Customizing Recording Options

You can customize the clickable elements that can be recorded during scenario recording. The scripts to modify are located in the following directories:

Chrome: <INSTALL>/eclipse/plugins/com.parasoft.xtest.libs.web_<version>/root/browsers/chrome/UserCustomizableOptios.js

Internet Explorer: <INSTALL>/plugins/com.parasoft.xtest.libs.web_<version>/root/browsers/ie/UserCustomizableOptions.js

Variables

SOAtest uses the array variables defined in **UserCustomizableOptions.js** during recording. The following are the variables currently available in this script.

ext.options.clickableAttributes

This variable defines the type of attribute of html elements that SOAtest is looking for when determining if it should record a click on this element. For example, the onclick is used to initiate script execution and is a good candidate in this array.

ext.options.clickableTags

This variable defines the html tags that SOAtest will consider when recording click actions. In an Ajax web application, there are cases where tags such as span or li are clicked that caused certain functionalities to execute on the client side. This variable is used to define such tags.

ext.options.clickableInputTypes

This variable defines the form input types that SOAtest will consider when recording clicks. Types such as text and textarea are not considered clickable by default because the user usually clicks on it just to gain focus and enter text.

ext.options.structuralTags / ext.options.containerTags

These variables work together to limit cases for which user actions are recorded against elements. User actions are not recorded for elements named in the former variable if those elements have at least one child element (either direct or indirect) that is specified in the latter variable.

For example, SOAtest normally records click actions on div elements. However, sometimes a div element might be used not as a UI element; instead, it could be used as a container element that contains a number of other elements (e.g., it could contain a table that contains many other elements). In this case, SOAtest could potentially record a click action against the container div that contains the table—but you might not want a click action recorded here because the div is used as a container element and not as a lower-level UI element. To tell SOAtest that it should not record the click action against the div in cases where the div contains a table element, you would ensure that the `ext.options.structuralTags` array (or equivalent) contains "div" and that the `ext.option.containerTags` array (or equivalent) contains "table".

ext.options.disallowedTags

This variable contains a list of tags that will never be recorded, even if they satisfy other recording criteria.

ext.options.locatorBuildersOrder

This variable defines the order in which SOAtest uses to create the locator. A locator is created to identify the html element on the page that the user action should take place. It is needed during playback to repeat the user action. The order is constructed such that visual attributes in an element are more favorable when creating the locator.

Choose **Parasoft> Preferences> Browser** and enable the **Print debugging information** option to enable debugging information in the console and see which locator builder is used.

Example message

```
Locator -  /descendant::img[@id='changeing'][1]. Used id builder.
```

This tells you what locator from LocatorBuilders.order is actually used (in the above example, `Used id builder` is used). See Browser Settings for more information about this option.

ext.options.preferredAttributesOrder

This field is used by the locator builders 'attributesXPath' and 'attributesXPathWithIndex' from the above ext.optionslocatorBuilders.order. When invoked, this array determines the preferred attributes used. You can add, remove, or reorder attribute names to this array to better customize the locators that get built for elements on your site. Removing an attribute will prevent it from being used by the aforementioned locator builders (which may be important if you know certain attributes on your web page are dynamic and should never be used). Re-ordering the attributes will change the priority of the attributes. The first attributes of the array, like 'id', take precedence to be used in locators over the last elements.

ext.options.displayDialogs

This variable determines whether SOAtest will display alert, confirm, and prompt dialogs during playback. Note that allowing these dialogs to display during playback requires the user close the dialogs some other way (such as with an AutoIT script).

ext.options.xpathLibrary

This variable lets you configure SOAtest to use a specific XPath library for resolving XPath locators used in web scenarios. It applies only to Internet Explorer and Chrome. This option has three possible values: ajaxlt, wgxpath, and default.

- **ajaxlt**: Uses the Google Ajaxlt XPath implementation: http://code.google.com/p/ajax-slt/
- **wgxpath**: Uses the Wicked Good XPath implementation at https://github.com/google/wicked-good-xpath
- **default**: Uses the default implementation (currently, Wicked Good XPath).