1. INSTALL THE SELENIUM BUILDER PLUGIN FOR FIREFOX

2. INSTALL THE CA APPLICATION TEST FOR SELENIUM BUILDER 1.0 PLUGIN

3. CREATE A RECORDING IN FIREFOX USING SELENIUM BUILDER

4. EXPORT THE RECORDING FROM SELENIUM BUILDER

5. IMPORT THE RECORDING TO DEVTEST WORKSTATION

6. EDIT THE RECORDED TEST STEPS IN DEVTEST WORKSTATION

7. INSTALL A WEB BROWSER DRIVER

8. RUN THE TEST IN DEVTEST WORKSTATION

**1. Install the Selenium Builder plugin for Firefox**

* Install the Mozilla Firefox Web-browser (version 43 to 45)
* https://www.mozilla.org/en-US/firefox/new/
* Install the Selenium Builder add-on for Firefox (version 3.0.9)
* https://addons.mozilla.org/en-US/firefox/addon/selenium-builder/
* Run the add-on at least once before installing the CA Application Test for Selenium Builder 1.0 Plugin
* From the Firefox menu select Developer ->Launch Selenium Builder

**2. Install the CA Application Test for Selenium Builder 1.0 Plugin**

* Plugins extend Selenium Builder's functionality (Sauce, GitHub, etc.)
* Go to the <DEVTEST\_HOME>\addons\sebuilder-plugin directory
* Typically C:\Program Files\CA\DevTest\addons\sebuilder-plugin
* Copy the entire lisa directory (and its contents)
* Go to your <Firefox profile>\SeBuilder3\plugins directory
* To find your Firefox Profile Folder: From Firefox, select Help  Troubleshooting Information
* Example:C:\Users\<USERNAME>\AppData\Roaming\Mozilla\Firefox\Profiles\eqpa9ail.default
* Paste the lisa directory into <Firefox profile>\SeBuilder3\plugins
* Verify that the plugin was successfully installed
* From the Firefox menu, select Developer ->Launch Selenium Builder -> Manage Plugins.

**3. Create a Recording in Firefox using Selenium Builder**

* In Firefox, select Developer
* Launch Selenium Builder
* Enter the URL of the web application you want to test in the Start recording at field
* Click Record
* Perform the actions you want to test on the Web application
* When you are done, return to Selenium Builder and click Stop Recording.

**4. Export the Recording from Selenium Builder**

* Click File ->Export ->Save as JSON
* Test cases recorded in the Selenium Builder plugin can be saved in various formats, but must be exported as JSON to be used with DevTest
* Once exported, the test cases can be imported into DevTest Workstation for editing and playback
* Test cases can be replayed in various Web browsers such as Firefox, Chrome, and Internet Explorer

NOTE: Selenium Builder test scripts are exported in JSON-based format for editing in DevTest

Notes about the JSON script Format

* Selenium Builder JSON-based scripts can be played back by a standalone interpreter so you can view and edit them in DevTest
* Selenium 2 JSON scripts contain
  + A list of steps (such as “get” or “clickElement”) which have named parameters
  + Locators which describe how to find a particular DOM element (a link to click, for example) and contain a type (id, name, link text, css selector, or xpath) and value
  + Negation for Assertion and Verification steps to invert their condition
  + Variables to store values referenced in parameters (such as checking that a page's title appears in the page itself)
  + Version information for compatibility between Selenium versions

**5. Import the Recording to DevTest Workstation**

* In DevTest Workstation, create a new test case
  + Right-Click on Tests -> Create New Test Case
* Then add the Selenium test steps you exported previously
* Right-Click in the blank space->Add Step->Selenium->Selenium Import/Export

**6. Edit the Recording in DevTest Workstation**

* Once a recording is imported, click on a test step and then click the Selenium Step tab in the right panel
* The details for the step are shown, including:
  + The JSON script that can be modified in-line
  + The step actions or gestures that can be modified
* Some actions can be edited using the XPath Editor – Click on the Locator link to open the XPath editor and edit the test step
* To add a new step
  + Right-click on a step and select Add Step after <STEP\_NAME>
  + You can drag-and-drop steps to change the order
  + You can import additional recordings and merge steps together into test scripts

**7. Install the Browser Driver**

* Recordings can be replayed in various Web browsers including Firefox, Chrome, Safari, and Internet Explorer
* To replay in a browser other than Firefox:
* Download the Web browser driver
  + Chrome – <https://sites.google.com/a/chromium.org/chromedriver/downloads>
  + IE -http://selenium-release.storage.googleapis.com/index.html?path=3.0
* Add the following properties to the project configuration file
* Key: selenium.browser.type – Value: Chrome (or IE, Safari)
* Key: selenium.chrome.driver.path – Value: <PATH\_TO\_DRIVER>
* Example: C:\lisa-se\chromedriver.exe

**8. Run the Test in DevTest Workstation**

* Save the Test by clicking Save on the toolbar in the top left corner
* Start an Interactive Test Run (ITR) and observe test steps as they are executed.
* Select ITR (gear icon) in the toolbar to open the ITR tab
* Click the Play button to start the test
* Observe the test actions in the Web browser and follow the steps in the ITR tab on the right

8. Run the Test in a Remote Browser

* On the local computer, add the following properties to the project configuration file you use for running test cases on the remote browser and make it active:
* Key: selenium.broswer.type -> Value: Chrome
* Key: selenium.chrome.driver.path -> Value: C:\lisa-se\chromedriver.exe
* Key: selenium.remote.url -> Value: http://<remote\_hostname>:4444/wd/hub

**NOTE:** If you plan to run Selenium Integration tests on multiple browsers (e.g., Chrome and IE), create a separate project configuration file for each browser type (e.g., SE\_Chrome.config and SE\_IE.config). You can then run the tests on multiple browsers by making different configuration files active for each test run

* Download the Selenium Server and save it to the remote computer
* Go to <http://www.seleniumhq.org/download/>
* Locate the Selenium Standalone Server .jar file and download it

selenium-server-standalone-3.0.1.jar

* Verify that the driver for the browser that you want to use is available on the remotecomputer

chromedriver.exe

* Install Java Runtime Environment on the remote computer
* http://www.oracle.com/technetwork/java/javase/downloads/index.html
* Install Chrome browser on the remote computer
* https://www.google.com/chrome/browser/desktop/
* On the remote computer, run the following command from a command prompt:
* java -jar selenium-server-standalone-3.0.1.jar -role hub
* On the remote computer, run the following command from a new command prompt:
* java -jar selenium-server-standalone-3.0.1.jar -role node –hub http://localhost:4444/grid/register
* Run the test and view the test being run on the remote computer in the Chrome browser
* View the test results on the local Workstation