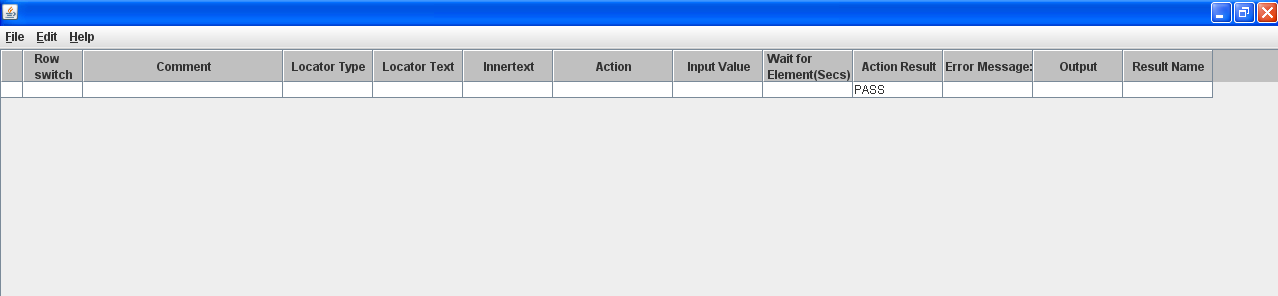
# Writing Test Scripts

Once you have successfully installed the Framework you are ready to start writing test scripts.

Open up the HTML EDITOR:

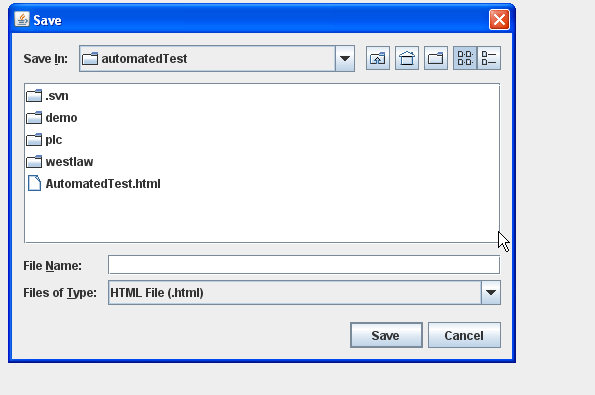
C:\Projects\testSeleniumFramework\framework\HtmlEditor2\dist\HtmlEditor2.jar



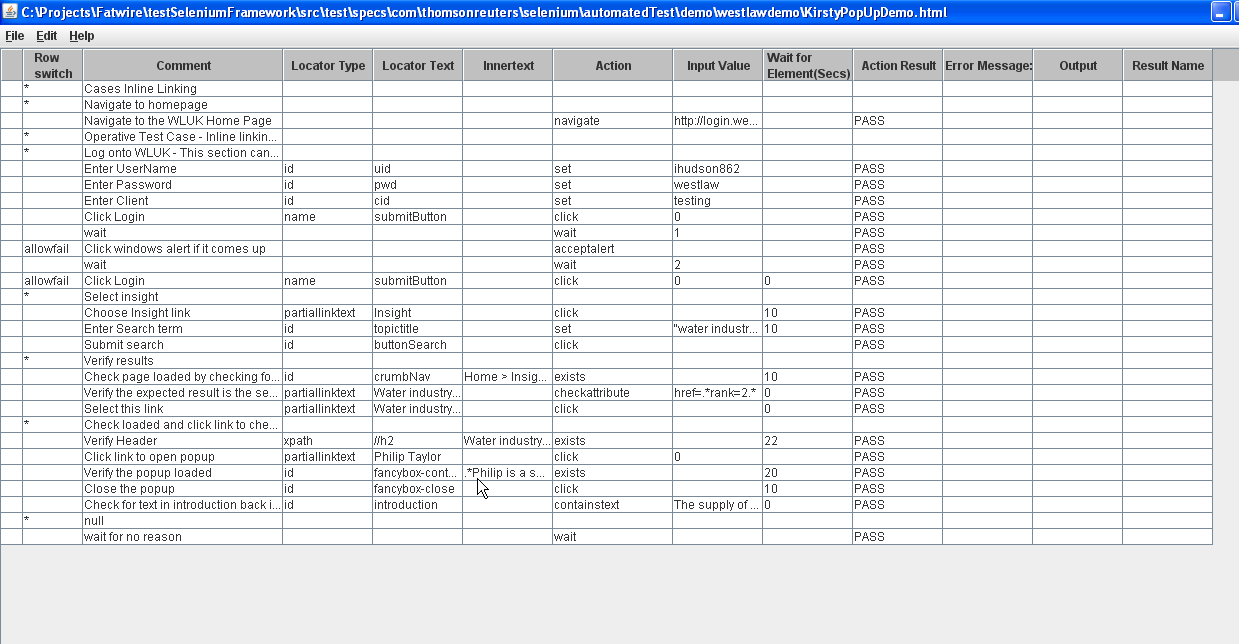
You can add rows, delete rows, insert rows above and below etc. Simply look in the “Edit” menu or right mouse click on an existing row:



You should save your script regularly while you are writing it. Click File>Save (or Save As if you have copied an existing test as a template) and save to the appropriate folder under automatedTest

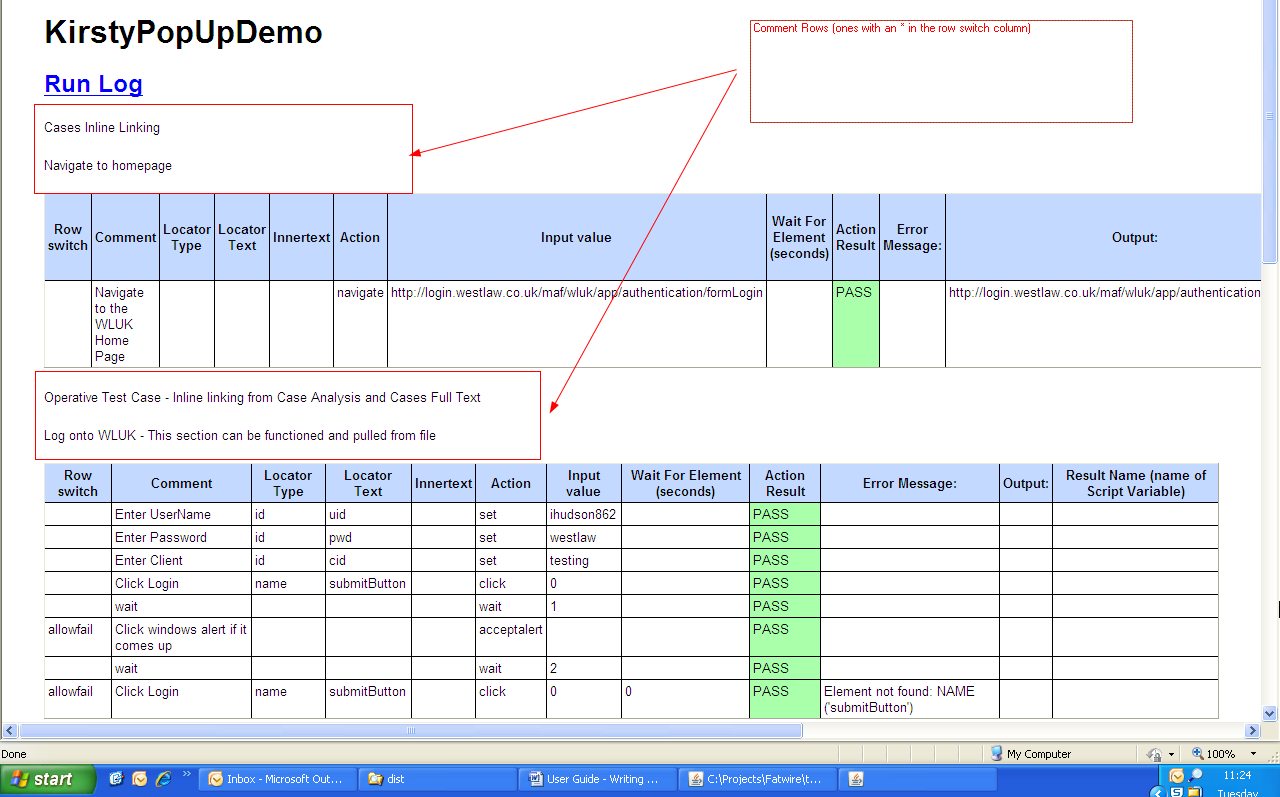


## What to enter in each column



### Row Switch

|  |  |
| --- | --- |
| Leave blank | Normal test row to be executed |
| \* | This is a comment row and you should only populate the comment column for this row |
| allowfail | You may need to use this if you need to cater for something which may or may not happen but you don’t want the test to fail. E.g. a windows alert message may or may not pop up when logging on to some websites so you want to click it if it comes up but it it does not come up you don’t want the test to fail. |
| // | This will “comment out” the row – it will not get executed at runtime. This is mainly used while writing tests to comment out lines temporarily. |
| Enter a formula | Advanced use of framework. If you enter a formula which evaluates to the boolean value false, the row will not execute. This is to allow conditional processing within a test script. |



### Comment

Please enter a comment for each row in your script to ensure it is clear what the test is trying to do.

### Locator Type, Locator Text, Innertext

These three columns are used to locate the element on the webpage. Please refer to the guide on locating elements Object Identification\_Selenium V.1.docx

You only need to use these columns if you are performing an action which is acting on an element E.g. click, set, checktext etc.

### Action, Input Value

You must fill out an action for every step which is not a “comment row” (i.e. row switch is \*).

You fill out the input value depending on the action.

Please refer to the documentation on Actions, Actions.xlsx.

### Wait

For any action which which needs to locate an element on a page, or switch to a new window etc there is a default timeout wait which can be overridden as follows:

|  |  |
| --- | --- |
| Leave blank | To use the default timeout wait |
| 0 | To reduce the timeout wait to zero. This is for when you are at a point in the test where you know the page has loaded so the test does not need to wait for the element to appear. Can be used to speed up tests especially if they fail – as otherwise the test will run slowly waiting each step for an element. |
