

SOASTA 53.05 (CloudTest/TouchTest 7419.145)

Oct 10, 2014

Table of Contents

SOASTA 53.05 (CloudTest/TouchTest 7419.xxx).....	1
Features	1
CloudTest/TouchTest	1
Test Suites and Nested Compositions	1
Nested Compositions	1
Test Suite Dashboard	1
Using the Launch Composition Clip Element.....	2
Launch Composition Options	4
Building a Test Suite	6
Test Suite Dashboard.....	8
Accessing the Test Suite Dashboard	11
Enhancements	12
TouchTest	12
sCommand: Change to DeviceAgent/DeviceCloud Parameter.....	12
Set the Device Agent or Cloud from sCommand	12
Set the Device Agent or Cloud from Jenkins.....	12
Bugs Fixed	14
CloudTest.....	14
TouchTest	15
SOASTA 53.04 (CloudTest/TouchTest 7419.118)	16
Features	16
CloudTest.....	16
WebSocket Support.....	16
WebSocket Recording.....	18
WebSocket Recording in Recording Editor	19
WebSocket Recording in Clip Editor	25

WebSocket Target Type	28
New Event Group and Wait Clip Elements	33
New Event Group Container	33
New Wait Clip Element	34
New WebSocket Clip Elements	35
WebSocket Event Handler - Events	35
WebSocket Actions	35
WebSocket Event Handler Extraction Methods	38
Message Type Extraction in WebSockets	39
Time Stamp Extraction in WebSockets	39
WebSocket Event Handler Options	41
WebSocket Analytics	41
Enhancements	45
CloudTest	45
User and Permission Management Enhancements	45
User Groups	45
Change or Transfer Ownership of Resource(s)	47
Full Control of Object(s) (User Admin)	49
New Monitor Administrator Privilege	53
Amazon EC2 Integration Enhancements	54
Virtual Private Cloud (VPC) Support	54
New Virtualization Type	56
New Instance Types	56
Support for HTML Reports via Template	57
Importing an HTML Report Template	59
Creating an HTML Report	60
Show servers across tenants to Tenant Administrators (75418)	61
Servers Are Restricted to Tenant (34800)	61
Support for On-Premises Results Databases (34799)	62

Find and Replace now matches HTTP Header names.....	64
Analytics.....	65
Widget-on-Widget Layout and Edge Constraints.....	65
General Dashboard Improvements and Changes.....	68
Ability to specify type and PIOPS when creating a volume.....	69
Exporting Widget data to Excel grabs all data in all trees (80301).....	70
Ability to specify type and PIOPS when creating a volume.....	70
Exporting Widget data to Excel grabs all data in all trees (80301).....	72
Bugs Fixed	73
CloudTest.....	73
TouchTest	80

SOASTA 53.05 (CloudTest/TouchTest 7419.145)

Features

CloudTest/TouchTest

Test Suites and Nested Compositions

This release introduces two important new features: Nested Compositions and the new Test Suite Dashboard. Both are relevant to functional testers using CloudTest and TouchTest.

In prior releases, functional testers were encouraged to get their test results/reports from Jenkins (or other CI tools integrated with CloudTest and/or TouchTest). However, most users found Jenkins CI reports challenging.

Now, CloudTest and TouchTest functional testers can launch compositions from other compositions using the new Nested Compositions feature, and view Test Suite results in the new Test Suite Dashboard. The ability to leverage Jenkins CI and other CI tools remains undiminished.

These two new features are distinct, but they can be used in tandem for debugging functional tests and to quickly identify regressions.

Nested Compositions

This feature introduces a new clip element type, Launch Composition. This clip element can be placed into the test clip the same as any other clip element.

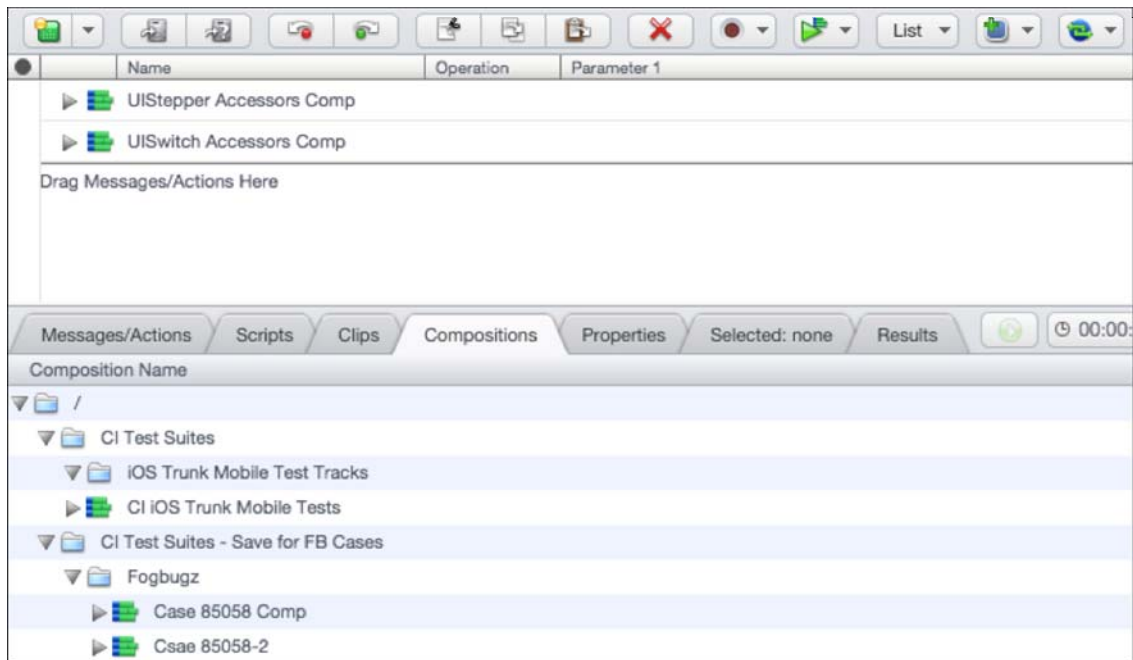
Test Suite Dashboard

This new System Dashboard is purpose-built for web and mobile Functional Testing and allows the tester to easily browse through the history of a test suite and the test compositions in it, right down to the action level where validations pass/fail, and where regressions can be detected and fixed.

Using the Launch Composition Clip Element

The Launch Composition clip element is a call to a Composition from another Composition (via a Test Clip in the calling Composition).

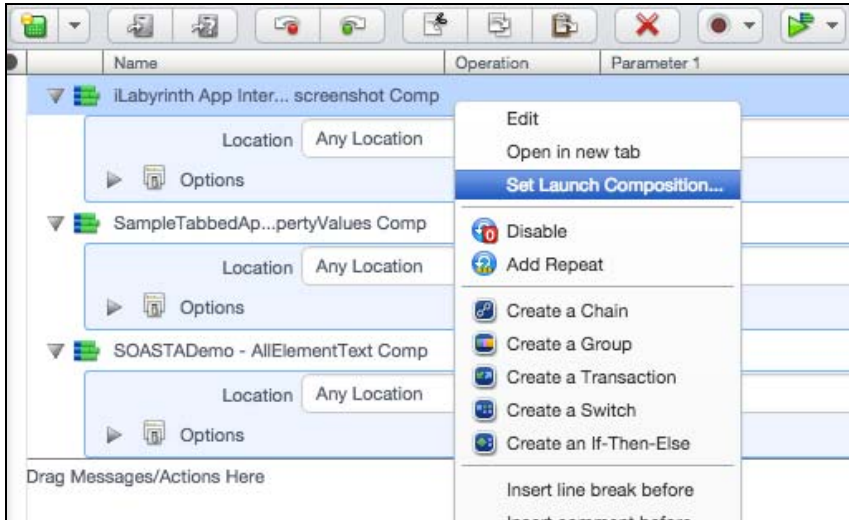
A Composition is added into a Test Clip via the Clip Editor lower panel, Compositions tab, and the resulting clip element is a Launch Composition clip element.



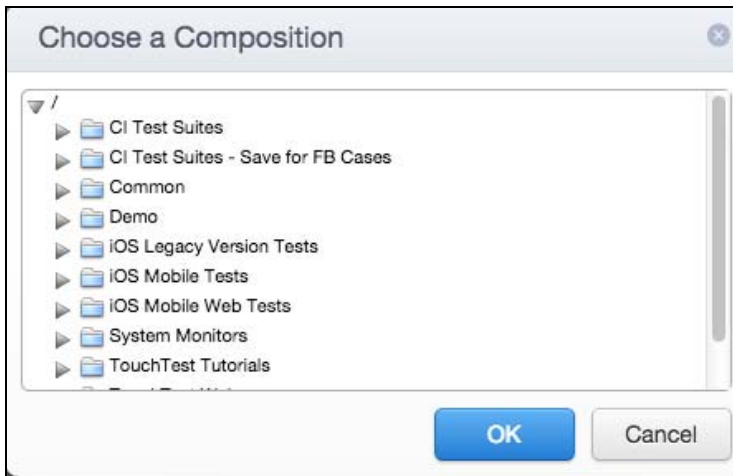
Adding a Launch Compositions clip element into the Clip works precisely the same as adding any other clip element—double click the item to place it at the insertion point, or, drag it into place in the Clip Editor workspace.

More than one Launch Composition clip element can be placed into a Test Clip(s) in order to create a Test Suite.

Additionally, the Set Launch Composition context command can be used to set another Composition as a Launch Composition.



When this command is selected, the Choose a Composition box appears. Select the Composition to set as a Launch Composition.



Launch Compositions have all of the following capabilities:

- They can Repeat, be placed in Transactions, have properties, and all of the things that Clip elements do.
- They have properties including “whether or not to wait for the comp to complete”, “whether failure should fail the parent”, “optional name for the result”, etc.
- When the parent Composition that contains the Clip is played its result can be viewed in the Test Suite Dashboard (which loads by default); an individual result is also created for each Launch Composition.
- Results of parent Compositions show the Launch Composition elements in the tree, but not its children

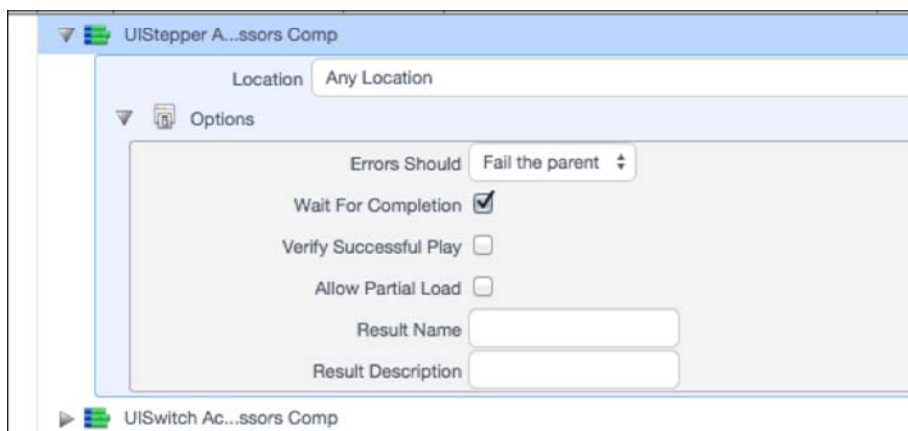
Note: In this release, Device Selection is done in the nested composition itself and cannot be overridden in the parent composition.

Launch Composition Options

Launch Compositions, like Nested Clips, have their own *in situ* properties that are applicable only to the nested instance (these options are distinct from the stand-alone properties that apply to the stand-alone Composition).

After adding a Launch Composition to your Clip, expand the new clip element in Clip Editor, List View to display its Options.

Alternately, select the Launch Composition clip element and view it in the Selected: <...> panel.



- **Location** – For non-functional tests specify the Maestro Location to use. Refer to [Specifying Maestro Locations](#) for more information. Locations doesn't apply to functional testing so functional testers can accept the default.

- **Errors Should** – This drop-down specifies which **failure action** to take if an error occurs in the Launch Composition. Select one of the standard Failure Actions that apply to all clip elements
 - If the **failure action** is set to record only, the validation outcome will be recorded in the result, and the launch composition will continue whether the validation passes or fails.
 - If the **failure action** is set to fail the parent, the parent clip will stop if the validation fails.
- **Wait For Completion** – Checked by default.
 - If checked, this Clip Element will not complete execution until the Composition completes.
 - If unchecked, this Clip Element will complete once the Composition has been started, and will not wait for its completion.
- **Verify Successful Play** – Unchecked by default.
 - If checked, this Clip Element will fail if the launched Composition does not complete with status *Completed*. What then happens is determined by the value of the *Failure Action* attribute.
 - This attribute only has meaning if the Wait For Completion attribute is checked. If that attribute is unchecked, this attribute is not relevant.
- **Allow Partial Load** – Unchecked by default.
 - Checked indicates that Composition play should proceed even if all of the necessary servers were not found. Locations doesn't apply to functional testing so functional testers can leave this unchecked.
- **Result Name** – Optionally, specify an alternate name for the Result name (e.g. other than the default name).
- **Result Description** – Optionally, specify a description for the result.

Building a Test Suite

Test Suites can be built in a number of ways, including adding all of the relevant Compositions as Launch Compositions into a test clip that resides in a parent test composition with no other test clips.

Playing this parent Composition creates a result for the Test Suite (although individual test composition results are also created).

More sophisticated Test Suites can be built that contain many such Clips, each of which contains one or more Launch Compositions.

For example, a Test Suite for mobile apps and devices such as the example below can be arranged by Track, each of which corresponds to a device type, OS version, or to a network type. The Clips that have Launch Compositions are organized into the appropriate Track.



When you play a parent Composition (e.g. your Test Suite), the Composition Editor opens to the Play tab with the Test Suite Dashboard in display.

Functional Summary: iOS Test Suite

Compositions	Validations	Clips
100% (30 / 30)	100% (131 / 131)	100% (45 / 45)

Compositions Overview

Composition	Last Result	Prior 10 Results	Start Time	Time To Complete
CI iOS 3G iPhone Devices	PASSED	●●●●●●●●●●	10/7/2014 12:19 am	54 s
CI iOS 7 iPad Devices	PASSED	●●●●●●●●●●	10/7/2014 12:19 am	1:18 m:s
CI iOS 8 iPad Devices	PASSED	●●●●●●●●●●	10/7/2014 12:19 am	2:28 m:s
CI iOS iPad Devices 1	PASSED	●●●●●●●●●●	10/7/2014 12:19 am	21 s
CI iOS iPad Devices 2	PASSED	●●●●●●●●●●	10/7/2014 12:19 am	45 s
CI iOS iPad Devices 3 1	PASSED	●●●●●●●●●●	10/7/2014 12:19 am	1:53 m:s
CI iOS iPad Devices 4	PASSED	●●●●●●●●●●	10/7/2014 12:19 am	1:10 m:s
CI iOS iPad Devices 5	PASSED	●●●●●●●●●●	10/7/2014 12:19 am	1:04 m:s
CI iOS iPad Devices 6	PASSED	●●●●●●●●●●	10/7/2014 12:19 am	1:24 m:s
CI iOS iPod_iPhone Devices 1	PASSED	●●●●●●●●●●	10/7/2014 12:19 am	46 s

Result Details

Element Status All | Element Type All | Operation All

- CI iOS 3G iPhone Devices
 - UIStepper Accessors Comp
 - UISwitch Accessors Comp

Summary | Events List

General

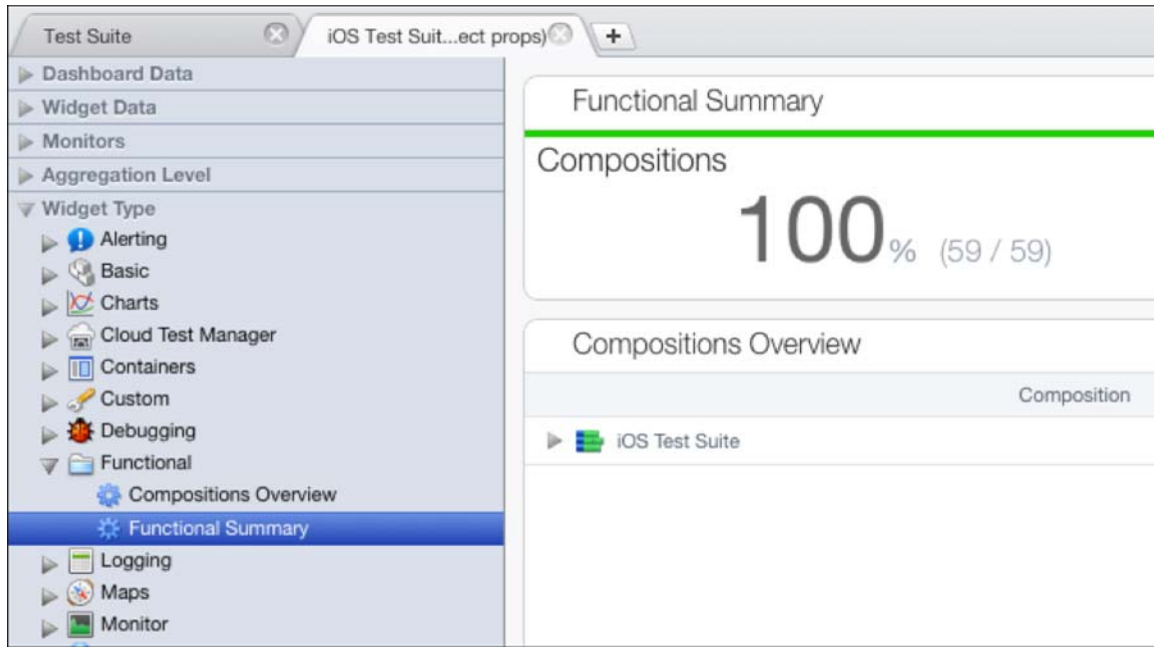
Name: CI iOS 3G iPhone Devices

Effective Duration	Avg. Response Time	Total Message Bytes	Effective Message Throughput
Full Duration	0 ms.	Sent: 0	0 msgs/sec.
53 sec.	Min: 0 ms.	Received: 0	0 bytes/s
Start: 538 ms.	Max: 0 ms.		0 bits/s
	Agg: 0 ms.		

The Test Suite Dashboard presents functional data as a summary of the output from a "test suite." The Test Suite is used to analyze the results of a functional test across many results, and in the TouchTest use case, mobile devices. The Test Suite Dashboard incorporates key metrics, and the results of a functional test, including Success Summary [the % and number of successful Compositions, Clips, Validations]; Details about the Compositions and its test results [pass/fail, time/date of last test run], and the ability to drill down into details.

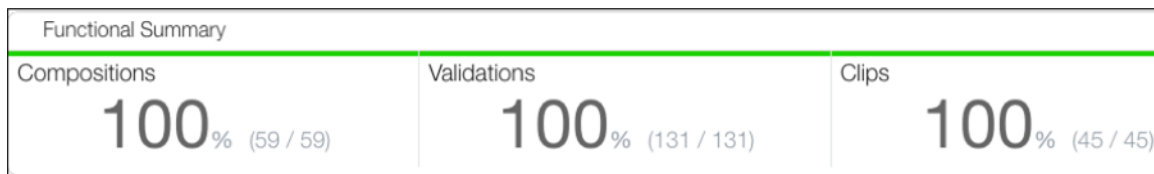
Test Suite Dashboard

As part of the Test Suite Dashboard, a new Widget Type category—Functional—has been added to the Widget Type list.



The Test Suite Dashboard includes the following widgets:

Functional Summary – This widget displays Summary Statistics, including the Success percentage of successful out of the total for Compositions, Validations, and Clips. The Functional Summary can also be added to custom dashboards.



Compositions Overview – This widget shows the components of the parent Composition and their history by component.

Compositions Overview					
Composition	Last Result	Prior 10 Results	Start Time	Time To Complete	
▶ CI iOS 3G iPhone Devices	● PASSED	●●●●●●●●●●	10/7/2014 12:19 am	54 s	
▶ CI iOS 7 iPad Devices	● PASSED	●●●●●●●●●●	10/7/2014 12:19 am	1:18 m:s	
▶ CI iOS 8 iPad Devices	● PASSED	●●●●●●●●●●	10/7/2014 12:19 am	2:28 m:s	
▶ CI iOS iPad Devices 1	● PASSED	●●●●●●●●●●	10/7/2014 12:19 am	21 s	
▶ CI iOS iPad Devices 2	● PASSED	●●●●●●●●●●	10/7/2014 12:19 am	45 s	
▶ CI iOS iPad Devices 3 1	● PASSED	●●●●●●●●●●	10/7/2014 12:19 am	1:53 m:s	
▶ CI iOS iPad Devices 4	● PASSED	●●●●●●●●●●	10/7/2014 12:19 am	1:10 m:s	
▶ CI iOS iPad Devices 5	● PASSED	●●●●●●●●●●	10/7/2014 12:19 am	1:04 m:s	
▶ CI iOS iPad Devices 6	● PASSED	●●●●●●●●●●	10/7/2014 12:19 am	1:24 m:s	
▶ CI iOS iPod_iPhone Devices 1	● PASSED	●●●●●●●●●●	10/7/2014 12:19 am	46 s	

Result Details

The Composition Overview widget includes the following columns:

- A Composition column that can be expanded to drill into child components
- A Last Result column that shows the most recent result (pass/fail) of the specific item for that line (e.g. in the result that was opened)
- A Prior 10 Results column that shows the pass/fail result for each row *before* the Last Result (most recent from left to right).
- A Start Time column that shows the last start time for the item in the row
- A Time to Complete column that shows the total time that the item (a Composition, Clip, or Validation) last took to execute

The Compositions Overview tree starts from the Clips inside the parent Composition. Only Compositions, Clips, and Validations are shown here, while Tracks, Bands, Messages, App, and all grouping Clip Elements are omitted.

The selected dot in Last 10 Results drives the Result Details selection below

▶ CI iOS 3G iPhone Devices	● PASSED	●●●●●●●●●●	10/7/20
▼ CI iOS 7 iPad Devices	● PASSED	●●●●●●●●●●	10/7/20
▼ iOS7 EverythingHasIds - Browser Actions Comp	● PASSED	●●●●●●●●●●	10/7/20
▼ EverythingHasIds - Browser Actions	● PASSED	●●●●●●●●●●	10/7/20
✓ verifyElementText	● PASSED	●●●●●●●●●●	10/7/20
✓ verifyElementText	● PASSED	●●●●●●●●●●	10/7/20
✓ verifyElementText	● PASSED	●●●●●●●●●●	10/7/20
✓ verifyElementText	● PASSED	●●●●●●●●●●	10/7/20
✓ verify r1-c1	● PASSED	●●●●●●●●●●	10/7/20
✓ verify r1-c2	● PASSED	●●●●●●●●●●	10/7/20

Result Details – This is the existing Result Details dashboard for that Result without the cover flow and with the left-hand tree starting at the selected element from above. Users can also select in the Result Details tree itself to navigate through test elements.

As already noted, the Result Details widget is also driven by the selected result dot in the Last 10 Results column (shown below).

