The most impressive aspect of using selenium IDE is that

* the user is not required to possess any prior programming knowledge.
* The minimum that the user needs is the little acquaintances with HTML, DOMS and JavaScript.

Being a Firefox plug-in, Selenium IDE supports only Firefox. A few more loopholes make this tool inappropriate to be used for complex test scripts.

Thus, other tools like Selenium RC, WebDriver comes into the picture.

Install mozill firefox

Install selenium [http://seleniumhq.org/](http://www.seleniumhq.org/" \o "http://www.seleniumhq.org/" \t "_blank)

Download IDE in mozilla

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-2.jpg)

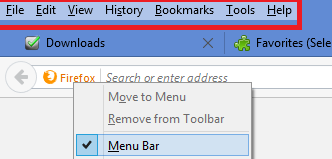
Or to

<https://addons.mozilla.org/en-US/firefox/>

Once the Firefox is booted and started again,

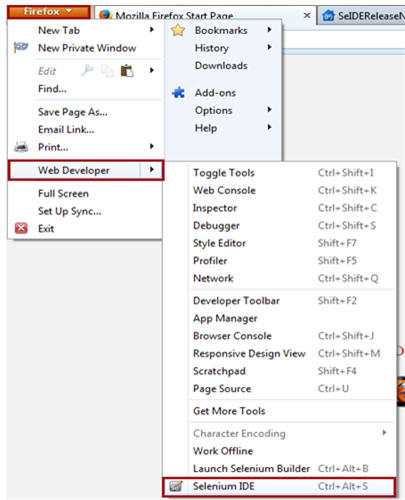
Right click over Firefox

Select Menu Bar, it will show at the top the menu bar

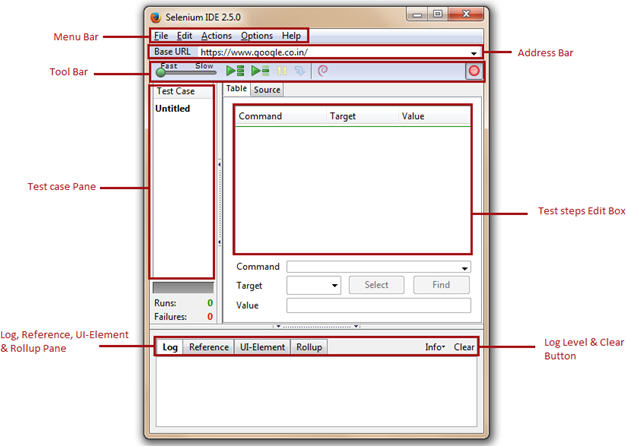


Go to tools

we can see selenium IDE indexed under menu bar -> Web Developer -> Selenium IDE.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-5-new.jpg)

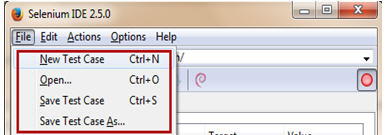
**Step #7:** As soon as we open Selenium IDE, the Selenium IDE window appears.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-6.jpg)

**#1. Menu Bar**

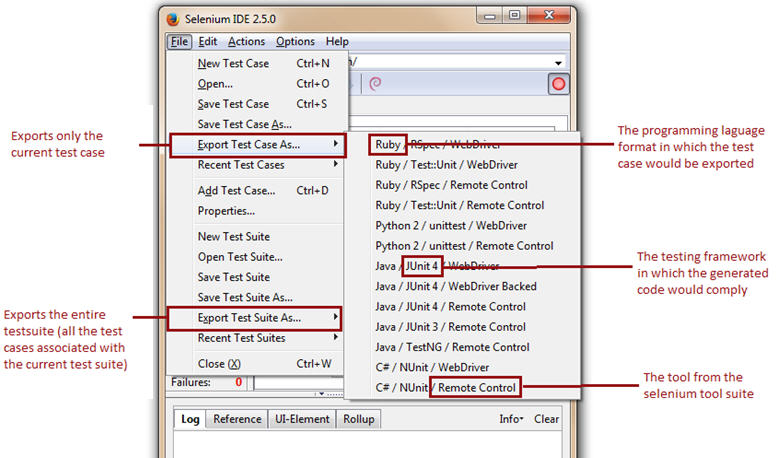
* File Menu
* Edit Menu
* Actions Menu
* Options Menu
* Help Menu

