

## Selenium Tutorial(IDE, Core, RC)

### CONTENT

#### About Selenium

1. Origin of Selenium
2. About Selenium
  - a. What is selenium
  - b. Advantage & Disadvantage of selenium
  - c. variants of selenium
3. General Concepts of Functional testing
4. General Concepts of Compatibility testing

#### Installation instruction of Selenium IDE

Mozilla Firefox – prerequisites

Download the selenium IDE

Firebug -Addon

#### Tutorial on Selenium IDE

1. Introduction
2. Sample test
3. How to use firebug
4. Advantages and Disadvantages of Selenium IDE

#### Installation instruction of Selenium Core

1. Install webserver (for ex: Install any of these webserver IIS, Apache, Tomcat)
2. Download Selenium Core
3. Working in Seleniumcore

#### Installation instruction of Selenium Remote Control(RC):

1. JDK 1.5 or greater version
2. Download Selenium Remote control

#### Tutorial on Selenium Remote Control

1. Introduction
2. working in command prompt
3. Selenium RC with multiple users
4. Integrating Selenium RC with Eclipse
  1. Installing Eclipse
  2. Configuring Eclipse with Selenium RC
5. Sample program
6. Supporting tools for Selenium RC (bromine, cubic test,etc..,)
7. Using Selenium RC with Multiple Users

#### Platforms supported by selenium

1. Browsers
2. Operating systems
3. Programming language
4. Testing Frameworks

#### Conclusion

## Selenium Tutorial(IDE, Core, RC)

### **About Selenium:**

Thought work is a global IT outsourcing company which focuses on Agile software development. The company has contributed many open source product and one of the product is called Selenium.

### **Selenium Contributors**

The story of Selenium starts at 2004 in chicago by Jason Huggin buildings core made as “Javascript Test Runner”. They created this core for the testing of an internal time & expenses application written in (python, plone). Automatic testing of any applications is core to ThoughtWork's style, given the Agile leanings of this consultancy. He has help from Paul Gross and Jie Tina Wang. For them, this was a day job.

Later Jason started giving demo to other colleague in his team and people got impressed by the testing framework & he got very good feedback for this work.

Soon after in 2004 fellow ThoughtWorker Paul Hammant saw the demo, and started discussions about the open sourcing of selenium, as well as defining a “driven” mode of Selenium where you'd get to use selenium over the wire from a language of your choice, that would get around the 'same origin policy'. Another colleague Aslak Hellesoy, experimented with different ideas for the 'Server' piece, including page rewriting to get around the same origin policy. Paul wrote the original server piece in Java and Aslak and obie fernandez ported that the client driver to Ruby setting the foundation for drivers in yet more languages.

### **Outside of ThoughtWork:**

At Bea, Dan Fabulich and Nelson Sproul came to the conclusion that the driver/server to browser architecture was not the most useful or flexible, so forked the driver coder and crafted that into a standalone server that leveraged and bundled MortBay's Jetty as a web-proxy. When the code was merged back it became known as "Selenium Remote Control" and the old driven codeline and capability was retired.

### **Selenium-IDE: Made in Japan:**

Shinya Kasataniin Japan became interested in Selenium, and realised that he could wrap the core code into an IDE module into the Firefox browser, and be able to record tests as well as play them back in the same plugin. This tool, turned out an eye opener in more ways that was originally thought as it is not bound to the same origin policy.

## Selenium Tutorial(IDE, Core, RC)

### Contributors of Selenium an overall:



**Beans, Erik**

1. Commiter on Selenium Core and Selenium Remote control
2. Involved since 2008
3. Organization: Google
4. Favorite Selenium method: start()

**Bevan Jeniffer**



1. Commiter on Selenium Remote Control
2. Involved since 2007
3. Organization: Google
4. Favorite Selenium method: setTimeout()



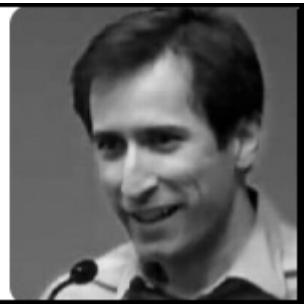
**Chai, Haw-Bin**

1. Creator of UI Element (an enhancement to selectors in the Selenium API)
2. Involved since 2007
3. Organization: CommerceHub
4. Favorite Selenium method: waitForCondition()

## Selenium Tutorial(IDE, Core, RC)

### **Fabulich, Dan**

Organization: Redfin



### **Hanrigou, Philippe (ph7)**



1. Creator of Selenium Grid, Commiter on Selenium Remote Control
2. Involved since 2006
3. Favorite Selenium method: `getHtmlSource()`

### **Huggins, Jason (hugs)**

1. Co-creator of Selenium Core, Creator of Selenium Ice (IE browser extension), Commiter on Selenium Core and Selenium Remote Control
2. Involved since 2004 (prototyped the original version of what became Selenium Core)
3. Organization: J. Randolph Amalgamated Industries, Inc.
4. Favorite Selenium method: `captureEntirePageScreenshot()`



### **Ifland, Orion0**



1. Commiter on Selenium website
2. Involved since 2008
3. Organization: Trinity Western University
4. Favorite Selenium method: `isElementPresent()`

## Selenium Tutorial(IDE, Core, RC)

### Kasatani, Shinya

1. Creator of Selenium IDE
2. Involved since 2005
3. Organization: KBMJ
4. Favorite Selenium method: `getEval()`



### Lightbody, Patrick



1. Founder of OpenQA, Co-creator of Selenium Remote Control
2. Involved since 2005
3. Self-employed
4. Favorite Selenium method: `dragAndDrop()`

### Sproul, Nelson

1. Selenium Committer
2. Involved since 2006
3. Organization: Oracle
4. Favorite method: who could choose?



### Stewart, Simon



1. Primary developer of WebDriver (which will eventually be compatible with Selenium)
2. Involved since 2008 (since 2006 with WebDriver)
3. Organization: Google
4. Favorite method: `WebDriver.findElement()`

## Selenium Tutorial(IDE, Core, RC)

### What is selenium?

Selenium is one of the most powerful open source automation tool for web application testing(even we can say acceptance testing for web application). This tool is primarily developed in Java Script and browser technologies such as DHTML and Frames and hence supports all the major browsers on all the platforms. For example, you can have your automation scripts written for Firefox on Windows and run them on Firefox in Mac. Most of the time, you will not need to change your scripts for them to work on Mac. It is more suited for the agile methodology of coding and testing. User can customize the selenium tool based on their requirement(code is available for the user). In terms of coverage for platform and browser, Selenium is probably one of the best tool available in the market for web applications. Selenium is an automate web application testing across many platforms . Especially for Compatibility and functional testing.

#### Advantage of selenium:

1. Opensource tool
2. No licensing cost associated with it
3. Customize according to our requirement

#### Disadvantage of selenium:

1. we cannot be assured of any add-ins which would be required if selenium fails to recognize objects
2. Also Support provided for Selenium would be very less ( Incase of QTP, Mercury provides standard solutions and supports to issues)

#### Variants of Selenium:

There are four variants of Selenium, which can be used in isolation or in combination to create complete automation suite for your web applications.

- Selenium IDE
- Selenium Core
- Selenium Remote Control
- Selenium Grid

#### General Functional Testing:

Functional testing tool with its powerful set of features allows automated GUI testing in minutes. Listed below are some of the features of functional testing.

## Selenium Tutorial(IDE, Core, RC)

- Scripting capabilities
- Portability
- Validation & Verification
- Playback Options
- Reporting Capabilities
- AJAX Testing

### Scripting Capabilities

Simplified Script Creation – In selenium, the default script is in html and it is the most common scripting language used among peoples.

- Keyword driven testing
- **Standard Scripting Language** allows you to easily create/update scripts without much programming knowledge.
- Object Repository is not available in selenium but it is one of the features of all Functional testing tool.
- Data driven Testing enables you to perform functional testing in different scenarios by just changing the test data in an external data source. Selenium supports Data driven testing.
- Unicode supports allows you to test multi-language deployments of your applications.
- Supports **pattern-matching capabilities** to improve test script reuse, if application changes frequently.

### Portability :

- Allows you to record scripts in **Windows** and replay it in **Linux** without re-creating the scripts.
- Browser Abstraction Layer allows scripts recorded in one browser to be replayed in all other supported browsers.
- Runtime locale option lets you to simultaneously **test all language** versions of your application with a single script.

### Validation and Verification

- Rich Library of **Built in function** added using the Function Generator tool. This helps you to construct function calls for specific testing requirements which include: In selenium, user has to create Built in function for their testing.
- **Checkpoints** that include **Text, GUI, XML, File** and **Database** enable thorough validation of your web applications/web sites. In selenium, there is a similar thing for checkpoint called verify or assert. But user can't able to check for images(bitmap). If user wants to check for database then they have to create a database connection with JavaScript and put it in user-extensions.js but not sure about creating check points in the database. I left the choice to the user.

## Selenium Tutorial(IDE, Core, RC)

### Playback Options

- Automatic/Customizable Error Recovery for unattended testing to handle unexpected window, like an ASSERT box, pop- ups, etc. But automatic error recovery for unattended testing is not available for selenium but user can write his/her own scripts to handle this.
- **Playback Synchronization** to handle the variation in time the application takes to load a new page. Ex: In selenium ClickAndWait is a command which will wait for the page to load and start executing the next command.
- Option to **chain scripts** for controlling the order of script execution. In selenium core, we can do the order of execution and if user wish he can execute everything as a batch file.
- Multiple Play back options that enables testers to debug test errors while creating and maintaining test scripts. Selenium supports multiple play back options.
- Allows execution of individual and/or groups of tests from one or many workstations.
- Provision for **command-line invocation** allows invocation from batch files or scheduling utilities.

### Reporting Capabilities

- Clear and Powerful Reports are provided to indicate the status of the test execution.
- Hyperlinks or color coding(RED color for failure) allow easy navigation through the report. This helps you to quickly identify application failures and clearly assess application quality.

**Supported Environments:** Internet Explorer 6, Firefox 1.5.x and Mozilla 1.7.x, Windows XP, 2000 and Linux.

### Web Functional Resources :

#### Compatibility Testing:

Today organizations trying to automate their web applications/web sites are looking for a uniform testing solution. This will help them *to reuse* test scripts to test it in different platforms/different browsers/different languages/different databases/different servers. The common testing issues are

Different Browsers

Different Browser Versions

Different Computer Types

Different Screen Sizes

Different Font Sizes

HTML Errors

Browser Bugs

## Selenium Tutorial(IDE, Core, RC)

### Different Browsers:

Your Web browser is a translation device. It takes a document written in the HTML language and translates it into a formatted Web page. The result of this translation is a little like giving two human translators a sentence written in French and asking them to translate it into English. Both will get the meaning across, but may not use the same words to do so.

The basic rules for translating HTML documents are established by the World Wide Web consortium, which publishes the official HTML standards. Though they defined standards, they missed out by providing certain information.

For example: HTML standards says that if you are using table, then CellSpacing(spaces allocated to each cell) should be defined in your code. But the standards didn't mention what is the default cellspacing. So developers defining their own value for Cellspacing. If the webdevelopers is not mentioning any value for cellspacing then the browser will take the default browser's value. So two browsers may use different amounts of white space in your table

In addition, the HTML standards usually run ahead of what the browsers support. No browser as yet supports 100% of the HTML Version 5 standard, but some browsers come closer than others. Over the past few years Internet Explorer has done a much better job of this than Netscape Navigator, though Opera has done arguably the best job.

latest HTML tags isn't universal, you could be building your pages with parts of the language that not all browsers understand. In that case the browser will ignore that part of your page it can't translate, and the way your page displays will be affected.

### Different Browser Versions

The major difference between two versions of the same browser is their support for newer portions of the HTML language. A new browser is generally better at displaying Web pages than an old one.

Suppose if I create a webpage using latest version of html standards and browsers(ex I.E 7) but the client (user) doesn't using the latest version of browser then they can't able to view the browser page properly.

A good rule of thumb is to design your pages to work for the last two versions of the major browsers.

### Different Computer Types

The Macintosh is still used by 12% of computer users, and has a very loyal following among graphic designers and publishers. In theory, if you view your page on both a PC and a Mac using the same version of the same browser, it should display the same, right?

In practice that's rarely the case. There are three reasons for this:

## Selenium Tutorial(IDE, Core, RC)

- **Font Availability:** When you tell your Web page to use a particular typeface, such as "Arial," you may not always get the font you want. Fonts are a computer resource, and not all computers have the same fonts as your computer. That's true even between different PCs, but it's especially true between the PC and the Mac. If the typeface of your page suddenly changes between these computers, you've probably used a font that isn't available on both computer types.
- **Font Size:** The Mac will generally render your typeface in a smaller pixel size than the PC will. That's especially true if you use the FONT tag to set your type size, since this tag uses abstract units to define size. You can avoid this problem if you use Cascading Style Sheets to set your font size in pixels.
- **Internet Explorer:** Microsoft outsources the development of Internet Explorer for the Mac, and so to a large extent this is a different browser from the PC version. In particular, the Mac version of Internet Explorer is prone to quirks and bugs that you won't see in the PC version. If you check your Web page under only one browser on the Mac, do so under Internet Explorer!

### Different Screen Sizes

If you don't test your pages using different screen resolutions, your page may be stretched to fit a large screen, or be cropped to fit a small screen.

Many experienced Web designers use HTML tables to control their page layout, yet they design their pages on large, 1024x768 pixel screens. When these pages are displayed on smaller computer screens, the browser may not be able to fit all the content onto the screen. In these cases, the content will scroll off the right of the page. While this may not sound like much of a problem, users hate scrolling left and right to view a page.

This problem should be easy to avoid, yet a surprising number of otherwise well-designed Web sites don't fit within the standard 800-pixel PC computer screen. This is especially a problem for pages built by graphic artists using a Macintosh, whose standard screen size is 1024x768. Many designers forget that designing for an 800-pixel screen means using roughly a 750-pixel layout.

Many less experienced Webmasters design their pages without tables, or dimension their tables to fit 100% of the screen width. If you've built your Web page this way on a computer with an 800-pixel screen, you may be surprised how your page is stretched to fit a larger screen. Text and graphics that were adjacent on your smaller screen may not be so on a large screen.

To avoid this, wrap the content of your page inside a single, large, one-cell table. Set the width of this table to be about 560-pixels, if you're designing for a 600 pixels layout, or 750-pixels if you're designing for an 800 pixel layout. Because the majority of Internet users have a screen resolution of 800 pixels or less, it's a bad idea to design your site for a larger screen.

### Different Font Sizes

## Selenium Tutorial(IDE, Core, RC)

Most browsers allow users to customize their default font size. Many users who work on computers all day do this to reduce eye strain. As a result, user preferences may cause the typeface that you used to design your Web page to increase as much as 50% larger in a user's browser. This increase in font size can hurt many carefully-planned page designs.

Everyone is eager to start testing the application in selenium. Now let us get to know about **Selenium IDE**.

### Selenium IDE:

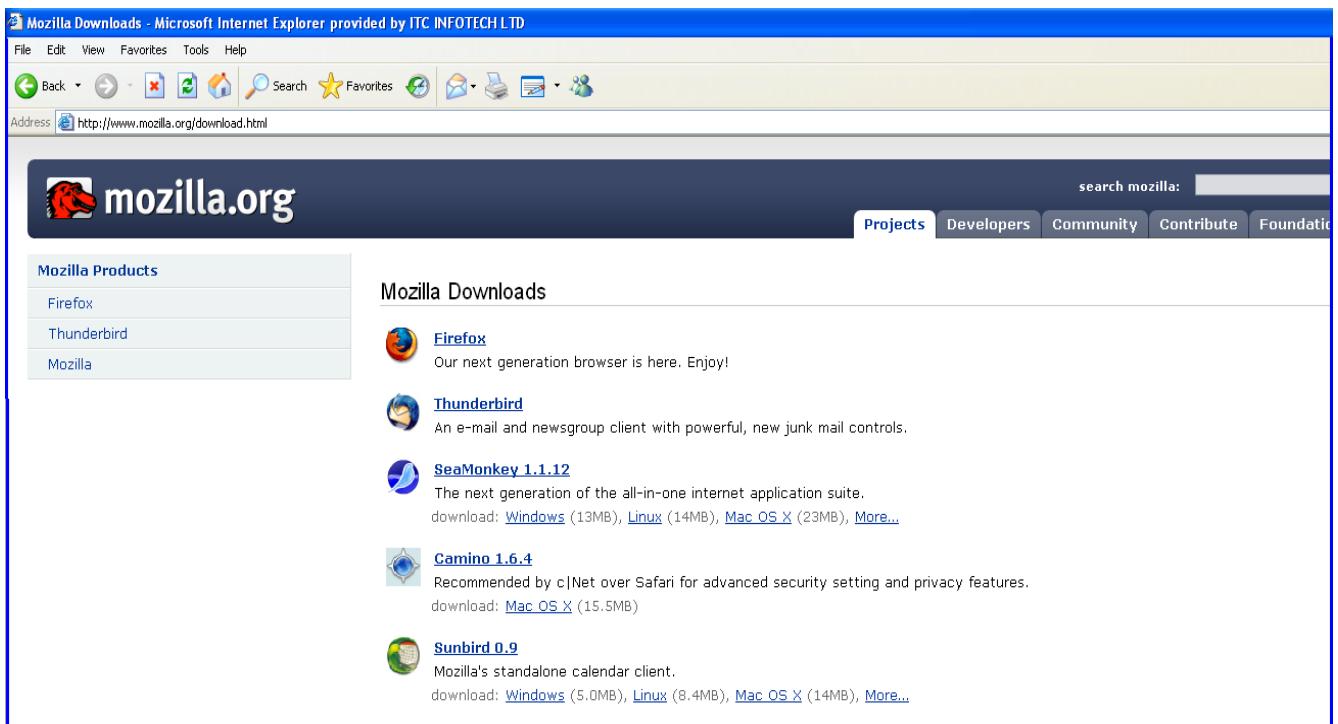
#### Installation instruction of Selenium IDE:

#### Prerequisites:

Mozilla Firefox – browser minimum version 1.5

**Step1:** Download the mozilla firefox from the following site:

<http://www.mozilla.org/download.html>



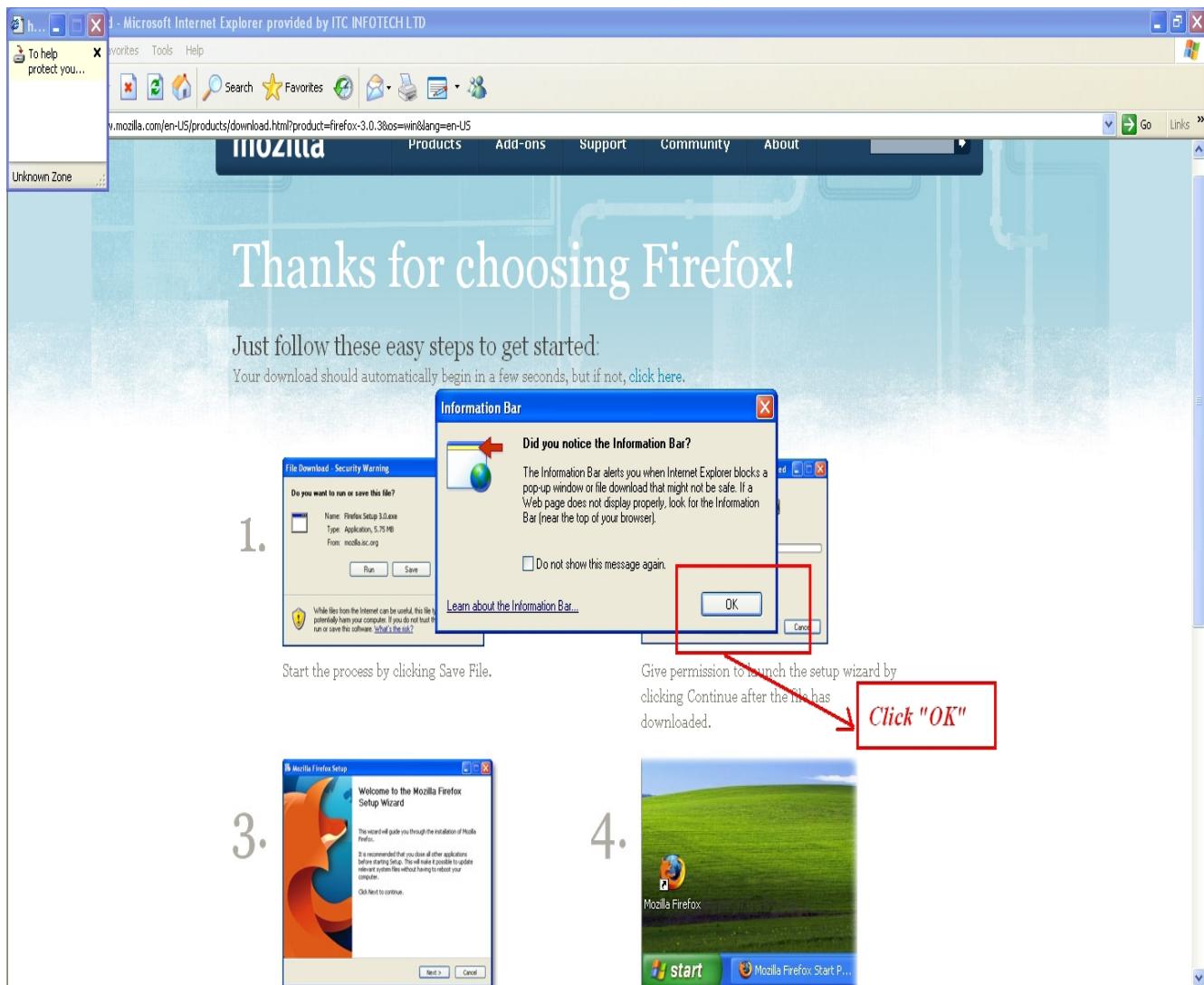
## Selenium Tutorial(IDE, Core, RC)

**Step 2:** Go to the latest version of Firefox page, click the download button. Here we are using version3 of mozilla.



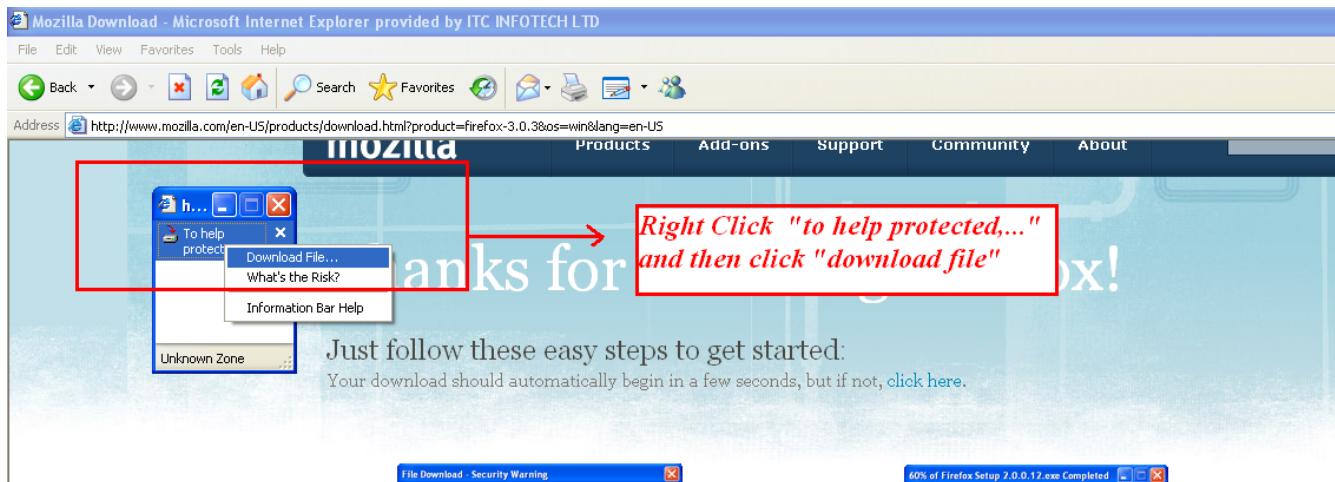
## Selenium Tutorial(IDE, Core, RC)

**Step 3:** Sometimes, Default browser “IE” is having certain restriction in downloading the software. Allow the software to download.

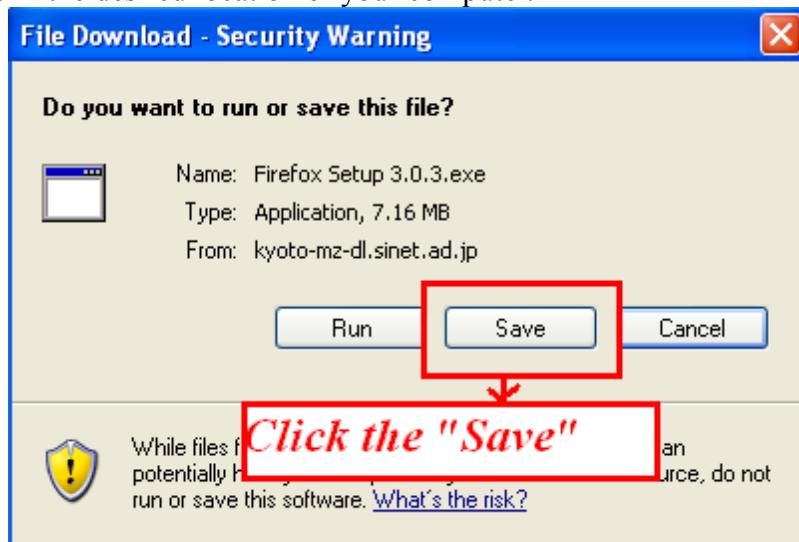


## Selenium Tutorial(IDE, Core, RC)

### Step 4: continue....



### Step 5: Save the file in the desired location of your computer.

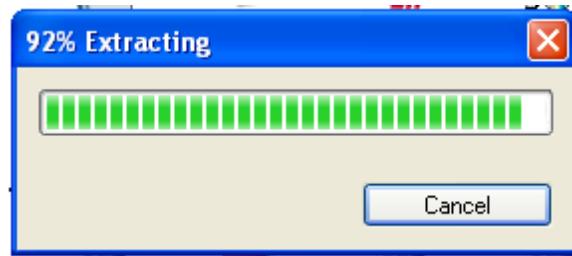


## Selenium Tutorial(IDE, Core, RC)

**Step 6:** Double click the executable(Firefox setup3.0.1.exe) file.



**Step 7:** After clicking the executable file, user will be getting the extraction dialog box. Just watch.



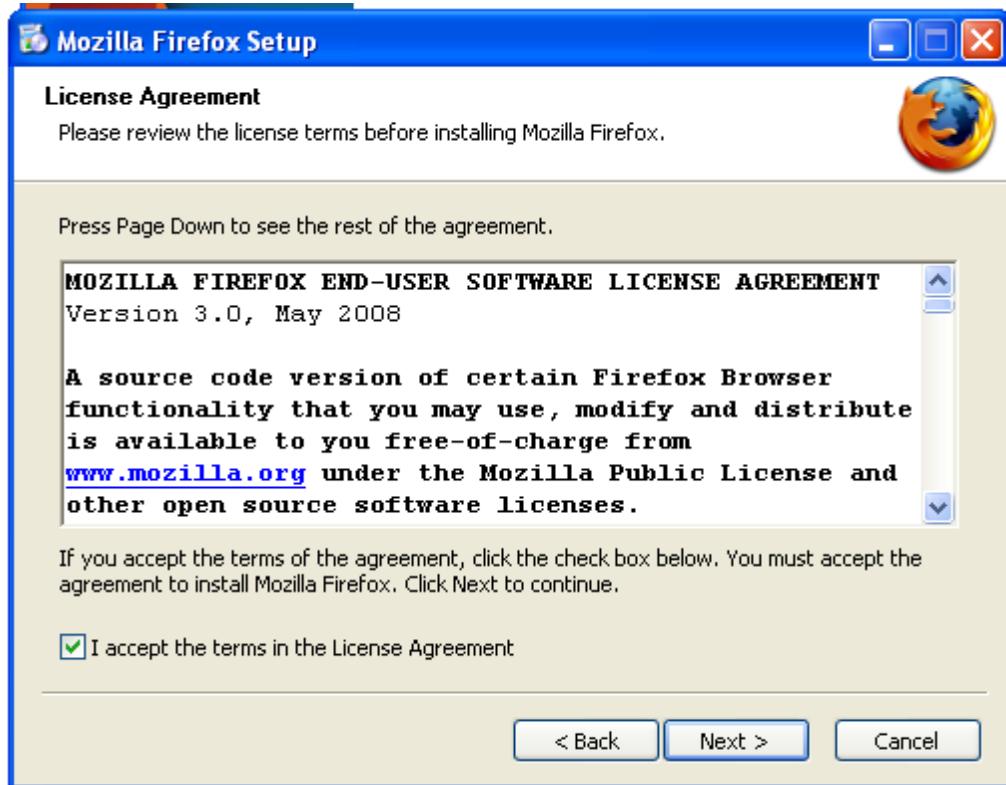
**Step 8:** Click “Next” button in the welcome window

## Selenium Tutorial(IDE, Core, RC)

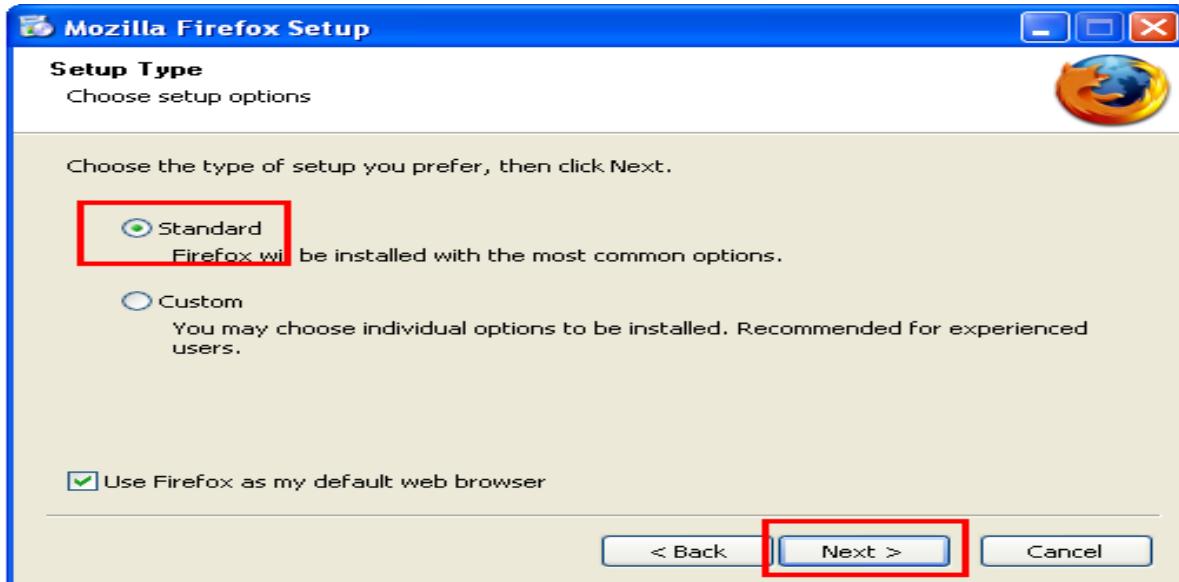


**Step 9:** Accept the license agreement

## Selenium Tutorial(IDE, Core, RC)



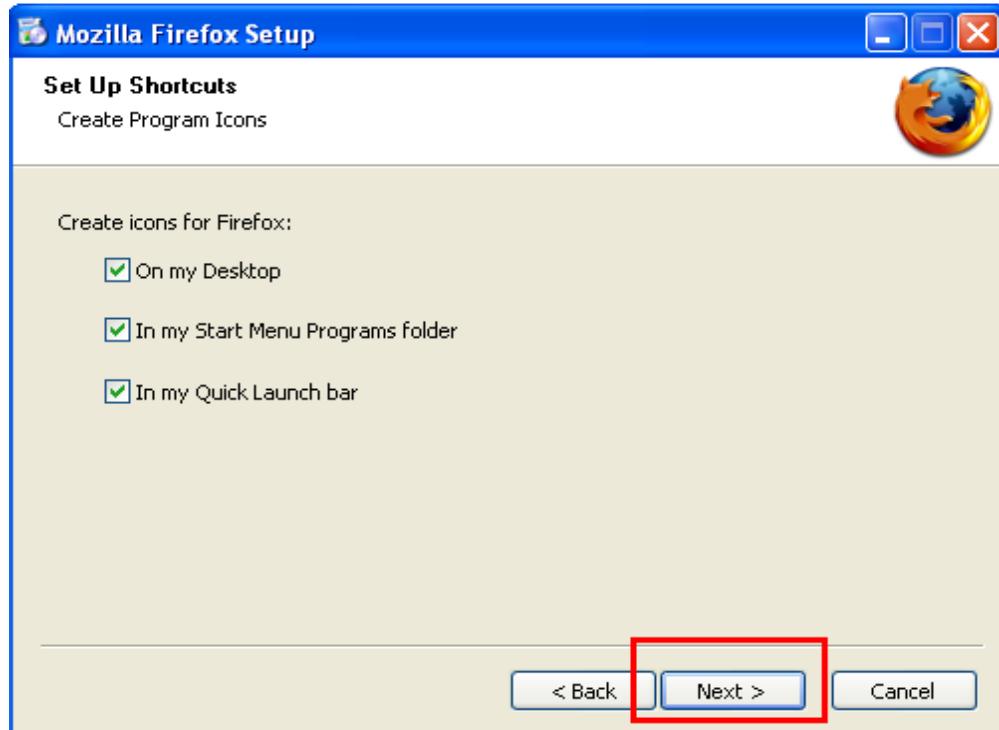
**Step 10:** Choose “Standard”, if the user prefers to install the software in “C:\Program Files\”, then click **next** button and proceed with the installation.If you want to install in different location of your system, then click custom and choose the directory where you wanted to install the software.



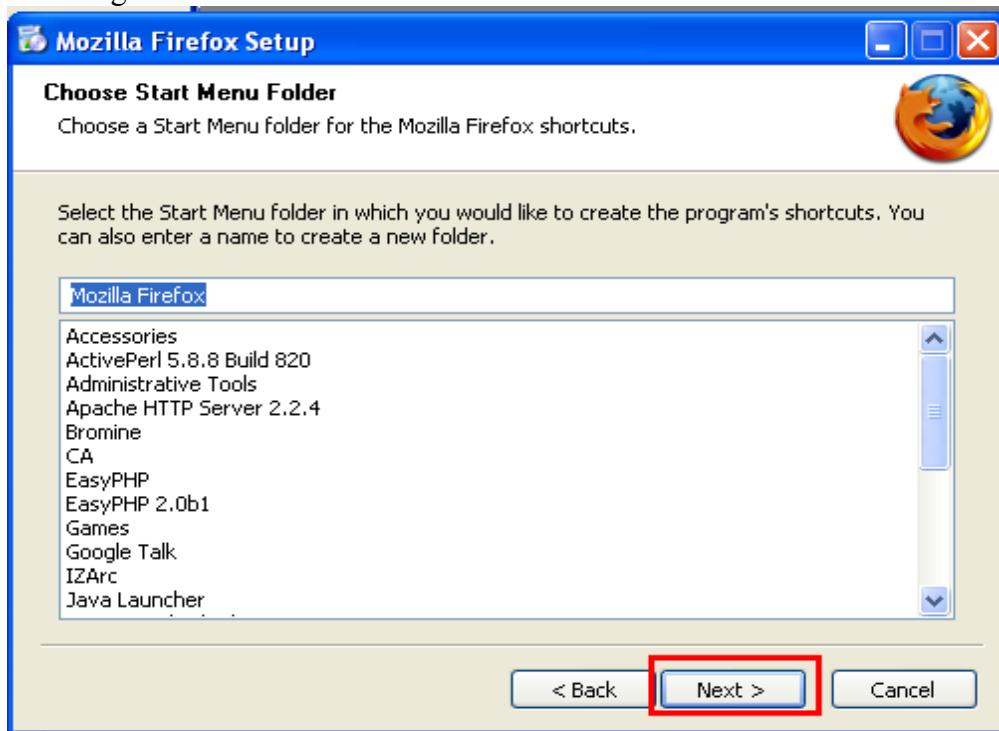
**Step 11:** If the user prefers to have shortcut icon in the following location, then enable “All”,

## Selenium Tutorial(IDE, Core, RC)

otherwise choose whichever is applicable to them.

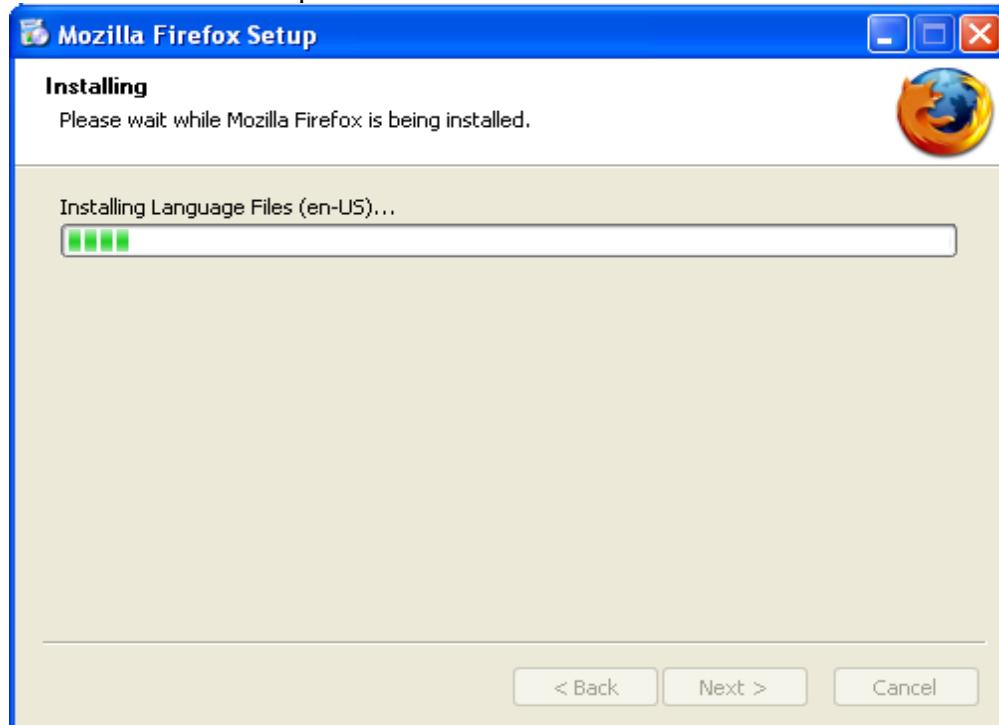


**Step 12:** This will create a start menu folder for the Mozilla , you can choose your own name for the folder. It's advisable give the default name.



