

HP LoadRunner

Software Version: 11.00

Ajax TruClient Tips & Tricks

Document Release Date: October 2010

Software Release Date: October 2010





Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© Copyright 2010 Hewlett-Packard Development Company, L.P.

Trademark Notices

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.



Table of Contents

Object Identification Issues.....	5
Overcoming Object Identification Challenges	5
The Difference Between Improve Object Identification and Replace Object.....	7
Objects are Shared Between Steps	7
What are the Different Object Identification Methods and which One Should be Used.....	8
How Related Objects Can Help the XPath and JavaScript Identification Methods	8
Interactive Replay Fails with Object Not Found Error	9
Interactive Replay Fails Due to Object Not Found Although Highlight Locates the Object.....	9
Some Tips Regarding Related Objects	9
How Can I Use JavaScript to Locate Objects?	10
Replay Fails to Select an Item from a List	10
Script Development	11
How to Select an Object in the Application when Mouse and Key Actions are Required.....	11
A Little about Parameters	11
Parameters Do Not Work when Running a Single Step	11
How to Check for Specific Text Including Branching.....	11
How to Check for Specific Text that is Case Insensitive.....	12
How to Select a Random Option from a List.....	12
How to Use External Functions in the Script.....	13
Some of the Events and Actions do not Appear in the Recorded Script	13
Dragging of a Slider or a Map does not Replay Correctly	14
Polling for an Object	14
How do I Create a WHILE Loop	15
What to do with the Word Verification Function in the Business Process.....	15
Load and Load Preparations	16
How to Make the Script as Stable as Possible	16



- How to Fix a Script that Replays in Interactive Development but Fails During Load..... 16
- How to Debug the Script in Load Mode..... 17
- How Many Ajax TruClient Vusers Can I Run on a Single Load Generator?..... 17
- Script Settings 18
 - How to set the Recording Options (e.g. Proxy)..... 18
 - How to Set Proxy Values for all the Ajax TruClient Scripts 18
 - How to Add a Plug-in and Some Bookmarks to all the Profiles used by the Different Scripts 18
 - Browsing Fails in the Ajax TruClient Firefox, Although Regular Firefox Browsing is OK..... 19
- Miscellaneous 20
 - Where is JavaScript Supported 20
 - How Can I Learn More about JavaScript..... 20
 - Additional Debugging Tips 20
 - General Tips Regarding Successful Interactive Replay 21
 - General Tips Regarding Successful Recording 21
 - Resolving Step Timeouts..... 21
 - Using Regular Expressions 21

Object Identification Issues

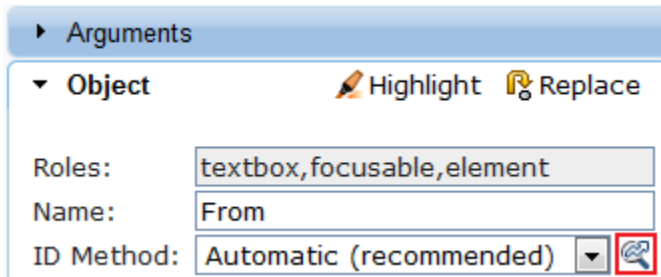
Overcoming Object Identification Challenges

One of the biggest challenges with recording and replaying Web 2.0 applications is object identification.

In Web 2.0 applications the object properties may change every time the application is accessed, making it more complicated to accurately locate an object.

Ajax TruClient includes very sophisticated mechanisms to overcome this challenge. It also enables manual actions to help improve identification when needed.

- Start by expanding the failed step and using the Highlight function to see if the required object is located. If it is, then this may be an issue of pacing and timing. Please refer to the other tips that discuss Object Not Found errors during replay.
- If the Highlight function fails, try using the Improve Object Identification function.