The screenshot shows a table of test results at the top, with the fourth row selected. Below the table is the 'Result Details' widget for the 'Verify Clicks' operation. The widget is divided into several sections: General, Waits And Validations, Custom Properties, Input, and Output. The 'Verify Clicks' operation is shown as 'noOp' with a start time of 14.406 sec and a response time of 741 ms. The 'Waits And Validations' section shows three 'verify r1-c1' operations, all of which passed. The 'Custom Properties' section shows 'No property changes'. The 'Input' section is empty. The 'Output' section shows a validation for the 'verify r1-c1' operation, with the command 'output-attributeValue' and the attribute locator 'id=r1-c1@style'. The expected and observed values are both 'background-color: rgb(187, 255, 187)'. The 'Verify Clicks' operation is shown as 'noOp' with a start time of 14.406 sec and a response time of 741 ms.

In this iteration of the Result Details widget, the Navigation Tree contains all Clip Elements (Groups, Chains, etc.) that are children of the Clip in the existing Result Details dashboard.

Note: Errors indicated in the Result Details UI will "bubble up" to the parent item(s) of the failed element. This was true of errors in the prior release *only if* the Failure Action for the given element was set to "Also fail the parent."

An element such as a Composition may succeed, but if any of its children have failed, then the Composition icon in the tree will show a red-x. This change applies to all contexts of the Result Details widget as of SOASTA 53.05—whether it is in view in the full Result Details Dashboard, here in the Test Suite Dashboard, or in any custom dashboard that includes the Result Details widget.

Additionally, bubble-up errors are shown all the way to the composition-level (in the prior release, they were shown only up to the track-level if the Failure Action was set to fail the parent).

Selection in the Compositions Overview will navigate to the corresponding object in Result Details below. Additionally, the Result Details widget in this context

shows only the relevant panels (for example, if the selection is a Validation, the relevant Accessor panels are shown).

Accessing the Test Suite Dashboard

The tester can enable to access the Test Suite Dashboard by any one of the following ways:

- Opening a parent Composition in the Composition Editor, Results tab.
- Opening the result of a parent Composition from Central (in which case, the Test Suite Dashboard is shown by default)
- Opening the Build in Jenkins and viewing the test result of a parent Composition

Enhancements

TouchTest

sCommand: Change to DeviceAgent/DeviceCloud Parameter

In the prior release(s), the DeviceAgent/DeviceCloud system property was used in sCommand via the systemproperty parameter as follows:

```
> ./scommand cmd=play wait username=jdoe1 password=secret
name=/tt/droidfish_s3 systemproperty="Band 1/Track
1/droidfish1/deviceS3 droidfish/DeviceCloud=cloud1"
```

Set the Device Agent or Cloud from sCommand

Now, this parameter can be used explicitly, apart from the System Property:

```
> ./scommand cmd=play wait username=jdoe1 password=secret
name=/tt/droidfish_s3 systemproperty="Band 1/Track
1/droidfish1/deviceS3 droidfish" devicecloud=cloud1
```

Set the Device Agent or Cloud from Jenkins

Use the following steps to override the composition Device Agent or Cloud from Jenkins:

1. Add the Play Composition(s) module to the Jenkins job.
2. Specify the CloudTest Server (from among the configured servers, refer to the [TouchTest Jenkins CI for iOS Tutorial](#) for complete coverage of using the CloudTest Jenkins Plugin, as well as for steps to add your CloudTest server to Jenkins CI).
3. Specify the composition path. If you are running a test suite, specify the path to the top-level test composition.
4. In the Additional Options field, specify the full path to the device agent or cloud, using lowercase letters only. For example:

```
systemproperty="Band 1/Track 1/droidfish1/deviceS3
droidfish" devicecloud=cloud1
```

```
systemproperty="Band 1/Track 1/droidfish1/deviceS3
droidfish" deviceagent=testTablet3
```


Bugs Fixed

CloudTest

85225: Jenkins Plugin fails CloudTest/TouchTest handshaking in some 53 builds

A regression in SOASTA 53 caused the 'Test Connection' button to fail in the Jenkins plugin.

85130: CloudTest does not respond to WebSocket server pings

The Cloud Test instance was unexpectedly unresponsive.

85091: Running Compositions can only be stopped from Player Status - Compositions

While attempting to stop a running composition/test, clicking the Stop button on the Composition Editor, Play tab had no effect. However, using the Central, Player Status - Compositions Stop succeeded.

84940: RSDB keeps showing "Cancelling" when terminated

After tear down was completed, the Central, RSDB list, Database State column continued to display "Cancelling" as the DB status.

84936: Failed to set the 'valueAsString' property on 'SVGLength': The value provided ('NaN') is invalid

A null check has been to stop a calculation from returning NaN.

84869: Server hours should not be checked during grid deploy, if "Track server hours" is disabled for the license

Replace failed servers

84449: Page resources with a tab character in them causes HTTP 404

Tab characters encountered as a page resource weren't handled as expected.

82853: Results Service query error ("No row with the given identifier exists:...)

Events can be written before the dimension values that they link to. This is already handled in the getEvents() query method. But we weren't handling the situation where the container's parent was a proxy

Copyright 2014. CloudTest is a registered trademark of SOASTA, Inc. and/or its affiliates. Other names may be trademarks of their respective owners.

TouchTest

84868: Android TouchTest Web: Initially opening a page is taking much longer than it did in previous versions [[requires new TouchTest Web](#) | [[requires new TouchTestDriver](#)]

This fixes the "slow loading webpage" problem.

84365: Android TouchTest Web: During execution of web clip an alert appears and blocks clip execution [[requires new TouchTest Web](#) | [requires new TouchTestDriver](#)]

This fixes the Alert pop-up concerning "Cannot call method 'appendChild' of null" during playback.

84058: Scroll action issue while playback [[requires new TouchTestDriver](#)]

Support for horizontal scroll on UIScrollView.

80212: Specific pop-up on Android application is not being recognized on Android Tablet only [[requires new TouchTestDriver](#)]

While running on a given Android tablet, a popup window wasn't recognized.

SOASTA 53.04 (CloudTest/TouchTest 7419.118)

Features

CloudTest

WebSocket Support

This release introduces full-fledged WebSocket protocol support for CloudTest and CloudTest Lite users, including the ability to record WebSocket traffic via the Recording Editor or directly into the Clip Editor.

As part of this support, this release introduces:

- A new WebSocket target type
- A new Event Group container
- A new WebSocket Event Handler in which all WebSocket event processing happens, but other WebSocket processing (such as Send, Close, and Open) can occur elsewhere.
 - The WebSocket Event Handler includes six new WebSocket Event Containers that work together to handle WebSocket-related events:
 - Open
 - Close
 - Error
 - Idle
 - Message
 - Default
- The new Web Socket actions include:
 - WebSocket Open
 - WebSocket Send
 - WebSocket Event Message
 - WebSocket Close

- A new Wait action is also provided in this release. It is useful in WebSocket sessions; however, it can also be used in CloudTest test clips other than those based on a WebSocket target.

WebSocket Recording

Users can now seamlessly record full-duplex WebSocket interactions between browser/client and server. WebSocket protocol connections are recorded in the CloudTest Recording Editor or directly into the clip from within the Clip Editor.

Since CloudTest auto-detects the WebSocket protocol using the same techniques used to record any HTTP(s) session it is not necessary to know the underlying details to successfully test your WebSocket site.

During HTTP(S) recording, CloudTest auto-detects the WebSocket format and, if found, any or all of the new objects listed in the section above.

Our recording techniques will automatically create any or all of the necessary underlying target(s) alongside other targets that are created, whether recording via the Recording Editor or directly into the Clip Editor.

Since WebSocket recording is as seamless as normal HTTP(S) recording, we will cover recording before we get into more detail about the new objects.

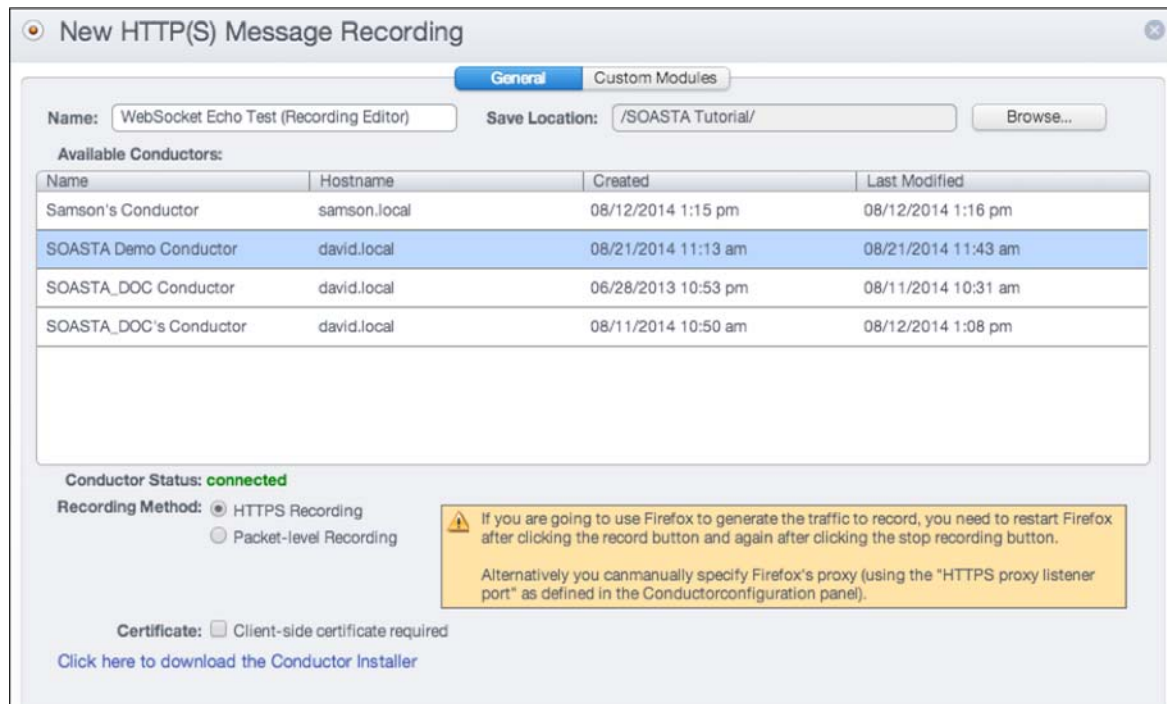
In the following sections, this capability is demonstrated using the basic [Echo Test at websocket.org](#). The steps to record either from the Recording Editor or Clip Editor are presented.

TIP: As with any HTTP(s) recording we will run CloudTest in one browser, and then record traffic from a second browser (in our example, we use Safari). In the second browser, ensure that the cache is cleared before proceeding. Refer to [Creating an HTTP Recording](#) for more about best recording practices.

WebSocket Recording in Recording Editor

Use the following Recording Editor steps to record WebSocket traffic from the example or your own WebSocket site.

1. Start the Conductor to use for recording.
2. In the first browser, launch CloudTest, and then navigate to Central > Recordings and click New.



1. Once ready, click Record in the New HTTP(S) Message Recording box.
2. With cache clear, use the second browser to launch your site (or use the Echo Test URL from above).

TIP: Give the recording a unique name, choose a Save Location, and then select a Conductor as you would for any recording.

3. Perform the WebSocket-relevant steps on your site. Or, if you're using the Echo Test example, do the following on the Echo Test demo page:
 - Click Connect
 - Click Send
 - Click Disconnect
 - Click Clear log
 - Check Use secure WebSocket (TLS)
 - Click Connect

- Click Send
 - Click Disconnect
 - Click Clear log
4. After the necessary steps are completed, click Stop on the Recording Editor toolbar.

Take a look at the resulting recording before converting it. For example, locate the first (WebSocket) Send action found in your recording. In the screenshot below, the sent text "Rock it with HTML5 WebSocket" is shown. Note the *Protocol: ws* value on the right.



5. Once ready, click Convert to a Clip. The Convert to Clip Wizard appears. See [Converting a Recording to a Clip](#) for more information about this wizard.
6. Proceed to the third page, Match Targets. Delete any non-relevant targets by selecting them individually and clicking the Delete (X) icon.

7. On the fourth page, Create New Targets, note the individual targets for your WebSocket site. These targets utilize a new target type provided in this release, WebSocket.

Convert To Clip Wizard

Name
Think Times
Match Targets
Create New Targets
Create Pages
Create Test Clip
Summary

Create New Targets

New Targets

Create	Target Name (Hover to see the complete target URL)
<input checked="" type="checkbox"/>	echo.websocket.org (websocket) (3)
<input checked="" type="checkbox"/>	echo.websocket.org (websocket) (3)

Potential SOAP Targets

Create	Target Name
No SOAP targets need to be created.	

Binary SOAP Endpoints (must enter a WSDL URL)

Create	Target Name
No binary SOAP targets need to be created.	

Proceed to create the new test clip and on the Summary screen, click the Continue button that corresponds to "Go to the Test Clip Editor for additional clip editing". If the Clip Editor is not already in List view, change to List View at this time.

If you used the websocket.org Echo Test, your clip should look similar to the clip shown below. Note the HTML page created for the page itself, and the two Event Groups representing the two cycles through Connect/Disconnect.

Name	Operation	Parameter 1
echo.html		
echo.html (1)	get	/echo.html
Delay1		3069 ms.
Delay4		5642 ms.
Delay2		5261 ms.
Event Group1		
Delay3		5930 ms.
Event Group2		

Drag Messages/Actions Here

1. In Clip Editor, List View, expand the first Event Group to examine it.

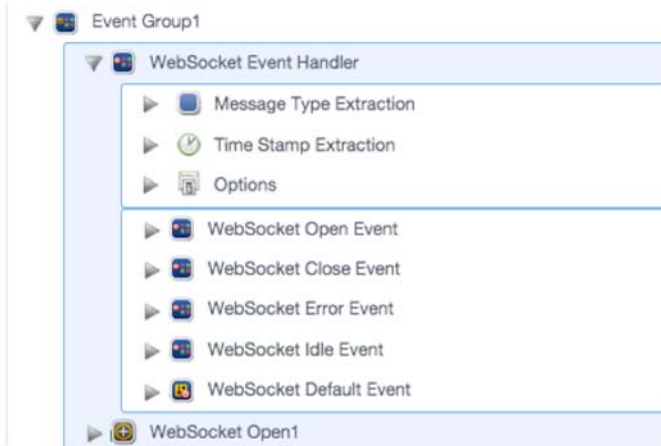
Event Group1		
WebSocket Event Handler		
WebSocket Open1		
echo.websocket.org		
echo.websocket.org (1)		
Delay3		
Event Group2		

Drag Messages/Actions Here

2. Note each of the following:

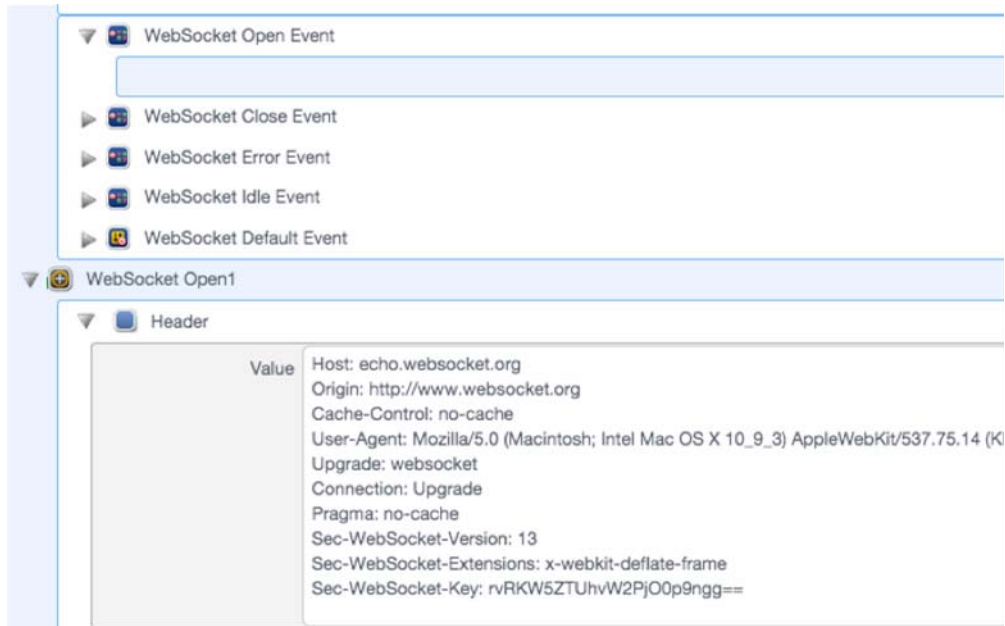
- The Event Group container

- The WebSocket Event Handler

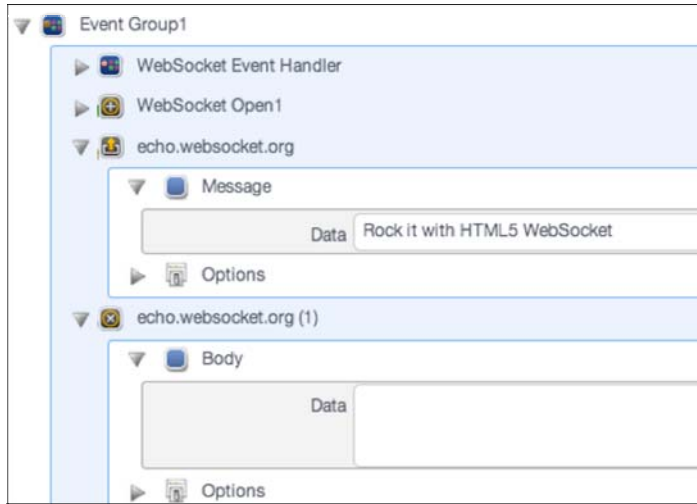


As noted elsewhere, all WebSocket Event Containers must be in WebSocket Event Handlers.

- The WebSocket Open Event (note that the Open Event and the Open Action are distinct from one another as shown below; where `WebSocket Open1` is the action)



- The WebSocket Send (echo.websocket.org, expanded below)
 - Locate and expand the first WebSocket Send (e.g. the same one from the Recording) and visually verify the *Rock it with HTML5 WebSocket* message, which is shown in the Data field



- The WebSocket Close (echo.websocket.org (1) shown above and below)

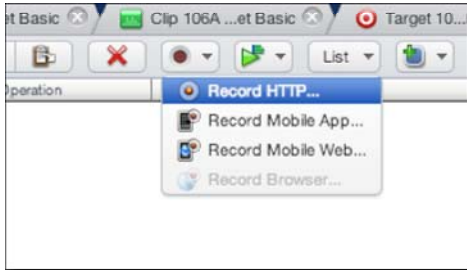


In subsequent sections, all of the new WebSocket objects are discussed. First, we'll show the Clip Editor recording steps. If you'd like to proceed with the play steps, you can skip the next section.

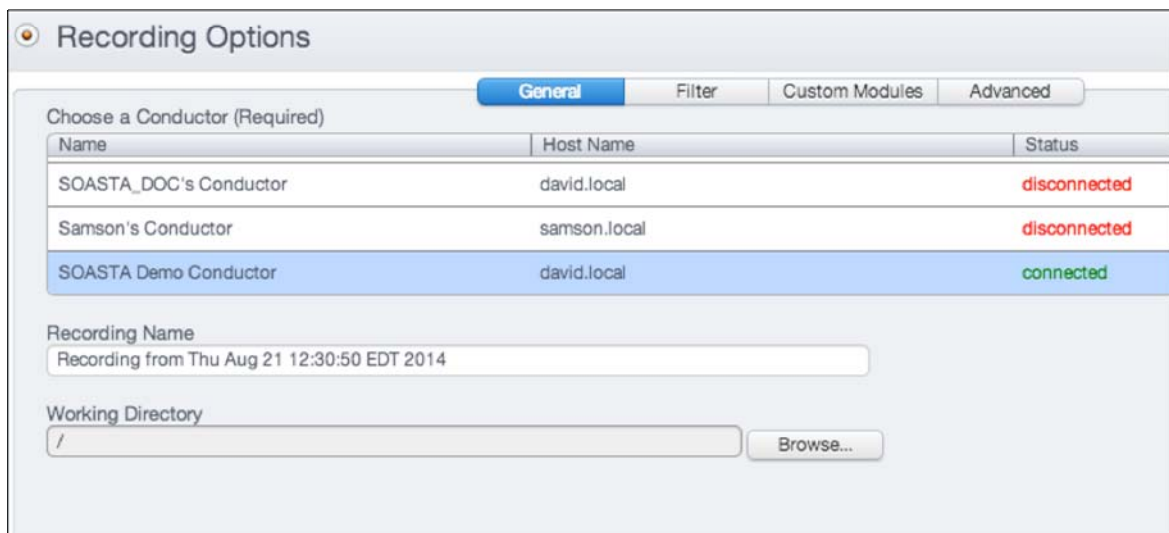
WebSocket Recording in Clip Editor

Use the following Recording Editor steps to record WebSocket traffic from the example or your own WebSocket site.

1. Start the Conductor to use for recording.
2. In the first browser, launch CloudTest, and then navigate to Central > Recordings and click New.



TIP: Give the recording a unique name, choose a Working Directory, and select a Conductor as you would for any recording.



3. With its cache cleared, open your site (or use the Echo Test URL from above) in the second browser.
8. Perform the WebSocket-relevant steps on your site. Or, if you're using the Echo Test example, do the following on the Echo Test demo page:
 - Click Connect
 - Click Send
 - Click Disconnect
 - Click Clear log

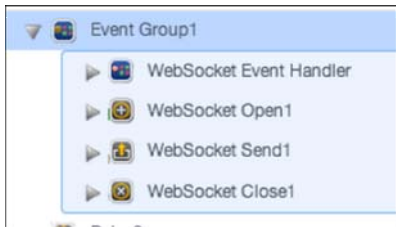
- Check Use secure WebSocket (TLS)
- Click Connect
- Click Send
- Click Disconnect
- Click Clear log

After the necessary steps are completed, click Stop on the Clip Editor toolbar. The resulting clip varies from the Recording Editor and Convert Clip version already created above only in the filtering (we deleted some unwanted targets along the way in that version).

If you used the websocket.org Echo Test, your clip should look similar to the clip shown below. Note the HTML page created for the page itself, and the two Event Group's representing the two cycles through Connect/Disconnect.



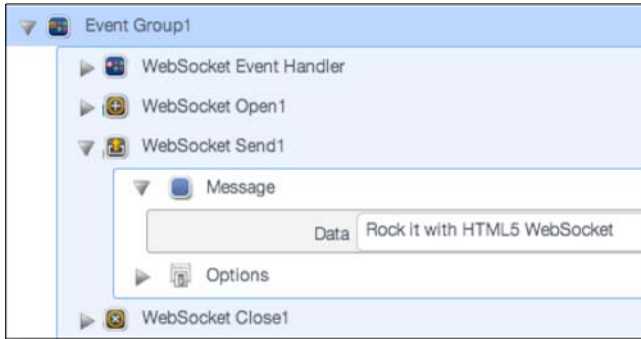
3. In Clip Editor, List View, expand the first Event Group to examine it.