## Selenium Tutorial(IDE, Core, RC)

**Step 13:** Screenshot of installation process

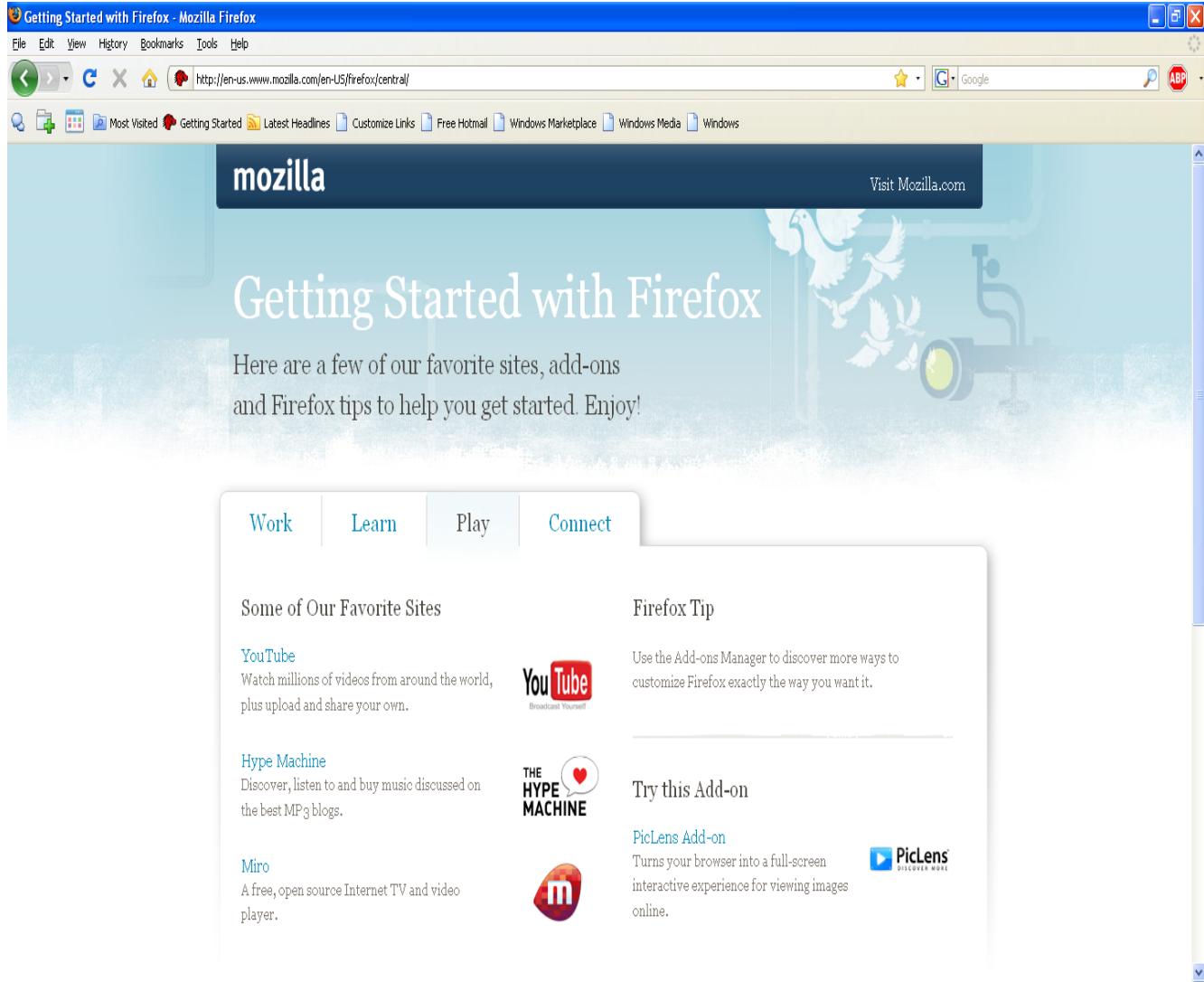


**Step 14:** Installation has been completed successfully. Click the “**finish**” button.



## Selenium Tutorial(IDE, Core, RC)

Now the Mozilla browser will open the default page of it.



Step 15: Once you clicked finish, automatically Mozilla browser will launch with the default page.

Congratulation, you successfully installed Mozilla firefox browser. Now let's proceed further for selenium IDE.

## **Selenium Tutorial(IDE, Core, RC)**

### **Download the Selenium IDE:**

After installing the Mozilla, the next step is to download the selenium IDE. Selenium IDE is available as **.xpi** file(executable), user can download it from the following site.

**Step1 :** Click the following link, user will be move to the Open QA selenium page, Choose the latest version of the selenium IDE for download.

<http://selenium-ide.openqa.org/download.jsp>

### **Other relevant sites for Selenium IDE:**

<http://selenium.openqa.org/>

<http://www.brothersoft.com/selenium-ide-163845.html>

<http://www.brothersoft.com/selenium-ide-download-163845.html>

## Selenium Tutorial(IDE, Core, RC)

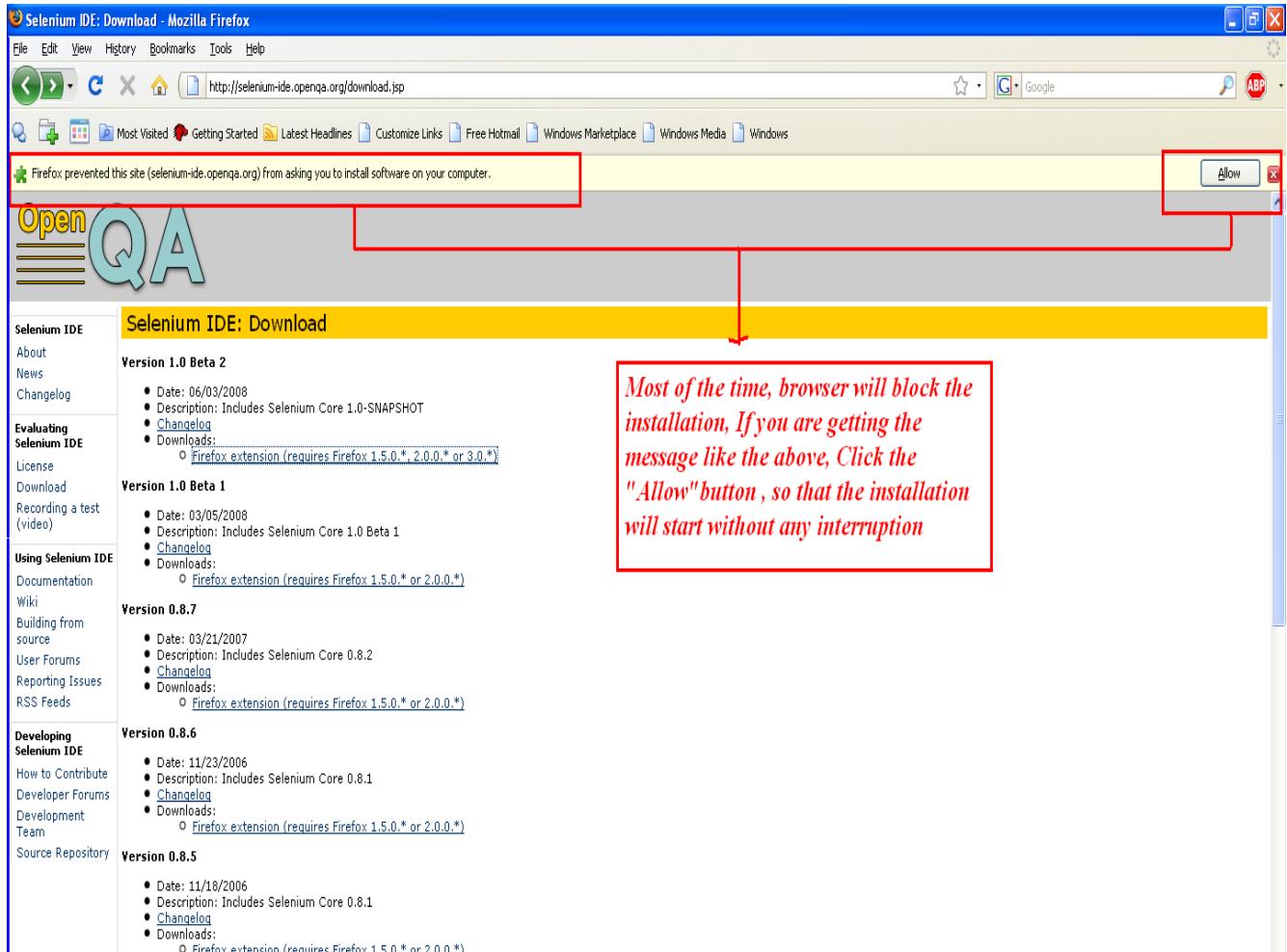
The screenshot shows the Selenium IDE download page on Mozilla Firefox. The URL in the address bar is <http://selenium-ide.openqa.org/download.jsp>. The page features a yellow header bar with the text "Selenium IDE: Download". Below the header, there's a sidebar with links for "Selenium IDE", "About", "News", "Changelog", "Evaluating Selenium IDE", "License", "Download", "Recording a test (video)", "Using Selenium IDE", "Documentation", "Wiki", "Building from source", "User Forums", "Reporting Issues", and "RSS Feeds". The main content area lists several versions of the Selenium IDE:

- Version 1.0 Beta 2**
  - Date: 06/03/2008
  - Description: Includes Selenium Core 1.0-SNAPSHOT
  - [Changelog](#)
  - Downloads:
    - [Firefox extension \(requires Firefox 1.5.0.\\* or 2.0.0.\\* or 3.0.\\*\)](#)
- Version 1.0 Beta 1**
  - Date: 03/05/2008
  - Description: Includes Selenium Core 1.0 Beta 1
  - [Changelog](#)
  - Downloads:
    - [Firefox extension \(requires Firefox 1.5.0.\\* or 2.0.0.\\*\)](#)
- Version 0.8.7**
  - Date: 03/21/2008
  - Description: Includes Selenium Core 0.8.2
  - [Changelog](#)
  - Downloads:
    - [Firefox extension \(requires Firefox 1.5.0.\\* or 2.0.0.\\*\)](#)
- Version 0.8.6**
  - Date: 11/23/2006
  - Description: Includes Selenium Core 0.8.1
  - [Changelog](#)
  - Downloads:
    - [Firefox extension \(requires Firefox 1.5.0.\\* or 2.0.0.\\*\)](#)
- Version 0.8.5**
  - Date: 11/18/2006
  - Description: Includes Selenium Core 0.8.1
  - [Changelog](#)
  - Downloads:
    - [Firefox extension \(requires Firefox 1.5.0.\\* or 2.0.0.\\*\)](#)
- Version 0.8.4**
  - Date: 10/27/2006
  - Description: Includes Selenium Core 0.8.0
  - [Changelog](#)
  - Downloads:
    - [Firefox extension \(requires Firefox 1.5.0.\\* or 2.0.0.\\*\)](#)

A red box highlights the "Firefox extension" link under Version 1.0 Beta 2, and a red arrow points from this box to a red box containing the text "Click here to download the Selenium IDE".

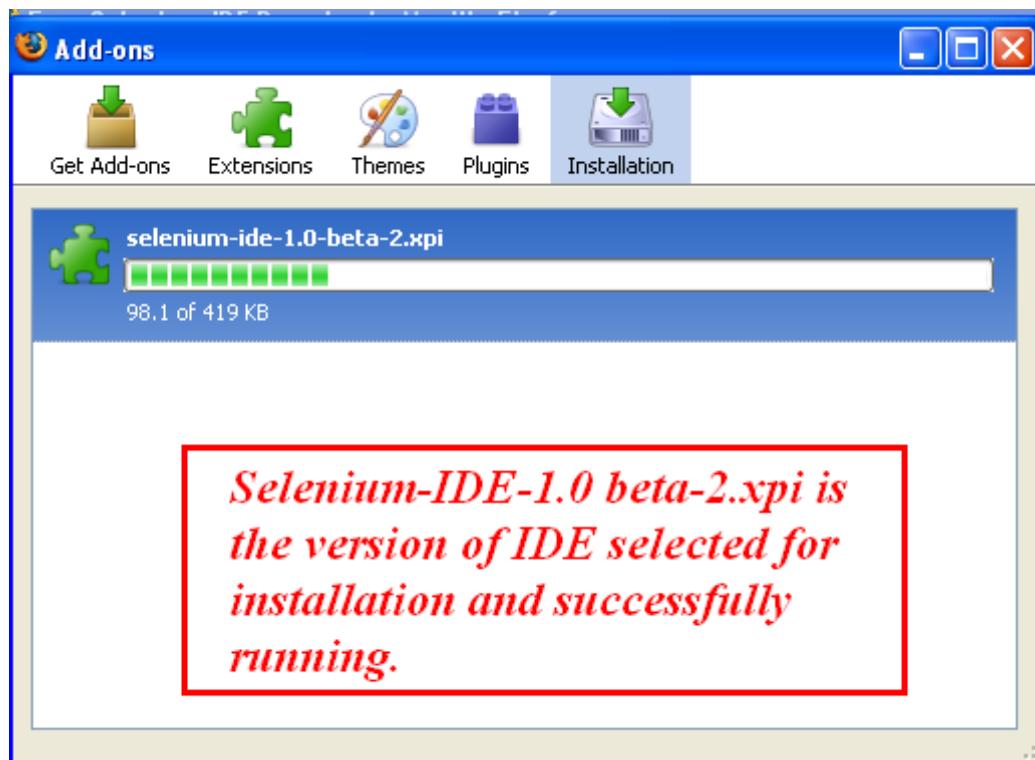
## Selenium Tutorial(IDE, Core, RC)

**Step 2:** Sometimes, browser may block the sites, Unblock and allow the sites for download.



## Selenium Tutorial(IDE, Core, RC)

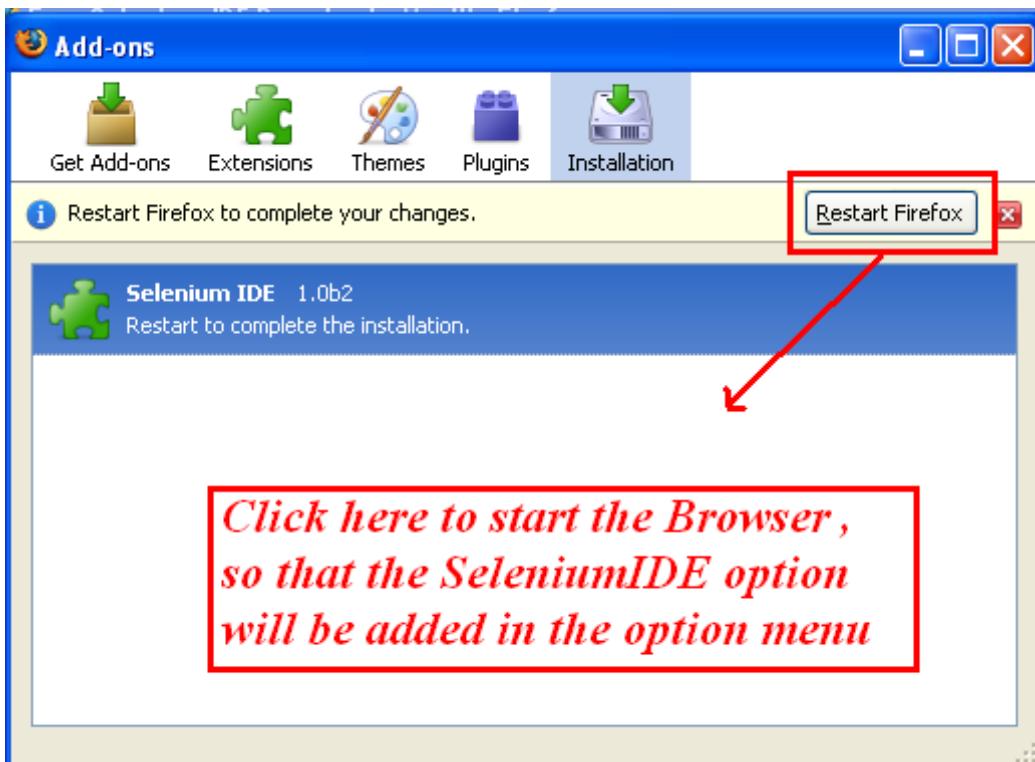
**Step 3:** Mozilla Firefox start adding the extensions in the Mozilla browser



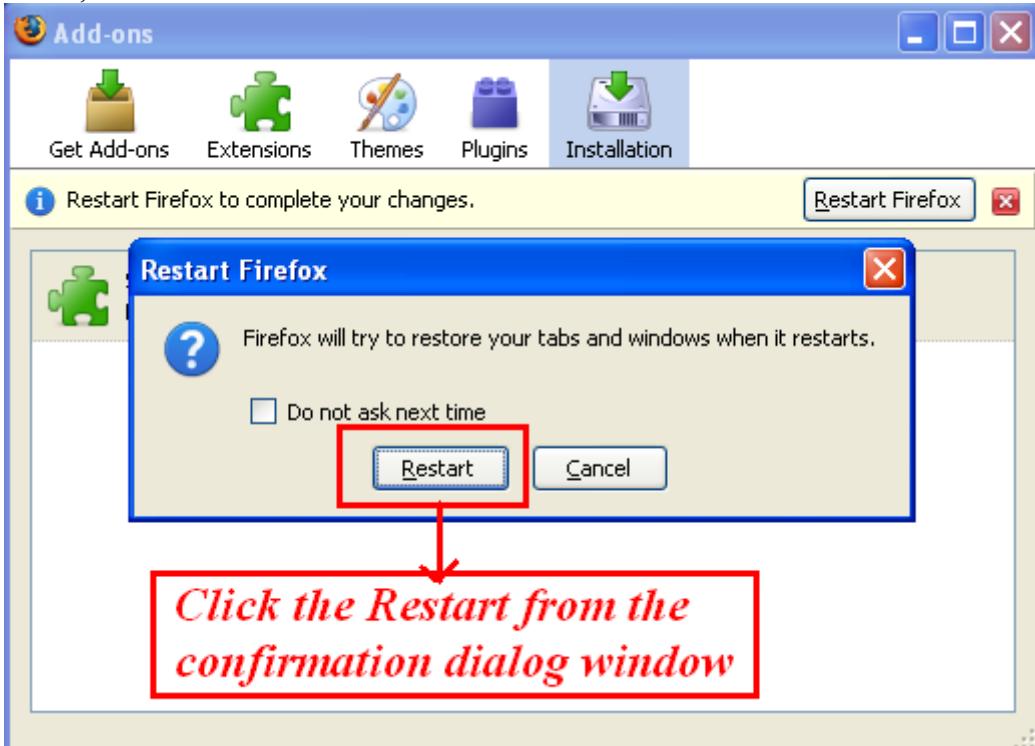
## **Selenium Tutorial(IDE, Core, RC)**

**Step 4:** Inorder to activate the Selenium IDE , restart the browser.

## Selenium Tutorial(IDE, Core, RC)

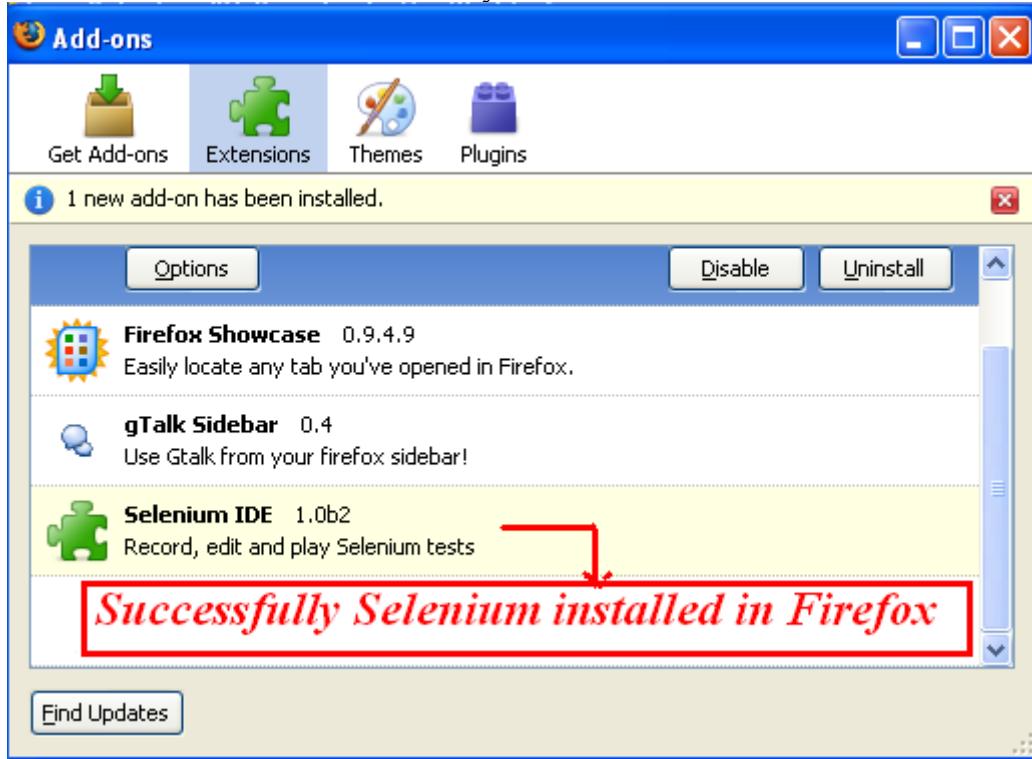


Step 5: Continue,...



