



Load Test Creation Tutorial

Copyright Notice

COPYRIGHT© 2015 SOASTA Inc. This document is unpublished and the foregoing notice is affixed to protect SOASTA Inc. in the event of inadvertent publication. All rights reserved. No part of this document may be reproduced in any form, including photocopying or transmission electronically to any computer, without prior written consent of SOASTA Inc. The information contained in this document is confidential and proprietary to SOASTA Inc. and may not be used or disclosed except as expressly authorized in writing by SOASTA Inc. Copyright protection includes material generated from our software programs displayed on the screen, such as icons, screen displays, and the like.

Trademarks

Technologies described herein are either covered by existing patents or patent applications are in progress. All brand and product names are trademarks or registered trademarks of their respective holders and are hereby acknowledged.

Confidentiality

The information in this document is subject to change without notice. This document contains information that is confidential and proprietary to SOASTA Inc. and may not be copied, published, or disclosed to others, or used for any purposes other than review, without written authorization of an officer of SOASTA Inc. Submission of this document does not represent a commitment to implement any portion of this specification in the products of the submitters.

Content Warranty

The information in this document is subject to change without notice. THIS DOCUMENT IS PROVIDED "AS IS" AND SOASTA MAKES NO WARRANTY, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO ALL WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOASTA Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Version

15

Table of Contents

About SOASTA, Inc.	1
Audience	1
Terminology	1
What Will You Learn?	3
CloudLink Community	4
Prerequisites.....	4
<i>SOASTA Conductor</i>	4
Recording User Scenarios	6
Recording Guidelines	6
Recording Steps – Option 1 (Using the Recording Editor)	6
Inspecting and Filtering Your Recording in the Recording Editor	11
Create a Test Clip from a Recording.....	14
Recording Steps – Option 2.....	23
Inspecting Your Recording in the Clip Editor	28
Using the Session Template Package Wizard	31
Scanning a Clip for Values to Replace.....	33
Configure session templates (SubString Parser and XPath).....	35
Inspecting Applied Session Template Values.....	43
Pre-test Page validation	47
Verify Pages are grouped correctly	47
Rename Pages (and messages) as appropriate	50
Playing and Debugging a Simple Test	56
Result Details Dashboard	59
Verifying Test Parameterization	63
Test editing and parameterization	66
Session Template Wizard	66
Session Template Parameterization: Under the covers.....	74
Finishing Parameterization	77
<i>Find and Replace</i>	78
Creating User-Defined Validations	80

Using Scripts	84
Using Transactions.....	91
Using Seed Data to Parameterize data.....	94
Test Execution.....	100
Setting Play Mode for Load Testing.....	100
Adding System Dashboards to tests.....	101
Advanced exercise.....	103
Next Steps...	103
Pages versus Transactions.....	103
Building Dashboards	104
Dynamic Pages	104
If-Then-Else and Switch	104
Monitoring.....	105

About SOASTA, Inc.

SOASTA's mission is to ensure that today's Web applications and services perform in a high quality, scalable, and predictable manner. The company's product, SOASTA CloudTest™, is available as an on-demand service in the Cloud or as an appliance (virtual and hardware), and enables developers and testers to test and monitor their Web applications and services at an affordable price. SOASTA CloudTest supports Load/Performance, Functional, and Web UI/Ajax testing. SOASTA is privately held and headquartered in Mountain View, California.

SOASTA also offers two additional products currently; TouchTest and mPulse For more information about SOASTA, please visit www.soasta.com.

Audience

This tutorial is designed to be an introduction to SOASTA CloudTest for individuals familiar with other proprietary or open source tools for developing and executing load/performance tests.

In addition to some experience in load/performance testing, experience with web-based applications based on the underlying HTTP request/response message structure will be beneficial.

The tutorial target application site (SOASTAStore.com) has been selected to demonstrate fundamental features found in many public-facing applications. Subsequent exercises demonstrate the need to address more complex application development techniques.

Terminology

- **Message** – Messages are the HTTP requests or XML sent to a web application or web service. Messages are represented visually in SOASTA CloudTest as boxes in the icon view and URL values in the list view.
- **Script** - SOASTA CloudTest Scripts are written in JavaScript and allow you to add custom logic to a Test Clip. For example, a script that modifies the input data for a message based upon a set of status codes from the previous message response extracts data or validates responses. For people who have used other products, those products reference the test itself as a Script. In CloudTest, a script is a JavaScript snippet of code that gets included inside a Test Clip.
- **Page** – Pages are groups of HTTP requests that enable CloudTest to simulate the order and timing of browser traffic. Pages can be either “static” or “dynamic”. Static pages

use the same requests captured during the recording process while dynamic pages parse the HTML response to identify resources associated with that page.

- **Transaction** – A transaction is a larger unit of work in a test than a single page or single chain. A transaction is often associated with a portion of a business flow and may include, for example, hitting a page, logging in, and performing some action. A transaction can include chains, pages, messages, and think times (delays).
- **Test Clip** - The building block of testing in SOASTA CloudTest is the Test Clip. A Test Clip is a visual script that is composed of a series of sequenced events (an event can be, for example, an HTTP request or a SOAP message). Once built and debugged, these Test Clips can be reused for any type of testing: functional testing, load testing, performance testing and Web UI testing where appropriate.
- **Composition** - A Test Composition is a multi-track interface where test clips can be placed at different times and tracks to create complex testing scenarios. This interface is similar to music and movie digital creation tools. CloudTest provides a number of methods to create Test Compositions including by using the "Open in Test Composition" command from within the Clip Editor (the method used in the tutorial steps below). Test Compositions can also be created manually by creating a new Test Composition and then dragging and dropping Test Clips into a sequence based on when they should execute.
- **Target** - A Target points to the application, service, or web site that you want to test, including its location (hostname) and any authentication information necessary to successfully connect to the target. A Target also contains a specification of the information needed to properly format and send messages to the given service or site.
- **Monitor Server** - A Monitor Server defines which hardware resources will be monitored. Typical Monitor Servers are Linux/Windows/Mac, database, and application servers. SNMP devices are also Monitor Servers. A Monitor Server does not define the metrics to be captured; just 'how' they will be captured (IP Addresses, ports, authentication, etc.).
- **Monitor** - A Monitor defines the resources for which you will capture data to be used in analytics. This data typically includes statistics on resource usage, processes and activity. Monitored data appears in the Dashboard on the same timeline as test metrics such as response times and error rates. Reference: [SOASTA Monitoring FAQ in CloudLink](#).
- **Dashboard** - A Dashboard is where you view and analyze your test results in real-time. SOASTA CloudTest provides a wide variety of analytic widgets to fill your Dashboards. You can use one or more widgets to track your test in real time and capture results for post-test analysis.
- **Conductor** - SOASTA Conductor is an agent that runs on a client machine (for HTTP/HTTPs recording) or servers (for monitoring server metrics like CPU usage, for example). The Conductor runs on Windows, Linux, and Mac OS X machines.

- **Session Template** – A session template is essentially a parameterization rule that defines how to extract a name/value pair from a response, put it into a clip property, so that it can substitute the data dynamically at runtime. The task of replacing dynamic values in CloudTest can be done manually or by creating a session template that can be applied and re-applied to like test clips.
- **Session Template Package** - A Session Template Package contains a set of session templates that were originally created using a selected test clip. A package can be reused across multiple test clips and eliminates the need to repeatedly define how to extract a particular dynamic name/value pair.
- **Session Template Wizard** – This wizard is started within the Test Clip editor and allows the parameterization of one name/value pair at a time. At the completion of the wizard, the dynamic values are replaced (by custom properties) and a reusable session template is created.
- **Session Template Package Wizard** – This wizard is started from either SOASTA Central or within the Test Clip Editor and parameterizes multiple name/value pairs at the same time. It also creates a Session Template Package as its output.
- **Virtual Users** - Software programs or services that, when executed iteratively, generate traffic from their location to target servers or services, as would real users of these same servers or services. These are concurrent users.
- **Load Testing** - (also known as Performance Testing) is the use of a coordinated community of virtual users to place “load” on servers or services simulating some future workload for evaluation. Many different categories or terms for performance testing with virtual users are in current use such as benchmarks, endurance, stress tests, failover tests, etc. In the context of this tutorial, “Load Testing” and “Performance Testing” will be used interchangeably.
- **Think Times** - A think time is used in a test to simulate the natural pauses that a user has while accessing a web application. A user doesn’t just blast through a bunch of URLs. For example, it might take a user 10-15 seconds to enter their credentials when a login page is loaded in the browser. That 10-15 second period is a think time. Think times are critical to a test as it makes the test more real world by simulating what a real user would do.

What Will You Learn?

In this tutorial, the goal is simple: Create and run a test to demonstrate the fundamental features of SOASTA CloudTest and develop understanding of using these features to be successful. At the end of this tutorial is a list of next steps to continue your education on SOASTA CloudTest.

The tutorial general process is as follow:

- Create an HTTP/HTTPs recording of the web application test scenario, either in the Clip Editor or Recording Editor (both methods are documented)

- Inspect the recorded clip
- Use the Session Template Package Wizard to start the correlation process
- Verify/rename Pages and Page contents, as necessary
- Debug test playback
- Additional parameterization using the Session Template Wizard
- Manual parameterization
- Using Find and Replace
- Creating user-defined validations
- Using Scripts
- Using Transactions
- Including Seed Data in a test
- Run a load test using Load Play Mode

CloudLink Community

SOASTA has created a community site, CloudLink, which is a primary source of technical information about how to use CloudTest. The site contains a KnowledgeBase, training materials, documentation, and a forum to ask questions. Both SOASTA employees and community members contribute to the content on the site. Throughout this Tutorial, CloudLink is referenced as a source of additional information on a variety of topics. Please use CloudLink to expand your knowledge of CloudTest and as a way to interact directly with SOASTA employees and other CloudTest users.

Prerequisites

The following items are prerequisites for the successful completion of this tutorial.

SOASTA Conductor

HTTP(s) Message recording requires that the SOASTA Conductor agent be installed on a client machine on which you want to do recording. This machine can be running any of the common operating systems: Windows, Linux, or Mac OS X.

To download the Conductor, log into CloudTest with the URL and account credentials provided by your CloudTest administrator.

If you haven't yet installed the Conductor, you can do so from the SOASTA CloudTest > Resources page, download section. The appropriate Conductor installation file should be listed.

The screenshot shows a list of download items under the heading 'Downloads'. The items listed are: Browser Recorder Firefox Add-on, CloudTest Command-Line Client, SCommand QC Integration VB Script, CloudTest Jenkins/Hudson Plugin, Scripting Custom Module Java API, Conductor, and Other Operating Systems... The 'Conductor' item is highlighted with a blue background.

If you need to download an installation file for a different operating system, click the "Other Operating Systems..." link.

The screenshot shows the 'SOASTA Conductor Downloads' page. It features a logo of a blue lightning bolt inside a circle. Below the logo, there are three download links: 'Windows' (represented by the Windows logo), 'Mac Universal' (represented by the Mac logo), and 'Linux' (represented by the Linux penguin logo). At the bottom of the page, there is a link to the 'SOASTA Conductor Installation Guide'.

Additional SOASTA Conductor configuration instructions can be accessed from *Installing SOASTA Conductor* in CloudLink.

Recording User Scenarios

SOASTA CloudTest's HTTP(s) recording feature provides a read-only view of a given user scenario as captured from the browser's HTTP(s) traffic. CloudTest offers two HTTP(S) recording modes—the first is performed using the Recording Editor, while the second mode records directly into the Clip Editor. The steps to record using both modes are presented in the Option 1 and Option 2 steps below

Recording Guidelines

To record a user scenario, use one browser for CloudTest (recommended: Firefox or Chrome) and a separate browser (recommended: Internet Explorer, Chrome, or Safari) to perform user actions.

Before recording, it is recommended you do the following things:

- Clear the browser's cache and cookies before recording user scenarios.
- Close any additional programs that use the network (e.g. mail, calendar, and all social media applications), and any other applications that access the Internet to eliminate unnecessary HTTP traffic in the recording.
- For a cleaner recording, make the default home page for the second browser about:blank.

The example scenario in this tutorial uses a demonstration site, SOASTA Store (www.soastastore.com). This site has been created to demonstrate some of the most common topics for training purposes.

Recording Steps – Option 1 (Using the Recording Editor)

The following steps detail how to capture user actions via the Recording Editor.

1. Locate and launch your SOASTA Conductor application. The SOASTA Conductor opens with the status “running” and the Conductor name in display.

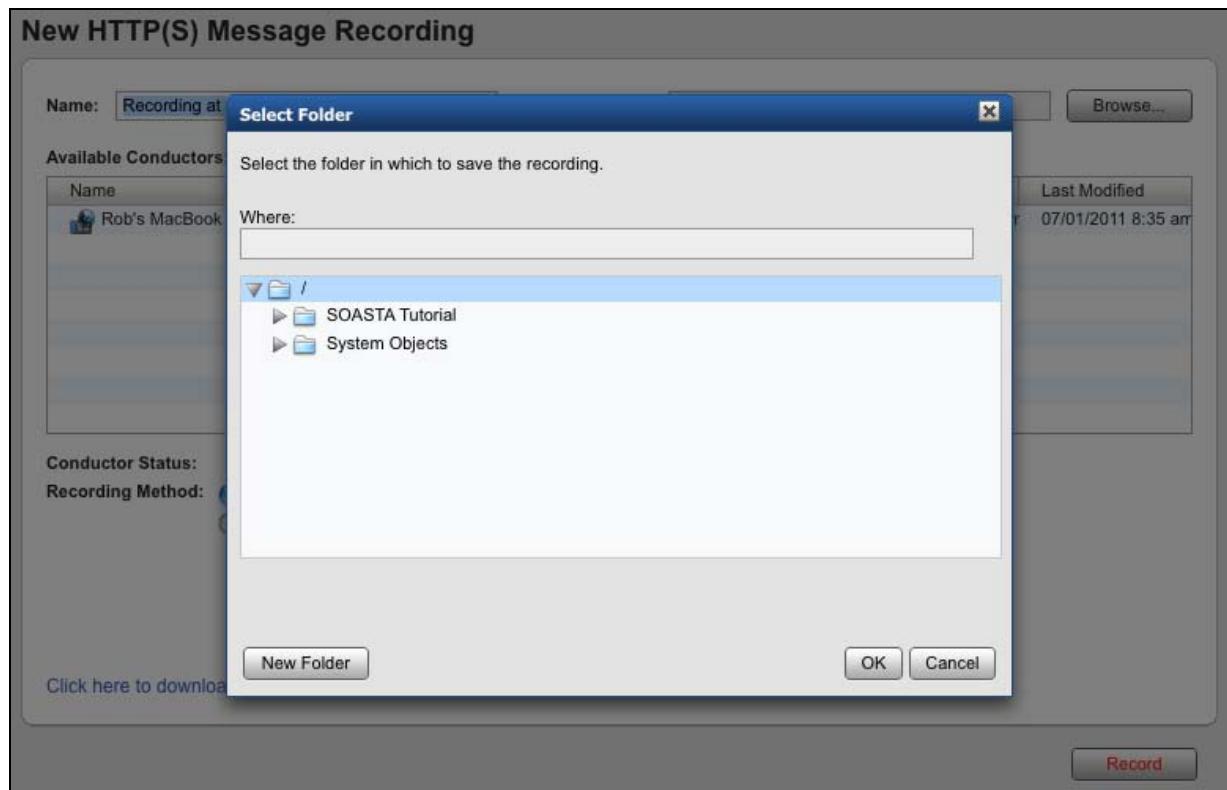
The Mac OSX version of Conductor is shown below.



In Windows, the Conductor will appear in the System Tray. Make sure the icon has a gold color. Otherwise, the Conductor is not connected to the CloudTest server.



2. Login to CloudTest using Firefox or Chrome. **For this tutorial, it is recommend that you use Firefox.**
3. Go to Central list > Recordings and then click New  in the toolbar.
The Recording Editor, New HTTP(S) Message Recording step appears.
4. Enter a name for the recording. For example, *SOASTA Store Post Forum Topic and Reply*.
5. Optionally, click Browse and specify a folder location for the recording using the Select Folder box. For example, *SOASTA Tutorial*. Click New Folder to create a folder if the desired folder doesn't exist.



- In the Available Conductors list, select the Conductor launched (in Step 1 above).

New HTTP(S) Message Recording

Name:	Recording at 10-14-2013 11:29:49 AM	Save Location:	/SOASTA Tutorial/	Browse...
Available Conductors:				
Name	Hostname	Created	Last Modified	
SOASTA_DOC Conductor	david.local		06/28/2013 10:53 pm	

Conductor Status: connected

Recording Method: HTTP(S) Recording
 Packet-level Recording

⚠ If you are going to use Firefox to generate the traffic to record, you need to restart Firefox after clicking the record button and again after clicking the stop recording button.

Alternatively you can manually specify Firefox's proxy (using the "HTTPS proxy listener port" as defined in the Conductor configuration panel).

Certificate: Client-side certificate required

[Click here to download the Conductor Installer](#)

Record

- Ensure that your SOASTA Conductor status is “connected”. If your SOASTA Conductor is not connected, the following steps will not work. Verify the Conductor connection settings and restart it.
- Accept the default Recording Method, *HTTP(S) Recording*.
 - HTTP(S) Recording is the default Recording Method in the Recording Editor and is required to record HTTPS sites. This enables an HTTP(S) proxy in SOASTA Conductor for recording.

When HTTP(S) Recording is checked, a reminder appears to notify you that HTTPS recording of the target site should be done from the Internet Explorer, Safari or Chrome browsers (not Firefox). If you must use Firefox, you will have to manually specify Firefox's proxy using the "HTTPS proxy listener port" setting in the Conductor configuration panel.

 - To record packet-level or HTTP traffic, click Packet-level Recording instead. This option is rarely used and should only be used if SOASTA support recommends it.
- Launch the second browser (e.g. Internet Explorer, Chrome, or Safari) and clear its cache and cookies. Close down any other unnecessary applications that use the

network. Set the second browser URL to “about:blank”. **For this tutorial, it is recommended that you use Chrome or Safari as the second browser.**

In the following example, a user will post a forum topic to the SOASTA Store site and reply to it. This tutorial will walk you through the process of creating this test.

4. Return to the Recording Editor in the first browser and click the Record button.



The Recording Editor appears and goes into listening mode.

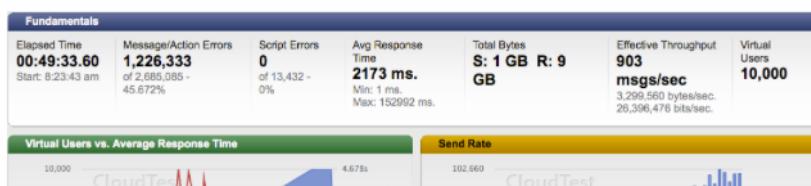
5. Open <http://www.soastastore.com/> in the second browser.

A screenshot of the SOASTA STORE website. The header features the "SOASTA STORE" logo. Below the header is a navigation bar with links: Home, Forum, Comment, Store, Checkout, Your Account, and About SOASTA. The main content area displays the title "Scalable, Affordable Testing of Web Apps" and a sub-section "Using SOASTAStore".

Scalable, Affordable Testing of Web Apps

Using SOASTAStore

This site is intended to provide a platform for learning SOASTA's CloudTest. It is comprised of a number of plug-ins that include a forum and a simple product store with a few books, DVDs and CDs for purchase. The site will change over time to illustrate new features in CloudTest.



Archives

- October 2013
- September 2013

Meta

- Register
- Log in

About You

- IP: 74.130.173.23
- Location: --
- Browser: Safari Generic
- OS: MacOSX

6. Messages start to appear in the Recording Editor.



7. In the second browser, click the "Forum" link.

A screenshot of the SOASTA STORE website showing the "Forum" section. The header includes the "SOASTA" logo and the "Forum" link. The main content area shows a list of forum topics.

The Forum page appears.

This screenshot shows the SOASTA Store Forum page. At the top, there is a message "You must be logged in to post" next to a user icon labeled "GUEST". To the right are "Login" and "Register" buttons. Below this is a navigation bar with "New/Recently Updated Topics" and "Select Forum" dropdown menus. A "Support" link and a "Group RSS" icon are also present. The main content area displays a table of forums. The first row shows "Website Issues" with 2 topics and 2 posts, last updated on June 22, 2011, at 12:37 pm by admin. The second row shows "Shipping Questions" with 0 topics and 0 posts. The table has columns for Forums, Last Post, Topics, and Posts.

Forums	Last Post	Topics	Posts
Website Issues	June 22, 2011 12:37 pm by admin	2	2
Shipping Questions	-	0	0

The SOASTA Store user accounts below will be used for this tutorial:

- testuser1 / soasta
 - testuser2 / soasta
 - testuser3 / soasta
 - testuser4 / soasta
 - testuser5 / soasta
8. Click the login button with the key icon to the left of it (**Login**) and enter the credentials for the first user in the list above.

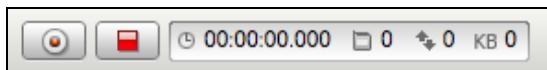
This screenshot shows the SOASTA Store Login page. It features a "You must be logged in to post" message and "Login" and "Register" buttons. The form fields include "Username" (testuser1) and "Password" (*****). There is also a "Remember me" checkbox and a "Login" button. Below the form are links for "Register?" and "Lost Your Password?".

9. Click the Login button.

This screenshot shows the SOASTA Store Member Profile page. It displays a message "Logged in as testuser1" and "Logout" and "Profile" buttons. It also shows "Members" and "Profile" icons. Below this is a "Last Visit: June 24, 2011" message and a set of small status icons. At the bottom is a navigation bar with "New/Recently Updated Topics" and "Select Forum" dropdown menus.

Confirm the login was successful. You will see a "Logged in as ..." message.

10. Click the “Product Questions” link in the “Pre-Sales” Forum Group.
11. Click the “Add a New Topic” link.
12. Enter “Post 1” as the Topic Name.
13. Enter “This is sample text for Post 1.” as the content for this post.
14. Click the “Post New Topic” button.
15. Click the “Reply to Post” link.
16. Enter “This is reply text for Post 1.” As the content for this post.
17. Click the “Post New Reply” button.
18. Click the “Logout” link at the top of the page.
19. Return to the first browser and then click the Stop button when you've finished your scenario.



Inspecting and Filtering Your Recording in the Recording Editor

1. Once the recording is finished, review the HTTP(S) messages that were recorded and verify that the recording looks accurate and complete. When scrolling left or right, messages are dynamically loaded into the Recording Editor.



Note: Message colors correspond to Content-Type. For a guide to content colors in CloudTest, see the CloudLink article, [Using the Recording Editor, Icon Colors for Content Type](#) section.

- **List**

You can inspect the recording in List view (after recording is stopped) to get a different view of what you've captured.



- Click any message in the recording to view detailed information about that HTTP request.

The screenshot shows the CloudTest application window. At the top, there's a menu bar with 'File', 'Central', 'SOASTA ...d Reply', and other options. Below the menu is a toolbar with icons for 'List', 'Convert to a Clip', and others. The main area has tabs for 'List' (selected), 'Central', and 'SOASTA ...d Reply'. A status bar at the bottom shows '00:03:00:388' and '111 KB 188,766'. The 'List' tab displays a table of 13 recorded messages. The 5th message is selected, showing its details. The 'Request Headers' section on the left lists various HTTP headers like Accept-Language, Host, Accept-Charset, Accept-Encoding, User-Agent, and Accept. The 'Response Headers' section on the right lists X-Pingback, Content-Length, Expires, Set-Cookie, Server, X-Powered-By, Cache-Control, and Pragma. There's also a dropdown menu for decoding binary encoded messages.

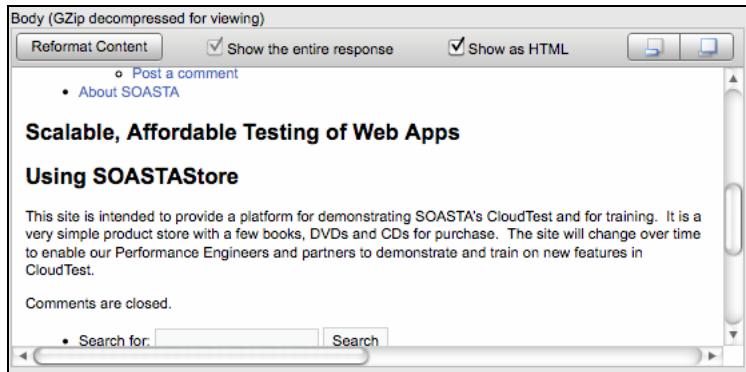
In the table of messages the following fields are shown:

- Sequence Number (Seq #)
- Response Code (for HTTP)
- Type (Content-Type)
- Method (GET, PUT, POST)
- Protocol
- Host Name
- URL (from root)
- Source IP
- Destination IP

The lower portion of this screen displays this information about each request:

- Request Headers
 - Request Body (content appears only for PUT and POST, generally)
 - Response Headers
 - Response Body
- Choose the first message that has / as its URL and check *Show as HTML*, which appears above the Response Body field (lower right).

The item appears as formatted HTML.



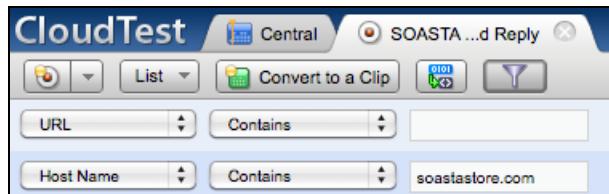
- **Filter**

Filtering allows you to eliminate messages that are not important for the test. Often this may include messages from Google Analytics or CDNs, depending on what is being tested. Removing these 3rd party requests will depend on the goals for the test.



You can filter by the following criteria by clicking the Filter icon: URL, Host Name, Destination IP, Source IP, Method, Duration, Count, Response Code, and Protocol.

1. In the Host Name filter row, enter *soastastore.com*.

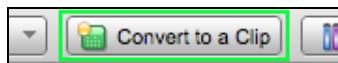


2. Click Apply to view the filter results.



Note: Filtering can be done by any of the drop-down fields. The messages in the recording can also be selected as a means of filtering the end result. For a complete discussion of filtering, see [Filtering Recordings](#) in CloudLink.

15. Once you have settled on a final set of filters, click Convert to a Clip. For this tutorial leave the single filter entered above as the only one applied to the recording.

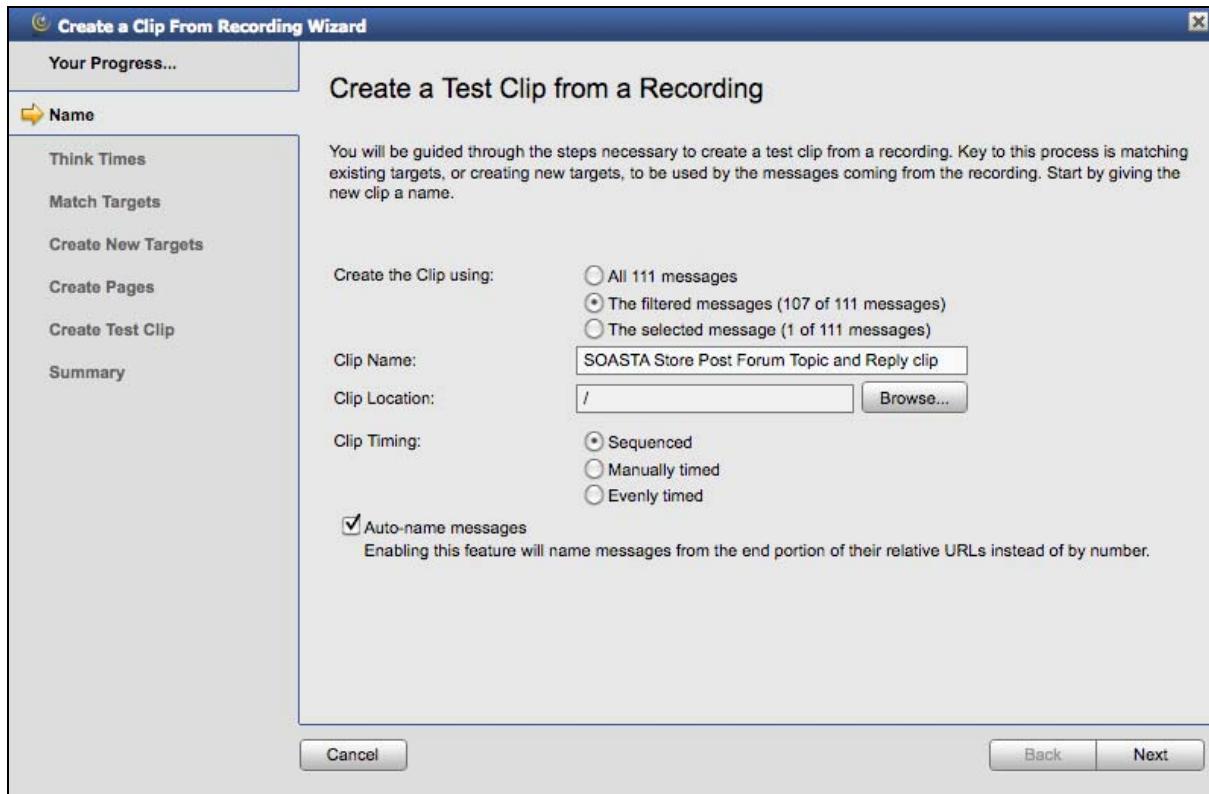


This will convert the read-only recording of the user scenario to an editable test. Test editing occurs in the Test Clip Editor after conversion is complete.

Create a Test Clip from a Recording

After you click Convert to a Clip, the Create a Test Clip from Recording box appears with the Name step selected to begin the conversion process.

Note: See [Test Clip](#) for more about this basic CloudTest building block.



7. Select the messages you want to include in the test clip. Leave “The filtered messages” selected for this tutorial.

You can choose to create the test clip using:

- All X messages (in the screenshot above there are 111). Note that the number of filtered messages as well as total messages may vary from that shown above.
- The filtered messages, if filters were used. In the screenshot above 107 of 111 messages were included by using the filter “Host Name > Contains > soastastore.com”. This second option is the default selection whenever filtering has been applied.
- The selected messages, if a manual selection of requests was made in the recording.

8. Select a clip timing type from among the following options. For this exercise, leave the default **Sequenced** selected.

- **Sequenced**

In a Sequenced clip, CloudTest sends request/page #1 in a test clip and request/page #2 isn't sent until a response to #1 is returned. This is the default selection for test clips created in SOASTA CloudTest and the clip type used by most CloudTest users.

- **Manually Timed**

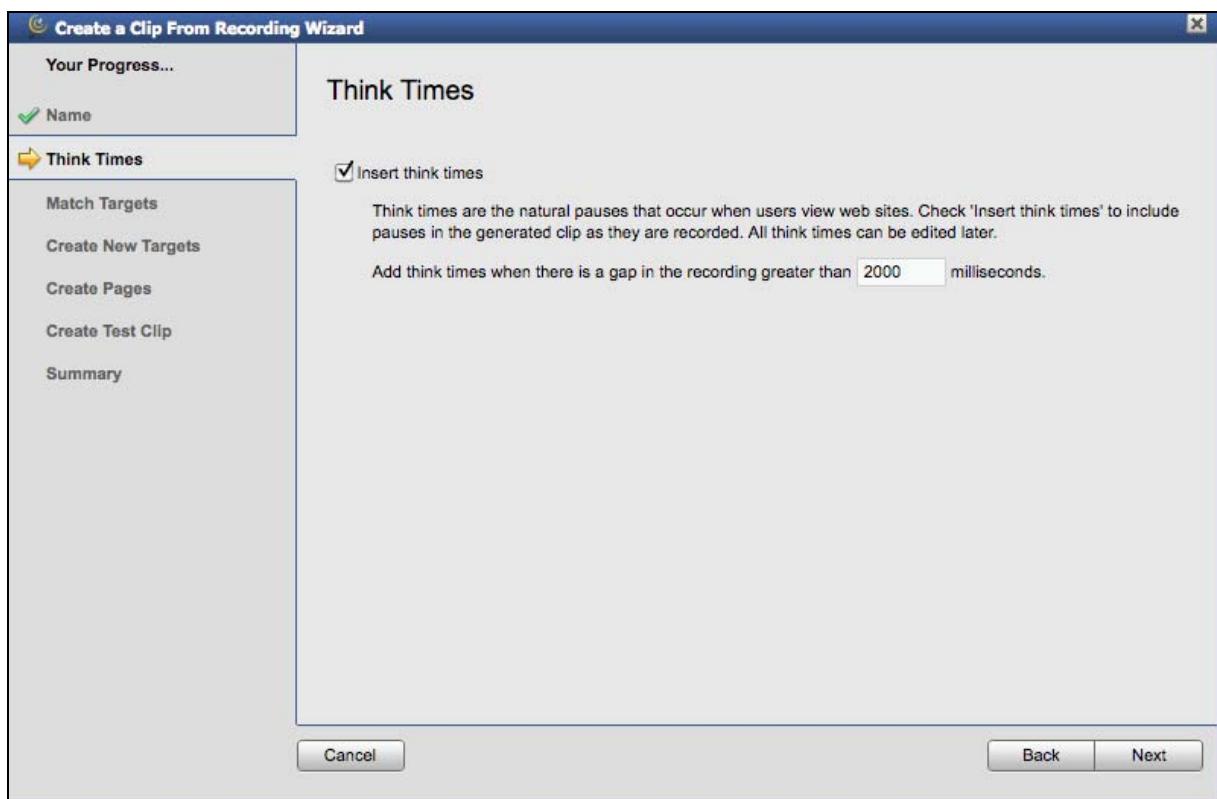
In a Manually Timed clip, you can specify down to the millisecond (ms) when a specific message is sent. Responses are not required before the next request is sent. For Manually Timed clips, an additional option to "Detect and insert bursts" appears. In the context of recording conversion, "bursting" is a series of messages that are probably due to a single event. For instance, a web page that loaded with 17 requests (for images, CSS, JavaScript files). Bursting isn't applicable for non-manual timing types. Most CloudTest users do not use manually timed clips.