Note the following:

- The new Event Group container
- The new WebSocket Event Handler

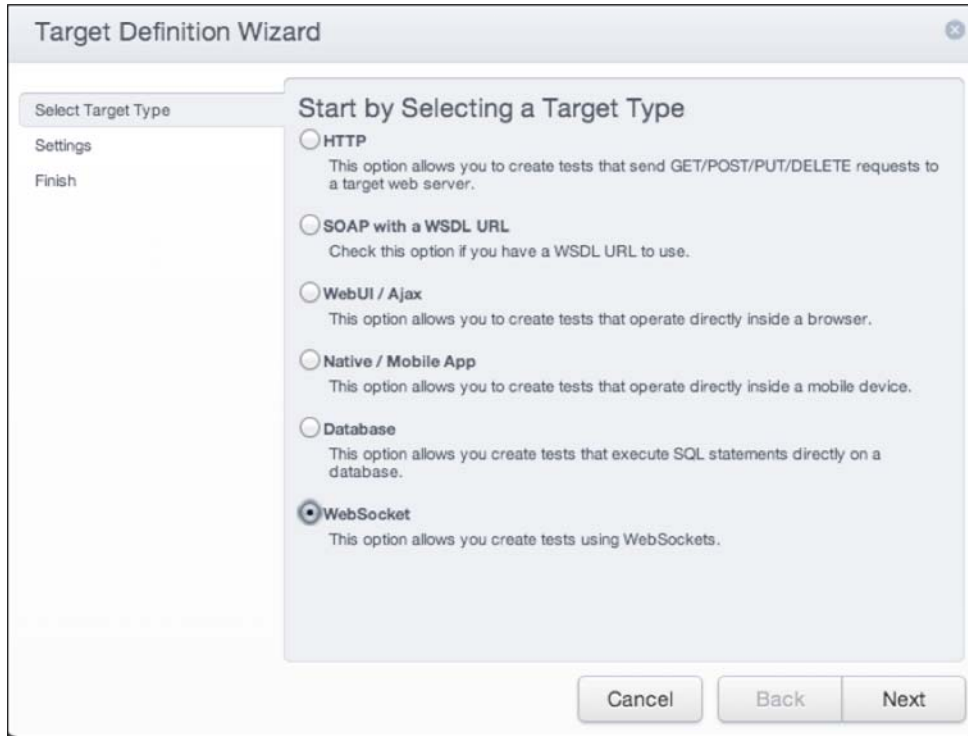
4. Locate and expand the first Send (e.g. the same one from the Recording) and verify the Rock it with HTML5 WebSocket text visually



In subsequent sections, all of the new WebSocket objects recorded above are detailed.

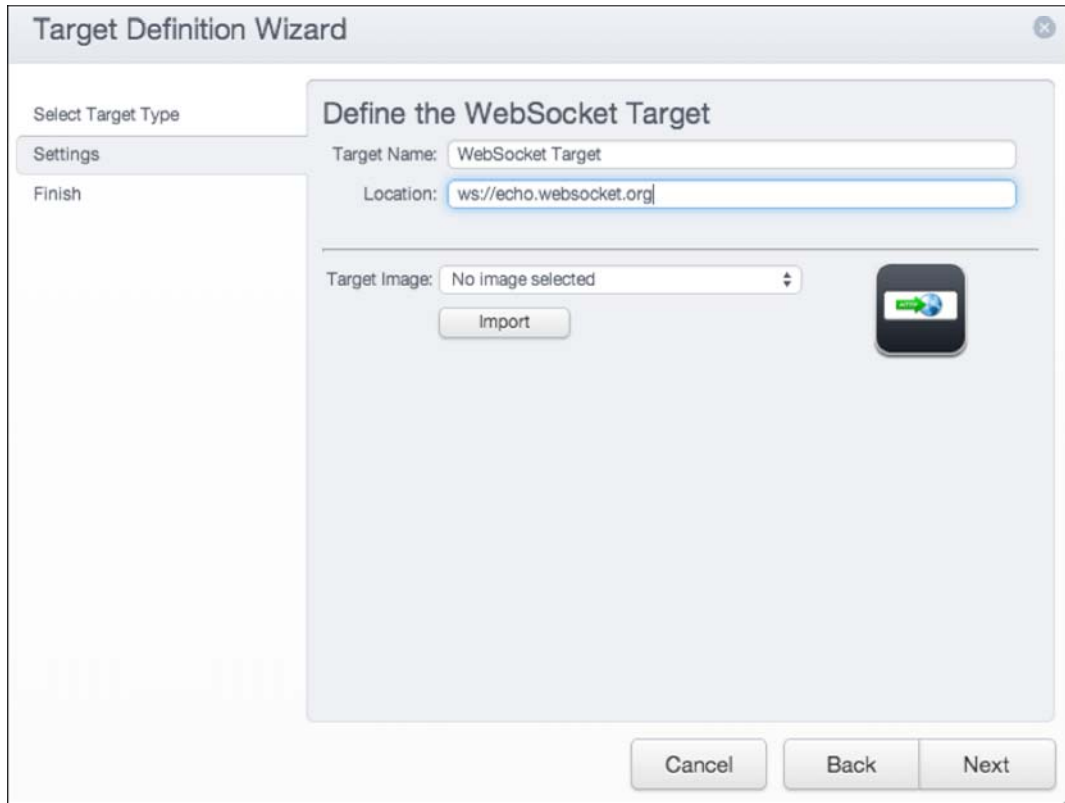
WebSocket Target Type

The new target type appears in the Target Definition Wizard alongside other target types.



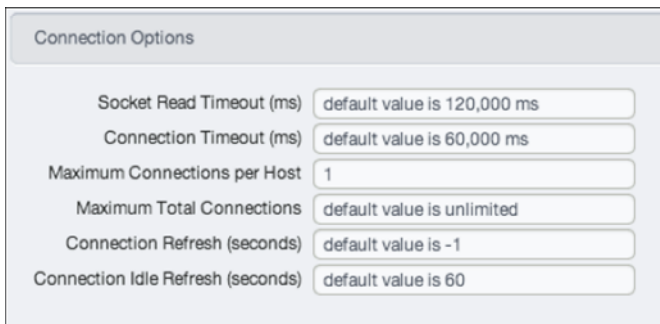
TIP: Since CloudTest will auto-detect WebSocket targets. It is *not necessary* to do manual target creation in most cases.

For manual creation of WebSocket targets, a URL must be provided.



CloudTest provides defaults for the following in WebSocket targets that are the same as those found in HTTP targets:

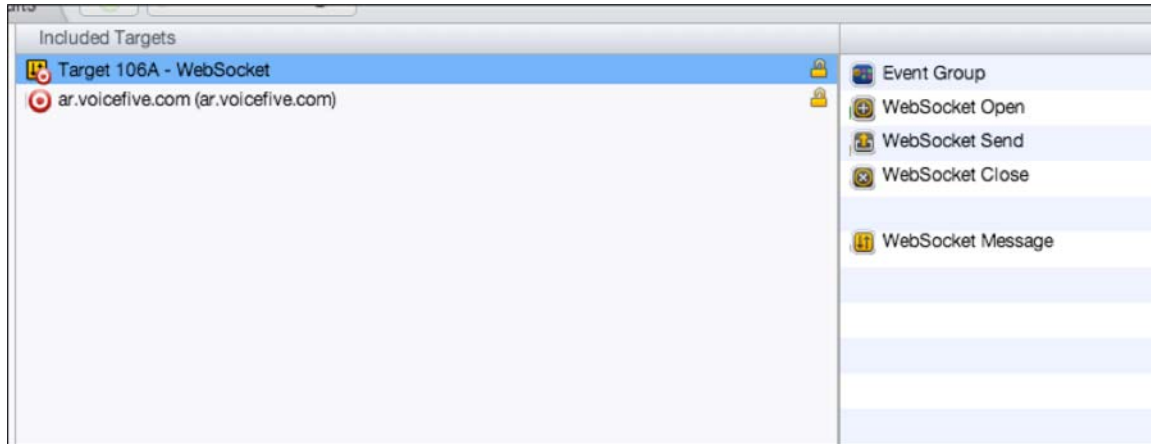
- Socket read timeout
- Connection timeout
- DNS options
- Max received content length



Unlike HTTP targets, WebSocket targets lack the concept of "operations," including no GET, POST, PUT, DELETE, or HEAD, and so, none are displayed. Instead the WebSocket target recognizes WebSocket Open, WebSocket Send, WebSocket Message, and WebSocket Close. Since the conventional HTTP

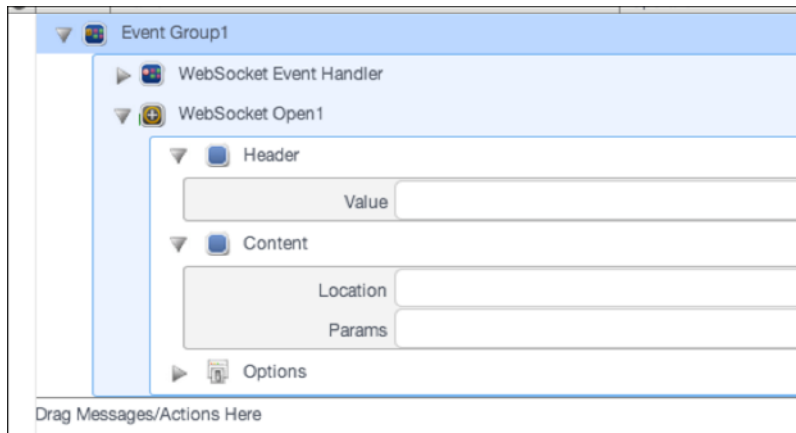
operations are not relevant, the WebSocket target UI is a single panel with no operations tree (as found in HTTP targets).

If a WebSocket target is selected in the Clip Editor lower panel, Messages/Actions tab, the new clip elements, including the actions WebSocket Open, WebSocket Send, WebSocket Message, and WebSocket Close, are shown.



In order to add WebSockets manually, you must first add an Event Group. The Event Group contains a Target Event Handler which contains the event containers. The Event Group does not contain the event containers directly.

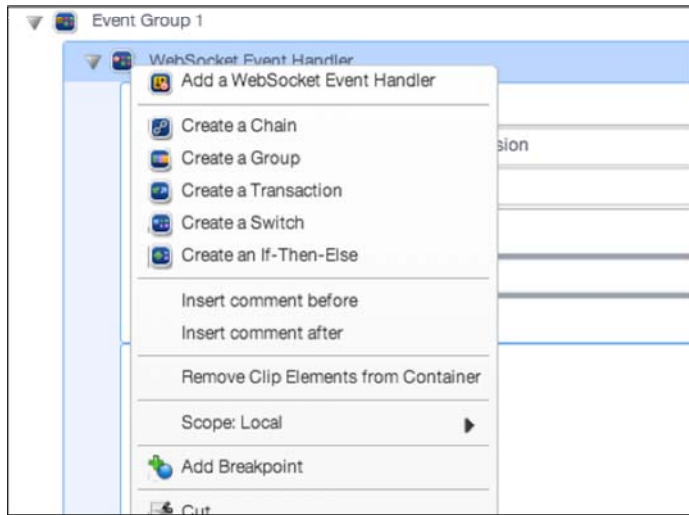
The Clip Editor status indicator enforces this required container model with a user-friendly reminder if a WebSocket action is not dropped into an Event Group.



To add an action, you can either select the WebSocket event to which to add the action, or you can expand the event container (in List View) and drag the action into it.

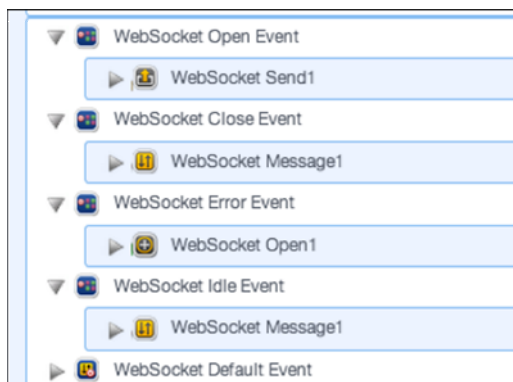
The generic Event Group contains one each of the six event handlers, but additional event handlers can be added via the context menu, Add a WebSocket Event Handler command, whenever the Event Group is selected.

An Event Group can contain any number of WebSocket Event Handlers.



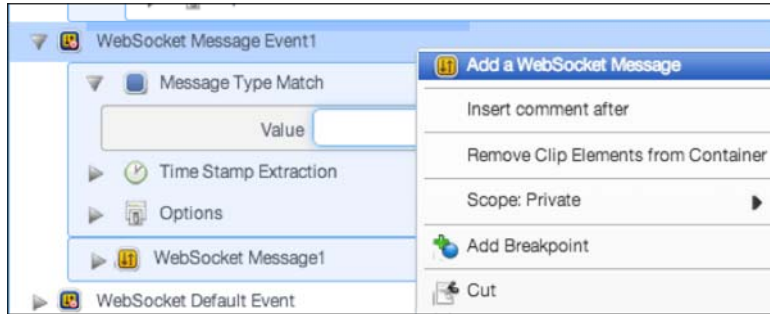
The following restrictions apply when building or modifying an Event Group for WebSocket testing:

- An Event Group cannot be nested inside another Event Group.
- Any WebSocket action must be directly or indirectly inside an Event Group.
- For WebSocket Message, it must reside in one of the six event containers
- For WebSocket Open, it must have a target
- For WebSocket Close, it must have a target.
- For Wait, it must have a wait expression tag.



If the add doesn't adhere to the container restrictions noted above, you will see a Clip Editor, Status Indicator message (top right) showing why it failed.

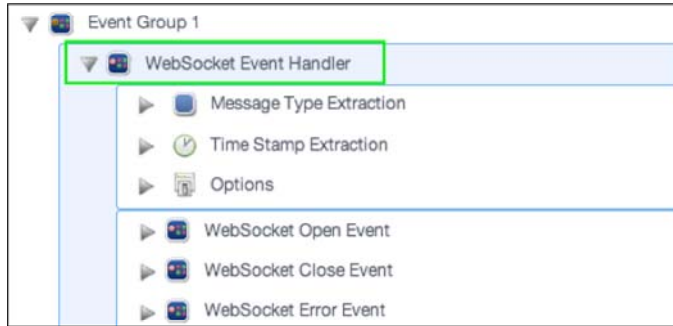
- Right-click an Event Container to add a WebSocket Message to it.



TIP: As noted above, SOASTA recommends recording your WebSocket session. You can then make any additions, deletions, or other changes—including application of Session Templates—once you've captured your test case via recording.

New Event Group and Wait Clip Elements

CloudTest's WebSocket support provides a new target and an Event Group container that includes a target event handler (shown below).



Users will encounter the new Event Group whenever a WebSocket is recorded. The Event Groups children include the new Event Handler, as well as the events that it supports, as well as the new actions that represent the asynchronous communication.

Note: The new Event Group is used for WebSocket targets *only* at this time but is built to support any Target that supports asynchronous events. In SOASTA 53, WebSocket is the first and only Target that supports asynchronous events included in CloudTest at this time.

The Wait clip element is independent of the WebSocket feature and can be used in all forms of testing as of this release.

New Event Group Container

Event Group	<p>An Event Group behaves similarly to a Group, except that at start it associates a Target Event Handler with some Target.</p> <p>Target Event Handlers are “embedded only” in SOASTA 53. Therefore, you don’t “specify” a Target Event Handler at this time, one is automatically embedded for you.</p> <p>The other items in the Event Group, aside from the Target Event Handler, are normal Clip Elements and will play just as in a Group container. Event handling occurs as long as the Event Group is playing. Once it ends, event handling ends, and the WebSocket</p>
-------------	--

	<p>connection is closed (if open).</p> <p>The Event Group contains all WebSocket connection info. WebSocket clip elements are always found in an Event Group. The Clip Editor will alert the user of any attempt to add or move these clip elements to an unsupported location.</p> <p>The Event Group can be put in sequenced or timed clips but not in a Page.</p>
--	--

New Wait Clip Element

This release includes a new Wait clip element that can be used with WebSocket tests, or any other kind of test. The Wait clip element is not expressly a part of the CloudTest WebSocket.

<p>Wait</p>	<p>A Wait pauses current execution until some expression involving Custom Properties evaluates to true or the maximum wait time specified by the user is reached. This clip element is provided so that synchronization between processing in the Target Event Handler and other parts of the Clip may occur. Specify a maximum wait time.</p>
-------------	--

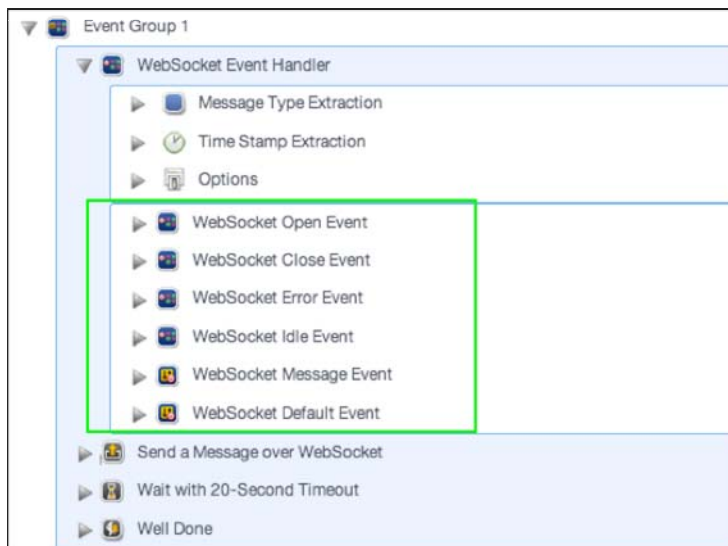
New WebSocket Clip Elements

This release introduces new WebSocket-related clip elements. There are two types:

- Events that can occur in the WebSocket Event Handler
- Actions that can occur in the Event Group.

WebSocket Event Handler - Events

The WebSocket Event Handler contains six new events. Open, Close, Error, Idle, Message, and Default (highlighted below).



WebSocket Actions

The Event Group actions include WebSocket Send, WebSocket Open, WebSocket Message, and WebSocket Close.

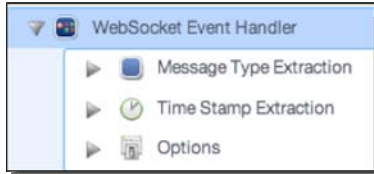
While recording, these clip element types are detected automatically while doing HTTP recording to any site in which WebSocket is detected. Results can be filtered via these clip element types. Lower panel or inline editors are provided in the Clip Editor for operations such as extraction,

WebSocket Event Handler	<p>Target Event Handlers can only contain the event containers, nothing else.</p> <p>This event handler contains the user-defined logic for handling asynchronous events from the WebSocket target.</p> <p>CloudTest doesn't support "nested" Event Groups. An Event Handler may not contain an Event Group or a Target Event Handler.</p>
The following WebSocket clip elements "events" can only be in the Target Event Handler. They cannot be in the Event Group itself:	
WebSocket Open Event	This standard "on open" event fires when the WebSocket connection is opened.
WebSocket Close Event	<p>This standard "on close" event fires when the WebSocket Connection is closed.</p> <p>The connection is closed automatically when the Event Group ends.</p> <p>The connection is opened when a new WebSocket Open clip element is executed.</p> <p>Close is valid only inside of an Event Group or Target Event Handler.</p>
WebSocket Error Event	This event fires if there is a WebSocket error.
WebSocket Idle Event	This "on idle" event fires if there is no Web Socket activity for some amount of time. The timeout length can be set through a System Property.
WebSocket Message Event	The message event fires when a WebSocket Message is received from the server and that message matches the message matching criteria for the event.
WebSocket Default Event	The default event fires when a WebSocket Message is received from the server and that message does not match the message matching criteria for any Message Event.
WebSocket Send	The Send action.

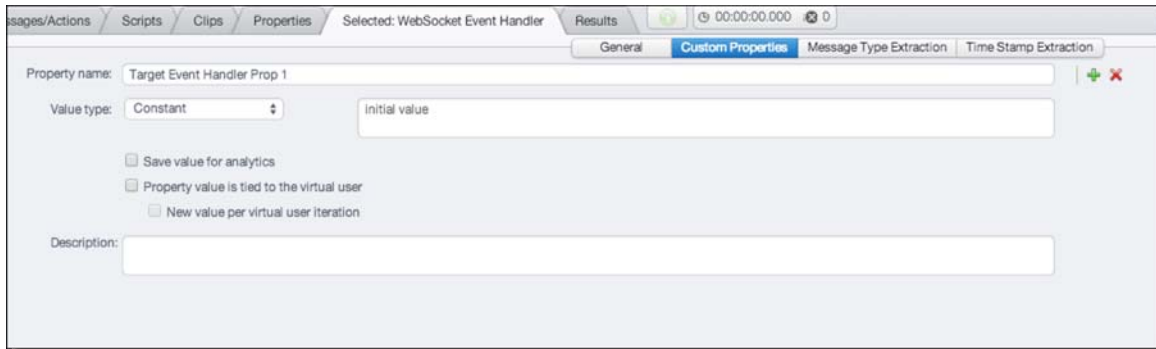
WebSocket Message	<p>The WebSocket Message contains any validations and property sets that should be done for the received message that caused the Message Event or Default Event to fire. Any validations and property sets are performed when the Event Message Clip Element executes as part of the event handling.</p> <p>The message can only repeat zero or one because it's used for handling one event message received.</p>
WebSocket Close	The close action.

WebSocket Event Handler Extraction Methods

The WebSocket Event Handler comes with two built-in extraction methods—one for Message and a second for Time Stamp as well as an Options section. In List View, these are shown inline.

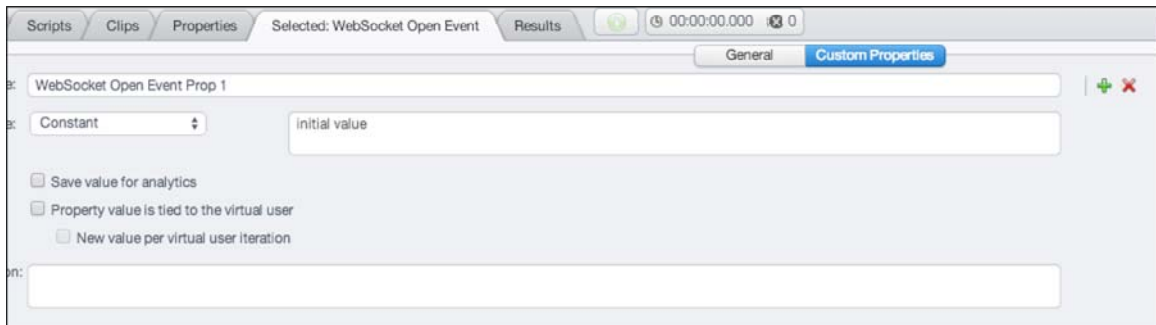


However, the WebSocket Event Handler can also be edited in the Clip Editor lower panel.



In this lower panel, you will also find a Custom Properties panel. As shown above, this Event Handler already has a Custom Property (Target Event Handler Prop 1) with Constant and `initial value` set.