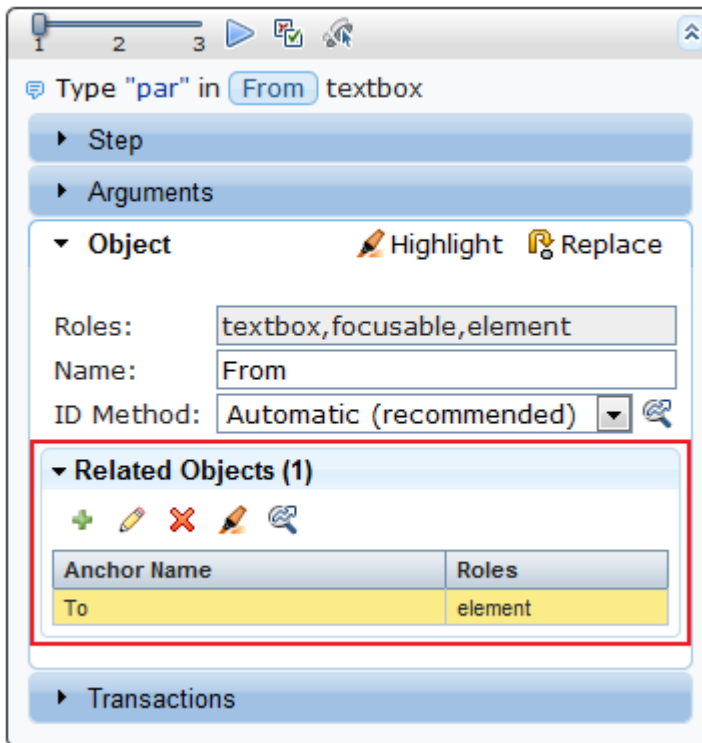
This will let TruClient relearn the properties of the object and compare them to the properties learned during recording. Based on the differences, the necessary adjustments can be made. Once you have done this, try replaying the step again to check whether the problem has been solved.

Note: Depending on how dynamic the application is, you may need to use the Improve Object Identification function more than once.

- If this fails, try using one of the alternative steps.
For example, you may be clicking on an option in a drop down list in which the text changes based on some value.
If you try to click based on the text, the step may fail.
If you use an alternative step that selects the item in the list based on the ordinal value of the option within the list, the click will succeed regardless of the text.

Before selecting one of the alternatives, try highlighting the object used by the alternative step and replaying it. This way you'll make sure the alternative step is replaying the necessary action.

- If the Improve Object Identification function does not solve the issue and neither do any of the alternative steps, try using the Related Objects option.



Using this functionality you can define one or more objects that appear on the screen in relation to the object you're looking for.

This provides an additional method of locating the required object.

Note: The Related Objects option may negatively affect performance.

- After you perform any of the changes, replay the single failed step in question and only afterwards replay the whole script again. This will help verify whether the change has solved the issue you encountered.
- In specific cases where you have a known XPath or JavaScript function that can locate the object, you can also change the ID Method used to XPath or JavaScript respectively. In most cases it is recommended to leave the ID Method on Automatic.
- There are also less common cases in which the application only loads objects when a scroll operation brings them into view.



If this is the case use the step to scroll object into view (use an object that always exist) or just use the scroll method based on pixels.

This will bring the required object into view and make sure the application loads it before a subsequent step attempts to use it.

- If replay stops with an error that says the object was not found try the following:
 - Select the failed step and press Highlight. If the correct object is not highlighted, there is an object identification issue and you should use Improve Identification to improve object identification
 - If the object is highlighted, it may be that the step was reached before the object appeared. Add a Wait or Wait for Object step before the problematic step.
- Sometimes you may need to choose an Alternative Step that is provided in order to solve the issue.

The Difference Between Improve Object Identification and Replace Object

Improve Object Identification will tell TruClient to relearn the properties of the object and compare them to the properties learned during recording. Based on the differences, the necessary adjustments can be made.

Replace Object will tell TruClient that the object currently referenced in the step is incorrect. TruClient will remove any current knowledge of the object and learn the object you select from scratch.

