**Introduction to Selenium IDE**

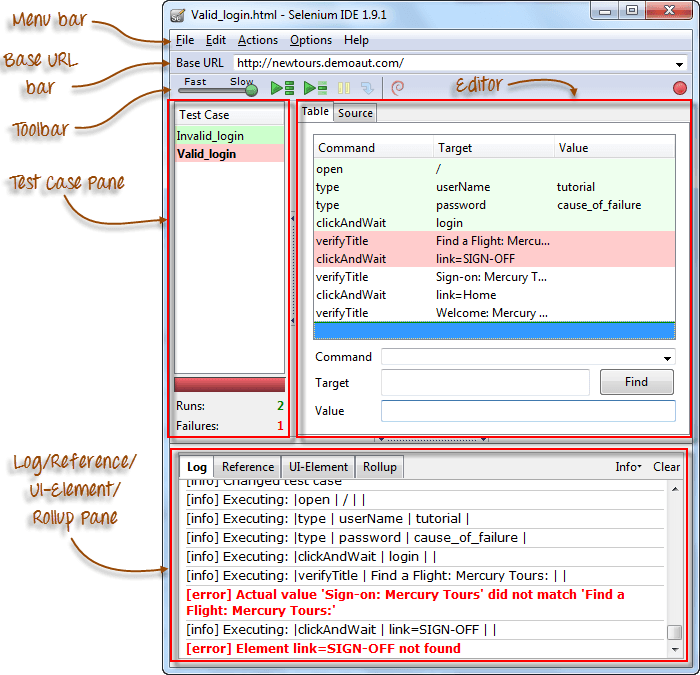
**Selenium IDE (Integrated Development Environment) is the simplest tool in the Selenium Suite**. It is **a Firefox add-on that creates tests very quickly through its record-and-playback functionality**.

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"**.

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.

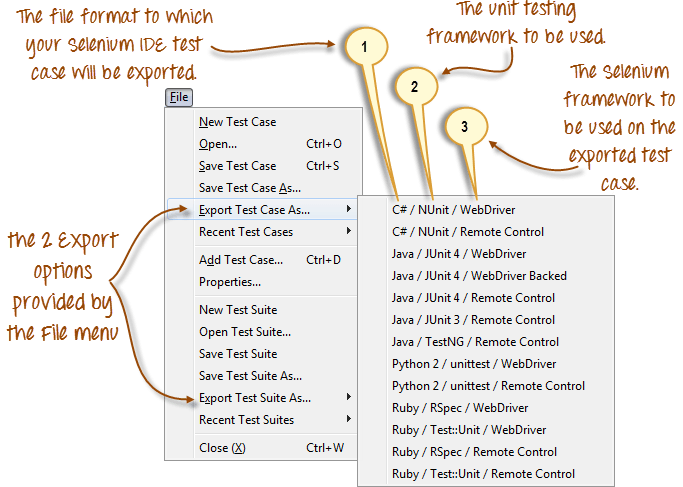
**Features of Selenium IDE**[](http://www.guru99.com/images/ide_-_labelled_parts.png)

**Menu Bar**

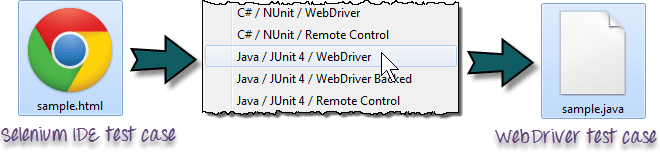
It is located at the **top most portion** of the IDE. The most commonly used menus are the File, Edit, and Options menus.

**File menu**

* 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.