| Any integer value | This will increase the default timeout wait by the amount specified. For example if you enter a “5” and the default timeout is 10 – the timeout wait for that step will be 15 seconds. You may need to do this if the step you are on is after an action which causes the page to reload (e.g. a click) and for some reason it is always really slow to load that page. |

When writing tests, remember that if you perform an action that causes a page load, the next step must be performed on a new object on that page to ensure you know the page loaded successfully. It is good practice to enter 0 into the Wait column when you are on a step where you know the page has loaded.

However it is good enough just to leave this column blank and only adjust if you are getting timing issues with your test.

### Action Result, Error Message, Output

These three columns should be left as they are. They are part of the test results only.

### Result Name

Some actions return output – which will appear in the Output column of the test results. E.g. gettext returns the text of an element, navigate returns the url you navigated to etc. If you want to store this output for later use in your test script, enter the name of a script variable in here. The name must begin with $ and the next character must be a letter (not a number). You can choose any name but giving it a meaningful name will help your test script is easier to read. E.g. $extracttext, $ukurl etc.

You can use the script variable later on in the test. E.g. to check some element has the text you extracted earlier in the test. See the guide How to Enter Formulae in Your Script Using javascript.docx for more information.

Script variables are case-sensitive, so if you name your variable $extract, then to refer to it later you must use $extract. $Extract will not work.

Therefore it is recommended to adopt a standard when naming these variables. E.g. all lowercase.

# Running Test Scripts

First of all you must ensure the properties file (C:\Projects\testSeleniumFramework\src\test\resources\TestConfig.properties) is set up correctly. The three top properties are the ones you need to check/change.

(NOTE: TestConfig.properties will always be locally updated so you should never commit this to your resource repository unless you are adding a new property. In which case, ensure you have the latest update, add the new property, then email all the testers to take the latest update and then re-apply their local changes.)