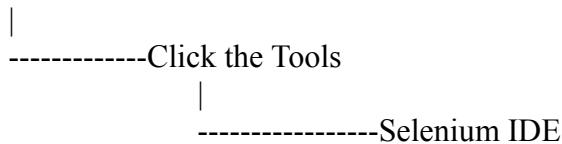
## Selenium Tutorial(IDE, Core, RC)

**Step 7:** Now Selenium IDE has been successfully added into the Mozilla Firefox browser.



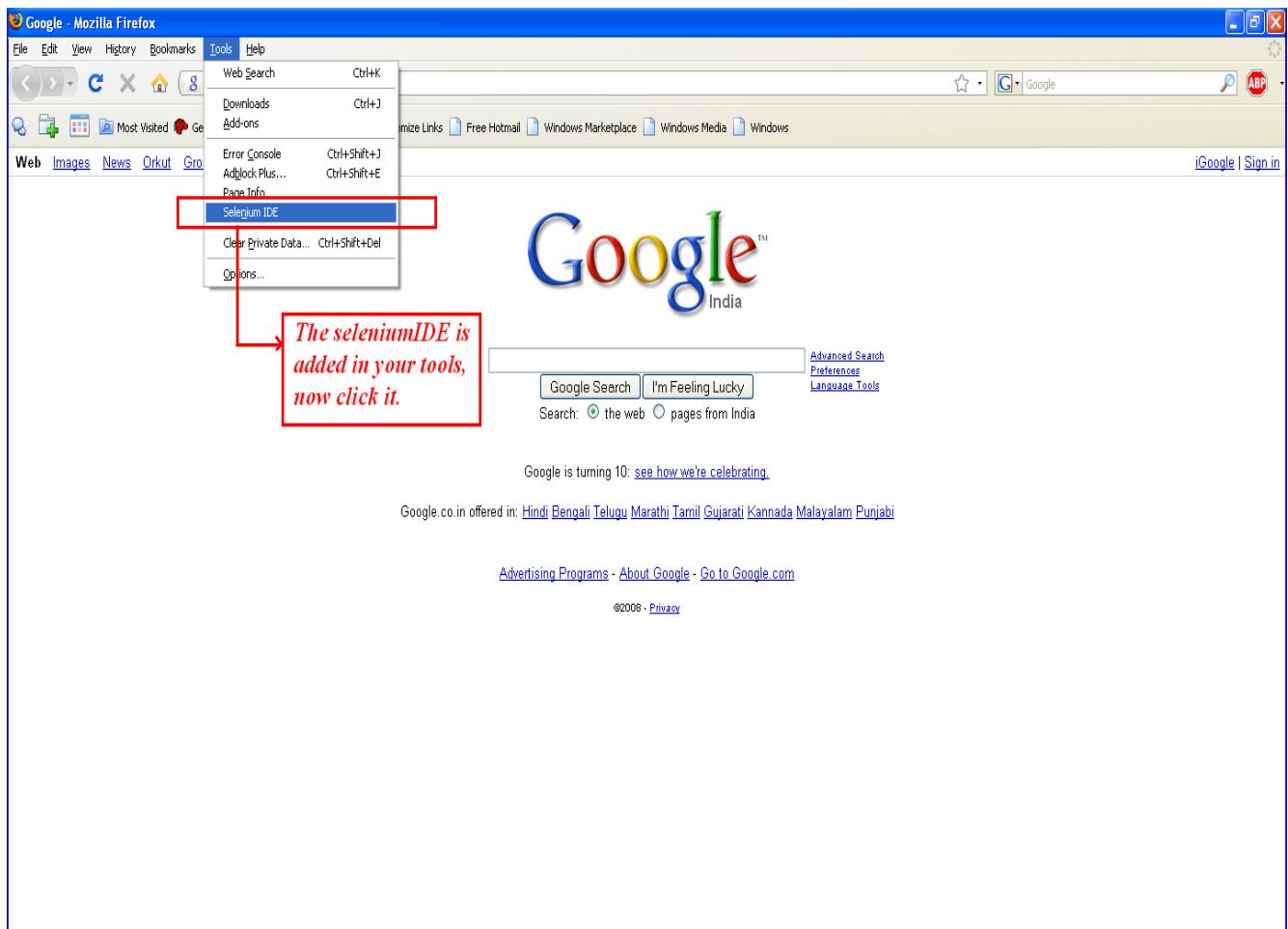
**Step 8:**

Open the Mozilla firefox



By default, selenium starts in recording mode.

## Selenium Tutorial(IDE, Core, RC)



**Congratulation! You successfully installed Selenium IDE.**

## **Selenium Tutorial(IDE, Core, RC)**

### **Tutorial on Selenium IDE:**

#### **Introduction about Selenium IDE:**

Selenium IDE is a Firefox plug in, which includes the entire Selenium Core, allows you to record, play back, edit, and debug tests in browser. It provides the simplest introduction to Selenium and is highly recommended for beginners.

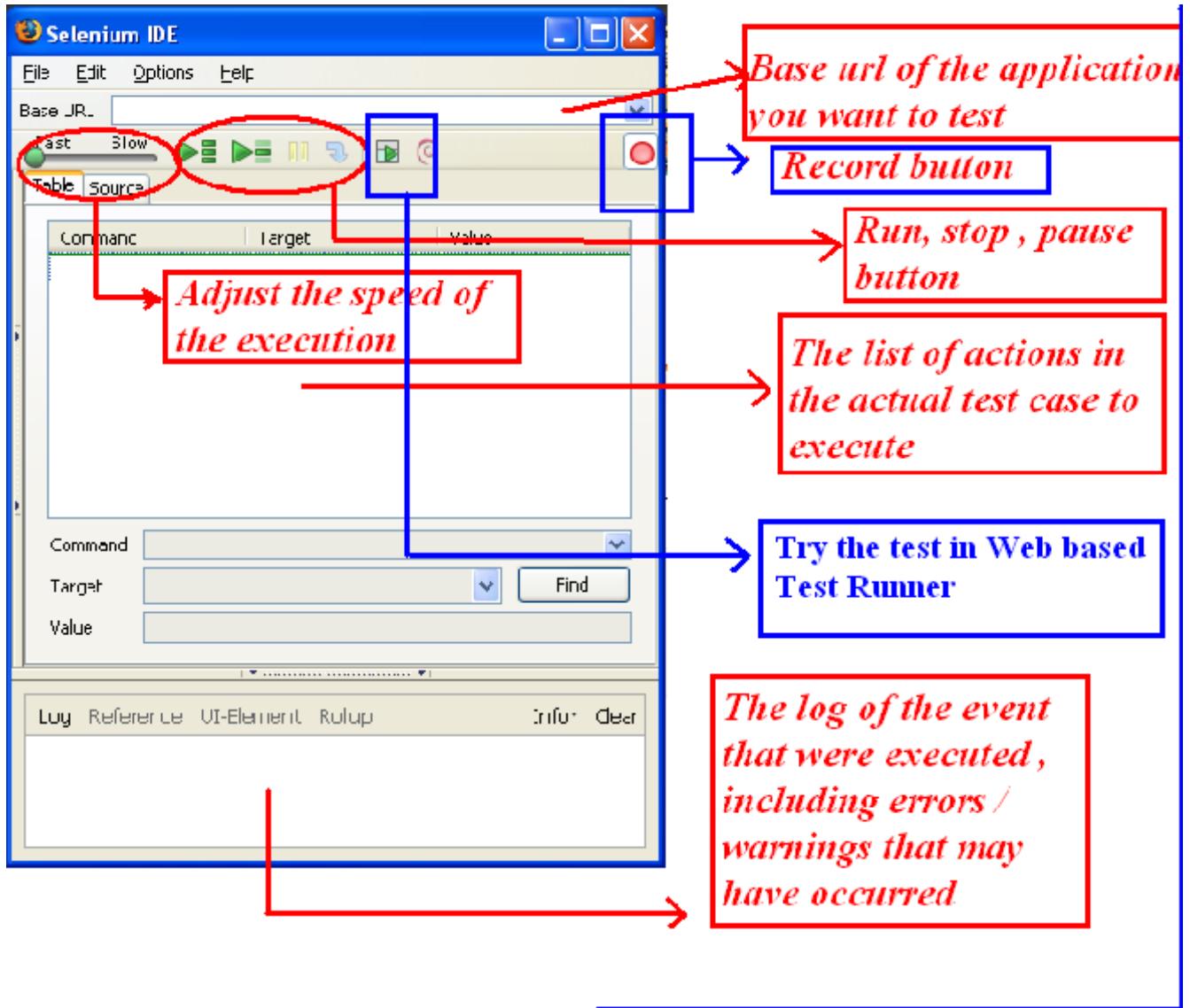
Selenium IDE is the only flavor of Selenium which allows you to record user action on browser window. It can also record user actions in most of the popular languages like Java, C#, Perl, Ruby etc. This eliminates the need of learning new vendor scripting language. However to run them in an automated fashion you need to use Selenium Remote Control.

Selenium is a powerful tool for controlling, automating, or testing websites. If there is any repetitive tasks or predictable task that you are always doing in browsers that you can handle it in Selenium IDE.

So if you are excited about the tool, Lets start playing with Selenium IDE now. If installed properly, Selenium can be accessed from tool --> Selenium IDE in your browser tool bar. By default, Selenium IDE will open in Record mode.

## Selenium Tutorial(IDE, Core, RC)

Figure 1: Selenium IDE window



The four execution buttons on the top are:

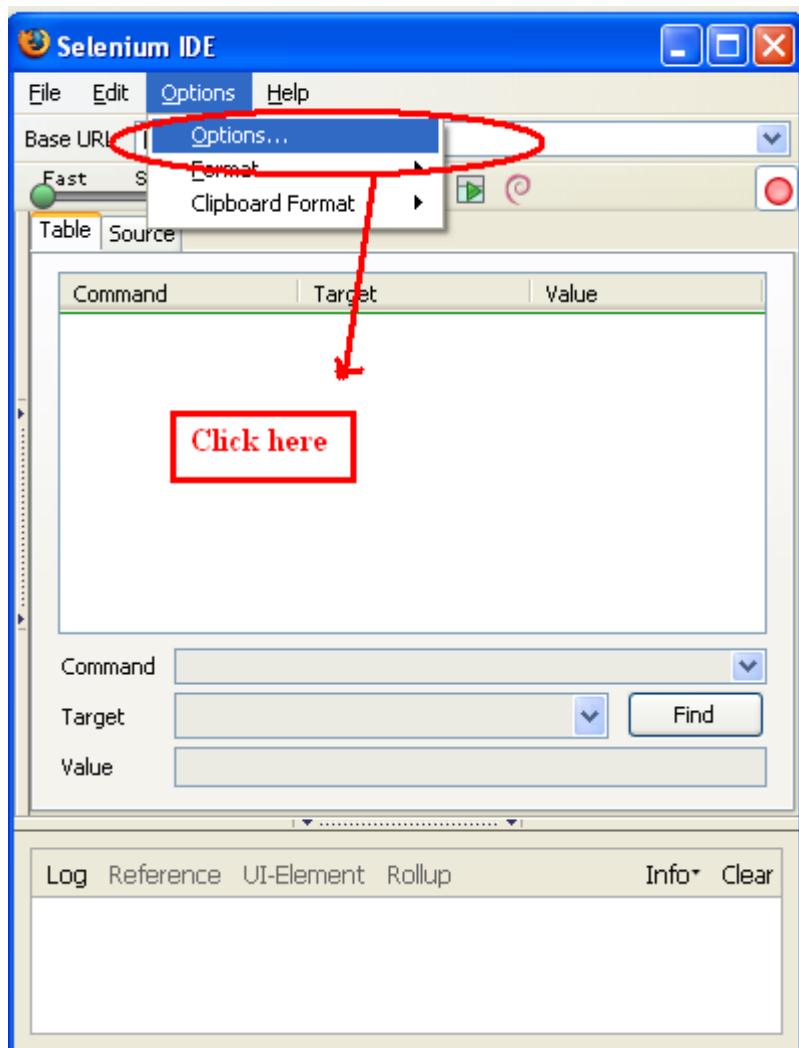
- Run All Tests: Run every test in the test suite.
- Run Selected Test: Click on one test in the test suite, then click here to run just that one test.
- Pause / Continue: While a test is running, press the Pause button to temporarily stop running commands. After you press Pause, you can press Continue to resume the test.

## Selenium Tutorial(IDE, Core, RC)

- Step: Click on a test in the test suite, then click on just one line of the test to set a "breakpoint." When the TestRunner reaches a breakpoint, it will automatically pause execution on that line. You may then press the Step button to run the next line of the test and pause again. You can also click on a breakpoint line to remove the breakpoint.

Figure 2:

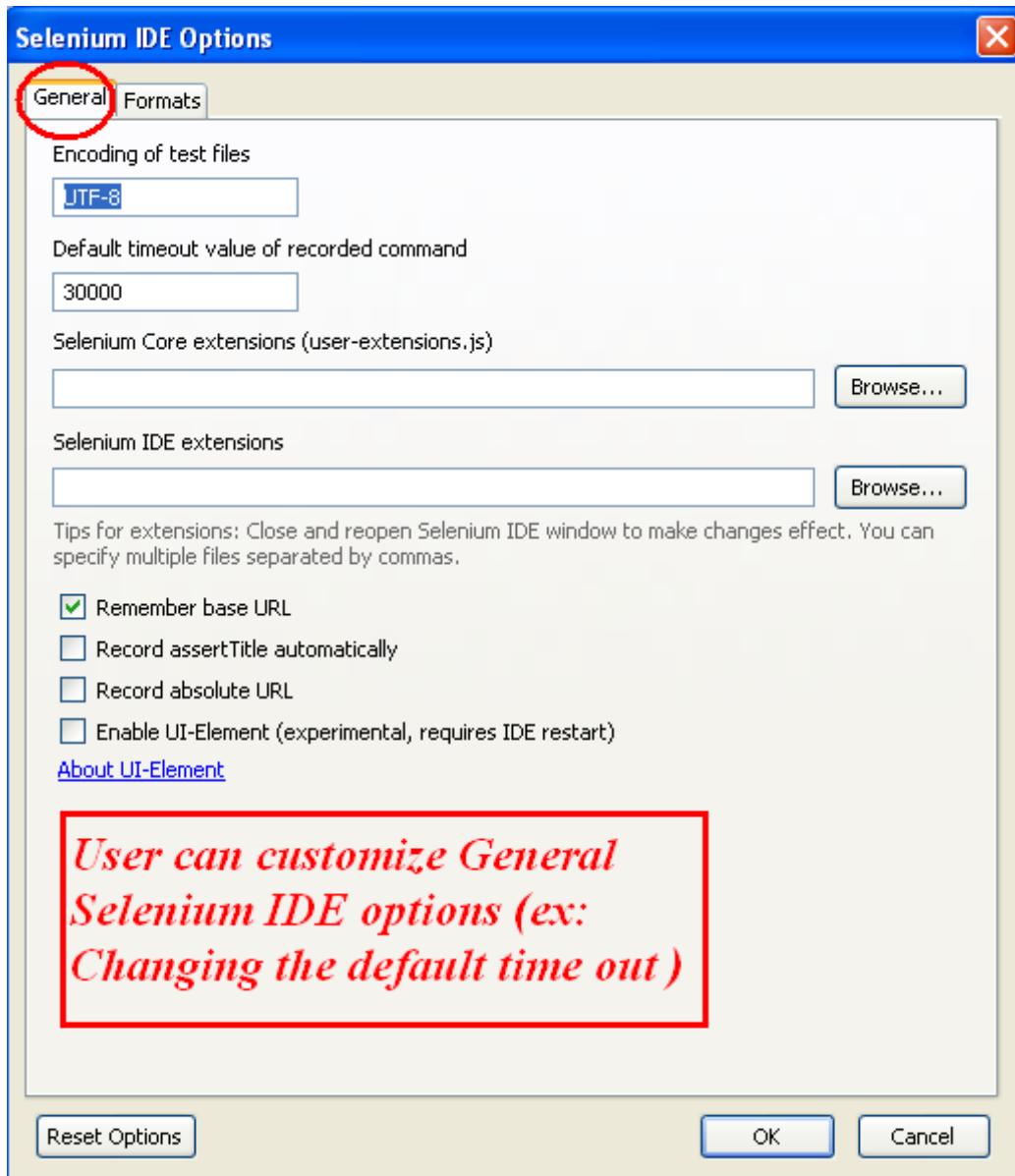
In selenium IDE , user can do certain customization that will be available as a default settings.For that, click the Options tab in the menu bar, there you can able to see “Options” , click this “Options.



## Selenium Tutorial(IDE, Core, RC)

Selenium IDE options window will open. User can able to see two tab.

1. General tab
2. Format tab



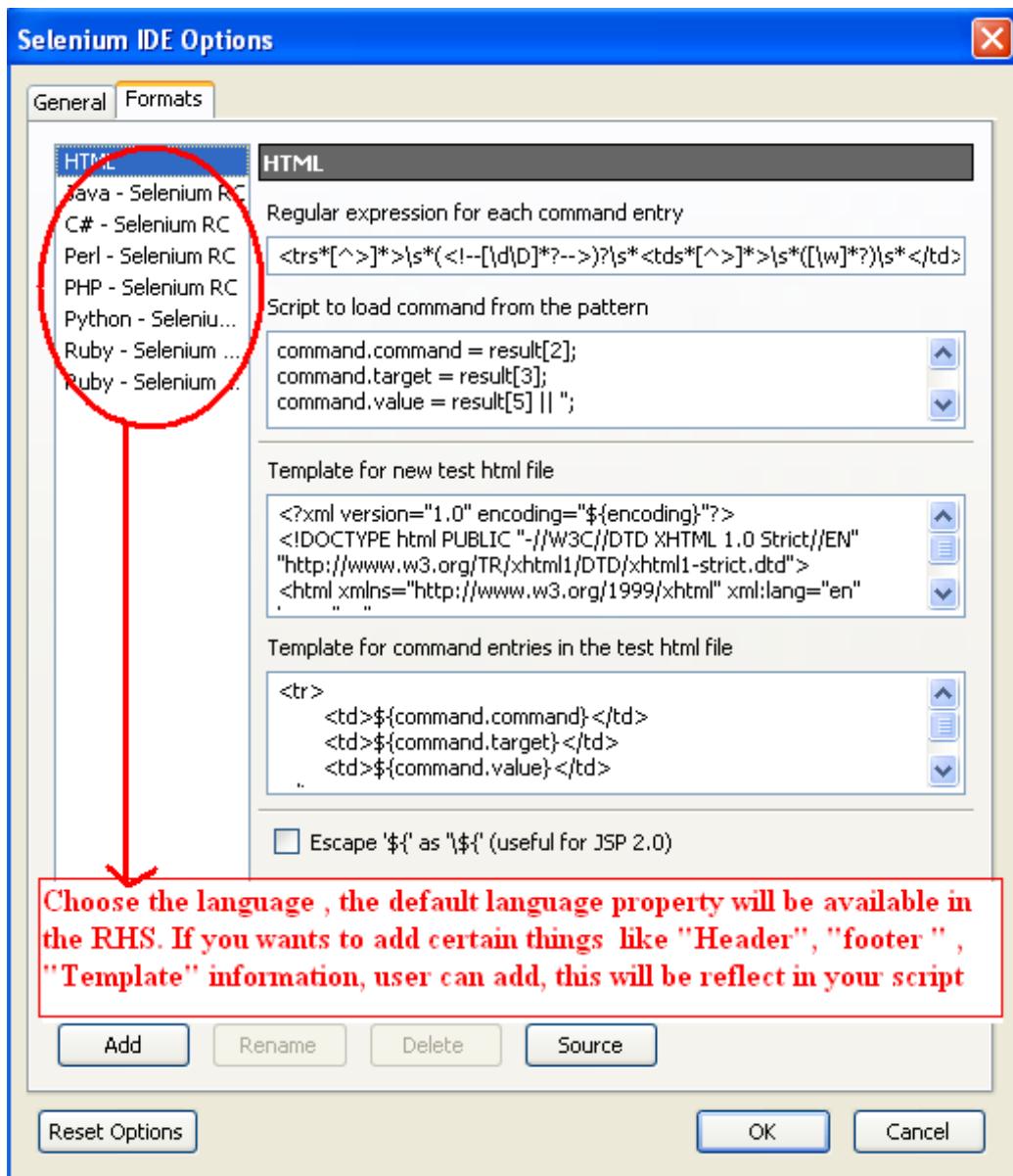
In general tab, you can change the setting like encoding format UTF-8 format into another format. By default Selenium IDE is available in UTF-8 (UTF-8 stands for Unicode Transformation Format-8. It is an octet (8-bit) lossless encoding of Unicode characters).