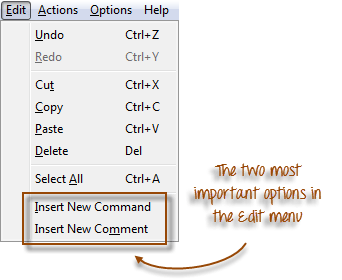
[](http://www.guru99.com/images/file_menu_b.png)

* 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)

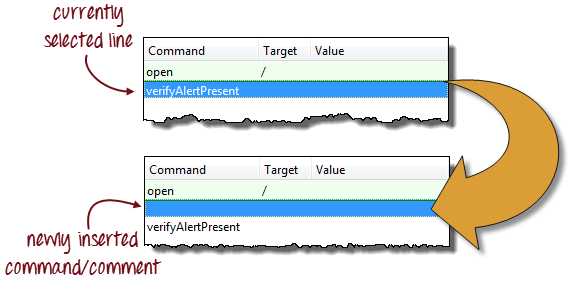
[](http://www.guru99.com/images/html_to_java_conversion.png)

**Edit Menu**

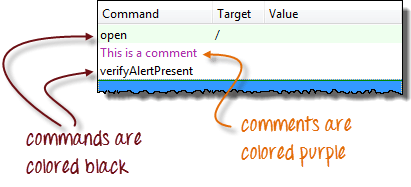
* It contains usual options like Undo, Redo, Cut, Copy, Paste, Delete, and Select All.
* The two most important options are the "**Insert New Command**" and "**Insert New Comment**".

[](http://www.guru99.com/images/edit_menu_-_most_impt.png)

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

[](http://www.guru99.com/images/insertion_-_before_and_after.png)

* **Commands**are colored **black**.
* **Comments**are colored **purple.**

[](http://www.guru99.com/images/command_and_comment_color.png)

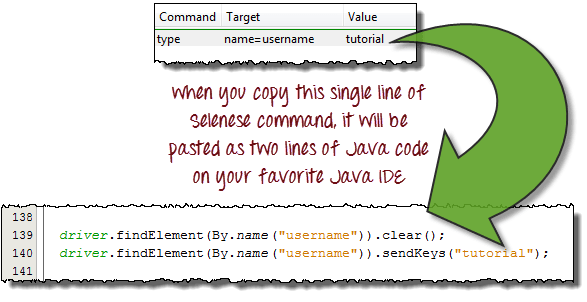
**Options menu**

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

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

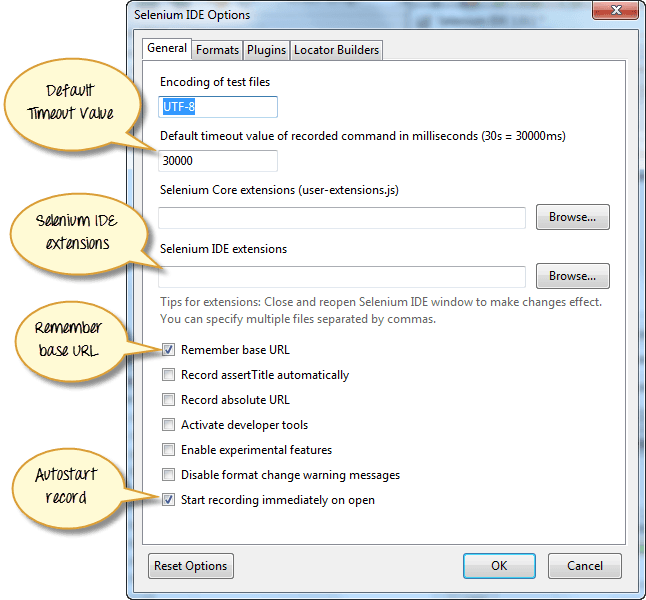
|  |  |
| --- | --- |
| [http://cdn.guru99.com/images/clipboard_format_menu_list_complete.png](http://www.guru99.com/images/clipboard_format_menu_list_complete.png) | **Clipboard Format**     * **The Clipboard Format allows you to copy a Selenese command from the editor and paste it as a code snippet**. * 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 below.

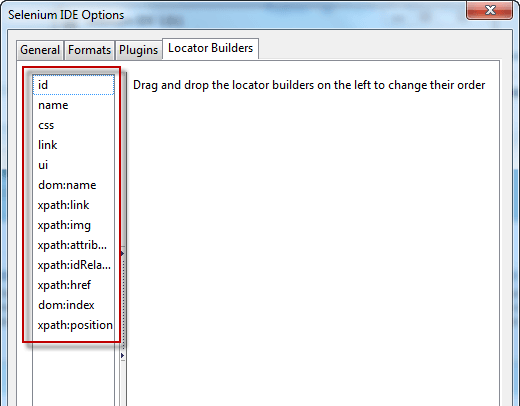
[](http://www.guru99.com/images/clipboard_format_-_copy_paste_illustration.png)

**Selenium IDE Options dialog box**

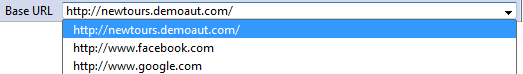
You can launch the Selenium IDE Options dialog box by clicking Options > Options... on the menu bar. Though there are many settings available, we will concentrate on the few important ones.

