

# CFSelenium 1.5 - With Some Handy Helper Methods

Posted At : June 22, 2011 2:27 PM | Posted By : Bob Silverberg

Related Categories: CFSelenium, Selenium

A new version of [CFSelenium](#) was recently released which includes a few helper methods that I've found useful. First off, I'd like to thank the fine folks on the [Mozilla Web QA](#) team, from whom I borrowed the idea for these methods. I wrote a couple of tests for them in Python, and their Python base page object contains these methods. While I agree that it makes sense to include these methods in a base page object, not everyone using CFSelenium is using the Page Object Model, and as the creator and co-maintainer of CFSelenium I figured I'd just make everyone's life simpler and include them directly in the client library.

The three methods, which are *waitForElementPresent*, *waitForElementVisible*, and *waitForElementNotVisible*, are helpful when testing applications that make use of AJAX, or even just simple JavaScript. When you want to check that a page has responded to a user interaction, and the response is generated by JavaScript, you can often run into timing issues in your Selenium tests. If you issue *aclick* command and then immediately include an assert with a *getText* command in it, it may be that the element that you are attempting to get hasn't been created by the browser yet. The same can be true when elements are shown or hidden. To address this issue the three methods mentioned above can be used, each of which is described in more detail below.

## Arguments

Each of these methods accepts one required and one optional argument. The first argument, *locator*, is just that; the locator of the element that you want to wait for. The second, optional, argument is *timeout*, which is a number, in milliseconds, that you want CFSelenium to wait for the element before it throws an exception. The default value for *timeout* is 30000 (or 30 seconds), and this can be overridden in one of two ways. You can override the default for your entire test case by passing a value into the *waitTimeout* argument of CFSelenium's *init* method. You can also override the default on any call to any of these methods by passing a value into the second argument of the methods, which is *timeout*.

## waitFor Methods

In case it isn't completely obvious, here is what the waitFor methods do:

- *waitForElementPresent* - Waits for the specified element to be added to the DOM. If the element is not present by the end of the timeout interval an exception is thrown.
- *waitForElementVisible* - Waits for the specified element to become visible. If the element is not visible by the end of the timeout interval an exception is thrown. Note that the element must be present in the DOM or the Selenium server will throw an exception immediately.
- *waitForElementNotVisible* - Waits for the specified element to become hidden. If the element is still visible by the end of the timeout interval an exception is thrown.

As always, the latest version is available via the [CFSelenium RIAForge page](#) and via the [CFSelenium GitHub Project](#). The latter also includes a detailed [Readme file](#) which can help you get started using CFSelenium.