**A) File Menu**

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-7.jpg)

File Menu is very much analogous to the file menu belonging to any other application. It allows user to:

* Create new test case, open existing test case, save the current test case.
* Export Test Case As and Export Test Suite As in any of the associated programming language compatible with Selenium RC and WebDriver. It also gives the liberty to the user to prefer amid the available unit testing frameworks like jUnit, TestNG etc. Thus an IDE test case can be exported for a chosen union of programming language, unit testing framework and tool from the selenium package.
* Export Test Case As option exports and converts only the currently opened Selenium IDE test case.
* Export Test Suite As option exports and converts all the test cases associated with the currently opened IDE test suite.
* Close the test case.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-8.jpg)

**The Selenium IDE test cases can be saved into following format:**

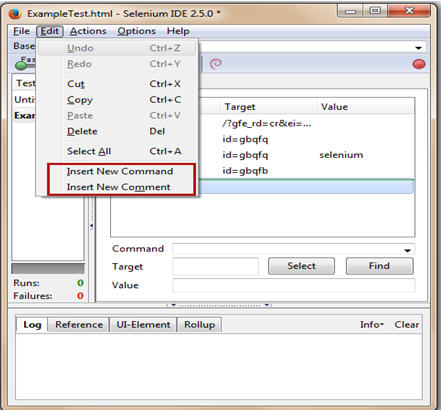
* HTML format

The Selenium IDE test cases can be exported into following formats/programming languages.

* java (IDE exported in Java)
* rb (IDE exported in Ruby)
* py (IDE exported in Python)
* cs (IDE exported in C#)

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-9.jpg)

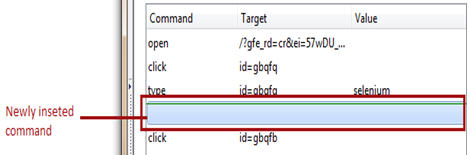
**B) Edit Menu**

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-10.jpg)

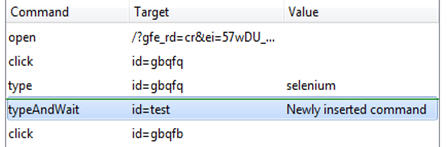
* Insert New Command
* Insert New Comment

**Insert New Command**

The new command would be inserted above the selected command/test step.

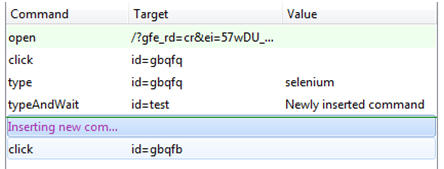
[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-11.jpg)

Now the user can insert the actual command action, target and value.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-12.jpg)

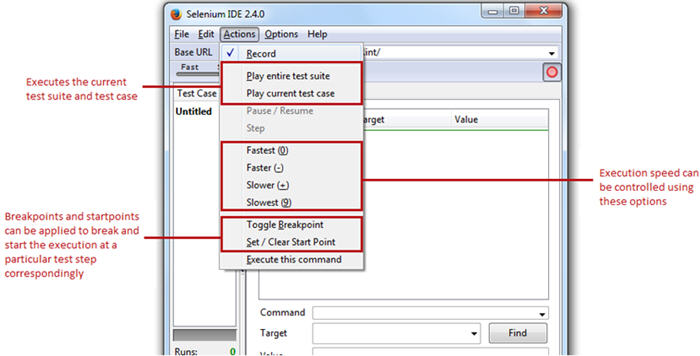
**Insert New Comment**

In the same way we can insert comments.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-13.jpg)

The purple color indicates that the text is representing a comment.