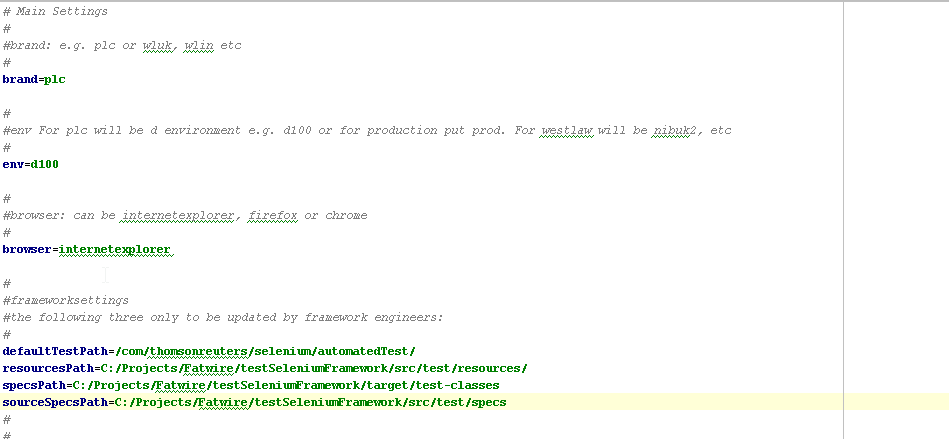
So going back to the point, the three properties you need to update/check are what you want are:

brand

env

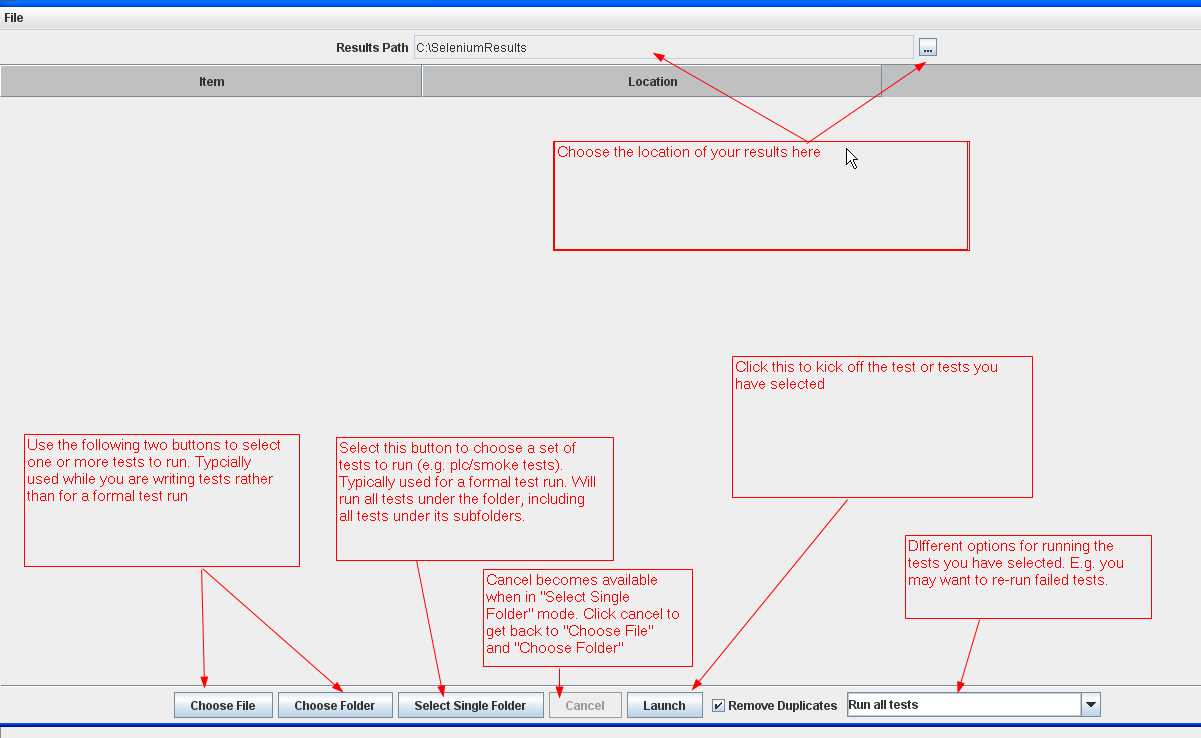
browser

These will determine which environment your test run will run on (brand.env) and which browser your test run will run on:



To run the test scripts use the FrontEnd tool:

C:\Projects\testSeleniumFramework\framework\FrontEnd\dist\FrontEnd.jar



## To stop a test or test run

To stop a test simply close down the test browser window. That test will then terminate.

To stop an entire test run, first close down the test browser window of the currently running test. Then close down the command window which opened when you kicked off the tests:

