

# HTML Automation Framework

Creation Date: 2/12/2022 10:55:00 PM

## Table Of Contents

Introduction .....	4
Maintenance .....	5
Chromedriver.exe .....	6
Msdriver.exe .....	6
Geckodriver.exe .....	6
Releases · mozilla/geckodriver · GitHub (download for win32) .....	6
Type Of Web Elements .....	7
HTML Action Set.....	8
InnerText or Value Attribute Validation Functions.....	14
Text Validation Of Other Attributes.....	18
Create a Runtime Object.....	23
Browser Functions .....	24
Launch Web Page.....	26
Assert Function .....	27
Browser Snapshot .....	28
Windows SendKeys function.....	29
HTML Page Source .....	30
Web Alert Handling.....	31
Page Navigation Functions.....	32
Quitting Web Pages .....	33
Change Page.....	34
Object Check .....	35
Page Load Time Wait .....	36
Explicit Wait Time .....	37
Change Frame .....	38
Bulk Object Check .....	39
Execute JavaScript.....	40
Smart Object Identification.....	41
Dynamic Parameterization.....	42
Table Value Parameters: .....	42
Traversing All Rows Of Parameter Sheet .....	44
Condition Value.....	45

Log File: .....	47
Pattern: .....	47
Random List: .....	47
Previous Block: .....	47
Pass Parameter And Code Based String Validations.....	48
Web Code.....	52
Script After Every Step .....	55

## Introduction

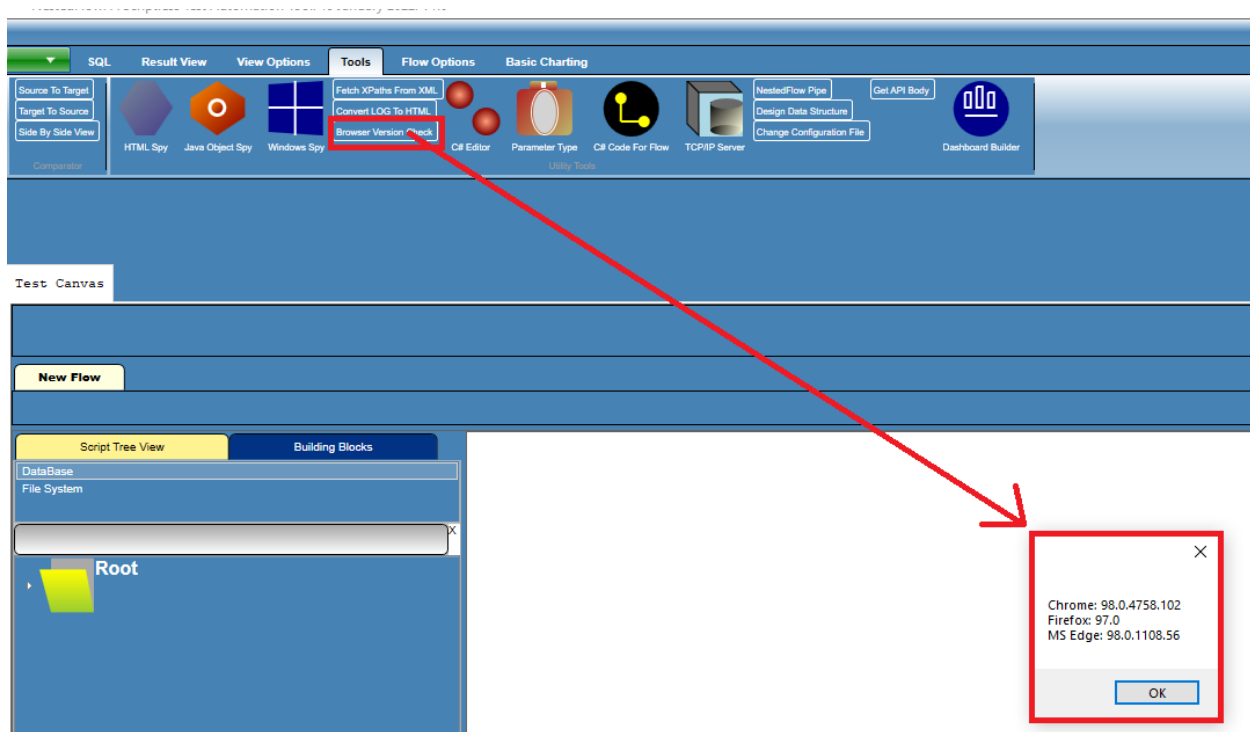
This document explains the HTML automation framework both coded and scripless which can be achieved using **Nested Flow** Automation Tool. It will cover

- Type Of Web elements
- Action Set
- Description of actions
- Browser Action Descriptions
- Other Automation functions

## Maintenance

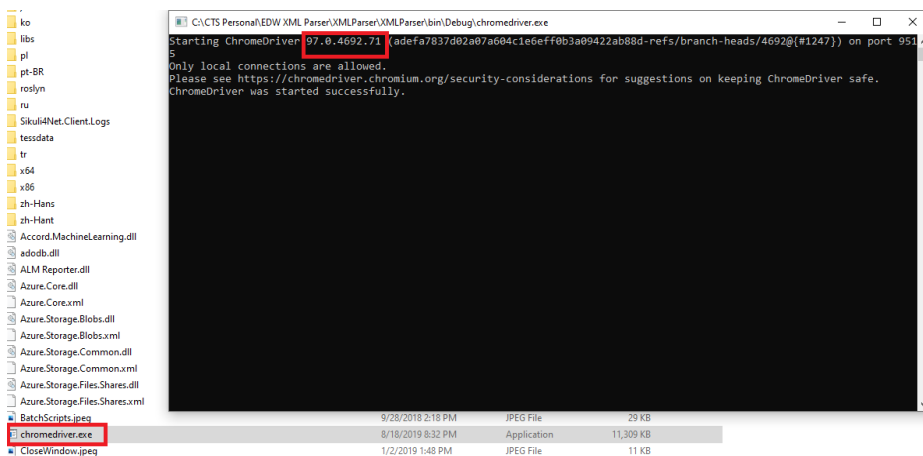
Tool comes with specific **chromedriver.exe**, **msedgedriver.exe**, and **geckodriver.exe** for supporting Chrome, MS Edge and FireFox browsers respectively. But browser updates (upgrades in particular) will cause tool to fail in identifying browsers. This section explains how to fix browser version issues.

### Navigate to Tools → Browser Version Check



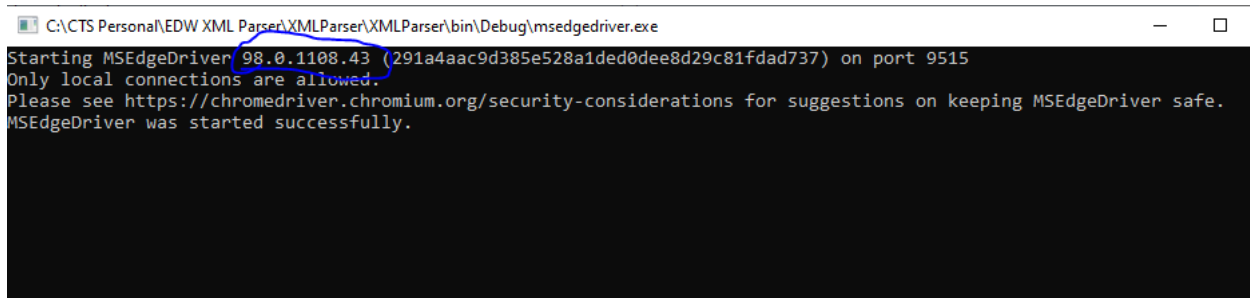
It will display the Chrome, Firefox and MS Edge versions. (Same can be achieved by opening each browser, navigating to Settings and then to About option)

In the Tool installation folder click on chromedriver.exe



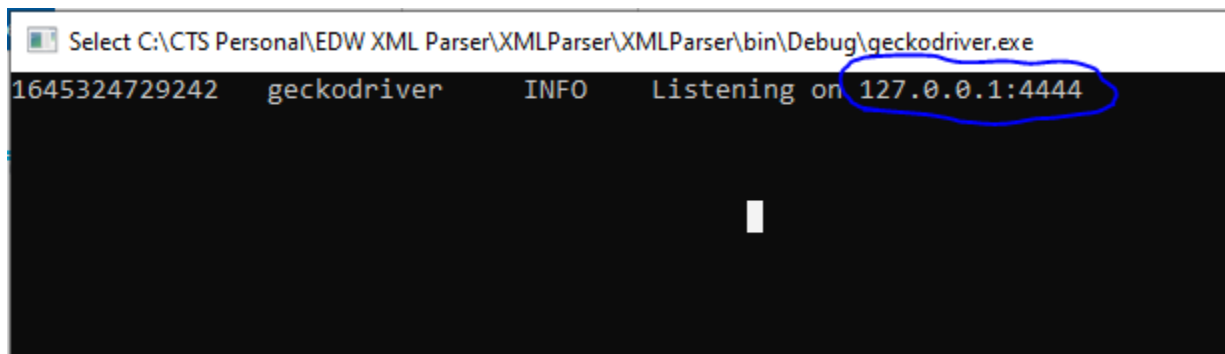
[ChromeDriver - WebDriver for Chrome - Downloads \(chromium.org\)](https://chromedriver.chromium.org/downloads) tells which chromedriver version matches chrome version

Click on msedgedriver.exe and geckodriver.exe as well



```
C:\CTS Personal\EDW XML Parser\XMLParser\XMLParser\bin\Debug\msedgedriver.exe
Starting MSEdgeDriver 98.0.1108.43 (291a4aac9d385e528a1ded0dee8d29c81fdad737) on port 9515
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping MSEdgeDriver safe.
MSEdgeDriver was started successfully.
```

Msedgedriver.exe version and browser version usually matches



```
Select C:\CTS Personal\EDW XML Parser\XMLParser\XMLParser\bin\Debug\geckodriver.exe
1645324729242  geckodriver  INFO  Listening on 127.0.0.1:4444
```

If one of the browsers do not work, download relevant drivers from below sites

Chromedriver.exe

[ChromeDriver - WebDriver for Chrome - Downloads \(chromium.org\)](https://chromedriver.chromium.org/downloads) (download for x86)






Msdriver.exe

[Microsoft Edge Driver - Microsoft Edge Developer](#) (download for x86)

Geckodriver.exe

[Releases · mozilla/geckodriver · GitHub](#) (download for win32)

## Type Of Web Elements





Object Type	Shape	Description
Web Element		HTML Web element with properties and actions as returned by object spy or created from component list
Last selected Web Element		Current Object which is selected during the last step where web object is used
Current Active Object On Page		Currently active object on page where focus exists
Smart Object		Smart intelligent AI based object which is still in beta state. Currently only few functions are supported here
Run time object		Dynamic object created at runtime

All the object types use same action sets which are explained in the next section.

## HTML Action Set

#	Action Name	Attribute	Description
1.	Choose Value	Text <b>string</b>	This is applicable to a Weblist wherein we choose one of the values using option Text
2.	Clear		Clears the contents of the text box
3.	Clear CheckBox		Uncheck a checkbox if selected.
4.	Click		Click on a web element
5.	Click And Hold		Click and hold on a web element
6.	Click By Offset	X offset, Y Offset (int, int)	Click after a X and Y offset from web element
7.	Click In Table	Column Number~Text Int~string	The web element in this table must always be a web table with a proper (number of columns same in all rows). Searches a link or button in a web table in each row for a given column number and clicks on it if found
8.	Do Nothing		Does Nothing
9.	Double Click		Double click on a web element
10.	Drag And Drop	XPath of target element	Drags a web element on another
11.	DSelect All		Unselects all options in web list
12.	DSelect Text	Text string	Unselects an option text from web list using visible text
13.	DSelect Value	string	Unselects an option text from web list using actual value
14.	Element Snapshot		Takes snapshot of a web element. May not work if frames are involved
15.	EnterText		Enters value to Text Box (web input)



16.	Focus		Focusses on a web element
17.	Get CSS Value		Get CSS identifier of the object (useful if it is dynamic object)
18.	Get Item Xpath		Get item Xpath (useful if it is dynamic object)
19.	Get List Of Vlues		Returns single column table with LOV item  text. Use  to save table and utilize in flow
20.	Get Property Value	HTML Attribute <b>string</b>	Gets a web element attribute
21.	GetTable		Returns all rows and column values of a proper web table. Use   to save table and utilize in flow
22.	Get TagName		Gets web element tag name
23.	GetText		Returns item text
24.	GetText And Validate	Text to validate <b>string</b>	Returns item text and validates with another provided text. PASS is the returned value if pass. FAIL if not
25.	GetText And Validate (Contains)	Text to validate <b>string</b>	Returns item text and validates if another provided text is contained by it. PASS is the returned value if pass. FAIL if not
26.	GetText And Validate (Starts With)	Text to validate <b>string</b>	Returns item text and validates if it starts with another provided text. PASS is the returned value if pass. FAIL if not
27.	GetText And Validate (Ends With)	Text to validate <b>string</b>	Returns item text and validates if it ends with another provided text. PASS is the returned value if pass. FAIL if not
28.	Highlight	Optional color name <b>string</b>	Highlights an object

29.	Is Displayed		PASS if web element is displayed. FAIL if not
30.	Is Enabled		PASS if web element is enabled. FAIL if not
31.	Is Exists		PASS if web element is existing. FAIL if not
32.	Is Selected		PASS if web element is selected. FAIL if not
33.	JavaScript: Choose Value	Text <b>string</b>	Works exactly like <a href="#">Choose Value</a> but through javascript
34.	JavaScript: Clear1		Works exactly like <a href="#">Clear</a> but through javascript
35.	JavaScript: Click		Works exactly like <a href="#">Click</a> but through javascript
36.	JavaScript: EnterText	<b>string</b>	Works exactly like <a href="#">EnterText</a> but through javascript
37.	JavaScript: Hover		Invokes onmouseover function of the object if exists
38.	KeyDown	<b>string</b> Possible values: "ADD" ALT ARROWDOWN ARROWLEFT ARROWRIGHT ARROWUP BACKSPACE CANCEL CLEAR COMMAND CONTROL DECIMAL DELETE DIVIDE DOWN END ENTER EQUALS ESCAPE F1 F10 F11 F12 F2	Performs Keydown on the object like pressing down SHIFT, ALT, ENTER etc