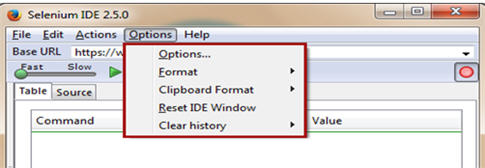
**C) Actions Menu**

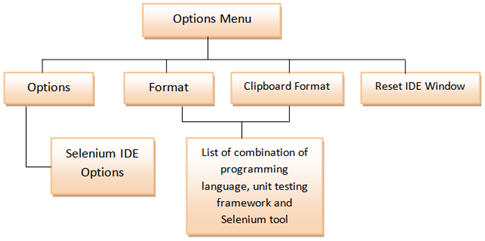
[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-14.jpg)

Actions Menu equips the user with the options like:

* **Record** – Record options fine tunes the Selenium IDE into the recording mode. Thus, any action made by the user on the Firefox browser would be recorded in IDE.
* **Play entire test suite** – The option plays all the Selenium IDE test cases associated with the current test suite.
* **Play current test case** – The option plays the current Selenium IDE test case that has been recorded/created by the user.
* **Pause / Resume –** User can Pause/Resume the test case at any point of time while execution.
* **Toggle Breakpoint** – User can set one or multiple breakpoint(s) to forcefully break the execution at any particular test step during execution.
* **Set / Clear Start Point** – User can also set start point at any particular test step for execution. This would enable user to execute the test case from the given start point for the subsequent runs.
* To deal with the page/element loads, the user can set the execution speed from fastest to lowest with respect to the responsiveness of the application under test.

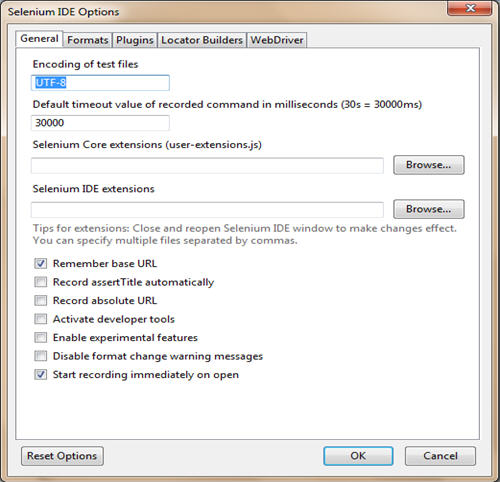
**D) Options Menu**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-15.jpg)

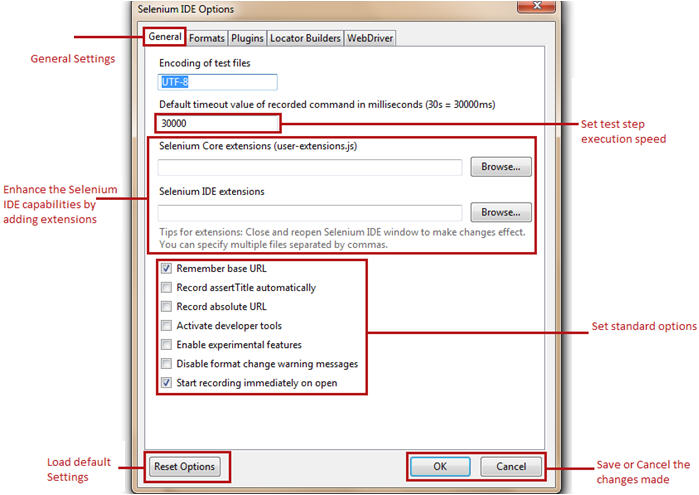
[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-16.jpg)