- **Evenly Timed**

In an Evenly Timed clip, whenever the test clip is resized in the SOASTA CloudTest Composition Editor, the test clip is re-timed (1 per second, 2 per second). The default is one message sent per second. Responses are not required before the next request is sent. Most CloudTest users do not use evenly timed clips.

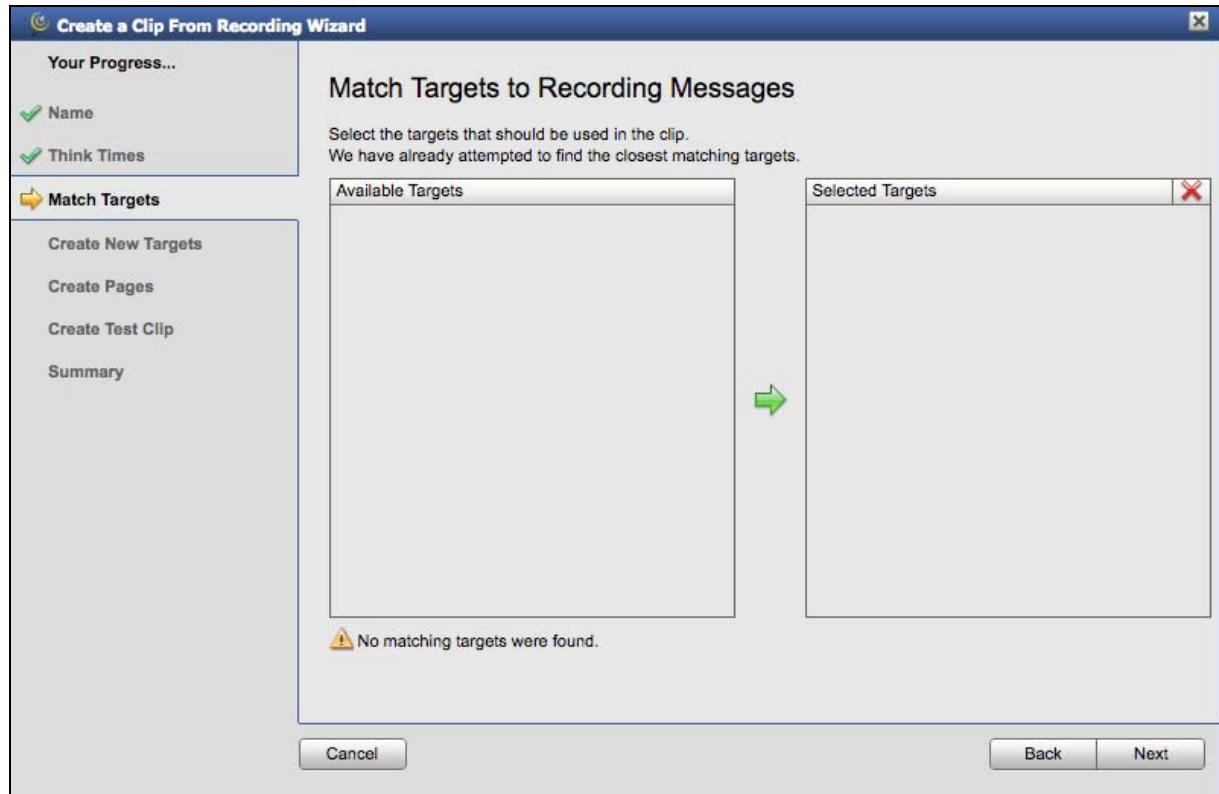
9. If you'd like, rename the test clip. This tutorial will assume the default name was used.
10. Leave the default *Auto-name messages* selected. This option will name the messages created during the recording-to-clip conversion using the end portion of their respective relative URLs, rather than by number (e.g. if this box is unchecked the messages will be sequentially named, Message1, Message2, etc.).

11. Click Next. The Think Times step appears. Think times simulate natural pauses while user's pause or read the contents of a page. CloudTest automatically detects the think times in the recorded user scenario. The default 2000ms threshold defined on this step indicates that any gap in time longer than 2000ms (e.g. 2 seconds) will be considered a think time and inserted into the converted test clip.
12. In most cases, the default of 2000ms is appropriate, so typically this value is left unchanged by CloudTest users. Note, however, that if the site you recorded was slow and there were gaps longer than 2 seconds while the page was loading, this will be considered a think time in the test clip—even though it technically wasn't a think time. You will be able to clean up these false think times during the test clip editing process. Additionally, the lengths of the default think times can be adjusted during test editing.



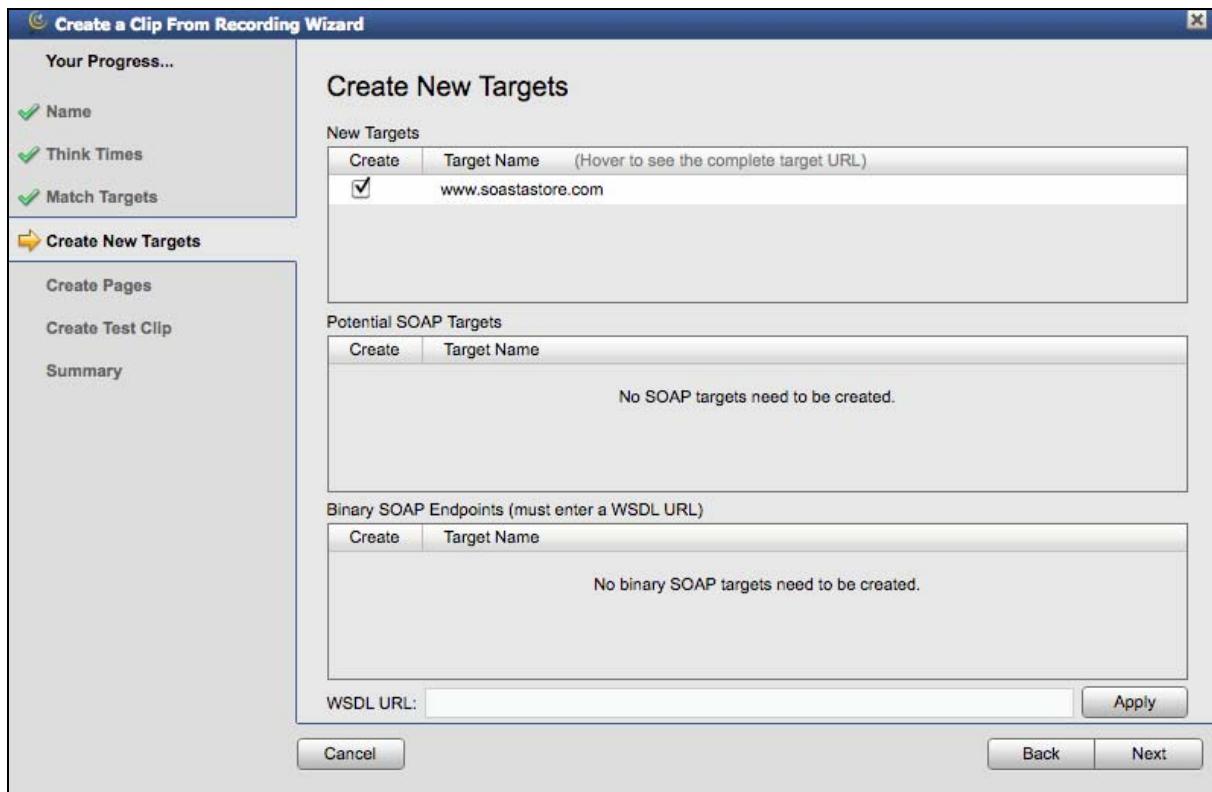
13. Click Next. The Match Targets to Recording Messages step appears. Targets are pointers to the hostnames being tested and control how CloudTest interacts with those hosts (number of concurrent connections, timeout values, etc.). For more about targets in CloudTest, see *Target* in CloudLink.

For new users, the Available Targets list will appear empty and a status message appears at the bottom of this list to indicate that no matches (with existing targets) were found.



- If any existing matching targets are detected, they are listed in the Selected Targets list on the right. If you want to remove any targets from the test, simply click the red X button next to the target you want removed and those targets will be filtered out when the test clip is created. If the targets do not exist, they will be created in the next step.

14. Click Next. The Create New Targets step appears. This step allows new targets to be reviewed one by one, and individually disabled (if a target is unchecked, the related target and its messages won't be created).



15. Click Next. The Create Pages step appears.

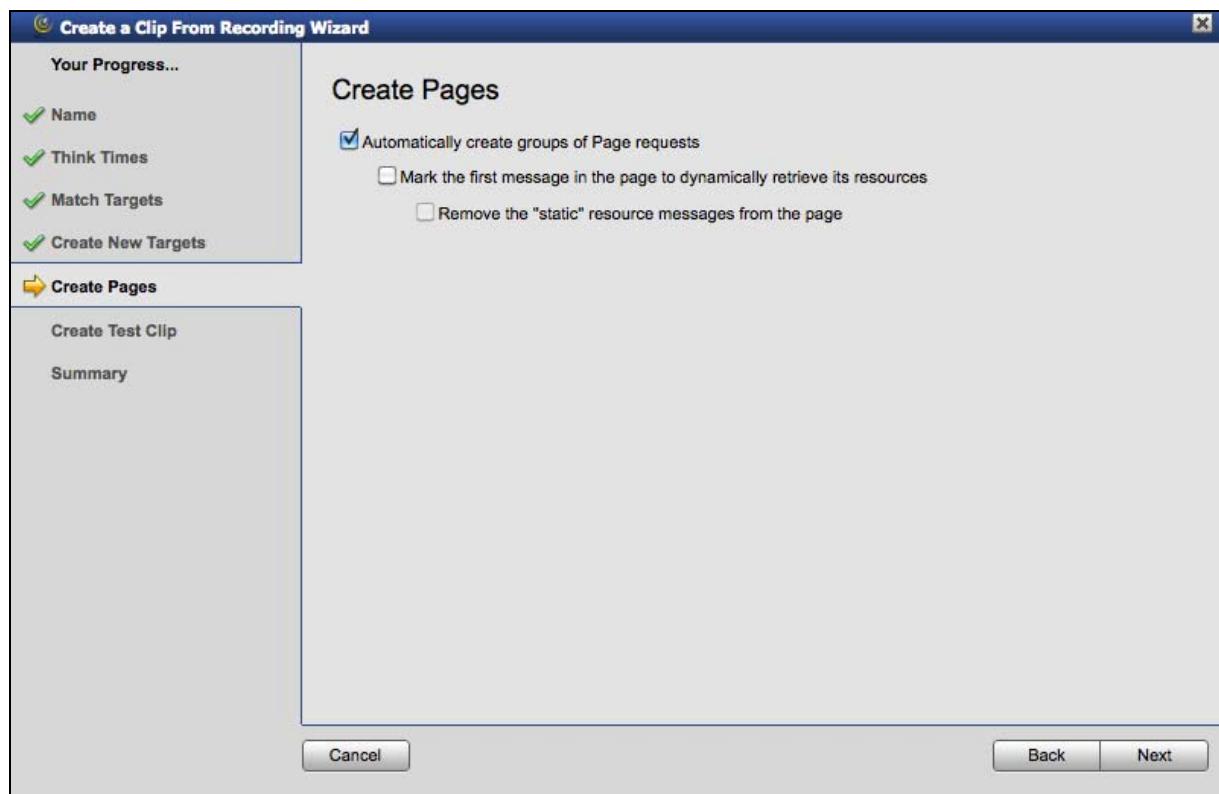
16. Pages are groups of HTTP requests that enable CloudTest to simulate the order and timing of browser traffic. Pages can be either “static” or “dynamic”.

Static pages use the same exact requests that were captured during the recording. Static pages are typically easier to get started with, as all the resources are listed and available for modification and/or deletion. Static pages are good to use on sites that are stable and not changing or if the tests clips have a limited life.

Dynamic pages make a request for the HTML page, parse the page, and dynamically extract the resources referenced in the page (e.g. CSS, images, JavaScript, etc.).

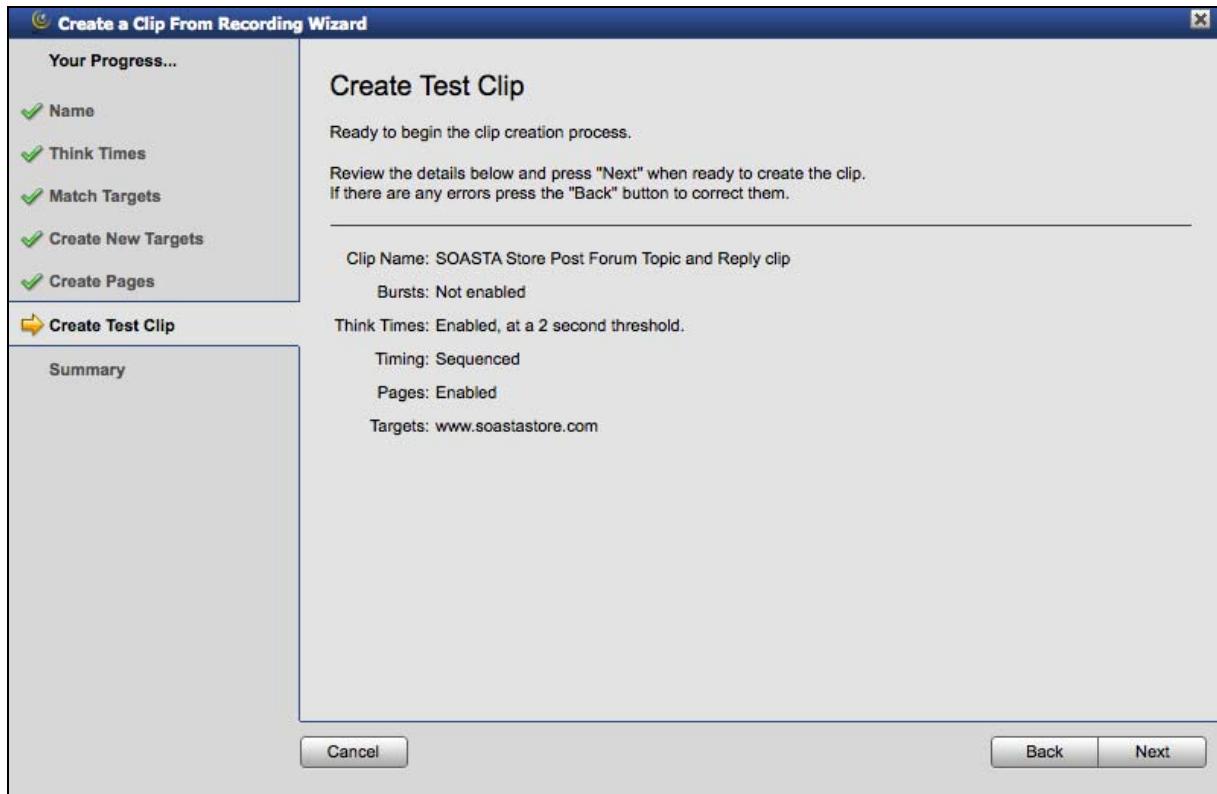
Dynamic pages typically take more time to setup because additional filters need to be applied to each page to remove any 3rd party resource requests that aren’t supposed to be played during the test (e.g. Google Analytics). Dynamic pages are good to use on sites that change frequently or when sites dynamically generate unique URLs each time the site is accessed (e.g. Drupal). Dynamic Pages are beyond the scope of this tutorial. Please reference CloudLink for more information.

For this tutorial, we will use static pages. Leave *Automatically create groups of Page requests* box checked. Click Next.



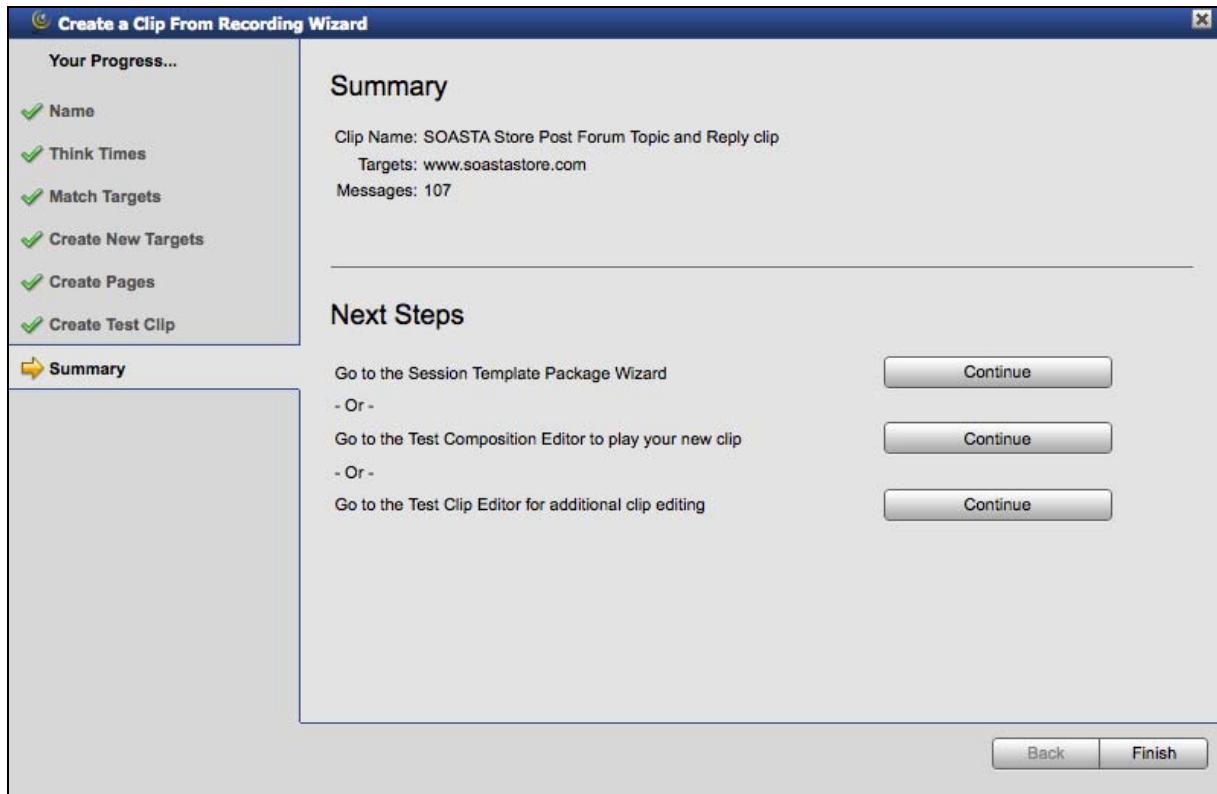
The Create Test Clip step appears with summary information about the new test clip selections.

In the example below, the clip name CloudTest will create is *SOASTA Store Post Forum Topic and Reply clip*, think times are enabled at a two second threshold, timing is sequenced, pages are enabled, and one target will be created.



17. Click Next. The status indicator appears to track clip creation progress.

18. The wizard creates the clip and then displays the Summary page. Click Go to the Test Clip Editor to perform additional clip editing tasks, such as user-specific parameterization.



19. The test clip is opened in the Clip Editor. The pages (and any other clip elements) are displayed in the center workspace, while the lower panel displays the default Messages/Actions tab with Available Targets and the Included Targets and Operations (not shown) lists to the right.



The first view shown above is called the "Icon" view. Clicking the plus icon on page will expand the Page so you can see the messages associated with that Page.



An alternative “List” view is available that shows each page request (or message) in a sequential view. The List view is a good way to edit tests as you have more visibility to the individual URLs the test is executing.. To change to this view, click the View drop down, which shows the word Icon by default, () and then change the view to “List” in the drop-down menu.



The clip will now look like the screenshot below. Expand the pages to see the requests within that page by clicking the grey triangle to the left of the page name. The HTML Document portion of the page is separated from the Resources.

Name	Operation	Parameter 1	Parameter 2	
soastastore.com				
HTML Document				
soastastore.com	get	/		HT
<input checked="" type="checkbox"/> Resources	<input type="checkbox"/> Static	<input type="checkbox"/> Dynamic		
thickbox.css	get	/wp-content/plugins/wp-e-com...e/wpsc-core/js/thickbox.css	ver=3.8.12.1.55f8cfa0d7	CS
compatibility.css	get	/wp-content/themes/2010-weaver/compatibility.css	ver=3.8.12.1.55f8cfa0d7	CS
style.css	get	/wp-content/themes/2010-weaver/style.css		CS
product_rater.css	get	/wp-content/plugins/wp-e-com...sc-core/s/product_rater.css	ver=3.8.12.1.55f8cfa0d7	CS
style-weaver.css	get	/wp-content/uploads/weaver-subthemes/style-weaver.css		CS
wpsc-default.css	get	/wp-content/themes/2010-weaver/wpsc-default.css	ver=3.8.12.1.55f8cfa0d7	CS
wp-e-commerce.js	get	/wp-content/plugins/wp-e-co...sc-core/s/wp-e-commerce.js	ver=3.8.12.1.55f8cfa0d7	JS
jquery.rating.js	get	/wp-content/plugins/wp-e-com...psc-core/s/jquery.rating.js	ver=3.8.12.1.55f8cfa0d7	JS
jquery.livequery.js	get	/wp-content/plugins/wp-e-com...admin/s/jquery.livequery.js	ver=1.0.3	JS
user.js	get	/wp-content/plugins/wp-e-commerce/wpsc-core/s/user.js	ver=3.8.12.1.55f8cfa0d7	JS
thickbox.js	get	/wp-content/plugins/wp-e-co...ce/wpsc-core/s/thickbox.js	ver=Instinct_e-commerce	JS
sf-forum.js	get	/wp-content/plugins/simple-fo...ces/script/forum/sf-forum.js	ver=3.5.2	JS
comment-reply.min.js	get	/wp-includes/s/comment-reply.min.js	ver=3.5.2	JS
jquery.js	get	/wp-includes/s/jquery/jquery.js	ver=1.8.3	JS
index.php	get	/index.php	wpsc_user_dynamic_js=true&ver=3.8.12.1.55f8cfa0d7	JS
soastastorelogo.jpg	get	/wp-content/uploads/2013/07/soastastorelogo.jpg		JP
dashboard.png	get	/wp-content/uploads/2011/06/dashboard.png		PN

The messages, most of which are within pages in the new test clip, were created from the recorded HTTP requests, while delays were calculated from the recorded think times.

Recording Steps – Option 2

HTTP Recording can be performed into the Clip Editor without the necessity to then convert the recording to a test clip. Messages are created inline in the Clip Editor. The following steps detail how to start a recording session in the Clip Editor.

1. Locate and launch your SOASTA Conductor application. The SOASTA Conductor opens with the status “running” and the Conductor name in display.

The Mac OSX version of Conductor is shown below.



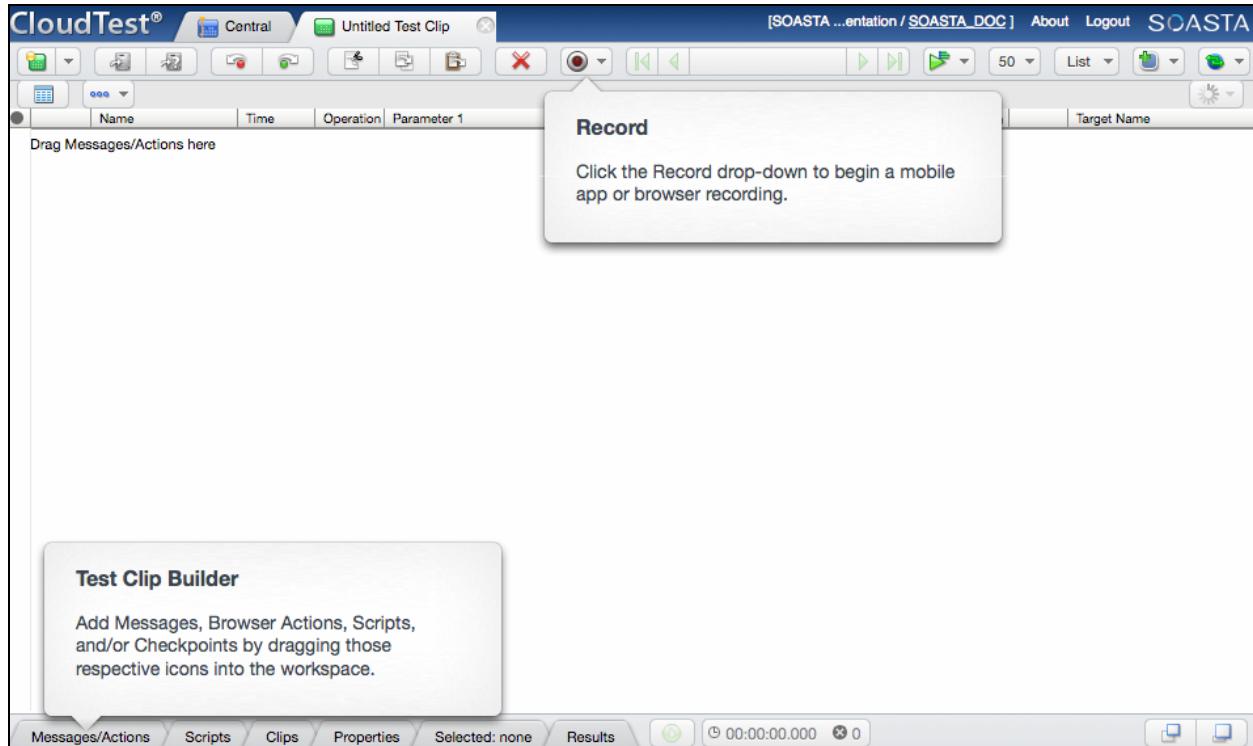
In Windows, the Conductor will appear in the System Tray. Make sure the icon has a gold color. Otherwise, the Conductor is not connected to the CloudTest server.



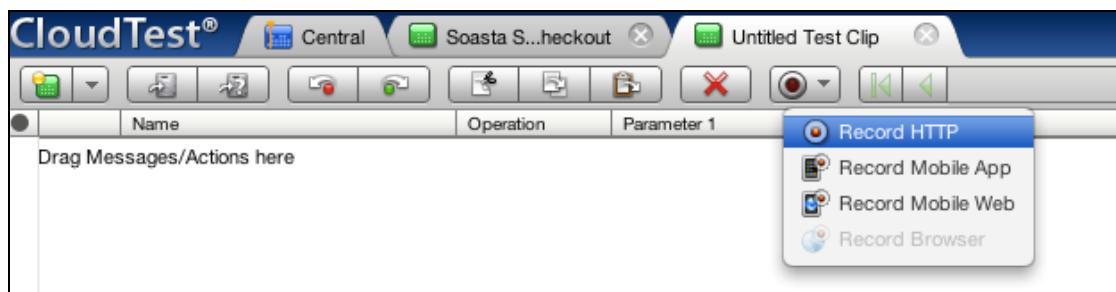
2. Login to CloudTest using Firefox or Chrome. **For this tutorial, it is recommended that you use Firefox.**

3. Go to Central list > Clip and then click New in the toolbar.

When you do so, the Clip Editor appears with a new Untitled Test Clip open. The Record button drop-down clearly indicates the button to use.



4. Click the Record drop-down button and choose the Record HTTP option.



5. The Choose a Conductor box appears. In the Available Conductors list, select the Conductor launched (in Step 1 above).

6. Ensure that your SOASTA Conductor status is “connected”. If your SOASTA Conductor is not connected, the following steps will not work. Verify the Conductor connection settings and restart it.
7. Click the Record button in the Recording Options box. The Clip Editor should now be in listening mode, and ready to record.



The Record button is bright red when Recording is active and the Status Indicator (top right) confirms that recording has started.



Note: When using this feature note that the clip is created as if it were recorded in the Recording Editor and then converted using the wizard conversion DEFAULTS. The underlying recording is still created and will be located in the Central > Recordings list.

1. Open <http://www.soastastore.com/> in the second browser.

The screenshot shows the SOASTA STORE website. At the top, there's a dark blue header with the "SOASTA STORE" logo. Below it is a navigation bar with links: Home, Forum, Comment, Store, Checkout, Your Account, and About SOASTA. The main content area has a title "Scalable, Affordable Testing of Web Apps" and a sub-section "Using SOASTAStore". A text block describes the site's purpose: "This site is intended to provide a platform for learning SOASTA's CloudTest. It is comprised of a number of plug-ins that include a forum and a simple product store with a few books, DVDs and CDs for purchase. The site will change over time to illustrate new features in CloudTest." To the right of the main content are several sidebar components: a search bar, an "Archives" section with links to October 2013 and September 2013, a "Meta" section with links to Register and Log in, and an "About You" section with IP information (74.130.173.23), location (---), browser (Safari Generic), and OS (MacOSX). At the bottom of the page is a "Fundamentals" summary table and a "Virtual Users vs. Average Response Time" chart.

Fundamentals						
Elapsed Time 00:49:33.60 Start: 8:23:43 am	Message/Action Errors 1,226,333 of 2,693,085 - 45.672%	Script Errors 0 of 13,432 - 0%	Avg Response Time 2173 ms. Min: 1 ms. Max: 152992 ms.	Total Bytes S: 1 GB R: 9 GB	Effective Throughput 903 msgs/sec 3,299,560 bytes/sec. 26,396,476 bits/sec.	Virtual Users 10,000

Virtual Users vs. Average Response Time

Send Rate

2. Messages start to appear in the Clip Editor.



3. In the second browser, click the "Forum" link.

The screenshot shows the SOASTA STORE website again, but this time the "Forum" link is highlighted in the navigation bar. The navigation bar includes links for Home, Forum, Comment, Store, and other account-related options.

The Forum page appears.

The SOASTA Store user accounts below are used for this tutorial:

- testuser1 / soasta
 - testuser2 / soasta
 - testuser3 / soasta
 - testuser4 / soasta
 - testuser5 / soasta
4. Click the Login button with the key icon to the left of it (**Login**) and enter the credentials for the first user in the list above.

5. Click the Login button.

Confirm the login was successful. You will see a "Logged in as ..." message.

6. Click the "Product Questions" link in the "Pre-Sales" Forum Group.

7. Click the “Add a New Topic” link.
8. Enter “Post 1” as the Topic Name.
9. Enter “This is sample text for Post 1.” as the content for this post.
10. Click the “Post New Topic” button.
11. Click the “Reply to Post” link.
12. Enter “This is reply text for Post 1.” As the content for this post.
13. Click the “Post New Reply” button.
14. Click the “Logout” link at the top of the page.
15. Return to the first browser, with the Clip Editor open, and click the Record button when you've finished your scenario. The Record button  should no longer be lit.

Inspecting Your Recording in the Clip Editor

1. Once the recording is finished, review the HTTP(S) messages that were recorded into the Clip Editor and verify that the recording looks accurate and complete. When scrolling left or right, messages are dynamically loaded into the Clip Editor.
2. The pages (and any other clip elements) are displayed in the center workspace, while the lower panel displays the default Messages/Actions tab with Available Targets and the Included Targets and Operations (not shown) lists to the right.

Note: Message colors correspond to Content-Type. For a guide to content colors in CloudTest, see the CloudLink article, [Using the Recording Editor, Icon Colors for Content Type](#) section.



Expand any collection in either Icon view or List view to view its contents.



An alternative “List” view is available that shows each page request (or message) in a sequential view. The List view is a good way to edit tests as you have more visibility to the individual URLs the test is executing.. To change to this view, click the button with “Icon” word on it () and click “List” in the drop-down.

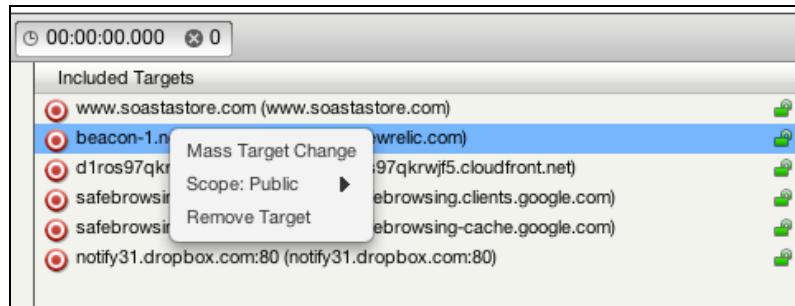


The clip will now look like the screenshot below. Expand the pages to see the requests within that page by clicking the grey triangle to the left of the page name. The HTML Document portion of the page is separated from the Resources.

CloudTest					
Central SOASTA ...d Reply SOASTA ...epli clip Robert Holcomb / SOA					
Name Type Target Name Operation					
soastastore.com					
	soastastore.com (7)	HTML	www.soastastore.com	get	/
	Resources	<input checked="" type="checkbox"/> Static <input type="checkbox"/> Dynamic			
	style.css	CSS	www.soastastore.com	get	/wp-content/themes/2010-weaver/style.css
	thickbox.css	CSS	www.soastastore.com	get	/wp-content/plugins/wp...c-core/js/thickbox.css ver=3.8.4.391083
	product_rater.css	CSS	www.soastastore.com	get	/wp-content/plugins/wp...e/j/s/product_rater.css ver=3.8.4.391083
	compatibility.css	CSS	www.soastastore.com	get	/wp-content/plugins/wp...heme/compatibility.css ver=3.8.4.391083
	l10n.js	JS	www.soastastore.com	get	/wp-includes/j/s/l10n.js ver=20101110
	wpsc-default.css	CSS	www.soastastore.com	get	/wp-content/themes/201 aver/wpsc-default.css ver=3.8.4.391083
	jquery.infieldlabel.min.js	JS	www.soastastore.com	get	/wp-content/plugins/wp...e/j/s/jquery.infieldlabel.min.js ver=3.8.4.391083
	ajax.js	JS	www.soastastore.com	get	/wp-content/plugins/wp...e/j/s/wpsc-core/j/s/ajax.js ver=3.8.4.391083
	wp-e-commerce.js	JS	www.soastastore.com	get	/wp-content/plugins/wp...e/j/s/wp-e-commerce.js ver=3.8.4.391083
	jquery.livequery.js	JS	www.soastastore.com	get	/wp-content/plugins/wp...e/j/s/jquery.livequery.js ver=1.0.3
	favicon.gif	GIF	www.soastastore.com	get	/wp-content/uploads/2011/06/favicon.gif
	Delay1				37597 ms.
	soastastore.com (1)				20252 ms.
	Delay2				

The messages, most of which are within pages in the new test clip, were created from the recorded HTTP requests, while delays were calculated from the recorded think times.

