Experiment No. - 8

Student Name: Vivek Kumar
Branch: BE-CSE(LEET)
Semester: 6th
Subject Name: Software Testing Lab

UID: 21BCS8129
Section/Group:20BCS-ST-801/B
Date of Performance: 28/04/2023
Subject Code: 20CSP-380

- **Aim/Overview of the practical:**
  Write a test suite using Selenium IDE with minimum 2 test cases**.**

- **Objective:** Selenium IDE (Integrated Development Environment) is the simplest tool in the Selenium Suite. It is a Firefox add-on that creates tests very quickly through its record-and-playback functionality. This feature is similar to that of QTP. It is effortless to install and easy to learn. Because of its simplicity, Selenium IDE should only be used as a prototyping tool, not an overall solution for developing and maintaining complex test suites.

  Though you will be able to use Selenium IDE without prior knowledge in programming, you should at least be familiar with HTML, JavaScript, and the DOM (Document Object Model) to utilize this tool to its full potential. Knowledge of JavaScript will be required when we get to the section about the Selenese command "runScript."

Input/Apparatus used

- **Requirement Analysis: -** Google Chrome, Selenium IDE, Online tool

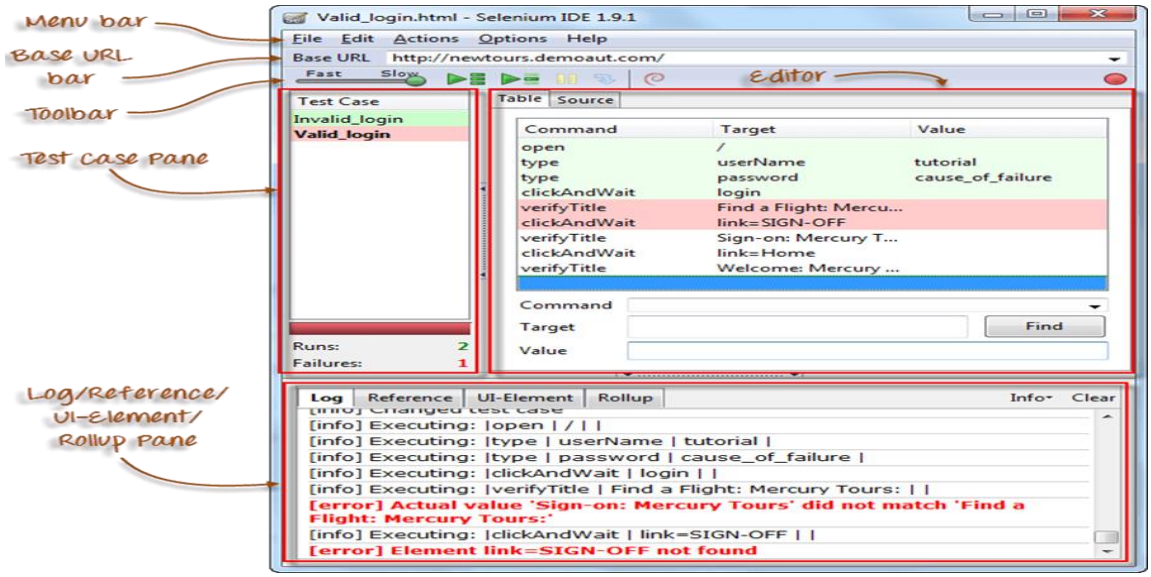
- **Hardware Requirement: -** Computer, Windows Power Supply.

Procedure/ Algorithm/ Code

Selenium IDE supports autocomplete mode when creating tests. This feature serves two purposes:

- It helps the tester to enter commands more quickly.
- It restricts the user from entering invalid commands.

Selenium IDE Features



Menu Bar

It is located at the **top most portion** of the IDE. The most commonly used menus are the File, Edit, and Options menus.

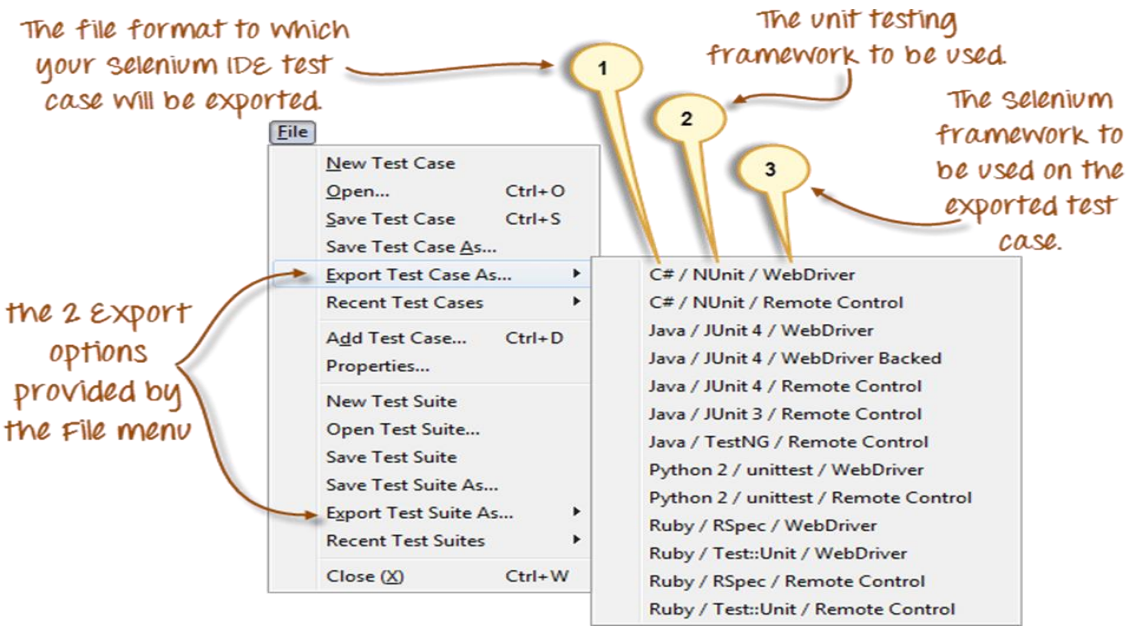
· It contains options to create, open, save and close tests.

· Tests are saved in HTML format.

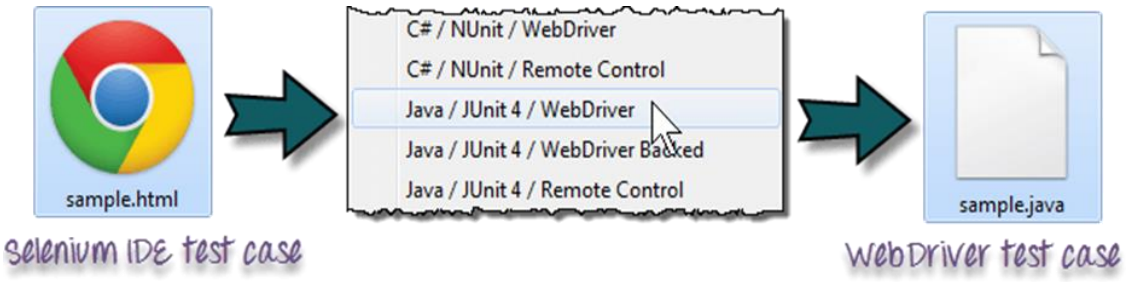
· The most useful option is "Export" because it allows you to turn your Selenium IDE test cases into file formats that can run on Selenium Remote Control and WebDriver

· "Export Test Case As…" will export only the currently opened test case.