[](http://www.guru99.com/images/options_dialog_with_lables.png)

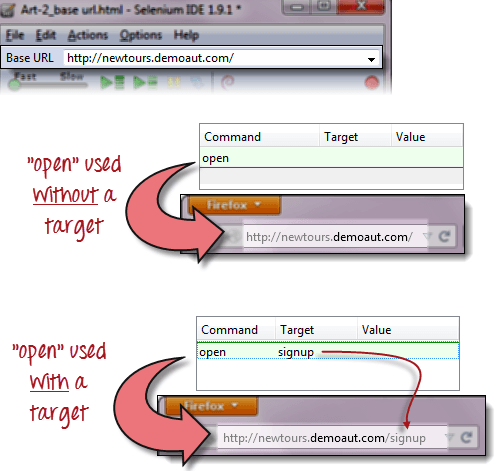
* **Default Timeout Value**. This refers to the time that Selenium has to wait for a certain element to appear or become accessible before it generates an error.  **Default timeout value is 30000ms**.
* **Selenium IDE extensions**. This is where you specify the extensions you want to use to extend Selenium 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.

[](http://www.guru99.com/images/locator_builders(1).png)

**Base URL Bar**

[](http://www.guru99.com/images/BaseURL.png)

* 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 [http://newtours.demoaut.com](http://newtours.demoaut.com/) as our Base URL. It is the site for Mercury Tours, a web application maintained by HP for web [testing](http://www.guru99.com/software-testing.html) 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 [http://newtours.demoaut.com](http://newtours.demoaut.com/). 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.

[](http://www.guru99.com/images/BaseURLTarget.png)

**Toolbar**

|  |  |
| --- | --- |
| [http://cdn.guru99.com/images/fast-slow.png](http://www.guru99.com/images/fast-slow.png) | **Playback Speed**. This controls the speed of your Test Script Execution. |
| [http://cdn.guru99.com/images/record.png](http://www.guru99.com/images/record.png) | **Record.**This starts/ends your recording session.  Each browser action is entered as a Selenese command in the Editor. |
| [http://cdn.guru99.com/images/play_entire_test_suite.png](http://www.guru99.com/images/play_entire_test_suite.png) | **Play entire test suite**. This will sequentially play all the test cases listed in the Test Case Pane. |
| [http://cdn.guru99.com/images/play_current_test_case.png](http://www.guru99.com/images/play_current_test_case.png) | **Play current test case**. This will play only the currently selected test case in the Test Case Pane. |
| [http://cdn.guru99.com/images/pause_resume_2.png](http://www.guru99.com/images/pause_resume_2.png) | **Pause/Resume**. This will pause or resume your playback. |
| [http://cdn.guru99.com/images/step.png](http://www.guru99.com/images/step.png) | **Step**. This button will allow you to step into each command in your test script. |
| [http://cdn.guru99.com/images/rollup_button.png](http://www.guru99.com/images/rollup_button.png) | **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**

|  |  |
| --- | --- |
| [http://cdn.guru99.com/images/test_case_pane_-_final.png](http://www.guru99.com/images/test_case_pane_-_final.png) | * In Selenium IDE, you can open **more than one test case at a time**. * **The test case pane shows you the list of currently opened test cases.** * When you open a test suite, the test case pane will **automatically list all the test cases** contained in it. * The test case written in **bold font** is the **currently selected test case** * After playback, **each test case is color-coded** to represent if it passed or failed.   + Green color means "Passed."   + Red color means "Failed." * At the bottom portion is a summary of the number of test cases that were run and failed. |