3. Click the Message/Actions tab in the lower left-hand corner. In the center column of this tab, all the targets created for this scenario are listed. Remove a target by right clicking on the target and selecting 'Remove Target'. All targets except 'www.soastastore.com' should be removed.



Most tests include unique session data (e.g. a login that creates a session or uses a token). The dynamic data values must be (replaced) before the test can succeed. The additional clip-editing steps in the next section will configure the test clip to extract dynamic session data, replace it with a custom clip property, and then use that property to place updated session data into the test at runtime.

Learn More

The example test clip above is comprised of three types of clip elements—pages, messages, and delays. Refer to the CloudLink, [KnowledgeBase](#), *Introduction* section for more about clip elements and other CloudTest basics.

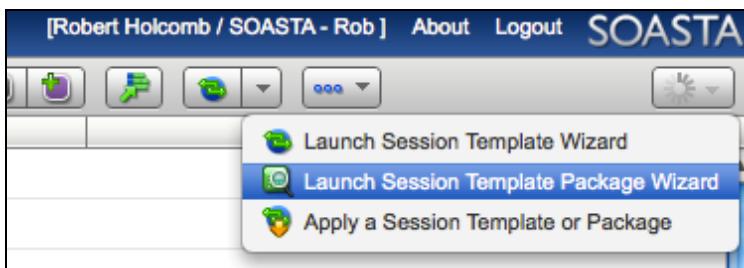
Using the Session Template Package Wizard

Now that a test clip has been created from the recording, the next step is to identify and substitute dynamic properties for those messages that require unique session data (e.g. parameters). In this section we will utilize the Session Template Package Wizard, which creates a re-usable “Package” that can be applied to other tests that use the same name/value pairs.

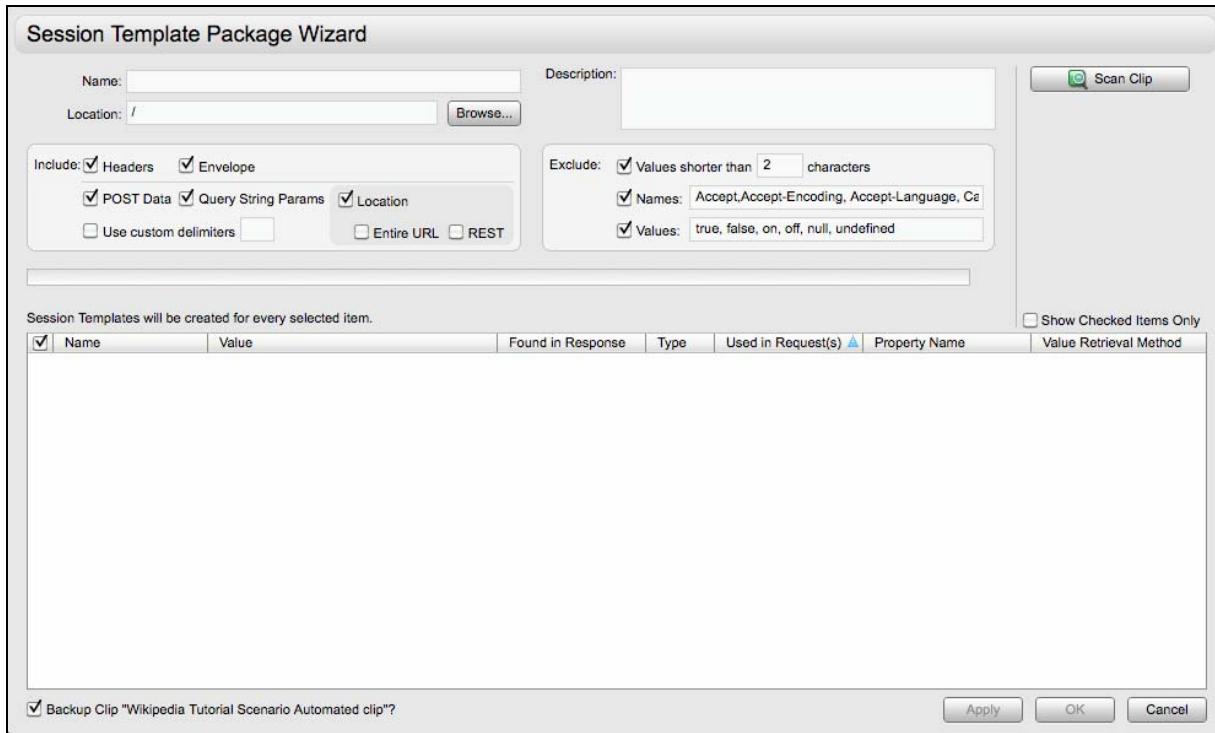
The Session Template Package Wizard uses a clip scanning method to search for potential name/value pairs that might be dynamic in the test. When the scan is complete, it provides a list of possible values to replace. Data can be extracted in the wizard using SubString parsing, XPath, by creating a new inline script, or by using an existing script.

The Session Template Package Wizard can be started from within the clip itself (as described below), or you can right-click on a clip in the clip list in SOASTA Central and choose to start the wizard from there. If starting the wizard from the clip list, make sure the clip is not open, as it will be considered locked if you don’t close it first.

1. Click the Session Template drop-down menu in the top right of the Clip Editor and then choose the Launch Session Template Package Wizard command.



When you do so, the Session Template Package Wizard appears.



1. First, give the session template package a name. For example, *SOASTA Store Session Template Package*.
2. Optionally, use the Browse button to specify the *SOASTA Tutorial* folder created above as the location where the Package should be saved.
3. Leave the Include/Exclude fields all checked. For more information about these fields, refer to *Session Template Packages* in CloudLink.
4. Leave the Backup Clip checkbox on the lower left checked. This ensures that the original test clip will be backed up.

Scanning a Clip for Values to Replace

In the following steps, the Clip Scanner will parse the clip in search of matching name/value pairs.

1. Click the Scan Clip button in the upper-right corner of the wizard.

The Scan Clip button becomes inactive and the scan begins. After the clip is scanned, the scan button automatically becomes re-enabled if the Include or Exclude options are changed. Scan progress is indicated by the green progress bar. Rows begin to appear as the scan progresses.

Once the scan is complete, the results are listed in the table below. Additionally, the green checkmark appears, and the table below is populated with name/value pairs the wizard thinks might need to be replaced. Some rows may appear before the scan completes.

Name	Value	Found in Response	Type	Used in Request(s)	Property Name	Value Retrieval Method
Referer	http://www.soastastore.com/	soastastore.com (7)	Header	style.css (+21)	Referer	SubString
ver	3.8.4.391083	soastastore.com (7)	HTML	thickbox.css (+9)	ver	SubString
ver	20101110	soastastore.com (7)	HTML	l10n.js	ver	SubString
ver	1.0.3	soastastore.com (7)	HTML	jquery.livequery.js	ver	SubString
ver	1.4.4	soastastore.com (7)	HTML	jquery.js	ver	SubString
ver	3.8.4391083	soastastore.com (7)	HTML	user.js	ver	SubString
ver	Instinct_e-commerce	soastastore.com (7)	HTML	thickbox.js	ver	SubString
ver	20090102	soastastore.com (7)	HTML	comment-reply.js	ver	SubString
ver	3.1.4	soastastore.com (7)	HTML	sf-forum.js (+5)	ver	SubString
page_id	167	soastastore.com (7)	HTML	soastast...com (8) (+2)	page_id	SubString
Referer	http://www.soastastore.com/?page_id=167	soastastore.com (8)	HTML	jquery.jqprint.js (+32)	Referer	XPath
log	testuser1	soastastore.com (8)	HTML	wp-login.php (2)	log	XPath
Origin	http://www.soastastore.com	soastastore.com (8)	Header	wp-login.php (2) (+2)	Origin	SubString

At this point, you will have to use your understanding of the application you are testing to determine which of these identified name/value pairs you need to replace in your test. Sometimes it is clear—sometimes it isn't. Experience will help you determine which of these values need to be replaced.

The scan of the clip normally returns lots of potential name/value pairs that aren't needed for the test to run correctly. If you aren't certain which values to replace, just do the ones you think are likely, play the test back and see what happens. If the test isn't working correctly, re-scan the test clip in the Session Template Package Wizard and replace additional name/value pairs.

The wizard allows you to configure multiple session templates at the same time. Just check the box to the left of the identified name/value pair and start the process for each session template. Let's walk through a couple examples that utilize the SubString parser and XPath to extract the dynamic values.

Configure session templates (SubString Parser and XPath)

Three name/value pairs from the response will be replaced in this section of the tutorial: `_wpnonce`, `forum-userform_addtopic`, and `forum-userform_addpost`. Let's look first at the `_wpnonce` name/value pair.

1. Locate the row for the name `_wpnonce`.

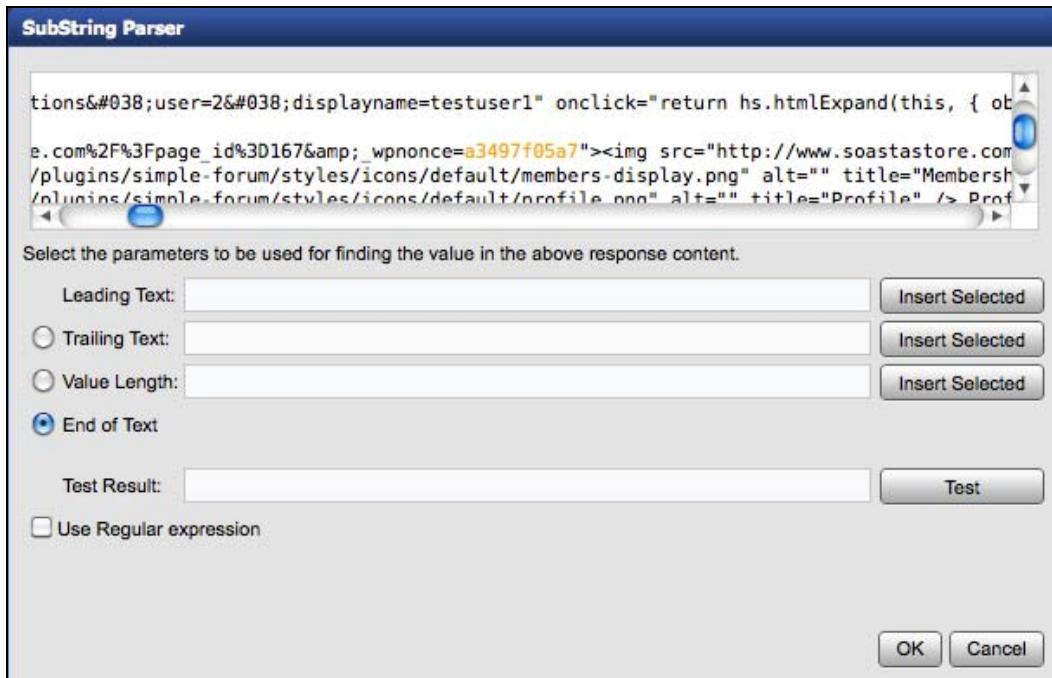
	Name	Value	Found in Response	Type	Used in Request(s)	Property Name	Value Retrieval Method
▶	forumslug	product-questions	soastastore.com (10)	HTML	index.php (4) (+1)	forumslug	XPath
▶	newtopicname	Post+1	soastastore.com (10)	HTML	index.php (4)	newtopicname	SubString
▶	sf_aahah	post	soastastore.com (10)	HTML	index.php (4) (+1)	sf_aahah	XPath
▶	page_id	167/product-questions/post-1/	index.php (4)	Header	soastastore.com (11) (+1)	page_id	SubString
▶	Referer	http://www.soastastore.com/questions/post-1/	soastastore.com (11)	HTML	sf-post.css (+20)	Referer	XPath
▶	topicslug	post-1	soastastore.com (11)	HTML	index.php (5)	topicslug	XPath
▶	forum-use...m_addpost	f783224f8a	soastastore.com (11)	HTML	index.php (5)	forum-userform_addpost	XPath
▶	postitem	This+is+reply+text+for+Post+1.	soastastore.com (10)	HTML	index.php (5)	postitem	SubString
▶	newpost	Saving+Post	soastastore.com (11)	HTML	index.php (5)	newpost	SubString
▶	If-Modified-Since	Fri, 01 Jul 2011 15:38:04 GMT	jcookie.js	Header	soastastore.com (12)	If-Modified-Since	SubString
▶	_wpnonce	a3497f05a7	soastastore.com (12)	HTML	wp-login.php (3)	_wpnonce	SubString
▶	action	logout	soastastore.com (12)	HTML	wp-login.php (3)	action	SubString
▶	redirect_to	http%3A%2F%2Fwww..3Fpage_id%3D167	soastastore.com (12)	HTML	wp-login.php (3)	redirect_to	SubString

Note the suggested Value Retrieval Method, which in this case is SubString. Depending on your application, you can choose either SubString, XPath, or script to extract the value from the response.

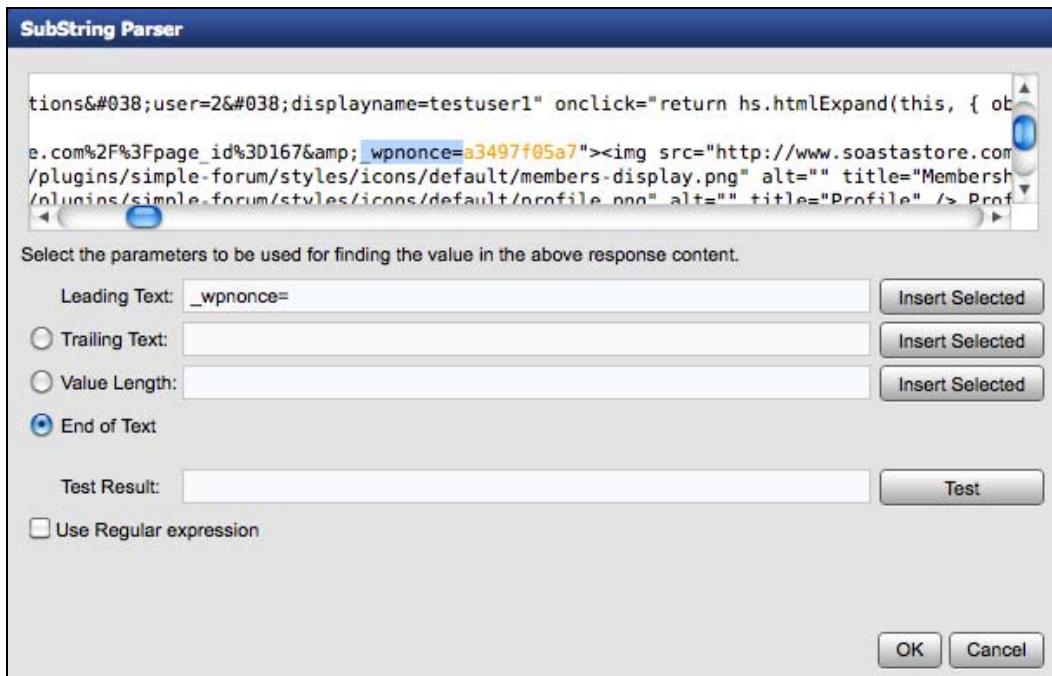
Note: XML Path Language, or XPath, is a method for selection within a document. Using XPath can be challenging in cases where a URL frequently changes, since the slightest change to the HTML code for that URL will render the XPath obsolete. SOASTA recommends using SubString whenever possible and XPath in cases where the page structure doesn't change much from test run to test run.

2. Click the checkbox to the left of the `_wpnonce` name to configure the SubString extraction method.

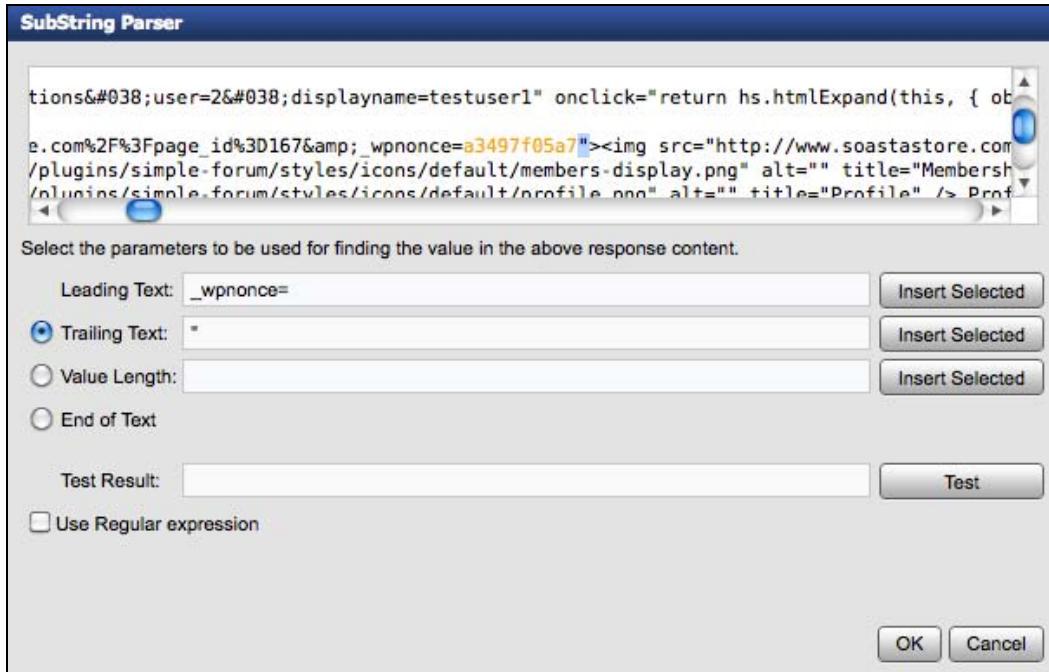
When you do so, the SubString Parser dialog box appears with the value of `_wpnonce` highlighted in orange in the top section.



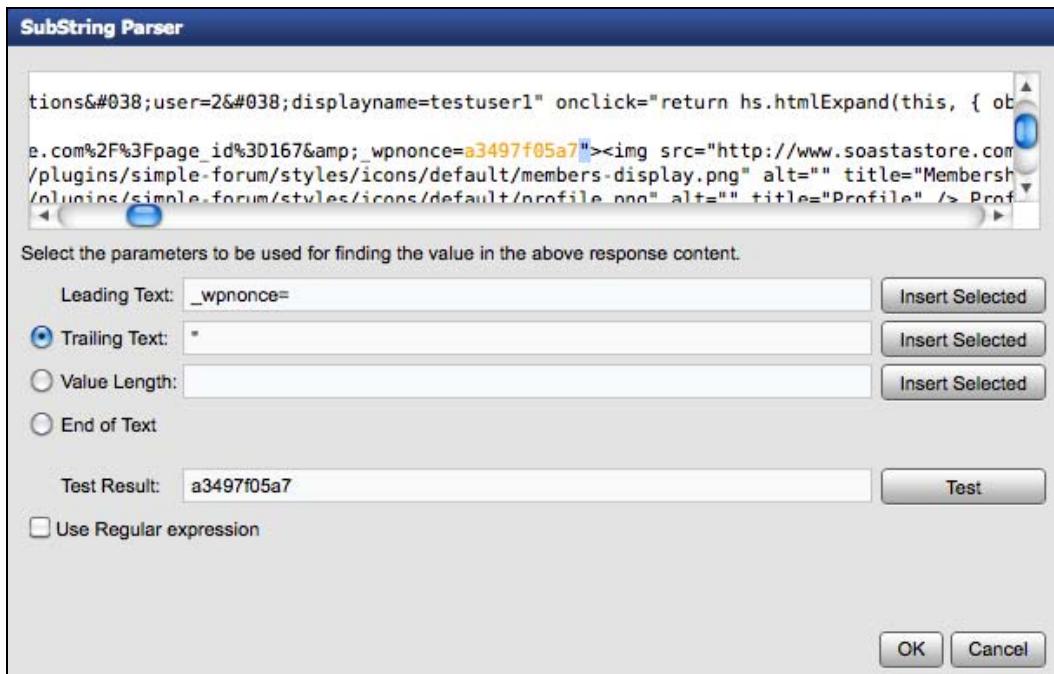
2. This interface is used to define how CloudTest identifies the substring that needs to be extracted. Use the mouse to highlight the leading text before the highlighted value and click the Insert Selected button on the Leading Text row. In this case, highlight `_wpnonce=`. This defines the string on the left side of the value that needs to be extracted.



3. Use the mouse to highlight the string to the right of the highlighted value that you want to extract. In this case, just use the double-quote string as the trailing text. Again, use your best judgment about what the trailing text should be, but try to use as short a string as possible. With the string highlighted, click the Insert Selected button on the Trailing Text row. Note that you can also type the leading and trailing text values directly into the text boxes—rather than highlighting them with the mouse.



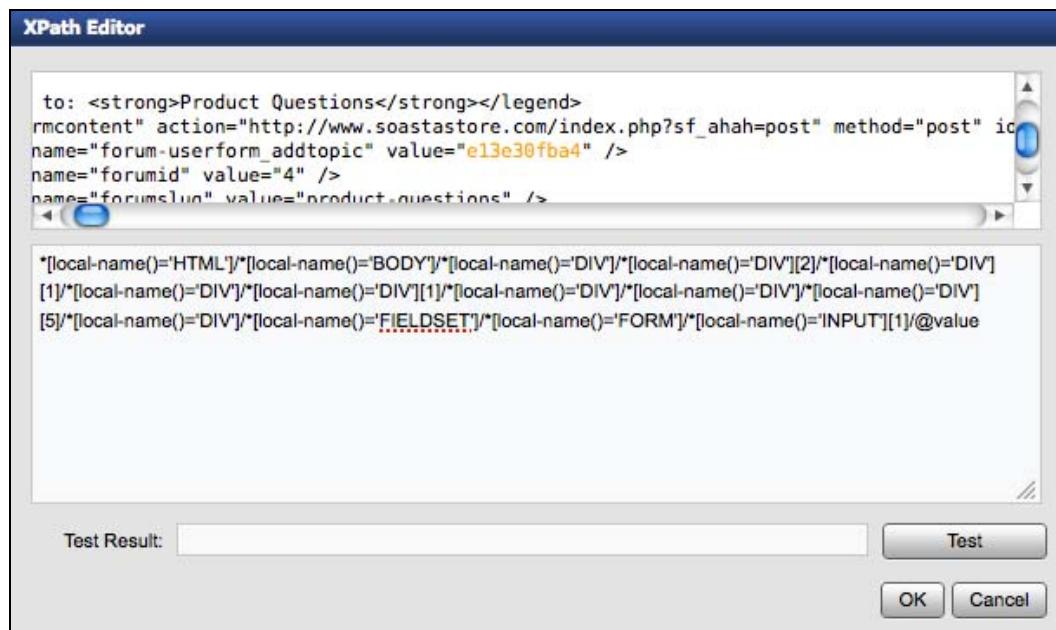
4. Click the Test button and ensure that the Test Result shown matches that of the highlighted token value as shown below.



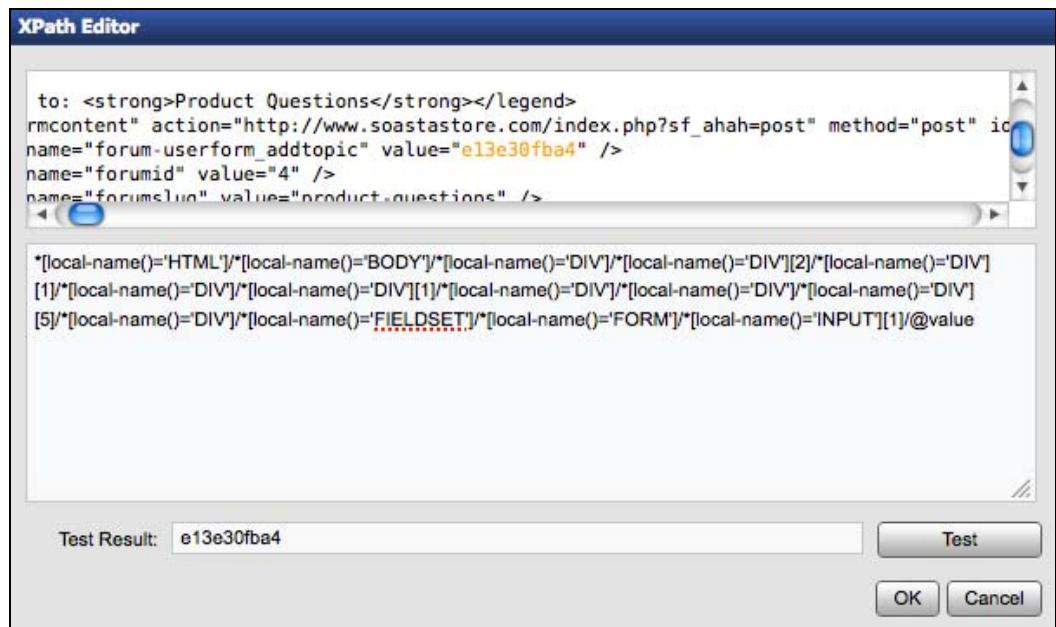
5. Click OK to exit the SubString Parser. Note that the `_wpnonce` row is now checked and that the specified value retrieval method is SubString Parser. The `_wpnonce` session template configuration is now complete.
6. Next, let's configure the `forum-userform_addtopic` session template. In this case, CloudTest has determined that this value can be extracted via XPath (as shown in the suggested Value Retrieval Method column).

	Name	Value	Found in Response	Type	Used in Request(s)	Property Name	Value Retrieval Method
▶	forum-user..._addtopic	e13e30fba4	soastastore.com (10)	HTML	index.php (4)	forum-userform_addtopic	XPath
▶	newtopic	Saving+Post	soastastore.com (10)	HTML	index.php (4)	newtopic	SubString
▶	forumslug	product-questions	soastastore.com (10)	HTML	index.php (4) (+1)	forumslug	XPath
▶	newtopicname	Post+1	soastastore.com (10)	HTML	index.php (4)	newtopicname	SubString
▶	sf_aahah	post	soastastore.com (10)	HTML	index.php (4) (+1)	sf_aahah	XPath
▶	page_id	167/product-questions/post-1/	index.php (4)	Header	soastast...om (11) (+1)	page_id	SubString
▶	Referer	http://www.soastasto...ct-questions/post-1/	soastastore.com (11)	HTML	sf-post.css (+20)	Referer	XPath
▶	topicslug	post-1	soastastore.com (11)	HTML	index.php (5)	topicslug	XPath
▶	forum-use...m_addpost	f783224f8a	soastastore.com (11)	HTML	index.php (5)	forum-userform_addpost	XPath
▶	postitem	This+is+reply+text+for+Post+1.	soastastore.com (10)	HTML	index.php (5)	postitem	SubString
▶	newpost	Saving+Post	soastastore.com (11)	HTML	index.php (5)	newpost	SubString
▶	If-Modified-Since	Fri, 01 Jul 2011 15:38:04 GMT	jcookie.js	Header	soastastore.com (12)	If-Modified-Since	SubString
▶	<input checked="" type="checkbox"/> _wpnonce	a3497f05a7	soastastore.com (12)	HTML	wp-login.php (3)	_wpnonce	SubString

7. Click the checkbox next to the *forum-userform_addtopic* name. Then, click the  icon next to XPath on the *forum-userform_addtopic* line. This will open the XPath Editor.

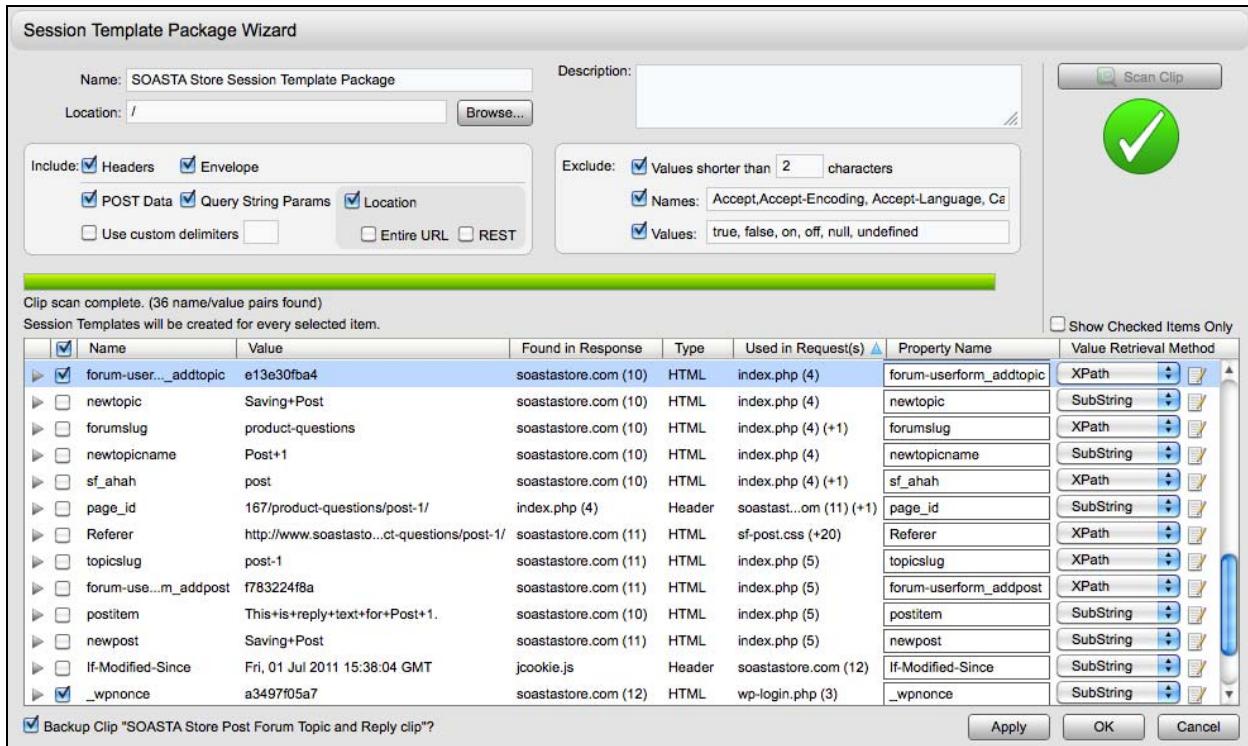


8. Click the Test button to confirm the XPath extraction works correctly. Verify the Test Result matches the highlighted value above.



9. Depending on your application, it may be necessary to update the default XPath that CloudTest generates. If you edit the XPath, make sure to click the Test button to confirm it is working correctly.

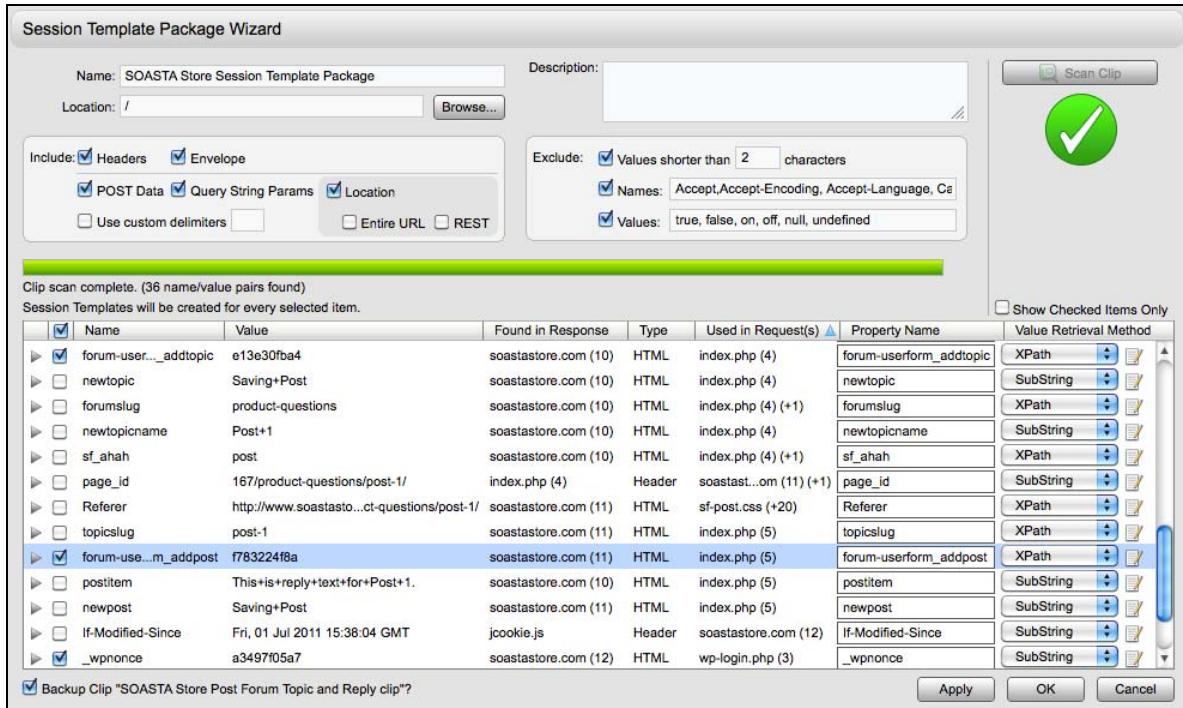
10. Click OK to close the XPath Editor.



