

# Browser Playback

## Browser Playback for SOAtest

The Browser Playback tool plays back a user action in a web browser on the current page (for example, clicking on a link or typing).

This topic includes:

- [Browser Playback for SOAtest](#)
- [Creating Browser Playback Tools](#)
- [Configuring Browser Playback](#)
- [Related Tutorials](#)
- [Browser Playback for Virtualize](#)
- [Creating Browser Playback Tools](#)
- [Configuring Browser Playback](#)

## Creating Browser Playback Tools

The Browser Playback tool is the basis for all Web tests recorded through the browser.

- For details on how to create these tests, see [Web Functional Testing - Overview](#) and [Browser Recording and Playback](#).

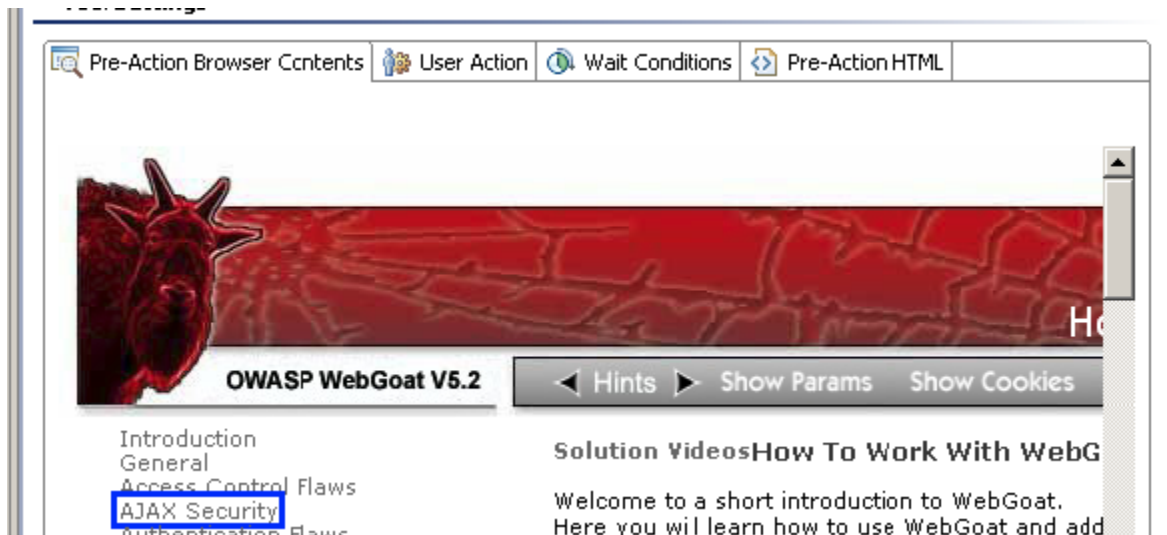
## Configuring Browser Playback

For details on how to configure and customize Browser Playback, see:

- [Modifying User Actions Simulated by a Web Scenario](#)
- [Specifying Wait Conditions](#)
- [Validating or Storing Values](#)
- [Stubbing Test Requests and Responses](#)
- [Customizing Recording Options](#)

The tool's configuration panel has the following tabs:

- **Pre-Action Browser Contents:** A rendered view of what was in the browser before the test executed. For an explanation of the colored borders used to mark notable elements, see [Colored Borders](#).
- **User action:** Allows you to view/modify the user actions simulated by this test. For details, see [Modifying User Actions Simulated by a Web Scenario](#).
- **Wait condition:** Allows you to view/modify the wait conditions for this test. For details, see [Configuring Wait Conditions](#).
- **Pre-Action HTML:** A code-level view of what was in the browser before the test executed. Use the **Browser Containers** drop-down to toggle between the available browser windows and frames.



Note that you can attach an output to an HTTP request response as well as add an output directly attached to the Browser Playback tool. The following tools can be attached as outputs:

- [Browser Contents Viewer](#)
- [Browser Data Bank](#)
- [Browser Validation](#)
- [Extension Tool for Custom Scripting](#)

These tools directly access the final browser contents for the test (that is, what you see in the post-action rendered view). Only one Browser Validation tool and one Browser Data Bank tool can be added as outputs to each Browser Playback tool.

