SOASTA

TouchTest^{TM} Web for Android Tutorial

SOASTA TouchTest™ Web for Android Tutorial

This document is for informational purposes only. SOASTA makes no warranties, express or implied, as to the information contained within this document.

Table of Contents

Why Mobile Web Testing?

CloudTest® Mobile, featuring TouchTest™ technology, delivers for the first time, complete functional test automation for mobile web sites using browser on your Android device. TouchTest™ technology delivers fast, precision functional testing while increasing the stability of automated tests across releases.

The TouchTest™ Web app makes web sites viewed with Android browser testable. TouchTest Web is launched using a lightweight web service called TouchTest™ Agent. Devices can be dedicated to testing in the lab, used as part of an external test, or crowd-sourced as part of a high volume, globally distributed test.

CloudTest Mobile support is provided for recording user actions within any Android 5.0 or greater device including iPhone, iPad, and iPod Touch. There is no need to jailbreak the Android device and the device can be untethered.

CloudTest® Basics

SOASTA CloudTest® provides fast, effective performance, load and functional test automation of any modern Web application, Web service, or mobile application in a lab, staging or production environment using unique visual programming and multi-track user interfaces. The CloudTest platform can utilize both public and private cloud resources to assure any web or mobile application won't fail under peak user traffic.

Central is the primary user interface in which CloudTest® runs, and contains CloudTest's primary features, organized by sections.

Central's first section—highlighted in light blue—contains the Welcome page, and the primary test building tools.

The **Composition** is the test itself as presented in the **Composition Editor**, and contains one or more **Clips** arranged on **Tracks** and governed by user-specified sequence and tempo. The Composition Editor is a player, debugger, and the dashboard where results are analyzed.

The **Clip** is the basic building block of a test as built in the **Clip Editor** and has a **Target** such as a web site, or in the case of TouchTest™, a mobile app. A clip can be thought of as a visual script that is composed of a series of timed or sequenced events, which correspond to gestures performed on the mobile device. It can contain messages, app or browser actions, and scripts, as well as delays and checkpoints—all of which can be organized into containers (i.e. groups, chains, transactions, etc.

TouchTest™ clips are recorded directly from the mobile app and added to the clip as you perform them on the mobile device.

What Does Touch Test™ Web Record?

TouchTest™ Web records the details of actual browser gestures and events on the mobile web site under test. These gestures and events are represented within the CloudTest/Lite **Clip Editor** as Browser Actions.

Each gesture you perform on a TouchTest Web-enabled device is recorded and added to the test clip as a browser action.

Like any clip element within CloudTest®, browser actions have attributes such as inputs and outputs, properties, waits, and validations that can be parameterized. Additionally, a browser action can be added to any containers (e.g. if-then-else, switch, chains, transactions, and groups).

TouchTest Web Prerequisites (Android)

TouchTest™ Web recording is performed by any user by accessing or deploying the following components on the desktop and on a Android device running 2.3.3 or later:

- The CloudTest® (or CloudTest® Lite instance) where the user has rights and will login to start recording.
- The TouchTest Web app installed onto the device.
- The TouchTest™ Agent app installed onto the device and pointed at the same CloudTest®/Lite instance from which testing will occur.

  Note: Both the TouchTest Agent and TouchTest Web apps will be installed from the CloudTest, Welcome page onto the given Android device.
- The Device Agent (i.e. on the CloudTest or CloudTest Lite instance) must have Mobile Web Capable set.
- The web page or site to test. For this tutorial, www.soasta.com is used as an example.

After the prerequisites are performed, we will easily run the TouchTest Web mobile app on a device, after which the remainder of this tutorial provides instructions for recording and playback of a fully functional test composition.

CloudTest or CloudTest Lite Access

The steps in this tutorial require a CloudTest user account on any CloudTest Edition, including a CloudTest Lite instance (with its default user account).

TIP: CloudTest Lite is a free downloadable edition of SOASTA CloudTest available as a VMWare® image that can be run in any virtual machine software that can open it. CloudTest Lite provides one free user account that can be used with one mobile device.

Android Device and Architecture Support

SOASTA TouchTest™ supports Android SDK 2.3.3 (Gingerbread) or later and the following devices.

- Samsung Galaxy SIII
- HTC One X (AT&T) or HTC One S (T-Mobile)
- Samsung Nexus 7
- Samsung Nexus
- Samsung Galaxy SII
- HTC EVO 4G (aging but still very popular. Equivalent and more recent: HTC Thunderbolt)
- HTC Desire (HTC Inspire from AT&T)

- Galaxy Tab 2
- Samsung Galaxy S
- Droid RAZR 2
- Motorola Droid X

Android Developers should note that the SOASTA TouchTest™ Driver is not compatible with the armv6 architecture. The following architectures are supported:

- armv7
- armv7s

Configuring the Android Device

Your Android device should have the following Settings:

- In the device settings, tap Developer Options and check the USB Debugging box.
- In the device settings, tap "Security" and then tap to check the Unknown sources box.

Note: Both TouchTest Web and TouchTest Agent must be installed on the device. Instructions to install both are included below.

Prerequisites for Android using Eclipse (Developer Only)

This guide presumes that the Android developer has a basic familiarity with the Eclipse development environment and has the following configuration:

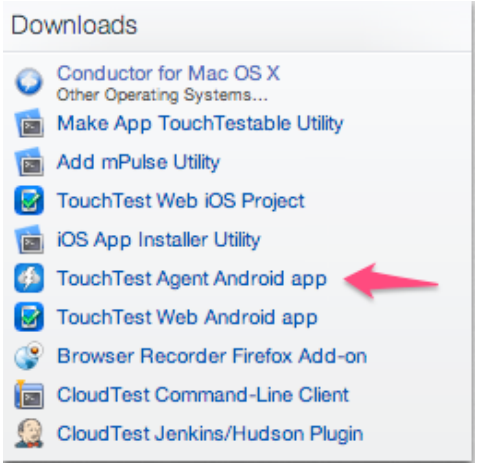
- Eclipse (version 3.6 or later) with Android SDK installed along with the Eclipse plugin. The Minimum Android Version supported for use with TouchTest™ is 2.3.3 (Gingerbread).

Installing TouchTest Agent & TouchTest Web on a Device (All Users)

The TouchTest™ Agent is responsible for launching the apps that are being tested on a given device. This Android app runs in Android devices running 2.3.3 or later. If these steps have already been performed on a given device, proceed to Recording a TouchTest Scenario below. The TouchTest Web app facilitates web action recording from the device browser.

To get started, install TouchTest Agent onto the mobile device and then perform the one-time registration steps that will enable your device for use with TouchTest, after which continue on to install TouchTest Web.

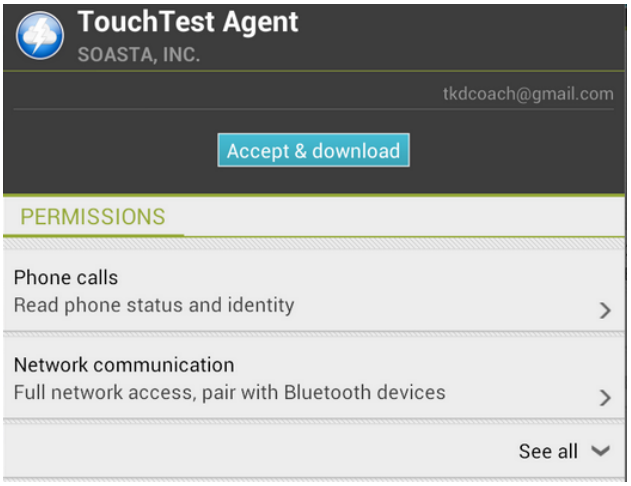
1. On your Android device, download the TouchTest Agent Android app from the Welcome page.



Downloads

Conductor for Mac OS X
Other Operating Systems...
Make App TouchTestable Utility
Add mPulse Utility
TouchTest Web iOS Project
iOS App Installer Utility
TouchTest Agent Android app
TouchTest Web Android app
Browser Recorder Firefox Add-on
CloudTest Command-Line Client
CloudTest Jenkins/Hudson Plugin

2. Tap on the downloaded TouchTestAgent.apk file (usually found in the device's notification tray) to start Install.
3. Tap TouchTest Agent to open the detail page and then tap.
4. When the installer appears, tap Accept & download to proceed.



TouchTest Agent
SOASTA, INC.

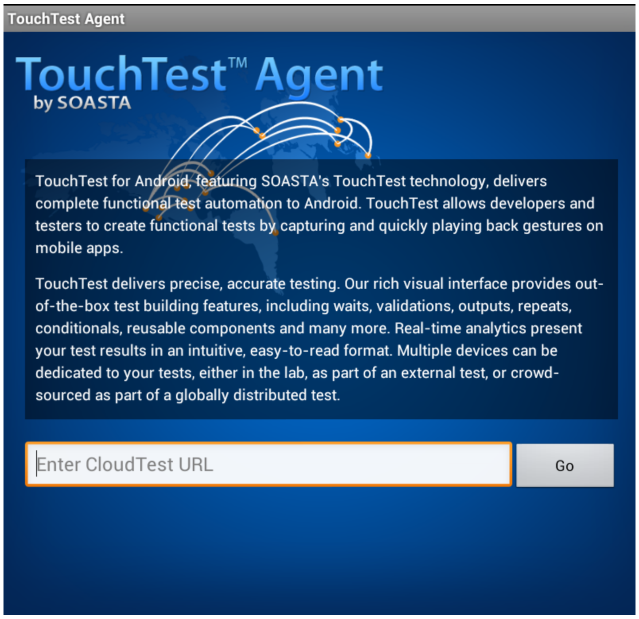
tkdcoach@gmail.com

PERMISSIONS

Phone calls
Read phone status and identity

Network communication
Full network access, pair with Bluetooth devices

5. Once install completes, tap Open to launch the TouchTest Agent app on the Android device. When you do so, the app launches and the TouchTest Agent splash screen displays.



TouchTest Agent

TouchTest™ Agent
by SOASTA

TouchTest for Android, featuring SOASTA's TouchTest technology, delivers complete functional test automation to Android. TouchTest allows developers and testers to create functional tests by capturing and quickly playing back gestures on mobile apps.

TouchTest delivers precise, accurate testing. Our rich visual interface provides out-of-the-box test building features, including waits, validations, outputs, repeats, conditionals, reusable components and many more. Real-time analytics present your test results in an intuitive, easy-to-read format. Multiple devices can be dedicated to your tests, either in the lab, as part of an external test, or crowd-sourced as part of a globally distributed test.

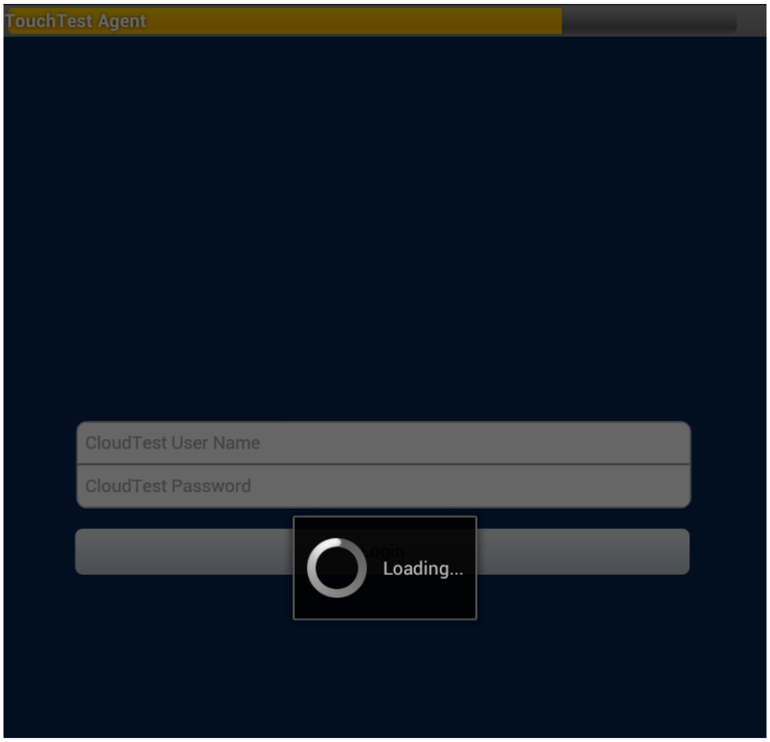
6. Enter the CloudTest URL for TouchTest, which is your own server with `/concerto/touchtest` appended:

   `http://<CloudTest server>/concerto/touchtest.`

   For example, a user with an Apple router running CloudTest Lite might connect to:

   `http://10.0.1.9/concerto/touchtest`

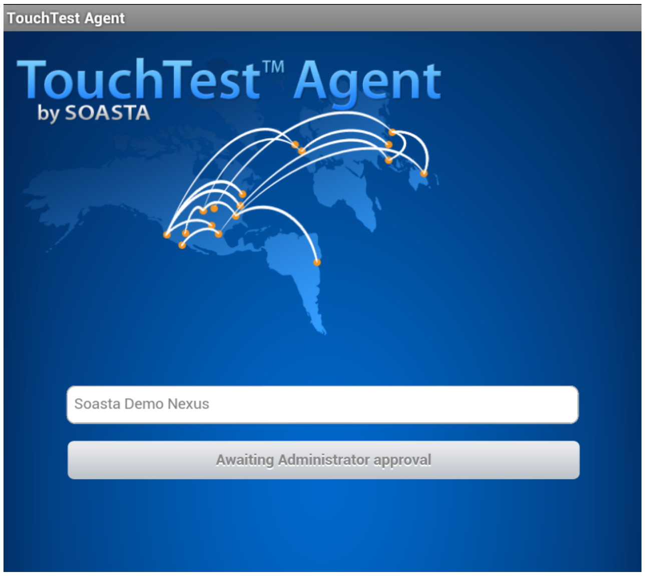
7. Tap Go. The TouchTest Agent opens the provided URL and a spinner appears as the app auto-registers itself.



8. Once the spinner disappears, enter your CloudTest user name and password and tap Login.
9. When prompted, give the TouchTest Agent a name. For example, *SOASTA Demo Nexus.* Note that this name will be used throughout the product to refer to this device. Once entered the device name can only be changed by an Administrator.
10. Click Submit for administrator approval.

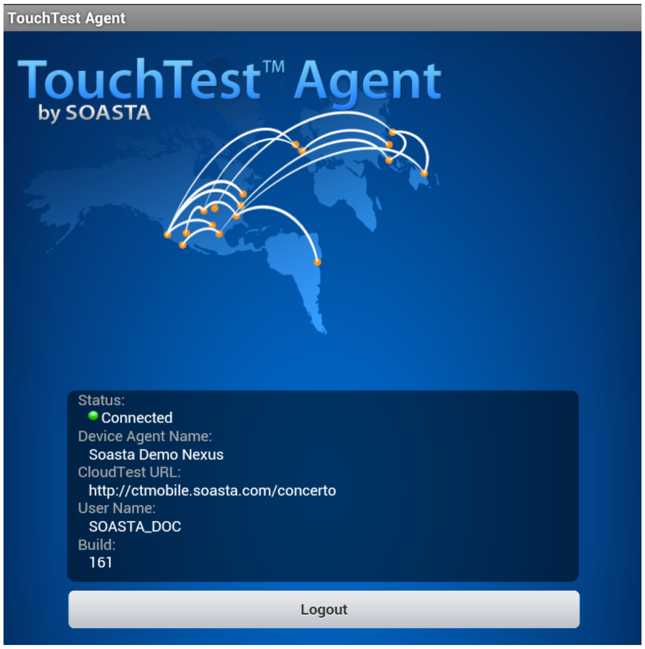
Note: **If you're a CloudTest Lite user**—you are already a Mobile Device Admin and you must approve this device yourself using the instructions in the next section.