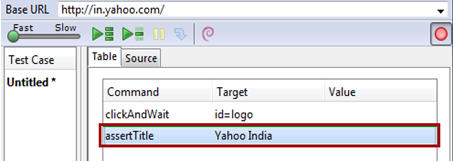
**Options**

Selenium IDE Options dialog box

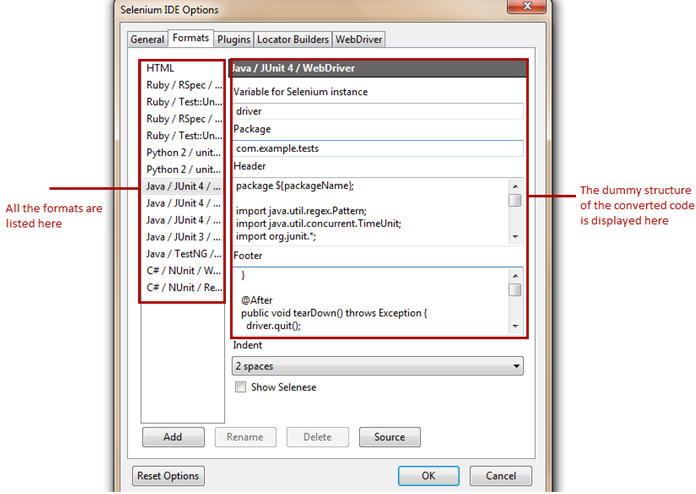
[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-17.jpg)

**General Settings**

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-19.jpg)

* **Default Timeout Value** – Default Timeout Value represents the time (in milliseconds) that selenium would wait for a test step to execute before generating an error. The standard timeout value is 30000 milliseconds i.e. 30 seconds.
* **Extensions** – Selenium IDE supports a wide range of extensions to enhance the capabilities of the core tool thereby multiplying its potential. These user extensions are simply the JavaScript files. They can set by mentioning their absolute path in the text boxes representing extensions in the Options dialog box.
* **Remember base URL** – Checking this option enables the Selenium IDE to remember the URL every time we launch it. Thus it is advisable to mark it checked. Un-checking this option will leave the base URL field as blank and it will be re-filled only when we launch another URL on the browser.
* **Record assertTitle automatically** – Checking this field inserts the assertTitle command automatically along with the target value for every visited web page.
* [](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-18.jpg)
* **Enable experimental features –** Checking this field for the first time imports the various available formats into the Selenium IDE.

**Formats**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-20.jpg)

Formats tab displays all the available formats with selenium IDE. User is levied with the choice to enable and disable any of the formats. Refer the following figure.

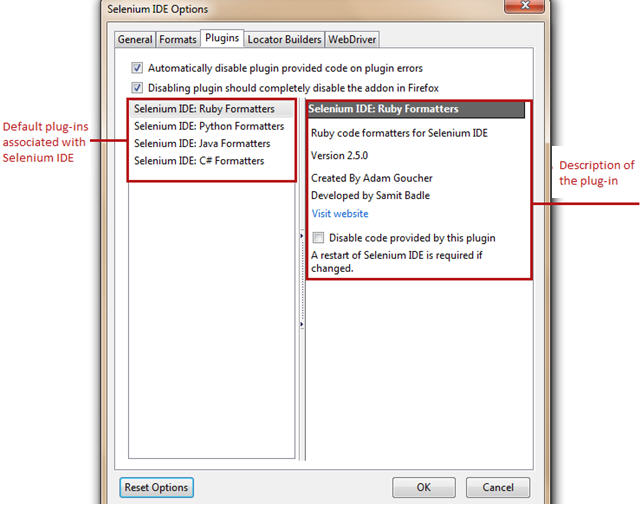
**Selenium IDE Plugins**

Plug-ins tab displays the supported Firefox plug-ins installed on our instance of Selenium IDE. There are a [number of plug-ins](http://docs.seleniumhq.org/projects/ide/plugins.jsp) available to cater different needs, thus we can install these add-ons like we do other plug-ins. One of the recently introduced plug-in is “File Logging”. In the end of this tutorial, we will witness how to install and use this plug-in.

With the standard distribution, Selenium IDE comes with a cluster of following plug-ins:

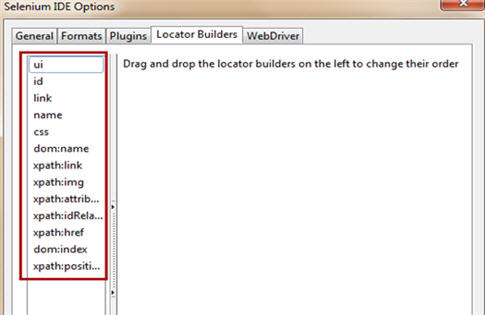
* Selenium IDE: Ruby Formatters
* Selenium IDE: Python Formatters
* Selenium IDE: Java Formatters
* Selenium IDE: C# Formatters