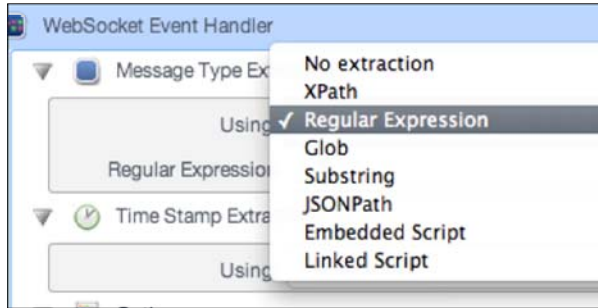
The same Custom Properties that apply to all other clip elements also applies to each of the event containers and other clip elements found in the WebSocket Event Handler. As always, Custom Properties can be set and edited in the lower panel.



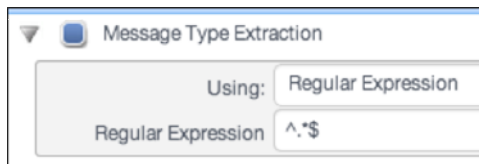
Message Type Extraction in WebSockets

Message Type Extraction is provided in order to extract a value from the received message that can be used to match against the values in the message events to determine which message event should be fired to handle the received message.

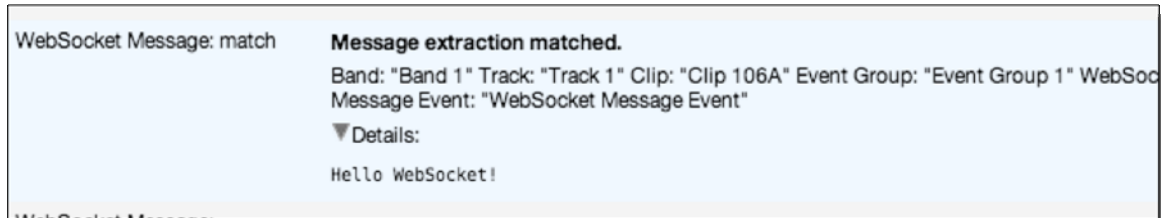
For Message Type Extraction, click Using to select from among the drop-down options. The extraction methods provided for WebSocket Message are similar to existing options provided elsewhere for XPath, JSONPath, Regular Expression, and so forth.



For the method selected, enter an appropriate extraction expression or options, using the conventions of the method itself. For example, using RegEx with the value `^.*s` will match everything in the message.



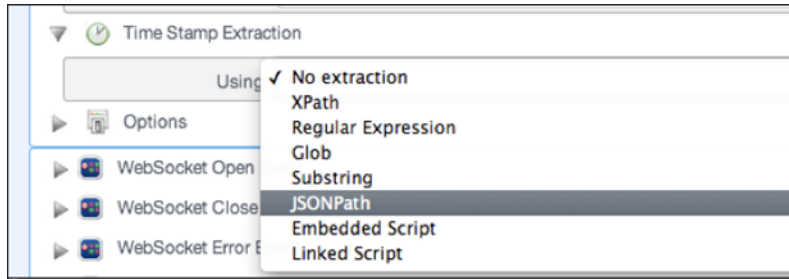
Using the value `^.*s` will result in a whole line match, such and will result in the message content being posted to the Events List.



Time Stamp Extraction in WebSockets

Time Stamp Extraction is provided in order to extract a time, if the server that sent the message inserted one, of the start of the processing for the message, so that we can compute the entire time to process this message, including time spent on the server that sent it. Since we can't know what the server is doing, the only way for us to get the total processing time is if the server sends us the start time as a timestamp. In addition, this assumes that the server's clock is synchronized to (or very close to) our clock.

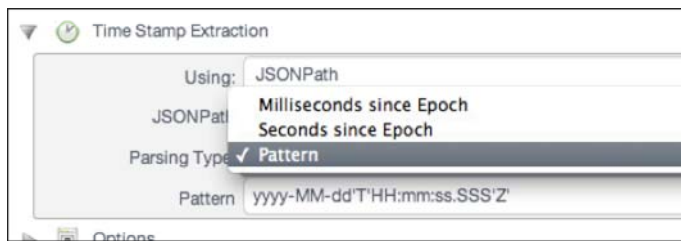
For Time Stamp Extraction, click Using to select from among the drop-down options. The extraction methods provided for WebSocket time stamps are similar to existing options provided elsewhere for XPath, JSONPath, Regular Expression, and so forth.



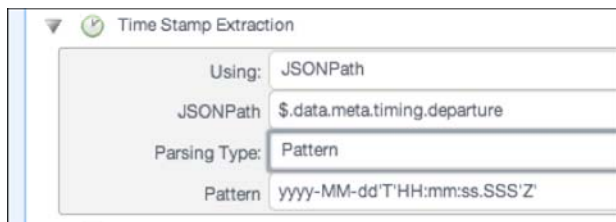
For the method selected, enter an appropriate value to extract, using the conventions of the method itself. For example, using JSONPath with the value `$.data.meta.timing.departure`.

For time stamp, you must also provide the date/time stamp Pattern to use from among Millisecond since Epoch, Seconds since Epoch, or by a specified Pattern (an example of which you should supply).

Time Stamp Extraction can be specified at WebSocket Message Event container(s) and WebSocket Default Event container. Time Stamp Extraction specified at an Event Container will override the Time Stamp Extraction specified at the WebSocket Event Handler.



For example, if you select Pattern and then enter this prescribed format, `yyyy-MM-dd'T'HH:mm:ss.SSS'Z'` (shown below).

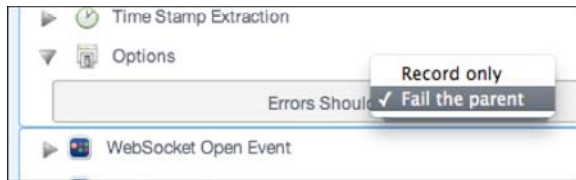


TIP: For more about supported timestamp formats, see the Java documentation for [SimpleDateFormat](#).

Will result in this output in the Result Details, Events List.

WebSocket Event Handler Options

The Event Handler options are failure actions to take (e.g. the same as those found elsewhere in CloudTest).

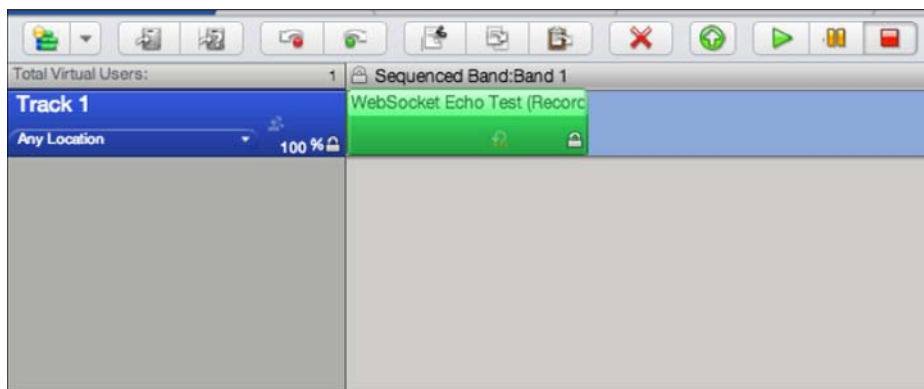


You can choose to Fail the parent on any failure or to Record the failure only.

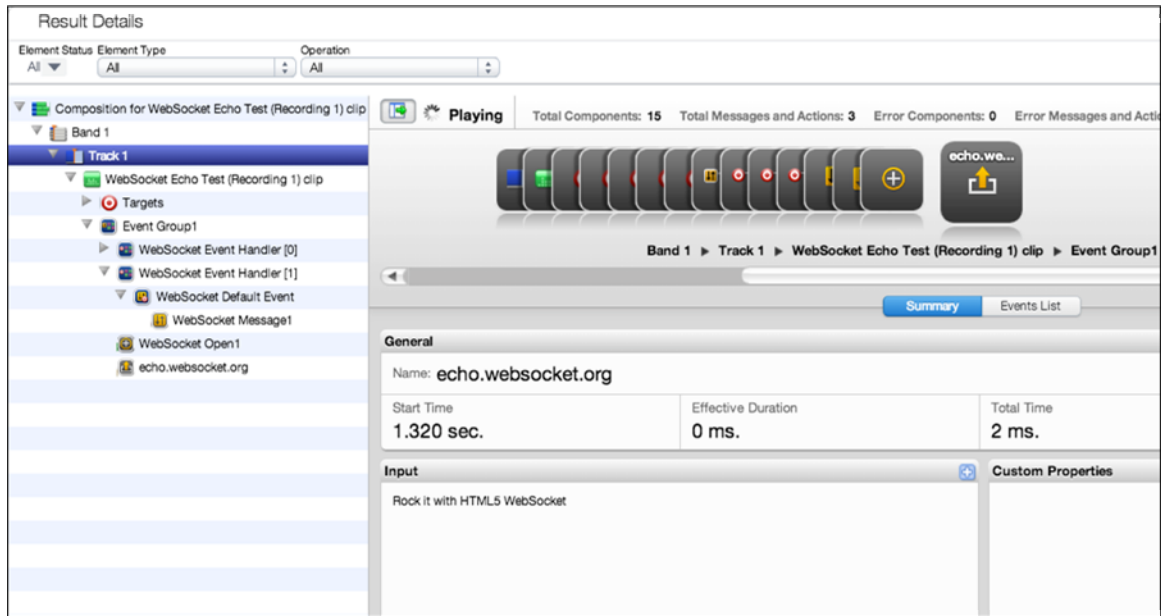
WebSocket Analytics

In this final WebSocket section, let's return to the test clips we created in the Recording section(s) above. In the examples below, we will show results for the filtered test clip based on the recording that we created using the Recording Editor.

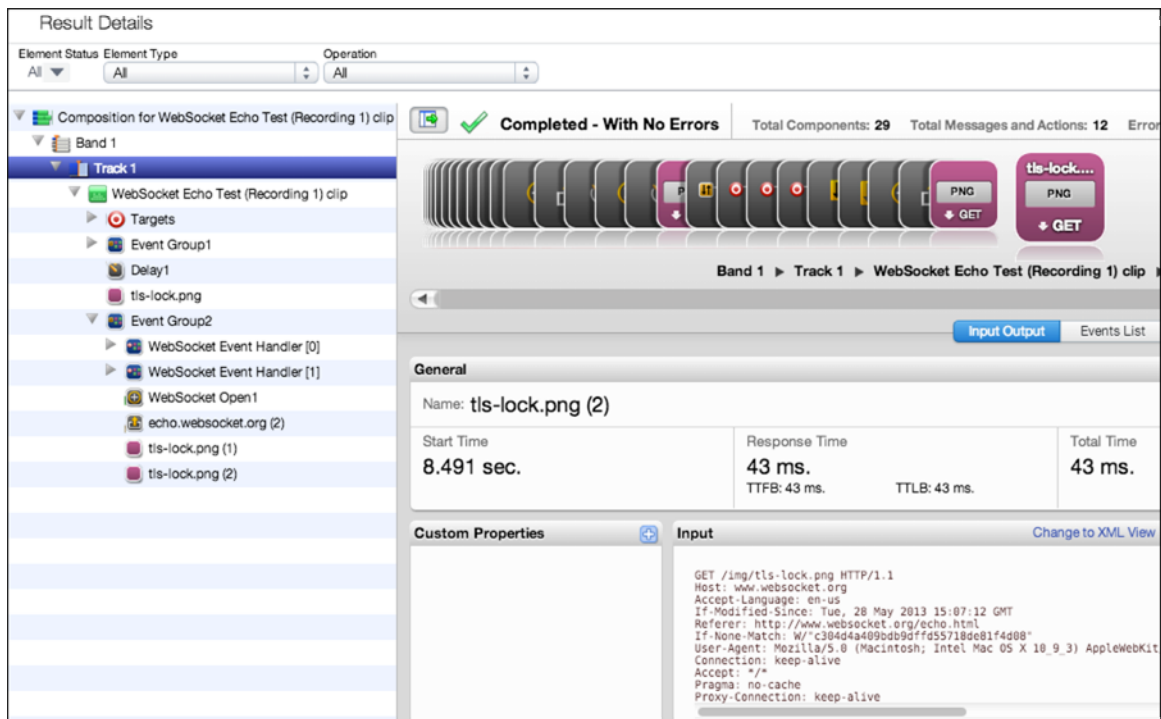
1. Open the test clip to use and select Use in Composition, Open in Test Composition from the Clip Editor toolbar. The test clip is added to a new untitled composition in the Composition Editor.
2. Click Play.



The Composition Editor – Play tab appears and the test plays. The WebSocket objects appear in the stream just as any other clip elements do.

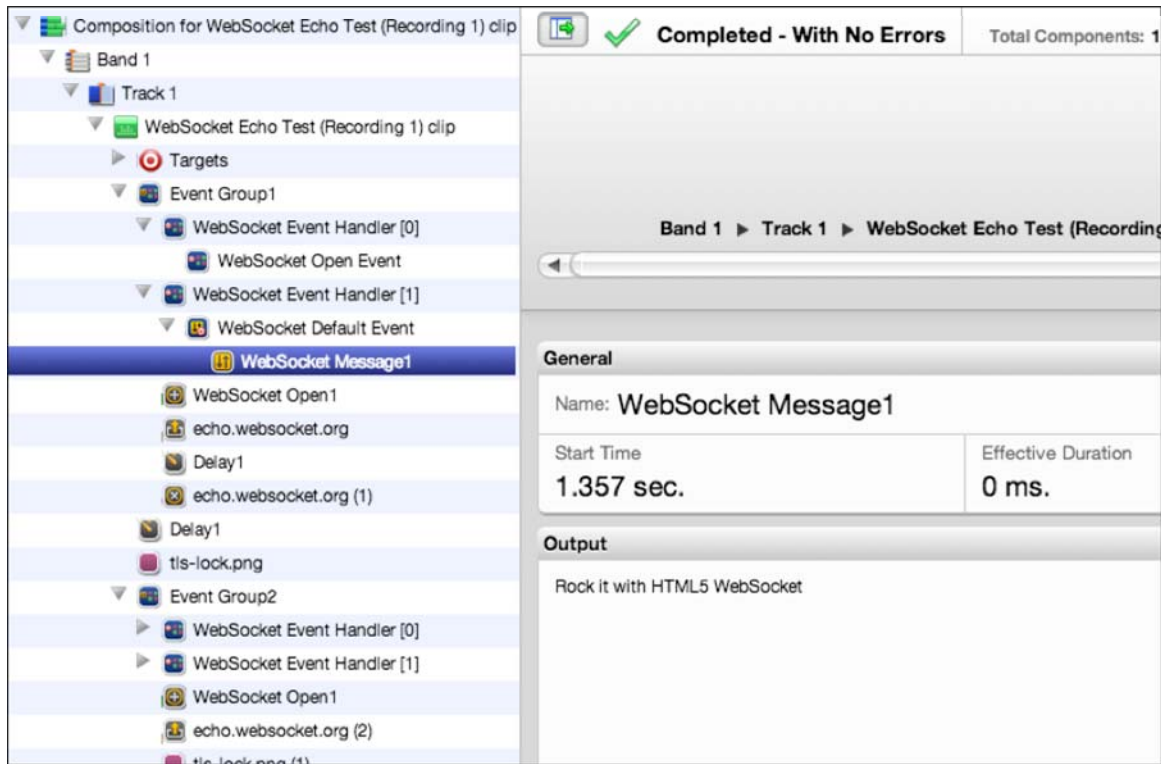


The test composition completes with no errors found.



- Expand the first Event Group, *Event Group1*. Note that the hierarchy reflected in the test clip from recording also appears here.

- Find the WebSocket Send that contained the canned text: *Rock it with HTML5 WebSocket*. In our test clip, it is placed in the WebSocket Event Handler [1], WebSocket Default Event, as *WebSocket Message 1*.

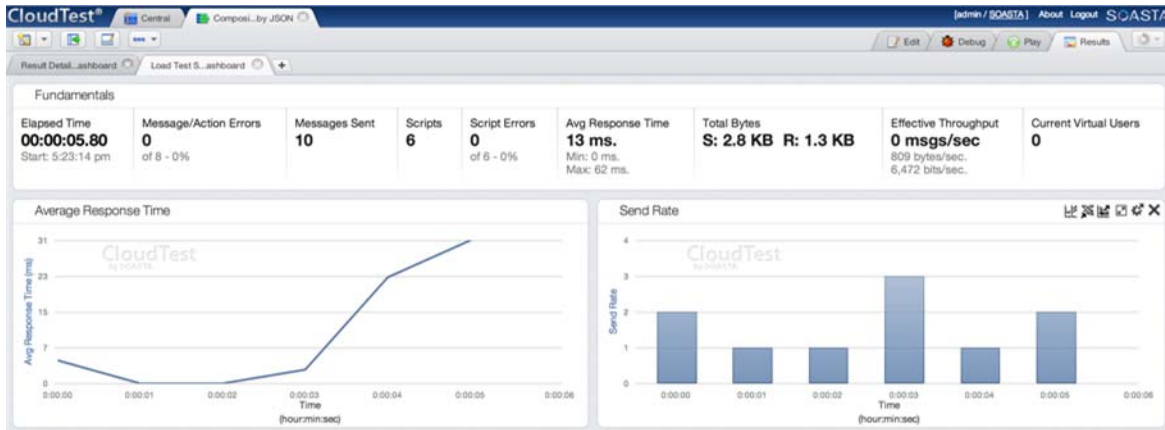


- With *WebSocket Message1* still selected, click the Events List tab and note the Event Code text (4th column from left) as well as the message content.

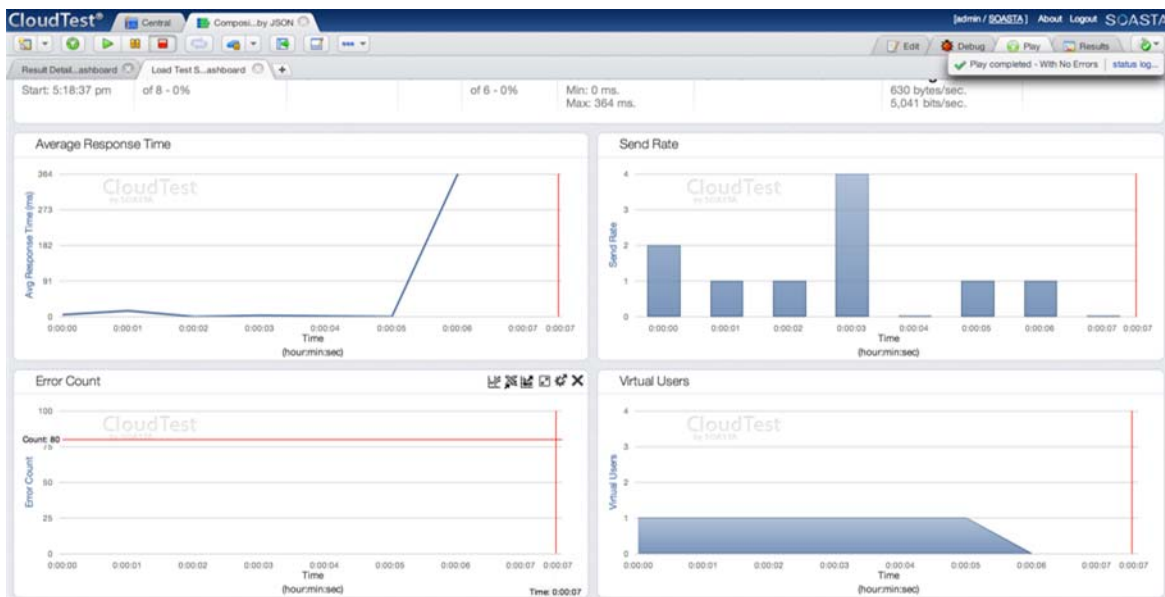
Note: The events in the Event Handler always appear grouped together in the tree underneath the Target Event Handler – they are not intermixed in chronological order with the other actions in the Event Group. There will be one repeat for each instance of an event that is fired.

Event(s)				
Event	Time	Level	Event Code	Description
	39	1357	Info	WebSocket Message: start WebSocket Message started. Band: "Band 1" Track: "Track 1" Clip: "WebSocket Echo Test (Recording 1) clip" Event Group: "WebSocket Event Handler [1]" WebSocket Default Event: "WebSocket Default Event"
	40	1357	Info	WebSocket Message: message WebSocket Message Band: "Band 1" Track: "Track 1" Clip: "WebSocket Echo Test (Recording 1) clip" Event Group: "WebSocket Event Handler [1]" WebSocket Default Event: "WebSocket Default Event" ▶ Details:
	41	1357	Info	WebSocket Message: end WebSocket Message completed, actual duration: 0 ms. Band: "Band 1" Track: "Track 1" Clip: "WebSocket Echo Test (Recording 1) clip" Event Group: "WebSocket Event Handler [1]" WebSocket Default Event: "WebSocket Default Event"
	42	1357	Statistics	WebSocket Message: stats WebSocket Message statistics. Band: "Band 1" Track: "Track 1" Clip: "WebSocket Echo Test (Recording 1) clip" Event Group: "WebSocket Event Handler [1]" WebSocket Default Event: "WebSocket Default Event" ▶ Details:

WebSocket results are also shown in the Load Test Summary Dashboard, including in the Fundamentals widget.



WebSocket results appear in the Load Summary Dashboard just as any message-based results do. Scroll in the dashboard to show the additional Load Summary Dashboard widgets.