Once the request for Administrator approval has been made, the TouchTest Agent will continue to poll CloudTest for approval.



Note: It is not necessary to keep the TouchTest Agent running while this approval is pending. The TouchTest Agent will resume polling for its approval once restarted.

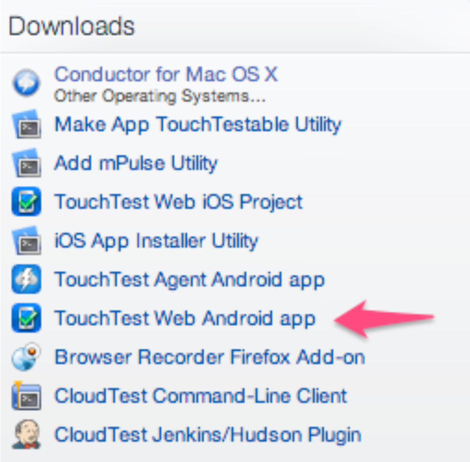
If the Mobile Device Administrator approves your device, the Connected status will appear. On subsequent launches click Login to Connect and Logout to Disconnect.



Status:
● Connected
Device Agent Name:
Soasta Demo Nexus
CloudTest URL:
http://ctmobile.soasta.com/concerto
User Name:
SOASTA_DOC
Build:
161

Note: In CloudTest Lite, the default user is also the Mobile Device Administrator for one device (covered below).

11. Return to the Welcome page in the mobile browser and then tap TouchTest Web Android app, and then tap Install.

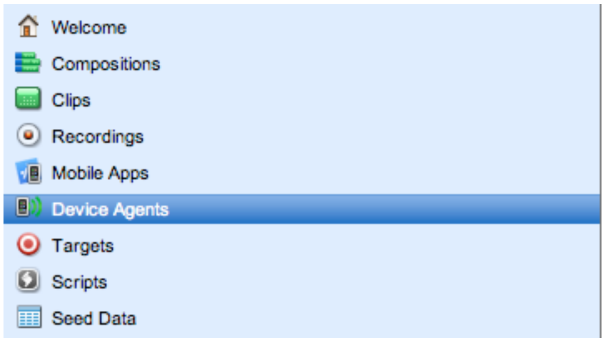

Downloads
Conductor for Mac OS X
Other Operating Systems...
Make App TouchTestable Utility
Add mPulse Utility
TouchTest Web iOS Project
iOS App Installer Utility
TouchTest Agent Android app
TouchTest Web Android app
Browser Recorder Firefox Add-on
CloudTest Command-Line Client
CloudTest Jenkins/Hudson Plugin

12. Tap on the downloaded TouchTestAgent.apk file (usually found in the device's notification tray) to start Install.
13. When the installer appears, tap Accept & download to proceed.
14. Once install completes, TouchTest Web opens. Exit and return to the TouchTest Agent app and ensure its status is "Connected" whenever you'd like to record or playback a mobile app test clip.

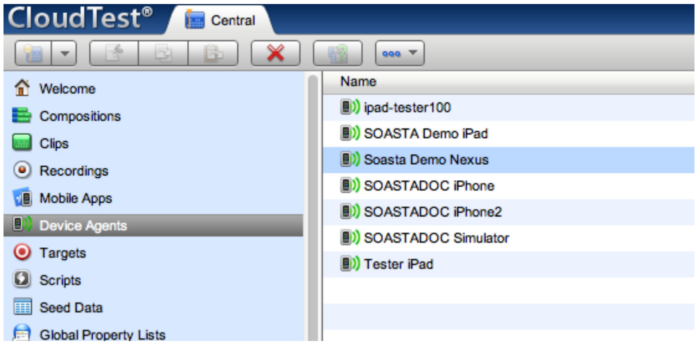
Approving a Mobile Device (Device Admin or CloudTest Lite user)

The TouchTest™ Mobile Device Administrator has the responsibility to approve or reject the devices attempting to join testing. If you are using CloudTest Lite, you are your own Device Administrator. Use the following steps to approve/reject the devices attempting to join.

1. Login as the user with mobile device administrative rights.
2. Click Central > Device Agents

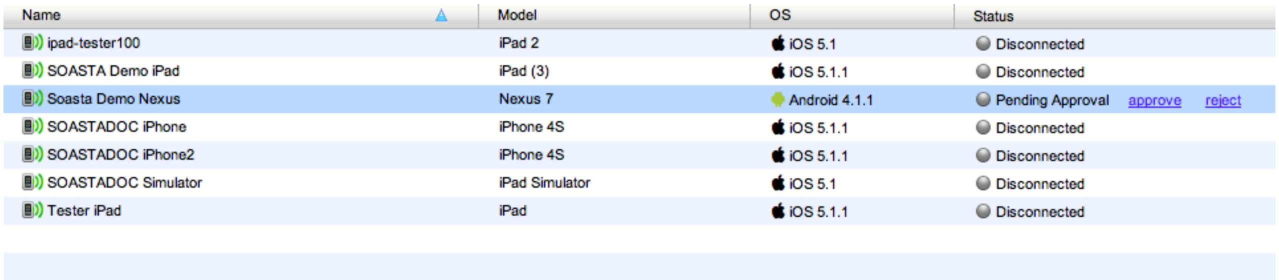


When you do so, the Device Agents list displays those devices in queue by name.



Name
ipad-tester100
SOASTA Demo iPad
Soasta Demo Nexus
SOASTADOC iPhone
SOASTADOC iPhone2
SOASTADOC Simulator
Tester iPad

Those devices that have the status Pending Approval need administrative attention.



Name	Model	OS	Status
ipad-tester100	iPad 2	iOS 5.1	Disconnected
SOASTA Demo iPad	iPad (3)	iOS 5.1.1	Disconnected
Soasta Demo Nexus	Nexus 7	Android 4.1.1	Pending Approval approve reject
SOASTADOC iPhone	iPhone 4S	iOS 5.1.1	Disconnected
SOASTADOC iPhone2	iPhone 4S	iOS 5.1.1	Disconnected
SOASTADOC Simulator	iPad Simulator	iOS 5.1	Disconnected
Tester iPad	iPad	iOS 5.1.1	Disconnected

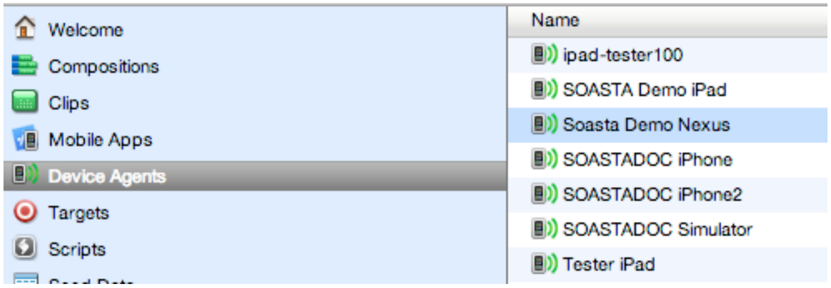
3. Click Approve to complete adding a device and Reject to deny its access.

    Note: CloudTest Lite/TouchTest Lite users may approve a single device only and that device cannot be removed. Approval should only be performed on the intended single device.

Setting Mobile Web Capable (All Users)

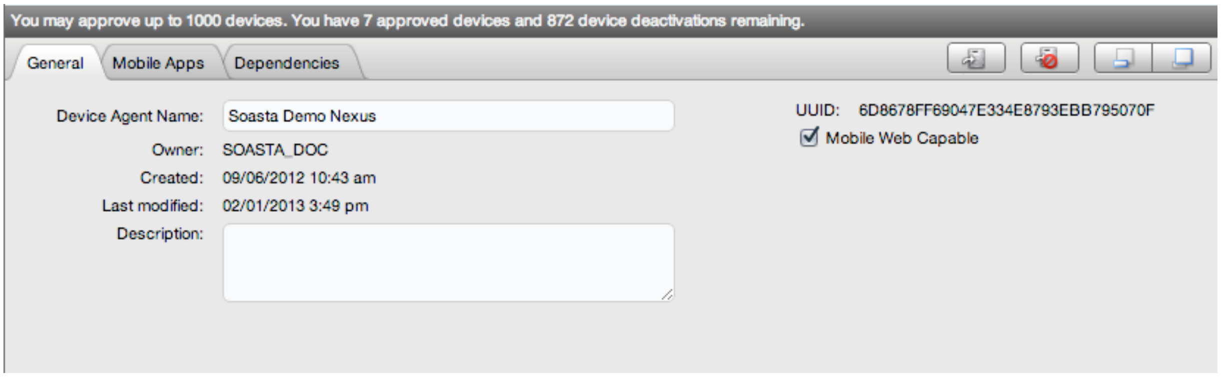
Before proceeding with the test creation steps, enable TouchTest Web for your Device Agent using the Mobile Web Capable checkbox. This should be done per device or simulator.

1. Locate your device agent by name in the Device Agents list.



Name
ipad-tester100
SOASTA Demo iPad
Soasta Demo Nexus
SOASTADOC iPhone
SOASTADOC iPhone2
SOASTADOC Simulator
Tester iPad

2. Check the Mobile Web Capable box in the lower panel, General tab.



You may approve up to 1000 devices. You have 7 approved devices and 872 device deactivations remaining.

Device Agent Name: Soasta Demo Nexus

Owner: SOASTA_DOC

Created: 09/06/2012 10:43 am

Last modified: 02/01/2013 3:49 pm

Description:

UUID: 6D8678FF69047E334E8793EBB795070F

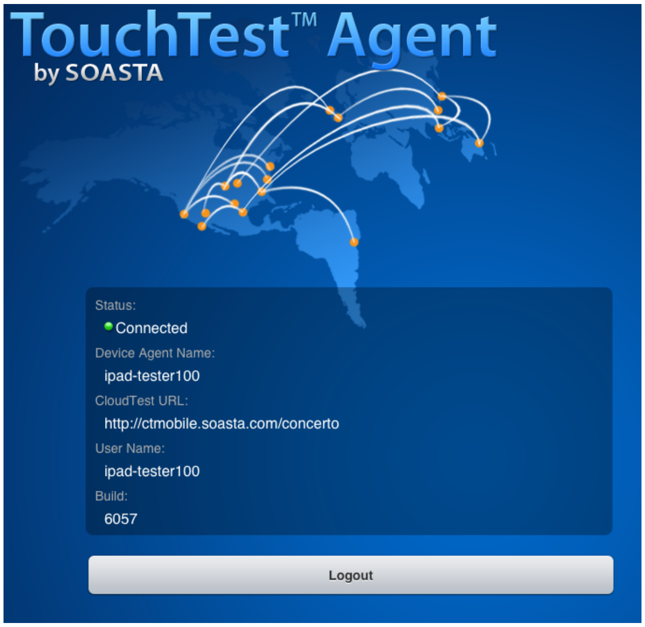
☑ Mobile Web Capable

3. Click the Save button on the lower panel toolbar.

Create a TouchTest™ Web Clip

Now that the device agent is mobile web capable, we can create a new clip using the following TouchTest Web recording techniques.

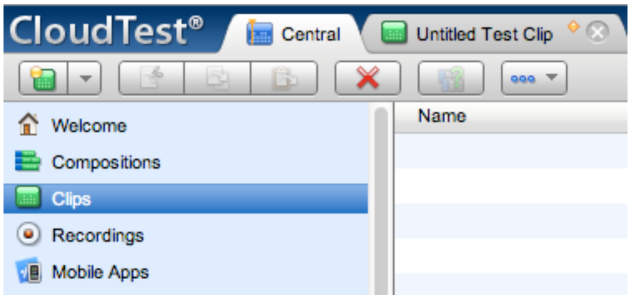
- Login to CloudTest on your desktop computer.
- Open the TouchTest Agent app on your mobile device, and click Login (your previously entered username and password will be auto-populated).



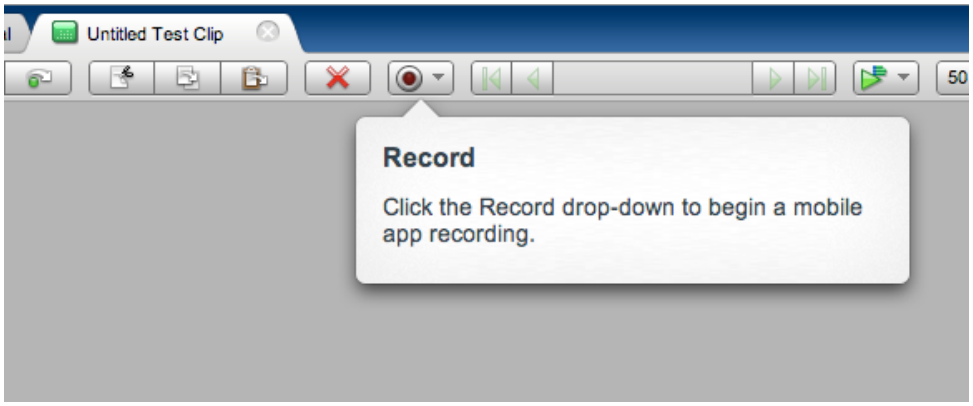
Status:
Connected
Device Agent Name:
ipad-tester100
CloudTest URL:
http://ctmobile.soasta.com/concerto
User Name:
ipad-tester100
Build:
6057

Once successfully logged on, the device Status will be *Connected*.

1. In the CloudTest, Central tab, select Clips, and then click New on the toolbar.



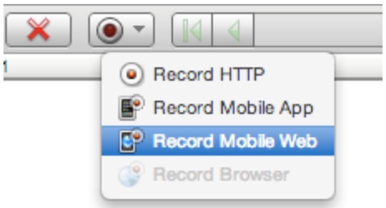
A new *Untitled Test Clip* opens in a Clip Editor tab. A Record pop-up identifies the Record drop-down.