Default timeout value of the action is 30000 milliseconds. user can change according to their requirement

## Selenium Tutorial(IDE, Core, RC)

Similarly they can do other modification according to their wish. Most of the information are self explanatory.

It is not mandatory , user should know about all these data's. If we know, it's additional information to the user , merely we can say it as GK(general knowledge)



## Selenium Tutorial(IDE, Core, RC)

Option format list out the available language of the selenium. User can customize the format according to their requirement.

**Steps:** choose the language,in the corresponding language format will be displayed in the RHS. Edit the default format and provide your own format to be appear in the language. The few are “Template”, “Command” entries. “regular expression” , “Header”, “Footer”, etc...,

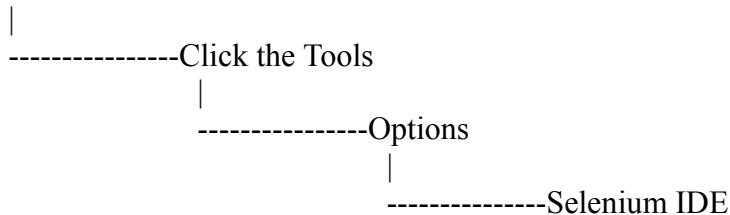
Any way selenium default scripts is html(added javascript) and that is the script we are going to use it in selenium IDE. The reason for availability of other language is , user can get the scripts for Selenium Remote Control. In selenium RC, user has to write the scripts on their own, instead of writing they can get the scripts(with little modification) from Selenium IDE and execute it in Selenium RC.

Now let's create a sample script in Selenium IDE , so that the user will get to know about the usefulness of Selenium IDE well.

Step1:

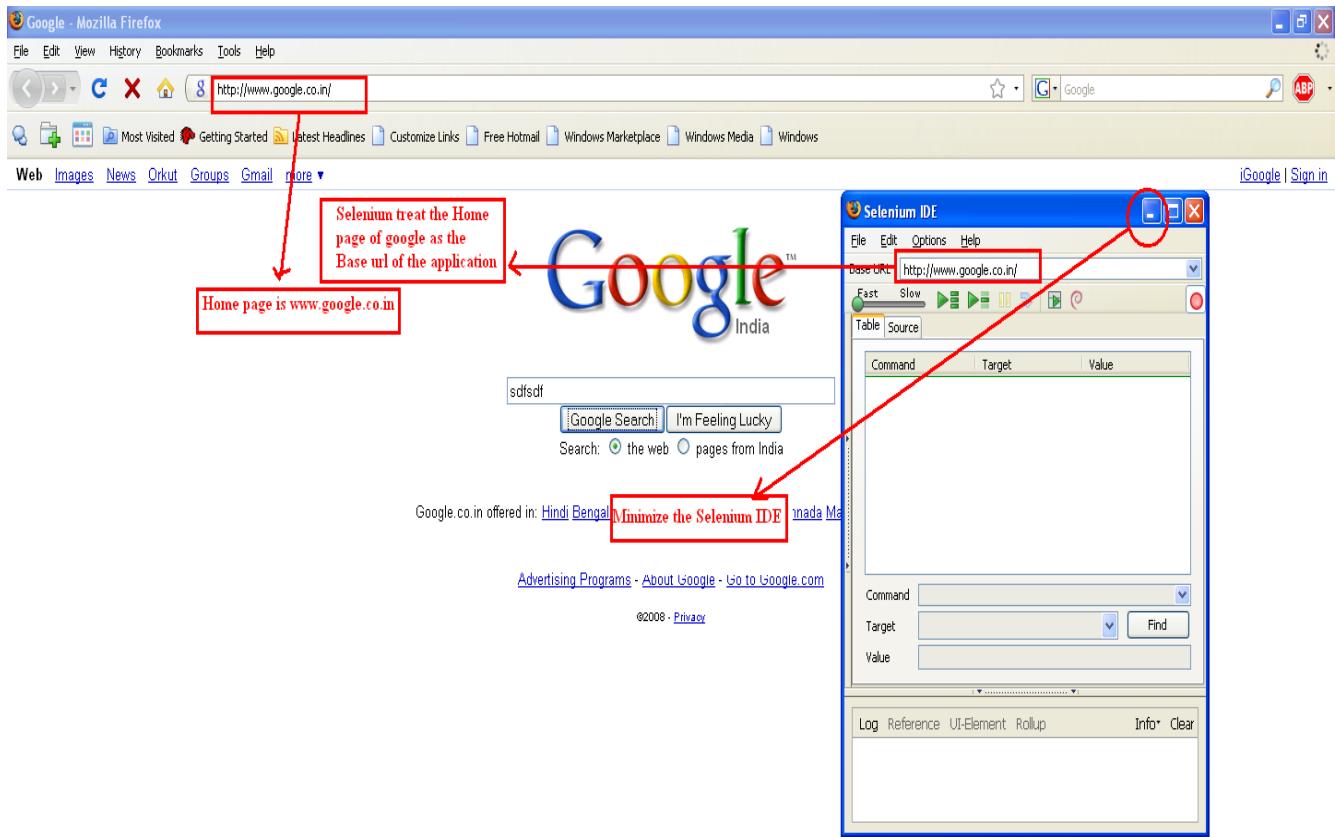
I am assuming that your home page is “[www.google.co.in](http://www.google.co.in)”

Open the mozilla firefox



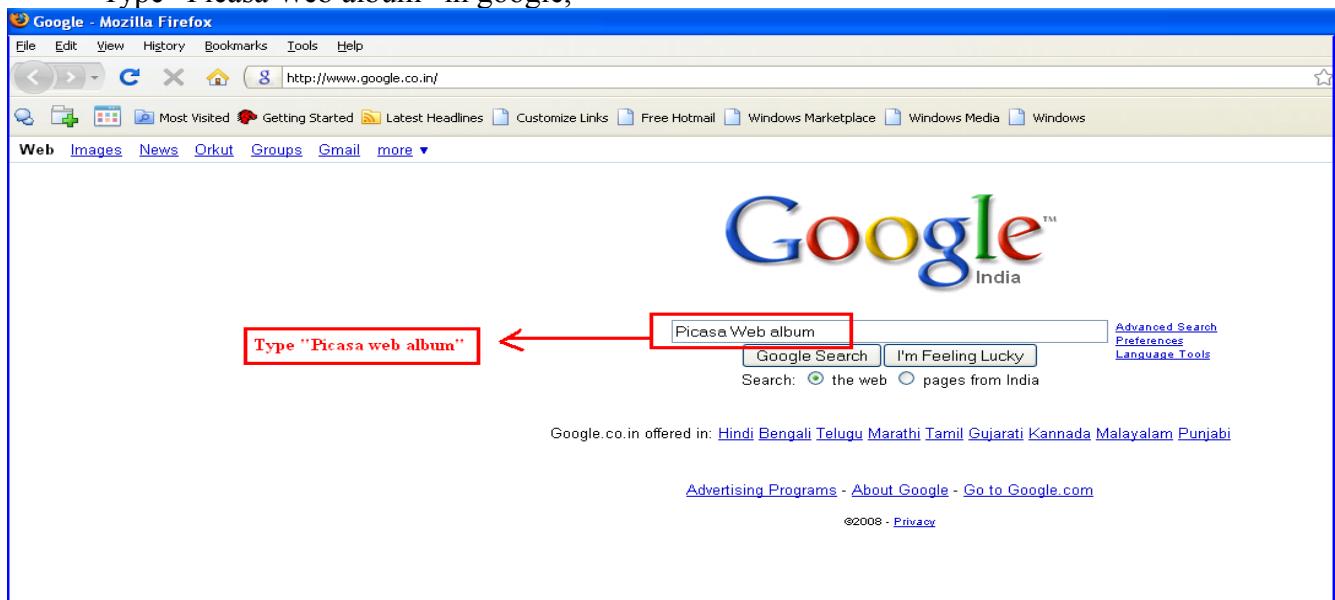
By default, Selenium IDE will open in record mode. No need to click the “Record” button. **Minimize** the Selenium IDE window.

## Selenium Tutorial(IDE, Core, RC)



### Step 2:

Type “Picasa Web album” in google,



## Selenium Tutorial(IDE, Core, RC)

**Step 3:** Now Google will list out the search result. Choose the Picasa webalbum which usually appears in the first of search. Click and move to the login page.

picasa web album - Google Search - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.google.co.in/search?hl=en&q=picasa+web+album&btnG=Google+Search&meta=

Most Visited Getting Started Latest Headlines Customize Links Free Hotmail Windows Marketplace Windows Me

Web Images News Orkut Groups Gmail more ▾

**Google** picasa web album Search Advanced Search Preferences

Search: the web pages from India

**Web**

**Picasa Web Albums:** free photo sharing from Google → **Click here to login to picasa webalbum.**

Fast and easy photo sharing from Google. Share with friends and family, or explore photos.

[picasaweb.google.com/](https://picasaweb.google.com/) - 15k - Cached - Similar pages

Download Picasa 3 (beta) Learn more about Picasa Web Albums  
Accounts Gmail  
Back I cannot access my account  
map your photos Google Image Labeler  
[More results from google.com >](#)

**Picasa Web Albums**  
One-click **web** upload directly from **Picasa**, or add photos using a **web** browser or your Mac.  
Share photos with friends and family Share **albums** exclusively with ...  
<https://www.google.com/accounts/ServiceLogin?service=lh2> - 12k - Cached - Similar pages

**Picasa 3:** Free download from Google  
Create Turn photos into movies, collages, slideshows and more. share. Share Upload  
seamlessly to **Picasa Web Albums** to share with friends, family & the world ...  
[picasa.google.com/](https://picasa.google.com/) - 9k - Cached - Similar pages

## Selenium Tutorial(IDE, Core, RC)

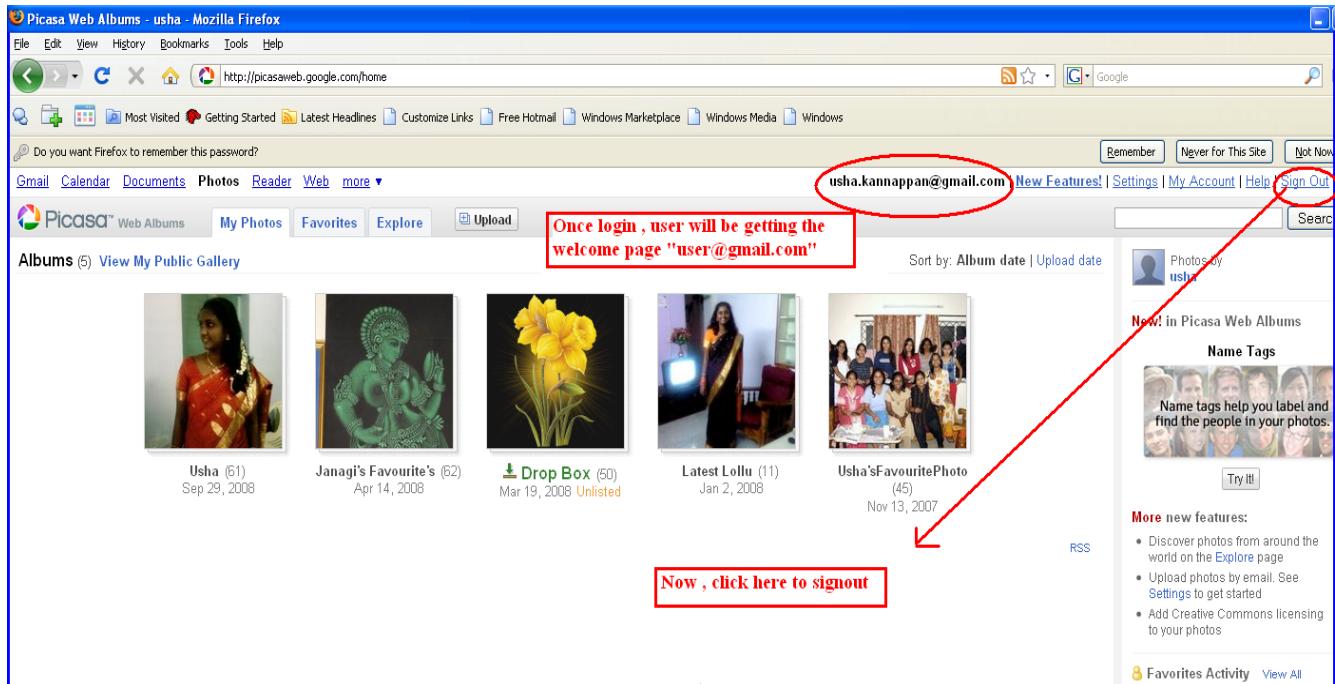
**Step 4:** This is the login page of the picasa web album. If you are having gmail account, then use the same username and password to login. If you doesn't have then try to create for this selenium test.

The screenshot shows the Picasa Web Albums login page within a Mozilla Firefox browser window. The URL in the address bar is <https://www.google.com/accounts/ServiceLogin?service=lh2>. The page title is "Picasa™ Web Albums" and the subtitle is "Fast and easy photo sharing from Google". On the left, there's a "Just added!" message and a "Search Photos" button. Below it is a thumbnail image of a photo gallery. To the right is the "Sign in to Picasa Web Albums with your Google Account" form, which includes fields for Email (containing "usha.kannappan") and Password (containing a redacted string), a "Remember me on this computer" checkbox, and a "Sign in" button. A red box highlights the text "If the user is having google account, use the same username and password to login into the Picasa". At the bottom right of the page, there's a link for users who don't have a Google Account: "Create a new Google Account".

## Selenium Tutorial(IDE, Core, RC)

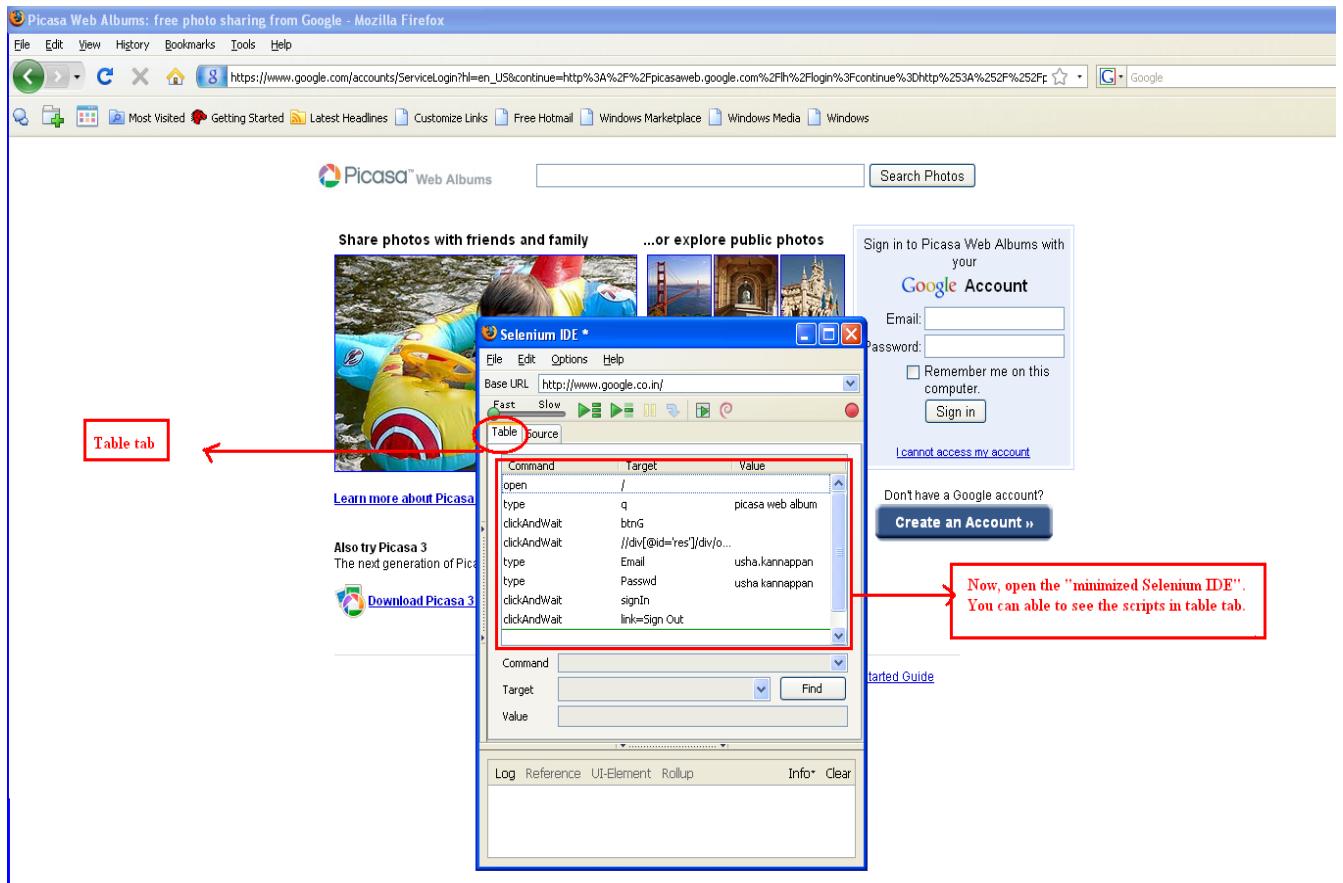
### Step 5:

Now login to the picasa webalbum. Whenever you login to any web page using your account, the next page is the welcome page for the user. Most of the website, display the username or else useraccount email id to display. In this example, it is displaying “[usha.kannappan@gmail.com](mailto:usha.kannappan@gmail.com)”. Next click the “signout” link.