Therefore, you should only use the Replace Object option if the object you used during recording was the wrong one.

If you recorded using the appropriate object you should use the Improve Object Identification option.

Objects are Shared Between Steps

If the same object in the application is used in multiple steps, Ajax TruClient shares the object among these steps.

This means that changing the object name in one step will also change it in the other steps and using Improve Identification in one step will affect the other steps also.

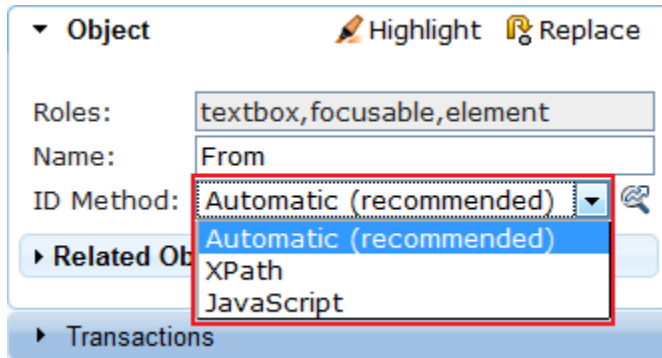
If the Replace function is used on one of these steps and a different object is selected, the objects will no longer be shared.



What are the Different Object Identification Methods and which One Should be Used

TruClient has 3 object identification methods: Automatic (recommended), XPath, and JavaScript.

These can be selected from the Object section of the step properties:



The Automatic method allows TruClient to use its internal advanced algorithms to locate the object.

The manual options of XPath and JavaScript provide a means to enter an XPath or JavaScript expression that will locate the object. These options override TruClient object location algorithms.

Note: TruClient automatically creates an initial value for the XPath identification. This can be replaced using the Regenerate Expression option.

In most cases it is recommended to use the Automatic option.

If Automatic identification fails, even after using Improve Identification and Related Objects, try using the XPath identification method.

For example, if you need to select the first search result, regardless of the term being searched for, using XPath identification may help.

How Related Objects Can Help the XPath and JavaScript Identification Methods

The XPath and JavaScript identification methods may return multiple elements depending on the expression used.

For example, if the XPath value is `//button`, and the web page in question includes multiple button elements, multiple objects will be returned.



To return a single object you can add a Related Object that will narrow down the identification.

Interactive Replay Fails with Object Not Found Error

- If replay stops with an error that says the object wasn't found try the following:
 - Select the failed step and press Highlight. If the correct object isn't highlighted there's an object identification issue and you should use Improve Identification to improve object identification
 - If the object is highlighted it may be that step was reached before the object appeared. Add a Wait or Wait for Object step before the problematic step.
- Sometimes you may need to choose an Alternative Step that is provided in order to solve the issue.

For example, you may be clicking on an option in a drop down list in which the text changes based on some value.

If you try to click based on the text, the step may fail.

If you use an alternative step that selects the item in the list based on the ordinal value of this option within the list, the click will succeed regardless of the text.

Interactive Replay Fails Due to Object Not Found Although Highlight Locates the Object

If the replay fails even though the Highlight option finds the correct object, this may be a case of pacing.

The object takes a little longer to load and the step is executed faster. Therefore, during execution the step cannot locate the object although, during debugging, the Highlight option on the failed step finds the object.

In this case it is recommended to "slow down" the script so there's enough time for the object to load. Use one of the following options to do this:

- Change the Object Timeout of the failing step. This is available via the Step section on the step's properties.
- Add a Wait or Wait for Object step before the failed step

Some Tips Regarding Related Objects

- Use this feature only if other identification methods have failed as it may be more resource intensive



- Use the minimum search area to improve performance.
- Related Objects are sensitive to window sizing. Resizing may alter object positions and relationships. This should be taken into account.
- Each identification method (Automatic, XPath, and JavaScript) has its own set of related objects. These related objects are not shared between identification methods.
- If several relations exist they all need to be found in order for the identification to succeed (AND relationship).