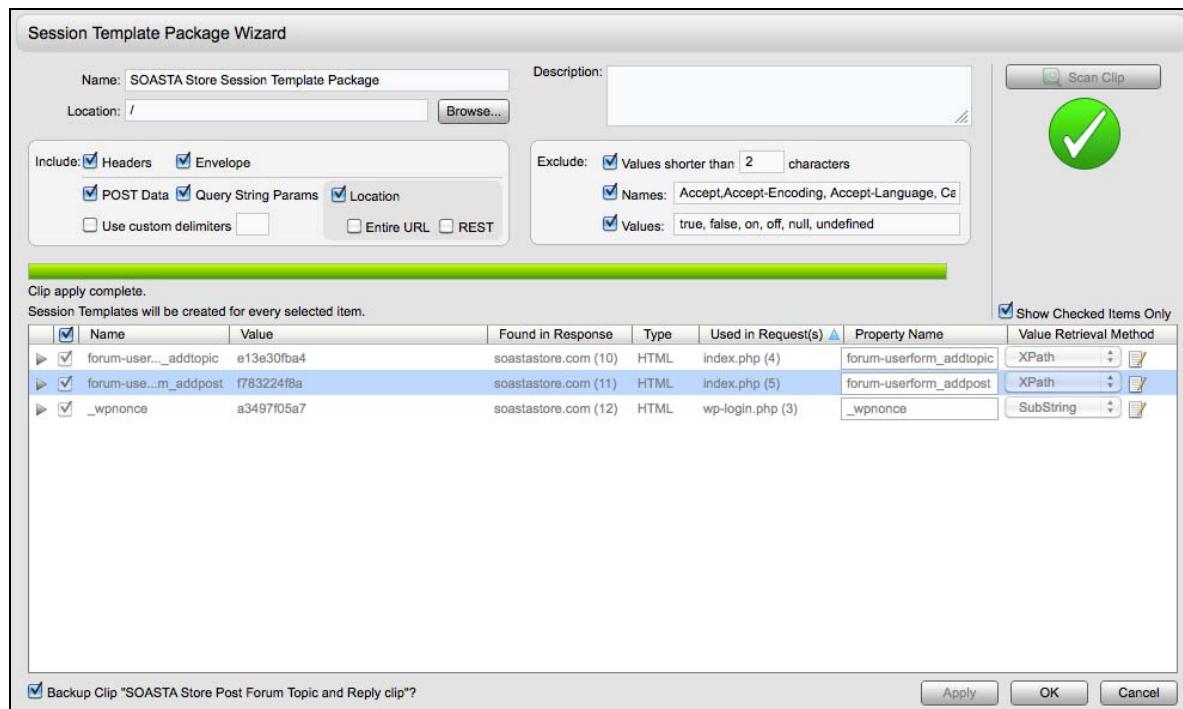
11. The *forum-userform_addtopic* session template is now configured.

12. Repeat the XPath configuration process for the *forum-userform_addpost* session template.

13. Once you have all three session templates configured, all should be checked in the Session Template Package Wizard interface.



14. Click the Apply button to apply the session templates to the test clip. This step creates the required properties (variables) in the test clip, inserts property sets to extract the dynamic data, and substitutes the property string into all the places where the previously-static strings were. The wizard filters the list to checked items only and applies those values. The green status bar indicates progress and once completed, the green checkmark icon appears.

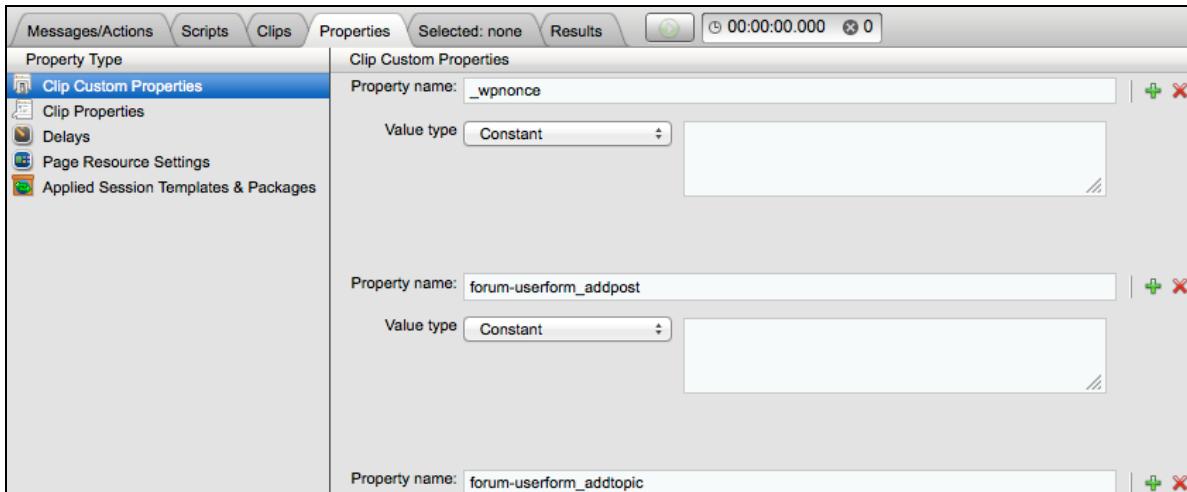


15. Click OK to exit the Session Template Package Wizard. This step creates the Session Template Package. For more information, search for “Session Template” in CloudLink.

Inspecting Applied Session Template Values

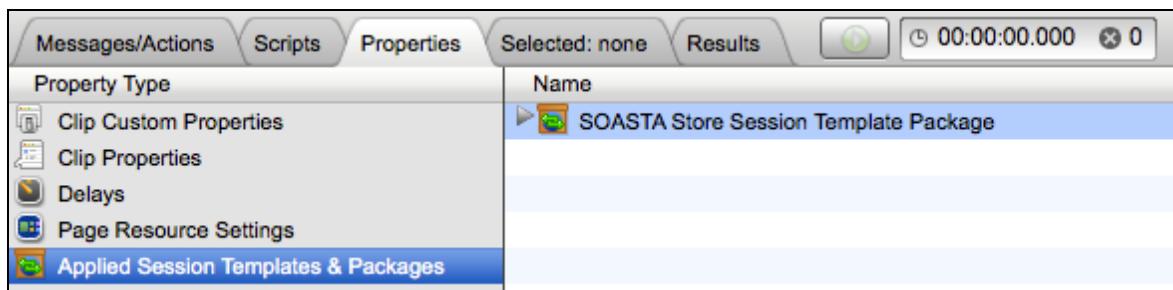
Now that the test clip has had the session templates applied, let's review the changes that were made to the clip by the wizard and inspect the applied template value.

1. Click the Properties tab in the lower panel of the Clip Editor

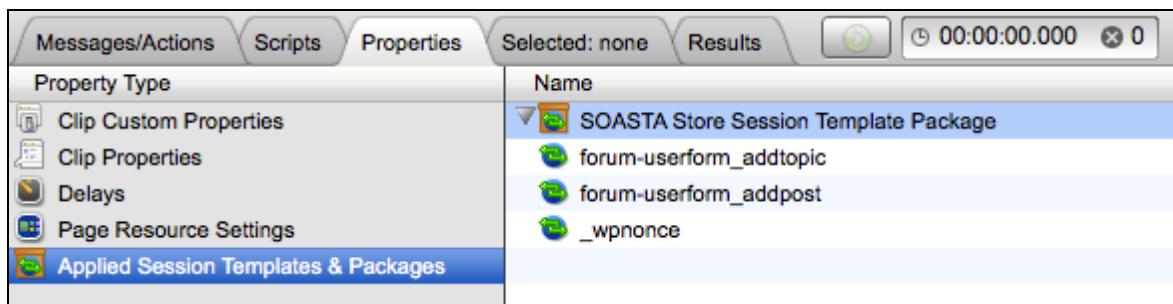


The custom properties have assumed the names extracted from the SOASTA Store site, which was identified in the Session Template Package Wizard above.

2. Click the Applied Session Templates & Packages node in the Property Type list.



3. The SOASTA Store Session Template Package appears in the Name list on the right. Click its arrow to expand the session templates within the package.



Note: The session template names correspond to the fields upon which they were based and also correspond to the automatically created custom clip properties.

- Click the *forum-userform_addtopic* item in the expanded session template.

Property Type	Name
Clip Custom Properties	SOASTA Store Session Template Package
Clip Properties	forum-userform_addtopic
Delays	forum-userform_addpost
Page Resource Settings	_wpnonce
Applied Session Templates & Packages	

- Review its Summary details on the far right pane. Note that the retrieval variable value method and XPath match the methods and selections used above.

Summary	
Applied by:	rholcomb@soasta.com on Sun Jul 03 15:24:42 PDT 2011
Retrieve variable value method:	XPATH *[local-name()='HTML']/*[local-name()='BODY']/*[local-name()='DIV']/*[local-name()='DIV'][2]/*[local-name()='DIV'][1]/*[local-name()='DIV']/*[local-name()='DIV'][1]/*[local-name()='DIV'][5]/*[local-name()='DIV']/*[local-name()='FIELDSET']/*[local-name()='FORM']/*[local-name()='INPUT'][1]/@value
Xpath:	
Message extracted from:	soastastore.com (10) <input type="button" value="Highlight"/>
Message(s) applied with ISSEs:	index.php (4)
Property Path:	forum-userform_addtopic
Path Type:	clip

The Summary identifies the first message from which the value was extracted and also any subsequent messages to which the value was applied by the use of a property path.

- Click the Highlight button. CloudTest changes the workspace display to show those messages that the value was extracted from and to which it was applied.

In List view, pages appear as shown below.

Name	Type	Target Name	Operation	URL
soastastore.com (10)	HTML	www.soastastore.com	get	/
index.php (4)	HTML	www.soastastore.com	post	/index.php

The first page shown above contains the message highlighted in green, *soastastore.com (10)*. This message is the “HTML Document” (or main message) of its page. This is the message where the *forum-userform_addtopic* value is extracted.

The second page shown above, *index.php*, contains the message highlighted in yellow, *index.php (4)*. This message is the HTML Document of its page.

- Double click the *index.php (4)* POST message in the expanded page to open it in the lower panel.

URL Values	/index.php
Query String	sf_ahah=post
Data	forum-userform_addtopic=%% prop : messageclip : forum-userform_addtopic forumslug=product-questions&newtopicname=Post+1&postItem=This+is+sample newtopic=Saving+Post

The message opens in a new tab in the lower panel (e.g. *Selected: index.php (4)*).

- Verify that the wizard replaced the appropriate string with the property path for *forum-userform_addtopic* (shown below).

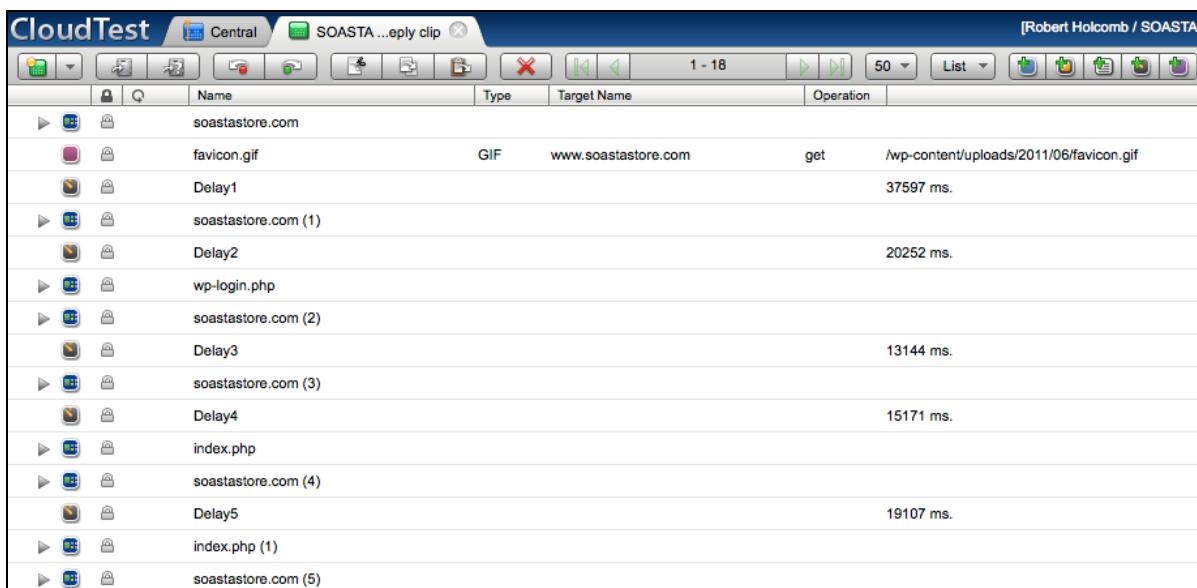
Operation: post	
URL Values	/index.php
Query String	sf_ahah=post
Data	forum-userform_addtopic={%% prop : messageclip : forum-userform_addtopic %%}&forumid=4&forumslug=product-questions&newtopicname=Post+1&postitem=This+is+sample+text+for+Post+1&topictags=&newtopic=Saving+Post

Pre-test Page validation

Verify Pages are grouped correctly

If the “Create Pages” checkbox was selected in the Convert to Clip wizard or in the Clip Editor, Recording Options box, the HTTP requests in your test clip have been grouped into Pages. Page creation is done based on the page referrer header and other rules. In most cases, the Pages pretty closely match the logical pages a user experiences while navigating the application being tested. In some applications that have non-standard referrer headers, unique AJAX calls, and other custom infrastructures, these pages may not match the way a user perceives pages in the application being tested. If that’s the case, you will need to modify the pages that were created during the conversion process.

The tutorial test clip looks something like this:



The screenshot shows the CloudTest interface with a list of recorded requests. The requests are grouped into pages, which are indicated by the page number in parentheses after the page name. The columns in the table are: Name, Type, Target Name, and Operation. The 'Name' column includes icons for each request, such as a lock, a padlock, and a file. The 'Type' column shows the request type (e.g., GET, POST). The 'Target Name' column shows the target URL. The 'Operation' column shows the duration of the request.

Name	Type	Target Name	Operation
soastastore.com			
favicon.gif	GIF	www.soastastore.com	get /wp-content/uploads/2011/06/favicon.gif
Delay1			37597 ms.
soastastore.com (1)			
Delay2			20252 ms.
wp-login.php			
soastastore.com (2)			
Delay3			13144 ms.
soastastore.com (3)			
Delay4			15171 ms.
index.php			
soastastore.com (4)			
Delay5			19107 ms.
index.php (1)			
soastastore.com (5)			

If you expand each page and look at the HTML Document and Resources, you will notice that the pages are generally pretty accurate in this recording. They correspond to each of the clicks and interactions we had when recording against SOASTA Store. However, one request, favicon.gif, might not be in a page (depending upon with browser you used to record). This should be included in the first page. We will include this request in the first page by dragging that request into the page as follows.

1. Expand the first page, likely named *soastastore.com*.

Name	Type	Target Name	Operation
soastastore.com			
HTML Document			
soastastore.com (7)	HTML	www.soastastore.com	get /
<input checked="" type="checkbox"/> Resources <input checked="" type="checkbox"/> Static <input type="checkbox"/> Dynamic			
style.css	CSS	www.soastastore.com	get /wp-content/themes/2010-weaver/style.css
thickbox.css	CSS	www.soastastore.com	get /wp-content/plugins/wp...c-core/js/thickbox.css
product_rater.css	CSS	www.soastastore.com	get /wp-content/plugins/wp...e/j/s/product_rater.css
compatibility.css	CSS	www.soastastore.com	get /wp-content/plugins/wp...heme/compatibility.css
l10n.js	JS	www.soastastore.com	get /wp-includes/js/l10n.js
wpsc-default.css	CSS	www.soastastore.com	get /wp-content/themes/201...aver/wpsc-default.css
jquery.infieldlabel.min.js	JS	www.soastastore.com	get /wp-content/plugins/wp-e...ery.infieldlabel.min.js
ajax.js	JS	www.soastastore.com	get /wp-content/plugins/wp...e/wpsc-core/js/ajax.js
wp-e-commerce.js	JS	www.soastastore.com	get /wp-content/plugins/wp...e/j/s/wp-e-commerce.js
jquery.livequery.js	JS	www.soastastore.com	get /wp-content/plugins/wp-e...js/jquery.livequery.js
favicon.gif	GIF	www.soastastore.com	get /wp-content/uploads/2011/06/favicon.gif

2. In the expanded Page area, you will see an “HTML Document” section and a “Resources” section. The HTML document is the main HTML page that is being called. You don’t want to put the resource there (nor can you—a page can only have one HTML document). We want to put the favicon.gif request inside the Resources section. To do this, scroll to the bottom of the Resources section. You will see the small vertical grey bar (scroll bar) on the left side of the Resources section.

Name	Type	Target Name	Operation
soastastore.com			
HTML Document			
soastastore.com (7)	HTML	www.soastastore.com	get /
<input checked="" type="checkbox"/> Resources <input checked="" type="checkbox"/> Static <input type="checkbox"/> Dynamic			
jquery.rating.js	JS	www.soastastore.com	get /wp-content/plugins/wp-e...ore/j/s/jquery.rating.js
user.js	JS	www.soastastore.com	get /wp-content/plugins/wp...e/wpsc-core/j/s/user.js
thickbox.js	JS	www.soastastore.com	get /wp-content/plugins/wp...psc-core/j/s/thickbox.js
index.php (2)	CSS	www.soastastore.com	get /index.php
comment-reply.js	JS	www.soastastore.com	get /wp-includes/j/s/comment-reply.js
sf-forum.js	JS	www.soastastore.com	get /wp-content/plugins/sim...cript/forum/sf-forum.js
style-weaver.css	CSS	www.soastastore.com	get /wp-content/uploads/w...emes/style-weaver.css
index.php (3)	JS	www.soastastore.com	get /index.php
soastastorelogo.jpg	JPEG	www.soastastore.com	get /wp-content/uploads/20...4/soastastorelogo.jpg
dashboard.png	PNG	www.soastastore.com	get /wp-content/uploads/2011/06/dashboard.png
favicon.gif	GIF	www.soastastore.com	get /wp-content/uploads/2011/06/favicon.gif

3. Once scrolled to the bottom of the Resources section, simply click the name of the message, favicon.gif, and drag it into the page as the last resource in the list. You may need to scroll down within the Resources section to confirm the move took place.

4. Click the Save icon on the toolbar to save the Test Clip.
5. If you collapse the first page by clicking the triangle icon, you will see that the favicon.gif request is now incorporated into the Page and no longer gets requested on its own.

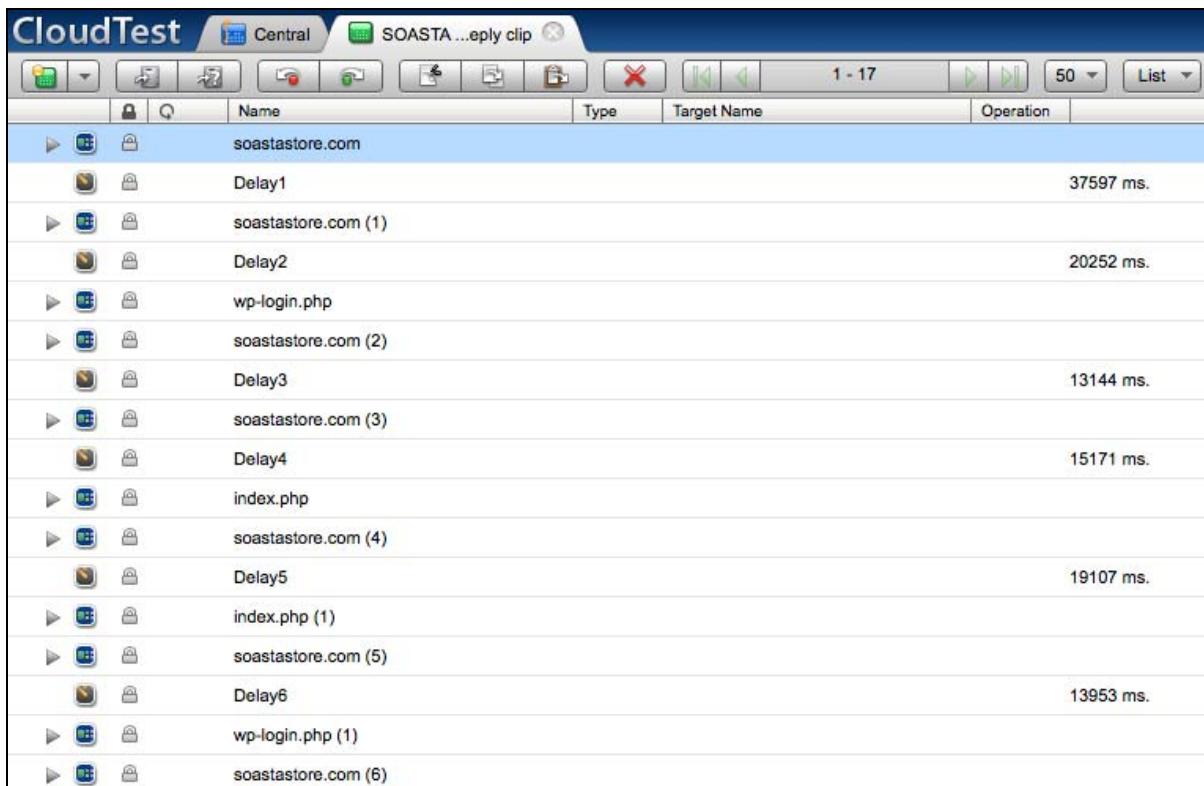
Name	Target Name	Time
Delay1		37597 ms.
soastastore.com (1)		
Delay2		20252 ms.
wp-login.php		
soastastore.com (2)		
Delay3		13144 ms.
soastastore.com (3)		

6. Depending on the factors listed above, you may need to do this zero or more times for each test clip.

Rename Pages (and messages) as appropriate

CloudTest auto-names each of the messages and pages in the test. This naming is done based on the last part of the URL.—whether this is done by the Convert to Clip wizard (Recording Option 1) or on-the-fly while doing HTTP recording in the Clip Editor (Recording Option 2).

In most cases, auto-naming matches the requests being made. In certain cases, however, it might be necessary to rename either pages or messages (or both) in a test. Although this does not change the functioning of the test clip, it will make the results easier to interpret. Take for example our tutorial test clip:



The screenshot shows the CloudTest interface with a toolbar at the top and a main window displaying a list of recorded test steps. The list includes various URLs and delays, with their execution times listed on the right.

Name	Type	Target Name	Operation
soastastore.com			
Delay1			37597 ms.
soastastore.com (1)			
Delay2			20252 ms.
wp-login.php			
soastastore.com (2)			
Delay3			13144 ms.
soastastore.com (3)			
Delay4			15171 ms.
index.php			
soastastore.com (4)			
Delay5			19107 ms.
index.php (1)			
soastastore.com (5)			
Delay6			13953 ms.
wp-login.php (1)			
soastastore.com (6)			

Since this sample application repeatedly uses the same URLs over and over again throughout the test (i.e. soastastore.com, wp-login.php, and index.php), the test clip ends up having similar Page names throughout the test. Anytime the same Page name is detected in the recording, a number is appended with an increasing integer value. So, the first time the test clip has a Page that references soatastore.com, it won't have any parenthesis. The second time it encounters that URL, it appends (1) as in *soastastore.com (1)*...and so on. In this case, the recording hit that page 8 times. It makes it harder to interpret the results if the appended numbers are the only differences between the Page names. Let's edit the Page names so they make more sense.

1. If you already know what the pages names should be, feel free to directly edit them in the UI (see below). To figure out what that page is doing, you will need to examine

the page more closely. Click the grey triangle to the left of the Page name to expand the first page of the Test Clip.

Name	Type	Target Name	Operation	Path
soastastore.com (7)	HTML	www.soastastore.com	get	/
style.css	CSS	www.soastastore.com	get	/wp-content/themes/2010-weaver/style.css
thickbox.css	CSS	www.soastastore.com	get	/wp-content/plugins/wp...c-core/js/thickbox.css
product_rater.css	CSS	www.soastastore.com	get	/wp-content/plugins/wp...e/s/product_rater.css
compatibility.css	CSS	www.soastastore.com	get	/wp-content/plugins/wp...heme/compatibility.css
l10n.js	JS	www.soastastore.com	get	/wp-includes/js/l10n.js
wpsc-default.css	CSS	www.soastastore.com	get	/wp-content/themes/201...aver/wpsc-default.css
jquery.infieldlabel.min.js	JS	www.soastastore.com	get	/wp-content/plugins/wp-e...ery.infieldlabel.min.js
ajax.js	JS	www.soastastore.com	get	/wp-content/plugins/wp...e/wpsc-core/js/ajax.js
wp-e-commerce.js	JS	www.soastastore.com	get	/wp-content/plugins/wp...e/js/wp-e-commerce.js
jquery.livequery.js	JS	www.soastastore.com	get	/wp-content/plugins/wp-e.../js/jquery.livequery.js

In this case, it is pretty clear the first page is just hitting [soastastore.com](http://www.soastastore.com) and getting the home page. So, we will want to change the name of this Page to "soastastore.com Home Page".

2. To change the name of the page, click on the text of the Page's current name (in this case "soastastore.com"). The Page name will become editable.

3. Enter the new name, *soastastore.com Home Page*, and hit the Enter/Return key on your keyboard to confirm the new name.

4. Save the test clip.
5. Although typically less important, you can also rename the individual messages within the page. If you want to change any message names, follow the same process as before: click the message name, enter the new name, and click the Enter/Return key on the keyboard to confirm the change.
6. Let's look at another Page where the appropriate name might not be as obvious.

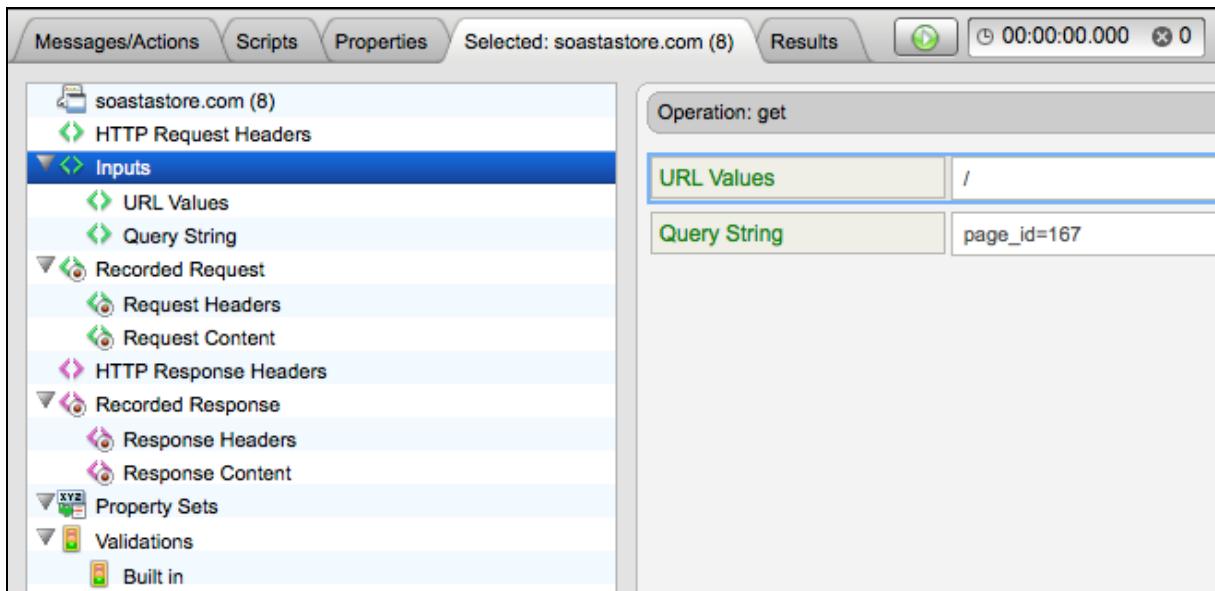
Name	Type	Target Name	Operation
soastastore.com Home Page			
Delay1			37597 ms.
soastastore.com (1)			
Delay2			20252 ms.
wp-login.php			
soastastore.com (2)			

In the image above, the test clip hits the soastastore.com Home Page, has a think time of almost 38 seconds, and then hits another page (currently named *soastastore.com (1)*). That page contains these requests:

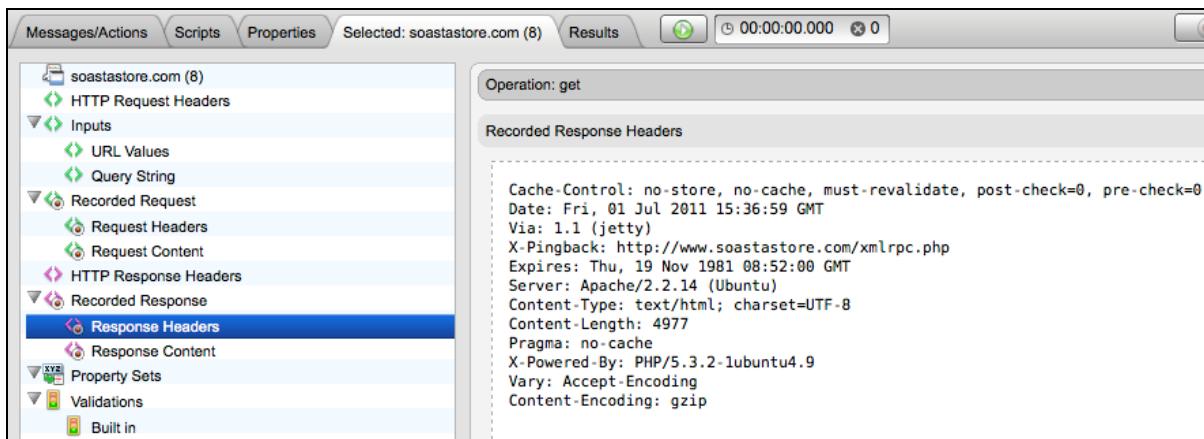
CloudTest [Central SOASTA ...epry clip]							[Robert Holcomb / SOASTA - Rob] About
Name	Type	Target Name	Operation				
soastastore.com Home Page							
Delay1				37597 ms.			
soastastore.com (1)							
HTML Document							
soastastore.com (8)	HTML	www.soastastore.com	get	/			page_id=167
Resources	Static	Dynamic					
jquery.jqprint.js	JS	www.soastastore.com	get	/wp-content/plugins/simple.../print/jquery.jqprint.js ver=3.1.4			
prettyCheckboxes.js	JS	www.soastastore.com	get	/wp-content/plugins/simple.../prettycheckboxes.js ver=3.1.4			
highslide.css	CSS	www.soastastore.com	get	/wp-content/plugins/simple.../highslide/highslide.css			
sf-tags.js	JS	www.soastastore.com	get	/wp-content/plugins/simple.../script/tags/sf-tags.js ver=3.1.4			
default.css	CSS	www.soastastore.com	get	/wp-content/plugins/simple.../skins/default/default.css			
highslide.js	JS	www.soastastore.com	get	/wp-content/plugins/simple.../highslide/highslide.js ver=3.1.4			
sf-framework.css	CSS	www.soastastore.com	get	/wp-content/plugins/simple.../skins/sf-framework.css			
login.png	PNG	www.soastastore.com	get	/wp-content/plugins/simple.../skins/default/login.png			
guestdefault.png	PNG	www.soastastore.com	get	/wp-content/forum-avatars/guestdefault.png			
register.png	PNG	www.soastastore.com	get	/wp-content/plugins/simple.../skins/default/register.png			

Note that the Page is accessing www.soastastore.com/page_id=167. That naming isn't really helpful unless you've memorized what page_id=167 means. It is possible that by referencing the test definition document you can deduce that this Page was a result of clicking the "Forum" link on the Exercises menu. For the sake of training, let's assume you couldn't figure that out from the test definition document. You will use other CloudTest features to figure it out.

- Double-click on the first request in the second Page. In these tutorial screenshots, this is the *soastastore.com (8)* HTML message in the *soastastore.com (1)* Page. This will open the properties pane at the bottom of the Clip Editor.



- Once you've double-clicked on a message, you have the ability to see the recorded request and response for that message. This is the request/response that occurred at the time of your recording. This is always available as a reference while you are building tests. Having this information available as part of the test clip is frequently used to compare the test you are editing to a "known good" execution of the test. That way you can compare headers, content, etc. to help troubleshoot any issues you might experience with a particular message. Click the 'Response Headers' item on the left.



- Many times this data is useful to determine if an HTTP 302 redirect occurred in the test—and what the redirect URL (location) might have been. This data is also useful when trying to determine if cookies were sent as part of this request and what their values might have been. In this case, the recorded response headers don't provide much information about what this Page is actually doing. So, click 'Response Content' to see the response content.

The response was 21651 characters long, showing the first 2000.

Show the entire response
 Show as HTML

```
<!DOCTYPE html>
<html dir="ltr" lang="en-US">
<head>
<meta charset="UTF-8" />
<title>Forum | SOASTA Store</title>
<link rel="profile" href="http://gmpg.org/xfn/11" />
<link rel="stylesheet" type="text/css" media="all" href="http://www.soastastore.com/wp-content/themes/2010-wear<br/><link rel="pingback" href="http://www.soastastore.com/xmlrpc.php" />
```

10. In this case, there is enough information in this response to indicate this page is the default page for the soastastore.com Forum. The highlighted text in the screenshot above shows the <title> tag in the HTML. Note that for performance reasons, only the first 2,000 characters in the response are shown in the window. If you want to see the full response, click the “Show the entire response” checkbox. Further, if you don’t want to read through the HTML code to determine what the page is, you can select the “Show as HTML” checkbox to render the HTML code as a web page that might be easier to review.

Wrap Text
The response was 29896 characters long, showing the first 2000.
[Show the entire response](#)
 Show as HTML

[Skip to content](#)

SOASTA STORE

[Skip to content](#)

11. You can then scroll within this window to review the recorded response content. If the window area is too small to view the whole page, click the expand pane button () on the right side of the properties pane bar. You can use the restore pane button () to return to the original size.

The screenshot shows a web proxy tool interface with the following details:

- Toolbar:** Messages/Actions, Scripts, Clips, Properties, Selected: soastastore.com, Results, a green play button icon, a timer showing 00:00:00.000, and a stop/cancel icon.
- Left Panel (Tree View):**
 - soastastore.com
 - HTTP Request Headers
 - Inputs
 - URL Values
 - Query String
 - Recorded Request
 - Request Headers
 - Request Content
 - HTTP Response Headers
 - Recorded Response
 - Response Headers
 - Response Content
 - Property Sets
 - Validations
 - Built in
- Right Panel (Operation: get):**
 - Recorded Response Content:**
 - Wrap Text
 - The response was 30022 characters long, showing the first 2000.
 - Show the entire response
 - Show as HTML
 - [Skip to content](#)
 - SOASTA STORE**
 - [Skip to content](#)
 - Navigation Menu:**
 - Home
 - Forum
 - Comment
 - Store
 - Checkout
 - Your Account
 - About SOASTA
 - Forum**
 - User Authentication:** You must be logged in to post. Login Register
 - User Status:** GUEST

12. Now that we know this Page is loading the default Forum page, we can name it accordingly. This tutorial has used the name *Forum Default Page (not logged in)*.