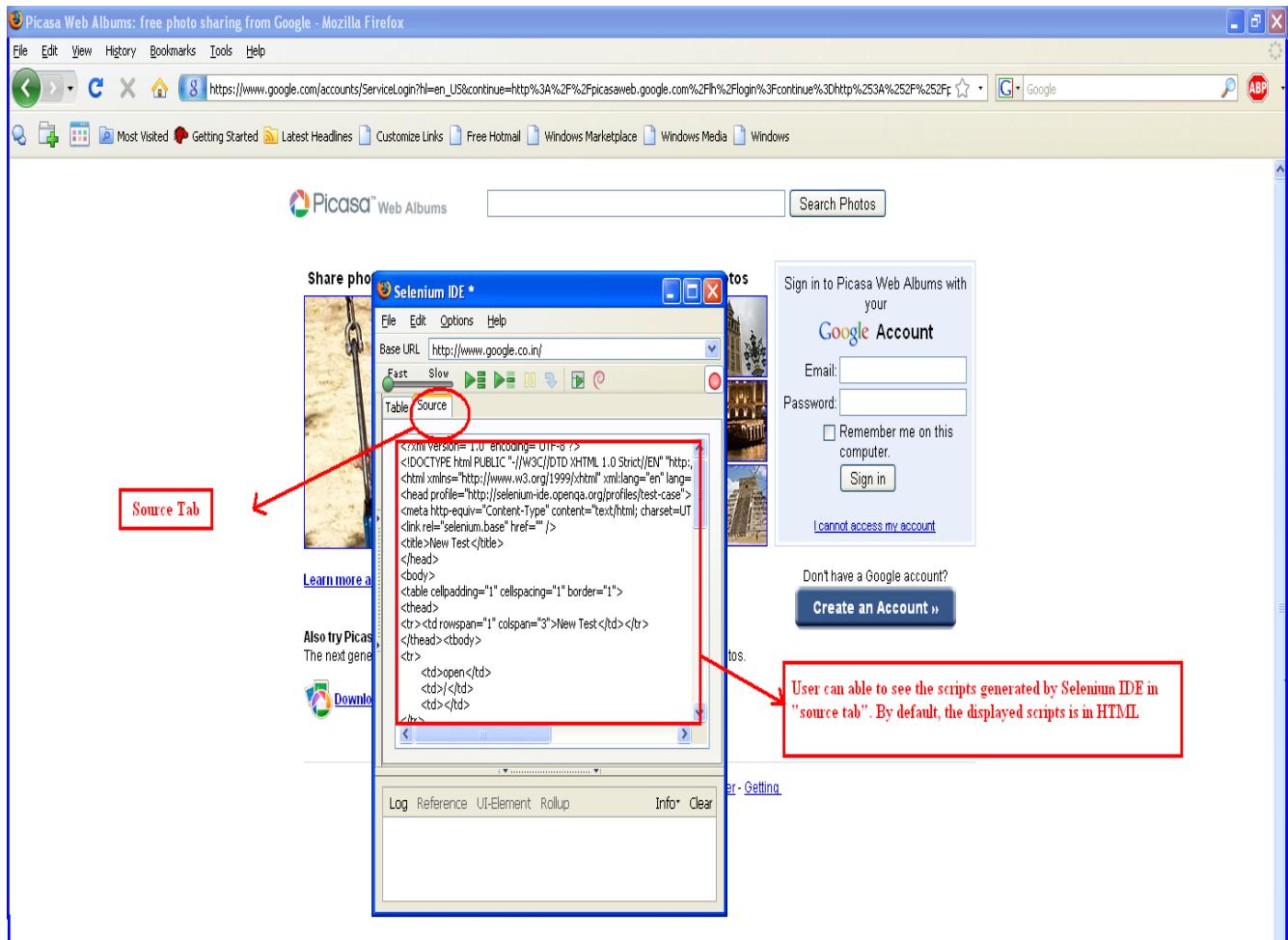


Now open the seleniumIDE, choose either “table” tab or “source” tab. In table tab, the steps has been generated in a table format. If you click the source tab, then the html + javascript coding has been generated.

## Selenium Tutorial(IDE, Core, RC)

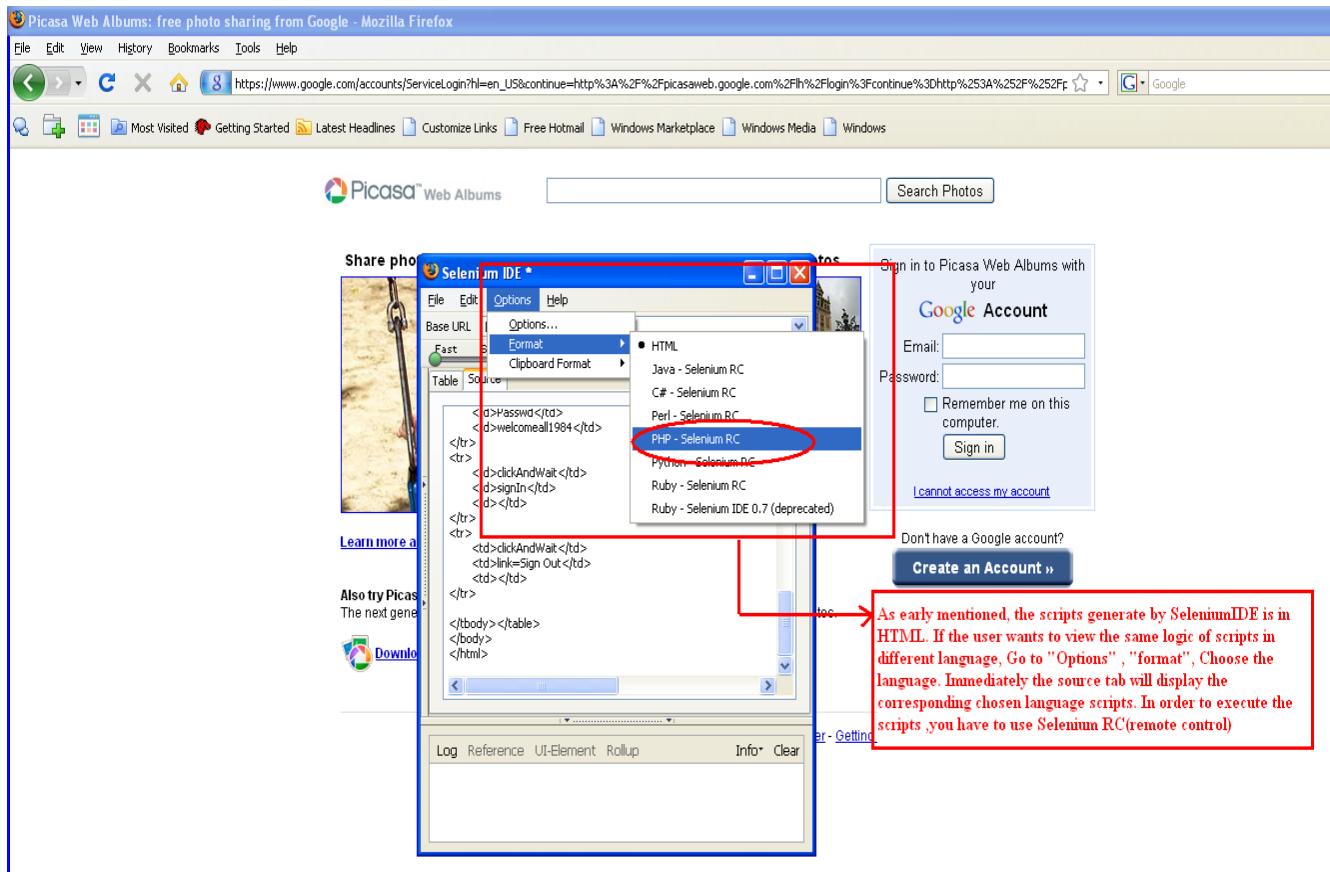


## Selenium Tutorial(IDE, Core, RC)



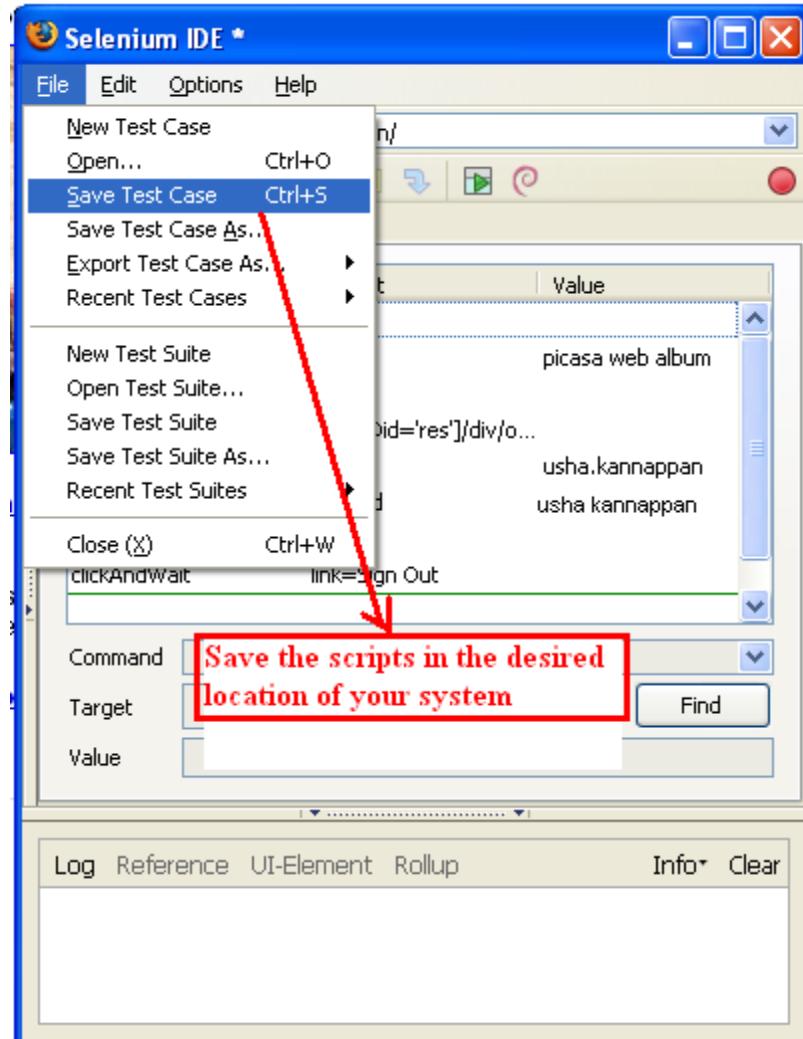
If the user wants to see the generated scripts in other programming/scripting language, Go to Option → format-> choose any other language. Source tab automatically update the scripts in the corresponding chosen language. Other programming scripts cannot be run in Selenium IDE, user has to use selenium RC(remote control) to execute the scripts. The purpose of this is, user can record and get the scripts for selenium RC not for selenium IDE. Here we are going to use HTML only.

## Selenium Tutorial(IDE, Core, RC)

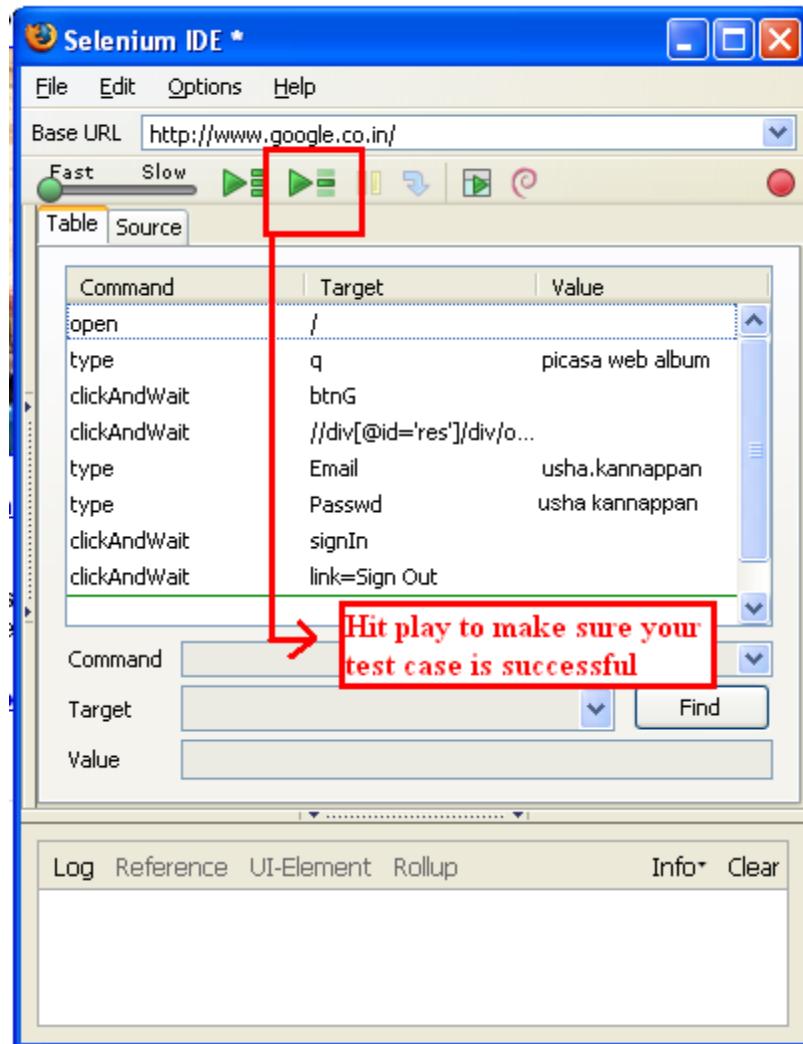


Now user can save the scripts in the desired location of the computer. Now click the run button and start executing the scripts. This will give an assurance to the user, the generated scripts is running properly. Later they can customize the script for validation, robustness, etc.,

## Selenium Tutorial(IDE, Core, RC)



## Selenium Tutorial(IDE, Core, RC)



Whenever Selenium execute, it will highlight the executing line in “Yellow” color , if the line executed successfully then the color will change to “Green” else failed line in “Red” color.

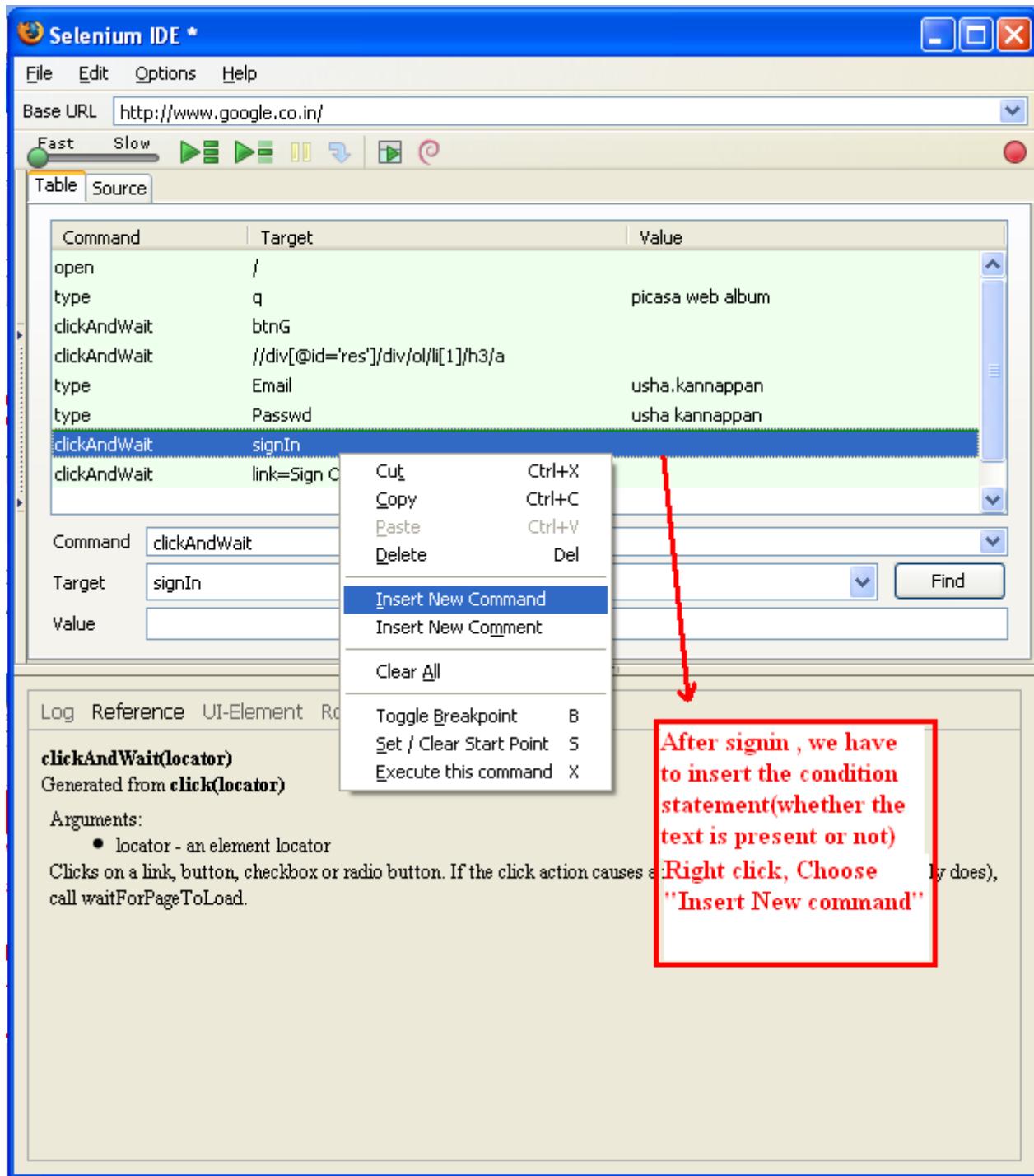
There are many things to check in webpage like text appeared or not, image displaying properly in different browser or not, button enabled or not etc.,,

Here in our example, we are going to check , after login the page is moving to welcome page. In welcome page, is the web page displaying the username (username or email id of the login) properly or not.

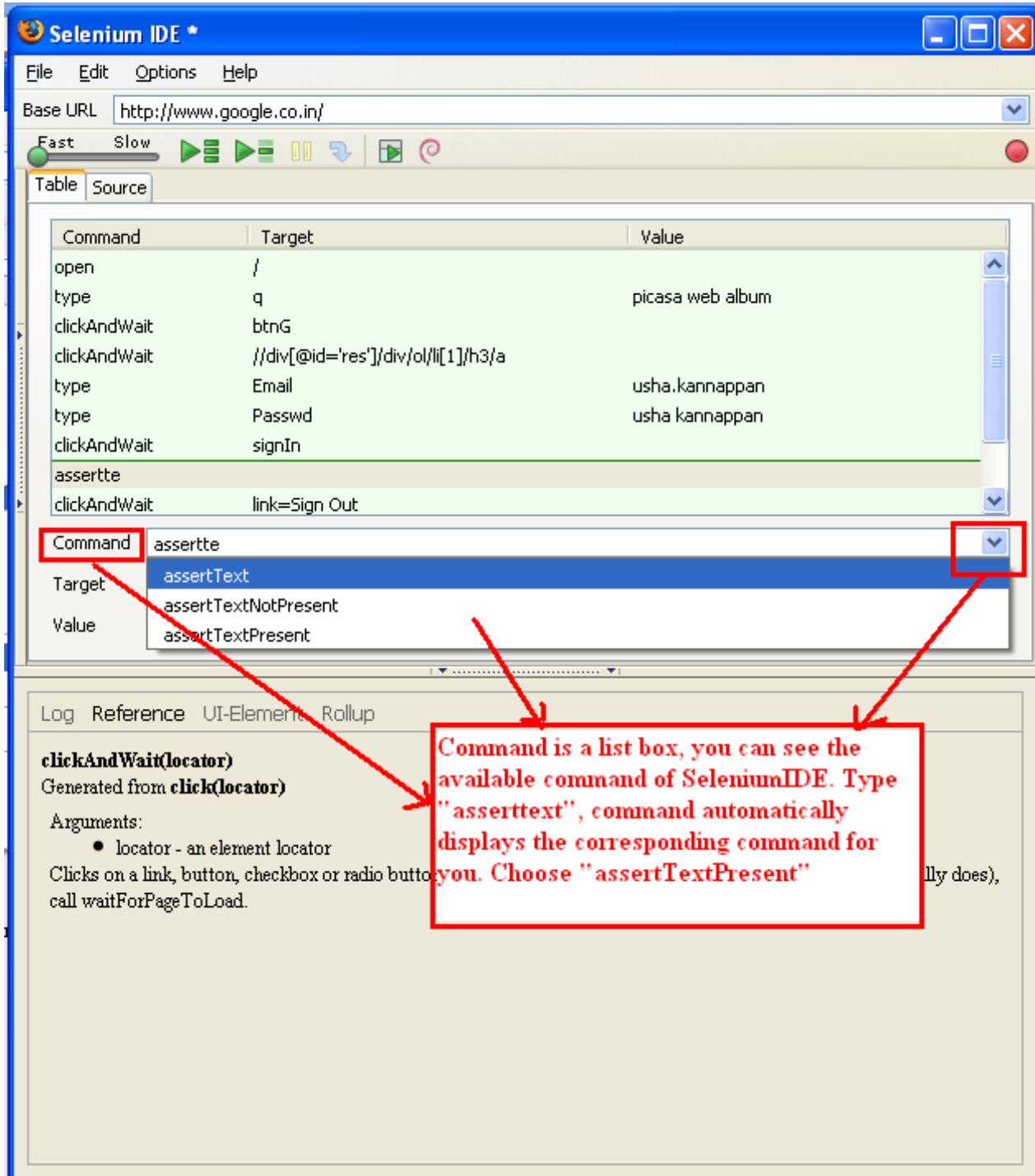
Now let's modify or tweak our generated scripts little bit.