How Can I Use JavaScript to Locate Objects?

Using the JavaScript identification method you can write JavaScript code that references the returned document and can use CSS selectors and other standard functions.

For example, the page returned by the server contains multiple links with the same “title” attribute (search results) and we want the script to randomly click on one of the available links.

Object identification for this case, using the JavaScript identification method, may look something like this:

```
var my_results = document.querySelectorAll('a[title="SearchResult"]');  
my_results[Math.floor(Math.random() * my_results.length)];
```

Replay Fails to Select an Item from a List

One of the common reasons for this is that the names of the items in the list are dynamic.

For example, the list may include a list of cities based on the text entered so far (auto-complete).

Based on the text types the list constantly changes.

There are 2 ways to solve this issue:

- Use an alternative step that selects an item from the list using the ordinal identifier instead of the text of the actual item.
- If the text is only partially dynamic you can use a regular expression to locate the required item based on partial text matching.

Regarding regular expressions see the relevant tips in this document.



Script Development

How to Select an Object in the Application when Mouse and Key Actions are Required

Highlight, Improve Identification, Replace, Related Objects etc. all require the user to select an object in the application.

There are cases in which various actions are required in the application to make the object visible (e.g. mouse over, mouse click etc.).

In these cases use the CTRL+ALT+F4 option to suspend the TruClient object selection mode until you've brought the object into view and press CTRL+ALT+F4 again to select the object.

A Little about Parameters

Parameters are added via VuGen, just like any other protocol.

They can be referenced in the script using the LR.getParam and LR.setParam functions.

When using parameters you may also need to change the way a step is replayed so it functions correctly regardless of the parameter value.

For example, let's assume we have an auto-complete list that has a different set of values based on the text that's typed.

Trying to select an option in the list based on the text of the option is bound to fail once the typed text is defined by a parameter. The options are changed every time the parameter value is updated.

In these cases the alternative step that uses an ordinal value is more appropriate.

Parameters Do Not Work when Running a Single Step

Parameter values are only used when the script is fully replayed.

The parameters are not used when you replay a single step.

How to Check for Specific Text Including Branching

One option is to add a Verify step from the Functions section of the Toolbox. In this step you can select various validation settings such as the object, the text to look for etc.



When you want to perform certain actions based on whether a validation succeeded or failed you add a Catch Error step from the Flow Control section of the Toolbox.

This way you can make sure the step continues even if the validation failed and within the Catch Error group you can define the set of steps that should be executed if the validation fails.

You can also take a more programmatic approach to validations. Using JavaScript you can access the DOM and validate any property you wish to verify. You can then add a conditional Break or Exit step (available via the Flow Control section of the Toolbox) based on this verification.

You can also check for the required text directly from the IF statement. The 'condition' argument of the IF statement is simply JavaScript code. You can use JavaScript code that accesses the window global object of the application under test. This can be done by referring to window.

Then you can manually verify if the text exists within the current page. For example, assuming a single frame application, you can write something like:

```
window.document.body.textContent.indexOf("Off") == -1
```

Where "Off" is the text you're looking for and -1 indicates that the text was not found.

The code in question is application specific.

You can optimize the code if you have further knowledge of the application (by getting the specific element).

How to Check for Specific Text that is Case Insensitive

By default, the Verify step is case sensitive. For example, looking for '**Test**' will fail if '**test**' is found.

If you'd like the Verify step to be case insensitive do the following:

- In the Verify step set the Condition argument to "Regular Expression"
- To check if the string '**test**' is contained in the text regardless to case you can use:
`RegExp("test", "i")`

How to Select a Random Option from a List

Set the Ordinal argument to 0. TruClient will automatically select a random option from the list.