13. Go through each of the remaining pages, determine what that page is doing, and name them accordingly. There should be about 9 Pages you need to rename. To best follow this tutorial, the following 4 pages need to have specific names:

- (a) Page that contains the POST for posting a new forum topic: *Post new forum topic*
- (b) Page immediately after the Page above: *Load newly posted forum topic*)
- (c) Page that contains the POST for posting the username: *Post credentials*
- (d) Page immediately after the Page above: *Forum Default Page (logged in)*

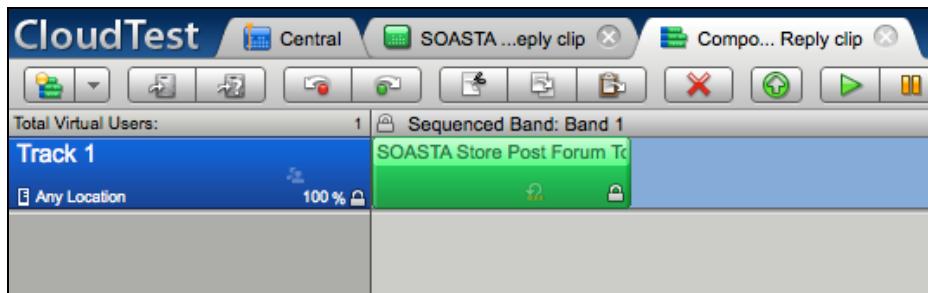
Playing and Debugging a Simple Test

As a general guideline, it is best to play and debug a test as changes are made to the Test Clip. This simplifies debugging tasks. Debugging a test is an iterative process and this tutorial walks you through a common test creation/debugging process.

1. Click the Open in Test Composition icon on the Clip Editor toolbar. For more information, see *Test Composition* in CloudLink.



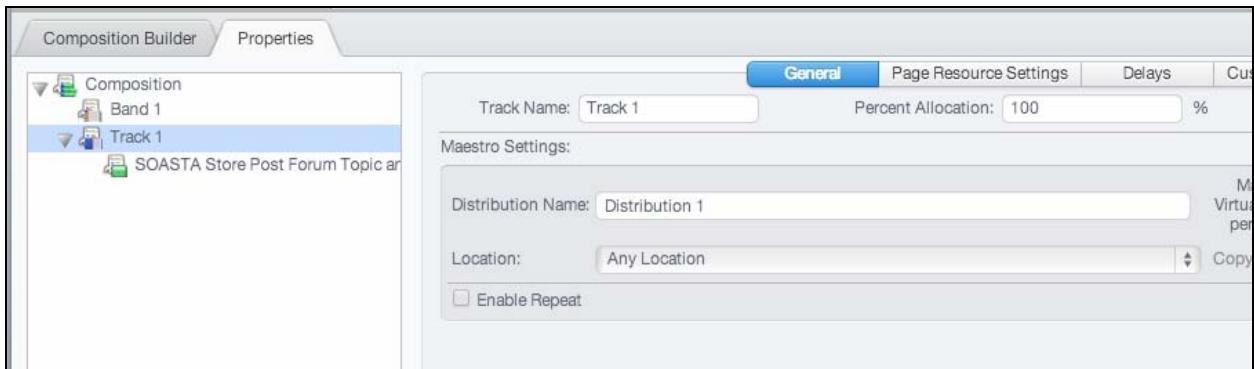
The test is added to Track 1 of a composition with the name *Composition for SOASTA Store Post Forum Topic and Reply clip*, which opens in the Composition Editor (in a new tab).



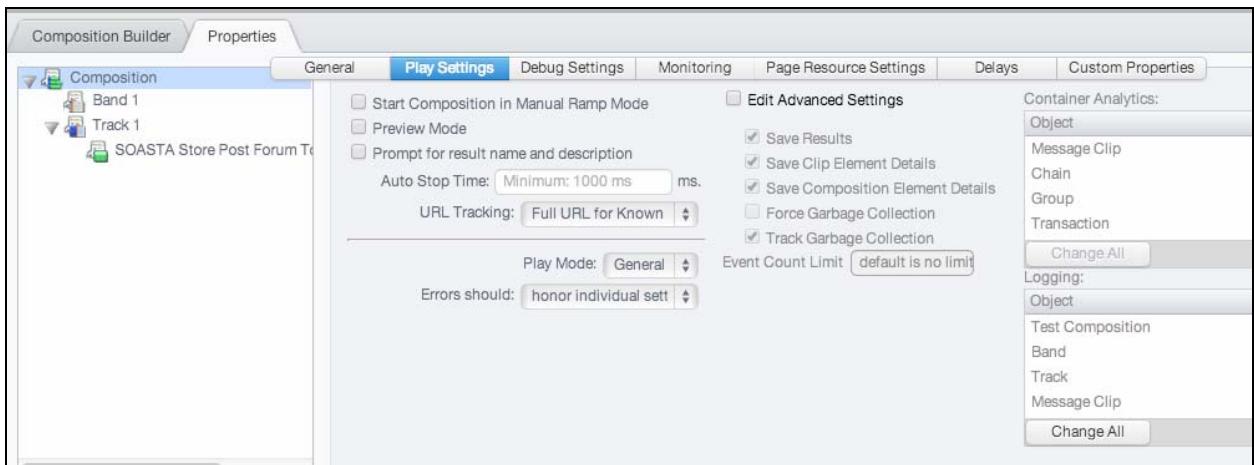
Click Save on the Composition Editor toolbar. The Save Composition box appears. Assign the composition a new name. For example, *SOASTA Store Forum Tutorial Composition*.

2. This tutorial test clip, like most tests, has think times and those think times can sometimes be quite long. A test might playback in a couple of seconds without think times, but might take many minutes with think times included. Think times are critical during load tests to ensure the HTTP traffic closely simulates real user traffic. However, think times are not necessary (in most cases) during the test debugging process. Many users override the think times in the clip with "0ms" think times. Note, however, there are a small number of applications that need think times to operate correctly. In those cases, users will typically override the think times with "1000ms". If you want to be conservative, simply override all think times with 1000ms—it covers your bases and still plays the test back for debugging purposes much more quickly. Now, let's set a composition-level think time override.

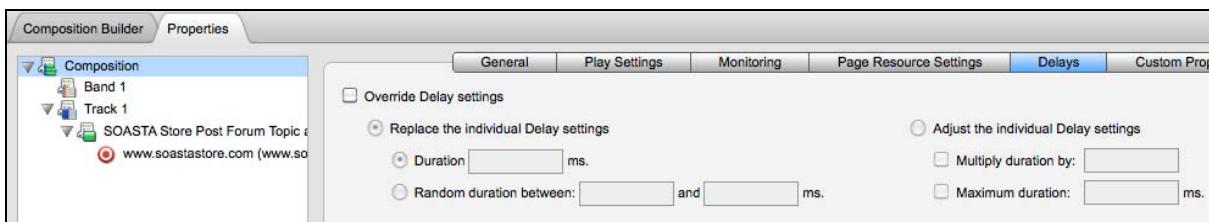
3. Click the “Properties” tab in the lower pane of the Composition Editor.



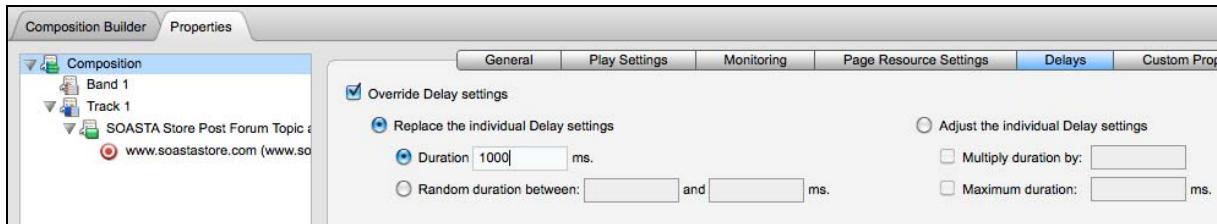
4. Click “Composition” in the left-hand navigation pane.



5. Click the “Delays” tab.



6. Click the “Override Delay settings” checkbox and set the “Duration” field as *1000* ms, which overrides each think time in the test with 1 second think time.



7. Save the composition.
8. Click the Play button (on the Composition Editor toolbar. The Composition Editor Play tab appears with the default dashboard in display while the test composition plays.

The Result Details dashboard appears with the status *Playing*.

Result Details Dashboard

The Result Details dashboard helps you determine the cause of errors in your test and has several methods for navigating through the test results.

Start by clicking the arrows next to the Track and Clip names to expand each Page in the test.

You can further expand individual Pages in the tree to drill down to individual message requests.

The screenshot shows the 'Result Details' window with the following details:

- Element Status:** is All
- Element Type:** is All
- Test Status:** Completed - With No Errors
- Total Components:** 129
- Total Messages and Actions:** 107
- Error Components:** 0

The left pane displays a hierarchical tree of test components:

- Draft of SOASTA Store Forum Tutorial Composition
 - Band 1
 - Track 1
 - SOASTA Store Post Forum Topic and Reply clip
 - soastastore.com Home Page
 - Delay1
 - Forum Default Page (not logged in)
 - Delay2
 - Post credentials
 - wp-login.php (2)
 - Forum Default Page (logged in)
 - soastastore.com (9)
 - userdefault.png
 - logout.png
 - subscribed.png
 - watching.png
 - profile.png
 - members-display.png
 - goinbox.png
 - Delay3
 - Forum Product Questions
 - Delay4
 - Post new forum topic
 - Load newly posted forum topic
 - Delay5
 - Post reply to forum topic
 - Load post with reply
 - Delay6
 - Logout
 - soastastore.com Home Page (after logout)

The right pane shows the summary for the 'Draft of SOASTA Store Forum Tutorial Composition' test:

Status	Effective Duration	Avg. Response Time	Total Message Bytes	Effective Message Throughput
Completed Composition completed.	2:07 (mm:ss) Start: Tue Jul 05 16:09:34 PDT 2011 75 ms. End: Tue Jul 05 16:12:33 PDT 2011 712 ms.	142 ms. Agg: 15 sec.	S: R: 70,963 526,092 Max number of threads used : 29	1 msgs/sec. 4,691 bytes/s 37,526 bits/s

Below the summary is a 'Component Summary' table:

Component	Retries due to unavailable local ports	Retry time due to unavailable local ports
Messages	107	Maximum 0 Minimum 0 Average 0 Total 0
Browser Actions	0	Maximum 0 Minimum 0 Average 0 Total 0
Scripts	0	Maximum 0 Minimum 0 Average 0 Total 0
Delays	6	Maximum 0 Minimum 0 Average 0 Total 0
Checkpoints	0	Maximum 0 Minimum 0 Average 0 Total 0
Events	634	Maximum 0 Minimum 0 Average 0 Total 0
Error Events	0	Maximum 0 Minimum 0 Average 0 Total 0
Error Components	0	Maximum 0 Minimum 0 Average 0 Total 0

Although errors didn't occur in this test playback, when errors do occur, the Result Details widget helps to easily identify both where and what errors occurred. A message will indicate that errors occurred during the test, how many errors there were, and any errors will be highlighted in red. Here are some sample screenshots of error conditions:

The screenshot shows the 'Result Details' window with the following details:

- All**
- Element Status:** is All
- Element Type:** is All
- Test Status:** Completed - With Errors

The left pane displays a hierarchical tree of test components:

- Draft of Composition for Wikipedia Tutorial Scenario clip
 - Band 1
 - Track 1

Result Details

Element Status: Is All Element Type: Is All

Failed Total Components: 7 Total Messages and Actions: 1 Error Components: 6 Error Messages and Actions: 1

Draft of SOASTA Store Forum Tutorial Composition

Band 1

Track 1

SOASTA Store Post Forum Topic and Reply clip

soastastore.com Home Page

soastastore.com (7)

soastastore.com (7)

Response did not pass validation.
Message send failed.

Start Time	Delta from sched	Response Time	Total Time	Total Message Bytes	Validations
21 sec.		914 ms. TTFB: 911 ms. TTLB: 914 ms.	915 ms.	S: R(Gzip): 3983,187	Validation of response body did not pass. Response did not pass validation.

Request Change to XML View Response Change to HTML View

Operation: get

GET / HTTP/1.1
Host: www.soastastore.com
Accept-Language: en-US,en;q=0.8
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
Accept-Encoding: gzip,deflate,sdch
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Proxy-Connection: keep-alive

URL Values /
Query String

Target:

HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must-revalidate, post
Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8
Date: Wed, 06 Jul 2011 20:41:05 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
Server: Apache/2.2.14 (Ubuntu)
Set-Cookie: PHPSESSID=d6pi6od0p2kv8o6af4lts1442; path=/
Vary: Accept-Encoding
X-Pingback: http://www.soastastore.com/xmlrpc.php
X-Powered-By: PHP/5.3.2-ubuntu4.9
Content-Length: 2684
Connection: keep-alive

The Navigation Tree view on the left is useful for navigating around multiple clips in a test composition. Click the arrow for any node to expand or collapse it.

Result Details

Element Status: Is All Element Type: Is All

Draft of SOASTA Store Forum Tutorial Composition

Band 1

Track 1

SOASTA Store Post Forum Topic and Reply clip

soastastore.com Home Page

Delay1

Forum Default Page (not logged in)

Delay2

Post credentials

wp-login.php (2)

Forum Default Page (logged in)

soastastore.com (9)

userdefault.png

logout.png

subscribed.png

watching.png

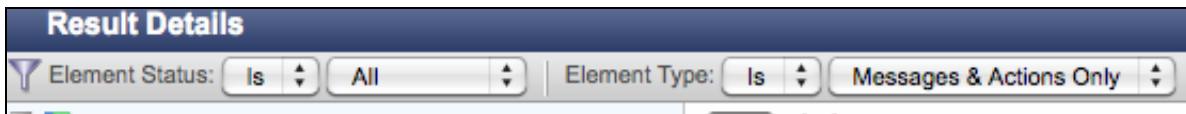
Completed - With

SOASTA Store

Draft of SOASTA Store Forum

Status	Effect
Completed	2:00
Composition completed.	

Use the filters to quickly display messages (or browser actions) only (within a single band, track, test clip, or chain); or, to display errors only.



- Click any Message or Action in the Cover Flow at the top to center it and display its request/response details and play statistics in the panes below.
- Use the scrollbar to browse the Cover Flow (if an item is selected in the Navigation Tree, the Cover Flow shows only items that pertain to the selection).
- Select any item in either the Navigation Tree or the Cover Flow to show its low-level details in the Input/Output or Events List tabs under the Cover Flow.
- Click the Events List tab on any object in the Cover Flow interface to see a sequential, event-by-event list of the given test result. The Events List displays the full request (headers/POST data) and response (headers/content) in the same view by clicking on Details node. This list provides the raw response data including hidden fields. The Events List is only available during debugging—not during a load test. This will be discussed in more detail later in the tutorial with regard to General Mode versus Load Mode.

Completed - With No Errors Total Components: 129 Total Messages and Actions: 107 Error Components: 0

Band 1 ► Track 1 ► SOASTA Store Post Forum Topic and Reply clip ► soastastore.com Home Page ► soastastore.com (7)

soastastore.com (7)

Event	Time	Level	Event Code	Description
9	51382	Info	Message: send	Sending Message. Band: "Band 1" Track: "Track 1" Clip: "SOASTA Store Post Forum Topic and Reply clip" Page: "soastastore.com Home Page" Message: "soastastore.com (7)"
10	51385	Verbose	Transport: send	Sending message "soastastore.com (7)" to Destination "www.soastastore.com (www.soastastore.com)", operation "get". Band: "Band 1" Track: "Track 1" Clip: "SOASTA Store Post Forum Topic and Reply clip" Target: "www.soastastore.com (www.soastastore.com)" Page: "soastastore.com Home Page" Message: "soastastore.com (7)" ► Details: Received response to message "soastastore.com (7)". Band: "Band 1" Track: "Track 1" Clip: "SOASTA Store Post Forum Topic and Reply clip" Target: "www.soastastore.com (www.soastastore.com)" Page: "soastastore.com Home Page" Message: "soastastore.com (7)" ▼ Details: <pre> HTTP/1.1 200 OK Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre Content-Encoding: gzip Content-Type: text/html; charset=UTF-8 Date: Tue, 05 Jul 2011 23:10:26 GMT Expires: Thu, 19 Nov 1981 08:52:00 GMT Pragma: no-cache Server: Apache/2.2.14 (Ubuntu) Set-Cookie: PHPSESSID=6nr2dn5t3s0mo9d1615mon07f2; path=/ Vary: Accept-Encoding X-Pingback: http://www.soastastore.com/xmlrpc.php X-Powered-By: PHP/5.3.2-lubuntu4.9 Content-Length: 2684 Connection: keep-alive </pre>
11	51849	Verbose	Transport: sent	

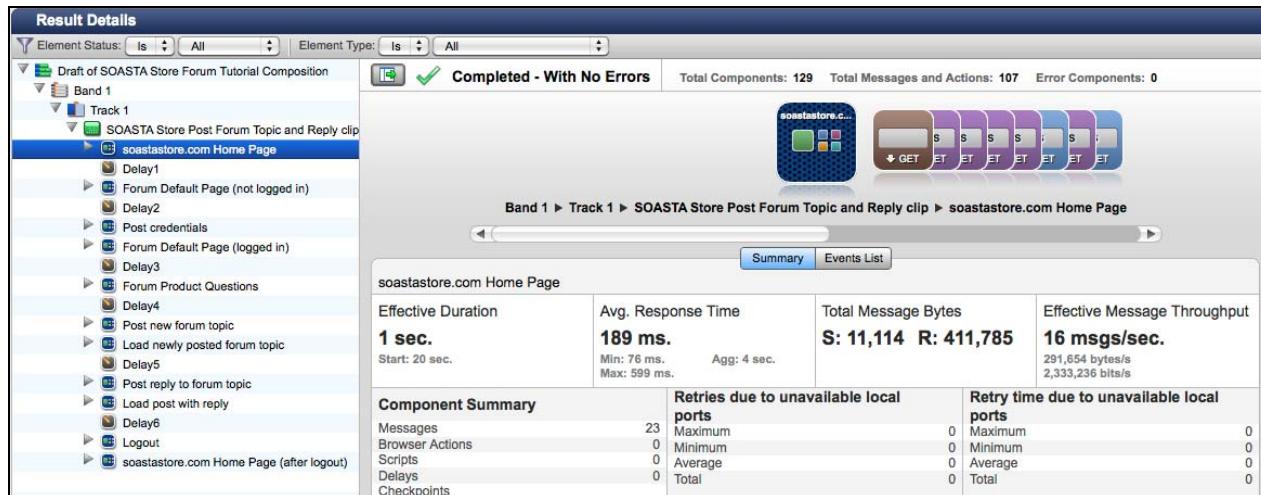
Verifying Test Parameterization

You will note that this test “Completed – With No Errors”. An important point to note is that **just because the test completed without errors does not necessarily mean the test was successful**. CloudTest automatically checks the HTTP Status Code for all HTTP requests in a test. If the response is an HTTP 2xx or 3xx response code, it is considered a success. If the response is an HTTP 4xx or 5xx response code, it is considered an error. Many applications, however, will not use these status codes to indicate if a request is successful or not. For example, many applications will return a valid response (i.e. HTTP 200 status code), but actually contain an error message on it—or return an error page with an HTTP 200 status code.

(A later section in this tutorial will walk you through the setup of custom validations to ensure pages and messages are coming back correctly.)

Until validations are setup in the test, the only way to ensure the test is working correctly is by reviewing each page request.

1. With the Result Details Dashboard in display, expand the Navigation Tree until all the pages are visible. Click the first page in the navigation page on the left (soastastore.com Home Page).



2. The dark-blue icon on the right (soastastore.com) represents the page. The data below the cover flow shows statistics associated with the playback of this page during the test. Click the brown “GET” icon to the right of the page and the Input Output tab.

Completed - With No Errors Total Components: 1 Total Messages and Actions: 1 Error Components: 0 [Jump to now](#)

soastast...
↓ GET

Band 1 ► Track 1 ► SOASTA Store Post Forum Topic and Reply clip ► soastastore.com Home Page ► soastastore.com

[Input Output](#) [Events List](#)

General

Name: **soastastore.com**

Start Time 1.584 sec.	Response Time 2.772 sec. TTFB: 2.754 sec.	Total Time 2.773 sec. TTLB: 2.754 sec.	Total Message Bytes Sent: 359 Received: 4,435(gzipped)
---------------------------------	--	---	--

Custom Properties [Change to XML View](#)

Input [Change to XML View](#)

```
GET / HTTP/1.1
Host: www.soastastore.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.116 Safari/537.36
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*
Proxy-Connection: keep-alive
```

Output [Change to HTML View](#)

Target:

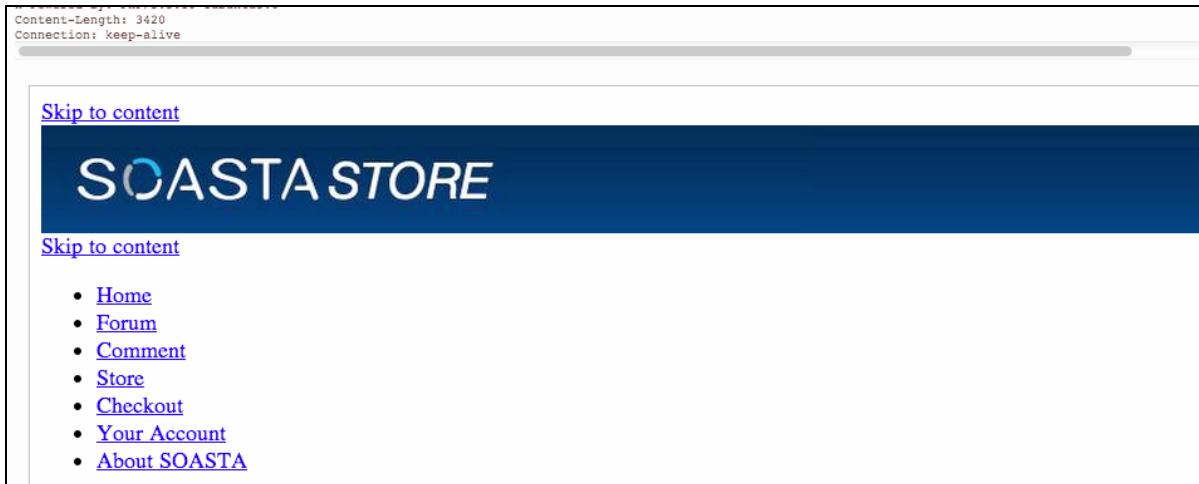
```
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must-revalidate, post-check=0
Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
Server: Apache/2.2.22 (Ubuntu)
Set-Cookie: website_mobile_detector=%7C0%7Cec51a1ccb6bcf87a32ce
Set-Cookie: PHPSESSID=s6j7geftlebtv5ra72luik4414; path=
Set-Cookie: wpsc_customer_cookie_fb2b19aaa84158875f98ab8f48187c0
Set-Cookie: AWSELB=B7458D1518CD511EC4A65A8C94305759B44044FEB7C74
Vary: Accept-Encoding
X-Pingback: http://www.soastastore.com/xmlrpc.php
X-Powered-By: PHP/5.3.10-lubuntu3.6
Content-Length: 3420
Connection: keep-alive
```

[Skip to content](#)

SOASTA STOR

[Skip to content](#)

3. This GET request represents the root request for index.html of soastastore.com. Scroll down in the response area to confirm the expected page contents are returned by the request. Note that this view is not an exact representation of the HTML rendering—consider it an interpreted rendering of the HTML. If you want to see the raw HTML that came back in this request, click the “Events List” tab.



4. Determine if the page response looks correct. In this case, the first page is coming back correctly.
5. Repeat steps 1-4 above for each of the pages in this test. Make sure the response content in the first GET (or POST) request of each page contains the expected information. Pay special attention to all POST requests and the GET requests that immediately follow the POST requests. This is where most replacement and parameterization errors typically occur.
6. At this point, there are two additional things you will need to fix before the test is complete. Before moving to the next step, see if you can identify those two things.

...
7. The answer: you haven't parameterized each step in the test. If you didn't read ahead, you should have found these two things:
 - After the new forum topic is posted, the GET request afterward is still requesting the originally recorded topic URL – not the one that was just created.
 - The reply POST isn't replying to the newly created topic—it is replying to the originally recorded topic.
8. We will correct these issues in the next couple sections.

Test editing and parameterization

Session Template Wizard

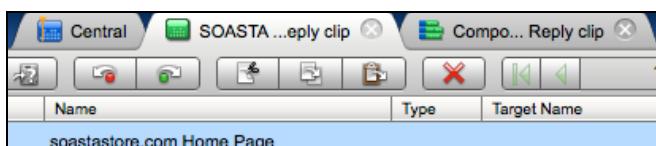
As you discovered in the previous section, although this test plays back with “No Errors” it isn’t working correctly. A couple of POSTs and related GET requests aren’t working 100% correctly. Let’s fix these errors.

There are a few ways these issues can be fixed:

- (a) Re-run the Session Template Package Wizard. There are other variables that need to be parameterized. This is typically the recommended approach, although it isn’t always the best solution for each situation.
- (b) Parameterize the dynamic values as they are discovered in the Clip Editor. This process is slower because each variable is done one at a time versus in batches. However, the Session Template Wizard makes this process very easy.
- (c) Manually parameterize the dynamic values by identifying the value to extract, manually creating extraction rules (or “property sets”), creating properties, and using the created properties in subsequent requests.

Since we’ve already seen option (a) this section of the tutorial will walk you through option (b) as described above for one variable (page_id). A later section will required that you replace an additional value using option (c) above.

1. Switch back to the Clip Editor on the tab bar. If you closed the test clip, re-open it from Central. Leave the composition open—you can edit the test clip without closing and reopening the Composition Editor.



2. Expand the grey triangle next to the pages for posting a new forum topic and the page immediately after that one (This tutorial used names: *Post new forum topic* and *Load newly posted forum topic*). The URL request after the post is hard-coded to something like */post-1/* or */post-1-1/*.

The screenshot shows the CloudTest Central interface. At the top, there's a toolbar with various icons. Below it is a navigation bar with tabs like 'Central', 'SOASTA ...', and 'Logout'. The main area displays two test steps:

- Post new forum topic**: An 'HTML Document' step with 'index.php (4)' as the file, 'www.soastastore.com' as the target, and 'post' as the operation. The query string is 'sf_ahead=post'.
- Load newly posted forum topic**: An 'HTML Document' step with 'soastastore.com (11)' as the file, 'www.soastastore.com' as the target, and 'get' as the operation. The URL is '/'. The query string is 'page_id=167/product-questions/post-1/'.

In the bottom right corner of the interface, there's a small icon of a globe with a play button, which is the 'Launch Session Template Wizard' button.

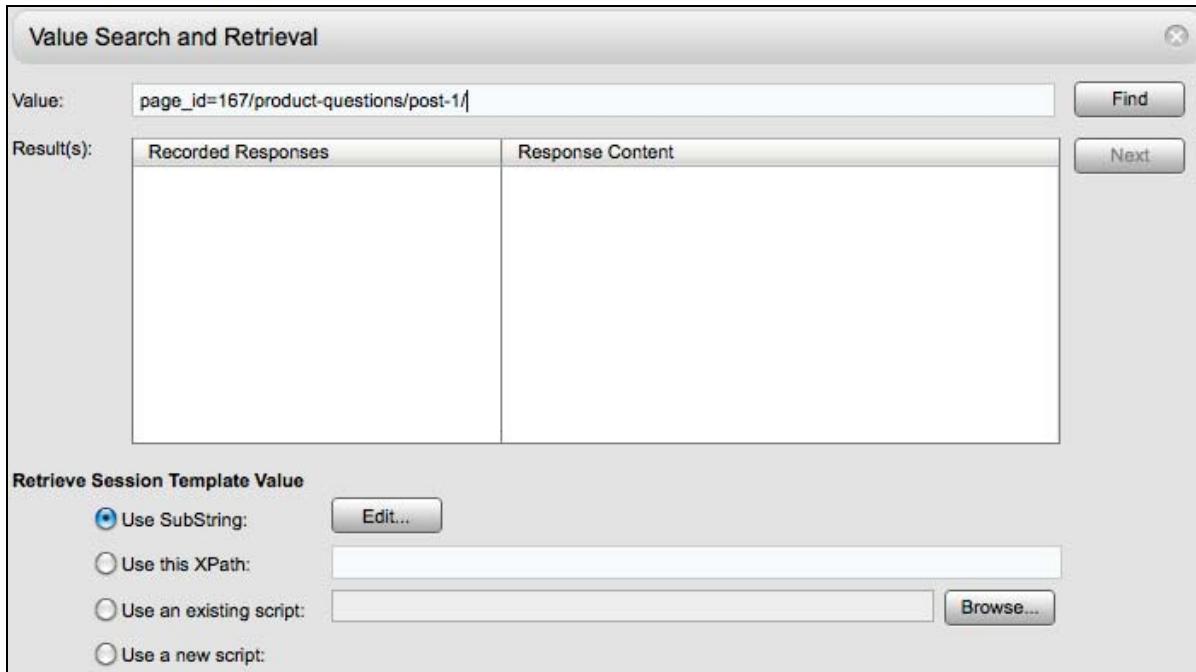
3. The next step is figuring out where the `/post-1/` value comes from and replacing it. To replace a single property in the test, we will use the Session Template Wizard that is accessed from within the test Clip Editor. Note that this is a different wizard from the Session Template Package Wizard we used earlier in the tutorial since we're only focusing on a single property. Highlight the portion of the Query String you want to replace. In this case, just highlight the entire Query String field. With the text highlighted, click the session template icon () in the upper-right corner of the Clip Editor and choose "Launch Session Template Wizard".

This screenshot shows the same test configuration as the previous one, but with a context menu open over the 'Query String' input field. The menu has three items:

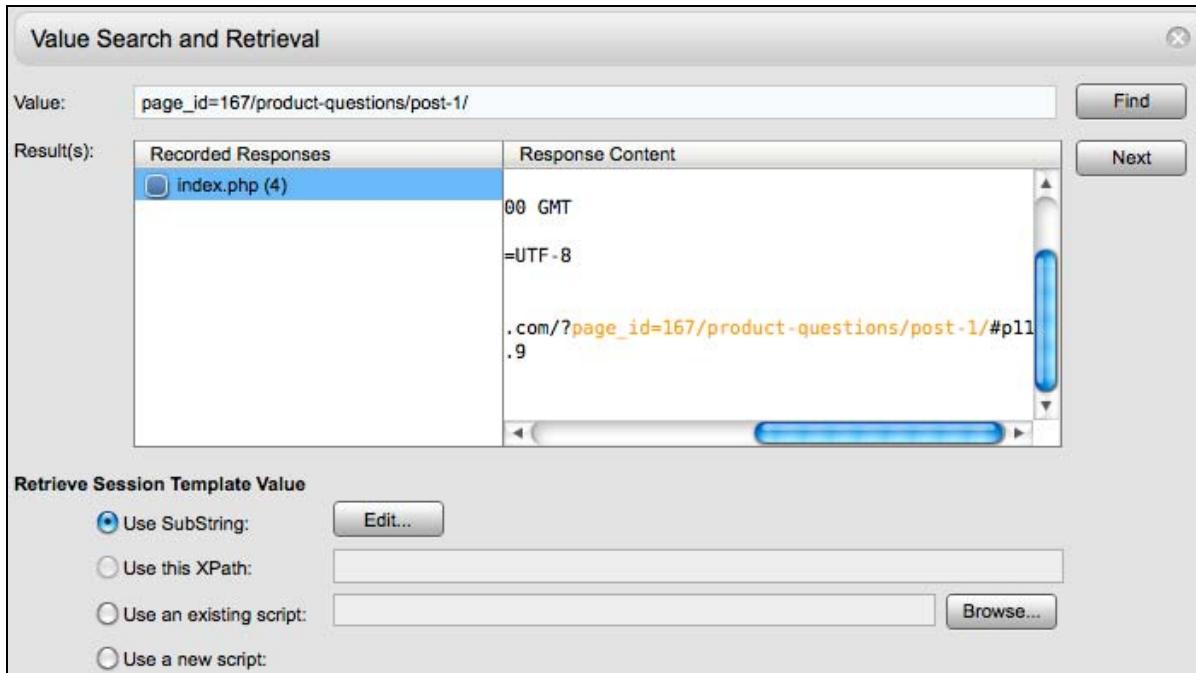
- Launch Session Template Wizard
- Launch Session Package Wizard
- Apply a Session Template or Package

The rest of the interface is identical to the first screenshot, showing the two test steps and the 'Inputs' section with the highlighted 'Query String' field.

4. The “Value Search and Retrieval” window will appear.



5. Click the “Find” button. This will search from the current message back to the front of the test. It will find the occurrence of this value in a response that preceded the selected message.