**“Assert text present” is the inbuilt command present in Selenium IDE. This command will check whether the webpage is having the text or not.**

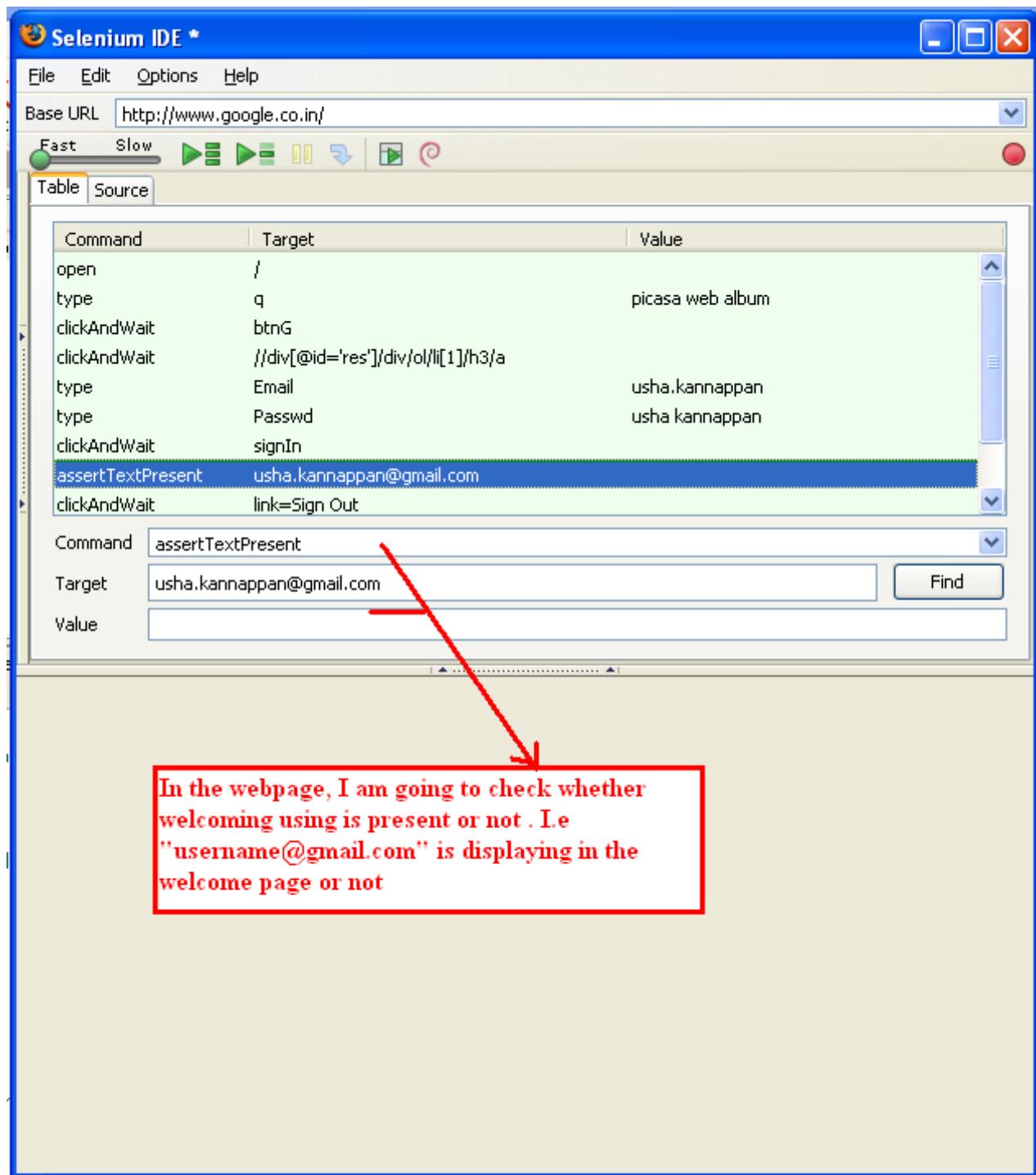
## Selenium Tutorial(IDE, Core, RC)



## Selenium Tutorial(IDE, Core, RC)



## Selenium Tutorial(IDE, Core, RC)



## Selenium Tutorial(IDE, Core, RC)

The screenshot shows the Selenium IDE interface with a script table and a status bar.

**Script Table:**

Command	Target	Value
open	/	
type	q	picasa web album
clickAndWait	btnG	
clickAndWait	//div[@id='res']/div/ol/li[1]/h3/a	
type	Email	usha.kannappan
type	Password	usha kannappan
clickAndWait	signIn	
assertTextPresent	usha.kannappan@gmail.com	
clickAndWait	link=Sign Out	

**Status Bar:**

- Log Reference UI-Element Rollup
- clickAndWait(locator)  
Generated from click(locator)
- Arguments:
  - locator - an element locator
- Clicks on a link, button, checkbox or r

**Annotations:**

- A red arrow points from the "assertTextPresent" row in the table to the status bar message.
- A red box highlights the status bar message: "The command executed successfully(Green color.)."
- A red box highlights the status bar message: "Most of the time, the command execute fastly by opening the window, typing username , password etc..,If the user wants to see specify steps, you can set breakpoint. When the line comes, it will wait , user can click the run button for further execution. Helpful for debugging"

After saving the scripts, user can open the scripts either in notepad or else in wordpad to view the scripts.

## Selenium Tutorial(IDE, Core, RC)

```
<?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 profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="" />
<title>New Test</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="3" colspan="3">New Test</td>
</tr>
<tr>
<td>open</td>
<td></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td></td>
<td>picasa web album</td>
</tr>
<tr>
<td>clickAndWait</td>
<td>btng</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>//div[@id='res']/div/ol/li[1]/h3/a</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>Email</td>
<td>usha.kannappan</td>
</tr>
<tr>
<td>type</td>
<td>Passwd</td>
<td>usha kannappan</td>
</tr>
<tr>
<td>clickAndWait</td>
<td>signIn</td>
<td></td>
</tr>
<tr>
<td>assertTextPresent</td>
<td>usha.kannappan@gmail.com</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>link=Sign Out</td>
<td></td>
</tr>
</tbody></table>
</body>
</html>
```

### Information:

This is an html command of selenium. Understanding this script is not a difficult task. The selenium tester should have the responsibility of making the scripts more robust. Though the tool is generating scripts, adding validation and identifying where needs to be checked are to be identified by tester. For example: if the scripts generates click then modify the line into ClickAndWait, so the selenium scripts will wait for the page to load. Otherwise it would result failed.

AssertTextPresent for checking whether the username is displayed in the welcome page or not

After asserting the scripts, now try to run the scripts. The scripts will execute successfully. If you give different username, the scripts will fail.

Congratulation!!! Now you can able to record and do testing in Selenium IDE.

## Selenium Tutorial(IDE, Core, RC)

Now we will see certain things in little advance.

1. The default commands generated by selenium when you are browsing the page as a normal user should be modified to make the test more robust and to add test cases to it.

For ex:

- Replace all **click** commands by **clickAndWait**. **click** simply clicks the specified link and goes on to execute the next command without waiting. On the other hand **clickAndWait** waits for the new page to loaded before executing the next command. **clickAndWait** should be used to make more robust test cases.
- Insert **assertTextNotPresent** command after each **clickAndWait** command to confirm a text must not be present in the browsed page.
- Use **assertTextPresent** command to confirm a text must be present in the browsed page.

Let's see the code present in Selenium IDE table tab.

The screenshot shows the Selenium IDE interface with the 'Table tab' selected. The table structure contains the following commands:

Command	Target	Value
open	/	
type	q	picasa web album
clickAndWait	btnG	
clickAndWait	//div[@id='res']/div/o...	
type	Email	usha.kannappan
type	Passwd	usha kannappan
clickAndWait	signIn	
clickAndWait	link=Sign Out	

Below the table, there is a search interface with fields for Command, Target, and Value, and a 'Find' button. A red box highlights the 'Table structure' field, which is the search input field.

At the bottom, the Log pane displays:

```
clickAndWait(locator)
Generated from click(locator)
Arguments:
```

## Selenium Tutorial(IDE, Core, RC)

When you look at the table , there are 3 columns

1. command
2. target
3. value

This is actually crux of selenium work.

Command	Target	Value
Command	Target	&nbsp (no value) , relevant value
Assertion	Expected	Actual

A command is what tells Selenium what to do. Selenium commands come in three 'flavors': Actions,Accessors and Assertions. Each command call is one line in the test table of the form.

Command	Target	Value
---------	--------	-------

### **Actions:**

Actions are commands that generally manipulate the state of the application. They do things like "click this link" and "select that option". If an Action fails, or has an error, the execution of the current test is stopped.

Many Actions can be called with the "AndWait" suffix, e.g. "clickAndWait". This suffix tells Selenium that the action will cause the browser to make a call to the server, and that Selenium should wait for a new page to load.

### **Accessors:**

Accessors examine the state of the application and store the results in variables, e.g. "storeTitle". They are also used to automatically generate Assertions.

### **Assertions:**

Assertions are like Accessors, but they verify that the state of the application conforms to what is expected. Examples include "make sure the page title is X" and "verify that this checkbox is checked".

All Selenium Assertions can be used in 3 modes

1. assert
2. verify
3. waitfor

## Selenium Tutorial(IDE, Core, RC)

For example, you can "**assertText**", "**verifyText**" and "**waitForText**". When an "**assert**" fails, the test is aborted. When a "**verify**" fails, the test will continue execution, logging the failure. This allows a single "**assert**" to ensure that the application is on the correct page, followed by a bunch of "**verify**" assertions to test form field values, labels, etc.

"**waitFor**" commands wait for some condition to become true (which can be useful for testing Ajax applications). They will succeed immediately if the condition is already true. However, they will fail and halt the test if the condition does not become true within the current timeout setting (see the **setTimeout** action below).

### TARGET:

Element Locators tell Selenium which HTML element a command refers to. The format of a locator is:

***locatorType =argument***

selenium supports the following strategies for locating element:

1. id
2. name
3. xpath
4. dom
5. identifier
6. link
7. css

### **id and name:**

Whenever you open any web page, right click and try to view the source code. You will come across id or name for the web object.

Based on this , user can uniquely identify the webobjects such as textbox, listbox, radio button etc., Each will be having it's own id , name.

In our example:

we used google as the webpage. “q” is the name for the “**google textbox**”. That is the reason,first line of our scripts had “q” as the target and “Picasaweb album “ as the value we typed in the text box

Command	Target	Value
open	/	
type	q	picasa web album

## Selenium Tutorial(IDE, Core, RC)

### xpath:

To put it in a simple way, xpath is like hierarchy or the order used to identify any object.

In out Example:

```
clickAndWait //div[@id='res']/div/o...
```

### Link:

Select the link (anchor) element which contains text matching the specified *pattern*.

- link=The link text

### CSS:

Select the element using css selectors.

```
css=a[href="#id3"]
```

### DOM:

dom=javascriptExpression.

User can find an element by evaluating the specified string.

```
dom=document.forms['myForm'].myDropdown
```

- dom=document.images[56]
- dom=function foo() { return document.links[1]; }; foo();

After reading all these , few tester would have an idea to drop down to use selenium. The reason might be, more of codes and finding the objects(element) unique name, id or else css name, etc.,,

No worries guys, No need to break your head to find it out.

There is a tool called “**FIREBUG**”. Using this tool, user can able to identify any **object's id, name,xpath, CSS, dom , link, etc..**,

Now let's look on to the firebug installation and continue our selenium IDE.

## Selenium Tutorial(IDE, Core, RC)

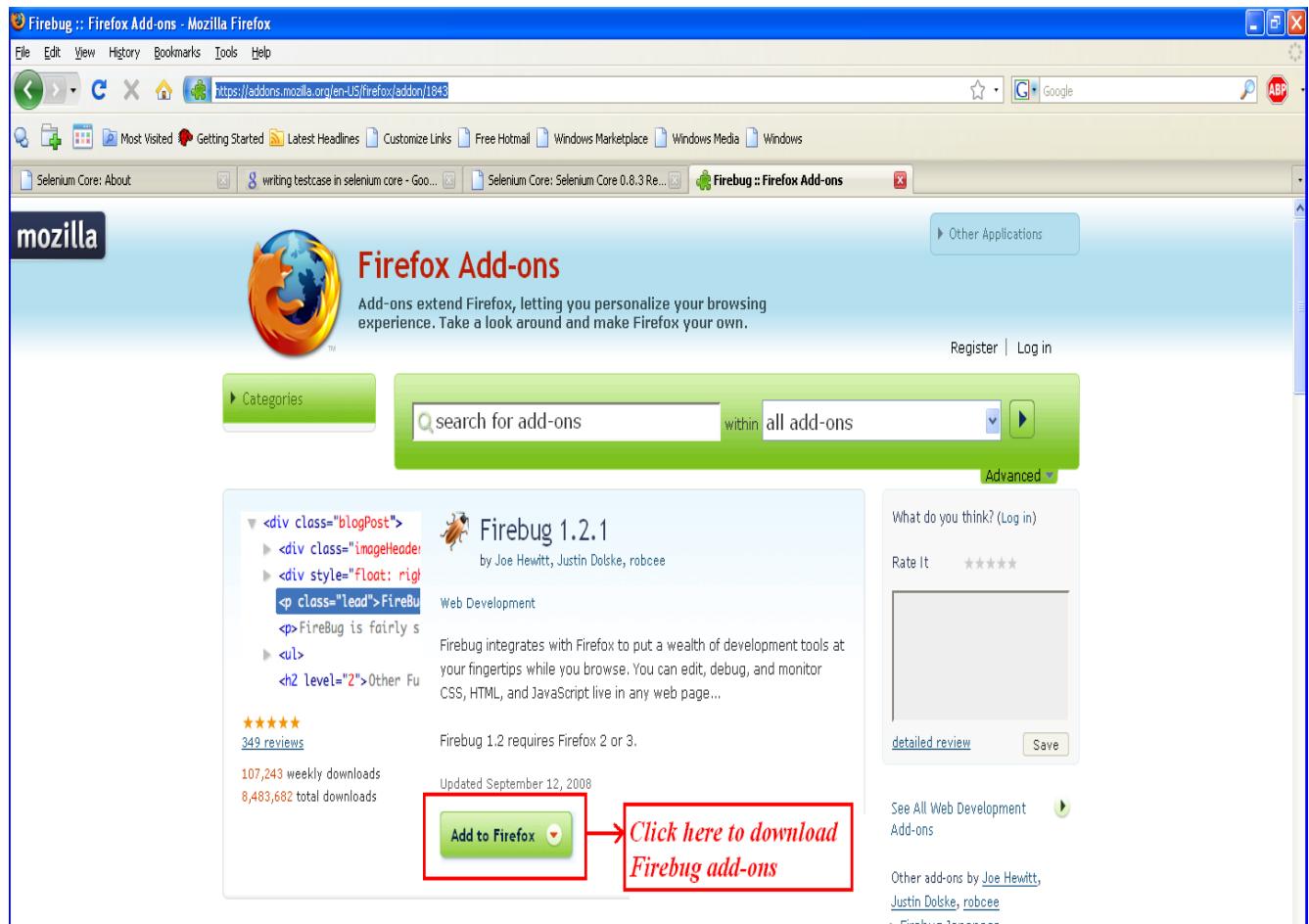
### FIREBUG:

Firebug is add on available for Mozilla Firefox.

#### Step1:

User can get it in the following link.

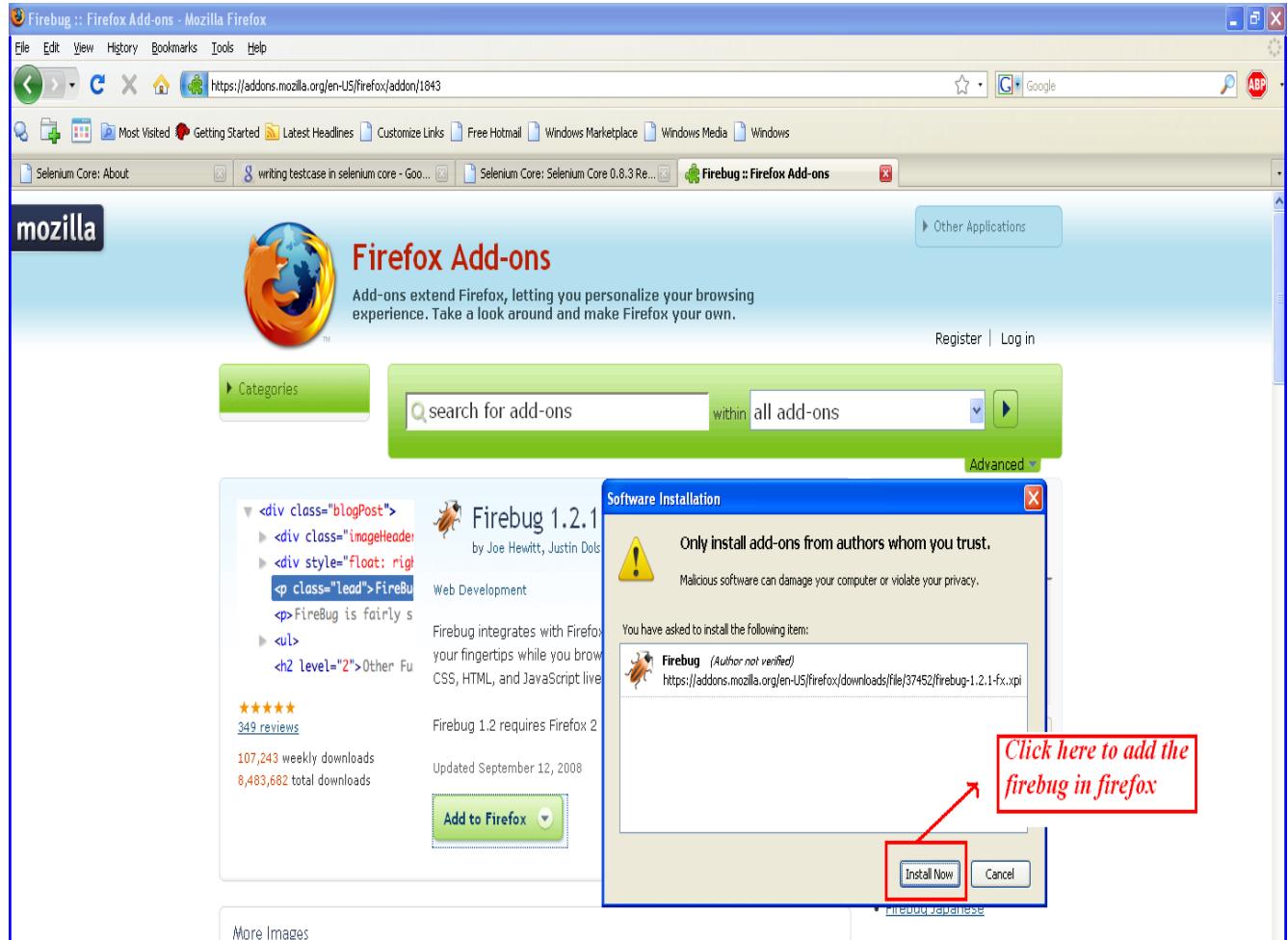
<https://addons.mozilla.org/en-US/firefox/addon/1843>