Enhancements

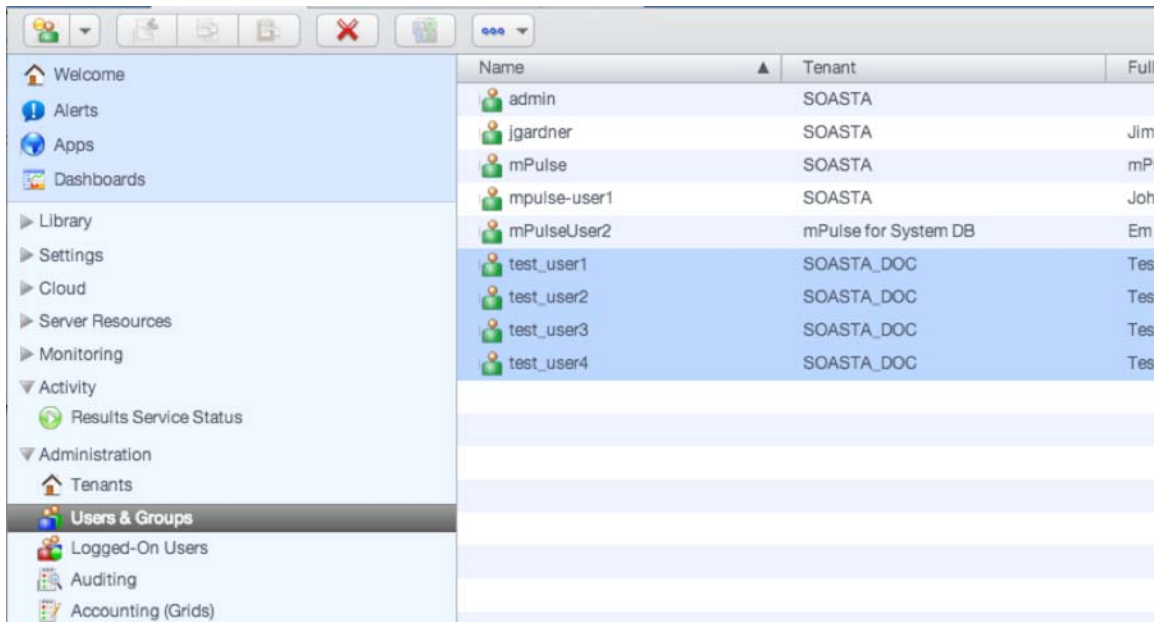
CloudTest

User and Permission Management Enhancements

User Groups

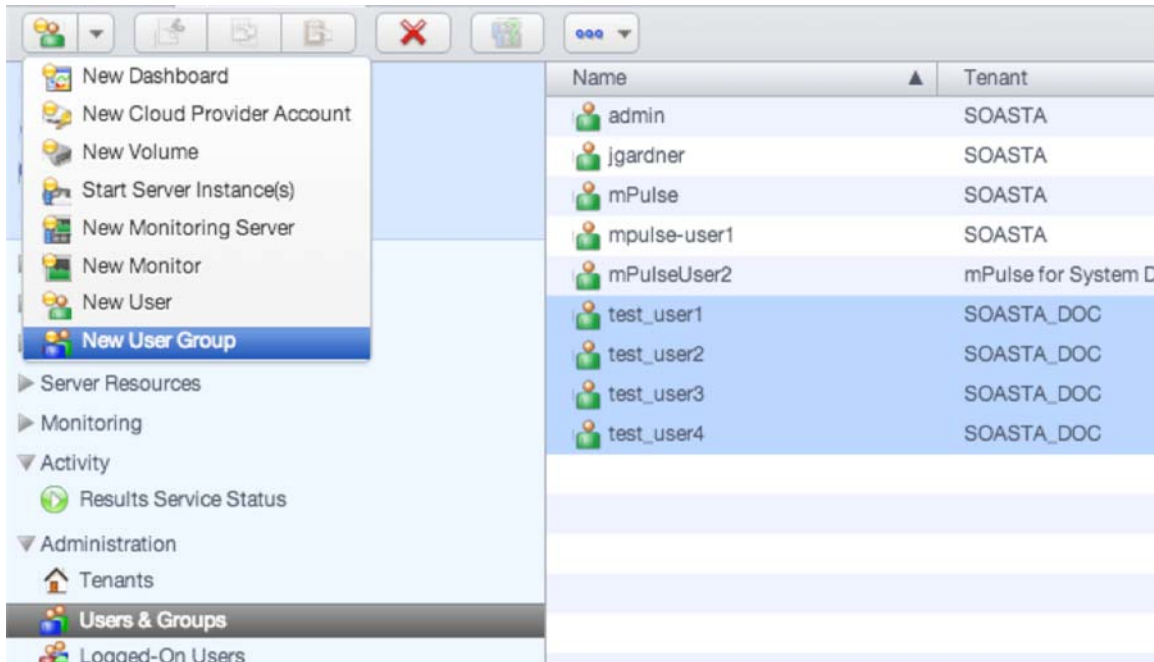
In previous releases, permissions (access control lists) were managed by adding lists of individual users to an ACL. As of SOASTA 53, administrators can create User Groups as well. User Groups are simply collections of users that can be used when building access control lists.

To this end, the Central > Users node has been renamed to Users & Groups. Users are created as they have been in SOASTA since the beginning.



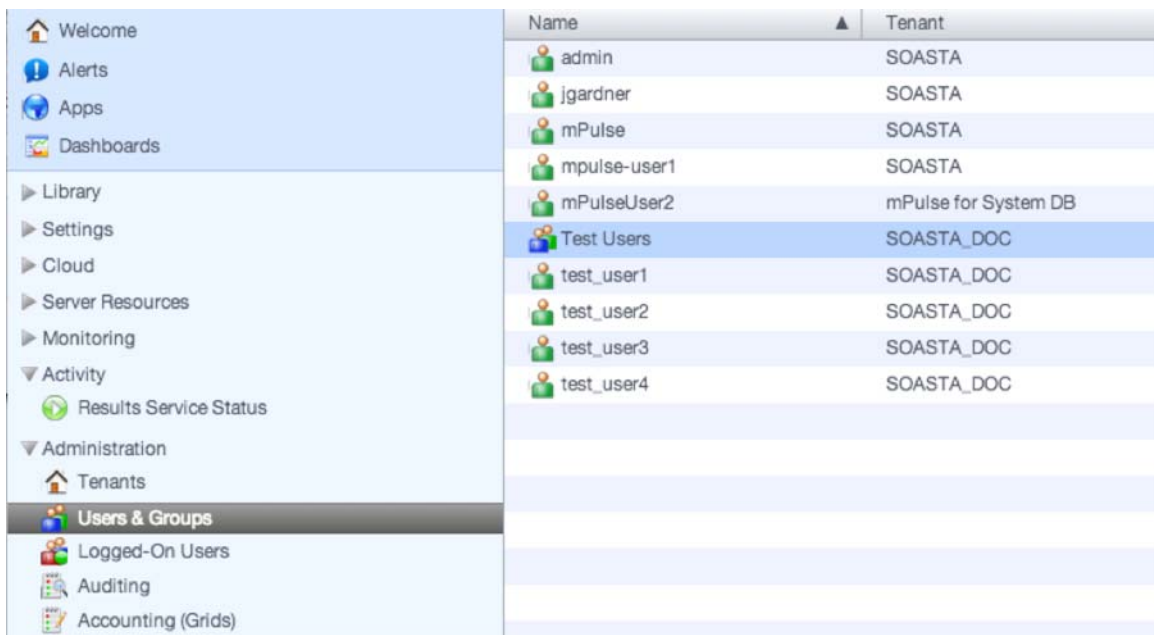
Name	Tenant	Full
admin	SOASTA	
jgardner	SOASTA	Jim
mPulse	SOASTA	mPu
mpulse-user1	SOASTA	Johi
mPulseUser2	mPulse for System DB	Em
test_user1	SOASTA_DOC	Test
test_user2	SOASTA_DOC	Test
test_user3	SOASTA_DOC	Test
test_user4	SOASTA_DOC	Test

1. To create a User Group, select Users & Groups and then choose New User Group from the New drop-down.



2. In the New User Group, give the new User Group a name and then check all the users to include.
3. Click OK to complete the User Group creation.

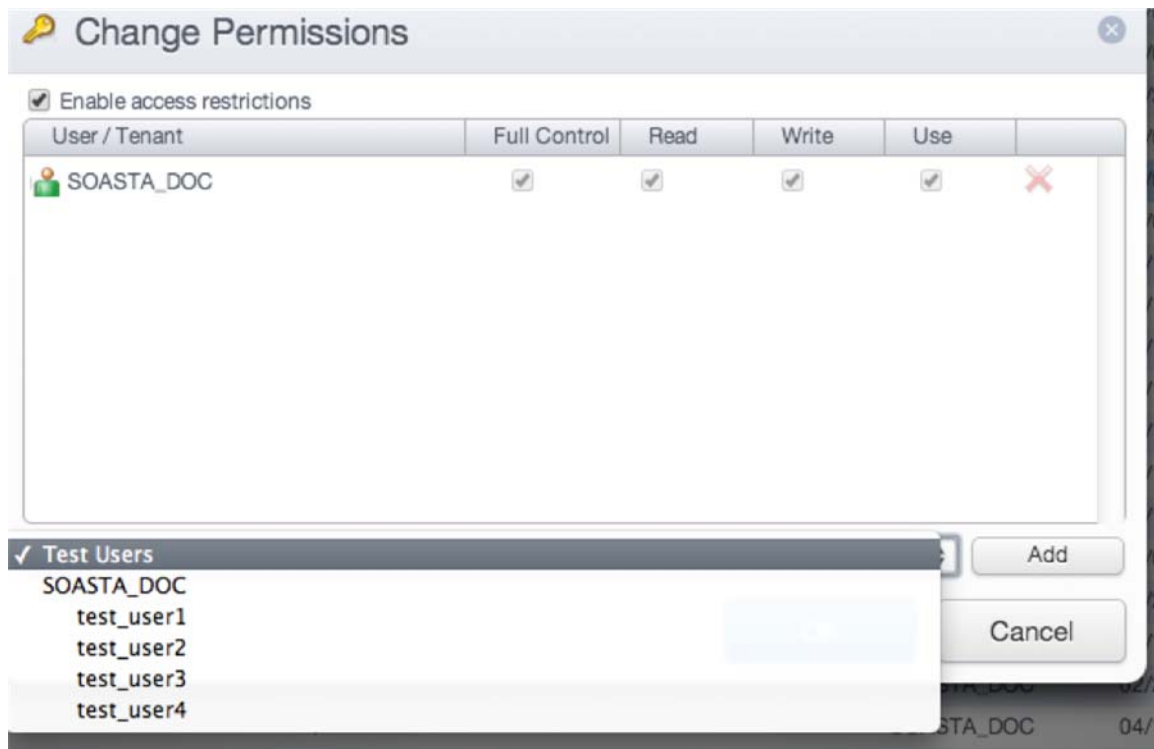
The New User Group appears in the Users & Groups list.



To edit an existing group, double-click it or right-click to open it in the User Group Editor. This box is precisely the same as the New User Group box. Un-check the user(s) to remove from the group.

Users can be included in multiple groups if desired. For example, the user joe@acme.com might be included in a New Hires group, but also in a Performance Engineers group. This allows a high level of precision when building access control lists.

Note that User Groups also appear in the Change Permissions box at the same logical level as User Admins. For example, in the drop-down (shown below) Test Users is a user group, while SOASTA_DOC is a User Admin.



Change or Transfer Ownership of Resource(s)

Ownership of a SOASTA repository object can now be transferred from one to another user by any user with User Admin privileges.

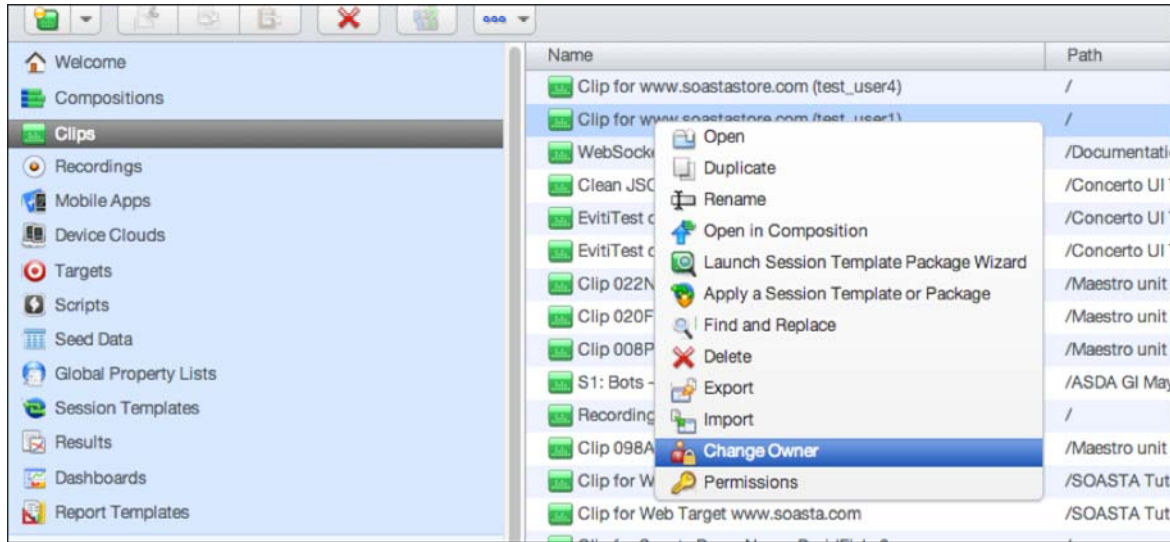
Change Owner allows a User Admin to clear a user's ownership of repository objects by moving ownership to a second user. This is, for example, necessary if that user has left the organization or team.

TIP: The User Admin can also take Full Control of any user object by selecting and choosing the Permissions command. Refer to the Full Control command also included in this release.

In prior releases, lacking this feature, the departed user couldn't be deleted until ownership of objects was removed, thus blocking the re-use of organizational equity.

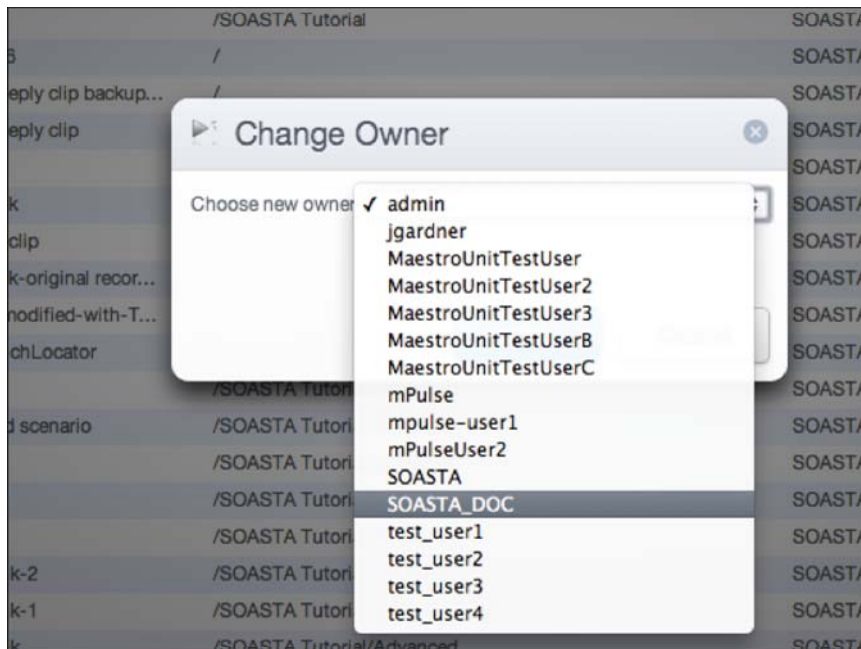
Now, the User Admin can simply transfer ownership of a directory or item in order to have the ability to delete a user, but still retain all of the items they have created for different projects.

To change the owner of one of more items, first select, and then right-click to select the Change Owner command from the menu.



When you do so, the Change Owner box appears.

Use this box to select an alternate user to whom ownership will be transferred.



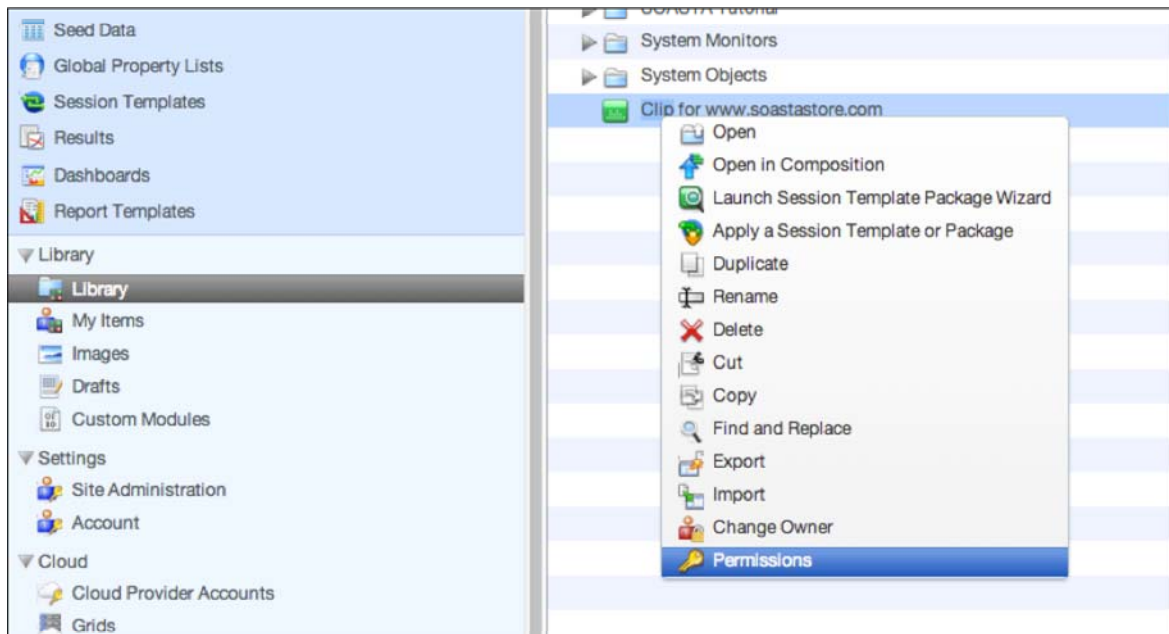
Full Control of Object(s) (User Admin)

This release introduces an enhanced ability for a User Administrator to take full control—using the existing Permissions box—of the underlying permissions formerly restricted to the original object owner. This ability is crucial to team collaboration in test compositions and with respect to other SOASTA repository objects, and is also useful if a team member is absent or leaves the team.

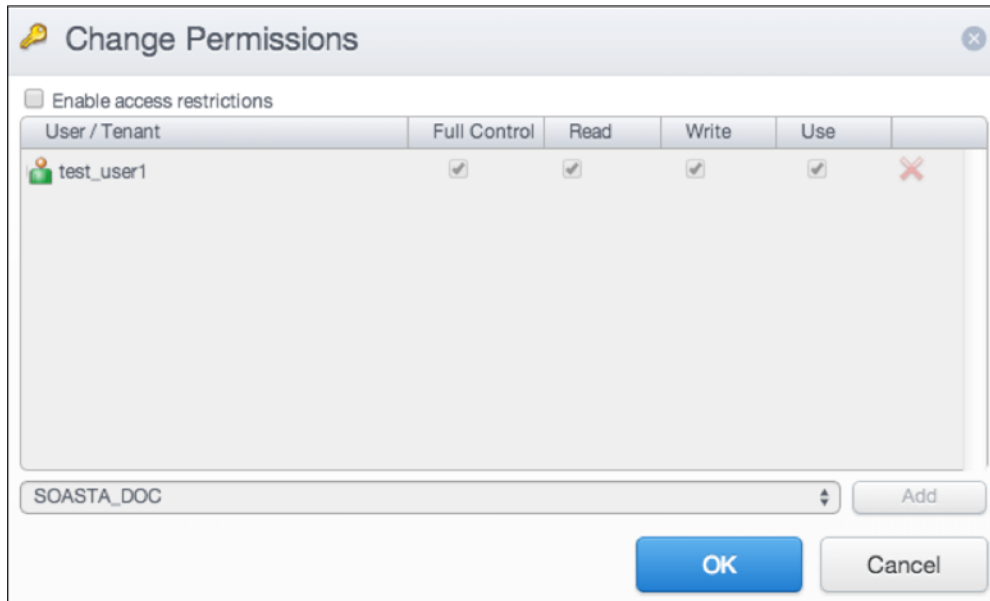
The User Admin can grant such full control per user(s), to herself, or to the Tenant. Any permissions granted to the Tenant applies to all the users in that tenant for the selection. However, the User Admin can supersede any tenant-level restrictions with additional user-level restrictions (or by applying Permissions one at a time per user—in either case, lower granularity means more specific control.

This new feature covers some edge cases. For example, if User A is not the owner of an object, but they have the ability to set permissions, and they apply permissions to User B, then the originating User A (a User Admin in this case) also gets those permissions. CloudTest always applies full-control permissions to User A when they change permissions.

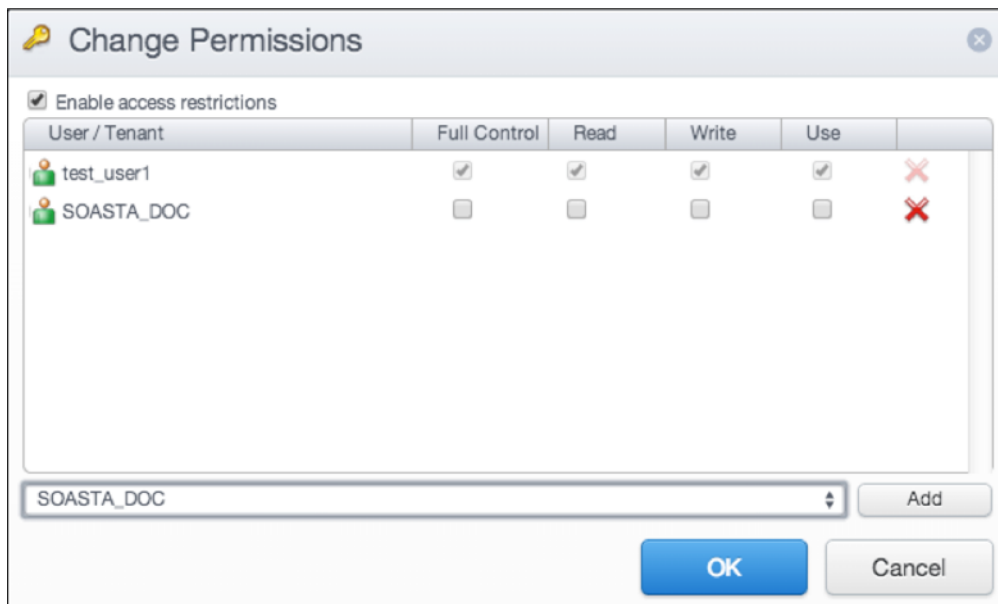
To take control of an object, the User Admin will select the object(s) and then access the Permissions from the right-click menu as in prior releases.



When you do so, the Change Permissions box appears displaying the current Permissions of the selection.

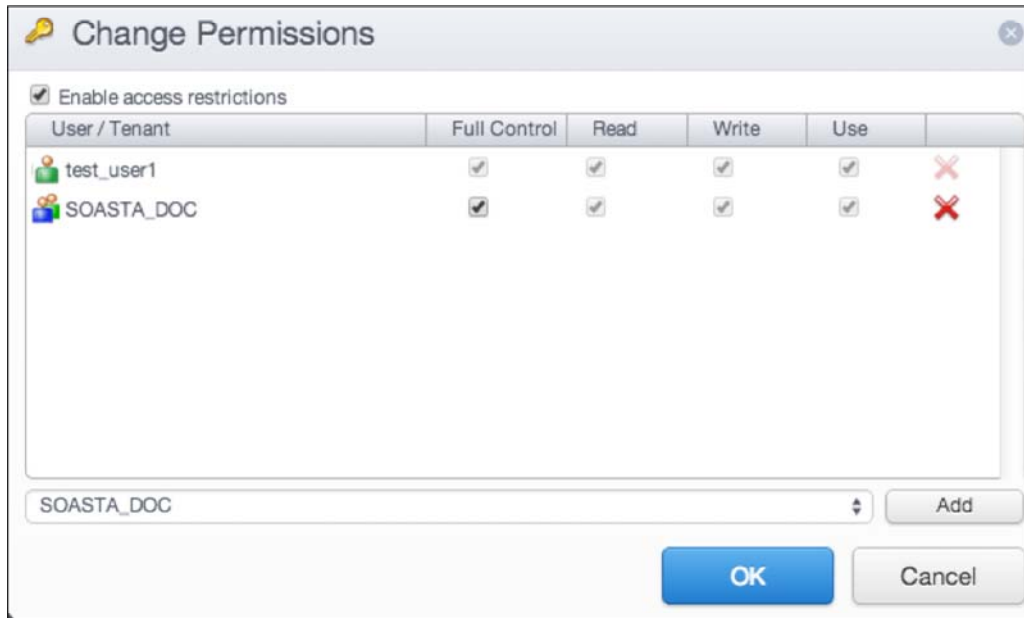


Select the tenant or user to add for whom Full Control (or other permissions) will be added or subtracted and then click Add.