· "Export Test Suite As…" will export all the test cases in the currently opened test suite.

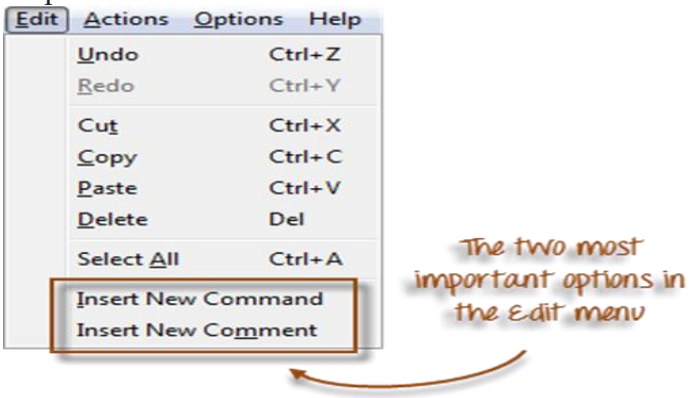


- As of Selenium IDE v1.9.1, test cases can be exported only to the following formats:
- .cs (C# source code)
- .java (Java source code)
- .py (Python source code)
- .rb (Ruby source code)

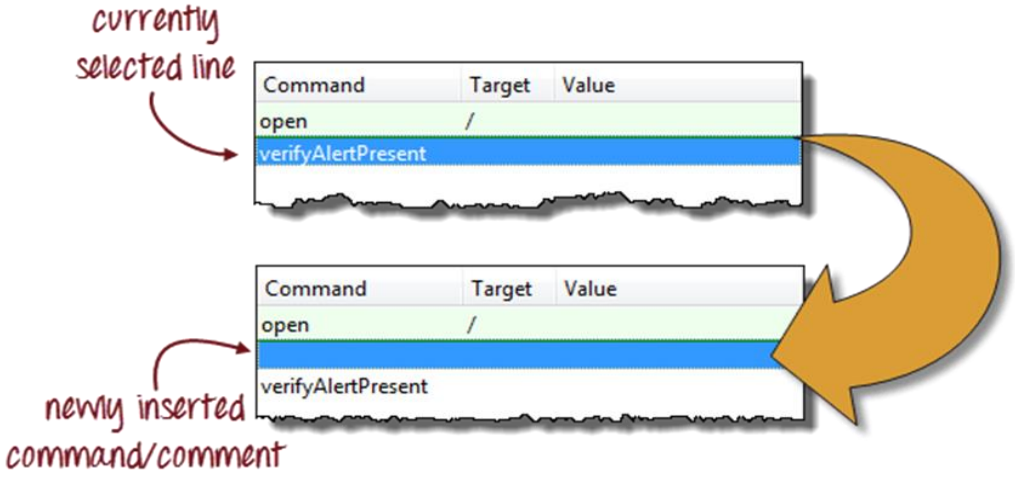


Edit Menu

- It contains usual options like Undo, Redo, Cut, Copy, Paste, Delete, and Select All.
- The two most important options are the “Insert New Command” and “Insert New Comment “.

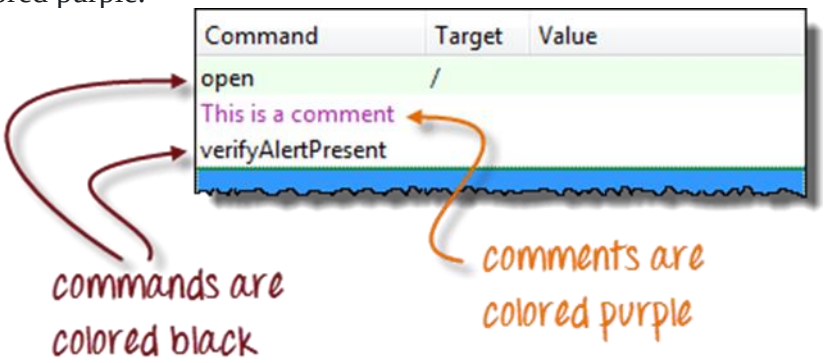


- The newly inserted command or comment will be placed on top of the currently selected line.



· Commands are colored black.

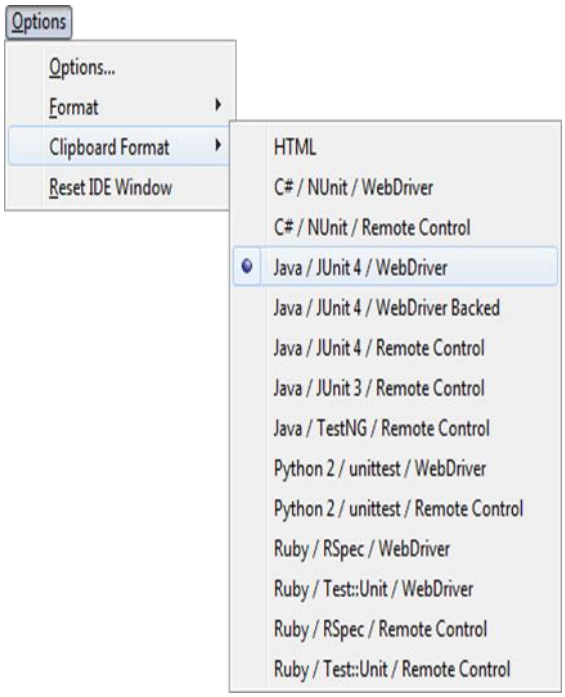
· Comments are colored purple.



Options menu

It provides the interface for configuring various settings of Selenium IDE.

We shall concentrate on the **Options** and **Clipboard Format** options.