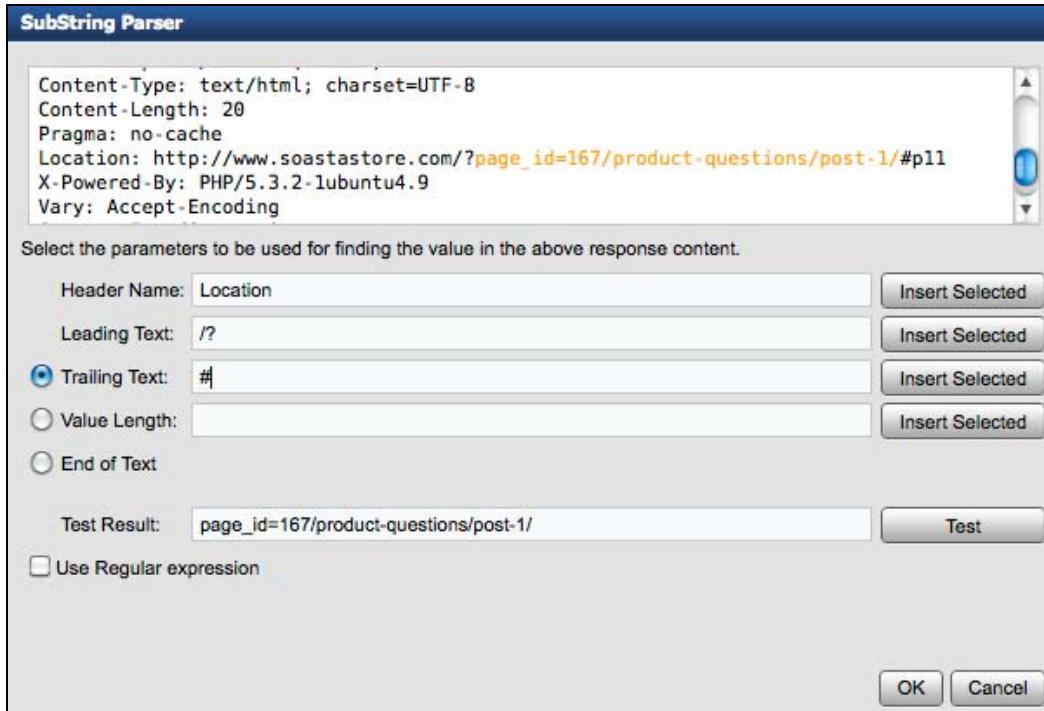
6. This string was found in the Location header of the response to the *index.php (4)* request. This is the page that POSTs the new topic. If you click the Next button repeatedly, it will search through the recorded responses of the clip to identify other

locations where this string was returned in a response. This clip has a few places where this string was found.

- For the purposes of this tutorial, select the first recorded response, *index.php (4)*. For this extraction, use a SubString. Click the “Edit” button next to the SubString option.

- The string you want to extract is coming back in the Location header of the response. The Header name has been pre-populated for you. Select the appropriate

leading and trailing text values to extract the orange text and insert them into the appropriate fields. Click the “Test” button to verify the extraction works correctly. Click OK when done.

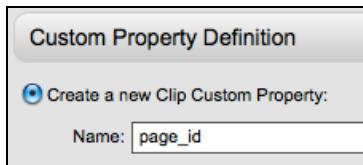


9. The SubString Parser window will close and you are returned to the wizard. Click the green arrow on the right of the wizard to move to the next step.

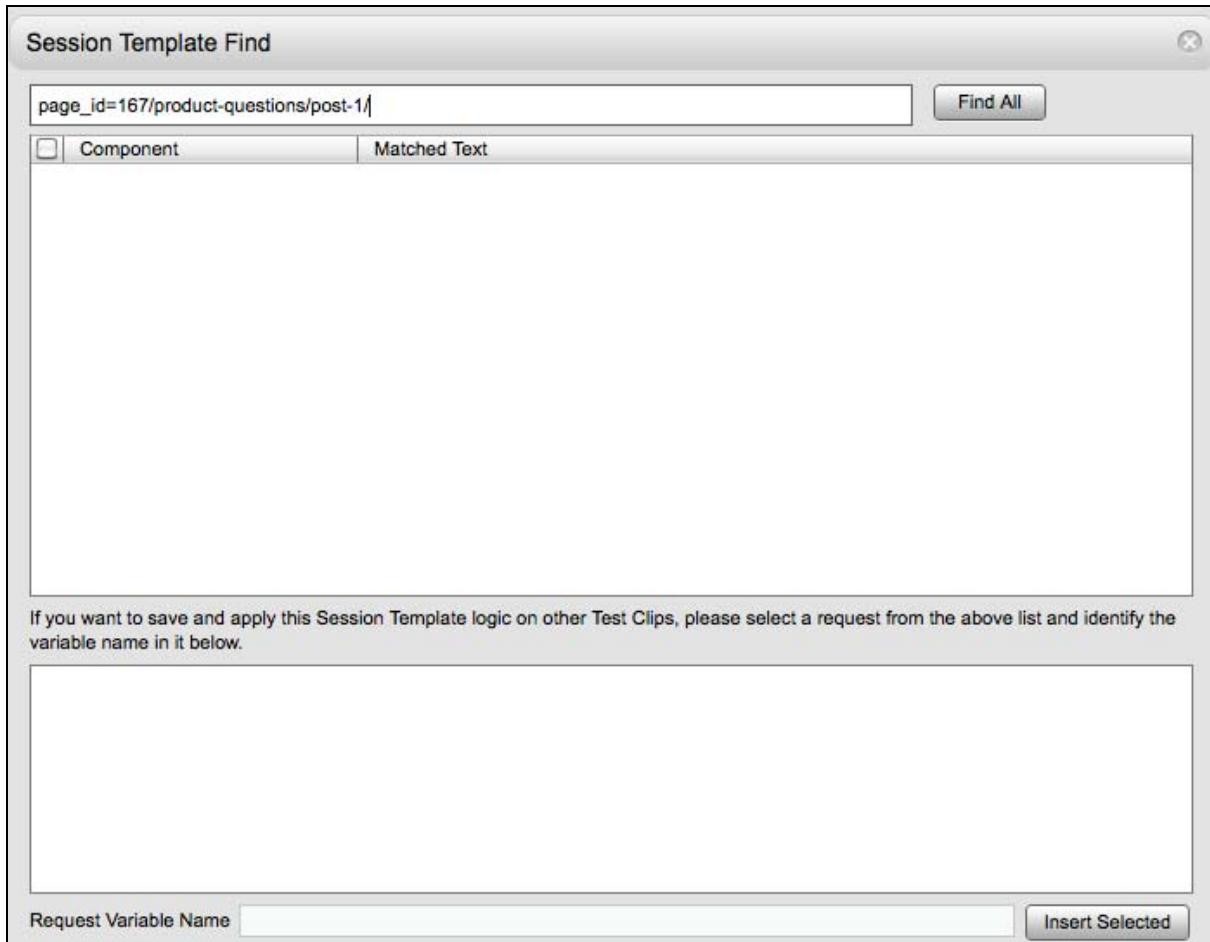


10. In this step you will define where you want the extracted data stored. You can either store the data in a new property or reuse an existing property. If you want to reuse an existing property from a previous replacement, you can do that, but be very careful as reusing properties can have unintended consequences when variable contents change. Since you are replacing this value for the first time, you want to

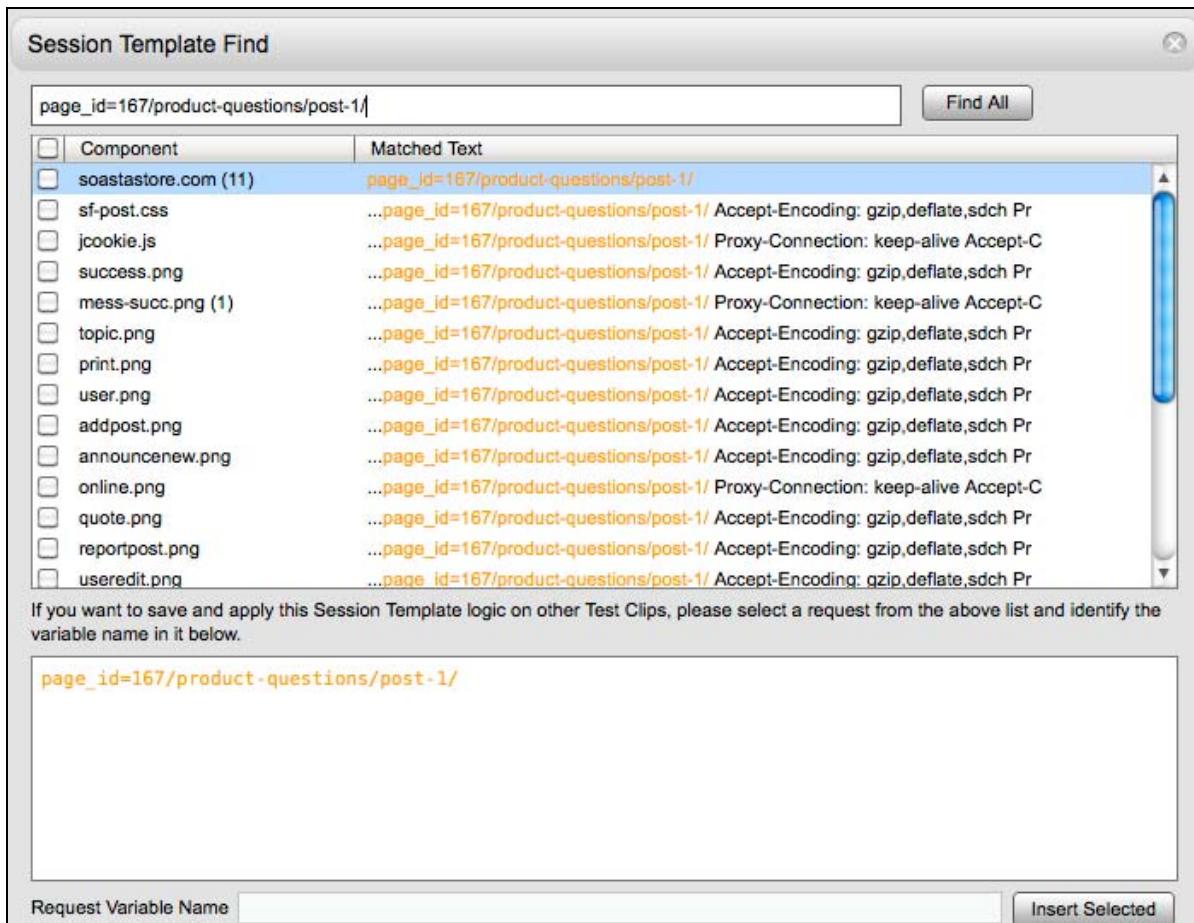
create a new clip custom property. Give it the name of *page_id*. It is recommended that you use new properties to avoid any contention issues.



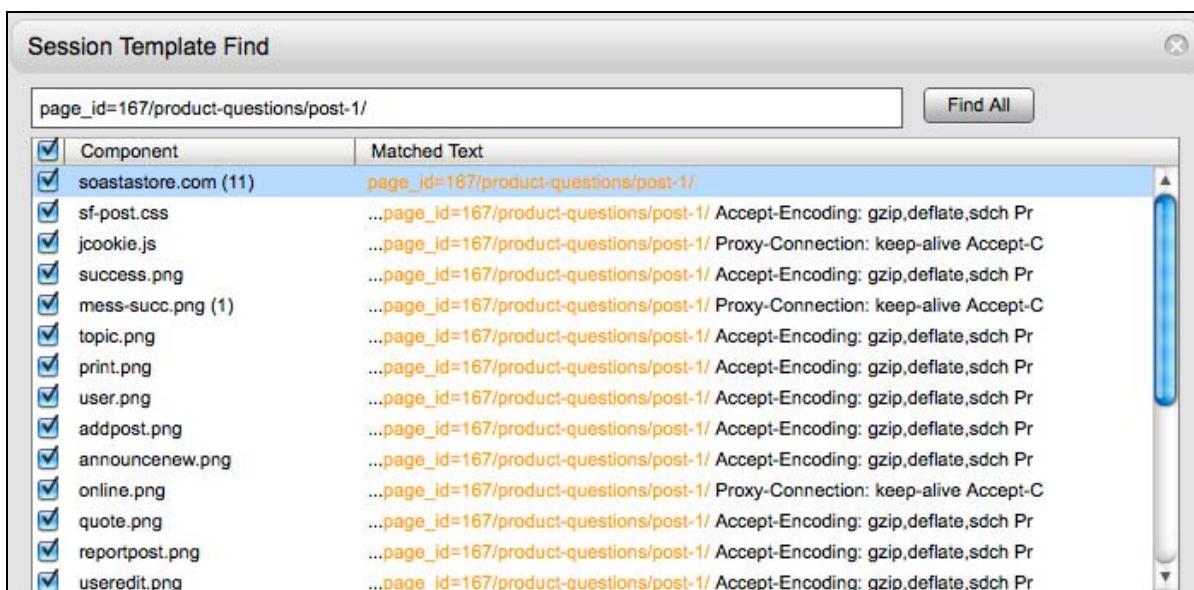
11. Click the green arrow on the right of the wizard to move to the next step.



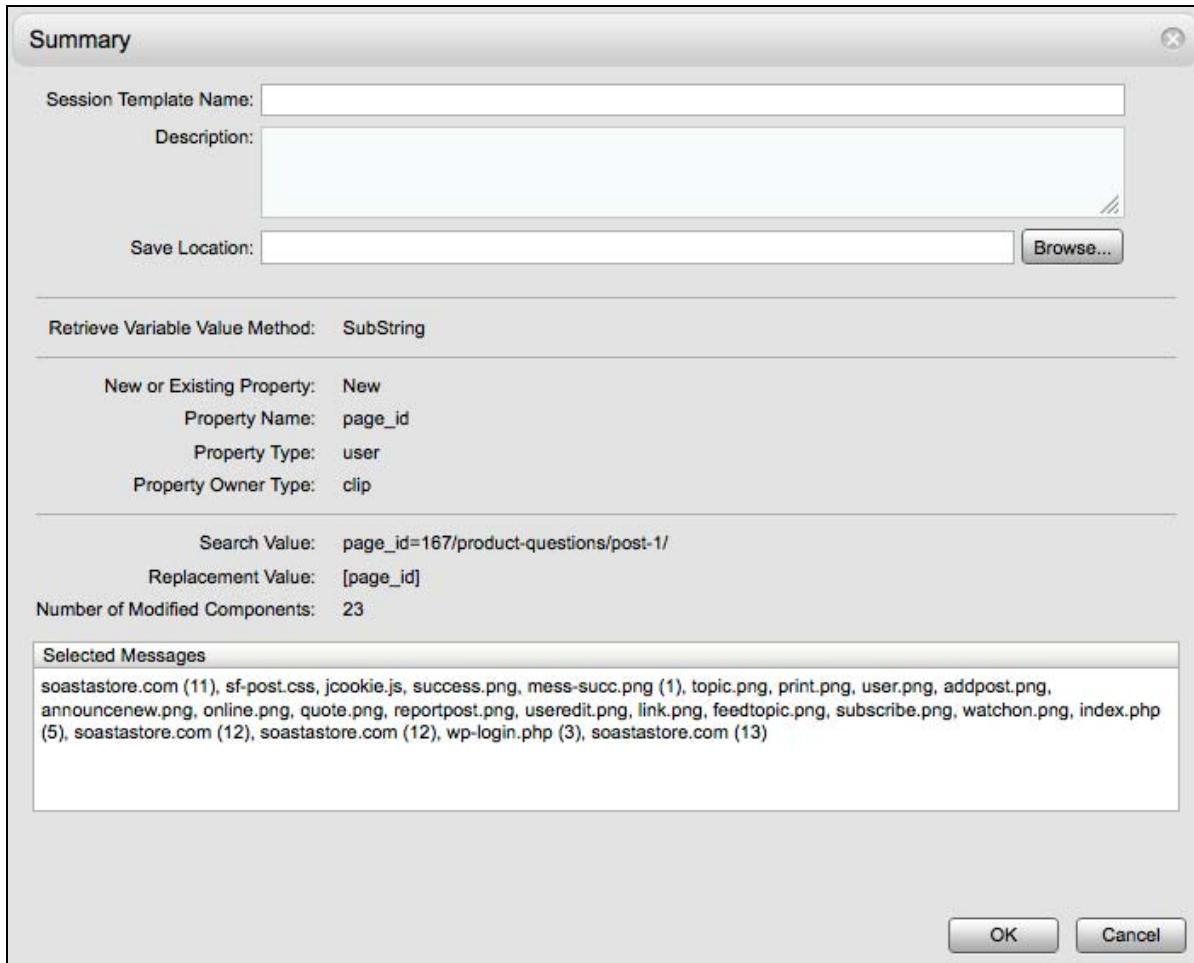
12. This step will search the entire clip to find all the locations where this string needs to be replaced. Click the "Find All" button.



13. Select each of the checkboxes for the requests you want the wizard to replace. In most cases, you will choose the top checkbox to select all of the values to be replaced with the dynamic property. Click the top checkbox to select all values to be replaced.



14. Click the green arrow on the right of the wizard to move to the next step.



15. Enter a name for this Session Template, *page_id*, click Browse... if you want to save to a specific location and click OK to close the wizard. After the wizard closes, the test clip will be reloaded with the new changes.

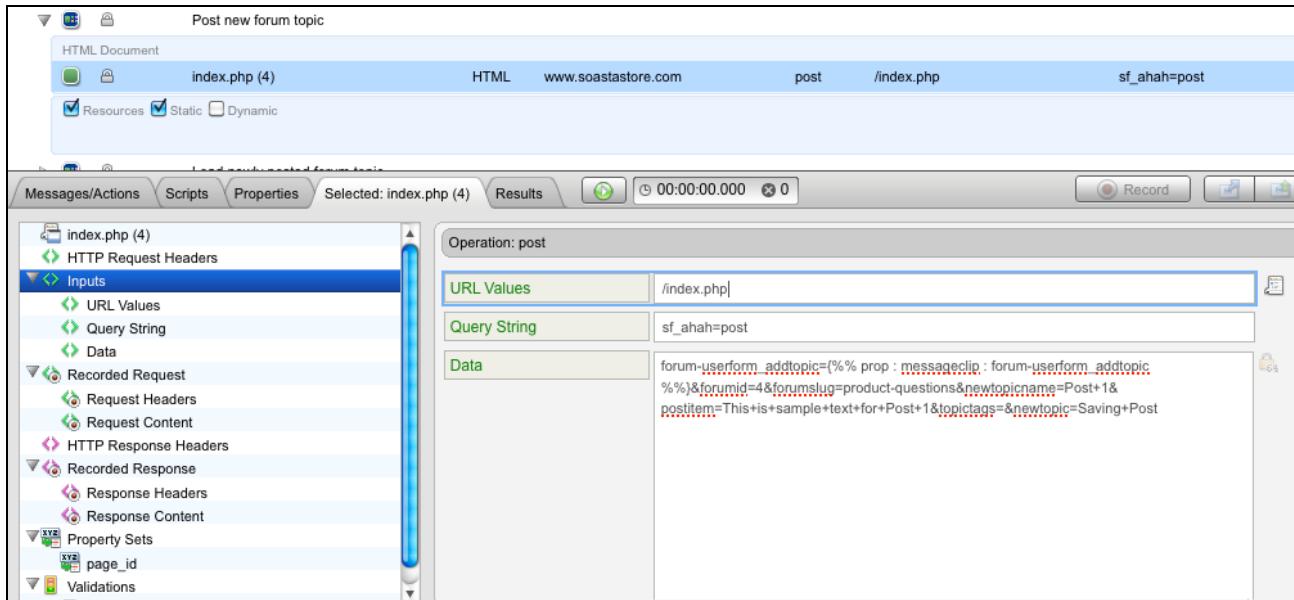
Session Template Parameterization: Under the covers

When you clicked OK to the complete the Session Template Wizard in the last section, several things happened:

- An extraction rule was defined (e.g. property set)
- A clip property was created
- The static string that needed to be replaced was replaced with a dynamic property
- A “session template” was created for re-use in other test clips (reusable session templates is a topic beyond the scope of this tutorial, please see CloudLink for more information)

Let's examine each of the places where the wizard altered the test in the test Clip Editor. Pay attention to the each of the interfaces described below. These are the same steps you will follow below to manually parameterize a test clip.

1. Expand the page that contains the POST that adds the new forum topic. In this tutorial, it was named *Post new forum topic*. Double-click on the POST request contained within the page.



2. In the lower pane, you will see “Property Sets” on the left. You will notice the *page_id* value that you just replaced. Click *page_id*.

The value of this field

Header: Location

Retrieved using: Substring

Leading text: /?

Trailing Text: #

Value Length:

Test on Recorded Response

Test on Played Response

Will be stored in this property

Type: Custom

Starting Element: Current Clip

Property Path: page_id

Match Options

If more than one value is matched, select the first

If no match is found, return null

3. The definition of the property set (e.g. extraction rule) is displayed above). Note that all of these configured settings can be changed in this user interface after the wizard completes. Additionally, notice the green + sign in the upper right-hand corner of this interface. You can create these extraction rules manually—without using the Session Template Package Wizard. This is used when manually correlating values.
4. This extraction rule is putting the extracted data into a clip property, *page_id*. Click the “Properties” tab to see the list of clip properties being used in this test clip. If it isn’t selected already, click “Clip Custom Properties” in the Property Type list.

Messages/Actions Scripts Properties Selected: index.php (4) Results 00:00:00.000

Property Type	Name
Clip Custom Properties	_wpnonce
Clip Properties	forum-userform_addpost
Delays	forum-userform_addtopic
Page Resource Settings	page_id
Applied Session Templates & Packages	

5. You will see all of the clip properties (created from both the Session Template Package Wizard and the Session Template Wizard) are listed in this interface.

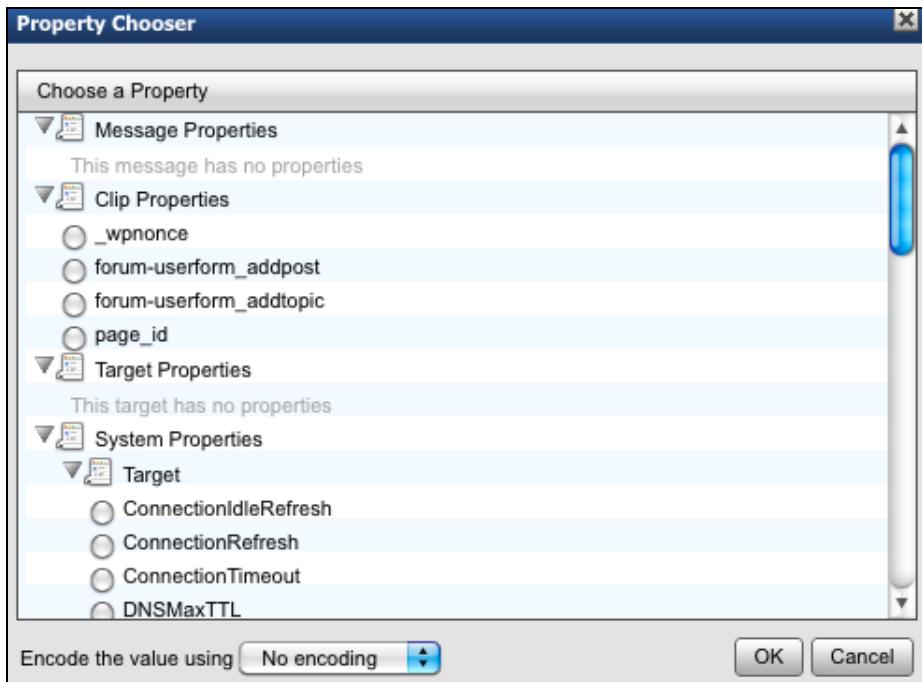
- If you are manually replacing values you can add or delete clip properties by clicking the + or - buttons in the upper-right corner of this interface. Give the new property a name and click the enter/return key on your keyboard to confirm the name.
- To see where this clip property is used (among other places), expand the page in the clip directly after this page that contains the POST. In this tutorial, it is named *Load newly posted forum topic*. Double-click on the first HTML Document request.

The screenshot shows the CloudTest application interface. At the top, there's a toolbar with various icons for file operations like Open, Save, Print, and a central search bar. The main window has two sections:

- Post new forum topic:** Shows one request: "index.php (4)" of type "HTML" to "www.soastastore.com" via "post" with URL "/index.php" and query parameter "sf_ahah=post".
- Load newly posted forum topic:** Shows a list of resources for "soastastore.com (11)". Under "Resources" (selected), there are 11 items: sf-post.css (CSS), jcookie.js (JS), success.png (PNG), mess-succ.png (1) (PNG), topic.png (PNG), print.png (PNG), user.png (PNG), addpost.png (PNG), announcenew.png (PNG), and online.png (PNG). The "sf-post.css" item is highlighted.

At the bottom, there's a toolbar with tabs for "Messages/Actions", "Scripts", "Properties", and "Selected: soastastore.com (11)". The "Properties" tab is active, showing a "Operation: get" dialog. In the "Query String" field, the value "{%% prop : clip : page_id %%}" is entered, with the placeholder text "%% prop : clip : page_id %%" highlighted in blue.

- You will see that this property is shown in both the list view above and the properties of the request below: `{%% prop : clip : page_id %%}`.
- Properties can be manually added to a request by highlighting the static text to be replaced in the Query String (or POST data), clicking the white Property Chooser icon () next to the text entry field, and selecting the property to be replaced (example shown below). When you click OK to the dialog, it will put the property into the test.



Finishing Parameterization

There are still more replacements you need to complete before the test is working correctly. Use a combination, if necessary, of the three methods described above to replace the remaining dynamic values (Session Template Package Wizard, Session Template Wizard, or manual parameterization).

You will need to replace the following name/value pairs:

- topicslug
- topicid (topicid is used in the POST message that replies to the forum topic)

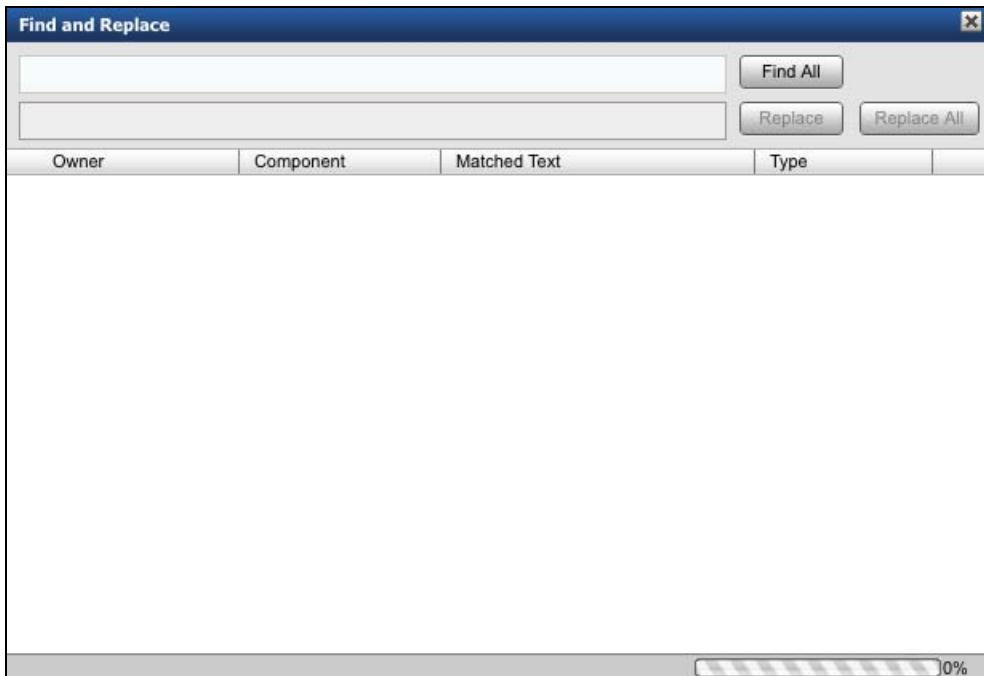
Hint: You can use any of the three methods to replace the topicslug value. However, you will need to use manual replacements to parameterize the topicid value.

When you think you have the test working, play it back in the composition. Manually log into the SOASTAStore.com Forum application and confirm that the reply is attached to the correct post. Run the composition 1-2 more times to make sure the tests works consistently. When you look at the Forum, you should see a post and a single reply in that post. If that isn't the case, something is wrong with the test. Go through the Result Details dashboard page-by-page to determine where your mistake was made.

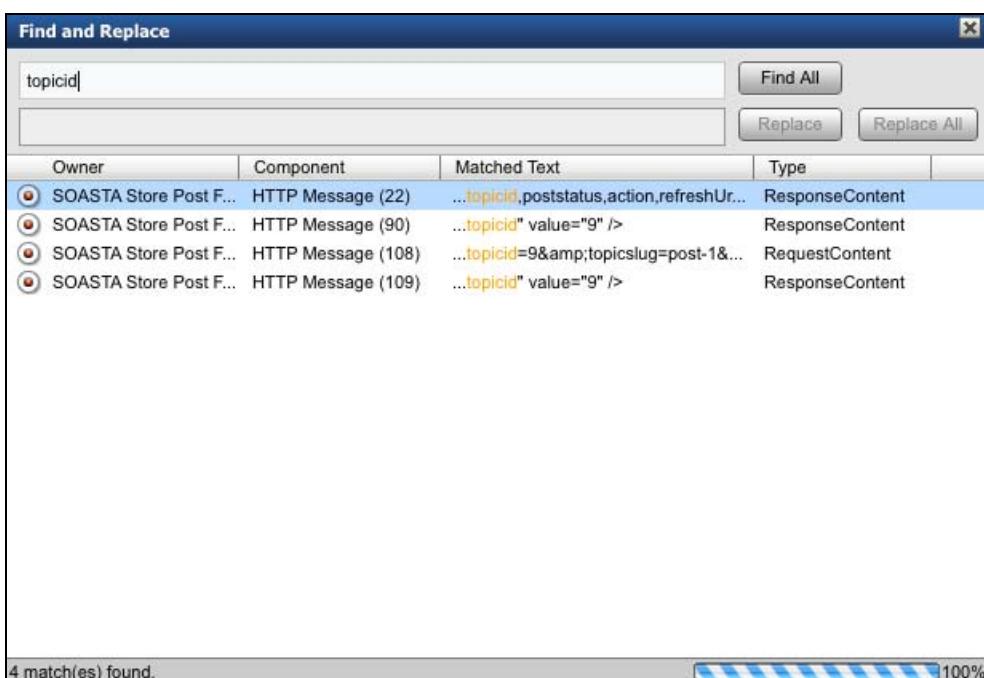
Find and Replace

HINT: You may find it useful to use the Find/Replace feature to determine which response the topicid value is coming back in.

1. In Central, right-click on the recording you made for this tutorial and select “Find and Replace”.



2. Enter “topicid” in the top field and click the “Find All” button.



1. This shows you each message where this string was found in the recording. You want to find the request where the topicid value first came back in a response. That is where you will create an extraction rule. Note the message number your response came back in. In this example, it is HTTP Message (90) that has the topicid value = 9.
 2. Close the Find and Replace dialog and open the recording (if it isn't open already). Scroll down to the Seq # you noted above (in this example, 90) and click it.

CloudTest Central SOASTA ...Reply clip Draft Composition SOASTA - Rob About Logout

Seq #	Resp. Code	Type	Method	Protocol	Host Name	URL	Source IP	Destination IP
87	200	GIF	GET	http	www.soastastore.co	/wp-content/plugins/simple-forum/styles/skins/default/checkbox.gif	127.0.0.1	174.129.15.12
88	200	PNG	GET	http	www.soastastore.co	/wp-content/plugins/simple-forum/styles/skins/default/mess-suucc.png	127.0.0.1	174.129.15.12
89	302	HTML	POST	http	www.soastastore.co	/index.php?sf_aahah=post	127.0.0.1	174.129.15.12
90	200	HTML	GET	http	www.soastastore.co	?page_id=167/product-questions/post-1/	127.0.0.1	174.129.15.12
91	200	CSS	GET	http	www.soastastore.co	/wp-content/plugins/simple-forum/styles/skins/default/sf-post.css	127.0.0.1	174.129.15.12
92	200	JS	GET	http	www.soastastore.co	/wp-content/plugins/simple-forum/resources/javascript/jquery/cookie.js?ver=3.1.4	127.0.0.1	174.129.15.12
93	200	PNG	GET	http	www.soastastore.co	/wp-content/plugins/simple-forum/styles/icons/default/success.png	127.0.0.1	174.129.15.12
94	200	PNG	GET	http	www.soastastore.co	/wp-content/plugins/simple-forum/styles/skins/default/mess-suucc.png	127.0.0.1	174.129.15.12
95	200	PNG	GET	http	www.soastastore.co	/wp-content/plugins/simple-forum/styles/icons/default/topic.png	127.0.0.1	174.129.15.12
96	200	PNG	GET	http	www.soastastore.co	/wp-content/plugins/simple-forum/styles/icons/default/print.png	127.0.0.1	174.129.15.12
97	200	PNG	GET	http	www.soastastore.co	/wp-content/plugins/simple-forum/styles/icons/default/user.png	127.0.0.1	174.129.15.12
98	200	PNG	GET	http	www.soastastore.co	/wp-content/plugins/simple-forum/styles/icons/default/addpost.png	127.0.0.1	174.129.15.12
99	200	PNG	GET	http	www.soastastore.co	/wp-content/plugins/simple-forum/styles/icon/icon-addpost.png	127.0.0.1	174.129.15.12

Decode this binary encoded message using: None

Request

Headers

```
Accept-Language: en-US,en;q=0.8
Cookie: PHPSESSID=mrpluni6777j6s3gahcllc55; wordpress_test_cookie=wordpress_logged_in_fb2b19aaa84158875f98ab8f48187c0e=testuser1
Host: www.soastastore.com
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
Referer: http://www.soastastore.com/?page_id=167/product-questions/
```

Body

Response

Headers

```
X-Pingback: http://www.soastastore.com/xmlrpc.php
Content-Length: 7283
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Last-Modified: Fri, 01 Jul 2011 15:38:04 GMT
Server: Apache/2.2.14 (Ubuntu)
X-Powered-By: PHP/5.3.2-1ubuntu4.9
Cache-Control: no-cache, must-revalidate, max-age=0
Pragma: no-cache
```

Body (GZip decompressed for viewing)

Raw Content Show the entire response Show as HTML

```
<fieldset>
<legend>Reply to Topic:</legend>
<div class="sfmessagestrip"><Post 1/><form class="sfpostformcontent" action="index.php?sf_aahah=post" method="post">
<input type="hidden" name="forum-userform_addpost" value="f783224f8a" />
<input type="hidden" name="forumid" value="4" />
<input type="hidden" name="forumslug" value="product-questions" />
<input type="hidden" name="topicid" value="0" />
<input type="hidden" name="topicslug" value="post-1" />
<table class="sfpostsavetable"><tr style="display:none"><td></td></tr></table>
<div class="quicktags">

</div>
<table class="sfpostsavetable">
<tr><td class="sfpostheading" width="100%" colspan="2">Smilies</td></tr><tr><td></td><td>...</td></tr>

```

3. Click the “Show the entire response” checkbox and using the browser’s find feature, locate the topicid in the response content. The value is what you need to extract using property sets.
 4. Note: When you use Find/Replace on a recording, you can only do a “Find” because it is read-only. If you use Find/Replace on a clip, you have the ability to replace strings in the test clip. Be careful, however, when using Find/Replace on a clip. You may want to back up the clip prior to doing a mass Find/Replace.