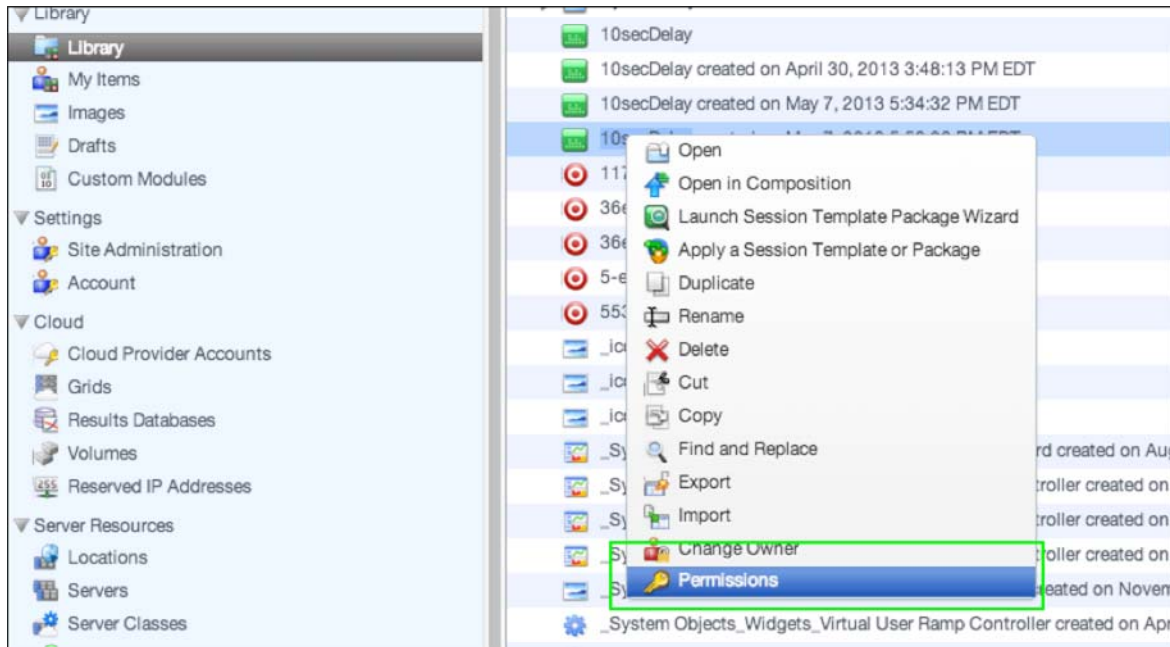


After adding the selected user to the Access Control List in the workspace, check the Full Control box.

When Full Control is checked, the Read, Write, and Use fields are grayed since Full Control includes all of these permissions.

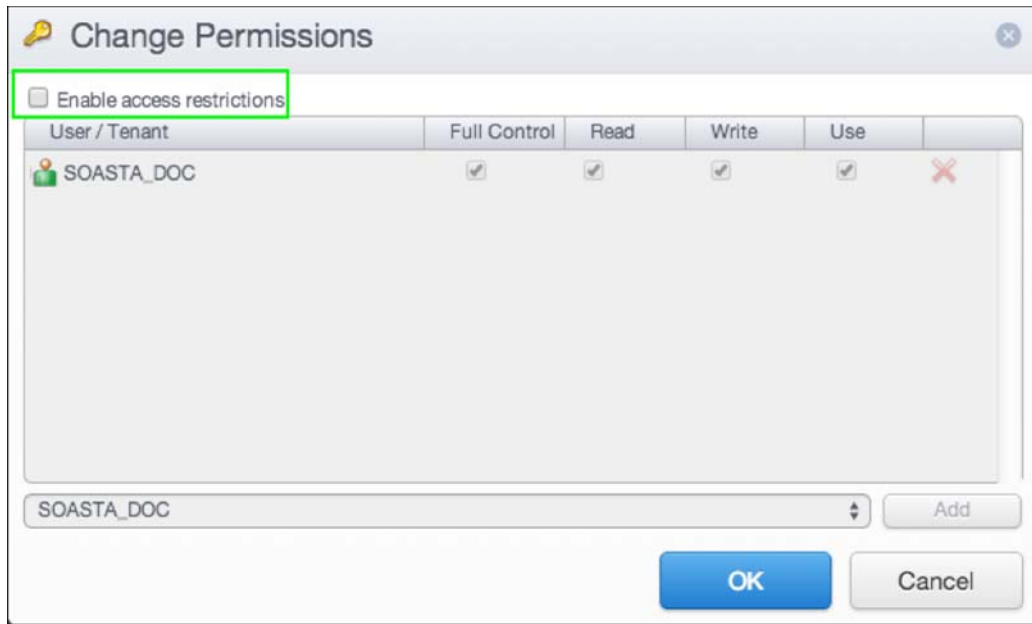


Click OK to complete granting Full Control to the listed user.



Once this command is selected, the Permissions dialog box appears. Access restrictions are disabled by default.

1. Check the Enable access restrictions box to start adding restrictions.



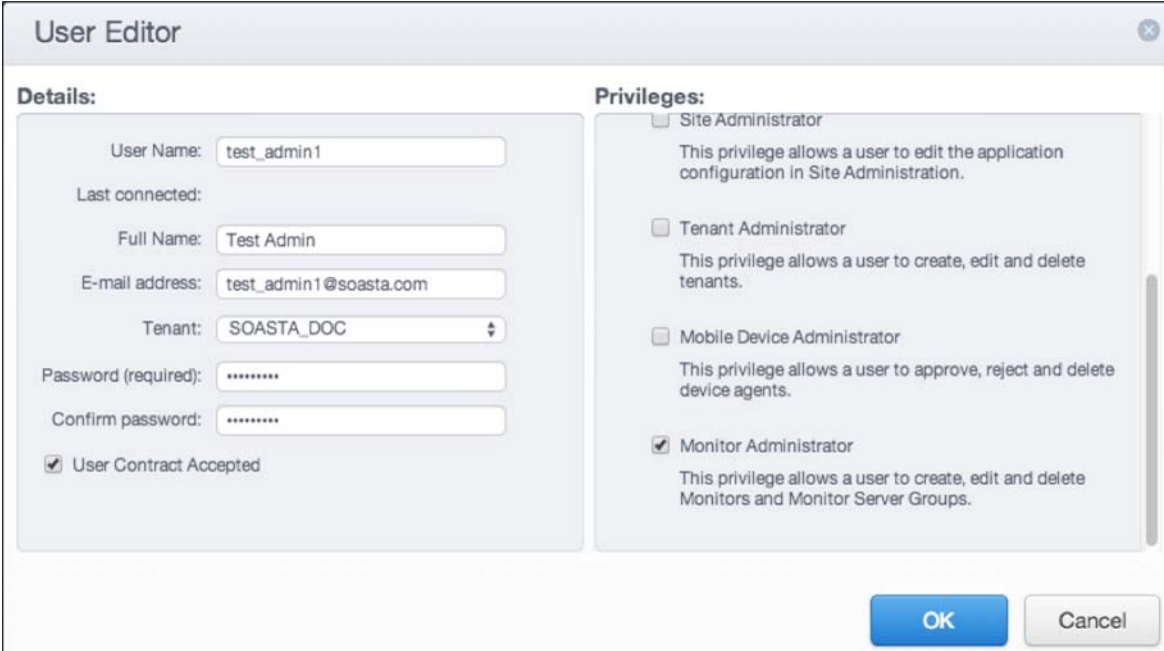
New Monitor Administrator Privilege

This enhancement permits the mPulse User Administrator to disable Monitor and Monitor Server Group creation per user in mPulse production environments.

Since Monitors can place additional stress on the read/write capabilities of the server, and significantly impact performance, Monitor creation is now reserved for those users with this privilege.

As of this release, the New User dialog box contains a new checkbox that the User Administrator can use to assign a new privilege, the Monitor Administrator.

To grant or disable the Monitor Administrator privilege per user, ensure that the box is either checked (enabled) or unchecked (disabled) for the given user.



The screenshot shows a 'User Editor' dialog box with two main sections: 'Details' and 'Privileges'. The 'Details' section contains fields for User Name (test_admin1), Last connected, Full Name (Test Admin), E-mail address (test_admin1@soasta.com), Tenant (SOASTA_DOC), Password (required) (masked with dots), Confirm password (masked with dots), and a checked checkbox for 'User Contract Accepted'. The 'Privileges' section lists four privileges: Site Administrator (unchecked), Tenant Administrator (unchecked), Mobile Device Administrator (unchecked), and Monitor Administrator (checked). Each privilege has a brief description of its capabilities. At the bottom right, there are 'OK' and 'Cancel' buttons.

Section	Field/Item	Value/Status
Details	User Name	test_admin1
	Last connected	
	Full Name	Test Admin
	E-mail address	test_admin1@soasta.com
	Tenant	SOASTA_DOC
	Password (required)	*****
	Confirm password	*****
User Contract Accepted		<input checked="" type="checkbox"/>
Privileges	Site Administrator	<input type="checkbox"/>
	Tenant Administrator	<input type="checkbox"/>
	Mobile Device Administrator	<input type="checkbox"/>
	Monitor Administrator	<input checked="" type="checkbox"/>

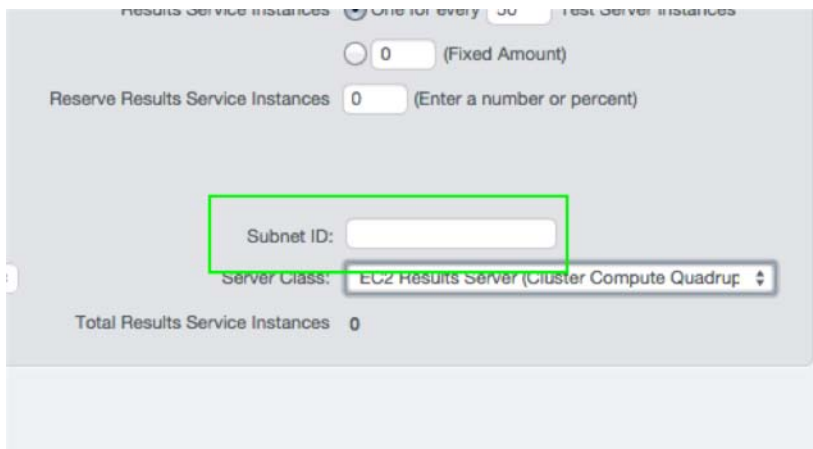
Amazon EC2 Integration Enhancements

The following enhancements provide improved performance for Amazon EC2 users.

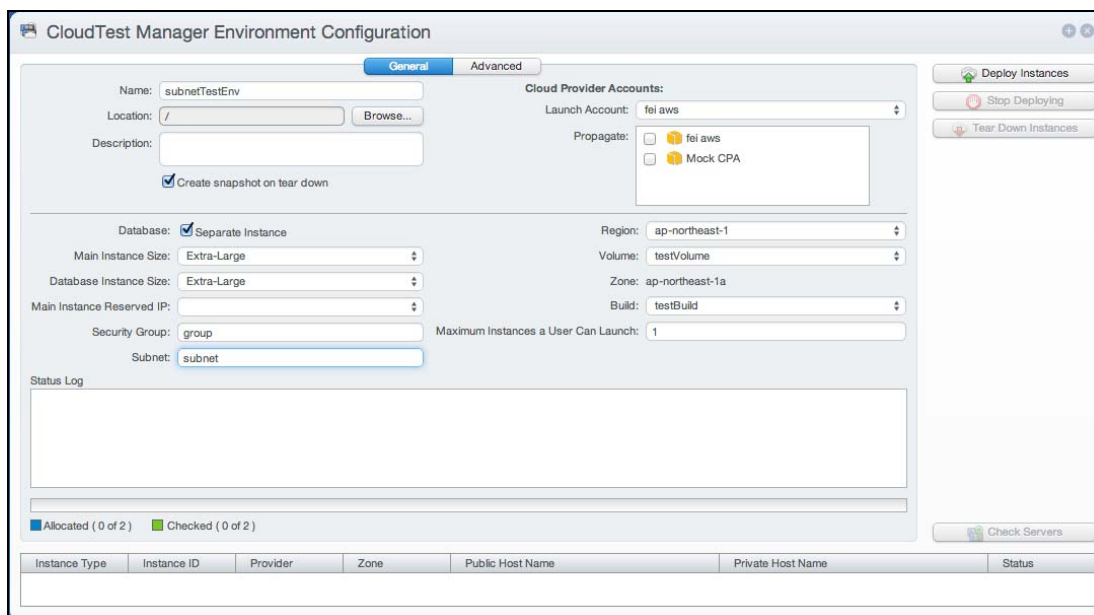
Virtual Private Cloud (VPC) Support

As of this release, CloudTest provides Environment, Grid, and Results Database support for launching Amazon EC2 instances within an EC2 Virtual Private Cloud (VPC) using the new Subnet ID entry field. This field is used to enter the EC2 SubnetID to use for the given VPC.

Users will find this new Subnet ID text input (similar to the input for security group in the Grid, Environment and RSDB editors) in the Environments, Grid, and RSDB UIs.



In the Environment Configuration box, enter the Vendor ID of the private network that you want to use into the Subnet field.



On the last step of the New Reserved IP address creation wizard, check the Private Network Address to use this new feature.

The screenshot shows a window titled "New Reserved IP Address" with a sidebar on the left containing three options: "New Reserved IP Address", "Import Reserved IP Address", and "Create Reserved IP Address". The "Create Reserved IP Address" option is selected. The main area is titled "Create Reserved IP Address" and contains the following fields and controls:

- Name:
- Location:
- Description:
- Region:
- Private Network Address

At the bottom of the window are three buttons: "Cancel", "Back", and "Finish".

New Virtualization Type

CloudTest now uses Hardware Virtual Machine (HVM) instead of Paravirtual (PV) across the board for Amazon Machine Images (AMI). This change applies to Grids, Results Databases, and Environments. HVM is the new generation of virtualization technology offered by Amazon. More detail about HVM technology can be found in the Amazon EC user guide's [Virtualization Types](#) article.

New Instance Types

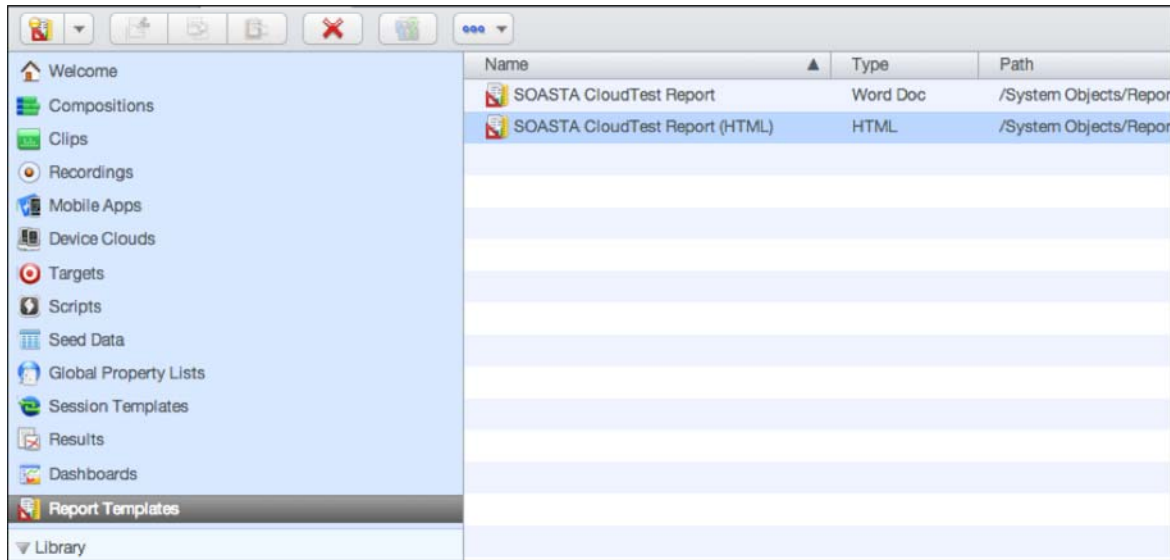
CloudTest now uses current-generation instance types across the board. The translation of CloudTest server sizes to EC2 instance types is as follows:

SOASTA Instance Type	Amazon EC2 Instance Type
<i>Large</i>	SOASTA 52 and earlier: m1.large
	SOASTA 53: m3.large
<i>Extra-Large</i>	SOASTA 52 and earlier: m1.xlarge
	SOASTA 53: m3.xlarge
<i>High-CPU Extra Large</i>	SOASTA 52 and earlier: c1.xlarge
	SOASTA 53: c3.xlarge
<i>High-CPU Double Extra Large</i>	SOASTA 52 and earlier: N/A
	SOASTA 53: c3.2xlarge
<i>High-CPU Quadruple Extra Large</i>	SOASTA 52 and earlier: N/A
	SOASTA 53: c3.4xlarge
<i>High-CPU Eight Extra Large</i>	SOASTA 52 and earlier: N/A
	SOASTA 53: c3.8xlarge
<i>High-Memory Extra Large</i>	SOASTA 52 and earlier: m2.xlarge
	SOASTA 53: r3.xlarge
<i>High-Memory Double Extra Large</i>	SOASTA 52 and earlier: m2.2xlarge
	SOASTA 53: r3.2xlarge
<i>High-Memory Quadruple Extra Large</i>	SOASTA 52 and earlier: m2.4xlarge
	SOASTA 53: r3.4xlarge
<i>High-Memory Eight Extra Large</i>	SOASTA 52 and earlier: N/A
	SOASTA 53: r3.8xlarge

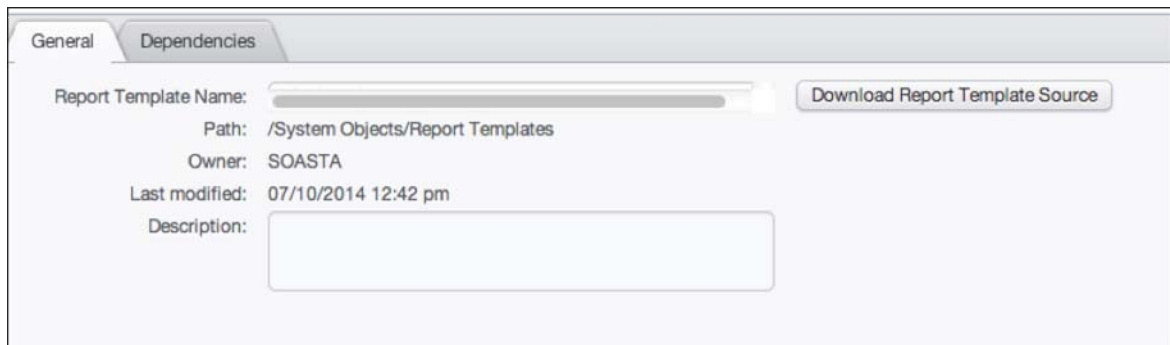
Support for HTML Reports via Template

This release adds HTML as a report generation document type, in addition to the long supported Word Doc format

HTML Reports are based on a default template that is now found in Central > Report Templates.



This template file can be downloaded via the lower panel.



Before revising, take a moment to review the template file as is in your browser.

SOASTA

CloudTest Summary Report

{%% general-text : customer name %%} {%% general-text : test type %%} Test

Test Date: {%% general-text : test date %%}

Performance Test Engineer:
{%% general-text : creator name %%}

Executive Summary

Introduction
description of the type of test being run, the participants in the test, the profile of the company, and the target of the test

Objectives/Goals
description of the purpose and goals for this test as well as a high level of the number of tests run and ...

- Objective 1
- Objective 2

Test Scenarios
X test scenarios were created:

- Test 1 (X% of load)
- Test 2 (X% of load)
- ...

additional information about scenarios, if appropriate

Conclusions
The site handles load gracefully in that the system does not become unstable or crash, and load time goes up relative to the amount of load being applied.

Next Steps and Recommendations
potential actions to take as a result of analyzing the tests, as well as recommendations for future test

- Recommendation 1
- Recommendation 2

Results
What we Learned
Recommendations

For revision, open the HTML report generation template in a source code editor.

```

<div class="summary_block_header">
  CloudTest Summary Report
</div>

<div class="customer_details_div">
  {%% general-text : customer name %%} {%% general-text : test type %%} Test
</div>

<p style="font-size:18px;">
  Test Date: {%% general-text : test date %%}
</p>

<p style="font-size:18px;">
  Performance Test Engineer: <br/>
  {%% general-text : creator name %%}
</p>

<div class="header1">Executive Summary</div>
<div class="header2">Introduction</div>
<div class="resultsblock">
  <div class="header1">Results</div>
  <div class="header2">What we Learned</div>
  <ul>
    <li>&nbsp;</li>
  </ul>

  <div class="header2">Recommendations</div>
  <ul>
    <li>&nbsp;</li>
  </ul>
</div>
<span class="replacegreen">description of the type of test being run, the participants in the test, the profile of the

<div class="header2">Objectives/Goals</div>
<span class="replacegreen">
  description of the purpose and goals for this test as well as a high level of the number of tests run and ...
  <ul>
    <li>Objective 1</li>
    <li>Objective 2</li>
  </ul>
</span>

```

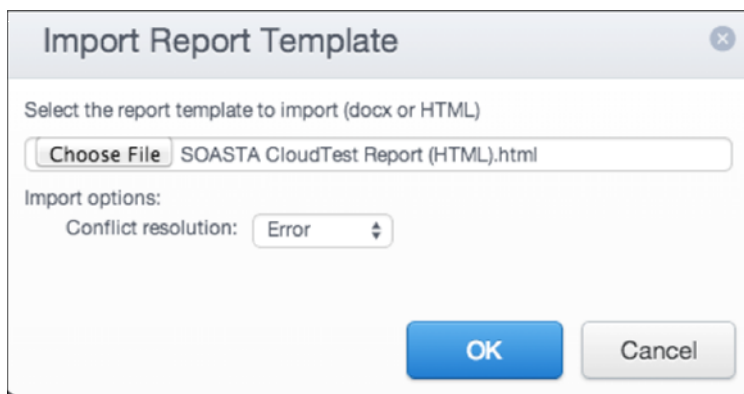
The HTML version of CloudTest's template language is the same as documented in the [Report Template Reference](#)—while using HTML tags instead of the curly brackets used with Word—with only one exception:

The "toc-targets" attribute of the "begin-for-each-result" isn't applicable. It can be there, but will just be ignored. That attribute is used to update the Word TOC, which isn't applicable to HTML templates.