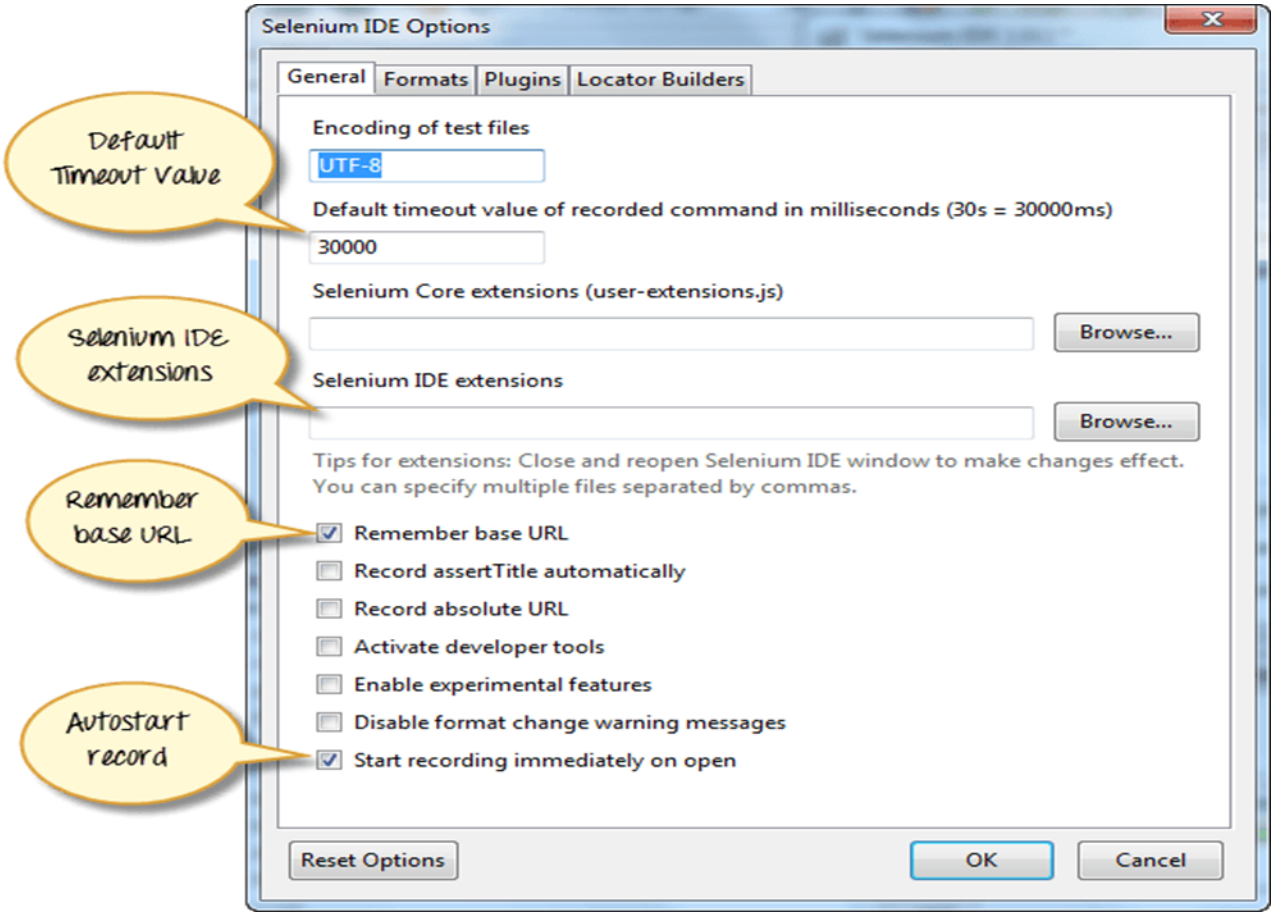
Clipboard Format

- **The Clipboard Format allows you to copy a Selenese command from the editor and paste it as a code snippet**.
- The format of the code follows the option you selected here in Clipboard Format's list.
- **HTML is the default selection.**

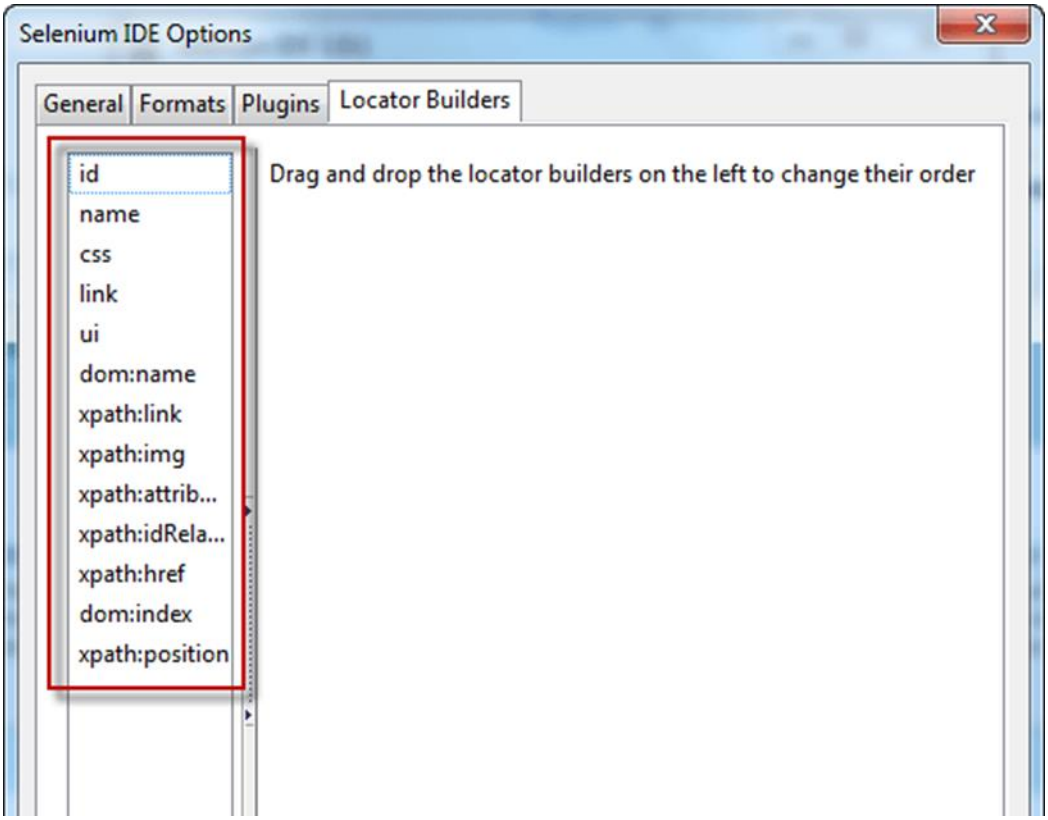
For example, when you choose **Java/JUnit 4/WebDriver** as your clipboard format, every Selenese command you copy from Selenium IDE's editor will be pasted as Java code. See the illustration

Selenium IDE Options dialog box

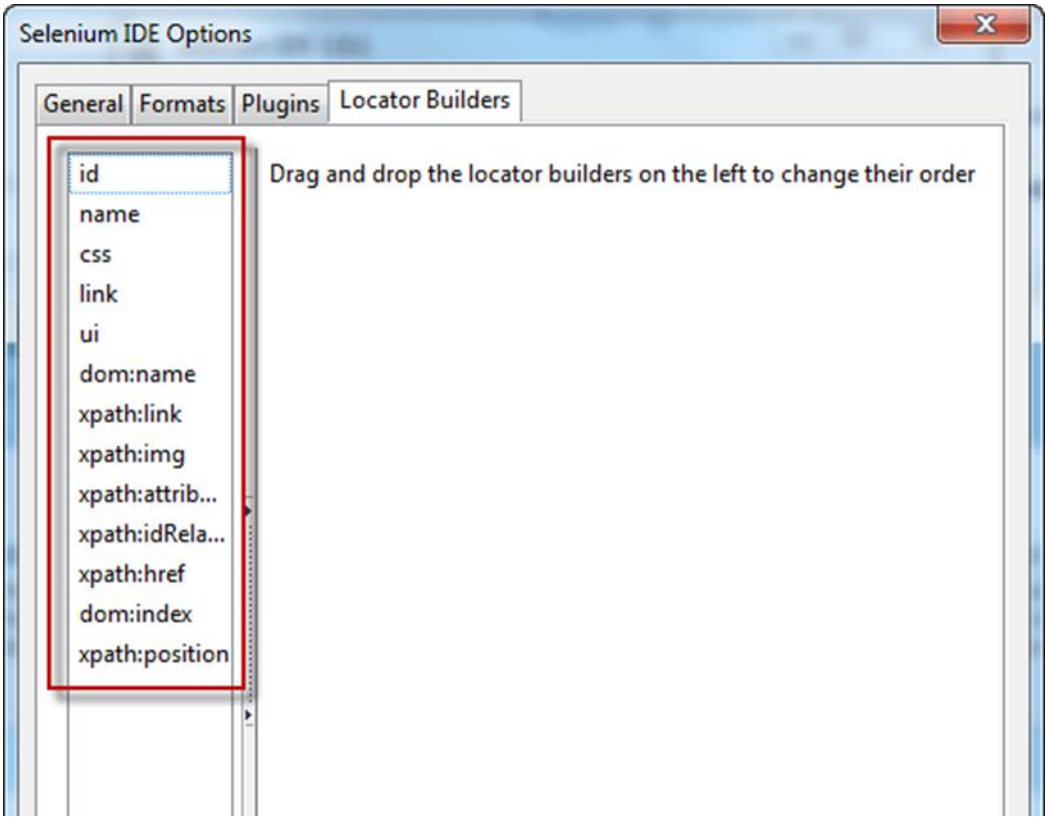
You can launch the Selenium IDE Options dialog box by clicking Options > Options… on the menu bar. Though there are many settings available, we will concentrate on the few important ones.



