

# Configuring Wait Conditions 1

This topic explains how to customize wait conditions for web scenarios in SOAtest and Virtualize.

Sections include:

- [Understanding Wait Conditions](#)
- [Specifying Wait Conditions](#)
- [Configuring the Order of Wait Conditions](#)

## Understanding Wait Conditions

You can customize how long SOAtest and/or Virtualize waits after performing a user action in order to move on to the validations/extractions phase of the current scenario step, and then to the next step in the scenario.

When a user interacts with a web page, the web page responds to whatever the user is doing. For example, when the user clicks a link, the page is reloaded with new content. When a user chooses a menu item, some other part of the page might refresh itself. The user instinctively waits until the page is done updating before continuing further use of the page. In fact, in most cases the user HAS TO wait for the update in order for the page element to be present with which the user is going to interact with next.

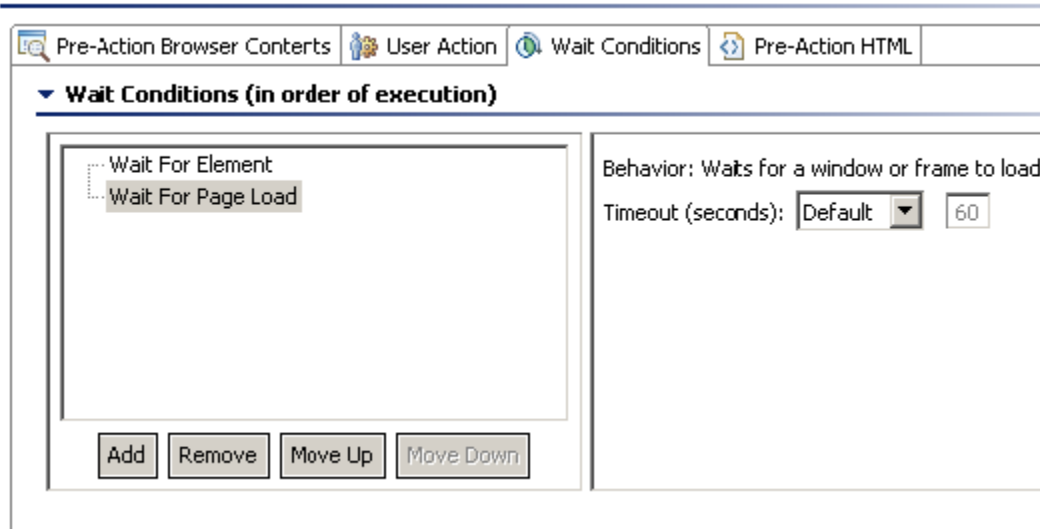
This decision has to be configured when test automation is implemented. The testing tool must wait long enough so that it does not try to continue with the scenario execution process before the page is ready, but at the same time it must also run quickly to achieve one of the benefits of automation—speed.

SOAtest and Virtualize automatically configures the wait conditions while it is recording. For each web scenario step, they automatically add a delay to wait for the element used in the next scenario step. This is done because you cannot have the next web scenario step click on a link unless that link first appears on the page.

If an action causes a new page to load during recording, then SOAtest and/or Virtualize adds a “Wait for Page Load” delay.

If network traffic that does not result in a page load is detected during recording (e.g., an AJAX request), then SOAtest and/or Virtualize adds a “Wait for Time Interval Without HTTP Traffic” delay to give the web application enough time during playback to make the appropriate requests.

However, you may want to manually adjust or modify the wait conditions in order to get the scenarios to perform as desired. In many cases, multiple wait conditions will be used for a single scenario step.



## Specifying Wait Conditions

The wait conditions captured during scenario recording can be viewed and modified in the scenario step's **Wait Conditions** tab. Available wait conditions include:

- **Wait for Page Load:** This wait condition waits until at least one page load has occurred. However, it will wait until all page loads that happen within one second of each other have finished. Once one second has passed without any new page loads starting, the wait is finished. A page load can either mean the entire page is reloading, or it can mean that a single frame is reloading. A Page Load wait condition is added during recording if SOAtest/Virtualize detects that a page load occurs after the particular user action that causes that scenario step to be recorded, and before the user action for the next step that is recorded.

- **Wait for Asynchronous Requests:** This wait condition waits until at least one asynchronous request has been made and a response is received for it. If any other asynchronous requests were begun while the first was in progress, then it waits until all asynchronous requests have completed. For this wait condition, an asynchronous request is defined as a request that is made while a page load is not occurring, and whose response is text-based. An Asynchronous Request wait condition is added to during recording if SOAtest/Virtualize detects that one or more asynchronous requests occur, outside the context of a page load, after the user action that caused that step to be recorded and before the user action for the next step is recorded.

#### Wait for Asynchronous Requests wait conditions are deprecated as of version 9.7

This means that you cannot add new ones, but existing ones will continue to function until you convert the associated scenarios to Selenium format. You can replace these wait conditions with "Wait for Interval without Traffic" wait conditions. Be sure to make the interval long enough that the asynchronous request will have occurred.