For example, let's assume you have an auto-complete list that shows a list of cities based on the typed text. You've currently selected the second option and the step is: **Select option #2 from City autocomplete.**



All you need to do is open the Arguments section of the step properties and change the Ordinal argument to 0. The step will now be: **Select a random option from City autocomplete.**

This option is very important when the typed text is a parameter and therefore you have no easy way of knowing in advance what values exist in the list and how many there are.

How to Use External Functions in the Script

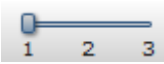
Add your JavaScript and C functions to the JS-Function.js and C-functions.c files that are part of the script and appear in the left navigation pane in VuGen.

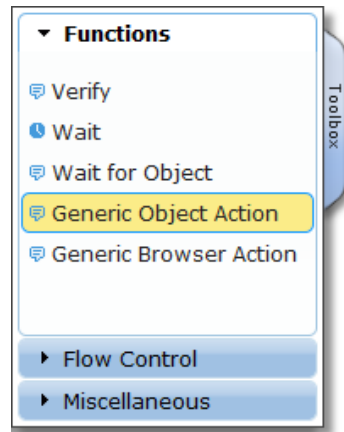
JavaScript functions can be called directly from the Ajax TruClient script, as all arguments and parameters support JavaScript. You can also add an Evaluate JavaScript step from the Toolbox for this purpose.

To call C functions add an Evaluate C step from the Toolbox.

Some of the Events and Actions do not Appear in the Recorded Script

Try solving this in one of the following ways:

- Ajax TruClient records all the events in the application. The event you're looking for may be in a different script level from the one being displayed.
 - You can tell that additional steps exist in other script levels if the steps in the viewed level are not numbered consecutively.
 - The current script level is set using the slider in the toolbar:  Try looking for the missing event or step in the other levels by changing the slider value.
 - Once you have found the required event you can change its level and make it part of level 1.
 - Change the script level back to level 1 and try to replay again.
 - Please consult the Ajax TruClient documentation for a full explanation of the script level concept.
- You can manually add a step to the script.
 - From the toolbox select the Generic Object Action and customize the step to perform the required action:



Dragging of a Slider or a Map does not Replay Correctly

If drag does not work (e.g. set option of slider, drag of map) and the result does not bring the control to the appropriate place try the following:

- Try using one of the Alternative Steps and see if it helps
- Set the values manually until they meet your needs (e.g. the precise number of pixels you'd like to drag in each of the relevant directions)
- Try using the "Drag to" capability (by changing the Action of the Drag step in the Step section of the step properties). This way you can drag your object to a relative position from another object.

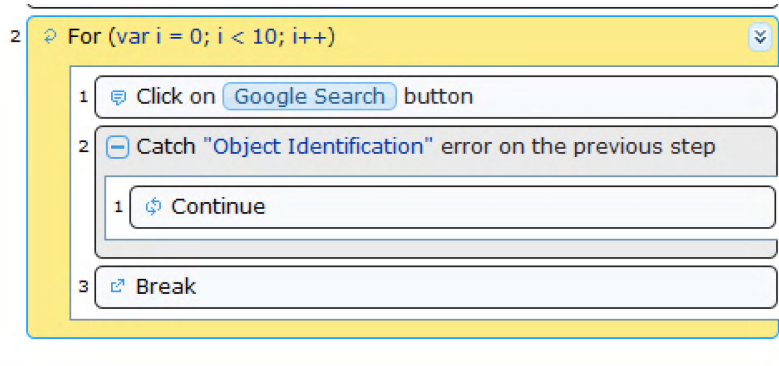
Polling for an Object

Create a loop that includes the following steps:

- A step that performs an action on an object.
- A Catch Error section that includes a Continue step.
- A Break step.

The Catch Error section will make the loop continue until the object is found and the step succeeds.

For example:



How do I Create a WHILE Loop

A “For” loop has 3 arguments: Init, Condition, and Increment.

A “While” is basically a “For” loop with only the Condition argument.