		F3 F4 F5 F6 F7 F8 F9 HELP HOME INSERT LEFT LEFTALT LEFTCONTROL LEFTSHIFT META MULTIPLY NULL NUMPAD0 NUMPAD1 NUMPAD2 NUMPAD3 NUMPAD4 NUMPAD5 NUMPAD6 NUMPAD7 NUMPAD8 NUMPAD9 PAGE_DOWN PAGE_UP PAUSE RETURN RIGHT SEMICOLON SEPARATOR SHIFT SPACE SUBTRACT TAB UP	
39.	KeyUp	<b>string</b> possible options are same as KeyDown explained above	Releases the specified key from web element.
40.	LOV Contains	<b>string</b>	Returns PASS if weblist has the specified value FAIL if not
41.	Move Mouse To Element		<b>DO NOT USE</b>

			Moves mouse to the web element (doesn't work properly)
42.	Move To Element		Move to web element
43.	Move To Frame		Can be performed only if the object is a frame. Moves to the frame
44.	Move To Last Window		Moves to the last window opened by the session
45.	Page Title		Gets web page title (doesn't matter which web element it is)
46.	Page URL		Gets web page URL (doesn't matter which web element it is)
47.	Right Click		Performs Right (Context) click on the web element
48.	Radio Group By Index	<b>int</b>	Click on the nth element (0 based index) on Radio group ( <b>function need not be used as the object spy returns correct XPath</b> )
49.	Radio Group By Value	<b>string</b>	Click on the element with entered text on Radio group ( <b>function need not be used as the object spy returns correct XPath</b> )
50.	Release Mouse		Release mouse on the web element
51.	Scroll To Object Y		Scrolls by amount of object Y position (vertical)
52.	Scroll To Object X		Scrolls by amount of object X position (horizontal)
53.	Scroll To Object		Scrolls by object X and Y positions (both vertical and horizontal)
54.	Select From List		Selects objects of tag "li" from listbox
55.	Select By ID	<b>int</b>	Chooses nth (0 based index) element from the web list

56.	Select By Value	<b>string</b> Item to be chosen	Chooses element from web list based on element value (not actual text)
57.	SendKeys	<b>string</b> possible values are same as <a href="#">KeyDown</a>	Sends a key once to the web element
58.	Slide	<b>int</b> (0-100) slider value	Amount by which to slider should be moved (web element must be a slider)
59.	Submit		Performs web submit action (rarely used)
60.	Switch Window	<b>int</b> or <b>string</b> int → 0 Switches to first window int → -2 Switches to last window int → +ve number : Moves to nth window open (0 based index) string → Moves to window with that title	This is a browser function but can be performed in element level as well
61.	Wait Till Clickable		Wait till web element is clickable
62.	Wait Till Displayed		Wait till web element is displayed
63.	Wait Till Exists	<b>Int</b> seconds	Wait for the web element to be existing till n seconds
64.	Wait Till Selectable		Wait Till web element is selectable
65.	Set Attribute	<b>string,string</b> attribute,value	Set weblement attribute with a given value

## InnerText or Value Attribute Validation Functions

The Nested Flow automation framework provides handy data validation functionality which can be used to quickly validate a textbox or link values without the need of writing C# functions in the framework.

Below are the functions which can be used to achieve that.

**Note: The logic looks for innerText and if not found looks for value. For LOVs it always goes for value**

#	Action Name	Attribute	Description
1.	GetText And Validate	Text to validate <b>string</b>	Returns item text and validates with another provided text. PASS is the returned value if pass. FAIL if not
2.	GetText And Validate (Contains)	Text to validate <b>string</b>	Returns item text and validates if another provided text is contained by it. PASS is the returned value if pass. FAIL if not
3.	GetText And Validate (Starts With)	Text to validate <b>string</b>	Returns item text and validates if it starts with another provided text. PASS is the returned value if pass. FAIL if not
4.	GetText And Validate (Ends With)	Text to validate <b>string</b>	Returns item text and validates if it ends with another provided text. PASS is the returned value if pass. FAIL if not
5.	GetText And Validate (Credit Card)		Returns <b>PASS</b> if the web item innerText or value is a valid credit card format or not else returns <b>FAIL</b>
6.	GetText And Validate (Is Alphanumeric)		Returns <b>PASS</b> if the web item innerText or value is alphanumeric (spaces included) else returns <b>FAIL</b>
7.	GetText And Validate (Is Blank)		Returns <b>PASS</b> if the web item innerText or value is blank else returns <b>FAIL</b>
8.	GetText And Validate (Is Not Blank)		Returns <b>PASS</b> if the web item innerText or value is not blank else returns <b>FAIL</b>

9.	GetText And Validate (Is Blank Or Spaces)		Returns <b>PASS</b> if the web item innerText or value is blank or white spaces else returns <b>FAIL</b>
10.	GetText And Validate (Is Not Blank)		Returns <b>PASS</b> if the web item innerText or value is not blank or white spaces else returns <b>FAIL</b>
11.	GetText And Validate (Is Decimal)		Returns <b>PASS</b> if the web item innerText or value is a valid decimal number (e.g: 100.67 or 1004 or 100,000) else returns <b>FAIL</b>
12.	GetText And Validate (Is Hex)		Returns <b>PASS</b> if the web item innerText or value is a valid hexadecimal number else returns <b>FAIL</b>
13.	GetText And Validate (Is In List)	Comma separated list <b>String,string.....</b>	Returns <b>PASS</b> if the web item innerText or value is one of the values in the list provided else returns <b>FAIL</b>
14.	GetText And Validate (Is IP)		Returns <b>PASS</b> if the if the web item innerText or value is a valid IP format else returns <b>FAIL</b>
15.	GetText And Validate (Is URL)		Returns <b>PASS</b> if the if the web item innerText or value is a valid URL format else returns <b>FAIL</b>
16.	GetText And Validate (Is InitCase)		Returns <b>PASS</b> if the if the web item innerText or value is having first letter uppercase and all others as lower case format else returns <b>FAIL</b>
17.	GetText And Validate (Is Numeric)		Returns <b>PASS</b> if the if the web item innerText or value is having all characters as digits else returns <b>FAIL</b>
18.	GetText And Validate (Is Letters)		Returns <b>PASS</b> if the if the web item innerText or value is having all characters as letters

			(upper or lower case) else returns <b>FAIL</b>
19.	GetText And Validate (Is Lower)		Returns <b>PASS</b> if the web item innerText or value is having all characters as lower case else returns <b>FAIL</b>
20.	GetText And Validate (Is Pattern)	Pattern <b>string</b> Pattern can have: N → digits n → digits C → upper case letters c → lower case letters / → Escape character /n = n /N = N // = / /c = c /C = C Sample pattern: Cc/cnnn-nn-nnnn Valid result: Azc234-78-9087	Returns <b>PASS</b> if the web item innerText or value is in adherence to the format value provided else returns <b>FAIL</b>
21.	GetText And Validate (Is Regular Expression)	Regex expression <b>string</b> Example: ^[a-zA-Z]+\$ If string contains only lower and uppercase alphabets	Returns <b>PASS</b> if the web item innerText or value is in adherence to the regular expression value provided else returns <b>FAIL</b>
22.	GetText And Validate (Is SSN)		Returns <b>PASS</b> if the web item innerText or value is a valid SSN else returns <b>FAIL</b>
23.	GetText And Validate (Is Upper)		Returns <b>PASS</b> if the web item innerText or value is having all characters as Upper case else returns <b>FAIL</b>
24.	GetText And Validate (Is Single Line)		Returns <b>PASS</b> if the web item innerText or value is not having new line character else returns <b>FAIL</b>
25.	GetText And Validate (Size)	Length <b>int</b>	Returns <b>PASS</b> if the web item innerText or value length is equal to the



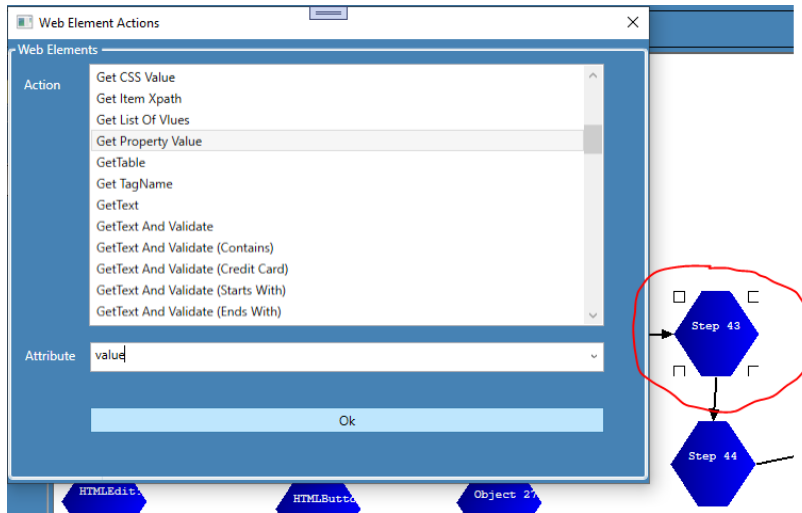
			value provided else returns <b>FAIL</b>
26.	GetText And Validate (Size Between)	Length start, Length End <b>int,int</b>	Returns <b>PASS</b> if the web item innerText or value length is in between the values provided (including the values) else returns <b>FAIL</b>
27.	GetText And Validate (Size Less)	Length <b>int</b>	Returns <b>PASS</b> if the web item innerText or value length is less than the value provided else returns <b>FAIL</b>
28.	GetText And Validate (Size More)	Length <b>int</b>	Returns <b>PASS</b> if the web item innerText or value length is more than the value provided else returns <b>FAIL</b>
29.	GetText And Validate (US Zip Code)		Returns <b>PASS</b> if the web item innerText or value is in valid US Zip code format else returns <b>FAIL</b>
30.	GetText And Validate (Canada Zip Code)		Returns <b>PASS</b> if the web item innerText or value is in valid Canada Zip code format else returns <b>FAIL</b>
31.	GetText And Validate (Is Email)		Returns <b>PASS</b> if the web item innerText or value is in valid email format else returns <b>FAIL</b>

## Text Validation Of Other Attributes

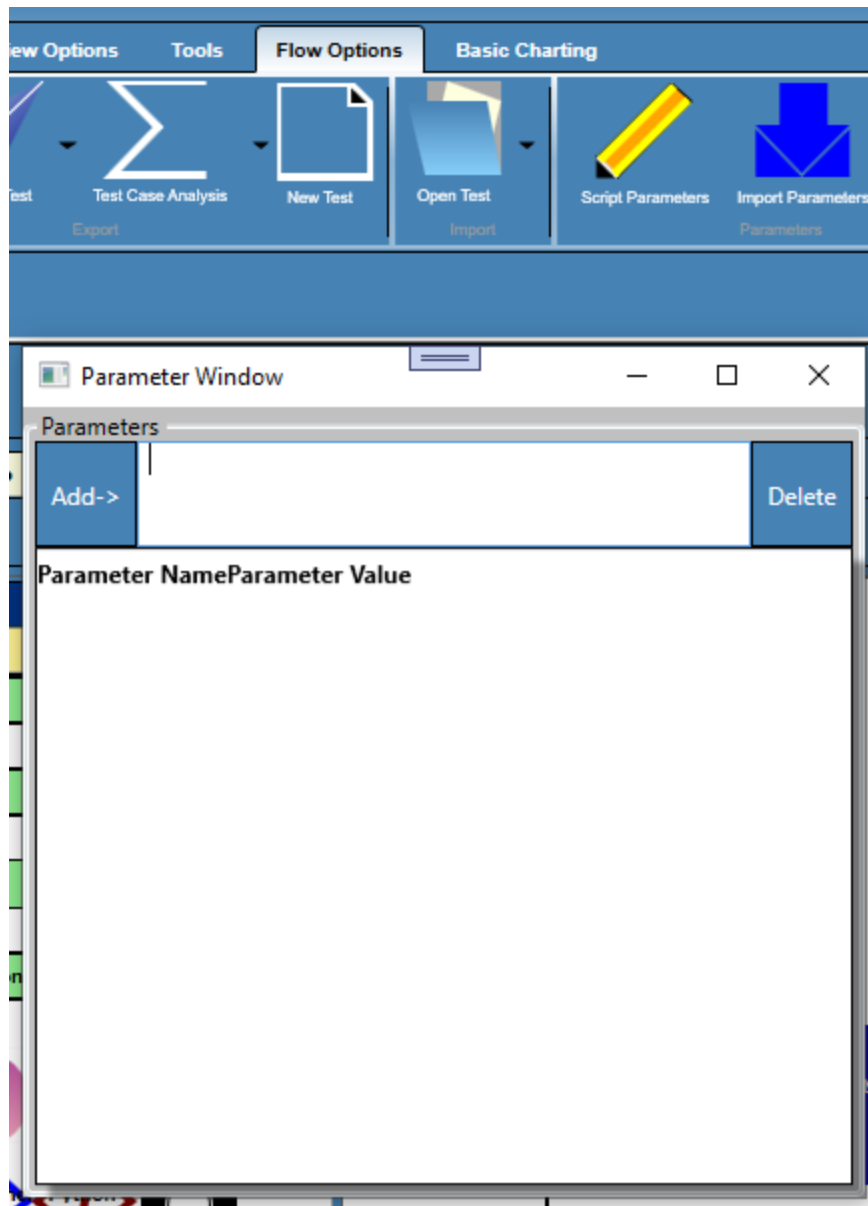
Sometimes it may be required to validate other attributes which are not innerText or Value. In such cases, same string validation functionality explained above are available in string functions.

In this section we will see how to use them.