- **Default Timeout Value**. This refers to the time that Selenium has to wait for a certain element to appear or become accessible before it generates an error. **Default timeout value is 30000ms**.
- **Selenium IDE extensions**. This is where you specify the extensions you want to use to extend Selenium IDE's capabilities. You can visit http://addons.mozilla.org/en-US/firefox/ and use "Selenium" as a keyword to search for the specific extensions.
- **Remember base URL.** Keep this checked if you want Selenium IDE to remember the Base URL every time you launch it. If you uncheck this, Selenium IDE will always launch with a blank value for the Base URL.
- **Autostart record.** If you check this, Selenium IDE will immediately record your browser actions upon startup.
- **Locator builders.** This is where you specify the order by which locators are generated while recording. Locators are ways to tell Selenium IDE which UI element should a Selenese command act upon. In the setup below, when you click on an element with an ID attribute, that element's ID will be used as the locator since "id" is the first one in the list. If that element does not have an ID attribute, Selenium will next look for the "name" attribute since it is second in the list. The list goes on and on until an appropriate one is found.



Base URL Bar

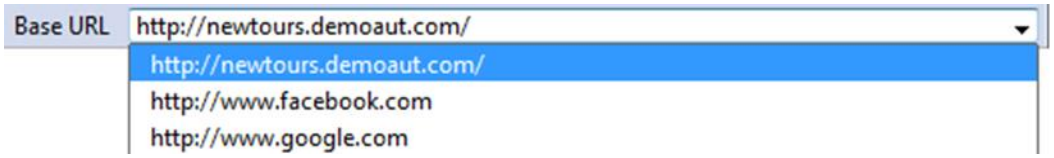


It has a dropdown menu that remembers all previous values for easy access.

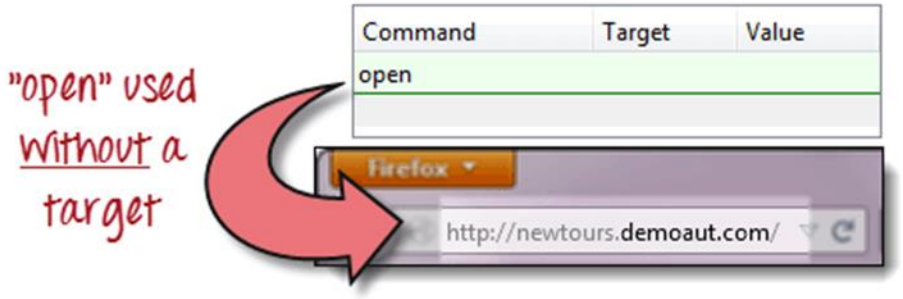
· The Selenese command **"open"** will take you to the URL that you specified in the Base URL.

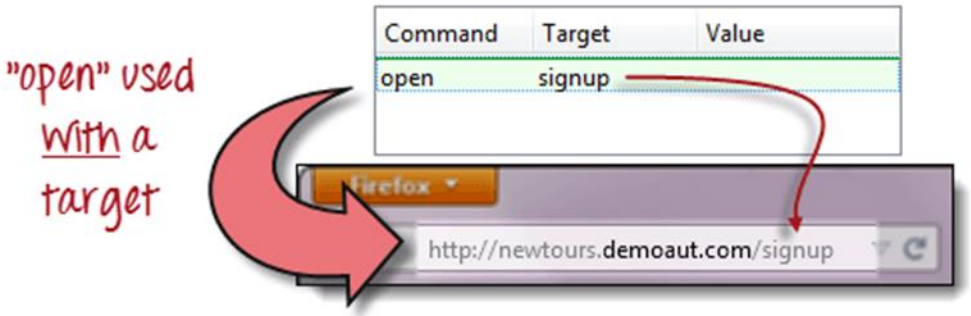
· In this tutorial series, we will be using https://demo.guru99.com/test/newtours/ as our Base URL. It is the site for Mercury Tours, a web application maintained by HP for web Testing purposes. We shall be using this application because it contains a complete set of elements that we need for the succeeding topics.

· The Base URL is very useful in accessing relative URLs. Suppose that your Base URL is set to https://demo.guru99.com/test/newtours/. When you execute the command "open" with the target value "signup," Selenium IDE will direct the browser to the sign-up page. See the illustration below.



Toolbar





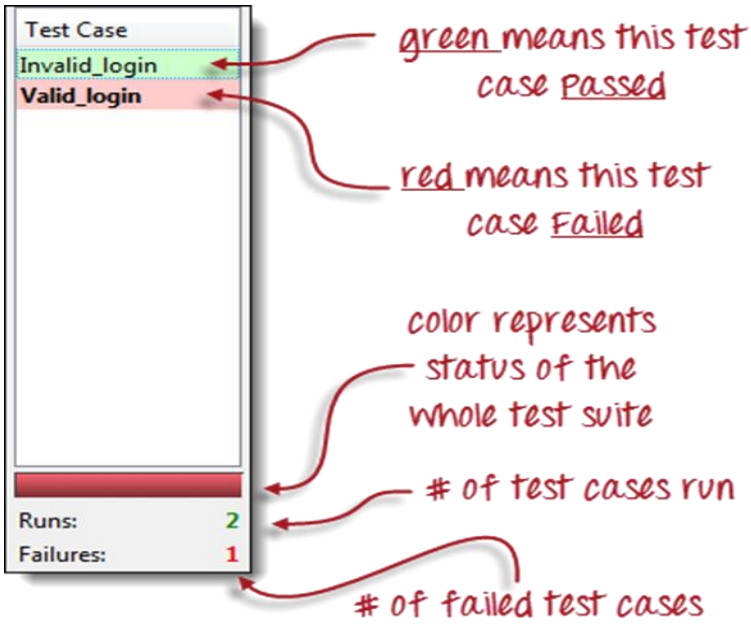
Playback Speed. This controls the speed of your Test Script Execution.
Record. This starts/ends your recording session. Each browser action is entered as a Selenese command in the Editor.
Play entire test suite. This will sequentially play all the test cases listed in the Test Case Pane.
Play current test case. This will play only the currently selected test case in the Test Case Pane.

Pause/Resume. This will pause or resume your playback.

Step. This button will allow you to step into each command in your test script.

Apply rollup rules. This is an advanced functionality. It allows you to group Selenese commands together and execute them as a single action.