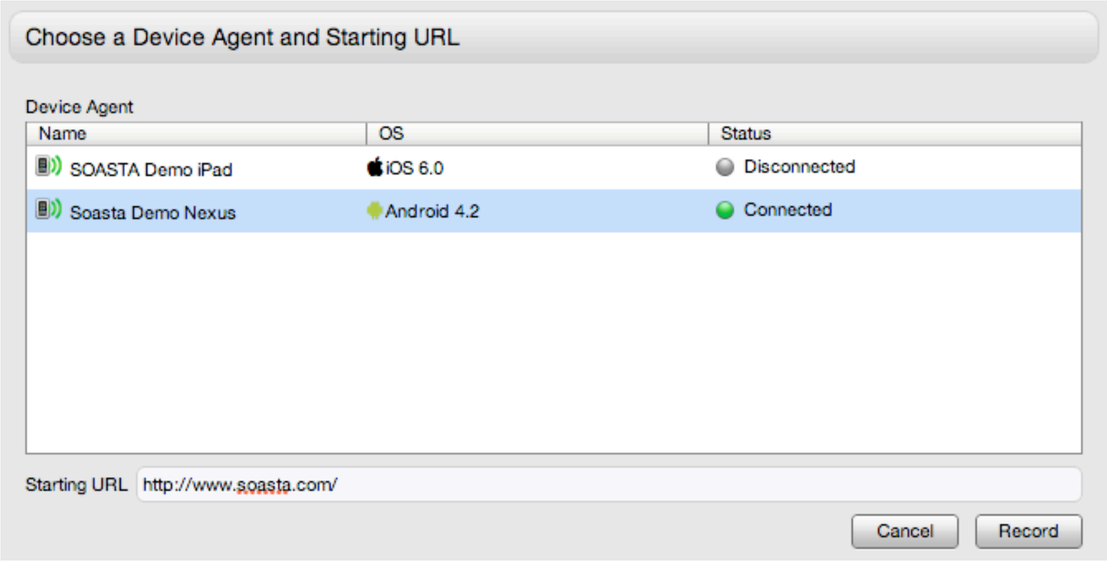
Record

Click the Record drop-down to begin a mobile app recording.

2. Click the Record button and then select Record Mobile Web.



The Choose a Device Agent and Starting URL box appears.



Choose a Device Agent and Starting URL

Device Agent

Name	OS	Status
SOASTA Demo iPad	iOS 6.0	● Disconnected
Soasta Demo Nexus	Android 4.2	● Connected

Starting URL: http://www.soasta.com/

TIP: If your Device Agent doesn't appear, open it in the Central > Device Agents list and ensure that Mobile Web Capable is set. If it is not set, make the change and save the device agent. If the device agent still doesn't appear it the list, exit the Choose a Device Agent box and try again.

3. Select the TouchTest Agent that you created above. For example, *SOASTA Demo Nexus*.

   Specify the Starting URL of the mobile web site to record.

   For example, enter the URL for the SOASTA web site:

   http://www.soasta.com/

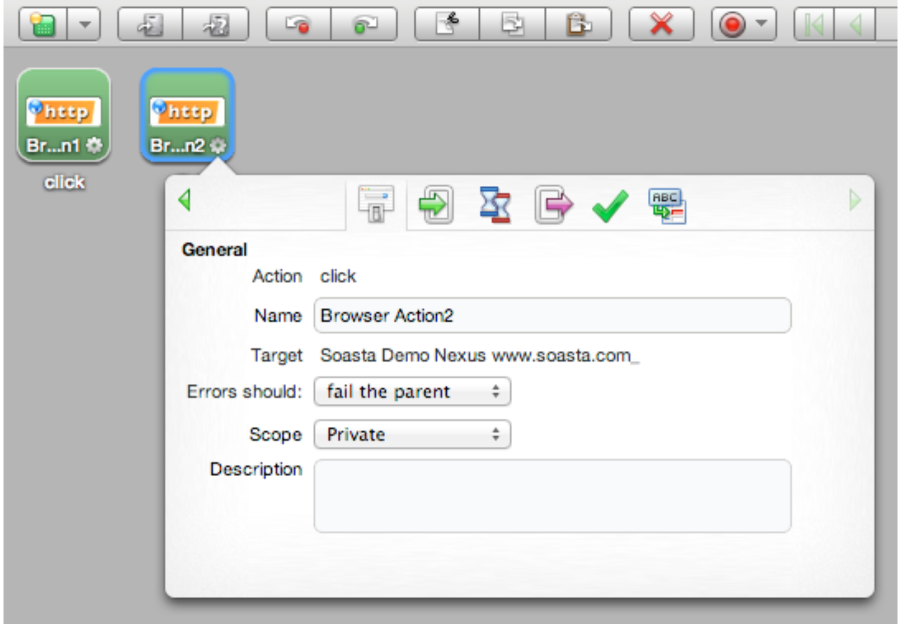
4. Once the Device Agent selected is "Connected" and the Starting URL has been entered, the Record button in the Choose box becomes active. Click it to continue.

   TouchTest Agent launches TouchTest Web, after which the target site appears in a browser window.

5. Perform the planned mobile app user interactions on your mobile device. For each action you perform in the browser, TouchTest Web adds an action to the clip.

   For example, in our demo clip, which is used in the remainder of this tutorial, we tapped the following sequence:

   a. On SOASTA home, tap menu, and then the first link, Solutions.
   b. Tap Mobile performance testing.
   c. Tap the SOASTA logo (top left).
   d. Tap Menu, Customers section, and then Case Studies.

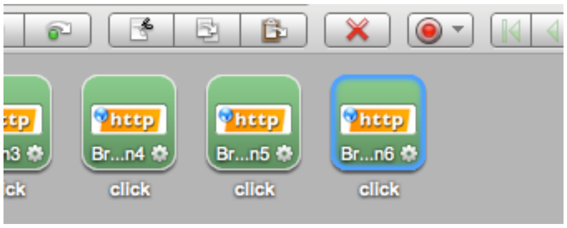


General
Action click
Name Browser Action2
Target Soasta Demo Nexus www.soasta.com_
Errors should: fail the parent
Scope Private
Description

The Info Window switches along the stream of actions as they are added. This window provides a convenient entry point to action details and can be used to change properties and add accessors such as validations. Actions can also be double-clicked to open them in the Clip Editor lower panel.

TIP: You can hide the Info Window using the Etc. (…) menu Show/Hide Info While Recording command. Click the Gear icon to Show Info for editing, or alternately, double-click to use the Action Editor in the lower panel.

6. Once the relevant interactions have been recorded, click the Record button again to stop the recording.



Create a Composition

With your test clip open in the Clip Editor, you are ready to create and play this simple test clip in a new test composition.

1. To create a new composition from your test clip, click the Use in Test Composition drop-down in the upper-right corner of the Clip Editor toolbar.

   Before making a selection note the menu commands:

   - **Open in Test Composition**

     Choose Open in Test Composition to add this clip to a new draft composition where additional composition parameters can be set in the Composition Editor, Edit tab before proceeding to play.

     This option is useful when composition-level properties need to set, and other composition-building tasks done before the test is ready to play.

   - **Play in Test Composition**

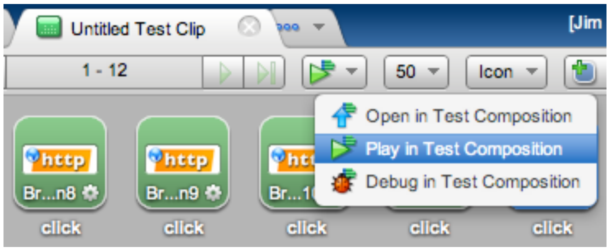
     Choose Play in Test Composition to add this clip to a new draft composition where it will immediately be played in the Composition, Play tab before proceeding to edit parameters or play.

     This option is useful for initial runs of a test prior to adding any necessary composition-level properties.

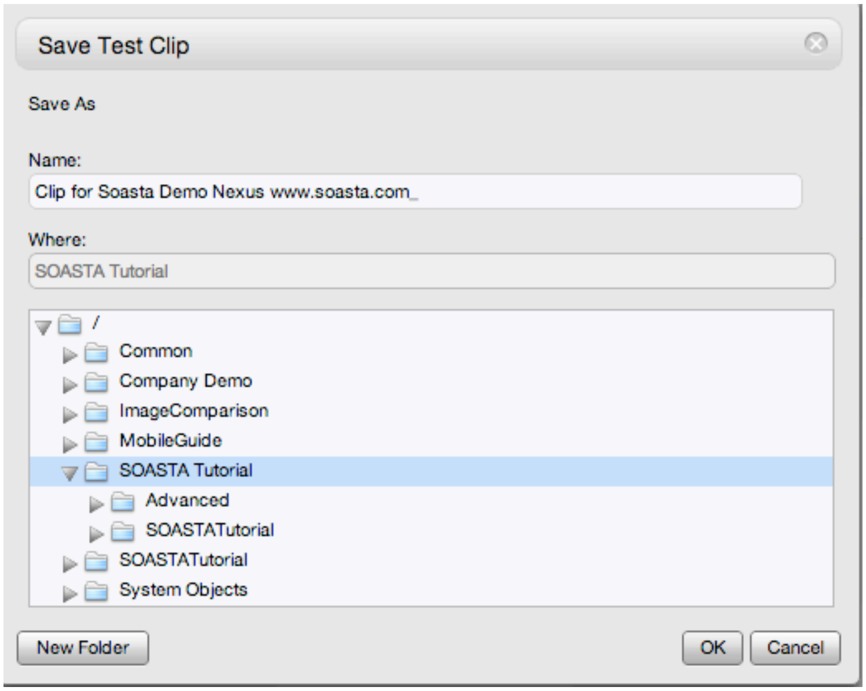
   - **Debug in Test Composition**

     Choose Debug in Test Composition to add this clip to a new draft composition where it can be debugged in the Composition, Debugging tab before proceeding to edit parameters or play based on debug actions.

2. After observing the menu commands, choose Play in Test Composition.



3. The Save Test Clip box appears. Accept the default test clip name and click OK.



Save Test Clip

Save As

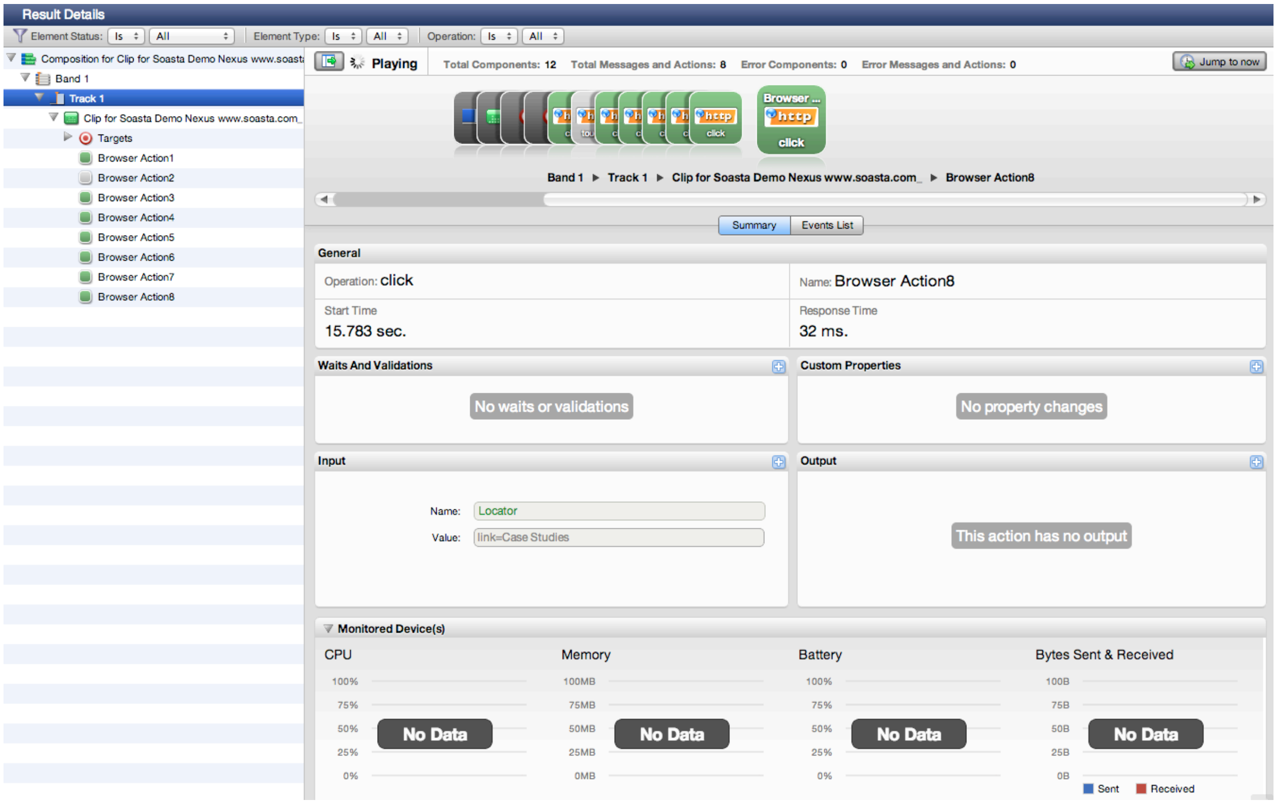
Name: Clip for Soasta Demo Nexus www.soasta.com_

Where: SOASTA Tutorial

4. A Composition Editor tab appears. As play starts, the Composition Editor switches immediately to the Play tab.

Result Details

The Result Details dashboard helps to discover the cause of errors in your test, if any. While play continues results are posted in the Composition Editor, Play tab, Result Details widget.



Result Details

Element Status: Is All Element Type: Is All Operation: Is All

Playing Total Components: 12 Total Messages and Actions: 8 Error Components: 0 Error Messages and Actions: 0

Band 1 ▶ Track 1 ▶ Clip for Soasta Demo Nexus www.soasta.com_ ▶ Browser Action8

General

Operation: click

Name: Browser Action8

Start Time 15.783 sec.

Response Time 32 ms.