These formatters are responsible to convert the HTML test cases into the desired programming formats.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-21.jpg)

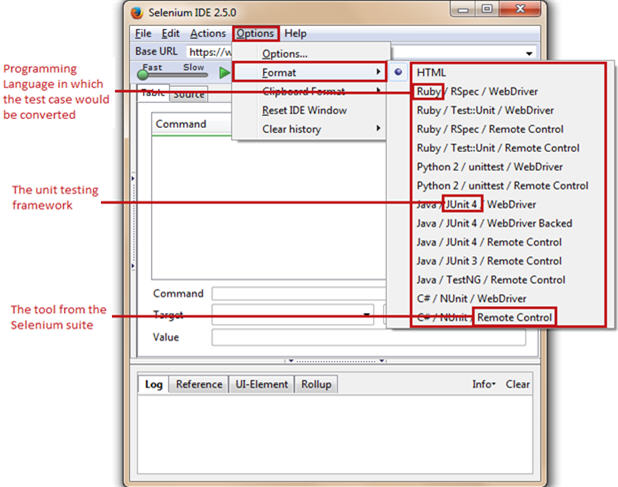
**Locator Builders**

Locator builders allow us to prioritize the order of locator types that are generated while recording the user actions. Locators are the set of standards by which we uniquely identify a web element on a web page.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-22.jpg)

**Formats**

Formats option allows user to convert the Selenium IDE test case (selenese commands) into desired format.

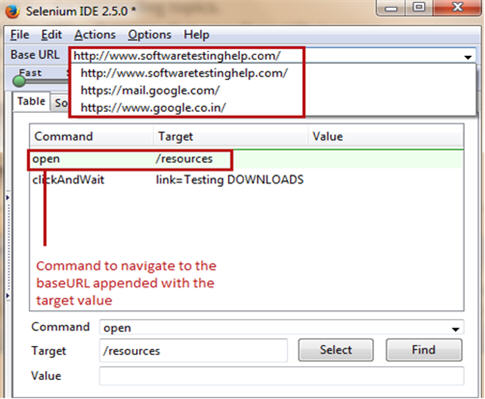
[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-23.jpg)

**E) Help Menu**

As Selenium has a wide community and user base, thus various documentations, release notes, guides etc.

**#2. Base URL Bar**

Base URL bar is principally same as that of an address bar. It remembers the previously visited websites so that the navigation becomes easy later on.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-24.jpg)

Now, whenever the user uses “open” command of Selenium IDE without a target value, the base URL would be launched on to the browser.

**Accessing relative paths**

To access relative paths, user simply needs to enter a target value like “/download” along with the “open” command. Thus, the base URL appended with “/downloads” (http://docs.seleniumhq.org/resources) would be launched on to the browser. The same is evident in the above depiction.

**#3. Toolbar**

[Selenium IDE 25](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-25.jpg)

Toolbar provides us varied options pertinent to the recording and execution of the test case.

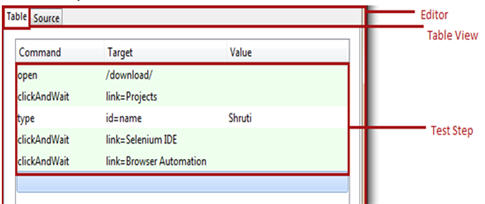
* **[Selenium IDE 26](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-26.jpg) Playback Speed** –to control the test case execution speed from fast to slow.
* **[Selenium IDE 27](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-27.jpg) Play test suite** –to execute all the test cases belonging to the current test suite sequentially.
* **[Selenium IDE 28](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-28.jpg) Play test case** –to execute the currently selected test case.
* **[Selenium IDE 29](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-29.jpg) Pause** –to pause the current execution.
* **[Selenium IDE 30](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-30.jpg) Step** –to step into the test step.
* **[Selenium IDE 31](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-31.jpg) Rollup**–to combine multiple test steps to act like a single command.
* [Selenium IDE 32](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-32.jpg) Record –to start/stop the recording of user actions. The hollow red ball indicates the start of the recording session whereas the solid red ball indicates the end of the recording session. By default, the Selenium IDE opens in the recording mode.