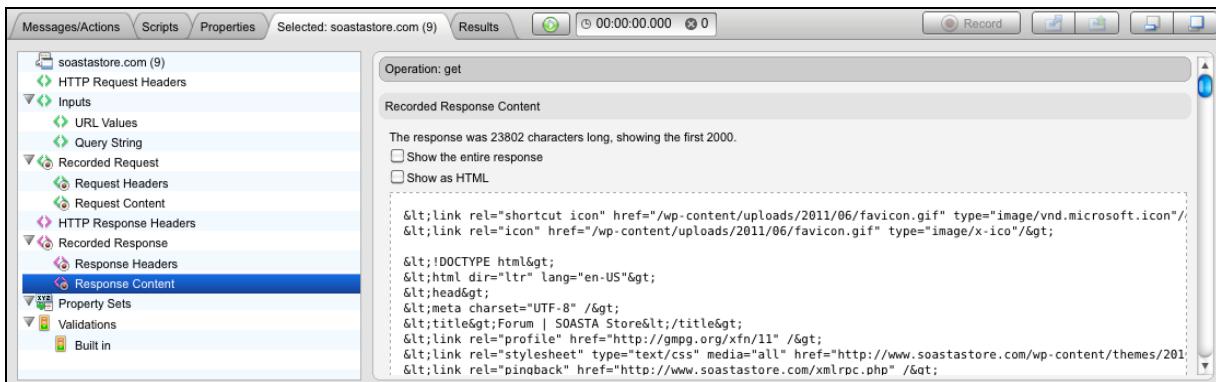
Creating User-Defined Validations

CloudTest provides the ability to create user-defined custom validations. Use the following steps to define a custom validation that will validate login success in the test clip.

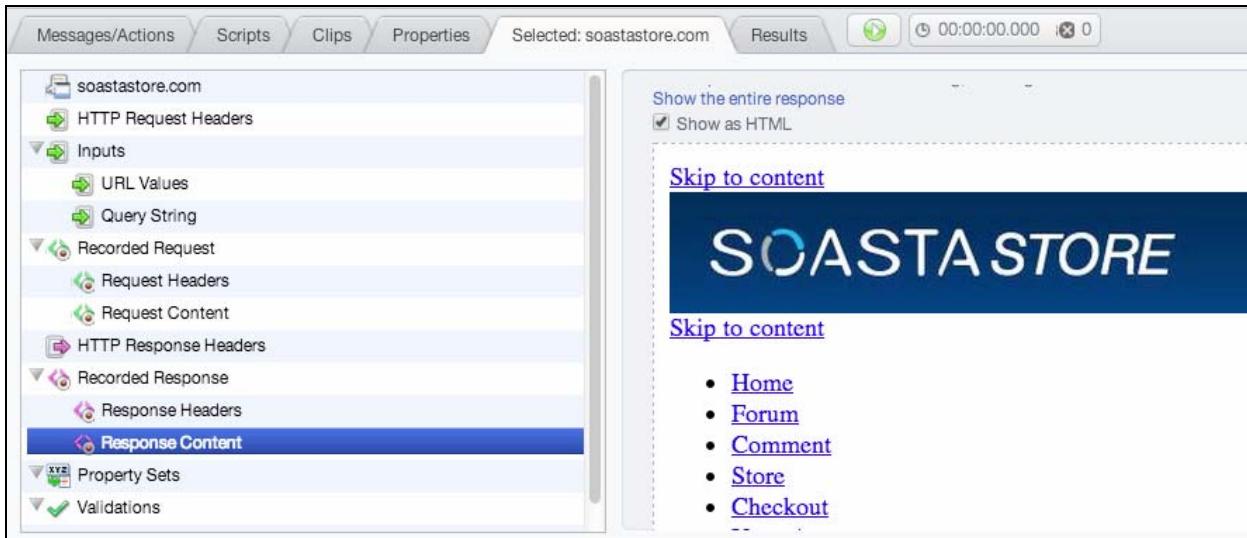
1. Open the page immediately after the POST message that sends the testuser1 credentials to the Forum. (We are validating the page after the POST because the POST response itself is just an HTTP 302 redirect—there isn't anything interesting to validate on that request.) In this tutorial, this page was named *Forum Default Page (logged in)*. Double-click the first message in the page, the HTML Document.

The screenshot shows the CloudTest application interface. The main window displays two sections: 'Post credentials' and 'Forum Default Page (logged in)'. The 'Post credentials' section contains a single item: 'wp-login.php (2)' with a 'post' operation to '/wp-login.php' and a query parameter 'action=login&view=forum'. The 'Forum Default Page (logged in)' section contains several items under 'soastastore.com (9)', including 'userdefault.png', 'subscribed.png', 'logout.png', 'watching.png', 'members-display.png', and 'profile.png', all with 'get' operations. The bottom-left pane shows a tree view of recorded requests and responses, with the 'Inputs' node expanded. The 'Inputs' tab in the bottom-right pane is active, showing an 'Operation: get' and two input fields: 'URL Values' (containing '/') and 'Query String' (containing 'page_id=167').

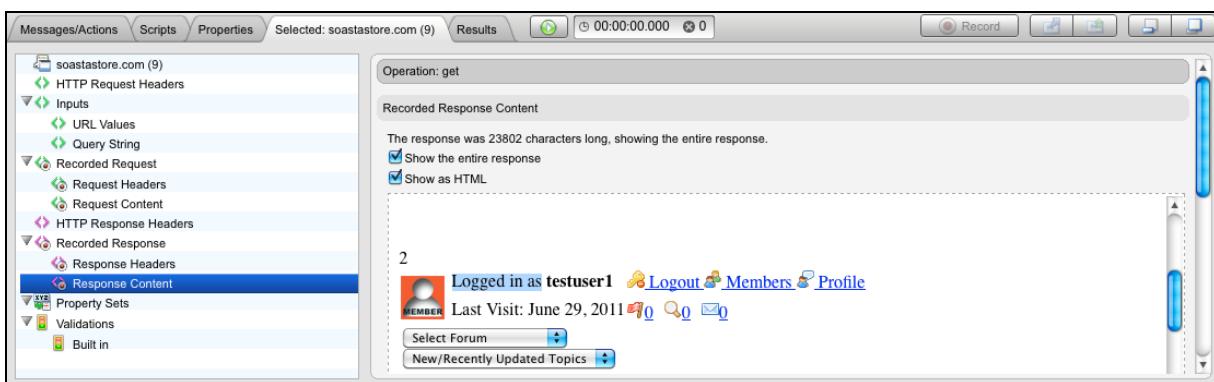
2. The first thing you need to determine is what text you want to validate. Meaning, what text must be in the response for this request to be considered a success? If you already know that, great. If not, you can use the Recorded Response Content to figure it out. Click Response Content in the left-hand lower pane.



- Click the "Show the entire response" and then "Show as HTML" checkbox.

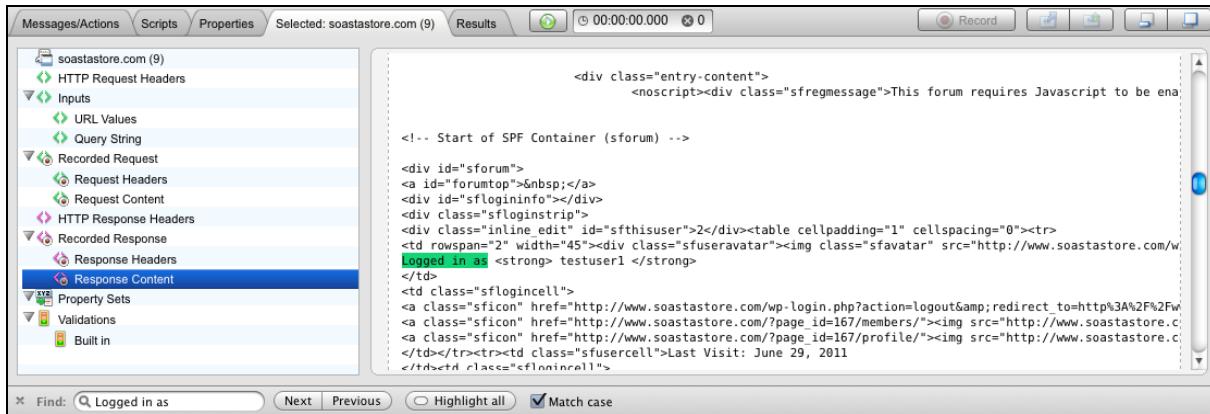


- Scroll down in the window to identify text you want to validate on. In this case, you will see the text "Logged in as".



- Now, be careful. You might want to just use the text "Logged in as" as your validation. However, remember this is a rendered version of the HTML. You need to make sure there aren't any hidden tags or other HTML artifacts contained in the actual HTML document. The validation is occurring on the raw HTML response – not the rendered HTML. To be sure this string actually shows up in the HTML

document, uncheck the “Show as HTML” checkbox and use your browser’s Find feature to try to locate the string you want to validate (in this case, “Logged in as”).

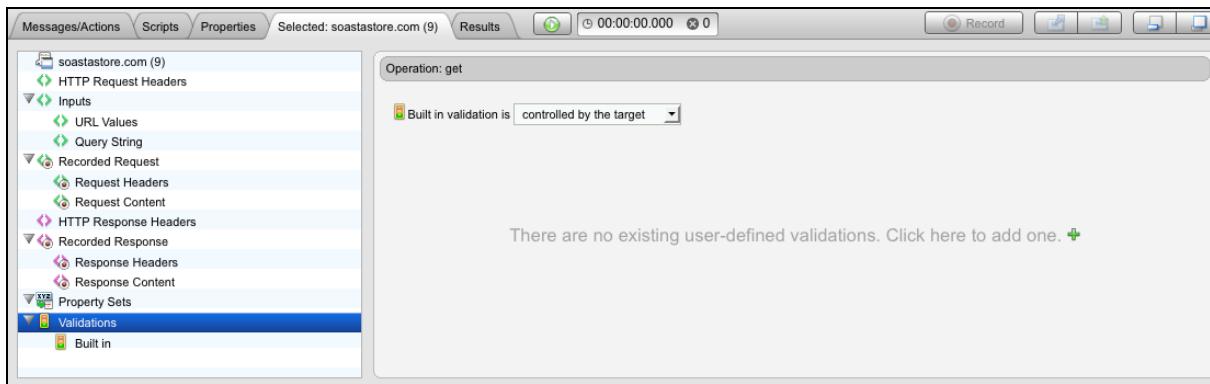


```

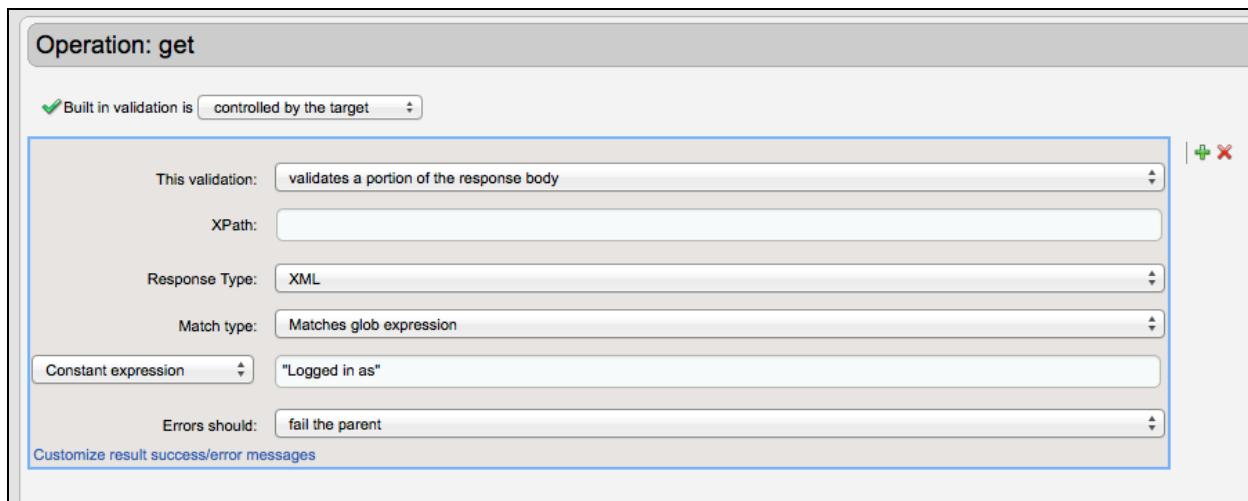
<div class="entry-content">
    <noscript><div class="sfregmessage">This forum requires Javascript to be enabled.
    </noscript>
    <!-- Start of SPF Container (sforum) -->
    <div id="sforum">
        <a id="forumtop">&nbsp;</a>
        <div id="slogininfo"></div>
        <div class="sfloginstrip">
            <div class="inline_edit" id="sfthisuser">2</div><table cellpadding="1" cellspacing="0"><tr>
                <td rowspan="2" width="45"><div class="sfuseravatar"></div></td>
                <td class="sflogincell">
                    <a class="sficon" href="http://www.soastastore.com/wp-login.php?action=logout&redirect_to=http%3A%2F%2Fwww.soastastore.com">Logout</a>
                    <a class="sficon" href="http://www.soastastore.com/?page_id=167/members/"></a>
                    <a class="sficon" href="http://www.soastastore.com/?page_id=167/profile/"></a>
                </td>
            </tr><tr>
                <td class="sflogincell">Last Visit: June 29, 2011</td>
            </tr>
        </table>
    </div>

```

6. In this case, you are fine. The text “Logged in as” shows up in the response, so we can use this string. Obviously, make sure this string **only** shows up when the test is successful, or the validation won’t be useful.
7. In the lower pane, click “Validations”.



8. Click the green plus sign (+) in the area on the right to add a user-defined validation.



This validation:	validates a portion of the response body
XPath:	<input type="text"/>
Response Type:	XML
Match type:	Matches glob expression
Constant expression	"Logged in as"
Errors should:	fail the parent

9. The Validation form appears. Accept the default value: *This validation: validates a portion of the response body*. Note that you can also validate response headers.
 10. Set the Response Type to *XML*.
 11. Set the Match Type to *Matches glob expression*.
 12. Enter the Glob Expression: **Logged in as**
- Note:** You are entering **Logged in as** using the asterisk wildcard character to start and end the expression; in order to match this as a Glob.
13. If this validation should stop the test clip from proceeding, leave the Failure Action set to *Fail the parent*.
 14. It is recommended that you update the success and failure messages. Set the Success Message as *Login Successful* and the Failure Message as *Login Failed*.

Operation: get

Built in validation is controlled by the target

This validation: validates a portion of the response body

XPath:

Response Type: XML

Match type: Matches glob expression

Constant expression *Logged in as*

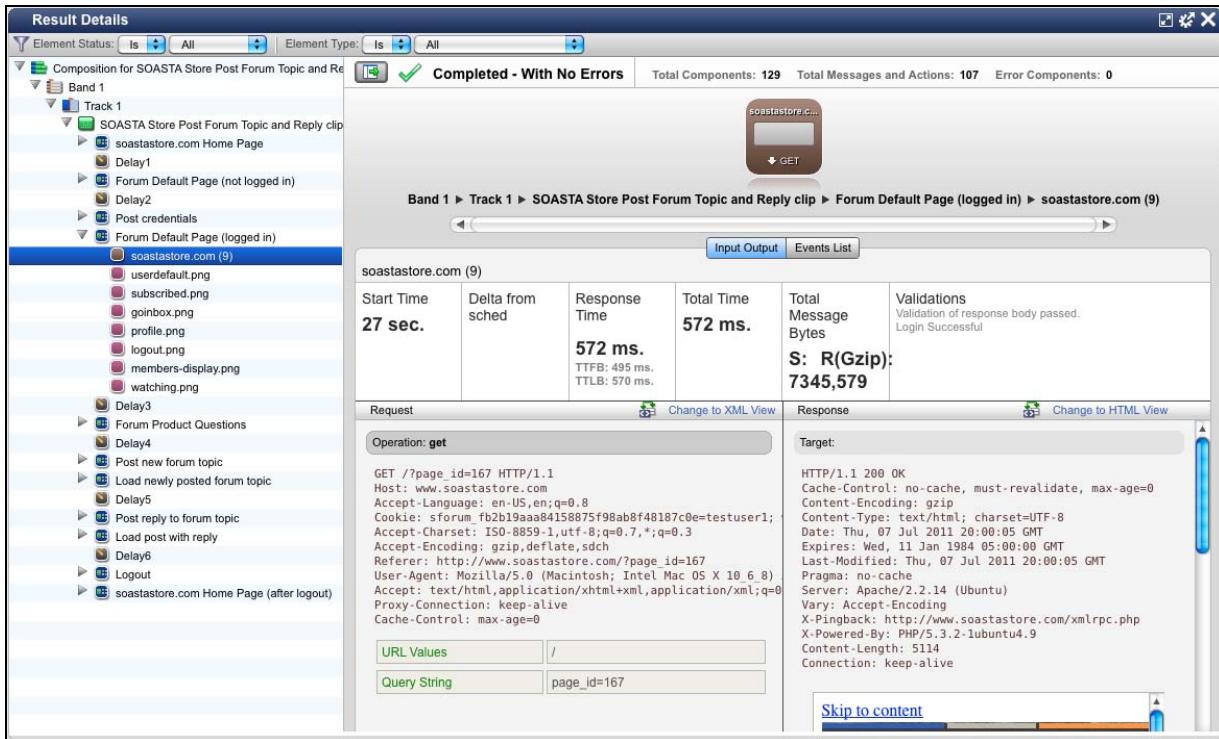
Failure Action: Fail the parent

Success Message: Login Successful

Failure Message: Login Failed

Hide result success/error messages

15. Click Save in the Clip Editor toolbar.
16. You need to confirm the validation is working properly. Play this clip in the composition editor (with delays/think times overridden as described earlier). Expand the navigation tree to find the message you just applied this validation to. Make sure you see the “Login Successful” text in the Validations section of the Result Details dashboard. If you see an error, that means that either (a) your validation is wrong or (b) your test clip isn’t actually logging in correctly. Inspect the login POST and the contents of this message to determine the problem.



17. Add additional validations to the topic forum POST and reply POST messages. Follow the process described above. Save the clip and verify the validations are working correctly in the test.

Using Scripts

CloudTest scripts are written in JavaScript for the CloudTest Object Model, which contains pre-defined JavaScript objects to which every script has access. Scripts in CloudTest can modify the current running test composition.

SOASTA CloudTest offers three property variables (`$prop`, `$sysprop`, and `$globalprop`) that simplify setting composition, system, and global properties.

We will use a script in this tutorial to generate a unique forum topic name that has a random number in it.

1. In Central, click on Scripts. Click the new button (). This opens a blank script-editing interface.
2. Copy and paste (or re-type) the code below into the script editor:

```

function generateRandomNumber(lowerExtreme, upperExtreme) {
    return Math.floor(Math.random() * (upperExtreme - lowerExtreme) + lowerExtreme);
}

var randomNumber = "" + generateRandomNumber(10000,99999);

$context.result.postMessage($context.result.LEVEL_INFO, "Random number: " +
randomNumber);

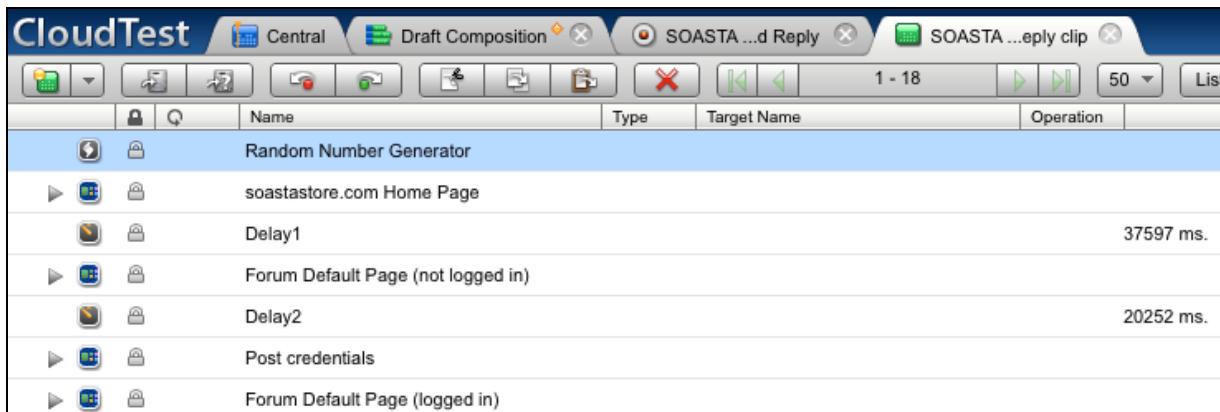
$prop.set("MessageClip", "randomNumber", randomNumber);

```

3. Save the script with the name “Random Number Generator”.
4. If you have the test clip open, close it and reopen it.
5. Click the “Scripts” tab in the Clip Editor lower pane, navigate to the Folder where the “Random Number Generator” script is located, and then select it.

Since, in this case, we want the random number generated at any point before we try to use it, it must be put into the test clip before we post the forum topic. A best practice is to put data generation scripts at the top of the test—as the first element in the test clip.

6. Drag the script into the test clip as the first item in the test. This script will now be executed before any of the HTTP requests are made.



7. You will notice in the script code that the last line in the script will put that random number into a clip property named “randomNumber”. Create that property now in the “Properties” tab of the lower pane.
8. The last step is to use the random number in an HTTP request. We will use it in the POST that creates the forum topic. Open the page for that POST. In this tutorial, it is named *Post new forum topic*.
9. Double-click the HTML Document in this Page.

The screenshot shows the CloudTest interface with a recorded script for 'index.php (4)'. The script details various steps including delays, forum product questions, posting a new topic, loading the topic, replying, and logging out. The 'Results' tab shows the recorded POST data for the index.php request, which includes URL Values (index.php), Query String (sf_ahah=post), and Data (forum-userform_addtopic=%% prop : messageclip : forum-userform_addtopic %%&forumid=4&forumslug=product-questions&newtopicname=Post+1&postitem=This+is+sample+text+for+Post+1&topictags=&newtopic=Saving+Post).

10. Of most interest is the POST Data.

The screenshot shows the CloudTest interface with the 'Data' section of the recorded POST request. The data field contains the following POST data:

```
forum-userform_addtopic=%% prop : messageclip : forum-userform_addtopic %%&forumid=4&forumslug=product-questions&newtopicname=Post+1&postitem=This+is+sample+text+for+Post+1&topictags=&newtopic=Saving+Post
```

11. Notice the attribute named “newtopicname” has a value of “Post+1”. This is the element we want to update. Replace the “1” with the randomNumber clip property. As you did above, do this by: highlighting the “1”, clicking the white property chooser icon, selecting the randomNumber property, and clicking OK. Repeat this process for the second occurrence of “1” in the last line of POST data. The final POST data should look something like this:

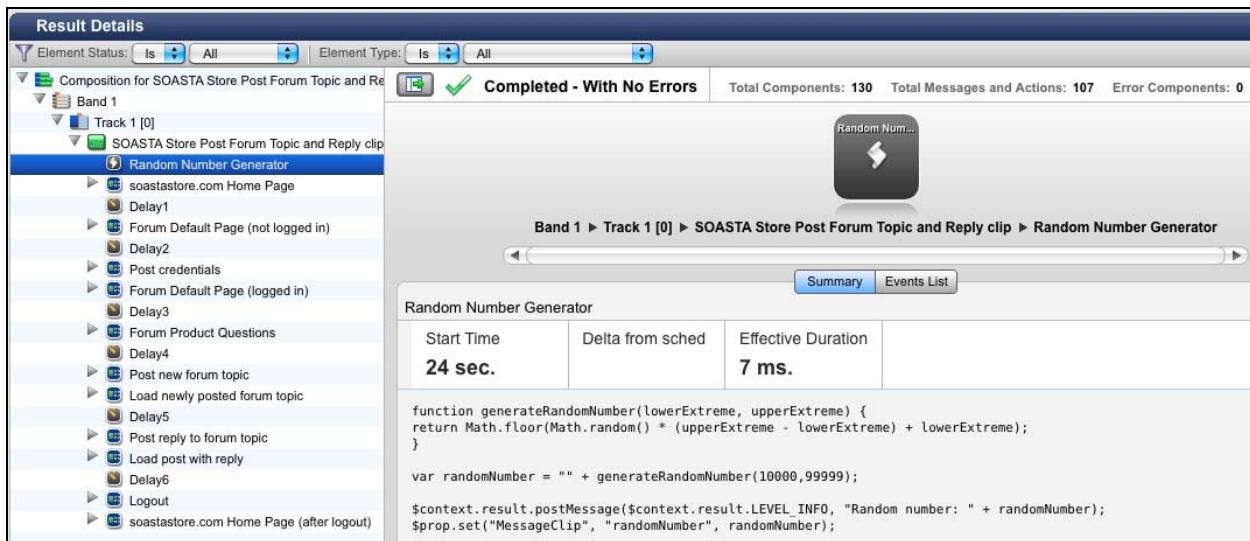
The screenshot shows the CloudTest interface with the 'Data' section of the recorded POST request after modification. The data field now contains:

```
forum-userform_addtopic=%% prop : messageclip : forum-userform_addtopic %%&forumid=4&forumslug=product-questions&newtopicname=Post+%% prop : clip : randomNumber %%&postitem=This+is+sample+text+for+Post+%% prop : clip : randomNumber %%&topictags=&newtopic=Saving+Post
```

12. Another “1” shows up in the POST that replies to the topic. You need to find that POST and insert the randomNumber property instead of the “1” in that message as well.

Note: If you are completing this tutorial in a classroom-training environment (or sharing the same SOASTAStore.com environment), you probably want to change the title of your forum topics to something unique. Meaning, instead of using “POST+[randomNumber]”, maybe you want to include your name like: “Rob+POST+[randomNumber]”. Make this change now in the POST that creates the new forum topic.

13. Save the test clip.
14. Play the test back in a composition (don’t forget to override the delays/think times).
15. Verify the random number that was generated by expanding the node next to the clip icon () in the Result Details dashboard. The first element in the clip is the “Random Number Generator” script; click it.



16. The script that was executed is shown on the right. Click the Events List button to see the events associated with this script.

The screenshot shows the Final Cut Pro X Events List interface. At the top, there's a navigation bar with the path: Band 1 ▶ Track 1 [0] ▶ SOASTA Store Post Forum Topic and Reply clip ▶ Random Number Generator. Below the navigation bar is a toolbar with a back arrow, forward arrow, and a search field labeled "Random Num...". Underneath the toolbar is a header bar with "Summary" and "Events List" buttons, where "Events List" is highlighted. The main area is titled "Random Number Generator" and contains a table with the following data:

Event	Time	Level	Event Code	Description
8	24855	Info	Transition: start	Script starting (linked Script "/Random Number Generator"). Band: "Band 1" Track: "Track 1" [0] Clip: "SOASTA Store Post Forum Topic and Reply clip" Script: "Random Number Generator"
9	24861	Info	Transition: scmsg	Random number: 32216 Band: "Band 1" Track: "Track 1" [0] Clip: "SOASTA Store Post Forum Topic and Reply clip" Script: "Random Number Generator"
11	24861	Verbose	Transition: prsetr	Property "randomNumber" of Clip "SOASTA Store Post Forum Topic and Reply clip", Track "Track 1" [0], Band "Band 1" set to new value. Band: "Band 1" Track: "Track 1" [0] Clip: "SOASTA Store Post Forum Topic and Reply clip" Script: "Random Number Generator"
				▶ Details:
12	24861	Info	Transition: end	Script completed. Band: "Band 1" Track: "Track 1" [0] Clip: "SOASTA Store Post Forum Topic and Reply clip" Script: "Random Number Generator"
13	24861	Statistics	Transition: stats	Script statistics. Band: "Band 1" Track: "Track 1" [0] Clip: "SOASTA Store Post Forum Topic and Reply clip" Script: "Random Number Generator"
				▶ Details:

17. The `$context.result.postMessage` line in the script causes the random number to be output to this Events List interface. This is very useful for debugging scripts to make sure they are doing what they should be doing. Note the random number that was generated by this script.
18. In the left-hand navigation pane, select the Page that posts the new topic. Click the brown-colored POST message in the cover flow interface at the top of the window. In the Events List, expand the details for the "Sending message..." event. Scroll to the right and you should see that same random number show up in the POST data for this message.

Result Details

Element Status: Is All | Element Type: Is All

Completed - With No Errors Total Components: 130 Total Messages and Actions: 107 Error Components: 0

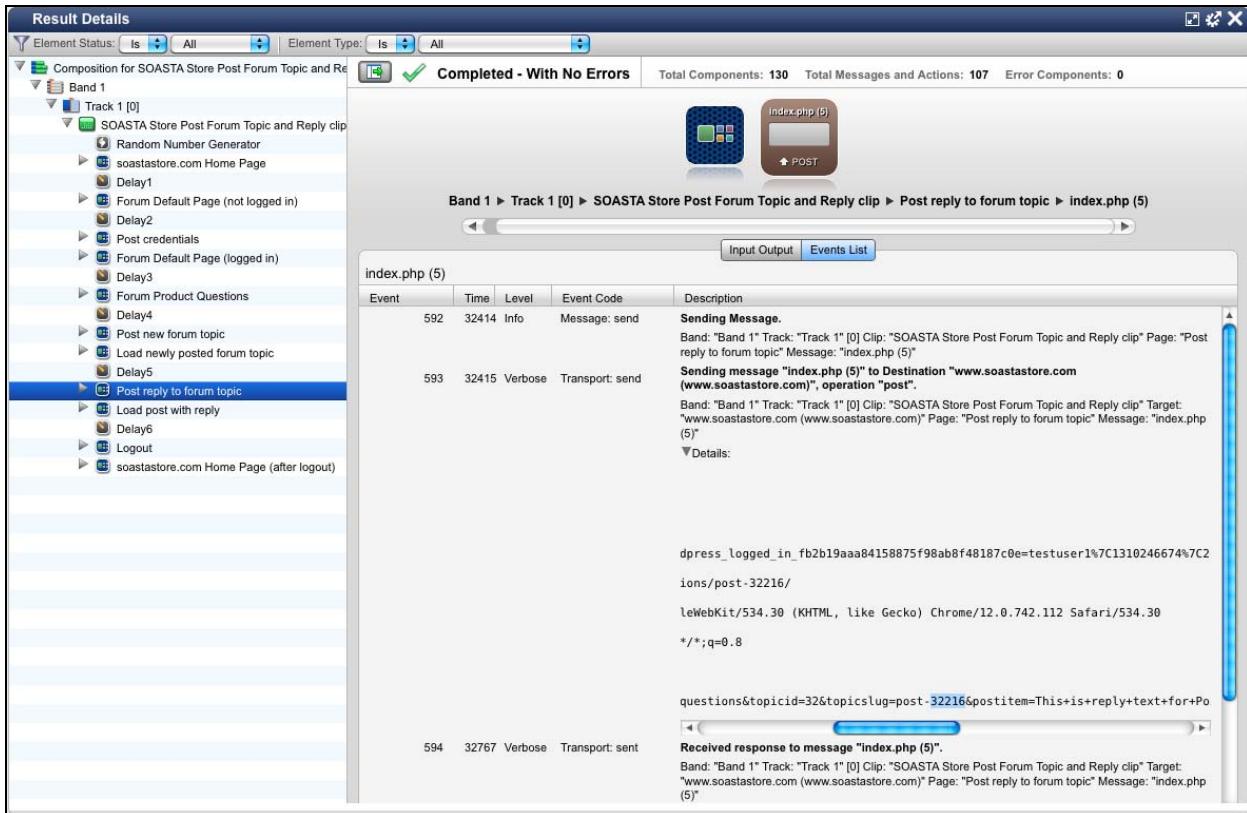
Band 1

- Track 1 [0]
 - SOASTA Store Post Forum Topic and Reply clip
 - Random Number Generator
 - soastastore.com Home Page
 - Delay1
 - Forum Default Page (not logged in)
 - Delay2
 - Post credentials
 - Forum Default Page (logged in)
 - Delay3
 - Forum Product Questions
 - Delay4
 - Post new forum topic
 - Load newly posted forum topic
 - Delay5
 - Post reply to forum topic
 - Load post with reply
 - Delay6
 - Logout
 - soastastore.com Home Page (after logout)

index.php (4)

Event	Time	Level	Event Code	Description
476	30391	Info	Message: send	Sending Message. Band: "Band 1" Track: "Track 1" [0] Clip: "SOASTA Store Post Forum Topic and Reply clip" Page: "Post new forum topic" Message: "index.php (4)"
477	30392	Verbose	Transport: send	Sending message "index.php (4)" to Destination "www.soastastore.com (www.soastastore.com)", operation "post". Band: "Band 1" Track: "Track 1" [0] Clip: "SOASTA Store Post Forum Topic and Reply clip" Target: "www.soastastore.com (www.soastastore.com)" Page: "Post new forum topic" Message: "index.php (4)" ►Details:
478	30751	Verbose	Transport: sent	s_logged_in_fb2b19aaa84158875f98ab8f48187c0e=testuser1%7C1310246674%7C2c6bd3 Kit/534.30 (KHTML, like Gecko) Chrome/12.0.742.112 Safari/534.30 =0.8 tions&newtopicname=Post+32216&postitem=This+is+sample+text+for+Post+32216&to Received response to message "index.php (4)". Band: "Band 1" Track: "Track 1" [0] Clip: "SOASTA Store Post Forum Topic and Reply clip" Target: "www.soastastore.com (www.soastastore.com)" Page: "Post new forum topic" Message: "index.php (4)" ►Details:

19. Repeat this process for the reply POST to make sure the same random number shows up in that post data.

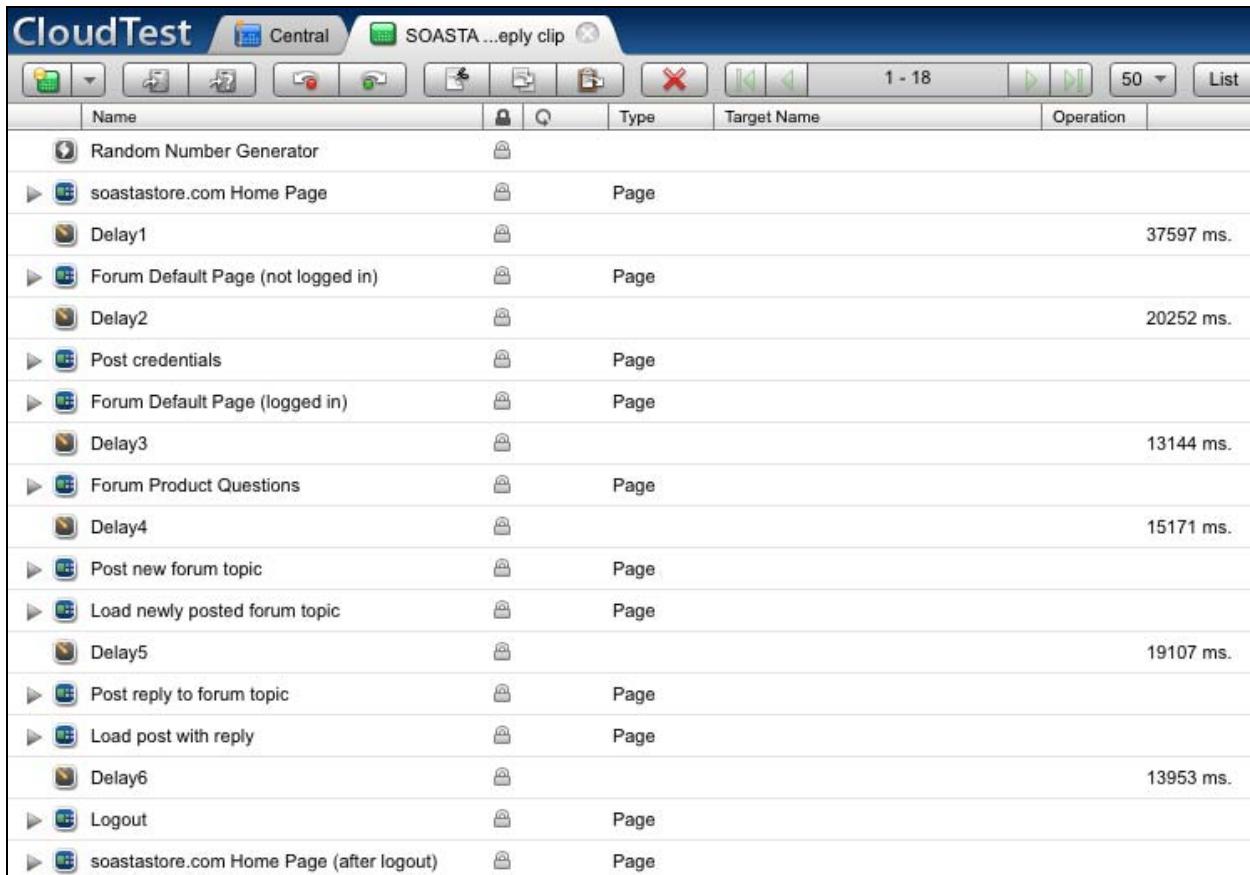


20. Once you've verified the test appears to be working from the CloudTest perspective, do a last verification by looking at the SOASTAStore.com Forum site. You should see your new post (look at the timestamp to confirm) with the same random number you saw in the CloudTest messages. Click on that topic and you should see the same random number in the topic text and reply text.