Waits And Validations: No waits or validations

Custom Properties: No property changes

Input

Name: Locator

Value: link=Case Studies

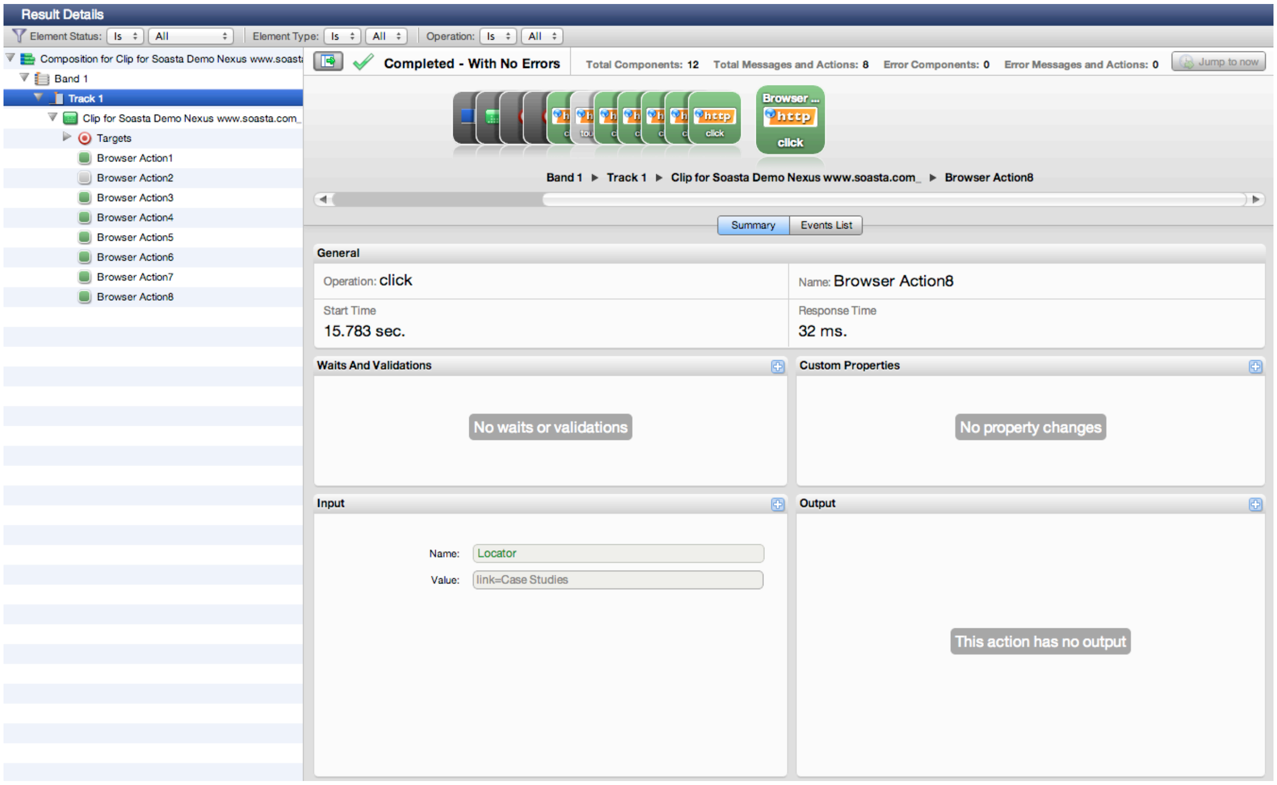
Output: This action has no output

Monitored Device(s)

The mobile Browser Actions performed when the clip was created are played back on the device. Click to expand the nodes in the Navigation Tree on the left as they appear.

- Result Details uses a Cover Flow (top panel center) to display the test composition's stream as it occurs.
- This stream is also shown in the Navigation Tree (on the left) as elements are executed. As play continues, the focus is set to the last executed element unless user interaction prevents it. The current container is expanded while the prior containers are closed.
- Clicking an element during play will halt this auto-focus-to-the-last-executed behavior. To resume auto-focus once interrupted, click Jump to Now in the upper right of the dashboard.

Once play completes, the final results are graded according to the default criteria. For example, in the shot below the test has the status “Completed – With No Errors.”



Result Details

Element Status: Is All | Element Type: Is All | Operation: Is All

- Composition for Clip for Soasta Demo Nexus www.soasta
  - Band 1
    - Track 1
      - Clip for Soasta Demo Nexus www.soasta.com_
        - Targets
        - Browser Action1
        - Browser Action2
        - Browser Action3
        - Browser Action4
        - Browser Action5
        - Browser Action6
        - Browser Action7
        - Browser Action8

Completed - With No Errors Total Components: 12 Total Messages and Actions: 8 Error Components: 0 Error Messages and Actions: 0 Jump to now

Band 1 ▶ Track 1 ▶ Clip for Soasta Demo Nexus www.soasta.com_ ▶ Browser Action8

Summary | Events List

General

Operation: click | Name: Browser Action8

Start Time 15.783 sec. | Response Time 32 ms.

Waits And Validations: No waits or validations

Custom Properties: No property changes

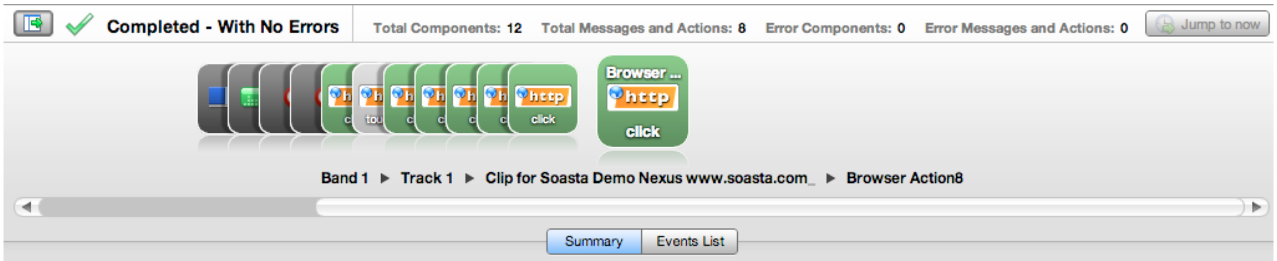
Input

Name: Locator

Value: link=Case Studies

Output: This action has no output

After the test completes, use the Cover Flow to examine test component details.

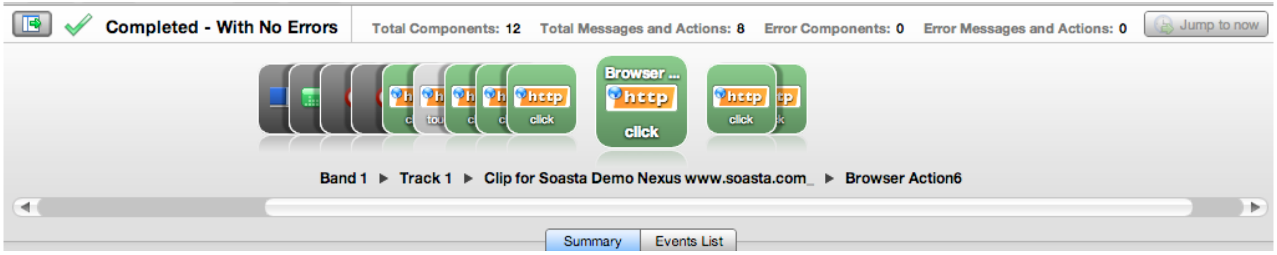


Completed - With No Errors Total Components: 12 Total Messages and Actions: 8 Error Components: 0 Error Messages and Actions: 0 Jump to now

Band 1 ▶ Track 1 ▶ Clip for Soasta Demo Nexus www.soasta.com_ ▶ Browser Action8

Summary | Events List

- Click any object in the Cover Flow at the top to center it and display its details and play statistics in the panes below.



Completed - With No Errors Total Components: 12 Total Messages and Actions: 8 Error Components: 0 Error Messages and Actions: 0 Jump to now

Band 1 ▶ Track 1 ▶ Clip for Soasta Demo Nexus www.soasta.com_ ▶ Browser Action6

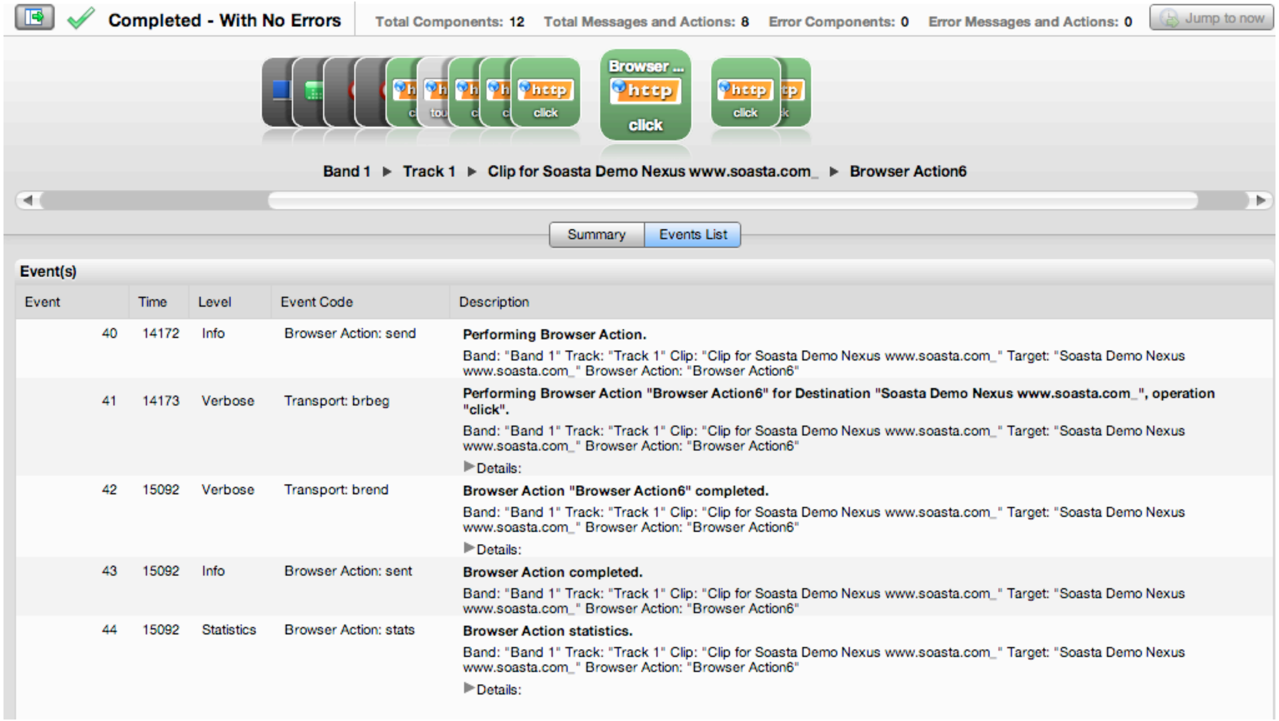
Summary | Events List

- Use the scrollbar to browse the flow. Select any item to show its low-level details.

Result Details has several methods for navigating through the test results. Click the checkboxes in the cover flow to quickly display messages (or Actions) only (within a single band, track, test clip, or chain).

> **TIP:** Before closing the composition, click the Composition Editor, Edit tab (top right of the Composition Editor) and then click Save on the Composition Editor toolbar. Give the composition a name. For example: “`Composition for <Clip Name>`.”

For selected items in either the Cover Flow or the Navigation Tree, additional details appear in the Summary detail sections (lower right) as well as in the Events List tab.



Completed - With No Errors
Total Components: 12
Total Messages and Actions: 8
Error Components: 0
Error Messages and Actions: 0
Jump to now
Browser ...
http
click
Band 1 ▶ Track 1 ▶ Clip for Soasta Demo Nexus www.soasta.com_ ▶ Browser Action6
Summary
Events List

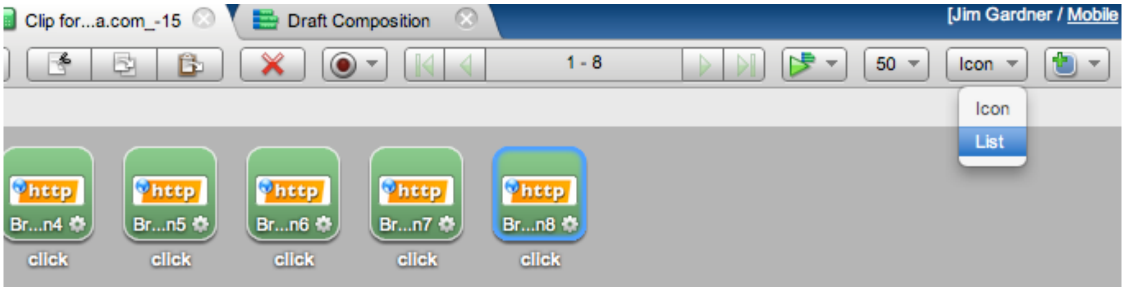
Event(s)

Event	Time	Level	Event Code	Description
40	14172	Info	Browser Action: send	**Performing Browser Action.** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action6"
41	14173	Verbose	Transport: brbeg	**Performing Browser Action "Browser Action6" for Destination "Soasta Demo Nexus www.soasta.com_", operation "click".** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action6" ▶Details:
42	15092	Verbose	Transport: brend	**Browser Action "Browser Action6" completed.** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action6" ▶Details:
43	15092	Info	Browser Action: sent	**Browser Action completed.** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action6"
44	15092	Statistics	Browser Action: stats	**Browser Action statistics.** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action6" ▶Details:

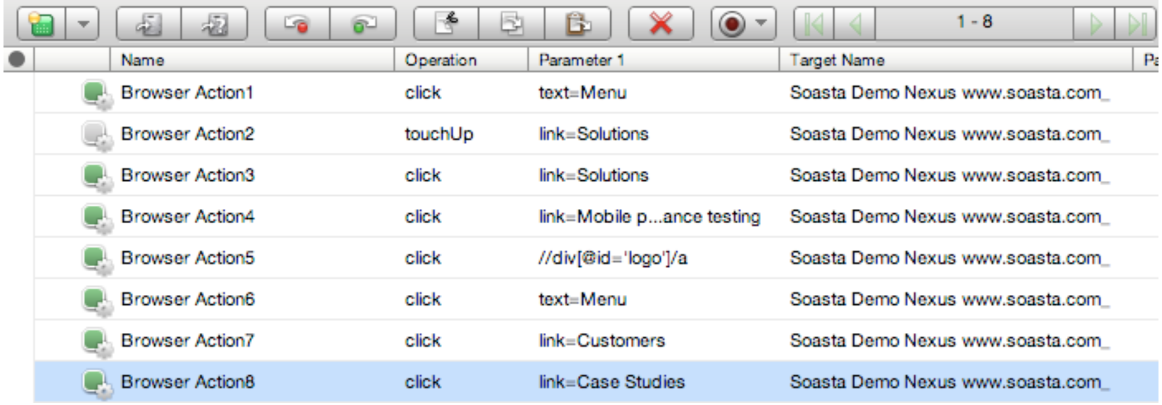
Advanced Clip Editing

Now that we've played this simple test composition successfully, and learned how CloudTest will check the success or failure of a given composition, let's return to the test clip to inspect the clip elements and do some additional parameterization.

- Click the Clip Editor tab if it's still open, or double click the test clip in the Navigation Tree to open it in a new Clip Editor tab.
- Once the Clip Editor tab is in view, click the drop down Icon/List button on the toolbar and then select List.



The List view is useful while clip editing, because it shows all the parameters and their corresponding inputs in one tabular view.