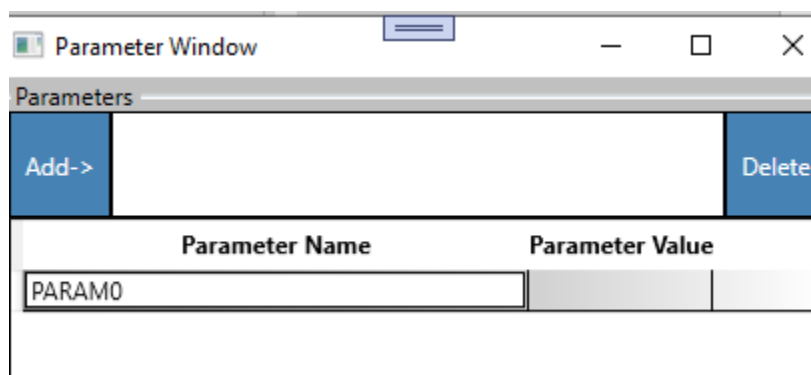
On a Text field get value Property and click on **Ok**



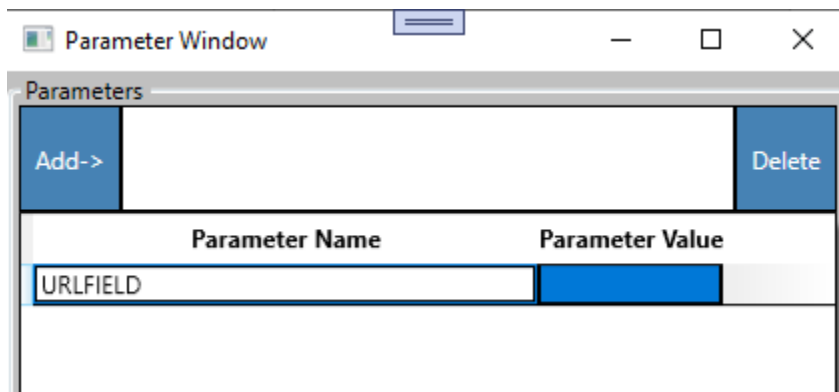
Click on **script Parameters** on the ribbon menu



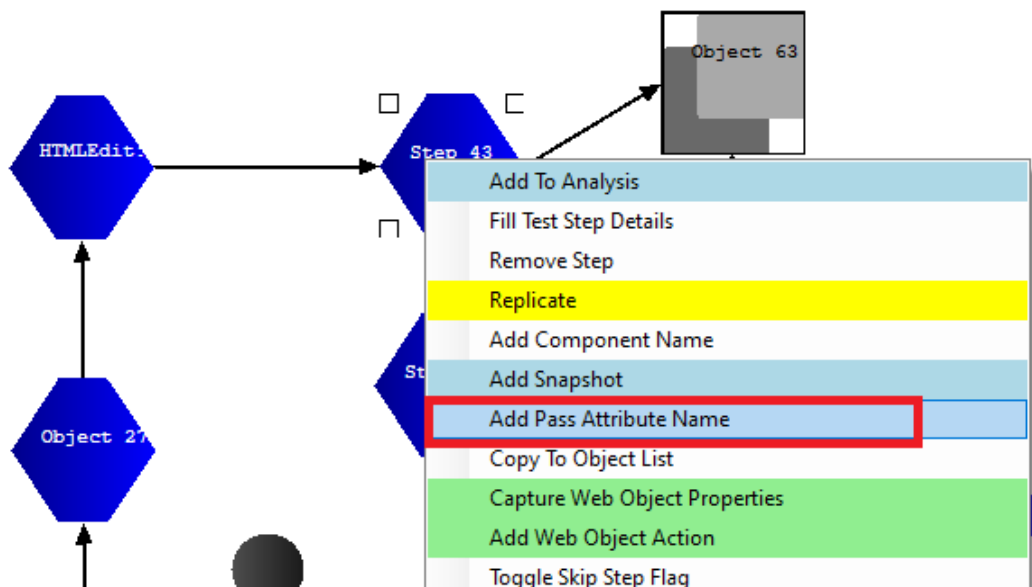
Add a white space in the text box. Click on **Add->** button. It will create a default parameter



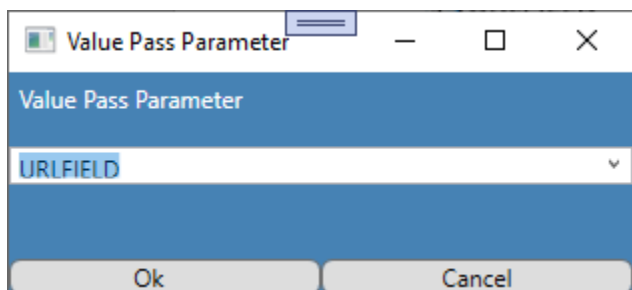
Change the name if you want and close the Parameter Window



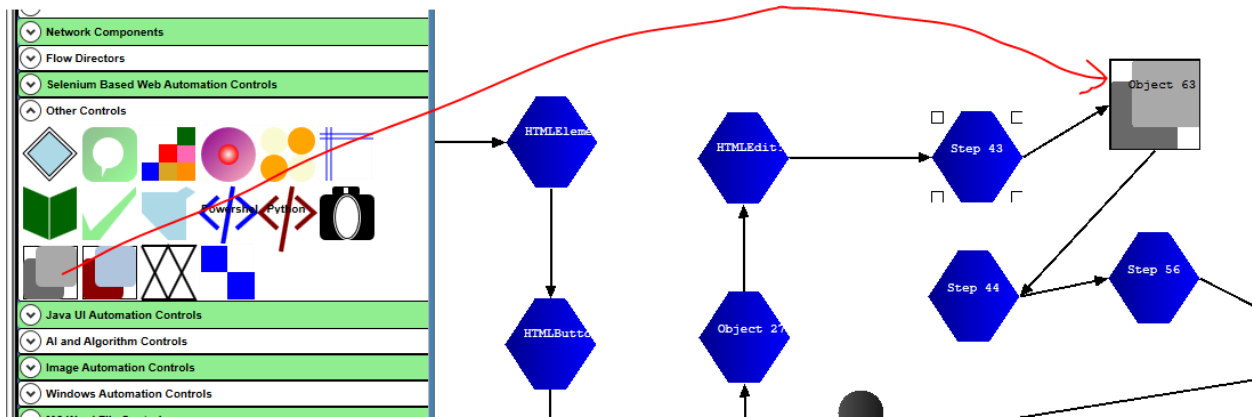
Right Click on the web element for which we are capturing the value property and click **Add Pass Attribute** menu



Choose the parameter you just created and click on **Ok** button.



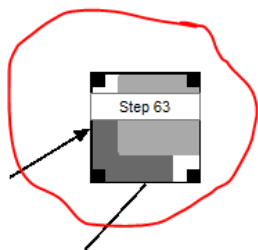
Create next step as generic string function which is in the **Other Controls**



Right Click on the Generic string function component and click on **Add String Function Options**

Choose parameter, Function and Pattern (in case of current function there is no Pattern) but if the selected function is "GetText And Validate (Contains)" the text to be checked will come in Pattern box above (It is equivalent to Attribute Value in above section)

If Mandatory Columns in property windows is entered as **FAIL** and if validation fails then the script is terminated.

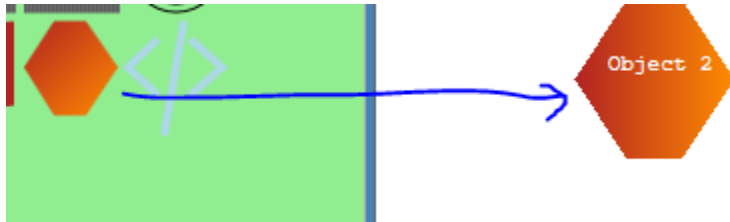


Object Type	Step
Object Sub Type	STRINGFUNCTIONS
Object Name	Step 63
Attributes XML	<NewDataSet> <Table1> <PARAMETER_x0020_NAME <FUNCTION_x0020_NAME <CHARACTER /> <LENGTH /> <START /> <CONTAINS /> </Table1> </NewDataSet>
Maximum Iterations	1
Mandatory Columns	FAIL

## Create a Runtime Object

Step 1:

Drag custom object to test case builder panel



Step 2:

Right Click and choose **Create Custom Object**

A screenshot of a dialog box titled 'Create Web Object'. The dialog has a blue header bar with the title. Below the header, there is a white input field. Underneath this field is a section labeled 'Sub Type' with a blue header and a white input field. At the bottom of the dialog, there are two buttons: 'Ok' and 'Cancel'.

Step 3:

Choose Sub type from LOV (XPath, Link Text, id or name) and enter Value on top field

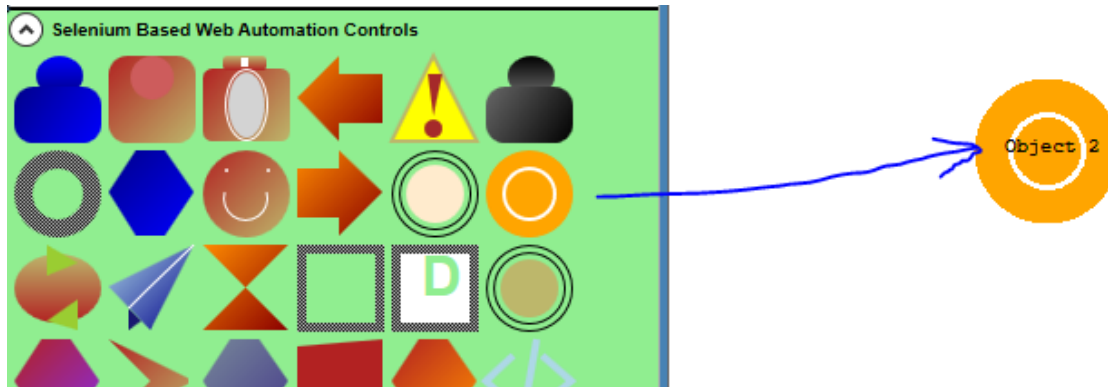
A screenshot of the 'Create Web Object' dialog box, showing it after data entry. The top white input field now contains the XPath expression: `//*[@id="leftmenuinnerinner"]/a[41]`. The 'Sub Type' section is expanded, showing a blue header and a white input field containing the text 'Xpath'. The 'Ok' and 'Cancel' buttons remain at the bottom.

Step 4:

Click on **OK** button

## Browser Functions

The Nested Flow framework provides bunch of Browser functions through the below component:



Right click on the control and choose **Add Browser Operation Data**



Note: All the web functions follow the launch component

Below are the Browser functions supported by the tool

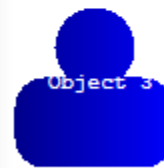
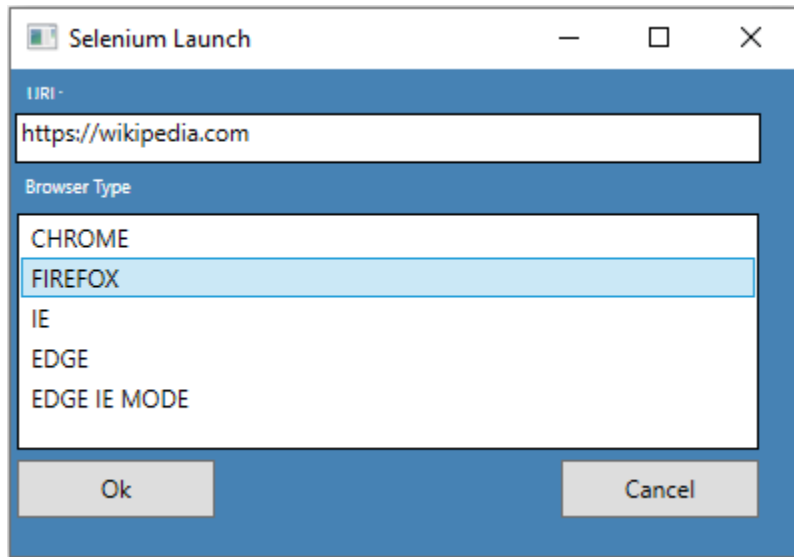
#	Action Name	Attribute	Description
1.	Browser Snapshot	Image Path <b>string</b>	Takes browser snapshot and saves the image to specified path
2.	Browser Alert	Alert Text <b>string</b>	Creates an Alert pop up with the specified message and closes after 2 seconds
3.	Browser Title		Returns browser title
4.	Browser Type		Returns browser type
5.	Browser URL		Returns browser URL
6.	Browser Scroll	Vertical scroll amount <b>int</b>	Scrolls the web page vertically by specified number of pixels
7.	Maximize		Maximizes the web page
8.	Minimize		Minimizes the web page
9.	Normal Size		Brings web page to default size
10.	Close Active Window		Closes the active window
11.	Navigate To	<b>string</b> URL	Navigate to specified URL
12.	Click At X,Y	<b>int, int</b> x,y	Click on web page at specified X,Y



13.	Page Source	<b>string</b> Path	Saves the page source HTML at specified path
14.	Wait Till Page Ready		For JQuery pages waits till page is ready
15.	Wait Till JQuery Stable		Waits for 30 seconds or till JQuery processing is done

## Launch Web Page

Launches a web page during script execution (Right click on the component and choose **Selenium Launch Parameters** option)



Takes 2 parameters URL and Browser Type

## Assert Function

Attribute	Value
Display Name Of The Object	
Code/Data Path	value
Object Type	Step
Object Sub Type	ASSERT
Object Name	Step 17
Attributes XML	{:TV>DATASHEET,0,UserID}
Maximum Iterations	1
Mandatory Columns	


This is for asserting one of the web element property attributes.

In this case **value** of the current web element operated on (in latest step) should be equal to Data Table **DATASHEET** row 0 **UserID** column value (The Parameterization will be explained in detail in the upcoming sections).