Test Case Pane

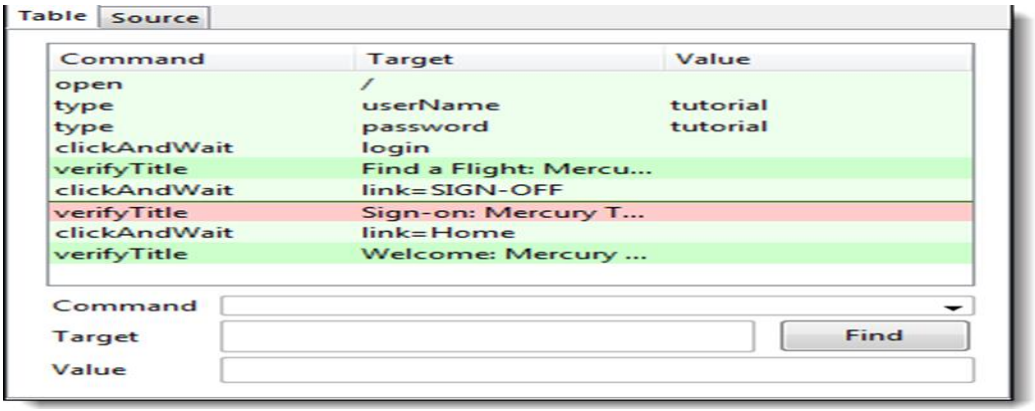


Editor

You can think of the editor as **the place where all the action happens**. It is available in two views: Table and Source.

Table View

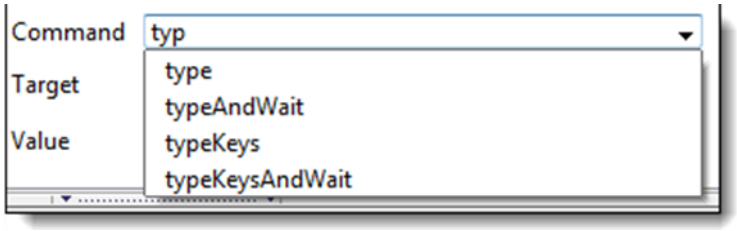
- Most of the time, you will work on Selenium IDE using the **Table View**.
- This is **where you create and modify Selenese commands.**
- After playback, each step is color-coded.



· To create steps, type the name of the command in the "Command" text box.

· **It displays a dropdown list of commands** that match with the entry that you are currently typing.

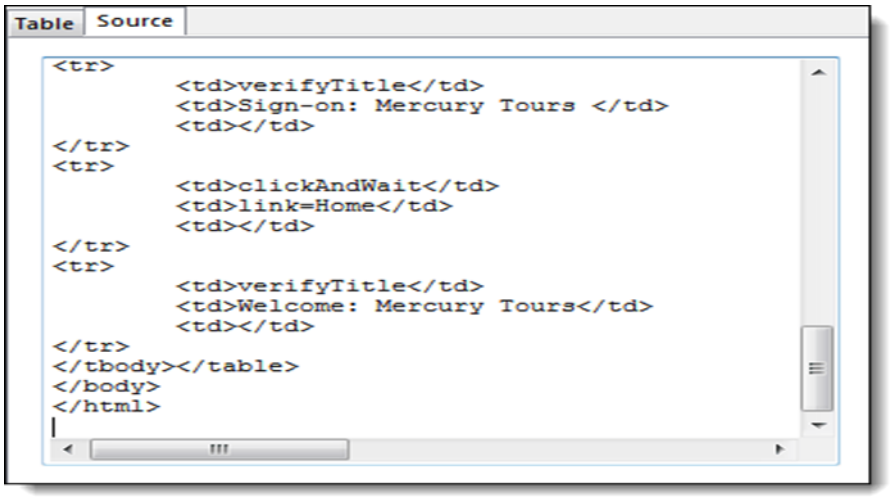
· Target is any parameter (like username, password) for a command and Value is the input value (like tom, 123pass) for those Targets.



Source View

· It displays the steps in HTML (default) format.

· It also allows you to edit your script just like in the Table View.



Log Pane

The Log Pane displays runtime messages during execution. It provides real-time updates as to what Selenium IDE is doing.

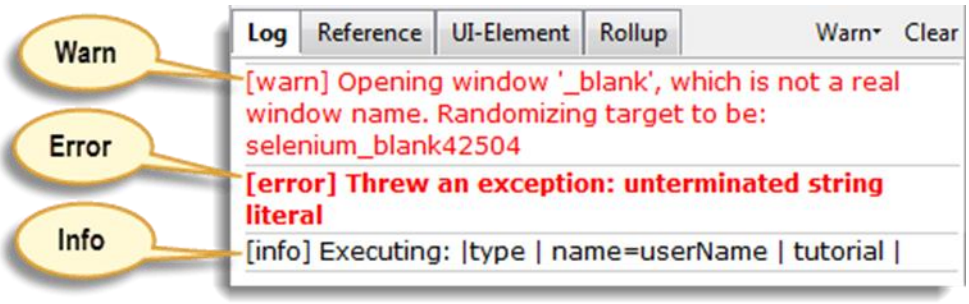
Logs are categorized into four types:

· Debug – By default, Debug messages are not displayed in the log panel. They show up only when you filter them. They provide technical information about what Selenium IDE is doing behind the scenes. It may display messages such as a specific module has done loading, a certain function is called, or an external JavaScript file was loaded as an extension.

· Info – It says which command Selenium IDE is currently executing.

· Warn – These are warning messages that are encountered in special situations.

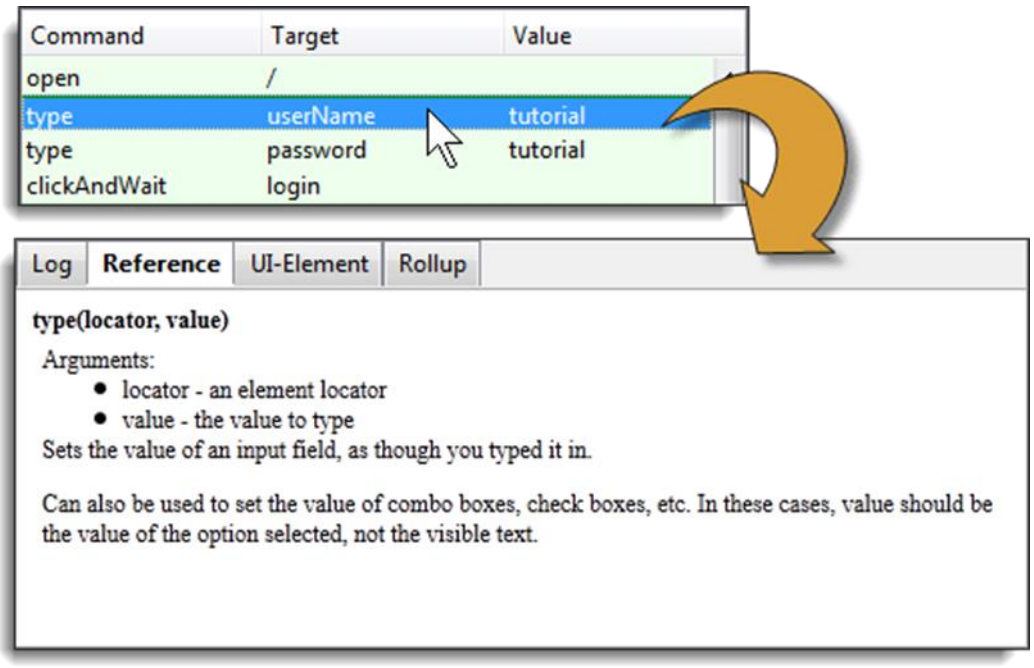
· Error – These are error messages generated when Selenium IDE fails to execute a command, or if a condition specified by "verify" or "assert" command is not met.