**#4. Editor**

Editor is a section where IDE records a test case. Each and every user action is recorded in the editor in the same order in which they are performed.

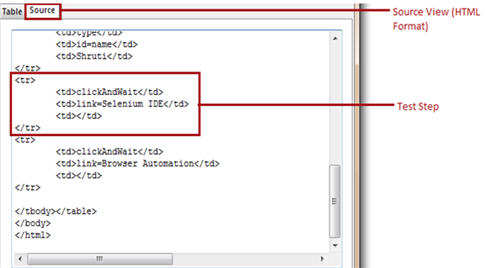
**The editor in IDE has two views, namely:**

**1) Table View**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-33.jpg)

It is the default view provided by Selenium IDE. The test case is represented in the tabular format. Each user action in the table view is a consolidation of “Command”, “Target” and “Value” where command, target and value refers to user action, web element with the unique identification and test data correspondingly. Besides recording it also allows user to insert, create and edit new Selenese commands with the help of the editor form present in the bottom.

**2) Source View**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-34.jpg)

The test case is represented in the HTML format.

Each test step is considered be a row <tr> which is a combination of command, target and value in the separate columns <td>.

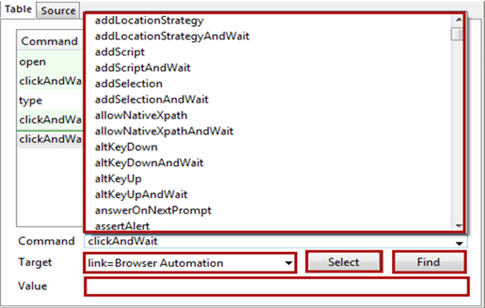
Like any HTML document, more rows and columns can be added to correspond to each Selenese command.

**Editor Form** to type any command.

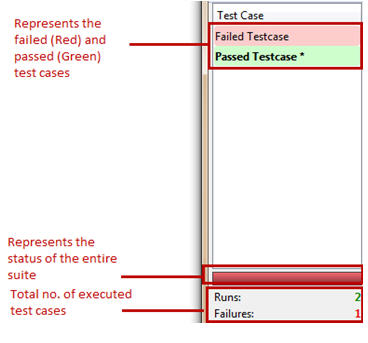
Select button lets the user to select any web element and its locator would be fetched automatically into the target field.

Find button lets the user find the web element on the web page against a defined target.

Value is the test input data entered into the targets with which we want to test the scenario.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-35.jpg)

**#5. Test case pane**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-36.jpg)

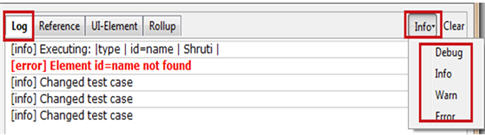
At the instance we open Selenium IDE interface, we see a left container titled “Test case” containing an untitled test case. Thus, this left container is entitled as Test case pane.

Test case pane contains all the test cases that are recorded by IDE. The tool has a capability of opening more than one test case at the same time under test case pane and the user can easily shuffle between the test cases. The test steps of these test cases are organized in the editor section.

Selenium IDE has a color coding ingredient for reporting purpose. After the execution, the test case in marked either in “red” or “green” color.

* Red color symbolizes the unsuccessful run i.e. failure of the test case.
* Green color symbolizes the successful run of the test case
* It also layouts the summary of the total number of test cases executed with the number of failed test cases.
* If we execute a test suite, all the associated test cases would be listed in the test case pane. Upon execution, the above color codes would be rendered accordingly.

**#6. Log Pane**

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-37.jpg)

Log pane gives the insight about current execution in the form of messages along with the log level in the real time. Thus, log messages enable a user to debug the issues in case of test case execution failures.