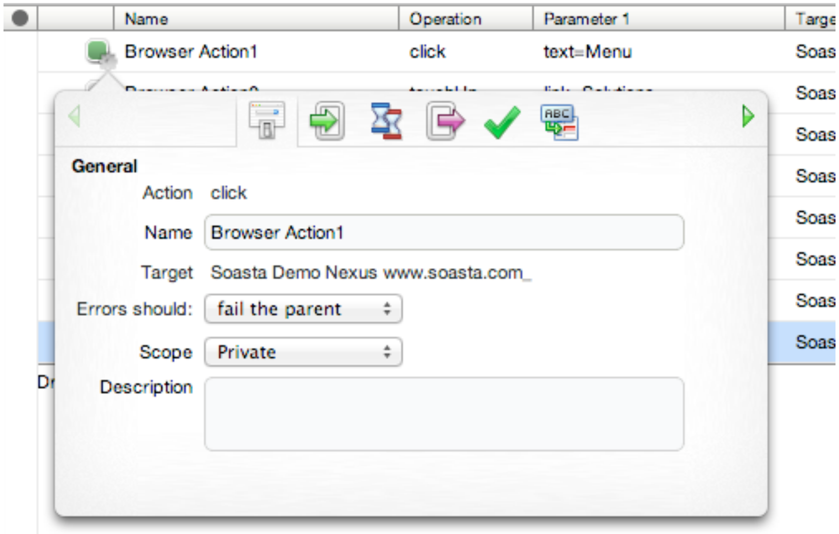
Name	Operation	Parameter 1	Target Name	Pa
Browser Action1	click	text=Menu	Soasta Demo Nexus www.soasta.com_	
Browser Action2	touchUp	link=Solutions	Soasta Demo Nexus www.soasta.com_	
Browser Action3	click	link=Solutions	Soasta Demo Nexus www.soasta.com_	
Browser Action4	click	link=Mobile p...ance testing	Soasta Demo Nexus www.soasta.com_	
Browser Action5	click	//div[@id='logo']/a	Soasta Demo Nexus www.soasta.com_	
Browser Action6	click	text=Menu	Soasta Demo Nexus www.soasta.com_	
Browser Action7	click	link=Customers	Soasta Demo Nexus www.soasta.com_	
Browser Action8	click	link=Case Studies	Soasta Demo Nexus www.soasta.com_	

When additional parameters are present they are displayed to the right of the Parameter 1 column.

Inspecting Browser Action Details

Browser Action details can be examined in the Clip Editor workspace by clicking the Gear icon or by opening an item to view it in the lower panel Action Editor. Readers should familiarize themselves with both techniques, keeping in mind that some action details such as those for repasts and custom properties can only be set in the lower panel.

When you click the Gear icon, the Info Window, General tab appears.



Name	Operation	Parameter 1	Targe
Browser Action1	click	text=Menu	Soas

General

Action click

Name Browser Action1

Target Soasta Demo Nexus www.soasta.com_

Errors should: fail the parent

Scope Private

Description

Click the tabs to browse the details for the given action.

In addition to the General tab, the Info Window tabs are Inputs, Waits, Outputs, Validations, and Property Sets.

- **Inputs** (Locator, Scale, Precision, Content Offset)

  Locators are unique characteristics that identify a specific element or object on a mobile device. Locators come in many forms, including links, IDs such as those defined within CSS, and XPath expressions.

- **Waits** (Pre-Action Waits , Post-Action Waits)

  Waits are commands that tell CloudTest not to execute an Action until a condition is met (pre-action waits), or to not continue processing the outputs, validations and property sets of the Action until a condition is met (post-action waits).

- **Outputs**

  Outputs specify what is to be shown in the Result Viewer for a given Action. Typical outputs include “captureScreenshot”, “outputElementText”, and “outputInnerHTML”. A single Action can have an unlimited number of outputs.

- **Validations**

  Validations verify some content or event occurred as expected and has a corresponding Failure Action. Browser Action validations can range from simple true/false conditions to more complex conditions. A single Browser Action can have an unlimited number of validations. Any validation failures will be exposed in the Results Dashboard.

- **Property Sets**

  Property Sets give you the ability to take text or data from the app you are testing and store it in a custom property for use in a subsequent action or message.

SOASTA CloudTest includes three property sets, all of which have relevance for refining and editing a selected Browser Action.

○ *Custom Properties*

Custom Properties are user-defined properties that are available to all clip elements, including Actions. Custom properties can be thought of as backdoors that allow access to portions of the object model more easily.

○ *System Properties*

System Properties are available to all clip elements, including Actions. SOASTA CloudTest defines system properties. For example, a test clip has system properties such as name, repeat timing, label, and more.

○*Global Properties*

Global properties are defined within the Central > Global Properties List and are “global” within the entire SOASTA CloudTest environment—and can be used across compositions.

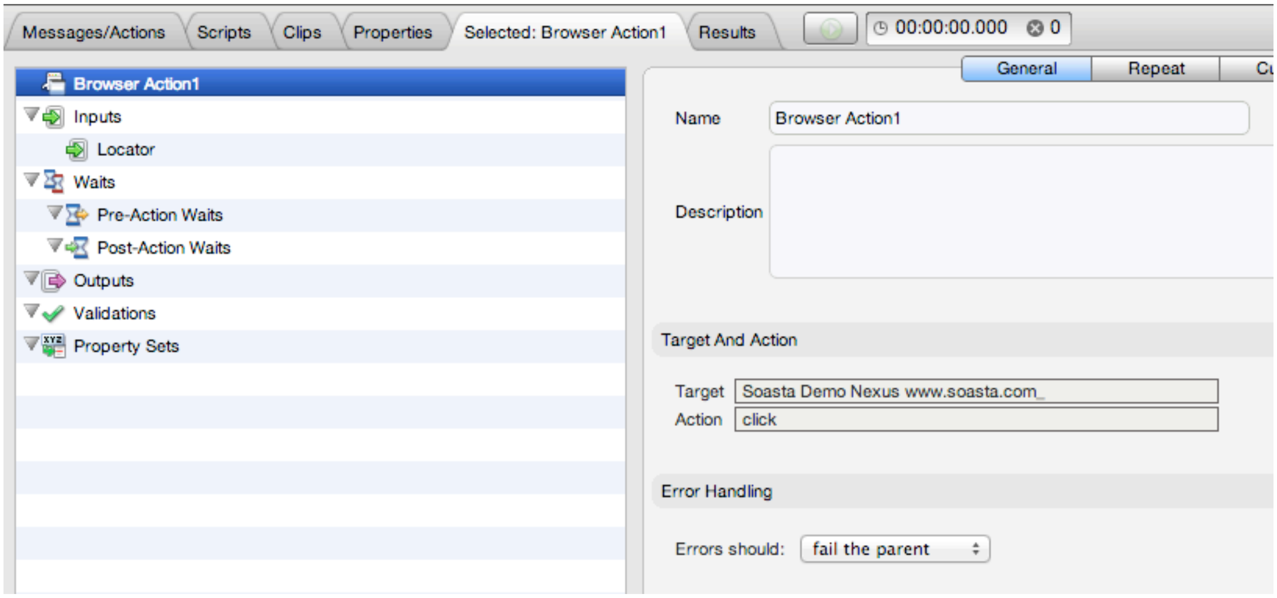
The Info Window disappears when the lower panel is open for a selection—and vice versa. The next section covers action attributes that appear only in the lower panel.

Browser Action Properties

Additional parameters, such as Custom Properties, can be set by double-clicking a browser action to open it in the Clip Editor lower panel. Action level properties are shown in the tree on the left.

1. Select the top-level node in the tree (as shown below)

- General, Repeat, and Custom Properties (for the action only; not for the entire clip) tabs appear on the right. Note that Error Handling here is set to *Errors should fail the parent* by default.



Name: Browser Action1

Target: Soasta Demo Nexus www.soasta.com_

Action: click

Errors should: fail the parent

Note in the shot above that when the Browser Action1 (top level node) is selected that the action's Error Handling settings are set to fail the parent by default. When you add accessors such as validations they are children of the action, and can have their own failure actions that differ from those of the parent action.

- In addition to the General tab clicking the appropriate tab can set attributes shown above, the action has Repeat and Custom Properties attributes that.

- Other settings, including Waits, Inputs, Outputs, Validations, and Property Sets that were seen in the Info Window steps above can also be set in the lower panel by clicking that node in the tree and then performing the desired action on the right.

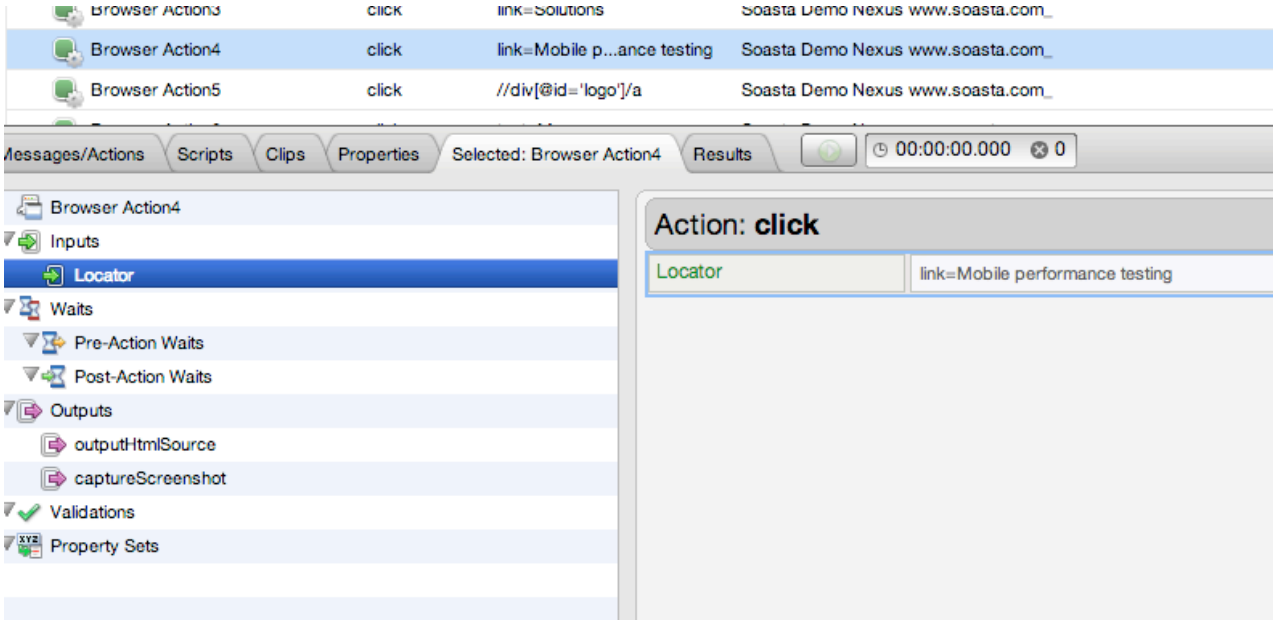
1. In the *Selected: Browser Action1* tab (or for any selected Browser Action), familiarize yourself with the available elements and properties.

Adding Outputs

In the following steps, we will add two outputs to the first Browser Action captured from the www.soasta.com site in the recorded clip above—one to output the HTML source in the test clip and another to capture a screenshot during the test.

TIP: If you are the author or programmer for a given site, you are likely already familiar with the source code for your site. However, if you are a tester who is not as familiar with the programming details of a site, getting an output of the source code provides insight into what to test. Getting the HTML source for a page is one of the best ways to decide what to validate in that page—whether they are UI elements or page content. The following steps can be performed in either the Info Window or, as shown below, in the lower panel.

1. Open Browser Action4 open in the lower panel Action Editor. Since your clip's order of actions may vary from our example, note that we are looking for the action corresponding to " link=Mobile performance testing."

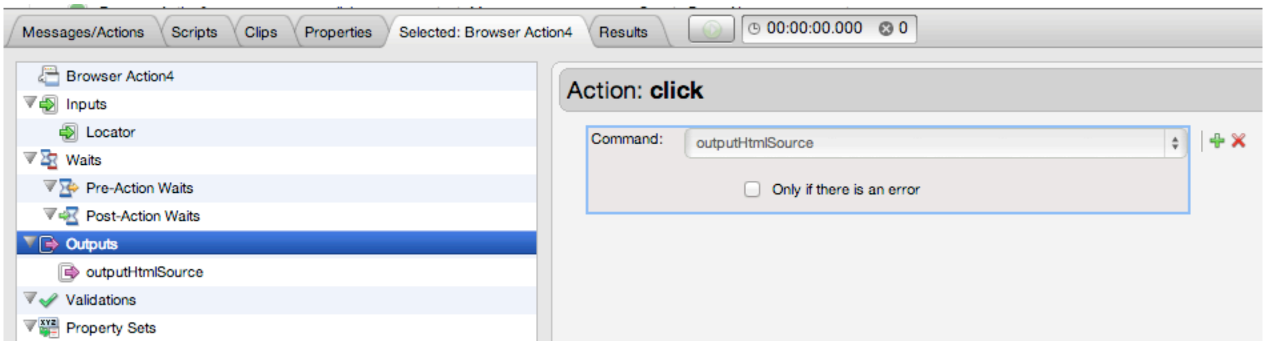


Browser Action3	click	link=Solutions	Soasta Demo Nexus www.soasta.com_
Browser Action4	click	link=Mobile p...ance testing	Soasta Demo Nexus www.soasta.com_
Browser Action5	click	//div[@id='logo']/a	Soasta Demo Nexus www.soasta.com_

Action: **click**

Locator: link=Mobile performance testing

2. Click Outputs in the lower panel list (on the left), and then in the *Action: type* field just to the right, click the green Plus sign. A new output form is added with the default output for a browser action, *outputHTMLSource*, shown.

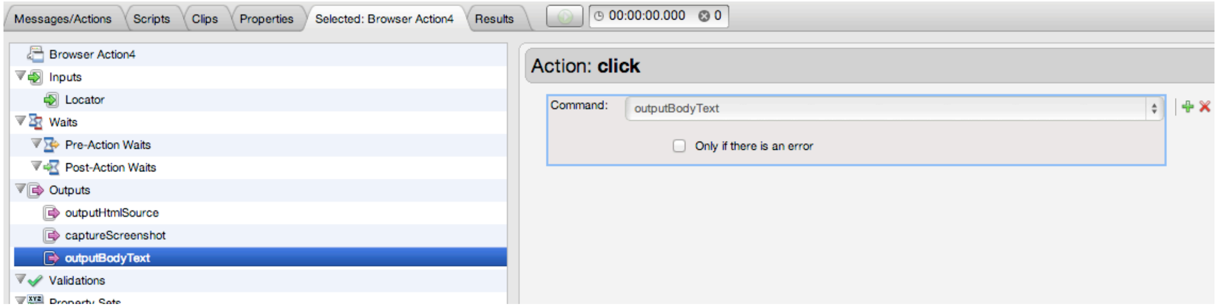


Action: **click**

Command: outputHtmlSource

□ Only if there is an error

2. Next, add a second output for outputBodyText. Unlike the HTML source output, this output gets only the <BODY> text, which in most cases accounts for all of the visible content of the page—that is, it's meaningful text, image, or other content.



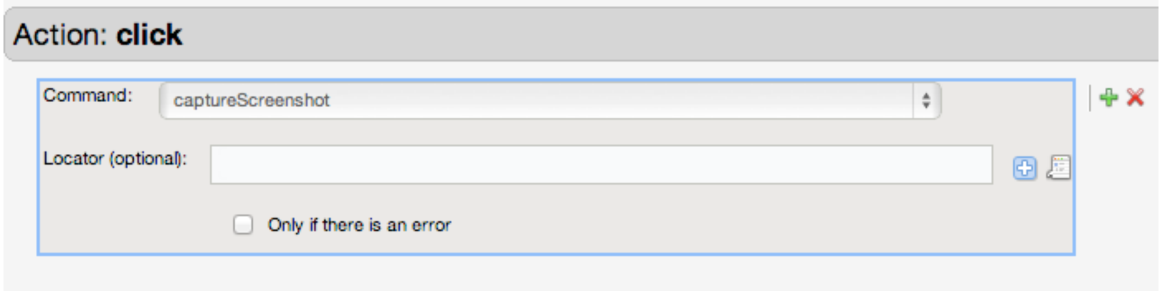
Action: **click**

Command: outputBodyText

□ Only if there is an error

3. Leave *Only if there is an error* unchecked for now, but note this checkbox for later use in cases where the output is desired only in the event this Browser Action produces an error.

3. Finally, select Browser Action1 and add an output, *captureScreenshot,* (it is the top entry in the drop down list). Leave the Locator field blank.



Action: **click**

Command: captureScreenshot

Locator (optional):

□ Only if there is an error

4. Click Save on the Clip Editor toolbar.

5. Return to the Composition Editor tab once again and click Play a second time.

TIP: If you closed the draft composition without saving it, use Play in Test Composition to create a new draft composition, or if you saved and closed the test composition, click the Central tab, Compositions list and re-open it from there.

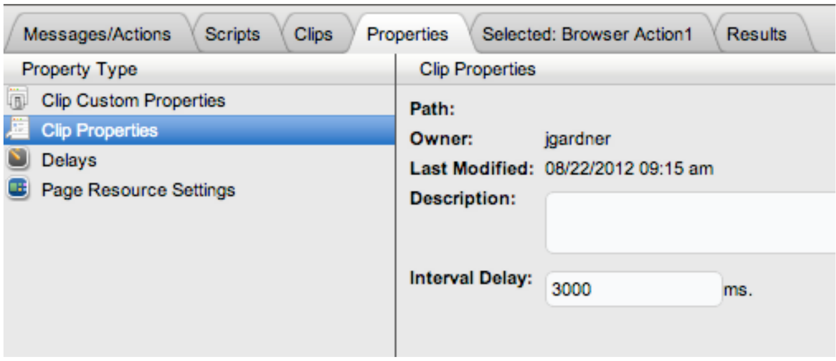
Adding an Interval Delay between Each Action

Out test clip can succeed as is when played in a test composition. In some cases, however, we may want to improve the playback on the device to better show the pace of the test as it plays. In the following steps, we will add an interval delay to the test clip.

This type of delay will stretch out the time between all the recorded app actions. In some cases, an interval delay can decrease simple errors that stem from network and device timing.

Imposing delays, either using the Interval Delay setting or by inserting Delay clip elements, can make the test more user friendly.

1. Click the Properties tab in the minimized sub-panel and then select the Clip tab at the top of the pane (the Clip tab may already be visible if properties are already open from the prior exercise).
2. In the Property Type list, click Clip Properties.
3. In the Clip Properties panel on the right, enter an Interval Delay in the given field. For example, *2000 ms* (or milliseconds). Entering *3000* adds a three second gap between each app action in the given test clip. This is in addition to any waits that we may also add between specific actions (or which TouchTest technology may build into some clips).



Path:
Owner: jgardner
Last Modified: 08/22/2012 09:15 am
Description:
Interval Delay: 3000 ms.

3

4. Click Save on the Clip Editor toolbar.

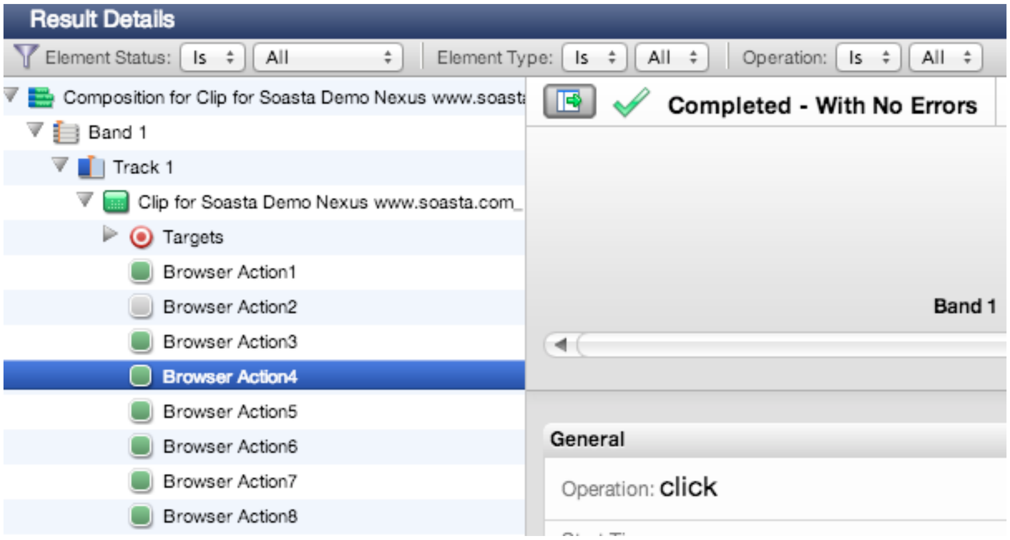
In the following section, we'll inspect results for the outputs that were set on Browser Action4 above.

Examining Outputs in Results

Now that we've added three outputs to the test clip, let's play the test composition again. Switch back to the Draft Composition tab if it's still open or use the Play in Test Composition command to start playing the saved clip in a new draft composition.

Once play completes, we will locate and examine the outputs we set above for Browser Action4.

1. In the Result Details dashboard, expand the Navigation Tree until Browser Action4 is in display.



Result Details

Element Status: Is All Element Type: Is All Operation: Is All

- Composition for Clip for Soasta Demo Nexus www.soast
  - Band 1
    - Track 1
      - Clip for Soasta Demo Nexus www.soasta.com_
        - Targets
        - Browser Action1
        - Browser Action2
        - Browser Action3
        - Browser Action4
        - Browser Action5
        - Browser Action6
        - Browser Action7
        - Browser Action8

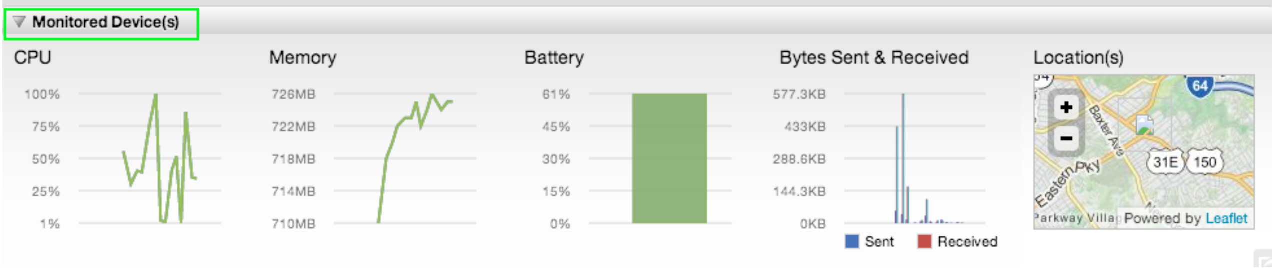
✓ Completed - With No Errors

Band 1

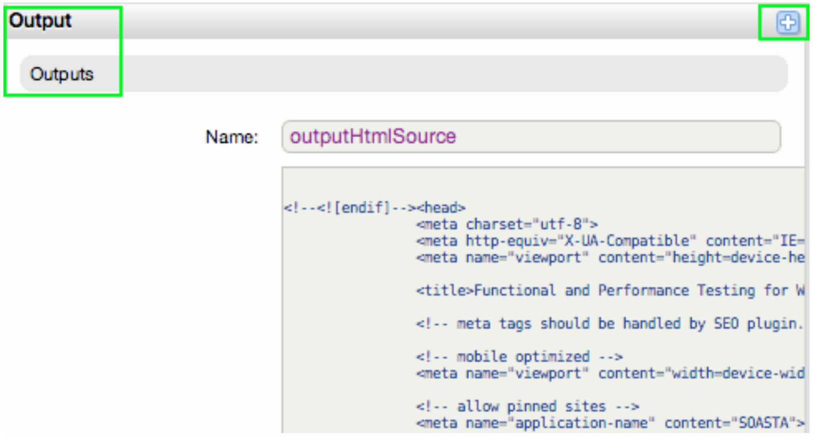
General

Operation: click

2. Collapse the Monitored Device(s) tab by clicking its down arrow.



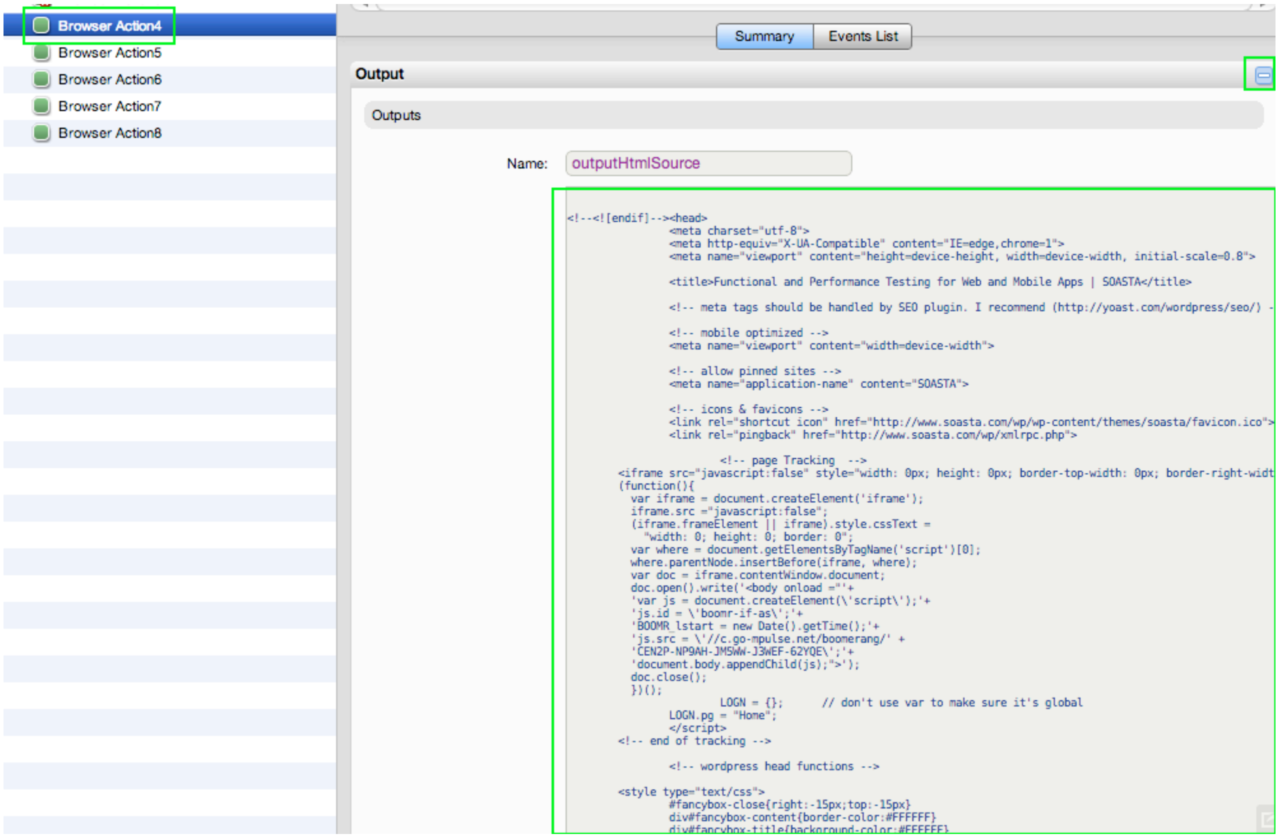
3. Click the Output section's Plus (+/-) icon to maximize it.



Name: outputHtmlSource

```
<!--<![endif]--><head>
                <meta charset="utf-8">
                <meta http-equiv="X-UA-Compatible" content="IE=
                <meta name="viewport" content="height=device-he

                <title>Functional and Performance Testing for W

                <!-- meta tags should be handled by SEO plugin.

                <!-- mobile optimized -->
                <meta name="viewport" content="width=device-wid

                <!-- allow pinned sites -->
                <meta name="application-name" content="SOASTA">
```

The Output section expands to use the whole workspace.

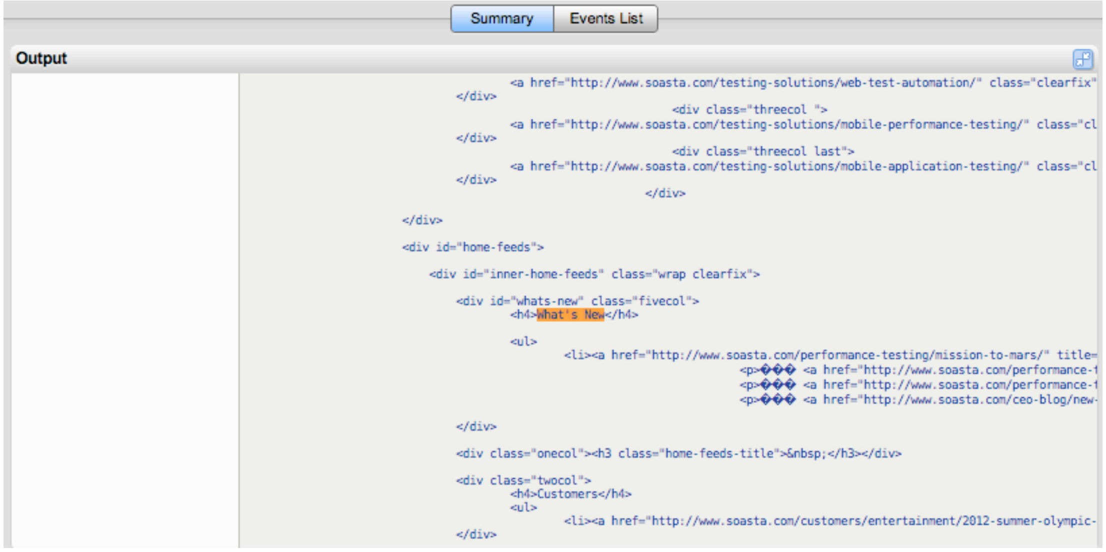


Name: outputHtmlSource

```
<!--<![endif]--><head>
                <meta charset="utf-8">
                <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
                <meta name="viewport" content="height=device-height, width=device-width, initial-scale=0.8">

                <title>Functional and Performance Testing for Web and Mobile Apps | SOASTA</title>

                <!-- meta tags should be handled by SEO plugin. I recommend (http://yoast.com/wordpress/seo/) -

                <!-- mobile optimized -->
                <meta name="viewport" content="width=device-width">

                <!-- allow pinned sites -->
                <meta name="application-name" content="SOASTA">

                <!-- icons & favicons -->
                <link rel="shortcut icon" href="http://www.soasta.com/wp/wp-content/themes/soasta/favicon.ico">
                <link rel="pingback" href="http://www.soasta.com/wp/xmlrpc.php">

                        <!-- page Tracking  -->
        <iframe src="javascript:false" style="width: 0px; height: 0px; border-top-width: 0px; border-right-widt
        (function(){
          var iframe = document.createElement('iframe');
          iframe.src ="javascript:false";
          (iframe.frameElement || iframe).style.cssText =
            "width: 0; height: 0; border: 0";
          var where = document.getElementsByTagName('script')[0];
          where.parentNode.insertBefore(iframe, where);
          var doc = iframe.contentWindow.document;
          doc.open().write('<body onload ="'+
          'var js = document.createElement(\'script\');'+
          'js.id = \'boomr-if-as\';'+
          'BOOMR_lstart = new Date().getTime();'+
          'js.src = \'//c.go-mpulse.net/boomerang/' +
          'CEN2P-NP9AH-JM5WW-J3WEF-62YQE\';'+
          'document.body.appendChild(js);">');
          doc.close();
          })();
                        LOGN = {};      // don't use var to make sure it's global
                LOGN.pg = "Home";
                </script>
        <!-- end of tracking -->

                <!-- wordpress head functions -->

        <style type="text/css">
                #fancybox-close{right:-15px;top:-15px}
                div#fancybox-content{border-color:#FFFFFF}
                div#fancybox-title{background-color:#FFFFFF}
```

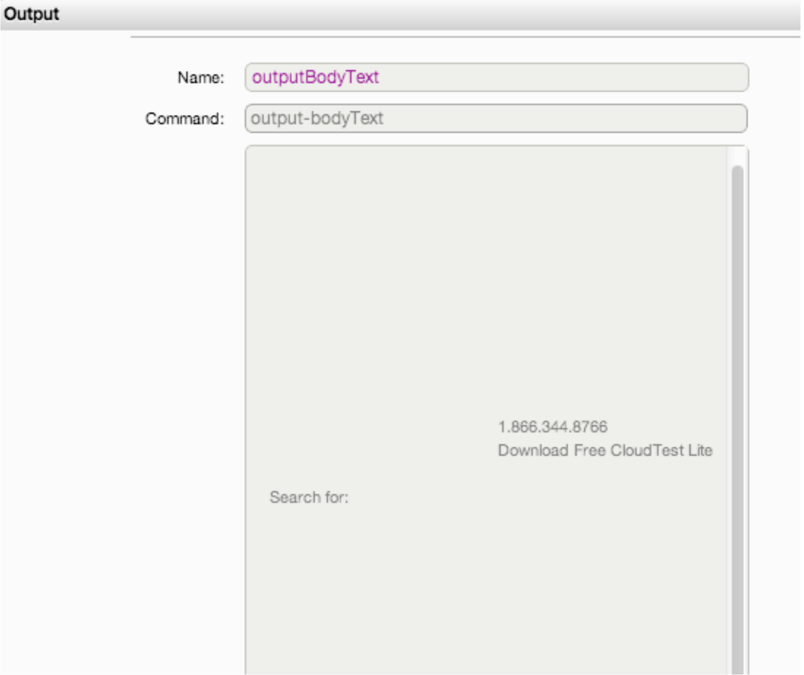
4. In the maximized Output section, outputHTMLSource, locate the phrase, *What’s New*.

   TIP: Use your browser’s search in page feature to locate specific text in HTML source.



```
                <a href="http://www.soasta.com/testing-solutions/web-test-automation/" class="clearfix"
            </div>
                                        <div class="threecol ">
                <a href="http://www.soasta.com/testing-solutions/mobile-performance-testing/" class="cl
            </div>
                                        <div class="threecol last">
                <a href="http://www.soasta.com/testing-solutions/mobile-application-testing/" class="cl
            </div>
                                    </div>

        </div>

        <div id="home-feeds">

            <div id="inner-home-feeds" class="wrap clearfix">

                <div id="whats-new" class="fivecol">
                    <h4>What's New</h4>

                    <ul>
                        <li><a href="http://www.soasta.com/performance-testing/mission-to-mars/" title=
                                                        <p>��� <a href="http://www.soasta.com/performance-t
                                                        <p>��� <a href="http://www.soasta.com/performance-t
                                                        <p>��� <a href="http://www.soasta.com/ceo-blog/new-

                </div>

                <div class="onecol"><h3 class="home-feeds-title">&nbsp;</h3></div>

                <div class="twocol">
                    <h4>Customers</h4>
                    <ul>
                        <li><a href="http://www.soasta.com/customers/entertainment/2012-summer-olympic-
                </div>
```

5. Next, scroll down and locate the body text output.



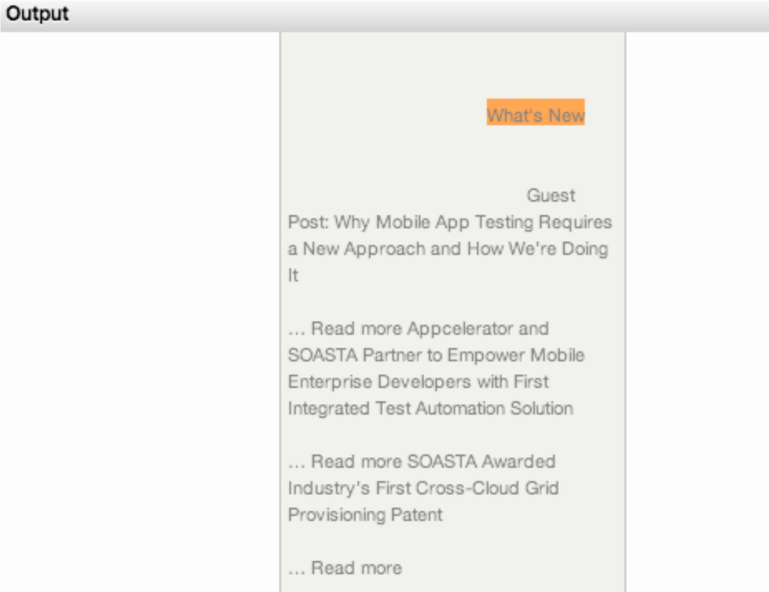
Name: outputBodyText

Command: output-bodyText

1.866.344.8766
Download Free CloudTest Lite

Search for:

In the Output section, outputBodyText, locate the phrase, *What's New*.



What's New

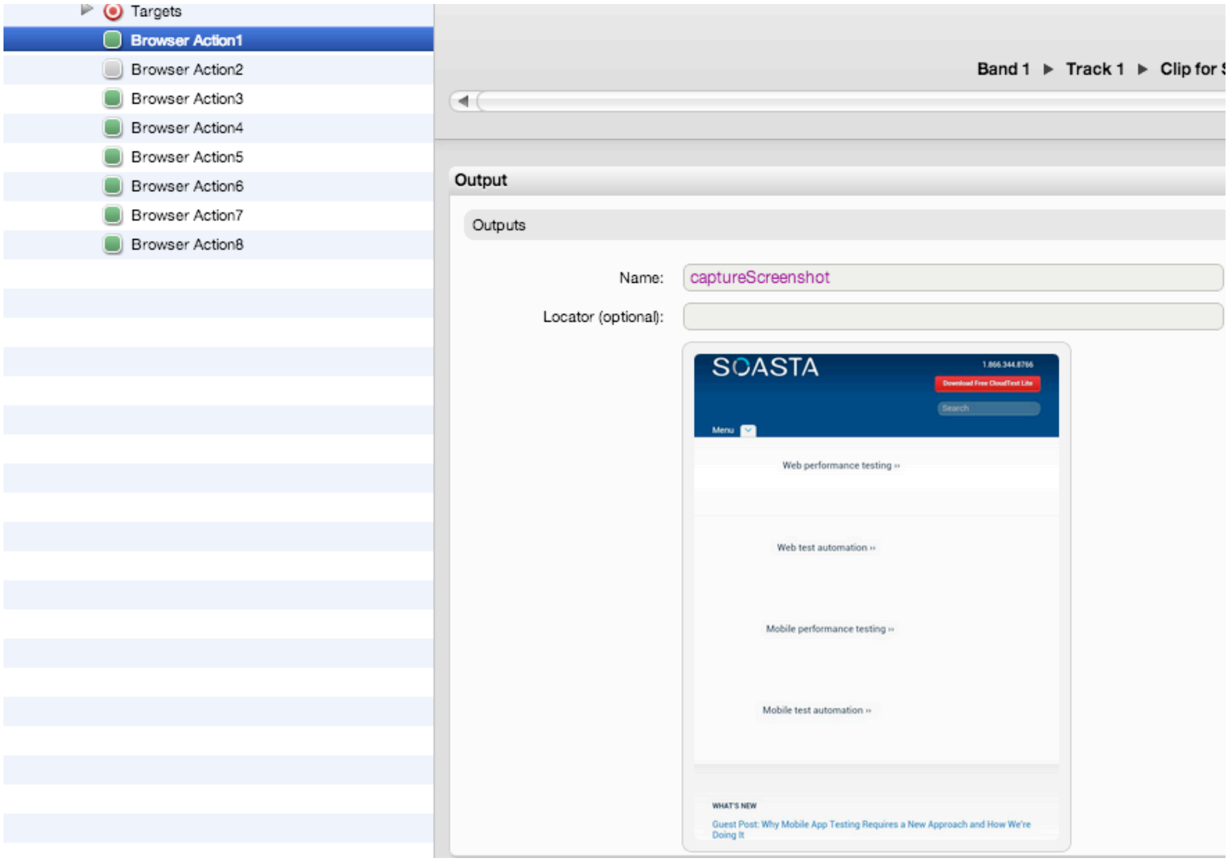
Guest Post: Why Mobile App Testing Requires a New Approach and How We're Doing It

... Read more Appcelerator and SOASTA Partner to Empower Mobile Enterprise Developers with First Integrated Test Automation Solution

... Read more SOASTA Awarded Industry's First Cross-Cloud Grid Provisioning Patent

... Read more

6. Now, select Browser Action1 in the tree and locate the captured screenshot in the Output section.



Name: captureScreenshot

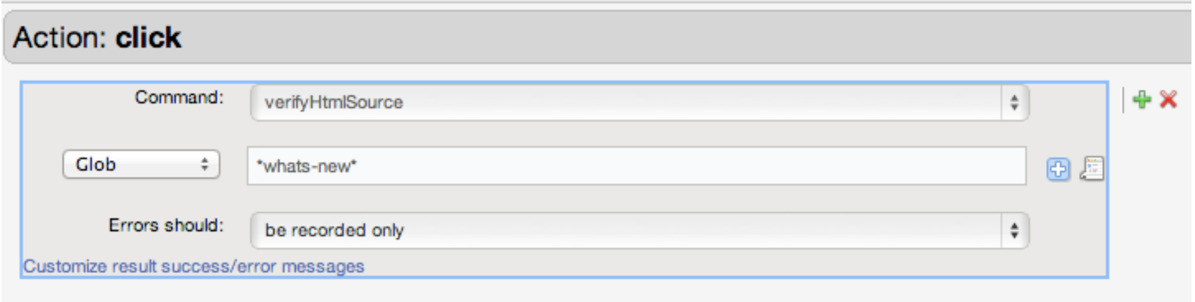
Locator (optional):

TIP: You can click the screenshot to pop it out to full view.

Adding a Text Validation on HTML Source

In this and the next section, we'll validate the What's New text where it occurs in HTML Source and in Body text.

1. In the Clip Editor, double-click Browser Action4 to open its properties panel.
2. Select Validations in the list, and click the green Plus icon to add a new validation form to the workspace.
3. Select verifyHTMLSource in the Command drop-down.



Action: **click**

Command: verifyHtmlSource

Glob: *whats-new*

Errors should: be recorded only

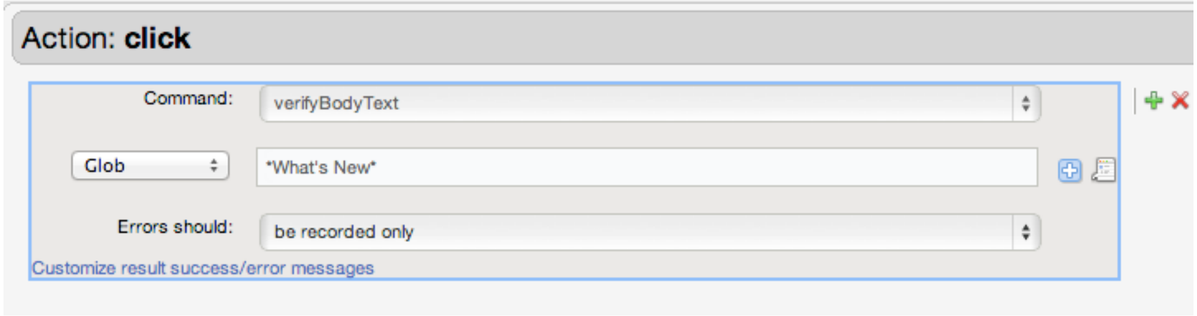
Customize result success/error messages

4. Click the Match field drop-down, and select *Glob* and enter the match text. Enter *whats-new* to find the class id text that we saw above after searching "What's New" in the outputHTMLSource item in results.
5. Click Save on the Clip Editor toolbar.

Adding a Validation on Body Text

Next, we will add a text validation on Browser Action4.

1. With Validations selected, click the green Plus icon to the right of the previous validation.
2. Select verifyBodyText in the Command drop-down.