## Related Tutorials

The following tutorial lesson demonstrates how to use this tool:

- [Web Functional Testing](#)

## Browser Playback for Virtualize

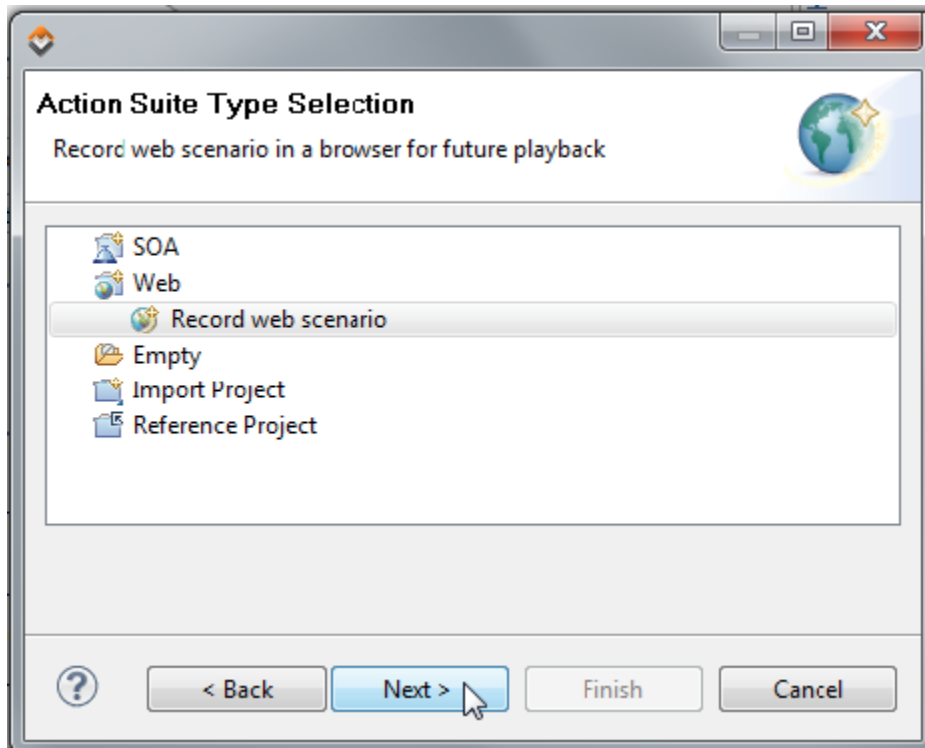
The Browser Playback tool plays back a user action in a web browser on the current page (for example, clicking on a link or typing).

This topic includes:

- [Browser Playback for SOAtest](#)
- [Creating Browser Playback Tools](#)
- [Configuring Browser Playback](#)
- [Related Tutorials](#)
- [Browser Playback for Virtualize](#)
- [Creating Browser Playback Tools](#)
- [Configuring Browser Playback](#)

## Creating Browser Playback Tools

Browser Playback tools are typically added from the Action Suite wizard's Web > Record Web scenario option:



From this wizard, you can click through the actions you want recorded, then Virtualize will create and configure an action suite with the Browser Playback tools necessary to replay the captured actions.

For details, see [Defining Provisioning Actions](#).

## Configuring Browser Playback

The tool's configuration panel has the following tabs:

- **Pre-Action Browser Contents:** A rendered view of what was in the browser before the test executed. For an explanation of the colored borders used to mark notable elements, see [Colored Borders](#).
- **User action:** Allows you to view/modify the user actions simulated by this test. For details, see [Modifying User Actions Simulated by a Web Scenario](#).
- **Wait condition:** Allows you to view/modify the wait conditions for this test. For details, see [Configuring Wait Conditions](#).
- **Pre-Action HTML:** A code-level view of what was in the browser before the test executed. Use the **Browser Containers** drop-down to toggle between the available browser windows and frames.