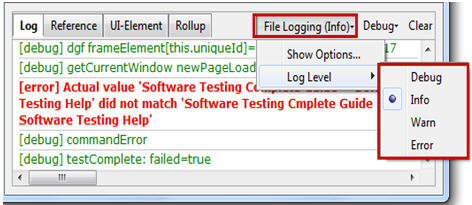
**The printing methods / log levels used for generating logs are:**

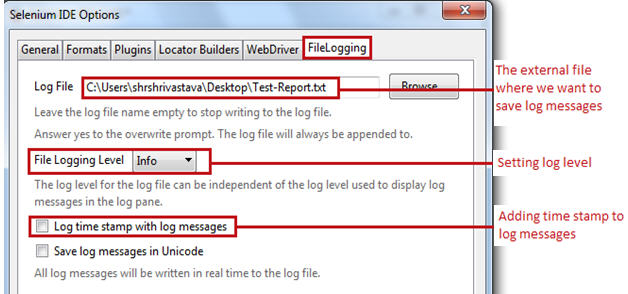
* Error – Error message gives information about the test step failure. It may be generated in the cases when element is not found, page is not loaded, verification/assertion fails etc.
* Warn – Warning message gives information about unexpected conditions.
* Info – Info message gives information about current test step execution.
* Debug – Debug messages gives information about the technicalities in the backdrop about the current test step.

Logs can be filtered with the help of a drop down located at the top-right corner of the footer beside the clear button. Clear button erases all the log messages generated in the current or previous run.

**Generating Logs in an external medium**

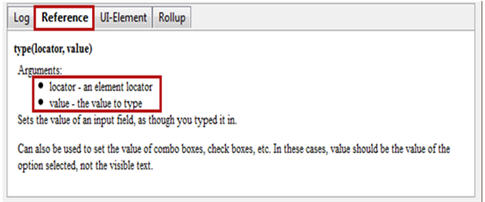
Recently introduced “File Logging” plug-in enables the user to save log messages into an external file. File Logging can be plugged in to IDE like any other plug-in. Upon installation, it can be found as a tab named “File Logging” in the footer beside the Clear button.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-38.jpg)

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-39.jpg)

**Reference Pane**

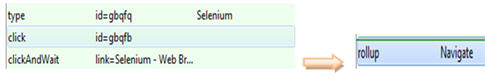
Reference Pane gives the brief description about the currently selected Selenese command along with its argument details.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-40.jpg)

**UI-Element Pane**

UI – Element Pane allows Selenium user to use JavaScript Object Notation acronym as JSON to access the page elements. More on this can be found in UI-Element Documentation under Help Menu.

**Rollup Pane**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-IDE-41.jpg)

Rollup Pane allows the user to roll up or combine multiple test steps to constitute a single command termed as “rollup”. The rollup in turn can be called multiple times across the test case.

**Conclusion**

Through this tutorial, our objective was to make you familiar and accustomed with the basic terminologies and nomenclatures of Selenium IDE. We also presented a detailed study on all the features of Selenium IDE.

**Here are the cruxes of this tutorial:**

* Selenium IDE is an automated testing tool which supports record and play back.
* User is not required to have any prior programming knowledge except for the basic understanding of HTML, JavaScript and DOM.
* The menu bar allows user to create, save, edit and convert the recorded Selenium IDE test scripts. It also allows user to set formats and plug-ins.
* Toolbar allows user to set the test execution speed, to pause and resume test case, to roll up commands etc.
* Roll ups combines more than one test step and thus the rolled up commands acts and executes as a single command.
* Editor allows user to record or create test scripts. Editor has two views “table” and “source”.
* In table view, each test step is comprised of a command, target and a value.
* Source view displays the test case in the HTML format.
* Test case pane shows a comprehensive list of failed and passed test cases with the relevant color-coding.
* Log Pane displays the test execution heath in the form of message.
* Log messages can be saved in a file using “File Logging” plug-in.
* Reference pane shows the description of every selected command.
* UI-Element and Rollup are generally used while creating advance Selenium IDE scripts.

[**Next Tutorial #3**](http://www.softwaretestinghelp.com/selenium-ide-script-selenium-tutorial-3/)**:** Now that we are acquainted and comfortable with Selenium IDE and its features, in the next tutorial we would practice these features by creating our own test script using Selenium IDE.\