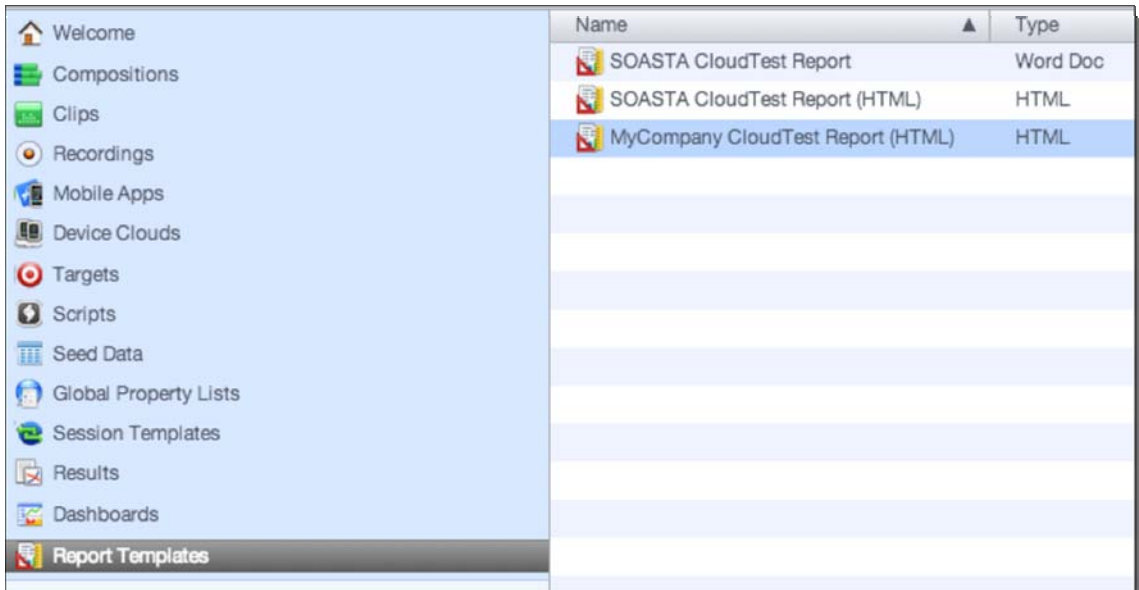
Importing an HTML Report Template

1. To import an HTML report template, select Central > Report Templates, and then click the "New" button and provide the HTML file.

The Import Report Template box appears. Select your revised template.

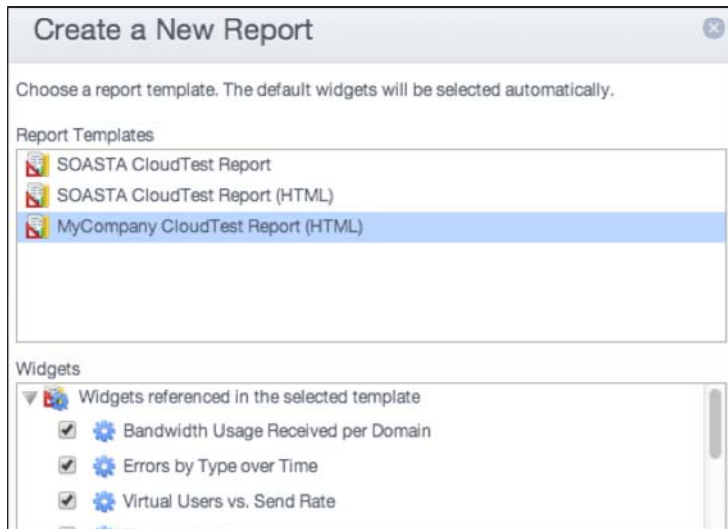


The new report template appears in the flat list along with the two default templates.



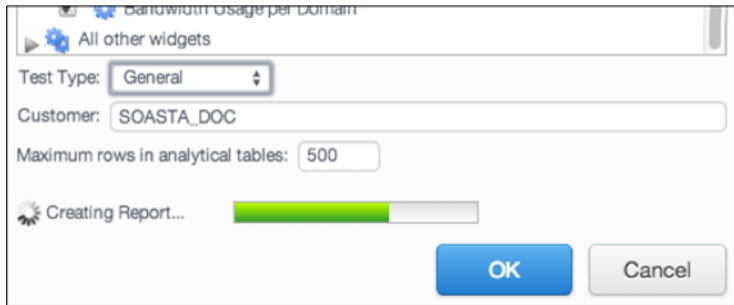
Creating an HTML Report

1. To create a report based upon your custom template, select it instead of either the default template(s). Selecting an HTML type report—whether the default HTML report or a custom one—while doing report generation, will create an HTML-format report.



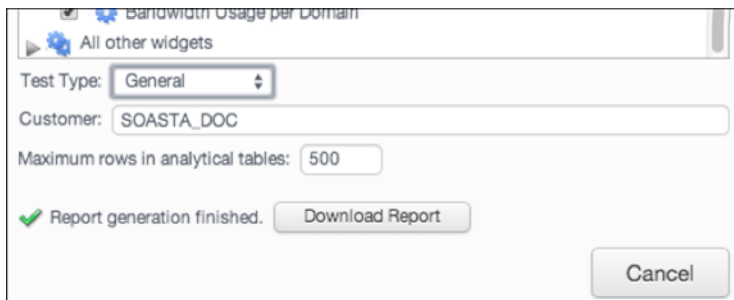
2. Specify the charts to export and the type of test report (e.g. Load, General, or Custom) just as you would for a Word-based report.

3. Click OK to begin report generation. After clicking OK, a preview of your report launches in your browser.



Subsequently, the Download report button appears.

4. Click to initiate the download.



The ZIP file download appears. This archive includes the report ("index.html") as well as an "images" folder for the included images.

Show servers across tenants to Tenant Administrators (75418)

In the prior releases (and going back many releases), the Tenant Admin could see servers across all Tenants in the Servers list, but not in the Player Status views and/or Results Service Status view.

Now, logged in Tenant Administrators will see ALL servers across all tenants for the given CloudTest Instance regardless of which list is in display.

Servers Are Restricted to Tenant (34800)

This release includes the ability to create a Server whose use is restricted to objects created by that Tenant.

To use this capability, select Central > Servers, and then click New.

When you do so, the New Server dialog box appears.

New Server

Location: generic location

Name: test server

Description:

Restrict to Current Tenant

Host Name:

IP Address:

Enabled

Verified

Maestro:

Repository:

Results:

Coordinator:

Monitor:

ExternalData:

Collector:

Reporting:

[Specify additional URLs accessible outside of the location](#)

OK Cancel

This setting can also be accomplished via the selected server's lower panel, Service tab.

The default behavior for new servers and for previously existing server is not to add this restriction.

General Services **Settings** Dependencies Check Server(s) Orphaned Server(s)

Enabled Verified Restrict to Current Tenant

<input type="checkbox"/> Maestro	<input type="text"/>		<input type="checkbox"/> Maestro
<input type="checkbox"/> Repository	<input type="text"/>		<input type="checkbox"/> Repository
<input type="checkbox"/> Results	<input type="text"/>		<input type="checkbox"/> Results
<input type="checkbox"/> Coordinator	<input type="text"/>		<input type="checkbox"/> Coordinator
<input type="checkbox"/> Monitor	<input type="text"/>		<input type="checkbox"/> Monitor
<input type="checkbox"/> ExternalData	<input type="text"/>		<input type="checkbox"/> ExternalData
<input type="checkbox"/> Collector	<input type="text"/>		<input type="checkbox"/> Collector
<input type="checkbox"/> Reporting	<input type="text"/>		<input type="checkbox"/> Reporting

Support for On-Premises Results Databases (34799)

This behind-the-firewall enhancement allows users to create and connect to Results Service Databases setup within a firewall.

1. To do so, first create a Location via Central > Locations using the CloudType, None.

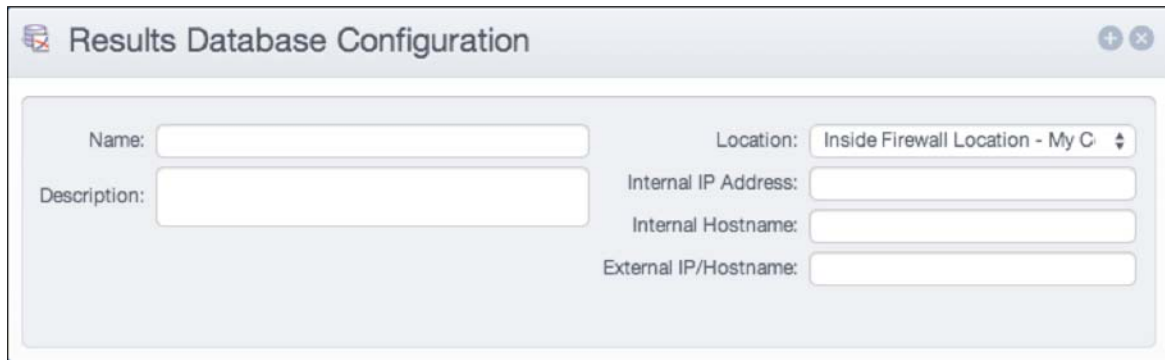
Note: If server locations for CloudTest components are in locations other than the standard location, specify them here. Otherwise, leave the "key" fields blank.

2. Click OK to complete adding the new location.
3. Return to Central, select Results Database (in the Cloud section of the tree), and then click New.

The Results Database Configuration box appears

4. In the Location field (top right), select the new Location you created in the steps above.

When you do so, the Results Database Configuration fields change (shown below) to display only those fields necessary for use with an inside-the-firewall Results Database.



The screenshot shows a window titled "Results Database Configuration". It contains the following fields:

- Name:
- Description:
- Location:
- Internal IP Address:
- Internal Hostname:
- External IP/Hostname:

5. Give the new Results Database a name and, optionally, provide a description.

TIP: If you'd like to return to the expanded UI, simply select one of the cloud locations.

6. Specify all of the following:
 - *Internal IP Address* – The internal IP address of the server machine on which to create the new Results Database.
 - *Internal Hostname* – The internal hostname that corresponds to the specified IP Address.
 - *External Hostname* – Specify an external hostname if this Results Database needs to communicate with some other component that may lie outside the firewall.
7. Click Save on the Results Database toolbar to create this Results Database.

Find and Replace now matches HTTP Header names

While searching HTTP Recordings, Find and Replace will now match HTTP Header Names and their values.

Analytics

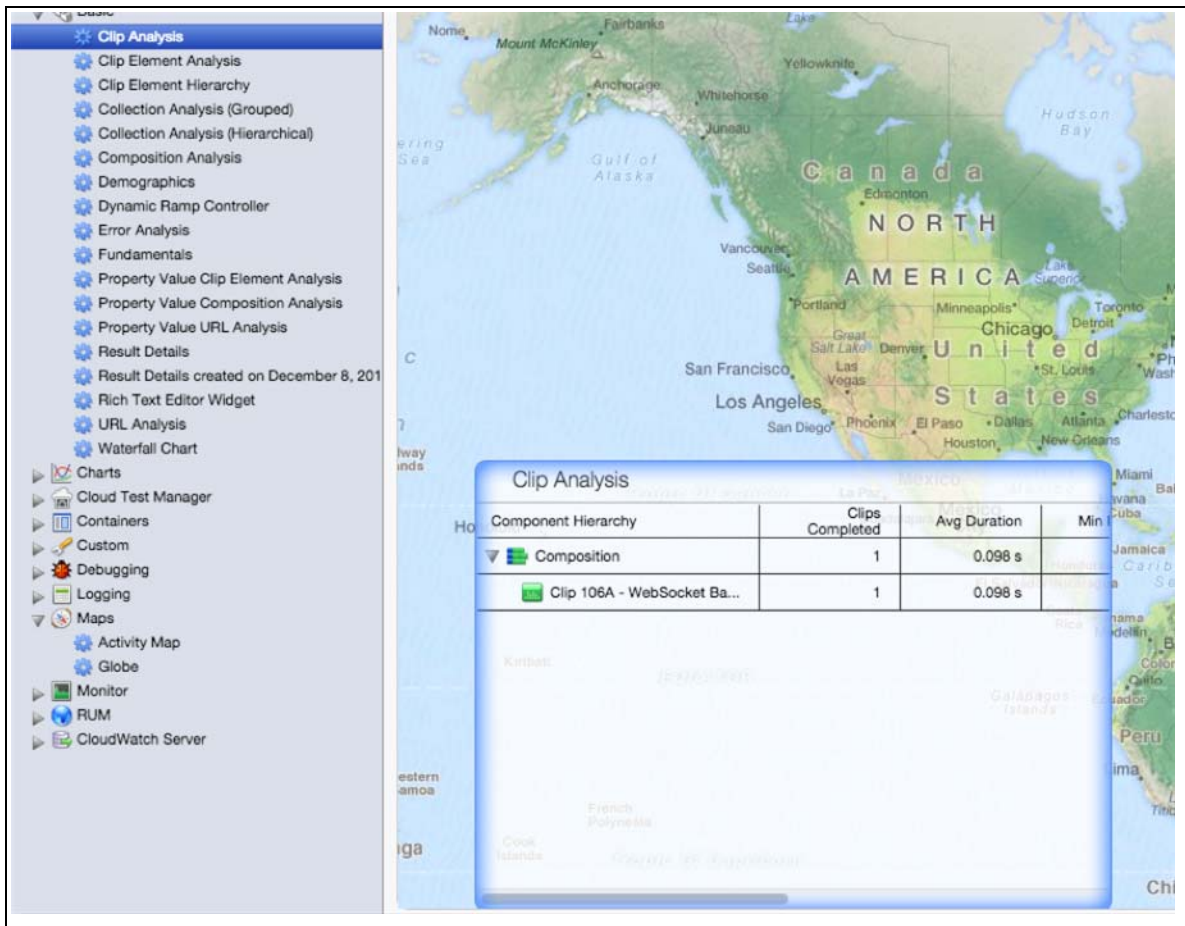
SOASTA 53 introduces the ability to Edge Constrain Widget-on-Widget (WoW) layout in CloudTest dashboards, as well as visual improvements, including a new, sleek Dashboard Filter Toolbar that can be applied in custom dashboards. These and additional general dashboard improvements are detailed in the following sections.

Widget-on-Widget Layout and Edge Constraints

This release resolves some issues with Widget-on-Widget layout that would result when the display was switched between monitors of varying resolutions. Additionally, This release adds the ability to edge constrain Widget-on-Widget (WoW) layouts in dashboards.

Edge constraint provides the ability to fix the widget's position relative to the left or right edge, or to the corner(s) of a dashboard.

1. To add edge constraints to a WoW in the dashboard, first select it.

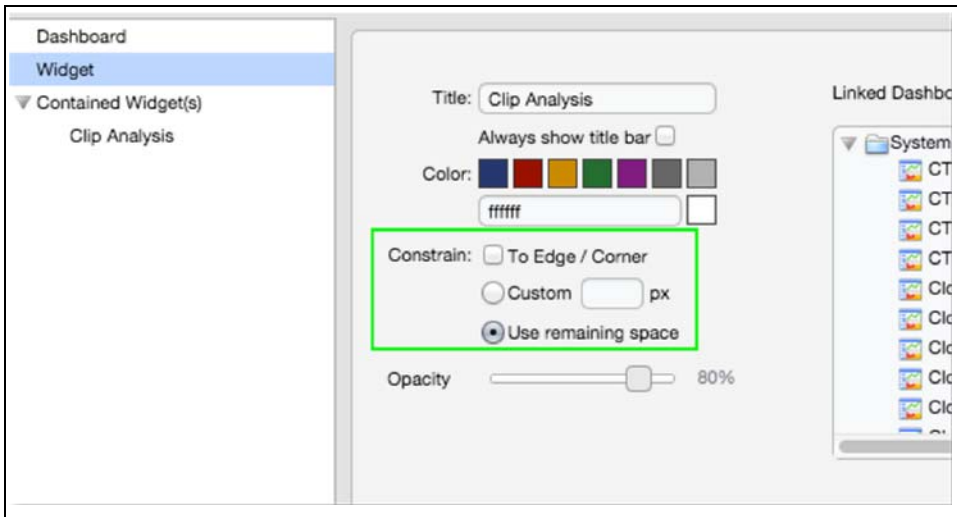


If you don't already have a WoW in the current dashboard, toggle the Widget Selection Panel on and add it now.

TIP: Refer to [Widget on Widget Layout](#) for more about this existing dashboard feature.

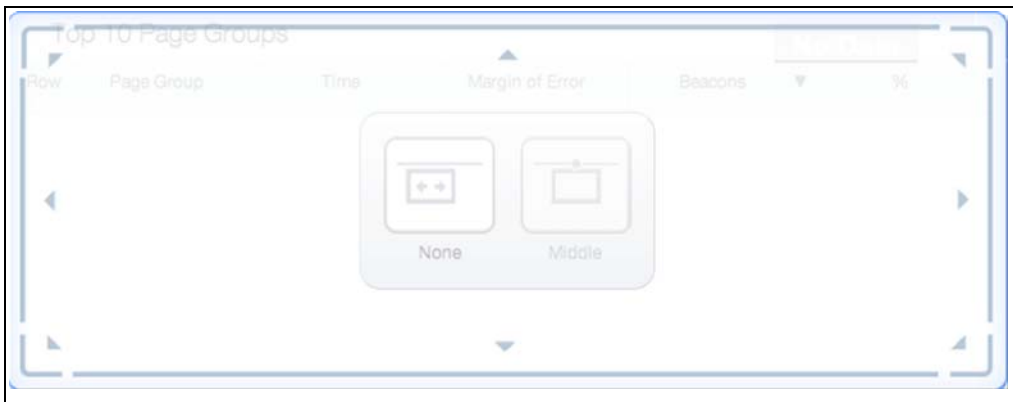
2. Click the Properties (Toggle Dashboard Edit Panel) button. The Dashboard Edit panel opens with the Widget node selected.

The Constrain settings section is shown below with the new To Edge/Corner checkbox.



The other two settings above existed in the prior release:

- *Custom Pixel Height* – Specify a custom pixel height for the selected widget (overrides To Edge/Corner)
 - *Use Remaining Space* – If no pixel height is specified, a WoW will expand and shrink to fit, it can also be drag resized or resized on Dashboard load (overrides To Edge/Corner)
3. Check the To Edge/Corner box.
 4. Click Apply (on the right of the panel). When you do so, the selected widget displays its Constrain Position Along Edge settings.

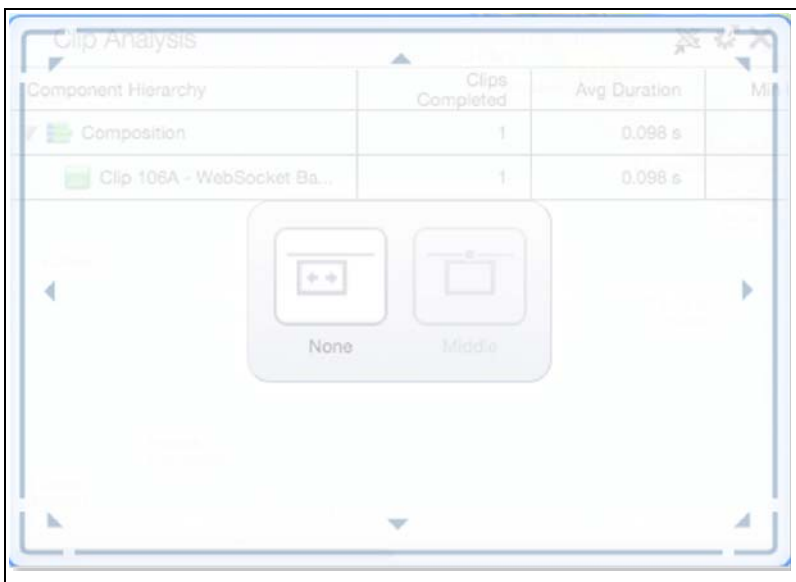


5. Choose a constraint mode.

- Click *Middle* to constrain widget to use a 50% vertical position (e.g. the widget will align to the vertical middle of the dashboard (e.g. from top to bottom)).
- Or, click *None*. In which case the widget is aligned to the position it was in prior to applying the edge constraint.

If *None*, was clicked, you can use the directional arrows in the Constrain Position box to snap-to any of the four directions: left, right, top, or bottom or to any of the corner positions.

When a given direction is selected a blue border appears along the selected widget's edge. For example, top is selected below.



Once you've selected one of the four directional positions (e.g. left, right, top, or bottom), you can additionally then click Middle again to adjust to the middle of the given direction.



Note that if the corner position is applied that the widget is then in an absolute position (e.g. it is not relative since the widget cannot move from there) so the subsequent Middle click doesn't apply.



6. Additionally, you can specify a Custom pixel position.
7. Or, click Use Remaining Space.

General Dashboard Improvements and Changes

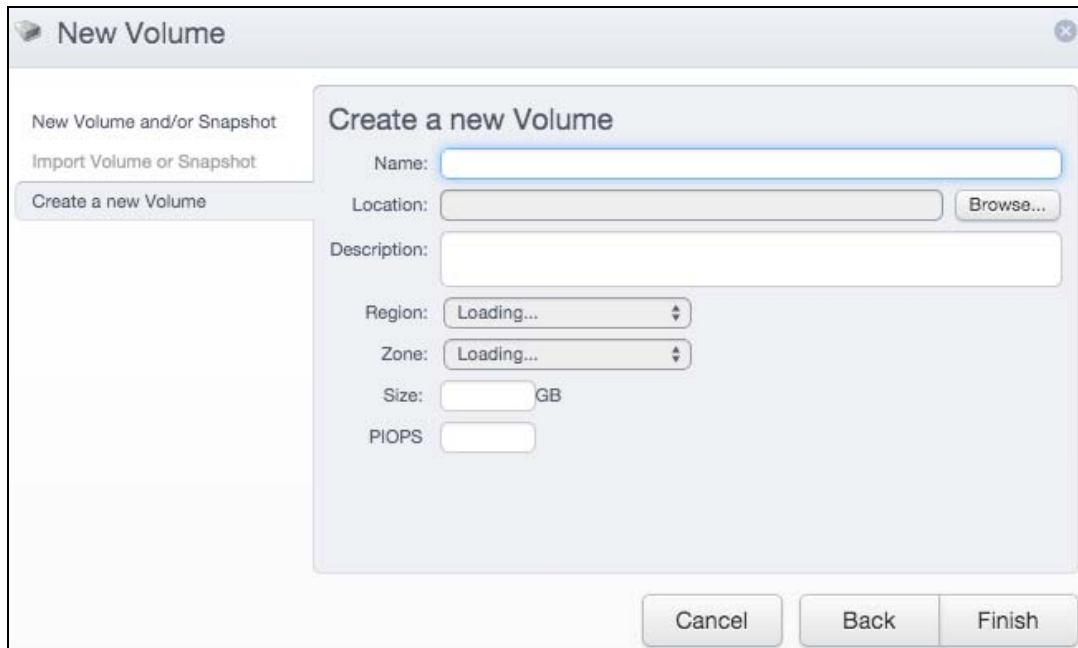
Other general dashboard improvements in this release include:

- New, more streamline Dashboard Filter Toolbar (where applicable in custom dashboards that have Display Filter Toolbar checked in Filter tab options).
- Visual improvements in Drag and Drop operations.