To create a while loop add a “For” loop (using the context menu or the toolbox), delete the Init and Increment arguments and specify the Condition.

What to do with the Word Verification Function in the Business Process

Many web sites use special Word Verification fields that displays some text that the user must enter in order to validate that an actual user is the one filling out the form.

This is designed to block crawlers, spiders etc. from using the site and taking up valuable system resources.

These text fields normally look something like this:



These fields are especially designed to block automatic tools such as LoadRunner.

In order to complete your business process automatically you must cancel this function on the web site against which you are running the load.



Load and Load Preparations

How to Make the Script as Stable as Possible

One of the characteristics of Web 2.0 applications is objects with dynamic attributes.

For example the ID of an object may have different values every time the application is accessed.

To make sure the script is stable and can run even when these changes occur, the following steps are recommended:

- After you have successfully replayed the script, make sure you replay it again several times in the Firefox interactive script development sidebar.
- If any of these replays fails, use the various methods for improving object identification (e.g. Improve Identification, Related Objects, and Alternative Steps).
- If the application objects change based on certain events (e.g. IIS Reset), make sure these events occur between replays, to validate the scripts stability.
- Use the Run option in VuGen to run the script again in load mode and validate that it is ready to be incorporated into a load test.

How to Fix a Script that Replays in Interactive Development but Fails During Load

It is highly recommended to use the Run option in VuGen to test the script in load mode before actually using it in a load test. The Run function plays the script the way it would be played in a load test and enables you validate it before including it in a load test.

If the script fails during Run in VuGen or in the actual load test but functions correctly in interactive replay please follow these steps:

- Look at the replay log to find out what the error is.
- If the error is “object not found” this may be due to pacing. Load mode replay is faster than interactive replay and the object may not be loaded in the application quickly enough.
- You can resolve this using one of the following methods:
 - Use the “Inter-step interval” run-time setting which allows the user to specify the time (in ms) between steps. This setting will be used for ALL the steps in the script.
 - Add a Wait or Wait for Object step before the step that failed. This will slow down the replay.



- Some applications may be sensitive to the browser size. When the script runs under load, a certain predefined window size is used which may cause object location failures. To solve this you can either add a Resize step to the script or set the initial width and height for non-interactive mode replay. This can be done using the 'Non-interactive window size' setting, located under 'Other Settings' in the Run-Time Settings.

How to Debug the Script in Load Mode

Consult the log and get a better understanding of the error.

To view the state of the application when the error occurred go to Run-Time Settings and select Snapshot on Error and replay the script. The snapshot will be generated when the error occurs and will be saved under the results directory based on the run logic (e.g. under Results > Load > [Action-Name] > [Iteration-Number]).

You can also add additional messages to the log by using the LR.log function inside the script itself.

For more information on this function please consult the LoadRunner Ajax TruClient documentation.

How Many Ajax TruClient Vusers Can I Run on a Single Load Generator?

The amount of Ajax TruClient Vusers that can run on a single load generator machine depends on the application under test and hardware parameters.

Internal HP benchmarks indicate that for various applications under test, a single Ajax TruClient Vuser can utilize around 60-120MB of memory (footprint) and consumes 3-30% of single CPU core.

We recommend assessing the resource utilization parameters per application to arrive at a realistic number for sizing of the load test.



Script Settings

How to set the Recording Options (e.g. Proxy)

The Run-time Settings are used for recording also.

Open the run-time settings for script and configure them. They will affect the recording.

How to Set Proxy Values for all the Ajax TruClient Scripts

LoadRunner launches Firefox in a temporary profile every time a script is opened for development.

To make the proxy settings available across all these profiles please use the global profile settings.

To do this:

- Open VuGen.
- Open an Ajax TruClient script.
- Go to Tools > Ajax TruClient Browser Options...
- In the dialog that opens (“Ajax TruClient Browser Configuration”), select the Proxy tab and define the appropriate Proxy settings.
- In the script, open the Run-Time Settings dialog and then select the General > Other Settings node.
- In the Proxy selection node select the “Use global proxy settings” option. This will make sure that every time the script is opened for development the proxy settings are refreshed from the Ajax TruClient Browser Options.