If the Assertion fails, the script ends there after closing the web driver (just like normal Assertion in the NUnit or other unit testing frameworks);

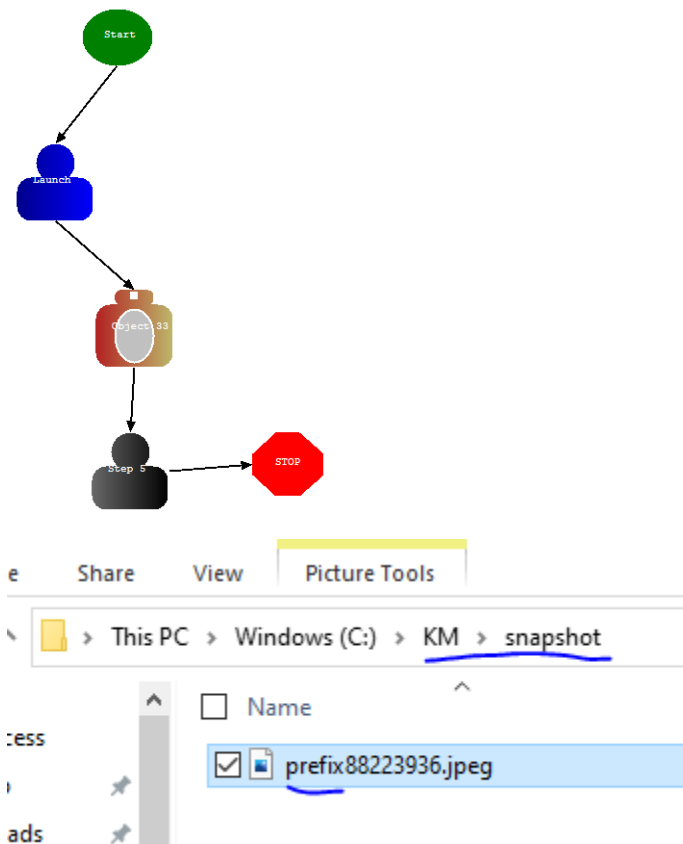
## Browser Snapshot

This is one of the many ways, browser snapshot can be taken



Attribute	Value
Display Name Of The Object	Object 33
Code/Data Path	c:\kmsnapshot
Object Type	Step
Object Sub Type	SNAPSHOT
Object Name	Step 2
Attributes XML	prefix
Maximum Iterations	1
Mandatory Columns	

A simple script using the component shown above will create a jpeg file as shown below. It creates the snapshot in the right path with right prefix and a random number added to its name

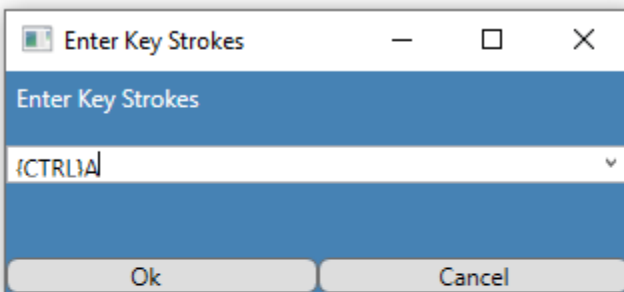


## Windows SendKeys function

There will be situations where during the testing of web pages we will encounter non-HTML objects and which are totally unidentified by any automation tool. In such cases we can utilize windows sendkeys function so that we can handle them with keystrokes. Below is the official Microsoft documentation on sendkeys.

[SendKeys Class \(System.Windows.Forms\) | Microsoft Docs](#)

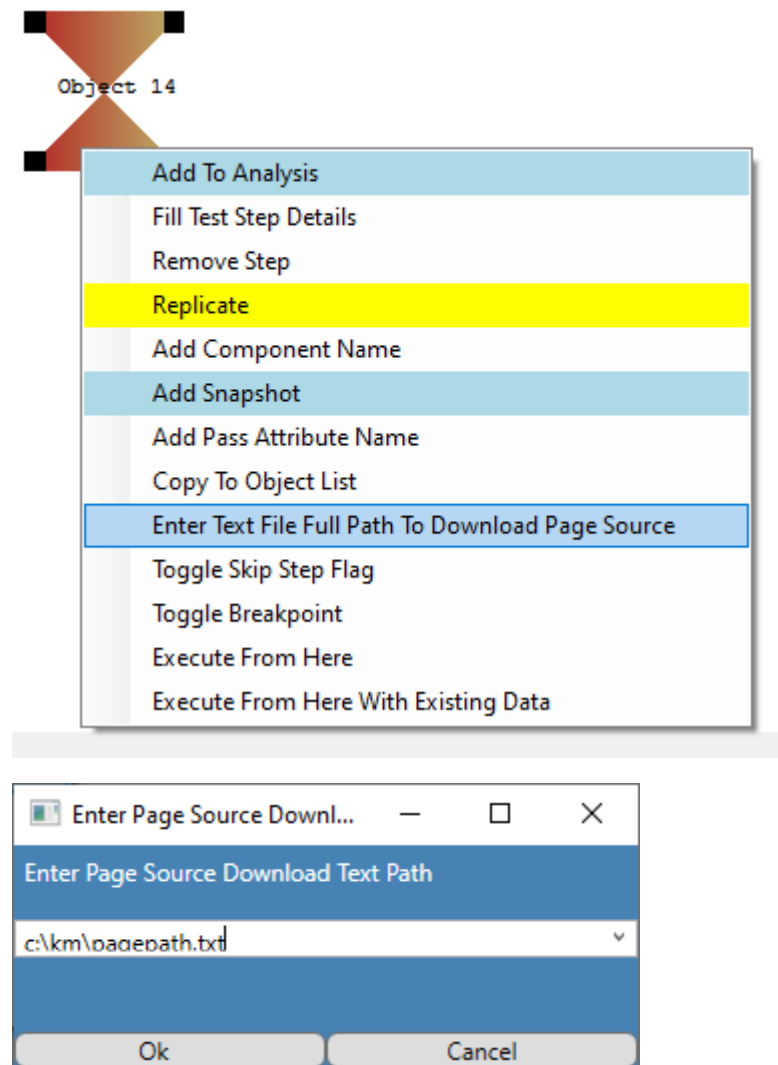
Drag the below component from the building blocks and Right Click on it and select **Enter Key Strokes To Be Simulated** option



In this example we are executing CTRL + A on the focused application

## HTML Page Source

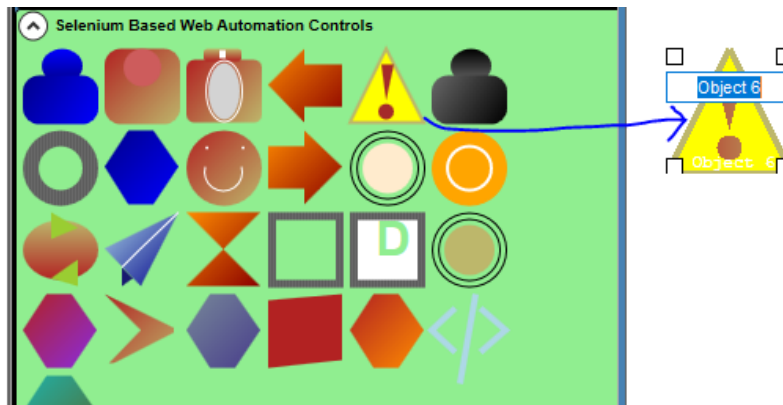
This is one another ways of getting HTML page source of the web page.



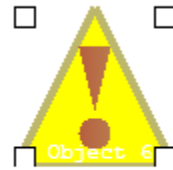
During Execution the HTML source will be saved on to the path selected.

## Web Alert Handling

Nested Flow automation framework allows handling of web alerts using the below component



This component takes only the Attribute XML attribute most of the time (except for SETTEXT option)



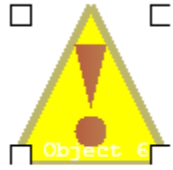
Attribute	Value
Display Name Of The Object	Step 6
Code/Data Path	
Object Type	Step
Object Sub Type	WEBALERT
Object Name	Step 6
Attributes XML	PASS
Maximum Iterations	1
Mandatory Columns	

**PASS** → Accepts the Alert (Clicks OK)

**FAIL** → Dismisses the Alert (Clicks Cancel)

**GETTEXT** → Gets alert Text

**SETTEXT** →



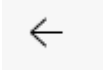
Attribute	Value
Display Name Of The Object	Step 6
Code/Data Path	Sends this Text
Object Type	Step
Object Sub Type	WEBALERT
Object Name	Step 6
Attributes XML	SETTEXT
Maximum Iterations	1
Mandatory Columns	

Sends the text in **Code/Data Path** to Alert


## Page Navigation Functions

These functions take no variables




This is Page backward function (equivalent to clicking on  on the browser)



This is Page Forward function (equivalent to clicking on  on the browser)



This is Page Refresh button (equivalent to clicking on  on the browser)

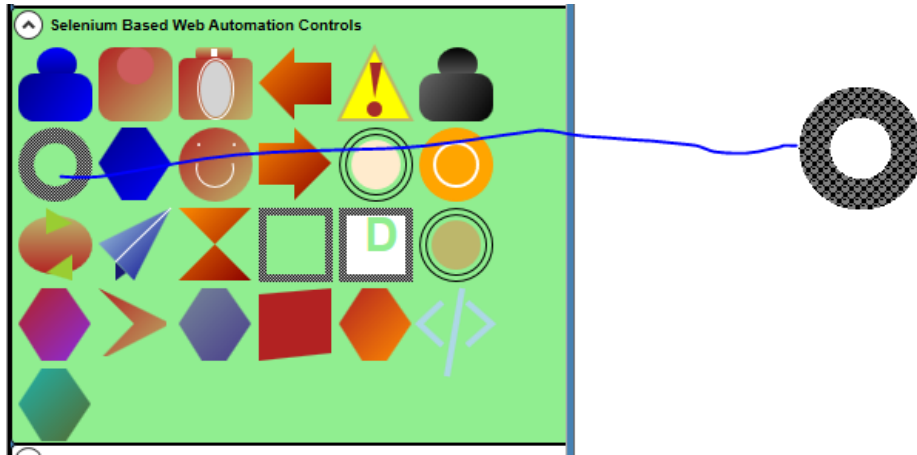


## Quitting Web Pages

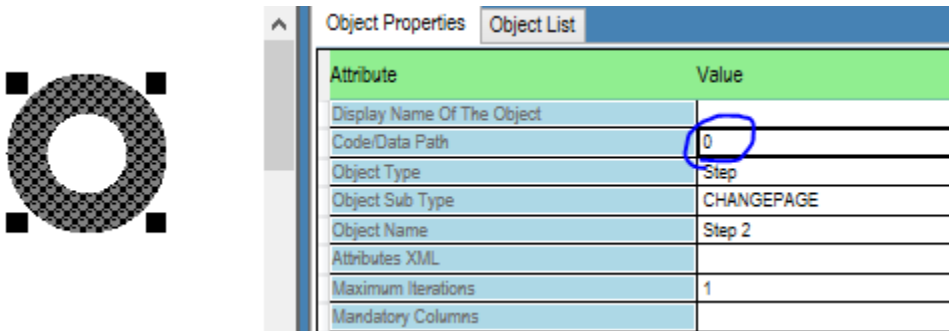
The below option (doesn't take variables) disposes the web driver



The component doesn't take any attribute



This is one of the ways to change the web page.



Code/Data Path → 0 Switches to first window

Code/Data Path → -2 Switches to last window

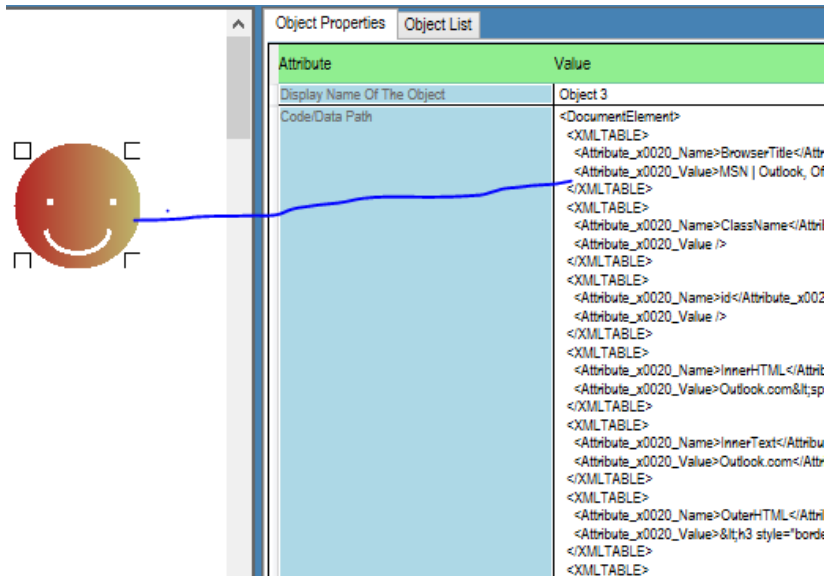
Code/Data Path → +ve number : Moves to nth window open (0 based index)

Code/Data Path → Moves to window with that title

## Object Check

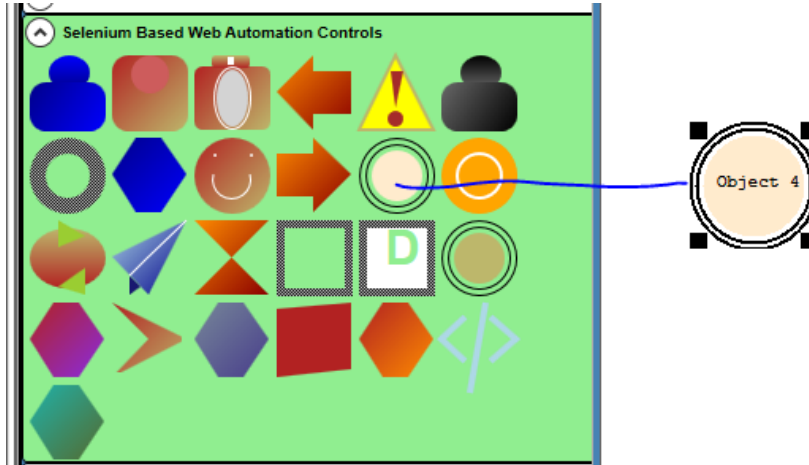
This option is used to check the object if it currently exists with same object properties. If object is existing PASS is returned else FAIL is returned.