Logs can be filtered by type. For example, if you choose to select the "Error" option from the dropdown list, the Log Pane will show error messages only.

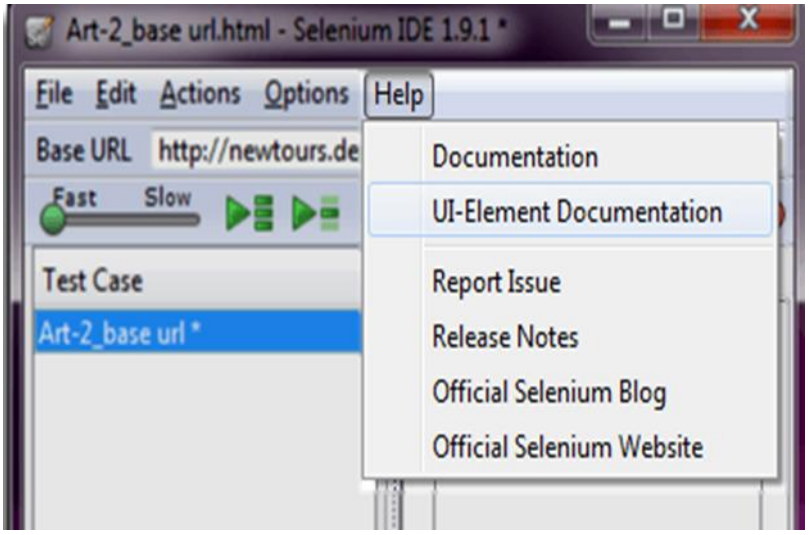
Reference Pane

The Reference Pane shows a concise description of the currently selected Sienese command in the Editor. It also shows the **description about the locator and value** to be used on that command.

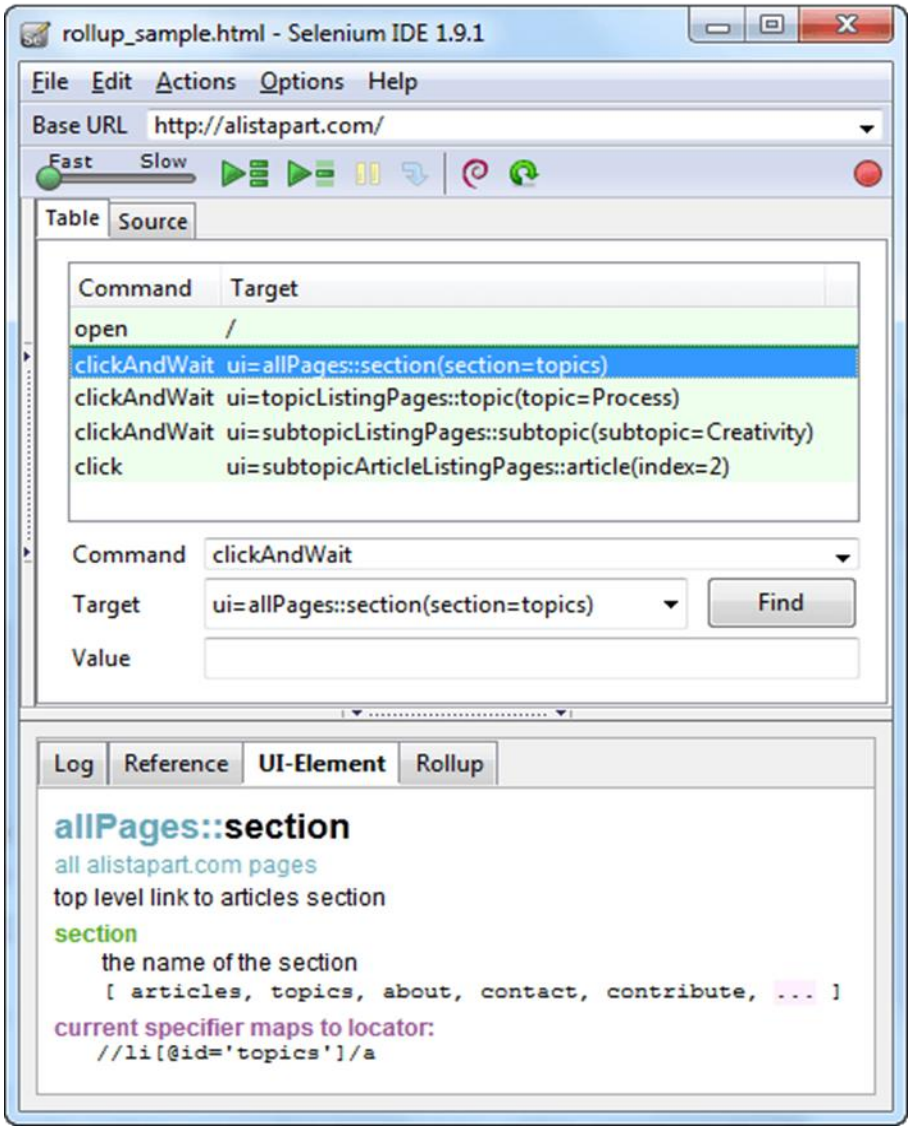


UI-Element Pane

The UI-Element is for advanced Selenium users. It uses JavaScript Object Notation (JSON) to define element mappings. The documentation and resources are found in the "UI Element Documentation" option under the Help menu of Selenium IDE



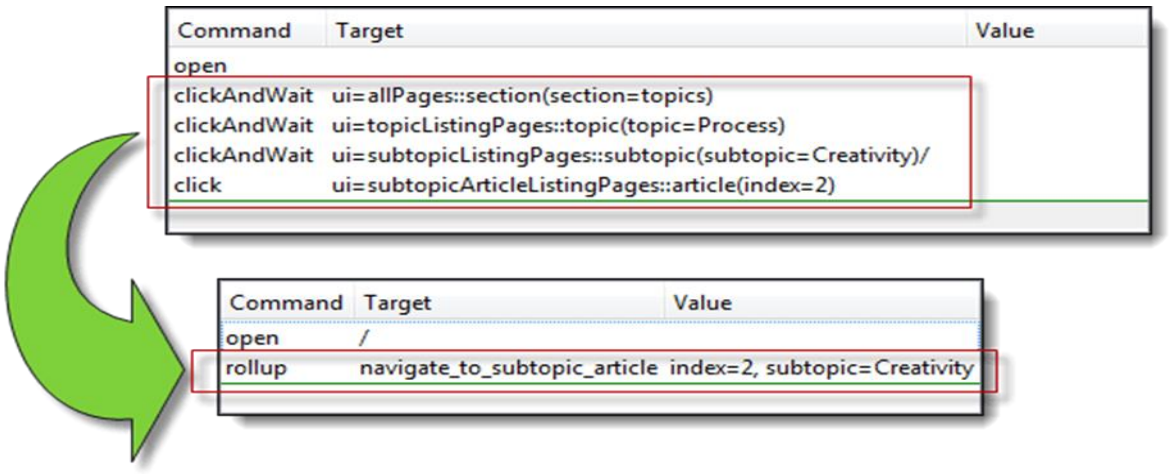
An example of a UI-element screen is shown below.