The Run-Time Settings also enable defining specific proxy settings for each script.

How to Add a Plug-in and Some Bookmarks to all the Profiles used by the Different Scripts

LoadRunner launches Firefox in a temporary profile every time a script is opened for development.

To make plug-ins and bookmarks available across all these profiles please use the global profile settings.

To do this:



- Open VuGen
- Open an Ajax TruClient script
- Go to Tools > Ajax TruClient Browser Options...
- In the dialog that opens (“Ajax TruClient Browser Configuration”) use the Extensions and Bookmarks tabs to define global plug-ins and bookmarks respectively.

Browsing Fails in the Ajax TruClient Firefox, Although Regular Firefox Browsing is OK

LoadRunner launches Firefox in a temporary profile every time a script is opened for development.

Failure may be due to incorrect proxy settings.

Check the proxy settings in your regular Firefox and then update them in the Ajax TruClient Browser Options dialog and in the script Run-Time Settings.

For more details see the tip: [How to set proxy values for all the Ajax TruClient scripts](#)



Miscellaneous

Where is JavaScript Supported

The arguments listed in the Arguments section of each step are all JavaScript based and can accept JavaScript expressions.

It is important to remember that to provide a string value, quotation marks are required.

For example: City will be interpreted as a variable whereas "City" or 'City' will be evaluated as a string.

All other sections (e.g. Step, Object, Transactions) are not JavaScript based, are not evaluated as JavaScript and do not support JavaScript expressions. The only exception is object identification using JavaScript.

Some examples are XPath identification, object names, timeout settings etc.

How Can I Learn More about JavaScript

JavaScript is THE scripting language of the Web.

JavaScript is used in millions of Web pages to add functionality, validate forms, detect browsers, and much more.

The Internet is full of resources for learning JavaScript and can be easily located using search engines.

An example of some tutorials and references:

- <http://www.javascriptkit.com/jsref/>
- <http://www.w3schools.com/js/default.asp>
- <http://www.learn-javascript-tutorial.com/>

Additional Debugging Tips

Since all the TruClient arguments support JavaScript you use the Alert function to display information during script development.

You can also reference any DOM element using regular functions such as location.



To further improve debugging capabilities you can install plug-ins such as DOM Inspector and Firebug that can provide additional information on the application object properties.

General Tips Regarding Successful Interactive Replay

- Do not resize the Firefox browser between record and replay and during replay. This can cause objects to move and interfere with VuGen's ability to locate them.
Note: This is especially important when the Related Objects feature is used, as resizing may change the relative position of the objects.
- Do not switch between applications during interactive replay. Keep Firefox in focus without working in the browser.

General Tips Regarding Successful Recording

- Do not use arrow keys, tab key, the escape key, or the middle mouse key when recording. Try to use their mouse equivalents.
- Do not resize the application during recording.

Resolving Step Timeouts

Steps may timeout due to several reasons:

- Application is responding slowly, possibly under load. This is actually an important test result.
- Step Timeout is incorrect and should be modified via the Step section of the step properties.
- The end event of the step is incorrect and the step is waiting for an event that doesn't occur. The end event should be changed via the Step section of the step properties. See more information regarding End Events in the LoadRunner documentation.

Using Regular Expressions

To use regular expressions, there are two options:

1. Use the '/' notation: Replace the quotation marks of a string with a slash.
For example: /LoadRunner/ is a regular expression that will match any string that contains the word "LoadRunner" in it.
2. If you need to dynamically create a regular expression (e.g. using a parameter), you can use the regular expression constructor and specify the string. For example, the equivalent of the above example is RegExp("LoadRunner").



The full list of supported regular expressions can be found here:

https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/RegExp