Right click on the object and click on **Capture Web Object Properties** option. Web object spy will open. Choose an object using the spy and click on **Return Object** button

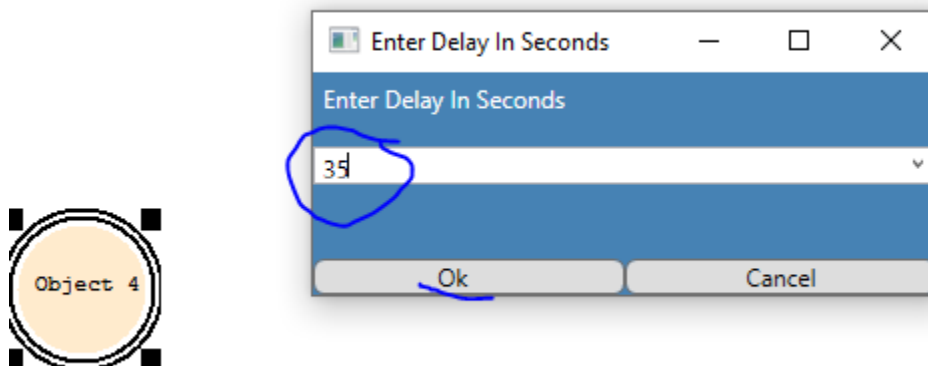


## Page Load Time Wait

The below function is used to set Page Load wait time for Selenium element and page operation purposes. Default wait time set by the tool is 30 seconds. **This option is practically not needed as tool sets default value.**

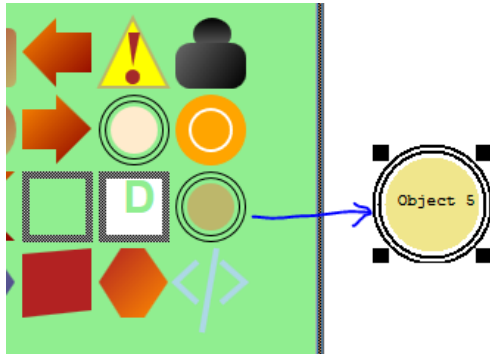


Right Click and select **Add Page Load Timeout Delay In Seconds**. Enter Delay and click on **Ok**.



## Explicit Wait Time

Default value set by the tool is 30 seconds (tool will wait for set amount of time for object to be available before throwing error). **This option is practically not needed as tool sets default value.**

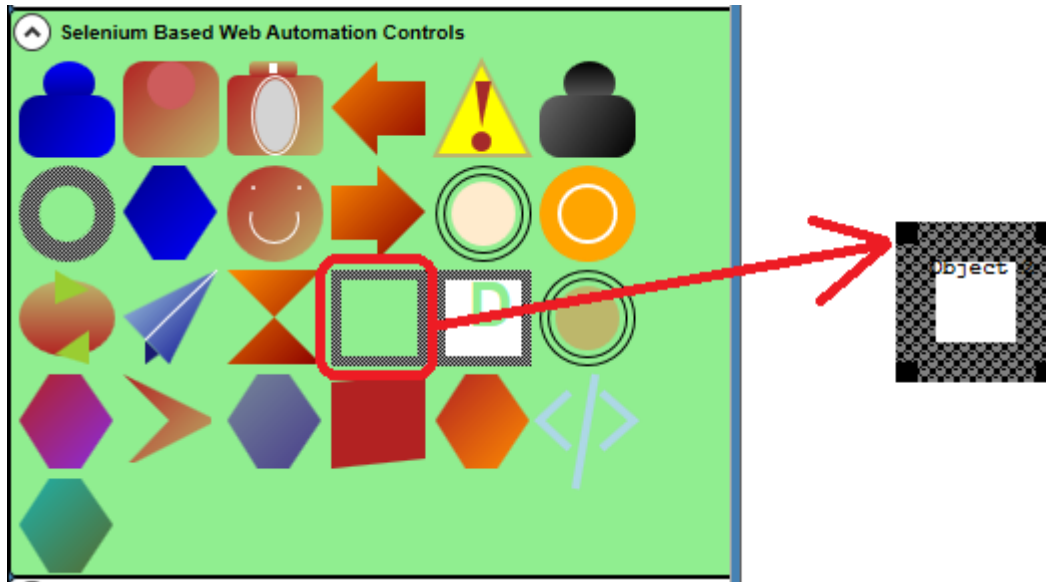


Right click on the component and select **Add Explicit Delay In Seconds** option.

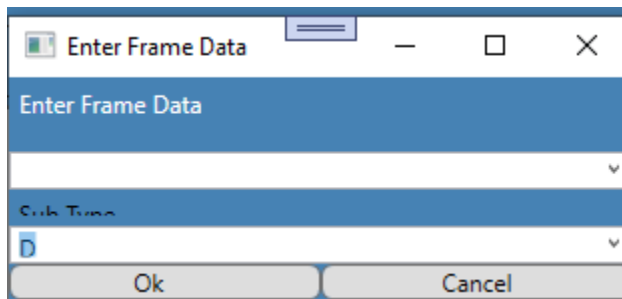
Enter Delay in seconds and click on **Ok** button

## Change Frame

Below function is one more way of changing frames (**do not use it unless necessary instead us [Move to Frame](#) function of web element**)



Right click on the object and select **Add Web Frame Data**.



If Sub Type = "D" then control moved to default frame

If Sub Type = blank and frame data is a string value then control is moved to frame with that name which is child of the frame in which the control currently exists

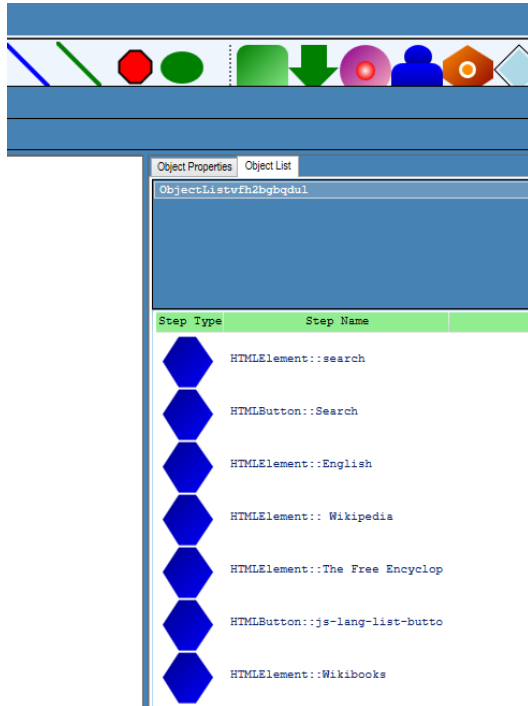
If Sub Type = blank and frame data is an integer (n) then control is moved to frame with that index which is child of the frame in which the control currently exists



→ this control moves the control to default content (top level of the web document)

## Bulk Object Check

In the document **Nested Flow HTML Object Spy.docx** page 5, it is explained how to generate the object page



Right click on any of the object and click on **Save**. It allows to save the page

Use the saved page (.xml) with the impact analysis object.



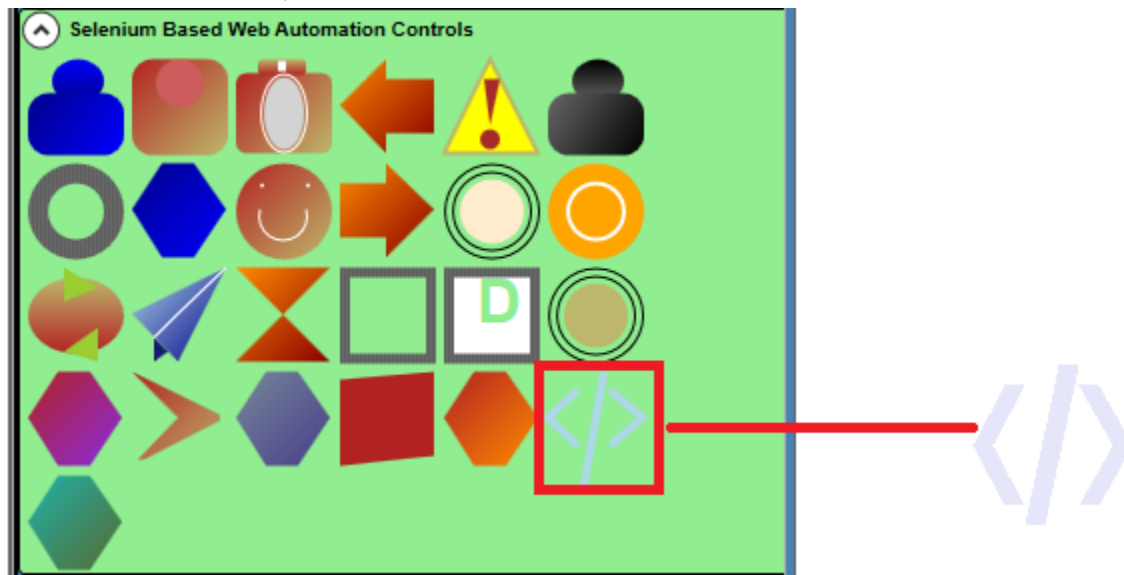
Run the object after the page is opened in the automation flow with the pages to be checked.

All the elements in the file with web object type are searched for existence on the web page.

All object checks are detailed out in the log file.

If any object check fails, the function stops and script ends.

## Execute JavaScript

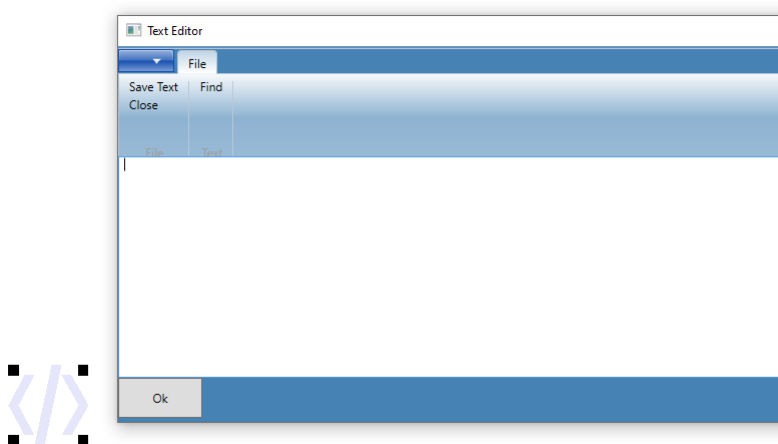


Following component is used to execute JavaScript during run time.

JavaScript can have one variable by name argument[0] (that is the current object selected last)

Or it can be plain JavaScript.

Right Click on the object and click **Add JavaScript** option



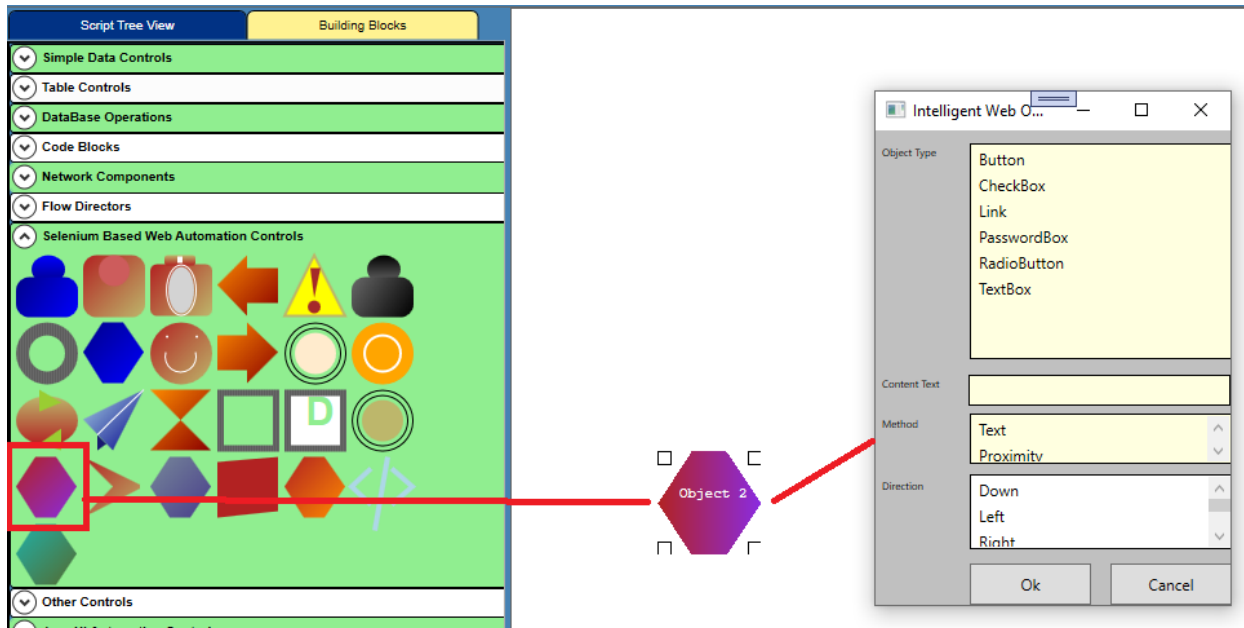
Enter JavaScript and click on **Ok**

**Example:** `window.globalparam = "1000";`



## Smart Object Identification

This feature is to be used only when there are a lot of frequent changes to web page and we are trying to create elements smartly so that most of our test case executes smoothly



Right click on the object and choose **Create Smart Web Element** option.

If Method = **Text**

Then **Direction** field is having no relevance

Tool will try to find object on web page whose type is the chosen **Object Type** with inner text = Content Text or id or name of the object contains that text

If Method = **Proximity**

Then object with text like **Content Text** is found and then closest element to it in the given **Direction** of object type given in **Object Type** is returned

## Dynamic Parameterization

### Table Value Parameters:

	A	B	C	D	E	F	G	H	I	J	K
1	URL	BROWSER	First Name	Last Name	Email	Phone	Address	City	Zip	WebSite	Description
2	<a href="https://demo.seleniumeasy.com/input-form-demo.html">https://demo.seleniumeasy.com/input-form-demo.html</a>	EDGE	James	Adams	<a href="mailto:jadam.71@fi.com">jadam.71@fi.com</a>	(484)319-8872	14612 Highland Cir	Little Elm	75068	<a href="https://www.wysite.com">https://www.wysite.com</a>	My Test Site
3											
4											

For creating a data driven script, we can use csv, Excel, xml or db queries.