Using Transactions

Depending on a test's goals, you may need transaction information in addition to page-level information. Individual pages can be grouped into transactions so that dashboard metrics can be used to report on the performance of a transaction—in addition to the performance of individual pages. This section will walk you through the process of creating transactions.

1. Open the test clip you've been working on. It should look like this:



The screenshot shows the CloudTest Central interface with a toolbar at the top and a main table below. The table has columns for Name, Type, Target Name, and Operation. The rows list various test steps, including Random Number Generator, soastastore.com Home Page, Delay1, Forum Default Page (not logged in), Delay2, Post credentials, Forum Default Page (logged in), Delay3, Forum Product Questions, Delay4, Post new forum topic, Load newly posted forum topic, Delay5, Post reply to forum topic, Load post with reply, Delay6, Logout, and soastastore.com Home Page (after logout). Some steps have associated times listed to their right.

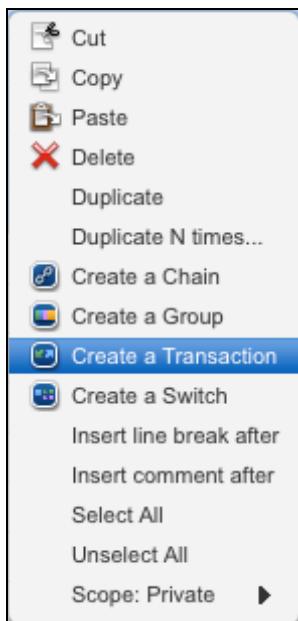
Name	Type	Target Name	Operation
Random Number Generator			
soastastore.com Home Page	Page		
Delay1			37597 ms.
Forum Default Page (not logged in)	Page		
Delay2			20252 ms.
Post credentials	Page		
Forum Default Page (logged in)	Page		
Delay3			13144 ms.
Forum Product Questions	Page		
Delay4			15171 ms.
Post new forum topic	Page		
Load newly posted forum topic	Page		
Delay5			19107 ms.
Post reply to forum topic	Page		
Load post with reply	Page		
Delay6			13953 ms.
Logout	Page		
soastastore.com Home Page (after logout)	Page		

2. You will create a transaction that includes loading the SOASTAStore.com home page, loading the Forum default page, logging in, and loading the Forum default page after login. Start by multi-selecting these pages and think times (delays) in the test clip.

CloudTest

Name	Type	Target Name	Operation
Random Number Generator			
soastastore.com Home Page	Page		
Delay1			37597 ms.
Forum Default Page (not logged in)	Page		
Delay2			20252 ms.
Post credentials	Page		
Forum Default Page (logged in)	Page		
Delay3			13144 ms.
Forum Product Questions	Page		
Delay4			15171 ms.
Post new forum topic	Page		

3. Right-click on the highlighted area and choose “Create a Transaction” from the menu.



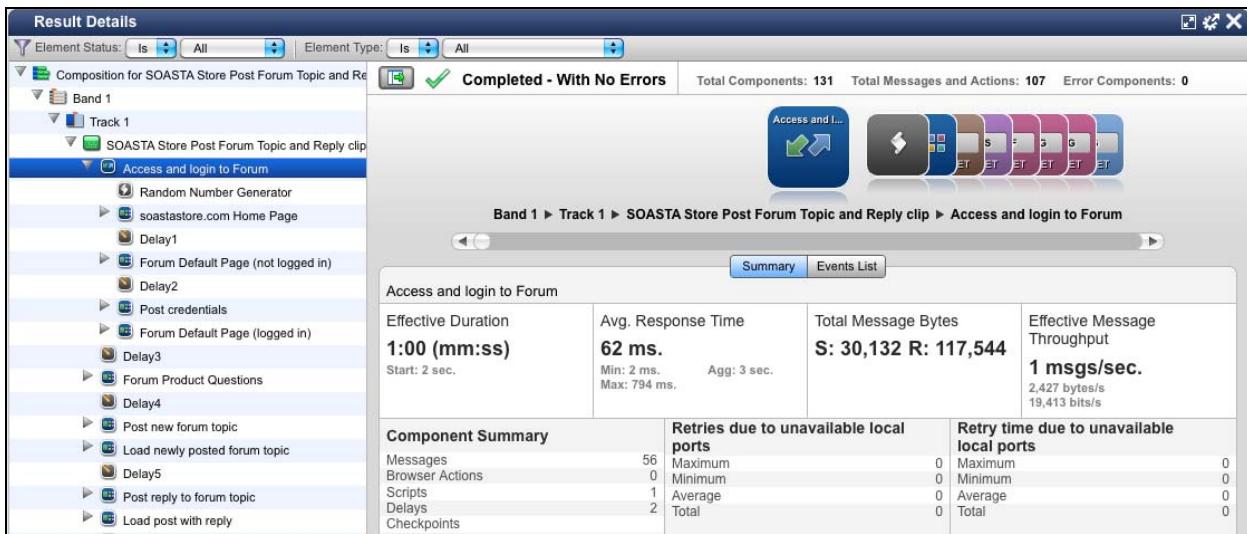
4. A transaction with a default name of “Transaction1” has been created.

Name	Type	Target Name	Operation
Transaction1	Transaction		
Random Number Generator			
soastastore.com Home Page	Page		
Delay1			37597 ms.
Forum Default Page (not logged in)	Page		
Delay2			20252 ms.
Post credentials	Page		
Forum Default Page (logged in)	Page		

5. Rename the transaction to *Access and login to Forum*.

Name	Type	Target Name	Operation
Access and login to Forum	Transaction		
Random Number Generator			
soastastore.com Home Page	Page		
Delay1			37597 ms.
Forum Default Page (not logged in)	Page		
Delay2			20252 ms.
Post credentials	Page		
Forum Default Page (logged in)	Page		

6. Save the test clip.
7. Repeat this process, as desired, to add additional transactions to this test clip.
8. Save the test clip.
9. Transaction-based metrics will now be available in the CloudTest dashboards. Here is the Result Details Dashboard with the transaction we created above (*Access and login to Forum*).

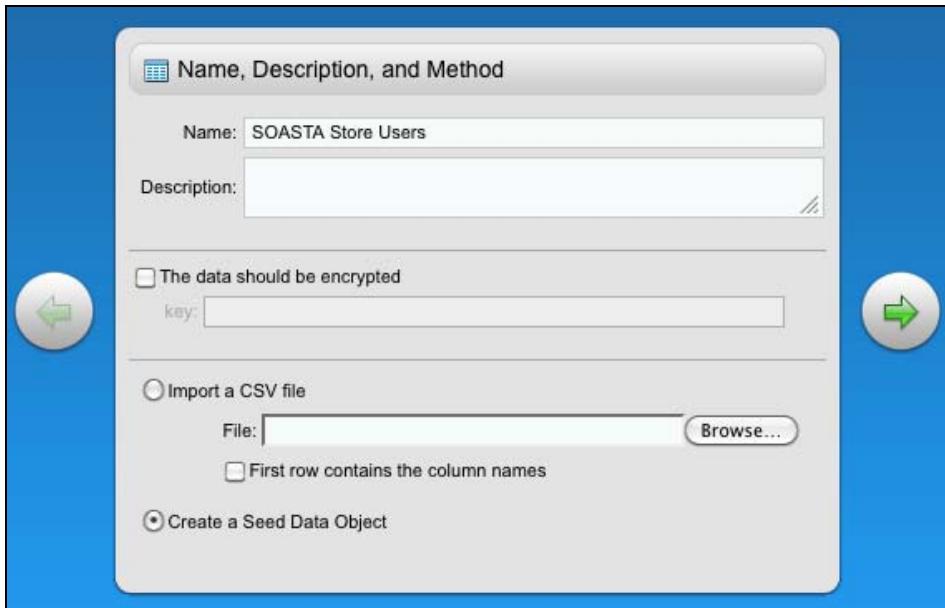


Using Seed Data to Parameterize data

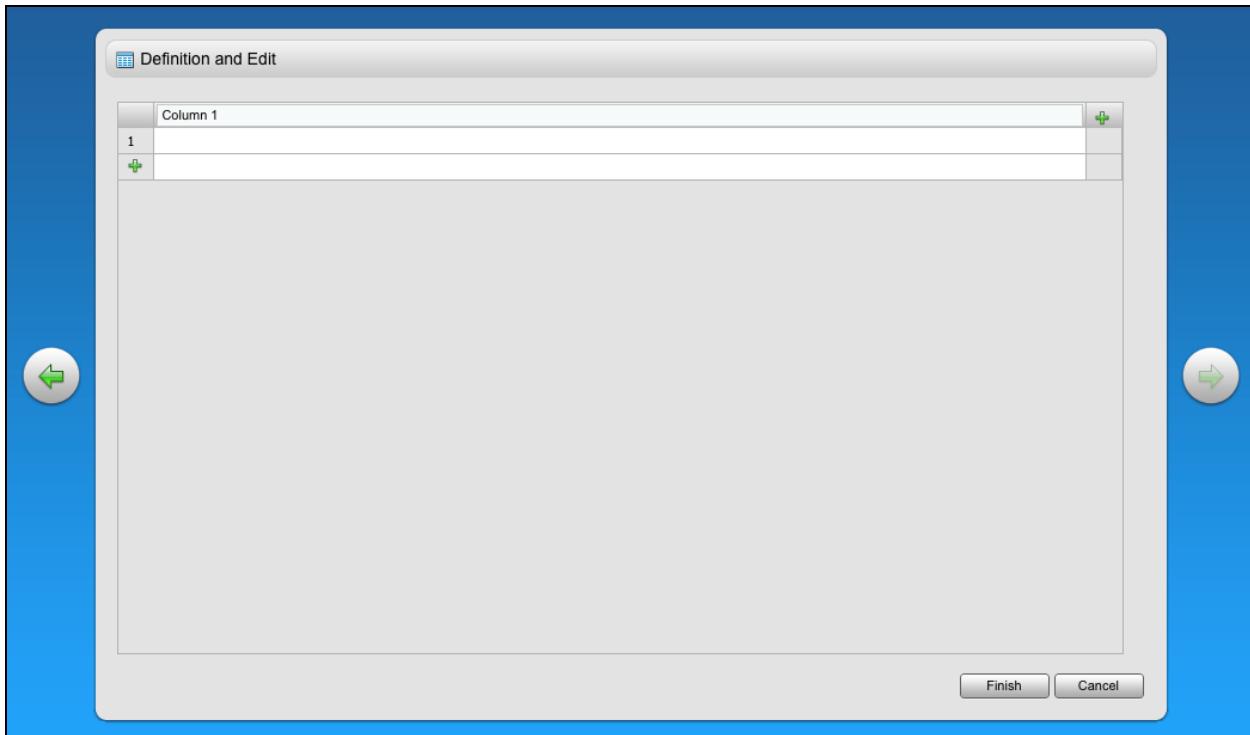
Seed data is used to parameterize a test clip. This tutorial will parameterize the user account. Once seed data has been setup, each test clip will login with a unique user account. There are three primary steps to setting up Seed Data: (1) create the seed data object in the CloudTest repository, (2) setup the clip in the composition to access the Seed Data object, and (3) configure the test clip to use the clip property created in the composition.

1. From Central, click the “Seed Data” item on the left.
2. Click the new icon () and the Seed Data wizard appears.

3. In the Name field, enter: *SOASTA Store Users*



4. Leave the other defaults alone and click the green arrow on the right.



5. Replace the default "Column 1" text with "username".

Definition and Edit	
	username
1	

- In the field to the right of the “1” and under the username column, enter “testuser1”. Click the green plus sign to add an additional row to the table. Repeat this process for each of the usernames: testuser2, testuser3, testuser4, and testuser5. It should look like the following when done:

Definition and Edit	
	username
1	testuser1
2	testuser2
3	testuser3
4	testuser4
5	testuser5

- Click the Finish button.
- Return to the clip in the Clip Editor.
- In the lower pane, click the Properties tab and then select “Clip Custom Properties” sub-tab on the left. You should see the clip custom properties created using the Session Template Package Wizard.

The screenshot shows the Clip Editor interface with the Properties tab selected. On the left, there's a sidebar with options: Messages/Actions, Scripts, Clips, Properties, and Results. The Properties tab is active. In the main pane, the "Clip Custom Properties" sub-tab is selected. A property named "_wpnonce" is listed, with its value type set to "Constant". Below the property, there are three checkboxes: "Save value for analytics", "Property value is tied to the virtual user", and "New value per virtual user iteration". There is also a "Description:" field at the bottom.

10. Scroll to the last custom property in the Clip Custom Properties panel on the right, and then click the green Plus (+) icon to add a new Clip Custom Property.

The screenshot shows the 'Custom Properties' section of a configuration interface. A new property is being added with the name 'clipProp' and a constant value. Advanced options include saving the value for analytics and tying it to a virtual user.

Property name:	clipProp	+	X	
Value type:	Constant			
<input type="checkbox"/> Save value for analytics				
<input type="checkbox"/> Property value is tied to the virtual user				
<input type="checkbox"/> New value per virtual user iteration				
Description:				

11. In the Property name field, enter: *username*

12. In the Value type drop-down, select: *Seed Data: Repository*

The screenshot shows the 'Custom Properties' tab with a property named 'username' selected. The value type is set to 'Seed Data: Repository' with the seed data object 'SOASTA Store Users'. Advanced settings allow specifying maximum rows per server, rows per track, column count, and starting row.

General	Repeat	Page Resource Settings	Delays	Custom Properties
Property name: username + X				
Value type: Seed Data: Repository Seed data object: SOASTA Store Users				
Advanced: Maximum rows per server: [] (default: 2000) Rows per track: [] (default: 1) Column count: [] (default: 1) <input type="checkbox"/> Use a subset of the seed data Starting Row: [] Number of rows: []				

13. In the Seed data object field, select *SOASTA Store Users*.

14. Leave the other defaults unchanged.

15. Save the clip.

16. If it isn't open already, open the test composition.

17. Open the Page that contains the login POST. In this case, it is the *Post credentials* Page in the *Access and login to Forum* transaction.

18. Double-click the HTML Document inside this page to see the POST data.

The screenshot shows the CloudTest interface with a transaction named "Access and login to Forum". The transaction details pane shows various steps: "soastastore.com Home Page", "Delay1", "Forum Default Page (not logged in)", "Delay2", "Post credentials" (selected), "Delay3", "Forum Product Questions", "Delay4", "Post new forum topic", "Load newly posted forum topic", "Delay5", and "Post reply to forum topic". The "Post credentials" step is expanded to show an "HTML Document" named "wp-login.php (2)". The document properties are: Type: HTML, Target Name: www.soastastore.com, Operation: post, URL: /wp-login.php, and Query String: action=login&view=forum. The "Inputs" section under "Operation: post" shows URL Values: /wp-login.php, Query String: action=login&view=forum, and Data: log=testuser1&pwd=soasta&submit=Login&redirect_to=%2F%3Fpage_id%3D167.

19. Review the POST data.

The screenshot shows the "Data" tab in CloudTest. The data field contains the POST data: log=testuser1&pwd=soasta&submit=Login&redirect_to=%2F%3Fpage_id%3D167.

20. We want to parameterize the "testuser1" text. To do that, replace the "testuser1" text with the property reference to the track property we just created. In this case, enter: `{%% prop : parent-clip : /username %%}`

21. Your POST data should look like this:

The screenshot shows the "Data" tab in CloudTest. The data field contains the parameterized POST data: log=[%% prop : parent-clip : /username %%]&pwd=soasta&submit=Login&redirect_to=%2F%3Fpage_id%3D167.

22. Save the test clip.

23. Switch back to the composition.

24. Increase the number of virtual users on the track to 5 by clicking the virtual user icon () and entering "5" in the field to the right of the icon.



25. Save the composition.
26. Play the composition (remember to disable delays/think times).
27. Inspect each of the login POSTs to confirm that unique user accounts are being used in the post messages. In the example below, testuser3 is correctly being parameterized in this test.

Result Details

Element Status	Element Type	Operation
All	All	All

Completed - With No Errors Total Components: 1 Total Messages and Actions: 1 Error Components: 0

wp-login...

Band 1 > Track 1 [1] > SOASTA Store Post Forum Topic and Reply clip > Access and login to Forum/Post credentials > wp-login.php

Input Output Events List

General

Name: wp-login.php

Start Time	Response Time	Total Time	Total Message Bytes
11.209 sec.	713 ms. TTFB: 713 ms.	714 ms.	Sent: 1,046 Received: 1,045 (gzipped)

Custom Properties

No property changes

Input

```
POST /wp-login.php?action=login&view=forum HTTP/1.1
Host: www.soastastore.com
Content-Length: 69
Content-Type: application/x-www-form-urlencoded
Accept-Language: en-us
Cookie: website_mobile_detector=47C0%7Ce51a1cbbcf87a32ce8d233c
Referer: http://www.soastastore.com/?page_id=167
Accept: */*
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/29.0.1471.64 Safari/537.36
Origin: http://www.soastastore.com
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*
Pragma: no-cache
Proxy-Connection: keep-alive
```

Output

```
HTTP/1.1 302 Found
Cache-Control: no-cache, must-revalidate, max-age=0
Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8
Date: Mon, 14 Jan 2014 18:57:01 GMT
Expires: Mon, 11 Jan 1964 03:00:00 GMT
Location: ?page_id=167
Pragma: no-cache
Server: Apache/2.2.22 (Ubuntu)
Set-Cookie: wordpress_test_cookie=WP-Cookie-check; path=/
Set-Cookie: wordpress_f2b219aaaa84158875f98ab8d48187c0e=teastuser3
Set-Cookie: wordpress_f2b219aaaa84158875f98ab8d48187c0e=teastuser3
Set-Cookie: wordpress_logged_in_f2b219aaaa84158875f98ab8d48187c0e=teastuser3;
Set-Cookie: session_f2b219aaaa84158875f98ab8d48187c0e=teastuser3;
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-Powered-By: PHP/5.3.10-jubuntu3.6
Content-Length: 20
Connection: keep-alive
```

Operation: post

```
POST /wp-login.php?action=login&view=forum HTTP/1.1
Host: www.soastastore.com
Content-Length: 69
Content-Type: application/x-www-form-urlencoded
Accept-Language: en-US,en;q=0.8
Cookie: PHPSESSID=avaugpsgbmu8dphgdla0hoojp3
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
Referer: http://www.soastastore.com/?page_id=167
Accept-Encoding: gzip,deflate,sdch
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/534.30
Origin: http://www.soastastore.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Proxy-Connection: keep-alive
Cache-Control: max-age=0
```

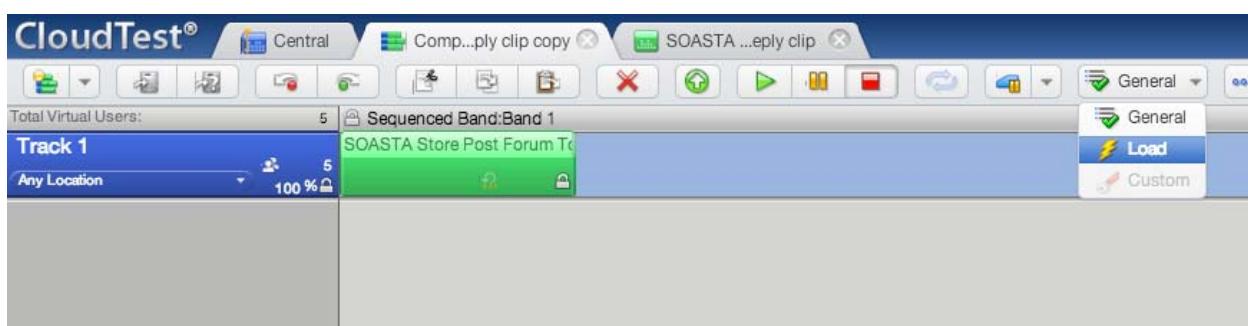
URL Values	/wp-login.php
Query String	action=login&view=forum
Data	log=testuser3&pwd=soasta&submit=Login&redirect_to=%2F%3Fp

Test Execution

Setting Play Mode for Load Testing

CloudTest provides play modes that are recommended settings for both debug (General) and load (Load) testing. General play mode provides verbose logging settings that enable you to debug tests. Load play mode omits verbose logging, but captures statistical information in real-time to help identify performance bottlenecks in a web application.

1. Before you run any load tests, in the Composition Editor you need to click the Play Mode drop-down and choose Load. This will run the test in an optimized way for load tests. Note that the debugging dashboard (Result Details) is not available in Load play mode. You will need to use other dashboards to review your data while in Load play mode.



Adding System Dashboards to tests

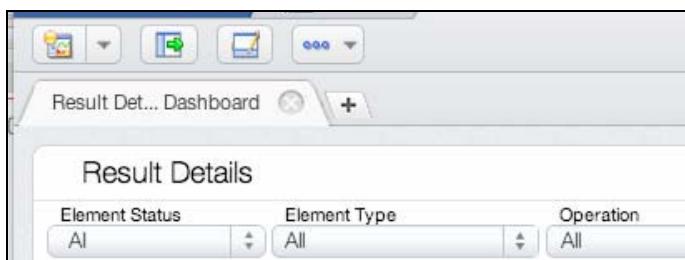
While running a test, there are a number of different dashboards you might use (load summary, error analysis, bandwidth analysis, etc.) to identify how a test is running and to identify any performance bottlenecks. The steps below show you how to add a dashboard, the Load Test Summary Dashboard, to the test composition. This dashboard is one of several built-in Dashboards available for you to use while running load tests. You can also build your own custom dashboards, but the steps for doing this are beyond the scope of this tutorial.

Note: The Result Details Dashboard's detailed information is only available if you've run a test with the "general" test mode. This mode is not available for load tests—only for test development and debugging. No more than 10 virtual users should be run in General mode.

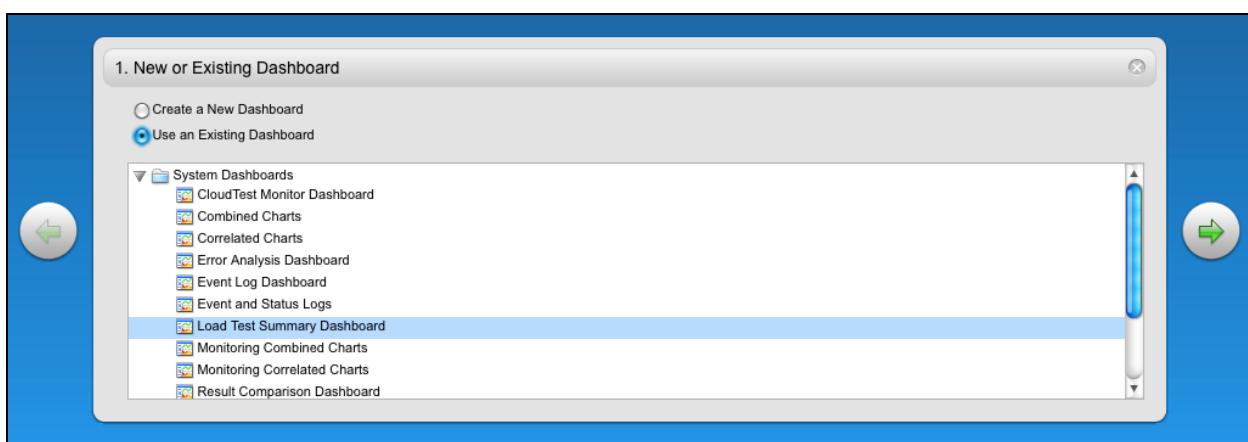
1. To add a dashboard, click the Play tab in the top right of the Composition Editor.



2. Click the Plus icon to the right of the Result Details dashboard. The New or Existing Dashboard wizard appears.

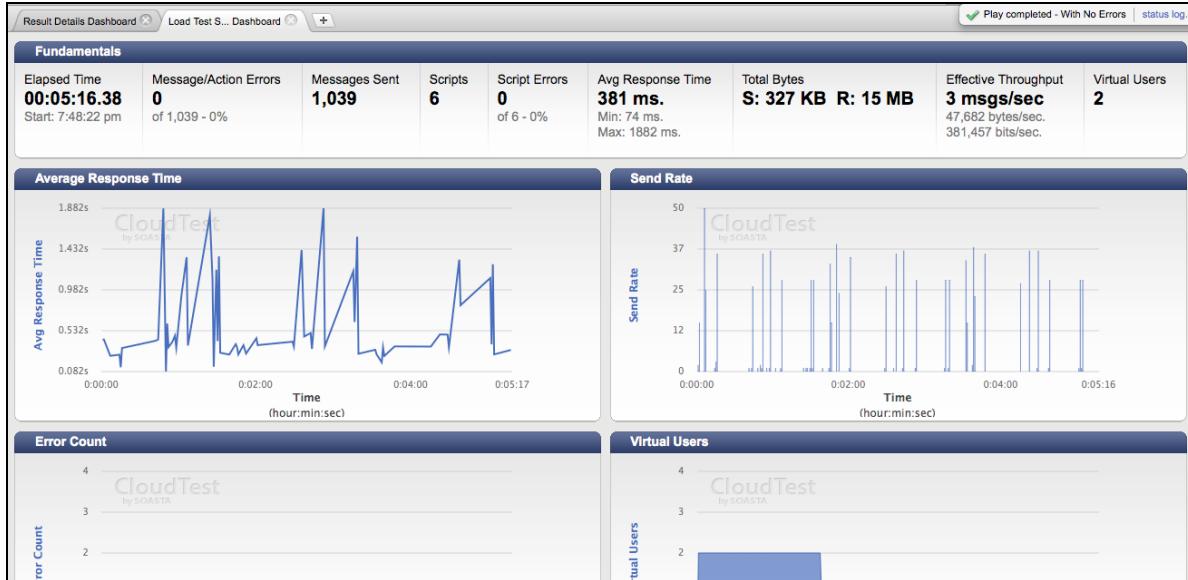


3. Click the Load Test Summary Dashboard from the System Dashboards list and click the right green arrow.



- The dashboard is added to the Composition Editor tab. The data from the last result for the test composition populates the widgets in the dashboard. If you are running a test when you add a new dashboard, the data for the currently running test will populate into this dashboard in real-time.

- Click Play in the Composition Editor toolbar.



- While the test composition plays, each of the widgets displays various animation effects indicating incoming data and reflects the real-time status of the test.
- Mouse over and click any item in the Average Response Time or Send Rate charts to view detail information.



- To zoom on a region of any chart, click-and-drag a rectangle. This zooms in on the selected chart AND all the other charts in the dashboard. The timestamps paired to make this deep investigation of the data possible. Double-click to un-zoom the chart.

This has been a brief introduction to adding Dashboards to a test composition. For more information about configuring dashboards and all the metrics available to analyze test results, please see CloudLink.

Advanced exercise

To enhance your understanding of CloudTest, try these advanced enhancements to the test:

- Instead of always choosing the same forum to post to, randomly choose a forum from the list of 4.
- Use a script to add a date/timestamp to the forum post content.

Next Steps...

Many topics are outside the scope of this introductory Load Test Creation Tutorial and deserve attention. Below are brief descriptions of the topics, but more information can be found online in CloudLink.

Pages versus Transactions

In this tutorial you created a test that utilized Pages. As described above, this is a grouping that most closely represents how a browser sends HTTP requests to a web application. An HTML document is retrieved, and then 6 threads in parallel retrieve the resources for that page. This is the most common way to create tests in CloudTest. It is important to note, however, that another grouping mechanism is available in test clips: Transactions.

Transactions provide a means to logically group and capture metrics for one or more tasks in a process. Transactions permit discrete business processes to be grouped accordingly. For example, a multi-step checkout process from an ecommerce website. In a site that includes a shopping cart, a user logs into that site, adds an item to their cart, and checks out. Each of these steps can be thought of as a transaction—the performance of which can be measured as a transaction.

Unlike pages, which can contain only one HTML page and which treat each HTML document as separate steps in a process, a Transaction permits all the parts of a logical group to be tested together.

Once logically grouped, the Transaction provides a convenient means to measure the amount of time it takes a server to perform a given defined task or set of tasks.

Note that it isn't an either/or situation with Pages versus Transactions. Many tests utilize a combination of both Pages and Transactions. For example, the Transaction shown above (in Icon view) contains multiple CloudTest pages that equates to a complete tasks in a web application. By constructing the test as a Transaction, metrics about the specific "transaction" (e.g. submitting a form) can be captured. Among the metrics captured is the server response time it takes to submit all or part of a Transaction. An individual Transaction can be measured again at various increments of load. The resulting metrics are useful in identifying potential bottlenecks as an outcome of testing discrete business processes. The **Transaction** above is shown in List view below..

Building Dashboards

Real-time analytics are the cornerstone of the CloudTest solution. The goal of any test is to understand how an application performs under differing conditions. The only way to determine how an application is performing during a test is by using real-time dashboards. Everyone looks for different things in a dashboard, so CloudTest provides an intuitive and flexible way for every user to create and edit their own dashboards. All metrics are available in real-time and can be exported when tests are complete.

As a CloudTest user who will be running tests, it is critical you understand which metrics (both test-related and monitoring-related) are available within CloudTest. This will make you more efficient while running a test and help you identify performance bottlenecks quicker. Take time to review each of the metrics available in CloudTest and understand what that data means and how to interpret it during a test.

Dynamic Pages

This tutorial used static pages in the test clip. It is a good idea to spend some time working with Dynamic Pages. As described above, these are very different as they dynamically parse out the Resources (CSS, JS, images, etc.) in an HTML document. It takes some time to become familiar with correctly filtering Dynamic Pages to make sure they are making the HTTP requests the test goals require (and more importantly, not making the requests for resources the test goals explicitly exclude).

If-Then-Else and Switch

The Clip Editor supports a visual mechanism for doing if-then-else and switch statements within a test. If you plan to include conditional logic in a test, these are important topics to review. These mechanisms provide the ability to create very complex test cases, but spend the time to make sure you understand how to use the features correctly.

Monitoring

Monitoring is a key part of any test. Monitoring data provides back-end metrics detailing the impact a test is having on the application or application infrastructure and helps the tester identify possible performance bottlenecks. CloudTest provides its own monitoring tools, in addition to integrating with 3rd-party tools like CA's Wily Introscope product. Spend some time installing a couple monitors and reviewing the Monitoring FAQ document to become familiar with the capabilities of CloudTest monitoring features.

SOASTA, Inc.
444 Castro Street, 4th Floor
Mountain View, CA 94041
866.344.8766
<http://www.soasta.com>