**Editor**

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

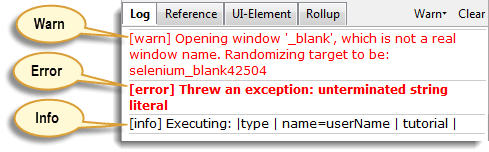
|  |  |
| --- | --- |
| [http://cdn.guru99.com/images/Editor.png](http://www.guru99.com/images/Editor.png) | **Table View**     * Most of the time, you will work on Selenium IDE using the **Table View**. * This is **where you create and modify Selenese commands.** * After playback, each step is color-coded. |
| [http://cdn.guru99.com/images/TableEditor.png](http://www.guru99.com/images/TableEditor.png) | * 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. |
| [http://cdn.guru99.com/images/editor_source_-_final.png](http://www.guru99.com/images/editor_source_-_final.png) | **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**

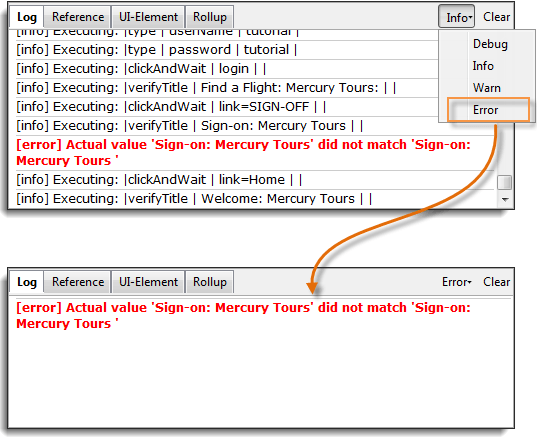
**The Log Pane displays runtime messages**during execution. It provides real-time updates as to what Selenium IDE is doing.

**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.

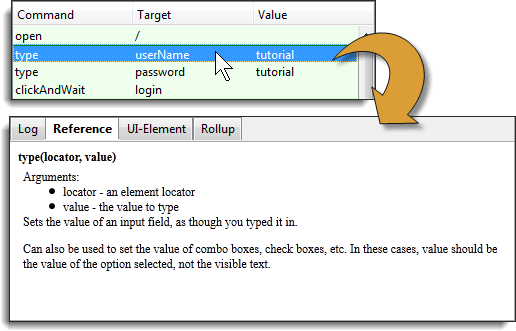
[](http://www.guru99.com/images/Log_pane_-_types_of_logs.png)

**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.

[](http://www.guru99.com/images/ErrorLog.png)

**Reference Pane**

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

[](http://www.guru99.com/images/complete.png)

**Additional Information**

Selenium-IDE will automatically insert commands into your test case based on your actions. Typically, this will include:

* clicking a link - *click* or *clickAndWait* commands
* entering values - *type* command
* selecting options from a drop-down listbox - *select* command
* clicking checkboxes or radio buttons - *click* command

## Commonly Used Selenium Commands

To conclude our introduction of Selenium, we’ll show you a few typical Selenium commands. These are probably the most commonly used commands for building tests.

open

opens a page using a URL.

click/clickAndWait

performs a click operation, and optionally waits for a new page to load.

verifyTitle/assertTitle

verifies an expected page title.

verifyTextPresent

verifies expected text is somewhere on the page.

verifyElementPresent

verifies an expected UI element, as defined by its HTML tag, is present on the page.

verifyText

verifies expected text and its corresponding HTML tag are present on the page.

verifyTable

verifies a table’s expected contents.

waitForPageToLoad

pauses execution until an expected new page loads. Called automatically when clickAndWait is used.

captureEntirePageScreenshot

PageScreenShot is stored as .png file and the path has to be specified in the target.

**Advanced Selenium IDE**

**Selenium IDE Options**

**Selenium IDE options can be set.**

**General:** can specify the time in millisecond.

* Remember baseUrl.
* Can assert Title automatically.
* Record absolute URL.
* Start recording immediately on open

**Formats:** By default, the Selenium IDE can be only in HTML format.

**Locators**: The order of the locators can be set by the users.

* Id
* Name
* CSS
* linkText
* Xpath.

WEBDRIVER:

The Selenium IDE file can be exported to many languages

Java,

Python,

Php

Ruby

**It can be run on Eclipse IDE downloading the selenium jar files.**