In this example we use: <https://demo.seleniumeasy.com/input-form-demo.html>

To enter the fields as mentioned in the sheet and click **Send**


### Input form with validations

**Contact Us Today!**


First Name

 First Name


Last Name

 Last Name


E-Mail

 E-Mail Address


Phone #

 (845)555-1212


Address

 Address


City

 city


State

 Please select your state ▼

Zip Code

 Zip Code


Website or domain name


 Website or domain name

Do you have hosting?

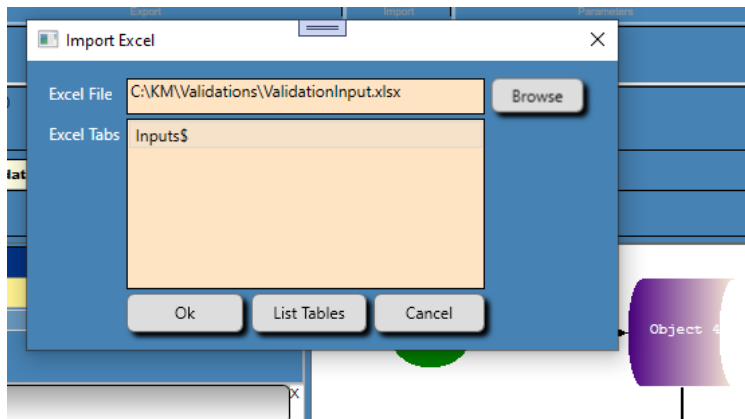
☐ Yes  
☐ No

Project Description

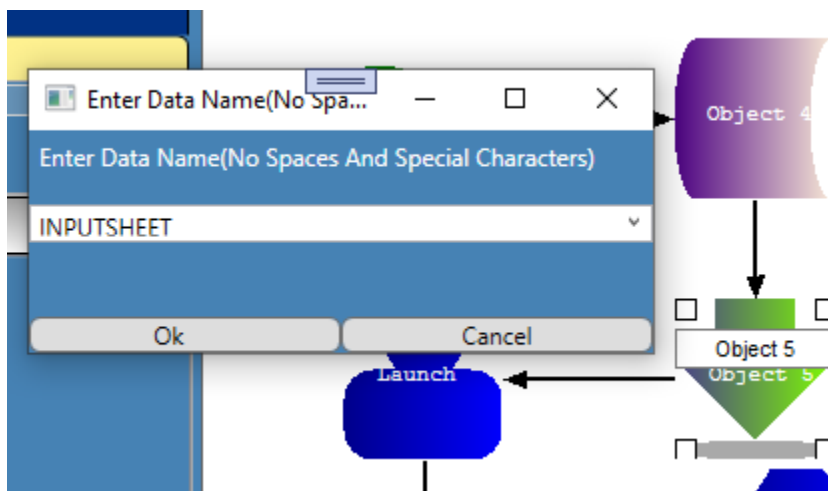
 Project Description

Send 

First the data sheet is chosen and proper sheet is selected. Either Click on **Ok** or double click on sheet name.

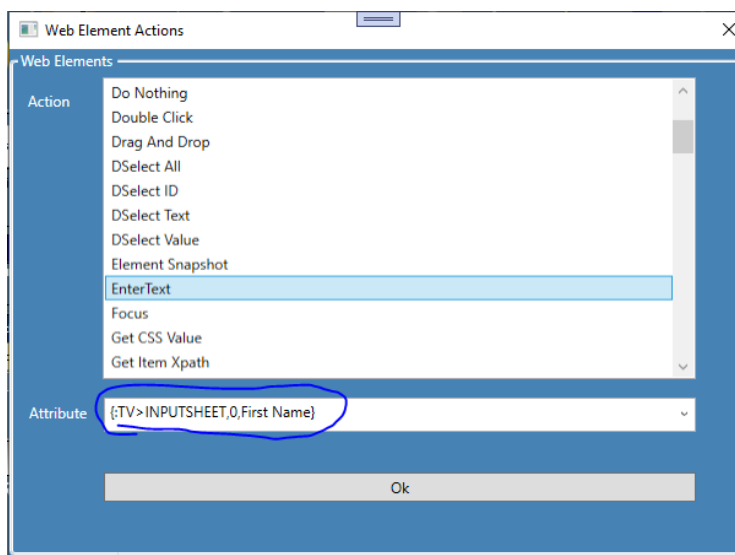


In the next step the excel sheet is saved with a name



(In this case the name is INPUTSHEET)

We use following field value to choose value from Excel



What it means is

TV → Table Value

INPUTSHEET → Table Name

0 → row number (first row which is not headers)

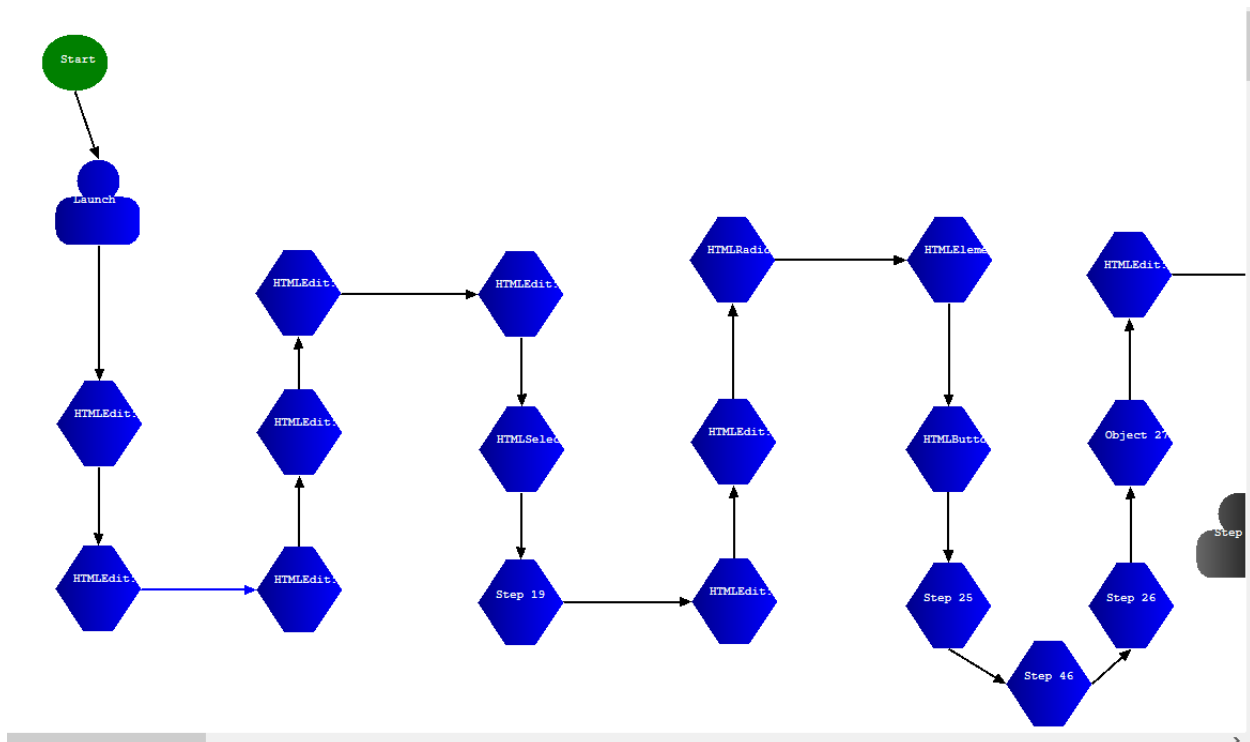
First Name → Column Header name

## Traversing All Rows Of Parameter Sheet

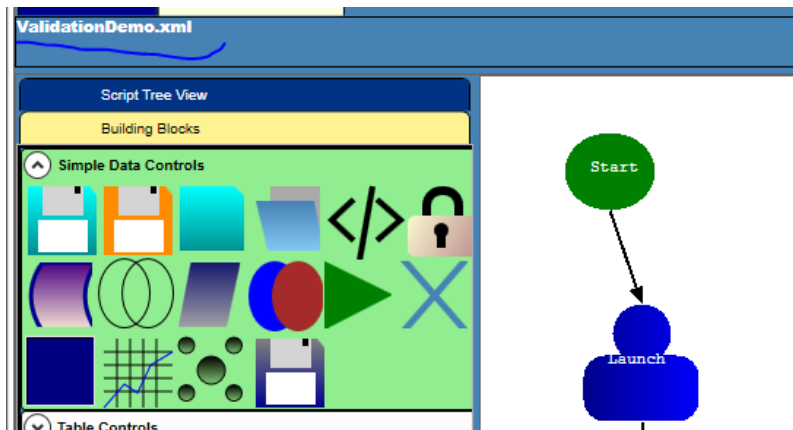
In this example we will traverse through all 3 rows of excel and perform the data entry tasks

#	A	B	C	D	E	F	G	H	I	J	K	L
1	URL	BROWSER	First Name	Last Name	Email	Phone	Address	City	State	Zip	WebSite	Description
2	<a href="https://demo.seleniumeasy.com/input-form-demo.html">https://demo.seleniumeasy.com/input-form-demo.html</a>	EDGE	James	Adams	<a href="mailto:jadam.71@fi.com">jadam.71@fi.com</a>	(484)319-8872	14612 Highland Cir	Little Elm	Texas	75068	<a href="https://www.wpsite.com">https://www.wpsite.com</a>	My Test Site
3	<a href="https://demo.seleniumeasy.com/input-form-demo.html">https://demo.seleniumeasy.com/input-form-demo.html</a>	EDGE	Aaron	Finch	<a href="mailto:Afinch.74567@fi.com">Afinch.74567@fi.com</a>	(484)319-9078	14619 Highland Cir	Little Elm	Texas	75068	<a href="https://www.afsite.com">https://www.afsite.com</a>	New Site
4	<a href="https://demo.seleniumeasy.com/input-form-demo.html">https://demo.seleniumeasy.com/input-form-demo.html</a>	EDGE	Michael	Pear	<a href="mailto:Mpear22345@fi.com">Mpear22345@fi.com</a>	(484)319-9078	15000 Highland Cir	Little Elm	Texas	75068	<a href="https://www.mpsite.com">https://www.mpsite.com</a>	My new Site

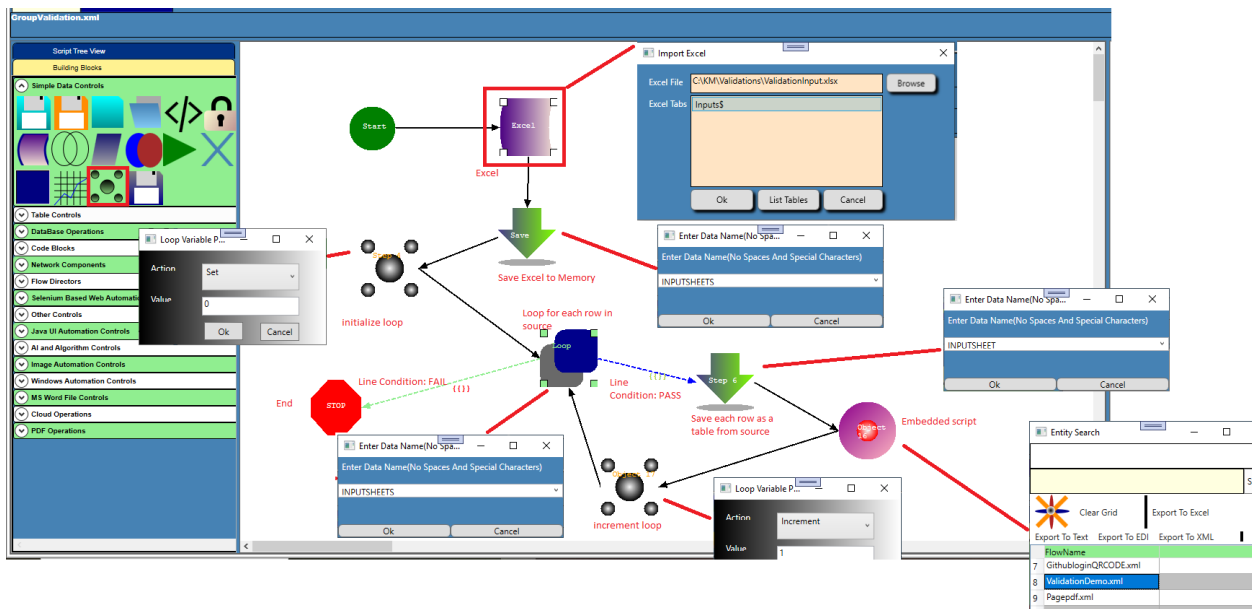
Remove the data sheet selection from the original script