Action: **click**

Command: verifyBodyText

Glob: *What's New*

Errors should: be recorded only

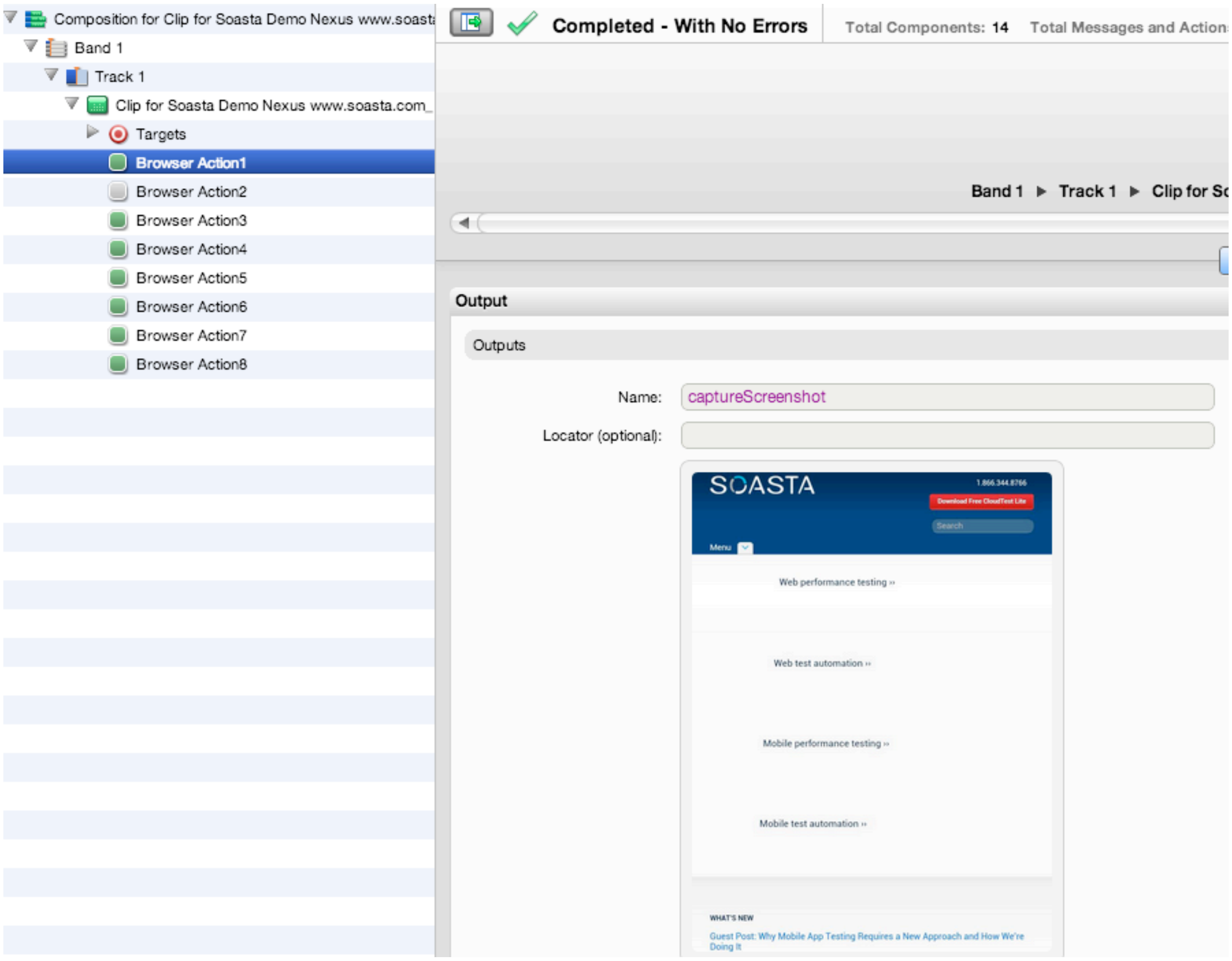
Customize result success/error messages

3. Click the Match field drop-down, and select *Glob* and enter the match text. Enter *What's New* to find the heading text that we saw above after searching "What's New" in the outputBodyText item in results.
4. Click Save on the Clip Editor toolbar.

Analyzing Validations in Results

Now that we've added three validations to the test clip, let's play the test composition again.

1. Switch back to the Draft Composition tab if it's still open and click the Play button, or, if it's not open use the Play in Test Composition command to play the saved clip in a new draft composition.
2. When Result Details appears, select Browser Action1 in the tree and then select it as shown maximize the Output panel.


Composition for Clip for Soasta Demo Nexus www.soast
Band 1
Track 1
Clip for Soasta Demo Nexus www.soasta.com_
Targets
Browser Action1
Browser Action2
Browser Action3
Browser Action4
Browser Action5
Browser Action6
Browser Action7
Browser Action8
Band 1 ▶ Track 1 ▶ Clip for Sc
Output
Outputs
SOASTA
1.866.344.8766
Download Free CloudTest Lite
Search
Menu
Web performance testing »
Web test automation »
Mobile performance testing »
Mobile test automation »
WHAT'S NEW
Guest Post: Why Mobile App Testing Requires a New Approach and How We're Doing It

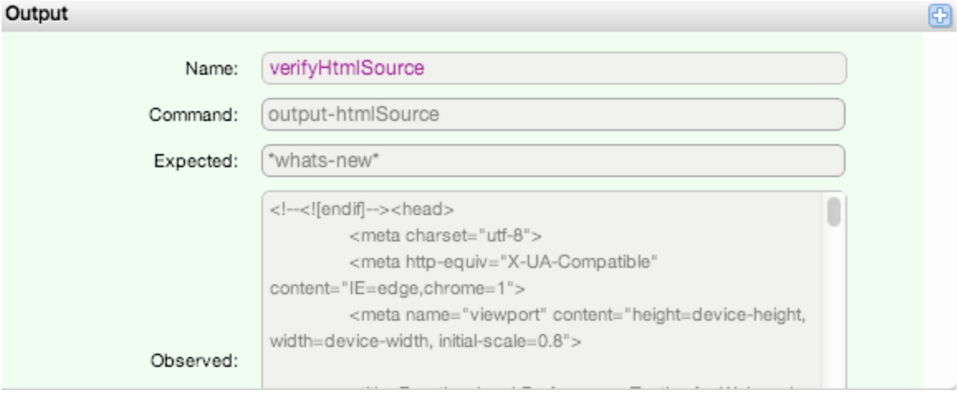
Completed - With No Errors

Total Components: 14 Total Messages and Action

Name: captureScreenshot

Locator (optional):

5. Select Browser Action4 and click each of its validations in the Waits and Validations panel. The Outputs panel shows the validation detail and flashes green for pass or red for fail.



Output

Name: verifyHtmlSource

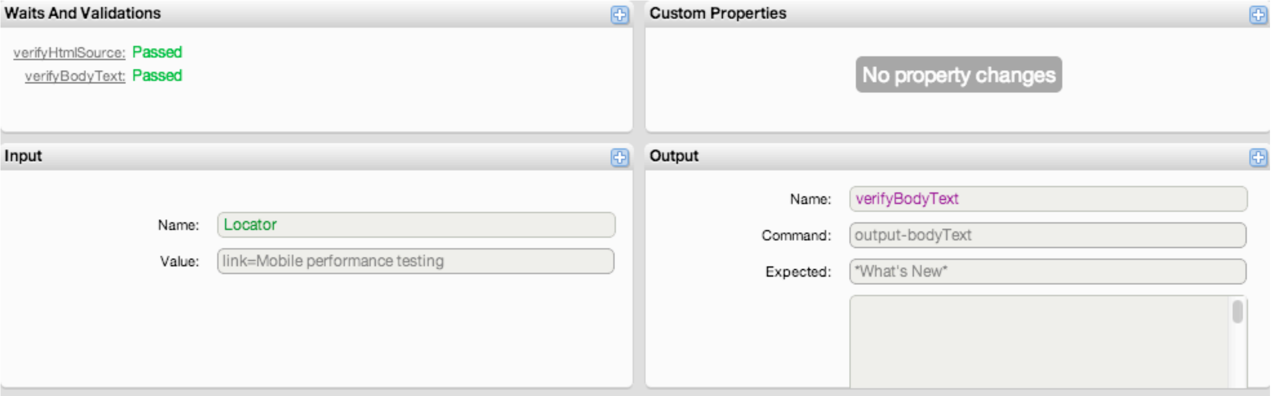
Command: output-htmlSource

Expected: *whats-new*

Observed:

```
<!--<![endif]--><head>
            <meta charset="utf-8">
            <meta http-equiv="X-UA-Compatible"
content="IE=edge,chrome=1">
            <meta name="viewport" content="height=device-height,
width=device-width, initial-scale=0.8">
```

3. Collapse the Monitored Device(s) panel to get a better view of the Waits and Validations and Output panels.



Waits And Validations

verifyHtmlSource: Passed
verifyBodyText: Passed

Custom Properties

No property changes

Input

Name: Locator

Value: link=Mobile performance testing

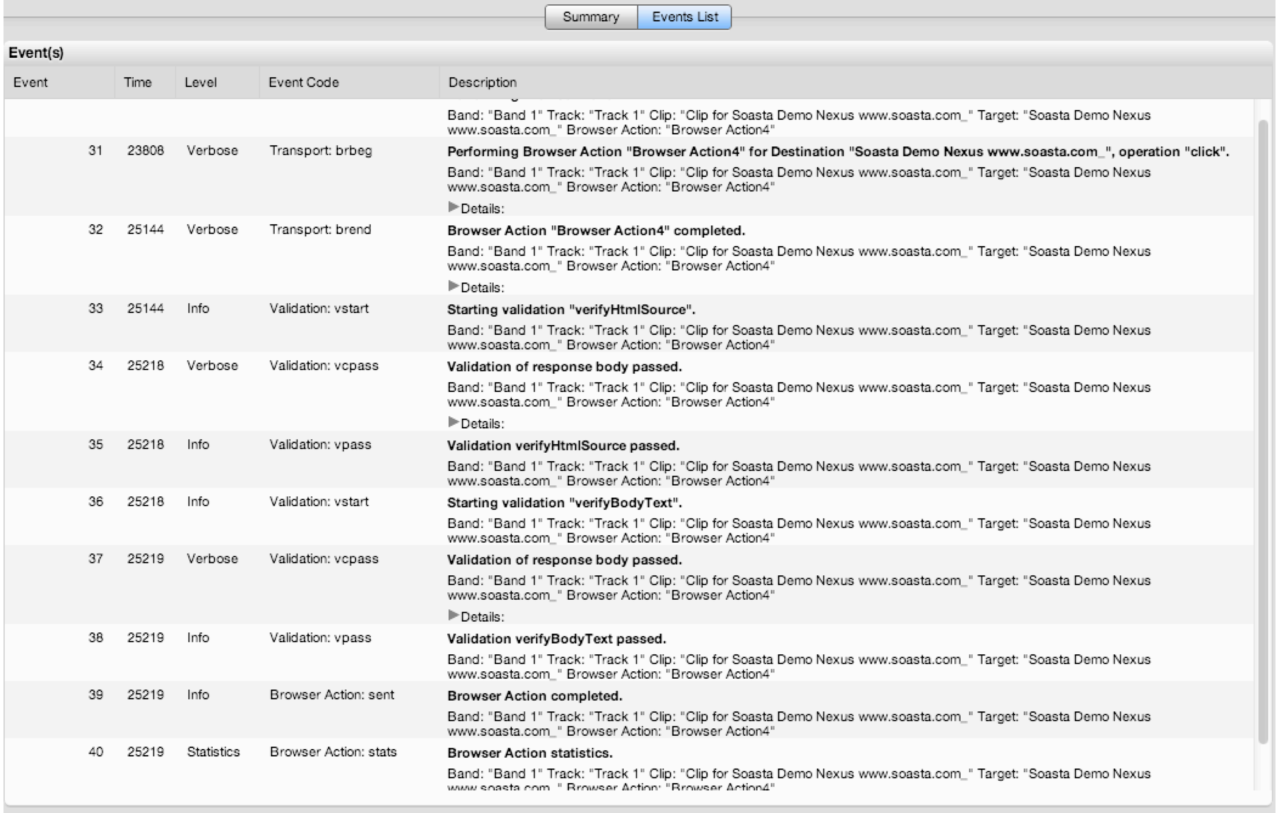
Output

Name: verifyBodyText

Command: output-bodyText

Expected: *What's New*

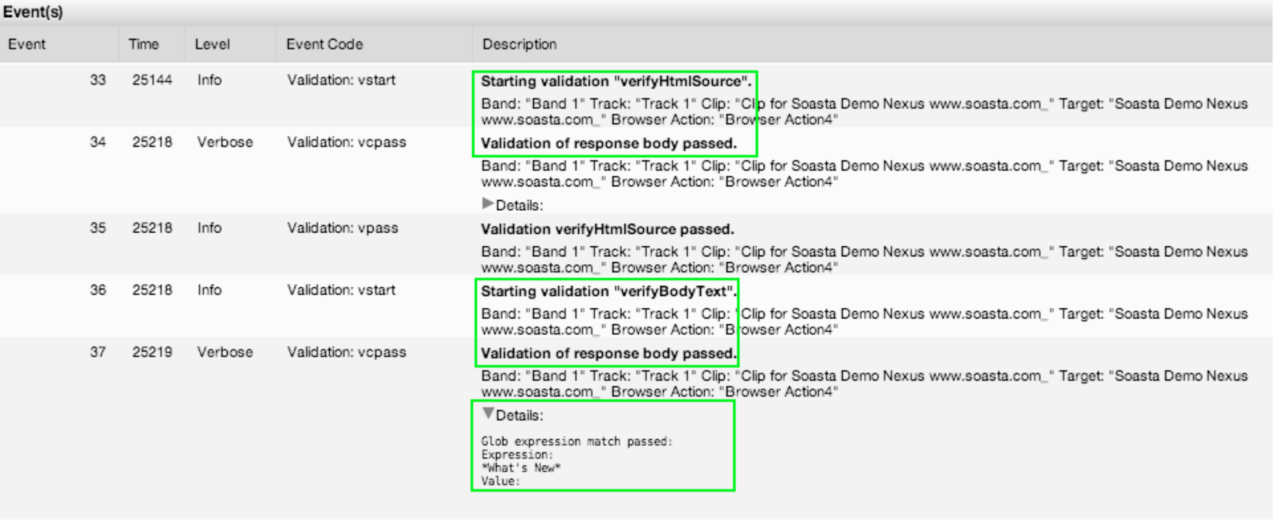
6. Click the Events list tab for the given selection to view action-related events, including validations.



Event(s)

Event	Time	Level	Event Code	Description
				Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4"
31	23808	Verbose	Transport: brbeg	**Performing Browser Action "Browser Action4" for Destination "Soasta Demo Nexus www.soasta.com_", operation "click".** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4" ▶Details:
32	25144	Verbose	Transport: brend	**Browser Action "Browser Action4" completed.** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4" ▶Details:
33	25144	Info	Validation: vstart	**Starting validation "verifyHtmlSource".** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4"
34	25218	Verbose	Validation: vcpass	**Validation of response body passed.** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4" ▶Details:
35	25218	Info	Validation: vpass	**Validation verifyHtmlSource passed.** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4"
36	25218	Info	Validation: vstart	**Starting validation "verifyBodyText".** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4"
37	25219	Verbose	Validation: vcpass	**Validation of response body passed.** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4" ▶Details:
38	25219	Info	Validation: vpass	**Validation verifyBodyText passed.** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4"
39	25219	Info	Browser Action: sent	**Browser Action completed.** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4"
40	25219	Statistics	Browser Action: stats	**Browser Action statistics.** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4"

Click the Details arrow to inspect validation event details.



Event(s)

Event	Time	Level	Event Code	Description
33	25144	Info	Validation: vstart	**Starting validation "verifyHtmlSource".** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4"
34	25218	Verbose	Validation: vcpass	**Validation of response body passed.** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4" ▶Details:
35	25218	Info	Validation: vpass	**Validation verifyHtmlSource passed.** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4"
36	25218	Info	Validation: vstart	**Starting validation "verifyBodyText".** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4"
37	25219	Verbose	Validation: vcpass	**Validation of response body passed.** Band: "Band 1" Track: "Track 1" Clip: "Clip for Soasta Demo Nexus www.soasta.com_" Target: "Soasta Demo Nexus www.soasta.com_" Browser Action: "Browser Action4" ▼Details: Glob expression match passed: Expression: *What's New* Value:

Appendix I: Adding TouchTestIDs to an Android (Hybrid) App

For HTML-based targets, including web sites recorded with TouchTest Web for Android, TouchTest Web utilizes the HTML `id` attribute as the TouchTest ID.

Developers and testers need only include the id attribute for any element on the given page where a TouchTestID is desired.

For example, in the following HTML code, the `id="hello"` will be recorded as the TouchTestID:

```
<input type="submit" value="Hello!" onClick="buttonClicked()” id=“hello">
```

SOASTA, Inc.
444 Castro St.
Mountain View, CA 94041
866.344.8766
http://www.soasta.com