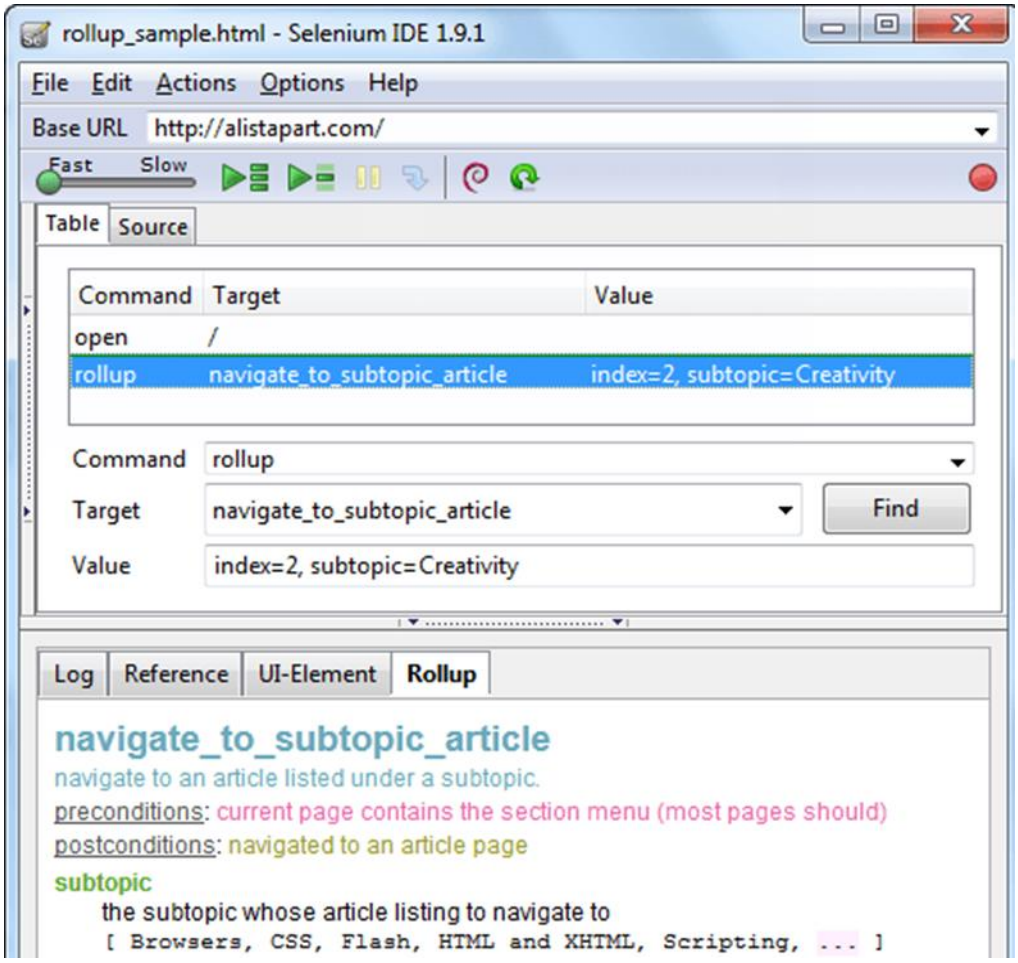


Rollup Pane

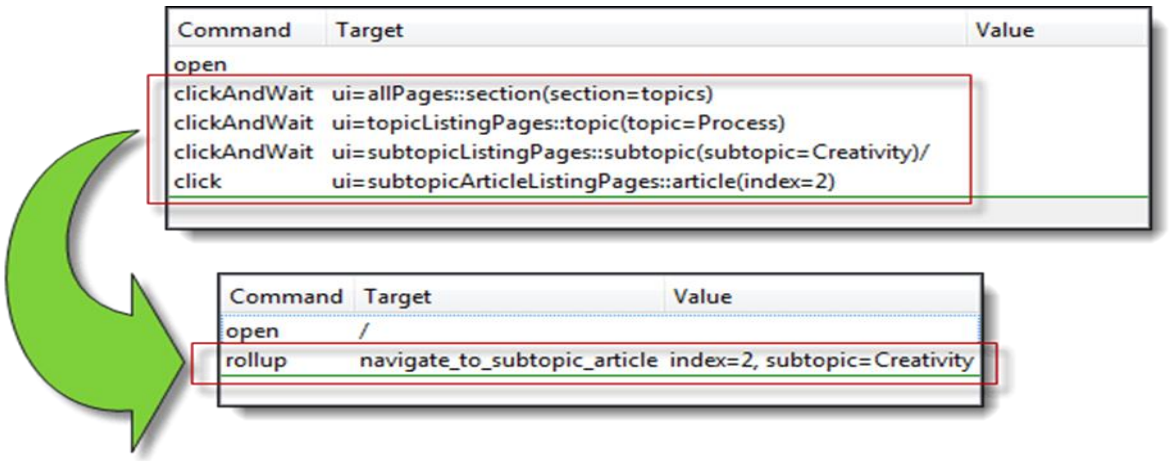
Rollup allows you to execute a group of commands in one step. A group of commands is simply called as a "rollup." It employs heavy use of JavaScript and UI-Element concepts to formulate a collection of commands that is similar to a "function" in programming languages.

Rollups are reusable; meaning, they can be used multiple times within the test case. Since rollups are groups of commands condensed into one, they contribute a lot in shortening your test script.





An example of how the contents of the rollup tab look like is shown below.



Summary

· Selenium IDE (Integrated Development Environment) is the simplest tool in the Selenium Suite.

· It must only be used as a prototyping tool.

· Knowledge of JavaScript and HTML is required for intermediate topics such as executing the "runScript" and "rollup" commands.A rollup is a collection of commands that you can reuse to shorten your test scripts significantly. Locators are identifiers that tell Selenium IDE how to access an element.

· Firebug (or any similar add-on) is used to obtain locator values.

· The menu bar is used in creating, modifying, and exporting test cases into formats useable by Selenium RC and WebDriver.

· The default format for Selenese commands is HTML.

· The "Options" menu provides access to various configurations for Selenium IDE.

· The Base URL is useful in accessing relative URLs.

· The Test Case Pane shows the list of currently opened test cases and a concise summary of test runs.

· The Editor provides the interface for your test scripts.

· The Table View shows your script in tabular format with "Command", "Target", and "Value" as the columns.

· The Source View shows your script in HTML format.

· The Log and Reference tabs give feedback and other useful information when executing tests.

· The UI-Element and Rollup tabs are for advanced Selenium IDE users only. They both require considerable effort in coding JavaScript.

· UI-Element allows you to conveniently map UI elements using JavaScript Object Notation (JSON).

To create a test suite in Selenium IDE:

1. Open Selenium IDE
2. Go to File>New Test Suite
3. To add test cases: Go to File>Add test case
4. Navigate to the location of your test case
5. Click on Add
6. Repeat steps 3-5 for more test cases

The Test suite that contains GoogleSignUpForm and GoogleSignUpErrors:

You can export the test suite to all the supported languages: HTML, Java, Groovy, C#, Perl, PHP, Phyton and Ruby.