Save the script



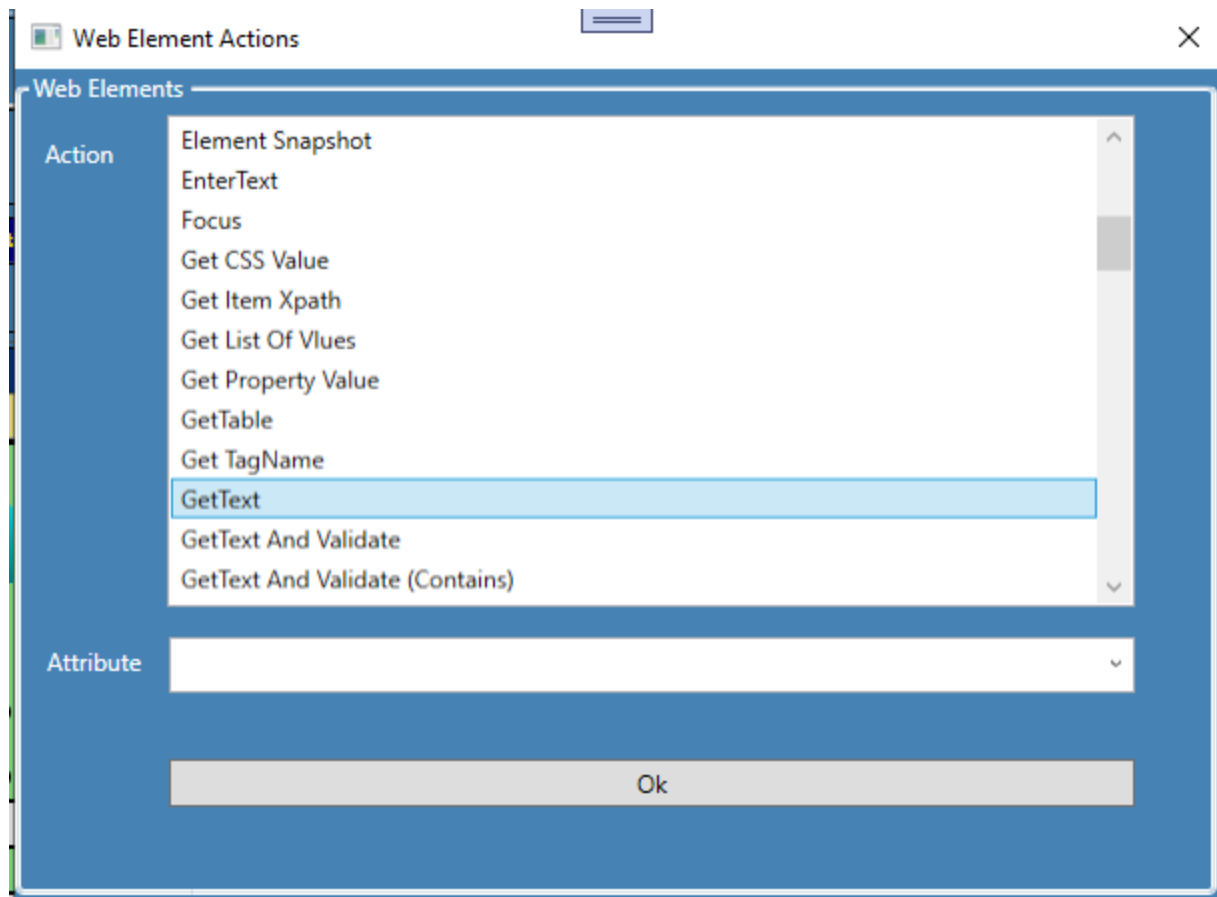
Create another script and loop through each row of the excel and call the data entry script as the embedded script as shown below



## Condition Value

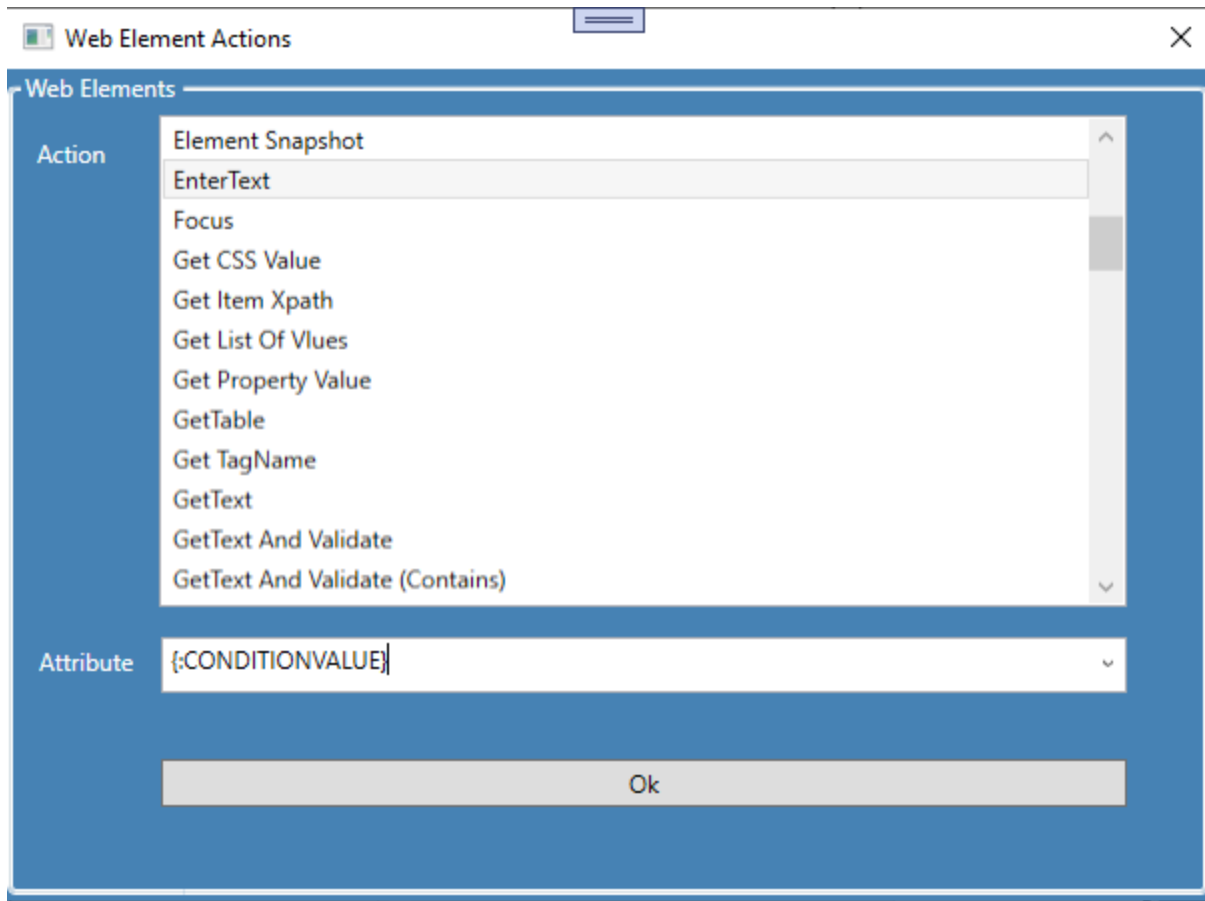
{:CONDITIONVALUE} will return the most recently returned value in the flow.

For example:



GetText function will get a Text from the web element.

If we need to use the captured value in next text box:



#### Log File:

{:LOGFILE} will return the value of execution log file

#### Pattern:

{:CS>Pattern} will return value according to a pattern

Example: {:CS>nnnnnnnnCCCCCcccccc/N} will return a string with 5 integers followed by 5 Uppercase letters followed by 7 lower case characters followed by letter N (/ is the escape character)

#### Random List:

{:RL>separator list} will return one random value from the list provided

Example: {:RL>,11,12,13,14} will take first letter after :RL> as separator and rest all as separator controlled list and while execution one of 11,12,13 and 14 is returned

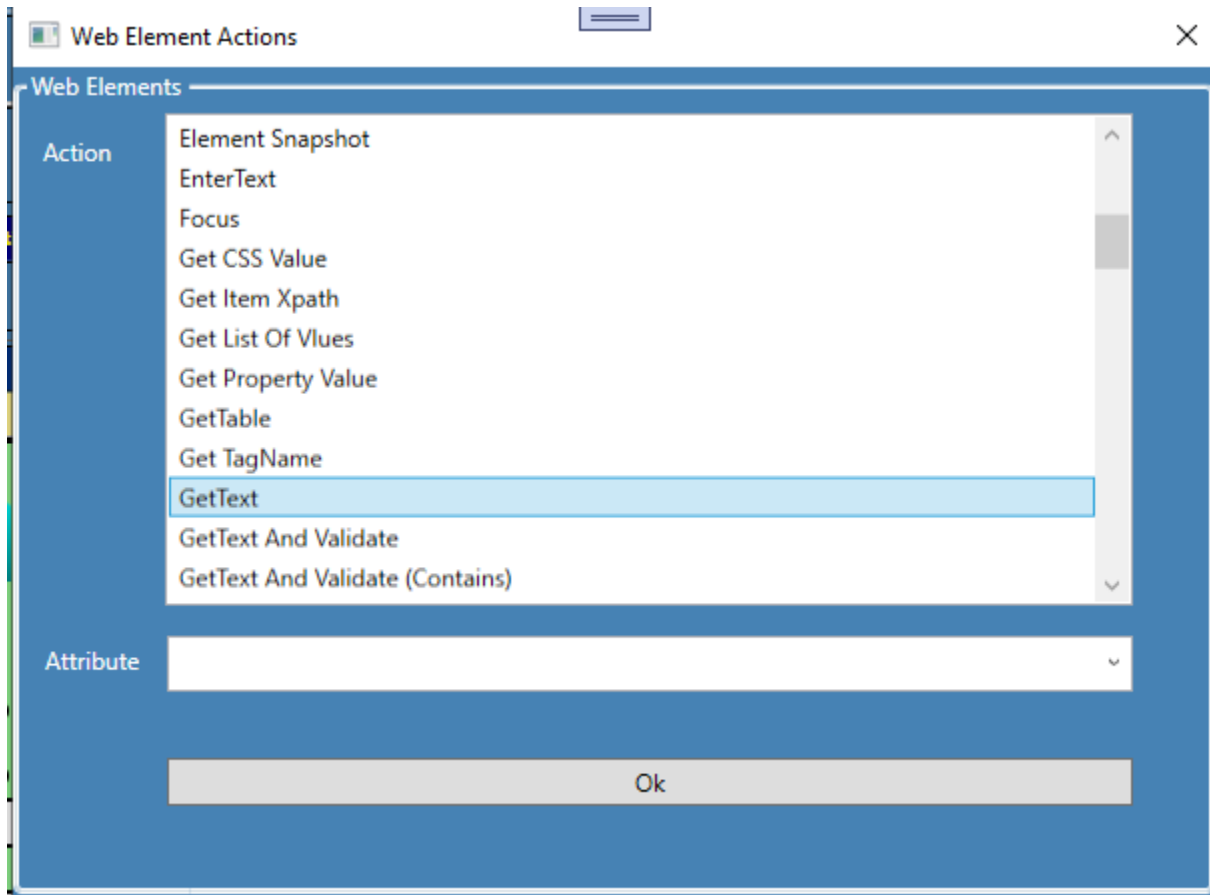
#### Previous Block:

{:PREVIOUSBLOCK} will return display name of the last executed step

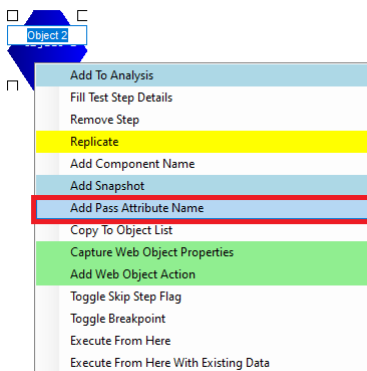
## Pass Parameter And Code Based String Validations

Nested Flow Automation Framework provides a lot of inbuilt string validation functions already explained in the previous sections. But there might be cases where available validations are not enough. In such cases the tool allows coded way of validating strings. This section will explain ways to do it.

Use **Get Property Value** or **GetText** methods to capture the web element attributes.

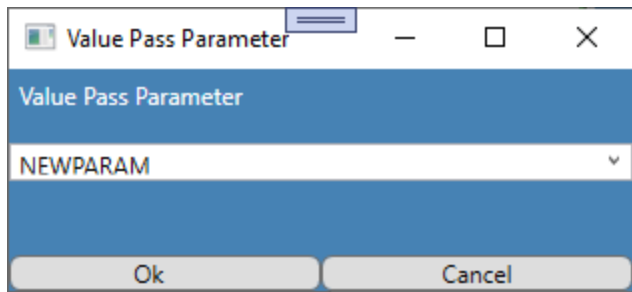


Right click on the element and choose **Add Pass Attribute Name** option



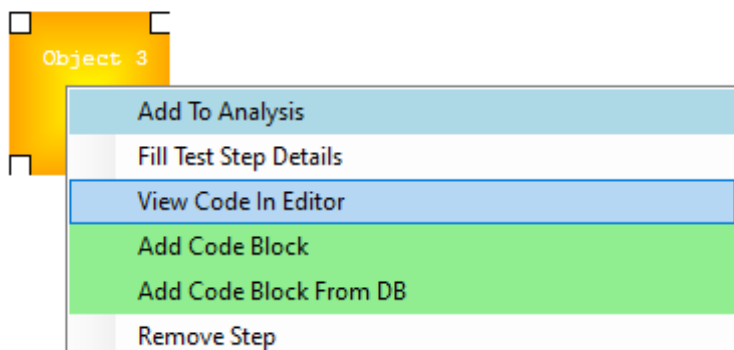
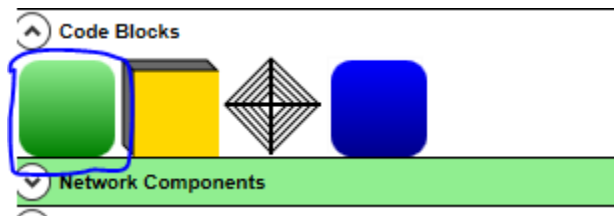
Give name to the parameter:



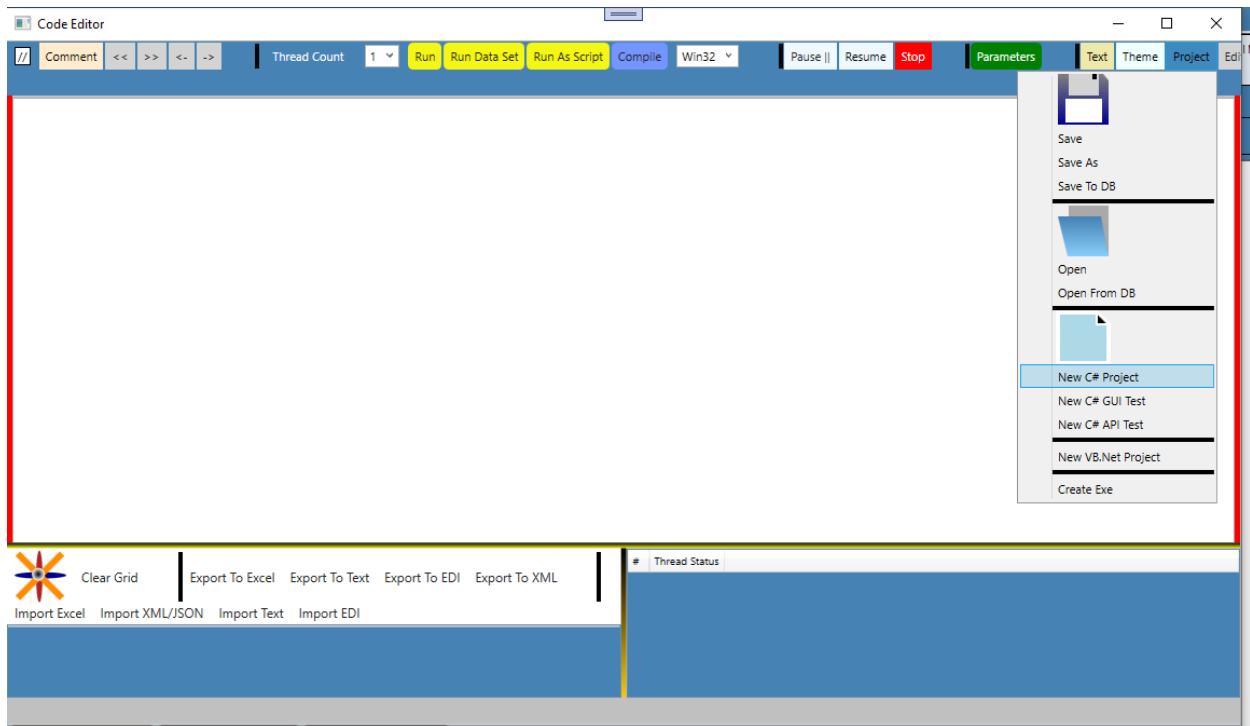


Click on **Ok** button

Drag a code block to the draw panel



Right click on the block and choose **View Code In Editor** option Click on Project → New C# Project



Bare bone code gets generated.

```

1  using System;
2  using System.Xml;
3  using System.Data;
4  using System.Data.SqlClient;
5  using System.Windows.Forms;
6  using System.Drawing;
7
8  using System.IO;
9
10 //Return value from Eval Code will be assigned to column
11 //DataTable T1: Whatever Data in Bottom Grid Of Query 1
12 //string s1 : Oracle User Name
13 //string s2 : Oracle Password
14 //string s3 : Second User Name
15 //string s4 : Second Password
16
17 namespace NSUpdateTable
18 {
19     public class CSUpdateTable
20     {
21         //Return value from Eval Code will be assigned to column
22         public object UpdateTable(DataTable T1, string s1, string s2, string s3, string s4, ref string s5)
23         {
24
25             return (T1);
26         }
27     }
28 }
29

```

New parameter is accessible in the code

```
//Return value from Eval Code will be assigned to column
public object UpdateTable(DataTable T1,string s1,string s2,string s3,string s4,ref string s5)
{
    string pstr = "(:NEWPARAM)";
    return (T1);
}
```

S5 in the code is the return parameter.

If validation is success assign s5="TRUE" else assign "FALSE"

(if the code block returns "FAIL" script will stop there itself – So it can be used as a good Assert scenario)

Save code **Project** → **Save** option

Save the file as .cs file

Close code editor

Right click on the block and choose **Add Code Block** and choose the .cs

## Web Code

Nested Flow Automation Framework also provides inbuilt support to Internet Explorer, FireFox, Chrome, MS Edge and Edge in IE Mode. But there will be needs when

- We need to connect to a selenium grid
- Some web element function not supported by the tool out of the box
- Connect to devices on Perfecto mobile etc. to run web scripts on mobile devices

To perform such activities we use web code functionality.

**Tools → C# Editor**

**Navigate to Project → New C# GUI Test**

```
New Project
1 using System;
2 using System.Xml;
3 using System.Data;
4 using System.Data.SqlClient;
5 using System.Windows.Forms;
6 using System.Drawing;
7
8 using System.IO;
9
10 using OpenQA.Selenium;
11
12 using OpenQA.Selenium.Support.UI;
13
14 using OpenQA.Selenium.IE;
15
16 using OpenQA.Selenium.Firefox;
17
18 using OpenQA.Selenium.Chrome;
19
20 using OpenQA.Selenium.Interactions;
21
22 using OpenQA.Selenium.Remote;
23
24 //Return value from Eval Code will be assigned to column
25 //DataTable T1: Whatever Data in Bottom Grid Of Query 1
26 //string s1 : Oracle User Name
27 //string s2 : Oracle Password
28 //string s3 : Second User Name
29 //string s4 : Second Password
30
31 namespace NSUpdateTable
32 {
33     public class CSUpdateTable
34     {
35         //Return value from Eval Code will be assigned to column
36         public object UpdateTable(DataTable T1,string s1,string s2,string s3,string s4,ref string s5)
37         {
38
39             XMLParser.LogFileActions.CreateLogFile(); //GetLogFile
40             XMLParser.Web_Automation_Framework.SeleniumFunctions SF = new XMLParser.Web_Automation_Framework.SeleniumFunctions();
41
42             string webtype = "";
43             string pid = "";
44             SF.InitializeWebDriver("CHROME",pid,out webtype);
45             bool retval;
46             SF.LaunchDriver("https://<URL>/", out retval);          return (T1);
47         }
48     }
49 }
50
```

Change first input parameter to IWebDriver IWeb

```

1  using System;
2  using System.Xml;
3  using System.Data;
4  using System.Data.SqlClient;
5  using System.Windows.Forms;
6  using System.Drawing;
7
8  using System.IO;
9
10 using OpenQA.Selenium;
11
12 using OpenQA.Selenium.Support.UI;
13
14 using OpenQA.Selenium.IE;
15
16 using OpenQA.Selenium.Firefox;
17
18 using OpenQA.Selenium.Chrome;
19
20 using OpenQA.Selenium.Interactions;
21
22 using OpenQA.Selenium.Remote;
23
24 //Return value from Eval Code will be assigned to column
25 //DataTable T1: Whatever Data in Bottom Grid Of Query 1
26 //string s1 : Oracle User Name
27 //string s2 : Oracle Password
28 //string s3 : Second User Name
29 //string s4 : Second Password
30
31 namespace NSUpdateTable
32 {
33     public class CSUpdateTable
34     {
35         //Return value from Eval Code will be assigned to column
36         public object UpdateTable(IWebDriver IWeb, string s1, string s2, string s3, string s4, ref string s5)
37         {
38
39             XMLParser.LogFileActions.CreateLogFile(); //GetLogFile
40             XMLParser.Web_Automation_Framework.SeleniumFunctions SF = new XMLParser.Web_Automation_Framework.SeleniumFunctions();
41
42             string webtype = "";
43             string pid = "";
44             SF.InitializeWebDriver("CHROME", pid, out webtype);
45             bool retval;
46             SF.LaunchDriver("https://<URL>/", out retval); return (IWeb);
47         }
48     }
49 }
50

```

Let us connect to a selenium grid in the script and save the script by clicking on **Project → Save**

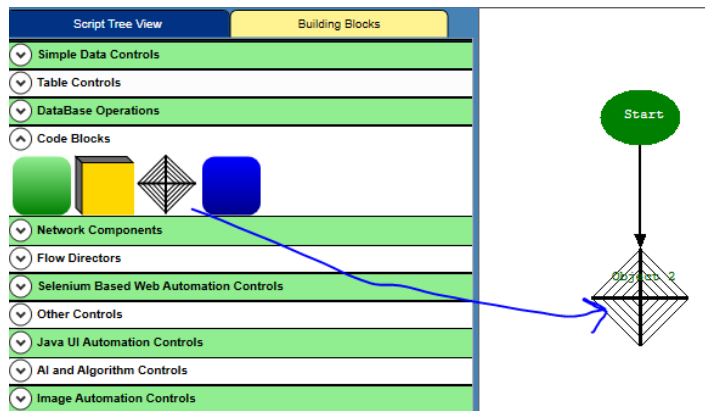
```

gridconnect
1  using System;
2  using System.Xml;
3  using System.Data;
4  using System.Data.SqlClient;
5  using System.Windows.Forms;
6  using System.Drawing;
7
8  using System.IO;
9
10 using OpenQA.Selenium;
11
12 using OpenQA.Selenium.Support.UI;
13
14 using OpenQA.Selenium.IE;
15
16 using OpenQA.Selenium.Firefox;
17
18 using OpenQA.Selenium.Chrome;
19
20 using OpenQA.Selenium.Interactions;
21
22 using OpenQA.Selenium.Remote;
23
24 //Return value from Eval Code will be assigned to column
25 //DataTable T1: Whatever Data in Bottom Grid Of Query 1
26 //string s1 : Oracle User Name
27 //string s2 : Oracle Password
28 //string s3 : Second User Name
29 //string s4 : Second Password
30
31 namespace NSUpdateTable
32 {
33     public class CSUpdateTable
34     {
35         //Return value from Eval Code will be assigned to column
36         public object UpdateTable(IWebDriver IWeb, string s1, string s2, string s3, string s4, ref string s5)
37         {
38
39             XMLParser.LogFileActions.CreateLogFile(); //GetLogFile
40             XMLParser.Web_Automation_Framework.SeleniumFunctions SF = new XMLParser.Web_Automation_Framework.SeleniumFunctions();
41
42             ChromeOptions options = new ChromeOptions();
43             options.AddArgument("no-sandbox");
44             options.AddAdditionalCapability("useAutomationExtension", false);
45             IWeb = new RemoteWebDriver(new Uri("http://<IP>:<PORT>/wd/hub"), options);
46             return (IWeb);
47         }
48     }
49 }
50

```

Exit C# script editor.


Drag web code to draw Panel



Right click on the object and choose **Add Code Block**

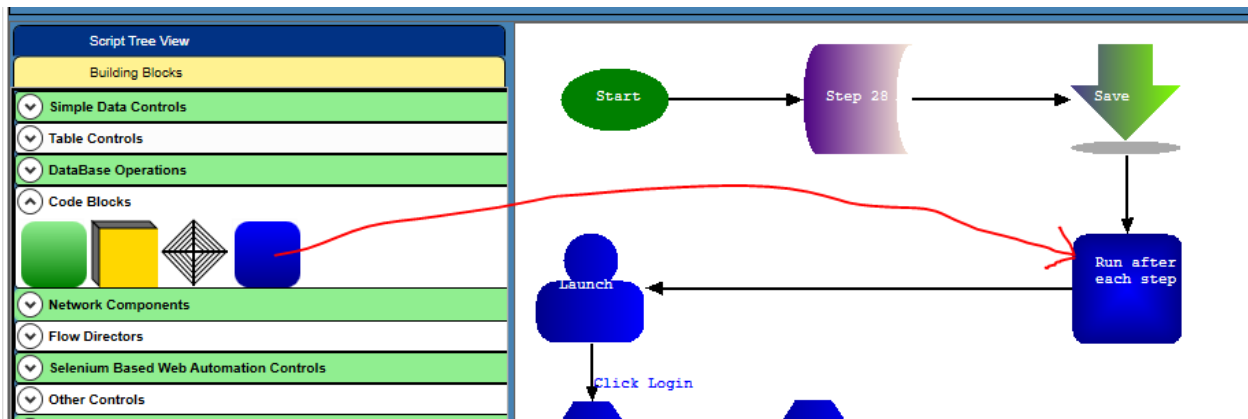
Choose the .cs script created.



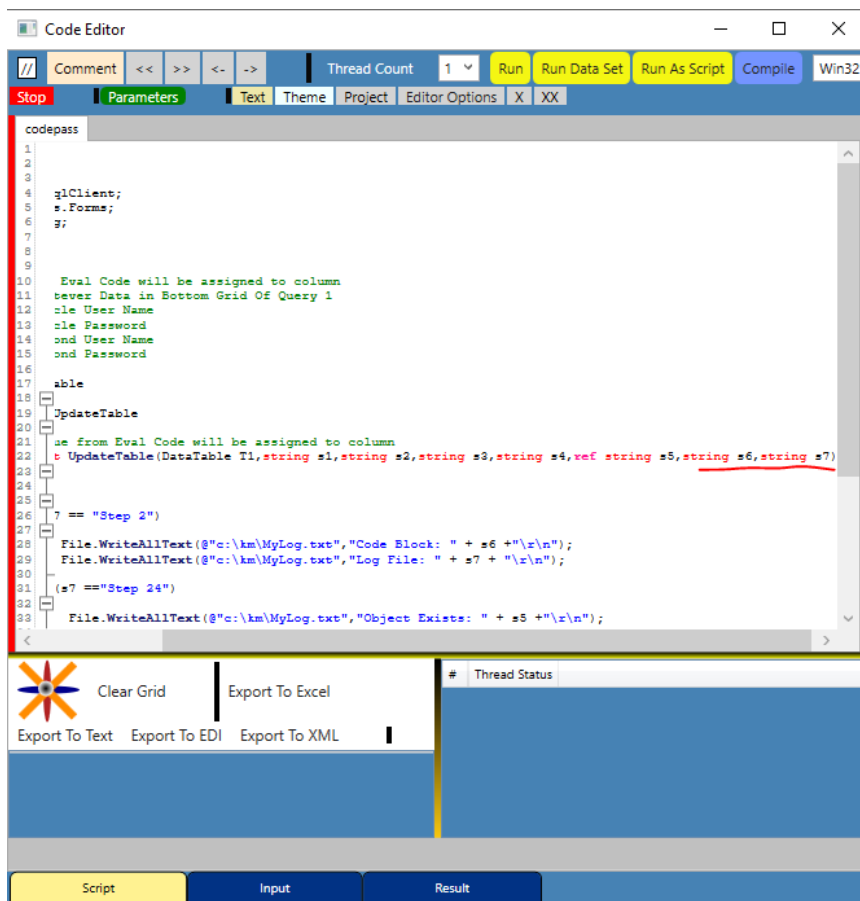
During the execution, instead of , web code block will initialize selenium and rest all object control functions remain the same.

## Script After Every Step

Just like any standard automation framework, Nested Flow also provides feature to run a code after each step.



The script should have 2 more parameters than created by default **Project** → **New C# Project**



S6 → Code Block Name (e.g.: Step 2, Step 3 and so on)

S7 → Execution Log File Path

This script can be capture parameters after each step and to perform some logging.

`File.AppendAllText(s7, "\r\n" + "<Message Here>" + "\r\n");` will put the log message directly on to the execution log message.