- **Wait for Element:** This wait condition waits until a specified page element meets a specific condition.
  - **Page Element:** The page element can be specified in two ways:
    - **Element for next user action:** This option automatically determines which element to wait for by looking at the next step in the scenario and using the element that is configured in that step's User Action tab. If the current step is the last step in a certain scenario, then this wait condition will not perform a wait.
    - **Element specified:** This option allows you to manually choose which element to wait for. You can use **Element Properties**, **XPath**, or **Script** to choose the element.
  - **Condition:** You can set the following element conditions to wait for:
    - **Present:** This condition waits until the element is present on the page. The element may or may not be visible to the user. This is preferred over the next condition, Visible, in cases where the element does not become visible until a user mouses over some other element. This is the default condition used for element waits that are automatically added while recording.
    - **Visible:** This condition waits until the element is visible on the page.
    - **Not visible:** This condition waits until the element is either not present on the page, or is on the page but is not visible.
    - **Editable:** This condition waits until an input element is editable. If this condition is used on an element that is not an input, it will eventually time out because elements that are not inputs are by definition not editable.
    - **Has value:** This condition waits until an element has an attribute with the specified value. The attribute could be any attribute supported by the element, or "text" for the text content of the element.

An Element wait condition (wait for element present) is added during recording as the last wait condition for all tests or actions except those that have a Script Dialog wait condition added to them. Element wait conditions are inactive (meaning that they don't wait for anything) in the following cases:

- There are no steps in the scenario after the current scenario step.
- There are no enabled Browser Playback tools in the scenario after the current scenario step
- The next Browser Playback tool does not use a page element in its user action.
- The next Browser Playback tool is configured to use test or actions suite logic. Element wait conditions are not used if the next step uses test or actions suite logic, since the logic can cause the next step to not be run
- **Wait for Script Dialog:** This wait condition waits until one of the following script dialogs is detected: alert, confirm, or prompt. A Script Dialog wait condition is added during recording as the last wait condition for all test or actions that cause a script dialog to appear.
- **Wait for Specified Time:** This wait condition simply waits for the specified number of milliseconds. This wait condition is not added automatically during recording.
- **Wait For Time Interval Without HTTP Traffic:** This wait condition waits until a specified number of seconds have passed without there being any traffic between the browser and the server. For example, if the specified time is 1 second, it finishes waiting once there has not been any traffic between the browser and the server for 1 second. This wait condition is added during recording only in conjunction with an Asynchronous Request wait condition, in cases where an asynchronous request is detected that causes other non-asynchronous requests to occur.

#### Wait For Time Interval Without HTTP Traffic was the default wait condition Prior to WebKing 6.0.5

This wait condition was available as a Request Wait Time and a UI Wait Time. The individual times were able to be customized manually, but the defaults were 4000 ms for Request Wait Time and 100 ms for UI Wait Time.

In addition to manually adding new wait conditions from this tab, you can also add them automatically from the Browser Contents Viewer tool.

## Adding a Wait Condition from the Browser Contents Viewer

The Browser Contents Viewer allows you to specify a wait condition graphically, from a view of the rendered Web page.

To add a new wait condition from the Browser Contents Viewer tool:

1. Right-click the element for which you want to specify a wait condition.
2. Choose **Add Wait for Element** from the shortcut menu.
3. In the Add Wait Condition dialog, specify the details for the new wait condition, then click **OK**.

The wait condition added will automatically be configured to wait until the clicked-on element is present.

## Identifying Elements Associated with Wait Conditions

The element that is the source of a wait condition will be highlighted with a solid green border in the Browser Contents viewer.

## Configuring the Order of Wait Conditions

The Wait Conditions appear in order of execution in the Wait Executions tab. You can Add, Remove, and change the order of conditions by clicking the appropriate buttons in the **Wait Conditions** tab.

*The order of the wait conditions is important!*

Wait conditions will execute in the order that they appear, regardless of whether any of the other wait conditions succeed or fail. If a wait condition fails (meaning the condition is not satisfied before the timeout for that condition), then an error message is generated. For example, for most web applications page loads typically happen before asynchronous requests are made. Therefore, the wait conditions for a scenario step that has both a page load and asynchronous requests typically should have a Page Load wait condition appearing before an Asynchronous Request wait condition. If the order of the conditions were switched, then the Page Load condition would fail, because the Asynchronous Request wait condition will wait for asynchronous requests which happen after any page loads occur. Then the Page Load wait condition would execute, but since there will be no more page loads it will end up timing out.

Each wait condition, other than the **Wait for Specified Time** condition, has a timeout. If the wait condition is not met within the timeout, an error message is reported so you know to adjust the wait conditions. However, the scenario step will continue even if wait conditions fail. The timeout can be set to use the default timeout that is set in the preferences, or it can be customized for that individual wait condition.