Additional detail about all of these enhancements is included in the following sections.

Ability to specify type and PIOPS when creating a volume

As of this release, CloudTest and CloudTest Manager support [Provisioned IOPS](#) for Amazon EC2 EBS volumes.



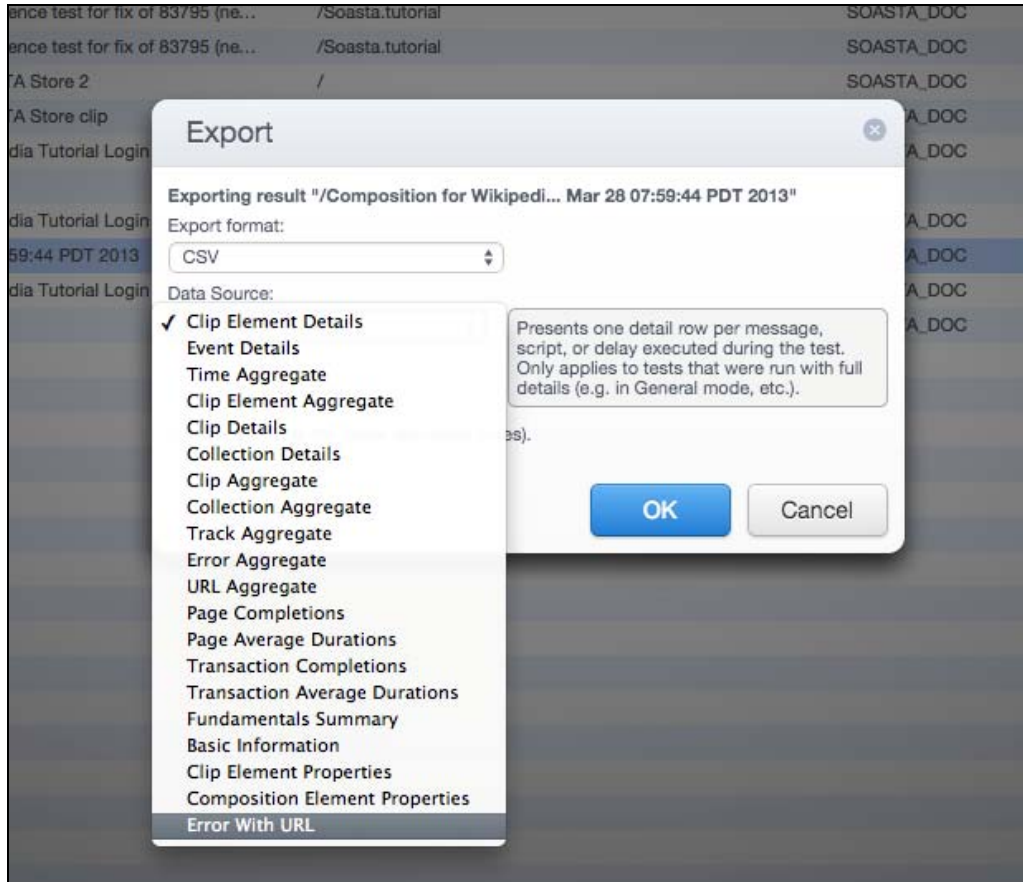
When creating a volume, you can optionally supply a number of PIOPS. If left blank, then a traditional EBS volume will be created. If a valid number is entered, however, then the volume type will be set to 'io1' and the specified number of PIOPS will be used.

Larger PIOPS will increase the Amazon storage costs. Consult the EBS documentation (linked above) for more details.

Exporting Widget data to Excel grabs all data in all trees (80301)

Getting errors in CloudTest reports into a single view proved difficult or impossible using previously supported CSV export formats.

For this release, a new Errors With URL option appears in the Export box, Data Source drop-down list whenever CSV format is selected.



This new option lists summary info about message and script activity for only those messages that had an error and includes full error detail including URL to support the user in correcting the given error(s).

Ability to specify type and PIOPS when creating a volume

As of this release, CloudTest and CloudTest Manager support [Provisioned IOPS](#) for Amazon EC2 EBS volumes.

New Volume and/or Snapshot

Import Volume or Snapshot

Create a new Volume

Create a new Volume

Name:

Location:

Description:

Region:

Zone:

Size: GB

PIOPS

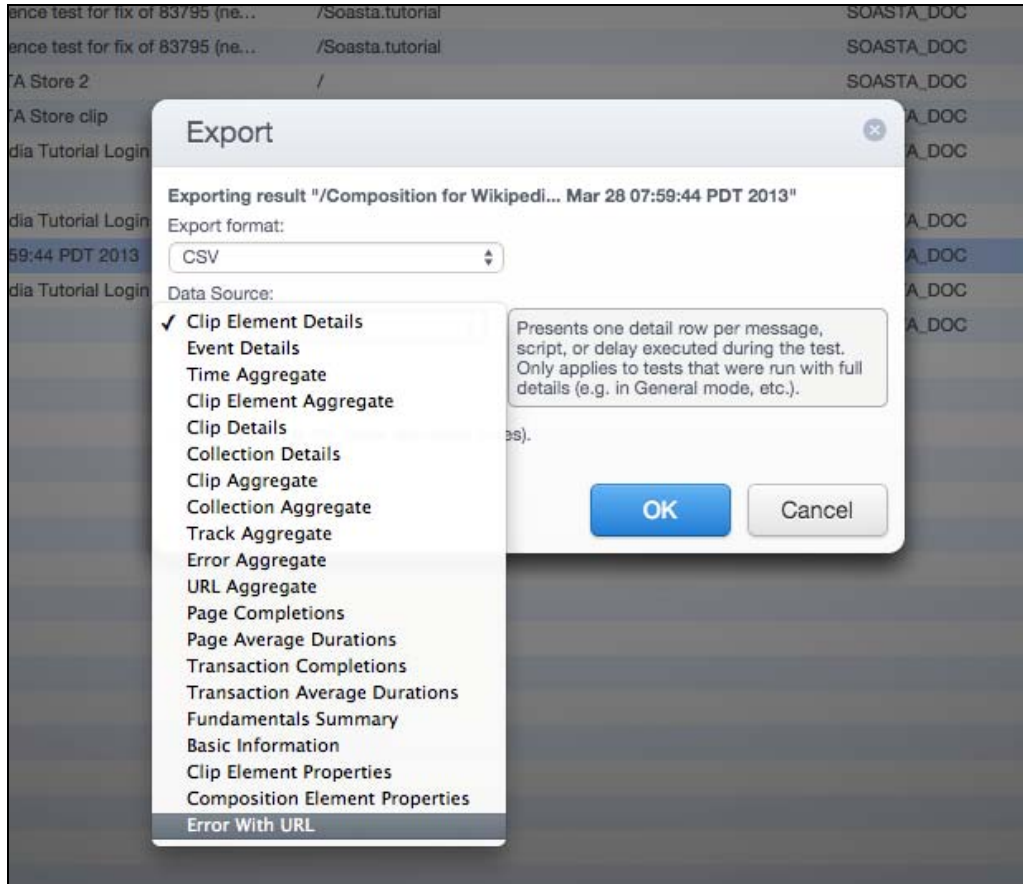
When creating a volume, you can optionally supply a number of PIOPS. If left blank, then a traditional EBS volume will be created. If a valid number is entered, however, then the volume type will be set to 'io1' and the specified number of PIOPS will be used.

Larger PIOPS will increase the Amazon storage costs. Consult the EBS documentation (linked above) for more details.

Exporting Widget data to Excel grabs all data in all trees (80301)

Getting errors in CloudTest reports into a single view proved difficult or impossible using previously supported CSV export formats.

For this release, a new Errors With URL option appears in the Export box, Data Source drop-down list whenever CSV format is selected.



This new option lists summary info about message and script activity for only those messages that had an error and includes full error detail including URL to support the user in correcting the given error(s).

Bugs Fixed

CloudTest

84808: WebSocket recorder does not work with NTLM (windows authentication)

Problems with NTLM authentication and WebSocket recording are now resolved.

84668: Blocked a frame from accessing a cross-origin frame

This java.lang.NullPointerException occurred in the Tenant Editor.

84561: Recording secure WebSocket (TLS) creates multiple duplicate targets that do not play back

An invalid target created during WebSocket recording also prevented playback.

84496: Namespace 'java' does not contain any functions (JS line 265)

This DOM-related error is now resolved.

84168: Invalid "Set-Cookie" header in response (unable to parse expiration date parameter: Sat)

An expiration date format whose value of expires was "Fri, 12 Sep 2014 21:39:11 -0000" was not handled.

This fix adds a date format string (below) to parse the date format:

"EEE, d MMM yyyy HH:mm:ss Z"

84097: Collection Aggregate Export Not exporting all it should

This fix improves aggregation performance for Browser Actions in order to improve scalability and fixes additional problems in thread synchronization for browser targets.

83942: Creating new Volume then RSDB fails during init and continue phase

Volume initialization unexpectedly failed while creating an RSDB.

83543: sCommand returns incorrect test duration time with Play Wait option

This fix resolves issues, with Transaction names that weren't lining up with query results as expected, and the second of which was that the response count was off by 1, resulting in the incorrect duration time (both issues are now resolved).

83461: Uncaught TypeError: Cannot read property 'getAttribute' of null; JS line 4247

Additional DOM element null checking has been added to detect similar occurrences of this error.

83196: Unable to Load SOASTA CloudTest (7198.180) UI using token

When using a token on the URL, also invalidate the Hash cookie, as the user may be changing.

82926: Lower panel does not correctly render international characters

Because of the way XHRs work, the server automatically encodes all Unicode characters as their HTML entity value - such as "〹" However, it doesn't work for HTML attributes such as the "value" attribute of an input. This is now handled.

82918: Rackspace servers don't show as running in CloudTest

This issue was caused by the "changed since" parameter of the list servers call used while polling pending servers to look for the state transition from pending to running. This setting had never seemed to work properly in that it seemed to return even servers that have not changed for a long time.

82549: TypeError: widgetTitleContainer is undefined; JS line 2173

This error occurred in the underlying jQuery object, which was assigned as widgetTitleContainer. On occasion, this was returning empty, so no index 0 and this error.

81899: java.lang.ArrayIndexOutOfBoundsException at PlaylistCompositionListFactory.getCompositionIDToOrdersMap

This exception could occur if a playlist was deleted while it was also still selected in Central, thus causing Central to list compositions for a deleted playlist.

81887: Uncaught TypeError: Cannot read property 'isVisible' of undefined; JS line 8564

Additional null checking has been added to detect any further occurrences of this issue.

81885: Uncaught TypeError: Cannot read property 'style' of null; JS line 11608

Additional null checking has been added to detect any further occurrences of this issue.

81848: Uncaught TypeError: Cannot read property 'style' of undefined; JS line 429

Additional null checking has been added to detect any further occurrences of this issue.

81820: For input string: "250125025075050050000070"

This fixes invalid long integer parsing in CloudTest Lite.

81809: Cannot read property 'length' of undefined; JS line 10994

Additional null checking has been added to detect further occurrences of this issue.

81745: Uncaught ReferenceError: L is not defined; JS line 1387

This error was likely due to an underlying timing issue on load.

81694: Unable to get property 'toUpperCase' of undefined or null reference; JS line 1699

This error occurred in the underlying keyboard controller, and was fixed by ensuring that an active element existed before transforming its tag name to uppercase.

81543: Uncaught ReferenceError: CheckServers is not defined; JS line 4426

Because of the asynchronous nature of the way HTML loads and executes JavaScript files, this timing issue happened occasionally. The code has been refactored to avoid this problem.

81483: Conductor installers are missing JDNC drivers for SQL Server and Oracle

This fix provides the missing indicated JDNC drivers.

81378: Activity Map JS error

This JavaScript error happened in the Activity Map.

81360: Waterfall Dashboard in a dashboard container causing a script error when clicking the bar

After adding a Waterfall widget to a Dashboard Container Widget, clicking the chart resulted in this error.

81288: Permiso denegado

This error message has been added to our throwaway messages list and will no longer appear.

81263: Null Start Time error

CloudTest will now handle a null start time by just using the last 5 minutes.

81127: Imported dashboards' widget headers are changed to white and the top menu in the widget gets invisible

In some cases after importing a dashboard, the Widget title bars appear white, which also made the top menu items invisible.

80722: Null check DOM - probably on XHR return

This null check occurred during an export into Excel format from the events and

status log.

81100: ReferenceError: CONTEXT_LOST_WEBGL is not defined; JS line 6450

An underlying Globe dashboard or widget error occurred due to dimension/size requirements that no longer apply.

80600: Uncaught NotFoundError: Failed to execute 'removeChild' on 'Node': The node to be removed is not a child of this node.; JS line

An underlying DOM error occurred.

80593: Uncaught ReferenceError: SoastaCentral is not defined; JS line

This Central error was a likely timing issue.

80426: Uncaught Error: Attempting to use a disconnected port object; JS line 69

This was caused by an extension that is throwing an error. The item has been added to the throwaway error list.

80128: External Data Service fault: 'External data source ID 0 is invalid.'

A user-friendly message has been added for this error, in which the External Data Source was found invalid, most likely from an expired session.

79844: 'undefined' is not an object (evaluating 'oParams.Param'); JS line 5738

This JavaScript error was encountered in the Clip Editor's inline editor while performing an action on a given supplement.

79256: Uncaught TypeError: Cannot read property 'OnLoadResultsSummary' of undefined; JS line 1044

This error occurred in the Clip Analysis widget likely while the Dashboard Edit Panel was open.

78985: Referential Integrity between Results Database and Server tables

This fix represents completion of Phase 1 of underlying referential integrity improvements, the lack of which caused customer-facing issues between RSDB and (Maestro) Server tables.

78969: Cannot delete a Results Database in the running state

A user attempt to delete an RSDB that was running failed unexpectedly. Now, the necessary shutdown and eventual deletion will occur as expected.

78907: Monitor chart with no resource type

Underlying conflicts led to a monitor that unexpectedly lacked a resource type.

78822: Unknown host

This DWR error occurred in a dashboard. Unknown Host has been added to the list of throwaway messages.

77883: ItemFlow is not defined

This error likely occurred while loading or unloading mobile tests.

77190: Failed to execute 'deleteCell' on 'HTMLTableRowElement'

This JavaScript error occurred when an empty fundamentals table JS was encountered.

77060: TypeError: f is null; JS line 3346

Sometimes a null object would be uncaught while reporting an error messages, causing this 2nd error message to occur instead.

76692: Uncaught TypeError: Cannot read property 'maxManualVUCount' of undefined; JS line 56

Additional initialization checking should guard against this error.

76163: ResultsServiceFault: Exception: could not resolve property

If you set a widget filter before combining, then combine, then change the type of the widget that had the filter it doesn't reset the filters.

74196: Property Chooser doesn't show columns for database seed data object

The Property Chooser UI will now also show Seed Data columns.

73863: Cannot open connection

This fix adds Exception: Cannot open connection to throwaway messages list.

73829: Unable to get property 'length' of undefined or null reference

Additional null checking to evaluate data has been added.

73824: Widget Filter: UnsupportedOperationException

App action was not included in the backend as a filter possibility.

73821: java.lang.NullPointerException

This happened if the user opened a composition that contained a clip to which the current user only had "use" permissions.

73715: Modifying seed data objects without write permission does not throw an exception

The original fix for this issue was to enforce read-only mode for Seed Data to which the user lacked sufficient permissions.

This subsequent follow-up work covers the edge case in which seed data write privileges existed on load, but then write privileges were removed while the seed data file was open. In this eventuality, the user will now no longer be able to make the changes.

73431: Tunnel Connection failed

This fix adds Tunnel Connection Failed to throwaway messages list.

71838: waitForElementText not showing observed output value in the input section

The wait element was only being included if the wait failed.

70023: Horizontal Alignment of Labels in a Table

The display of labels in dashboard tables (i.e. the General widget, Effective Duration section) has been fixed by adding vertical-align:top; to elements with this issue in the XSL template.

69006: Safety checks to avoid "TypeError: 'null' is not an object" and "'undefined' is not a function"

This issue has been resolved by adding safety checks. This fix also resolves cases 62573 and 57554.

63938: Cannot undo 1 seed data operation(s) (exceeds the total number of operations that have been performed)

The Save button in the Seed Data Wizard was not clearing out the UI version of the command stack, so clicking undo after saving (which clears all commands and persists them) was sending an undo request when there was nothing to undo. Now, this is handled as expected.

63018: "Volume is currently in use" when trying to delete a volume

A new user-friendly error message has been added, along with an underlying network error dimension check.

60631: Changed grid journal message per case

The Grid Manager's, Step 3, Journaling tab now displays Journaling messages by case if Migration fails during Test Environment deployment.

56259: Long delay between compositions run in a playlist

This fix introduces significant back-end performance improvements—including the use of new XFS disc formatting—that correct the increasing time detected between compositions executed from a CloudTest playlist. The time to create tables and indexes was confirmed as taking up to several seconds each more than expected. Additionally, the overall composition duration went from very short to a couple of minutes.

55051: Exception: The results database "Shared Cloud ResultsDB" is not running

The UI now handles this error, which would occur while deleting compositions, monitors, or results from Central.

54381: Changing your user name doesn't update the username in the upper right hand corner

The expected user name will now appear.

53519: Monitor charts seem to appear in Random

Monitor charts are now alphabetized in the Central > View Analytics node.

51768: Mass Target Change menu is appearing behind the Choose Target and Action menu and not selectable

Opening the Choose Target and Action, and right clicking on the target pops up the Mass Target Change menu behind the current window.

51389: Delete the parent folder while a Script is open eventually causes "internal error"

Deletion of the parent folder will now result in a user-friendly error message.

54479: Mass Target Change does not display tab has changed warning when closing the tab

After a Mass Target Change was used, closing the Clip Editor tab failed to prompt the user that unsaved changes existed, after which reopening the clip the change wasn't saved.

50263: Choose Target and Action highlights incorrect included Target

The Choose Target and Action command highlighted an incorrect target in the Include Targets list.

45064: No matching SOAP target found for recorded message

This fix modifies clip generation so that binaries with missing SOAP targets are skipped in the conversion, rather than throwing an error.

43547: Recording Editor needs to handle DNS failures

The recording proxy didn't gracefully handle failures to get the IP address.

38905: unhandledExceptionTermination still around in the server code

This legacy code has now been removed.

24988: Firefox extension should include "CloudTest" in the name

The Firefox browser extension no longer refers to itself as being a part of CloudTest.

6972: List of Monitored Servers should be alphabetized

The Central, Monitoring Server Groups list was not alphabetized.

4100: Deleting a folder does not remove contained playlists from the UI

Folder deletion left contained playlists orphaned. For example, after creating a Playlist and moving it inside a Folder, folder deletion would leave the Playlist in displayed in the Central, Playlists list (e.g. the one in the tree rather than the detail list view).

1901: Incorrect spacing in Safari if service has WSDL documentation

In Safari, creating a target by importing a WSDL that has documentation (e.g. the Coordinator Service) would result in a big space in the middle.

TouchTest

[Release [requires new TouchTestDriver](#)]

84717: Fix the evaluation of the output messages that are not in JSON format

Fixed the evaluation of the output messages that are not in JSON format.

84644: TouchTestable app is not working on iOS8

Problems running a given mobile app under iOS8 were encountered.

84579: Outputs configured on target level are not shown on the composition execution

Target-level outputs didn't work as expected.

84549: Clips hangs indefinitely on action with builtIn-waitForElementPresent

The test would stall while waiting for the built-in wait action.

84365: Android TouchTest Web: During execution of clip an alert appears and blocks clip execution [\[requires new TouchTest Web\]](#)

Fixes alert pop-up concerning "Cannot call method 'appendChild' of null" during playback.

84365: Android TouchTest Web: During execution of web clip an alert appears and blocks clip execution

Changed logic to seek for when onload is true for when to check for this value.

84363: iOS TT ScrollToVisible is always failing with builtIn-waitForGestureComplete has failed

This fix includes additional support for scrollToVisible for UICollectionView (vertical scrolling only).

84299: TTIDs change on TableView scroll

Refactored code was added in order to find TTIDs in a view in its own function.

84275: Android Hybrid app crash after search button clicked [\[requires new TouchTest Web\]](#)

Using an eval() with a JavaScript proxy caused Android 4.3 (JellyBean) WebView to crash with a signal 11 sigfault in specific circumstances when calling loadUrl on the WebView.

84199: IOS - TouchTest not Recording Native App

If the app used the method "application:openURL:sourceApplication:annotation:" to perform some task, this error was invoked. The fix is to simply not load TTD again. We should never load multiple TTD instances in the same session.

84150: iOS: Clicking on Image on Homepage navigates back to TouchTest Agent

Clicking an image on the homepage of a given app would cause recording to stop.

83951: iOS TT: App crashes when taking screenshot of embedded map

Adding captureScreenshot at the target level crashed the mobile app.

83906: Android_Phone: Tap action doesn't work under Titanium Calendar component

After recording, the error "The element located by 'id=calendar_Day_20140526_Button' is not visible, not accessible or not enabled" would appear. Selecting any of the other recorded locators also didn't produce the expected result.

83897: Adding logging in Coordinator to specify how long we will wait for a response on any specific action

A workaround has been provided that utilizes the Auto-Stop option in the Composition Editor. Additionally, logging has been added to detect further occurrences should this behavior recur.

83894: String index out of range: 8 exception while using MakeAppTouchable

Regression due to the change supporting Japanese chars in IPA names.

83851: Fails to playback on KeyBdReturn on Android

Playback does not show error on KeyBdReturn but does not play it back, and then shows an error on the next step.

83758: Actions locator: UIWWEBVIEW NOT READY, delete this and try again

Attempt to re-inject JS in both recording and playback, if the JS is found to be missing in the middle of recording/playback.

83501: iOS - Unable to play back application on Menu buttons

For performance reasons, we optimize locators after the action is finished. In this case, the table was being redrawn, re-using menu items so that what used to be a correctly recorded locator became an inaccurate locator.

83195: Recording does not work on custom keyboard

When opening a custom keyboard while using Android 4.2 and 4.3, the app will no longer crash.

82743: Android build fails to locate items in GNM list during replay

After replaying the recorded steps, the last clip will fail with this error: "The element located by "id=globalNavBar_transfers_Label" is not visible, not accessible or not enabled."

82062: iOS TT: Scroll to Visible doesn't seem to scroll far enough to tap the item

In Google Maps, after searching for a location, scrollToVisible was used to scroll the text="Report a problem" within the location menu. It seems to scroll to the correct link, but not far enough so that we can tap on it on the next action. Which subsequently fails in the next action playback.

80257: Touch Locator is not returning all strategies

While recording, TouchTest doesn't record all location strategies for performance reasons, but the Touch Locator tool should always return all possible location strategies.

79946: Memory Leak in Clip Editor seen on FF and Chrome

Memory leaks occurred in SDK-based builds in Firefox and Chrome test environments using Titanium Studio.

79754: Password not recorded as webTypePassword

When recording, the string the user types into the 'Password Input' field is not recorded as a webTypePassword, as it should be.

78384: Touchtest URL is overlapping on "User Name" field in "Touchtest Agent"

The TouchTest URL overlapped the "User Name" field in "TouchTest Agent."