Test suite exported as HTML

1. After you added test cases to your test suite, to export it as HTML, simply:
2. Go to File>Export as HTML
3. Enter a file name
4. Click on Save

In the HTML file, the test suite is a table in which each entry is a test case link:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <meta content="text/html; charset=UTF-8" http-equiv="content-type" />
  <title>Test Suite</title>
</head>
<body>
<table id="suiteTable" cellpadding="1" cellspacing="1" border="1" class="selenium"><tbody>
<tr><td><b>Test Suite</b></td></tr>
<tr><td><a href="file:///C:\SeleniumTestSuite\GoogleSignUpForm.html>GoogleSig-
nUpForm</a></td></tr>
```

<tr><td><a **href="file:///C:\SeleniumTestSuite\GoogleSignUpErrors.html**">GoogleSignUpErrors</td></tr>
</tbody></table>
</body>
</html>

You need to add "**file:///**" at the beginning of the path location of your test cases to run them from the command line, otherwise you will get this error:
"Firefox doesn't know how to open this address, because the protocol(c) isn't associated with any program"

Run a HTML test suite using Selenium-RC

After you generated your HTML suite file, you can run the suite using –htmlSuite command.

-htmlSuite requires you to specify:

- browserString (e.g. *firefox)
- startURL (e.g. "http://www.google.com)
- suiteFile (e.g. c:\absolute\path\to\my\HTMLSuite.html)
- resultFile (e.g. c:\absolute\path\to\my\results.html)

Follow these steps to run the suite:

- Create a HTML file that will keep your test results
- Open a command line
- cd to the selenium-server.jar location.

For example: cd C:\selenium-remote-control-1.0.3\selenium-server-1.0.3

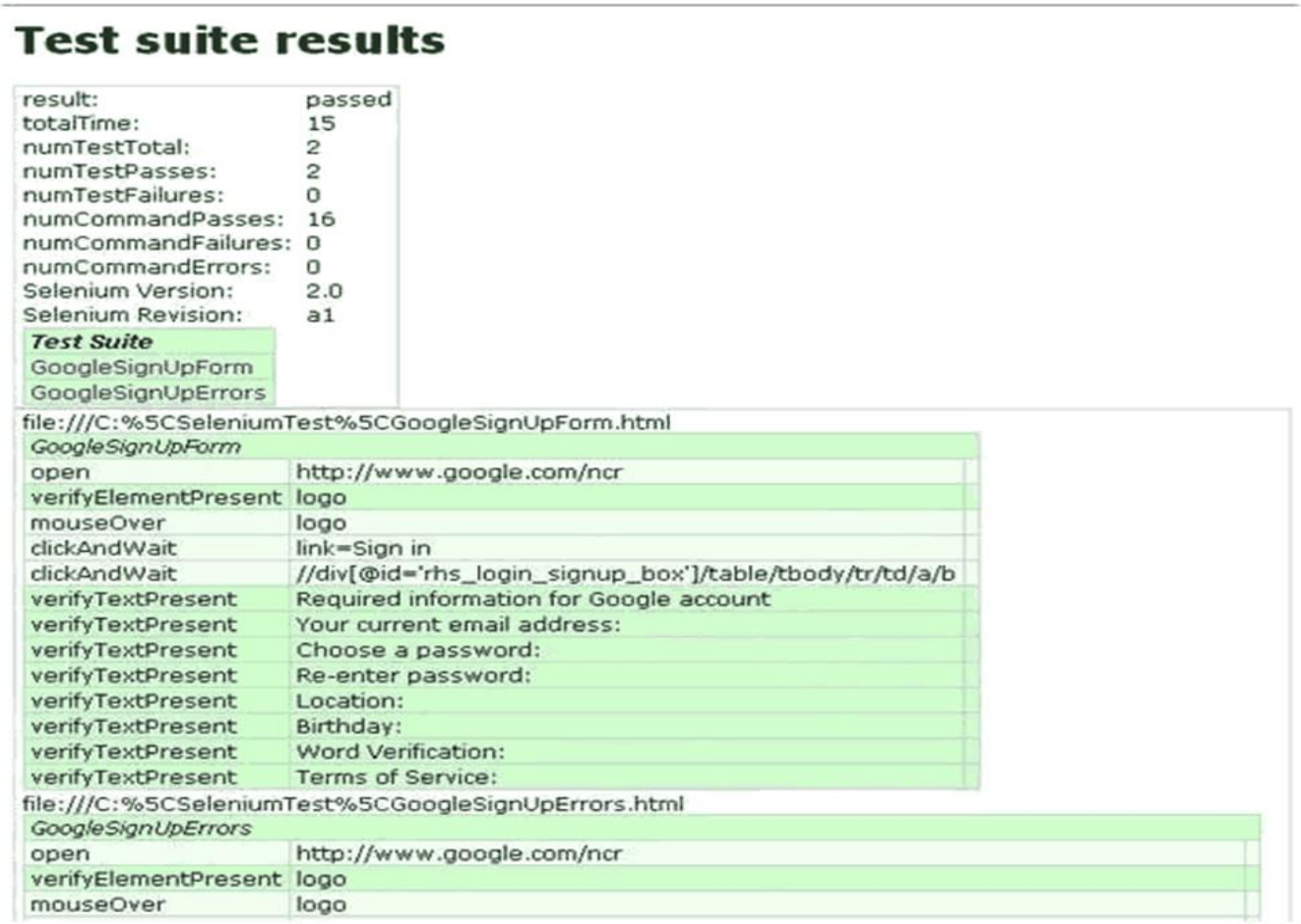
- Run the command with the required arguments. For example:

java -jar selenium-server.jar -port 4546 -htmlSuite *firefox "http://www.google.com" "c:\SeleniumTest\TestSuite.html" "C:\test.html"

You will see how Test runner is invoked and starts running the test in Firefox.

In case Test runner does not start running the test cases, ensure pop-up windows are enabled in your browser.

After the test script is done, you can check the test results in the HTML file you provided as argument.



Test suite results

result:	passed
totalTime:	15
numTestTotal:	2
numTestPasses:	2
numTestFailures:	0
numCommandPasses:	16
numCommandFailures:	0
numCommandErrors:	0
Selenium Version:	2.0
Selenium Revision:	a1

Test Suite
GoogleSignUpForm
GoogleSignUpErrors

file:///C:%5CSeleniumTest%5CGoogleSignUpForm.html

GoogleSignUpForm	
open	http://www.google.com/ncr
verifyElementPresent	logo
mouseOver	logo
clickAndWait	link=Sign in
clickAndWait	//div[@id='rhs_login_signup_box']/table/tbody/tr/td/a/b
verifyTextPresent	Required information for Google account
verifyTextPresent	Your current email address:
verifyTextPresent	Choose a password:
verifyTextPresent	Re-enter password:
verifyTextPresent	Location:
verifyTextPresent	Birthday:
verifyTextPresent	Word Verification:
verifyTextPresent	Terms of Service:

file:///C:%5CSeleniumTest%5CGoogleSignUpErrors.html

GoogleSignUpErrors	
open	http://www.google.com/ncr
verifyElementPresent	logo
mouseOver	logo

Learning outcomes (What I have learnt):

We have learned the writing a test suite using Selenium IDE.
We have learned the how to use Selenium IDE in The Firefox Browser.

Evaluation Grid (To be created per the faculty's SOP and Assessment guidelines):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Worksheet completion including writing learning objectives/Outcomes. (To be submitted at the end of the day).		
2.	Post-Lab Quiz Result.		
3.	Student Engagement in Simulation/Demonstration/Performance and Controls/Pre-Lab Questions.		
	Signature of Faculty (with Date):	Total Marks Obtained:	