## Selenium Tutorial(IDE, Core, RC)

### Step 2:

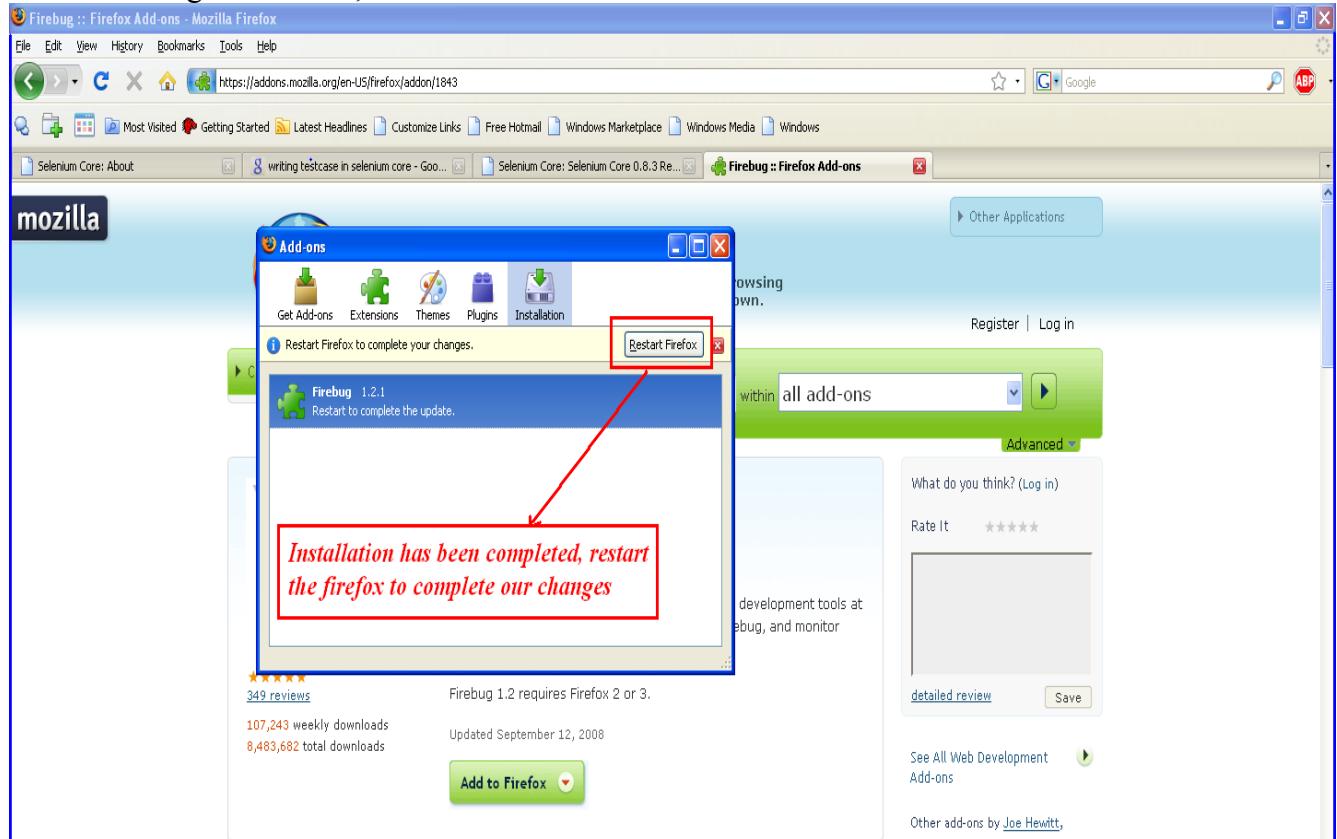
Sometimes software will be blocked by the browser , So click and allow for downloading and installation.



## Selenium Tutorial(IDE, Core, RC)

### Step 3:

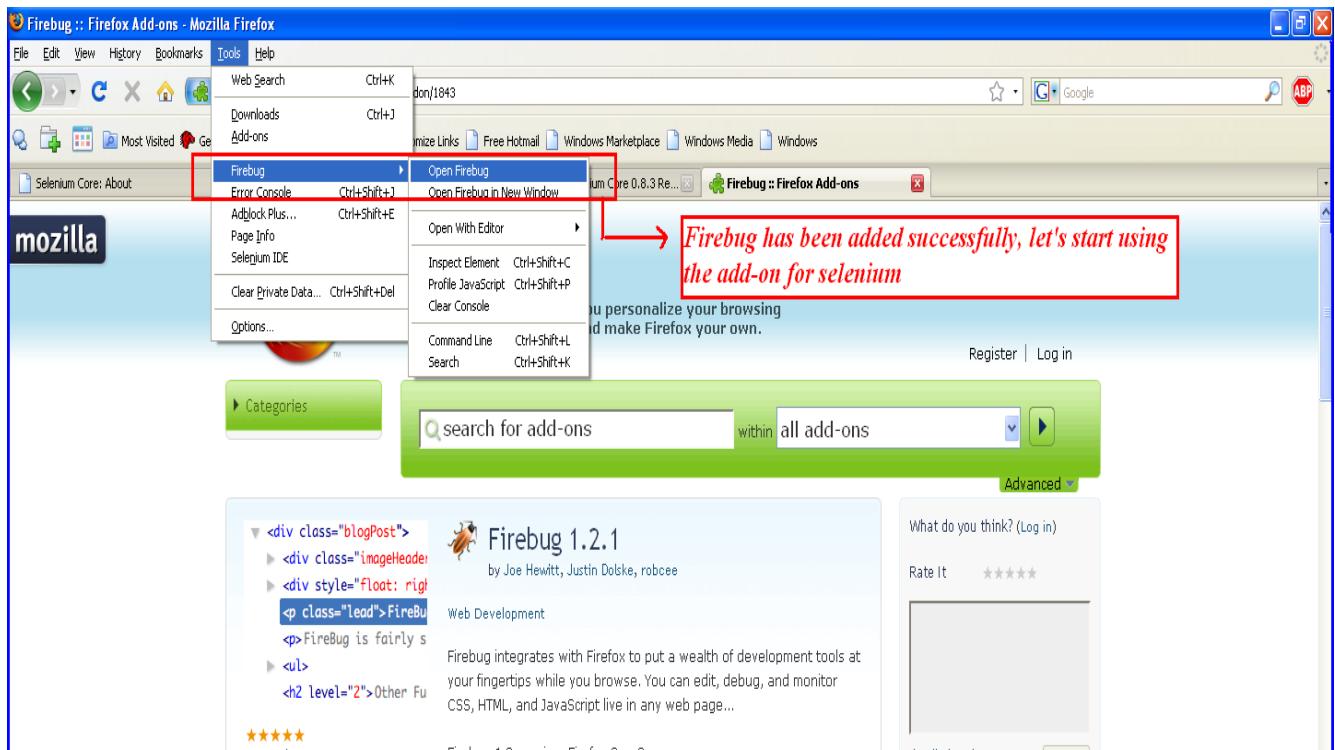
After installing the add-on , restart the browser.



## Selenium Tutorial(IDE, Core, RC)

### Step 4:

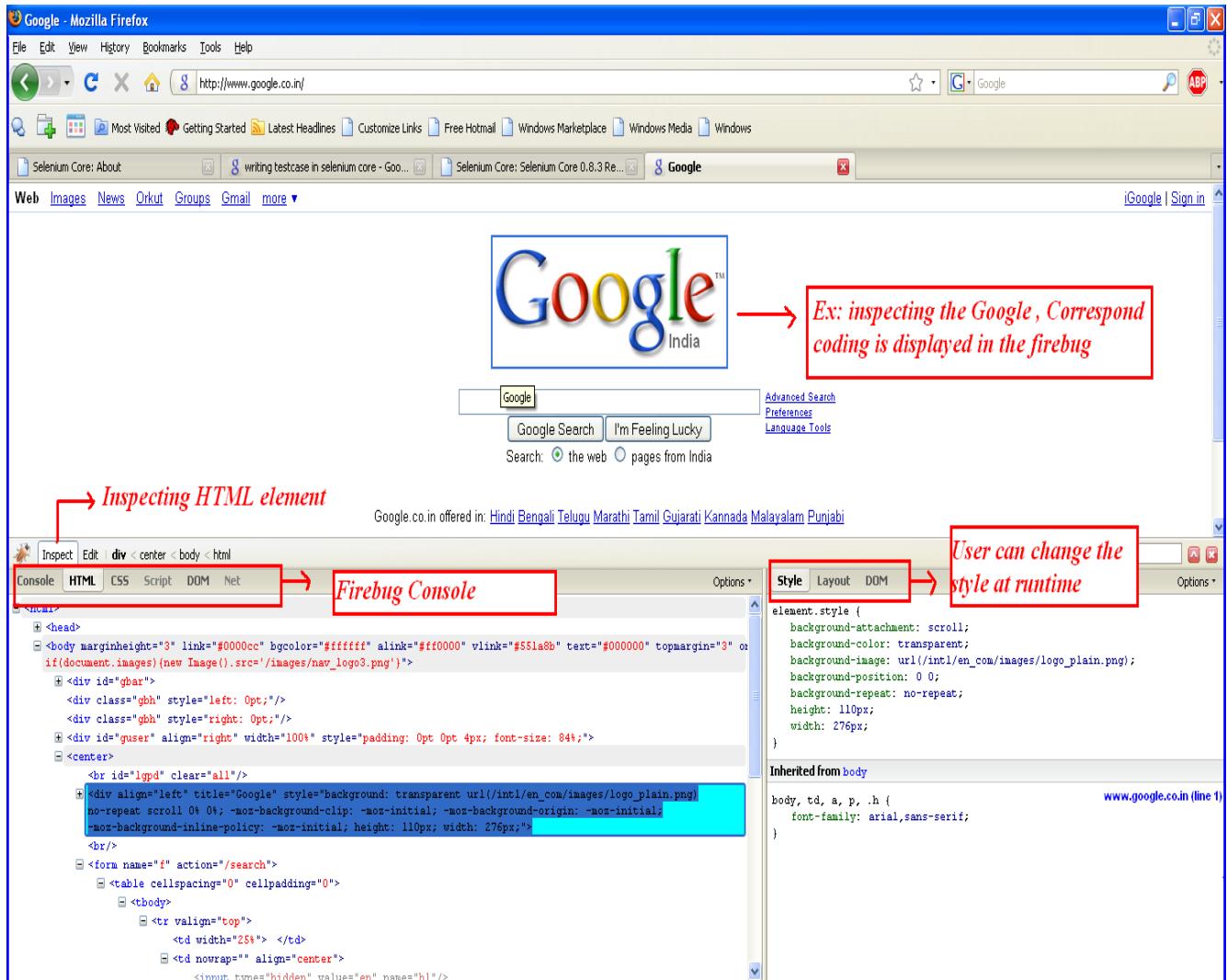
Installation has been successfully done. Click the tools , firebug and open firebug.



## Selenium Tutorial(IDE, Core, RC)

### Step 5:

Now user can see the Firebug window in the lower part of the browser. Click the inspect , then wherever you are moving the corresponding coding will be displayed in the console of firebug.



Using this user can find the target value and make the testcases more robust.

## Selenium Tutorial(IDE, Core, RC)

### **Advantages:**

- It is a Freeware
- Simple, Easy to install, Easy to work
- Selenium IDE is the only flavor of Selenium which allows you to record user action on browser window
- Can also record user actions in most of the popular languages like Java, C#, Perl, Ruby
- It will not record any operation that you do on your computer apart from the events on Firefox browser window
- During recording if you right click on any element it will show all the selenium commands available
- You can also edit existing command, by selecting it and editing on the boxes available
- You can also insert/delete commands by choosing appropriate option after right clicking
- Choose appropriate run option - i.e walk, run or test runner and review your results

### **Disadvantages:**

1. Limitation in terms of browser support (It runs only in Mozilla)
2. We can't run recorded script if it is converted to Java, C#, Ruby etc.
3. Not allowed to write manual scripts like conditions and Loops for Data Driven Testing
4. **There is no option to verify images.**

**Congratulation Reader!!! Now you can start exploring SeleniumIDE on your own and do testing. The content given above is just a glimpse not for the expert level of SeleniumIDE. The more user practicing the scripts in real time will give more idea of using the Selenium IDE. “ALL THE BEST” for it.**

## Selenium Tutorial(IDE, Core, RC)

### SELENIUM CORE:

Selenium Core requires a web server and the Selenium Core zip file. Here I am using Apache as my web server. User can even use other web server (**IIS, Tomcat, etc..**) for selenium core

### Installation instruction for Selenium Core:

1. Web server-Apache
2. Selenium -Core

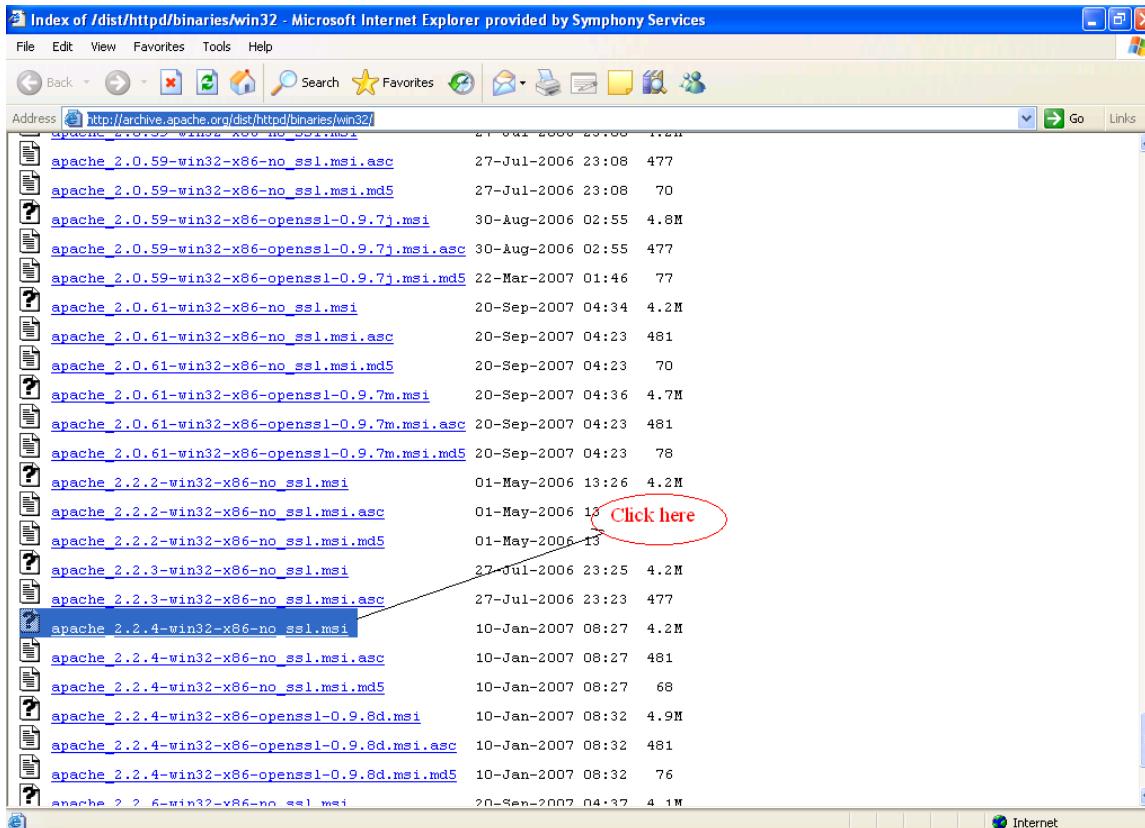
1. Download and install apache2.2.4(download the latest version of

Click the following link and download the Apache2.2.4:

<http://archive.apache.org/dist/httpd/binaries/win32/>

Search for the following **msi**( windows installer)

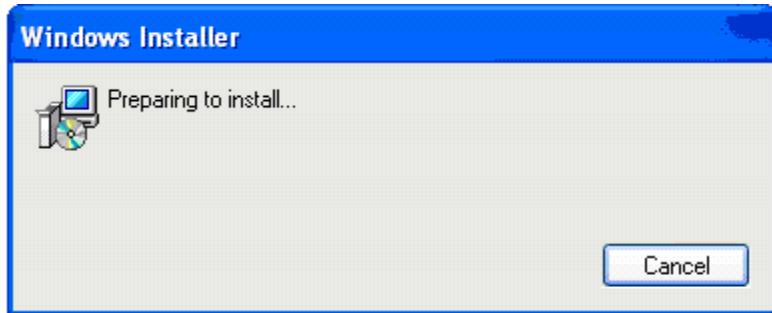
**apache\_2.2.4-win32-x86-no\_ssl.msi**



## Selenium Tutorial(IDE, Core, RC)

### Step 1:

Now we're ready to install. “**Double-click**” the installer.



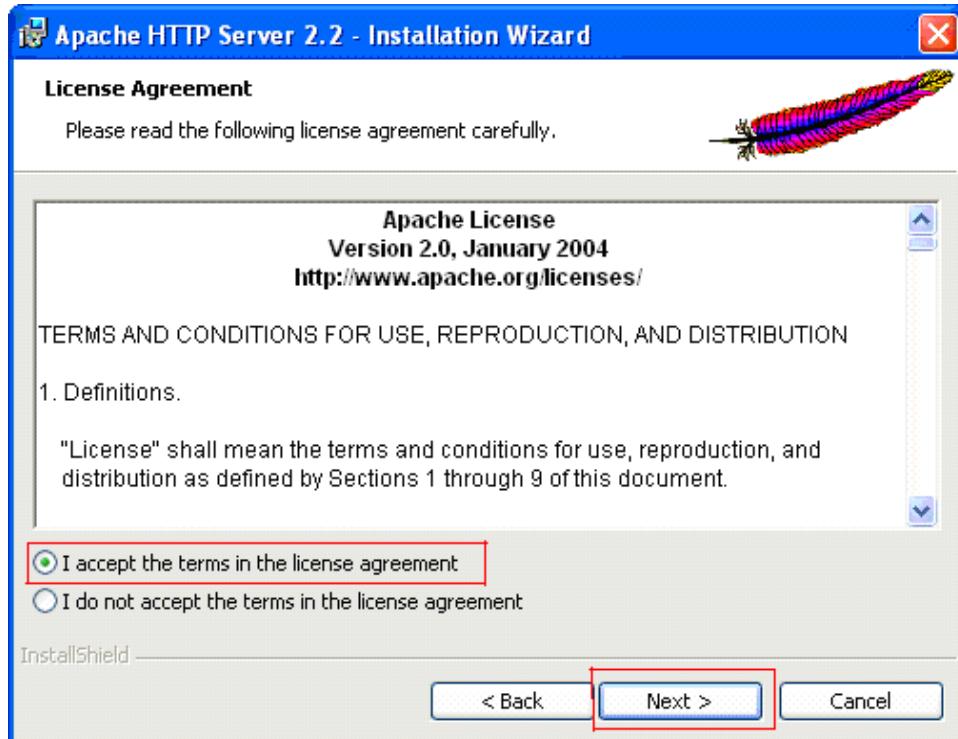
### Step 2:

Click the “**next**” button in the Installation Wizard



## Selenium Tutorial(IDE, Core, RC)

**Step 3:** Accept the License agreement and click the “next” button



**Step 4:**

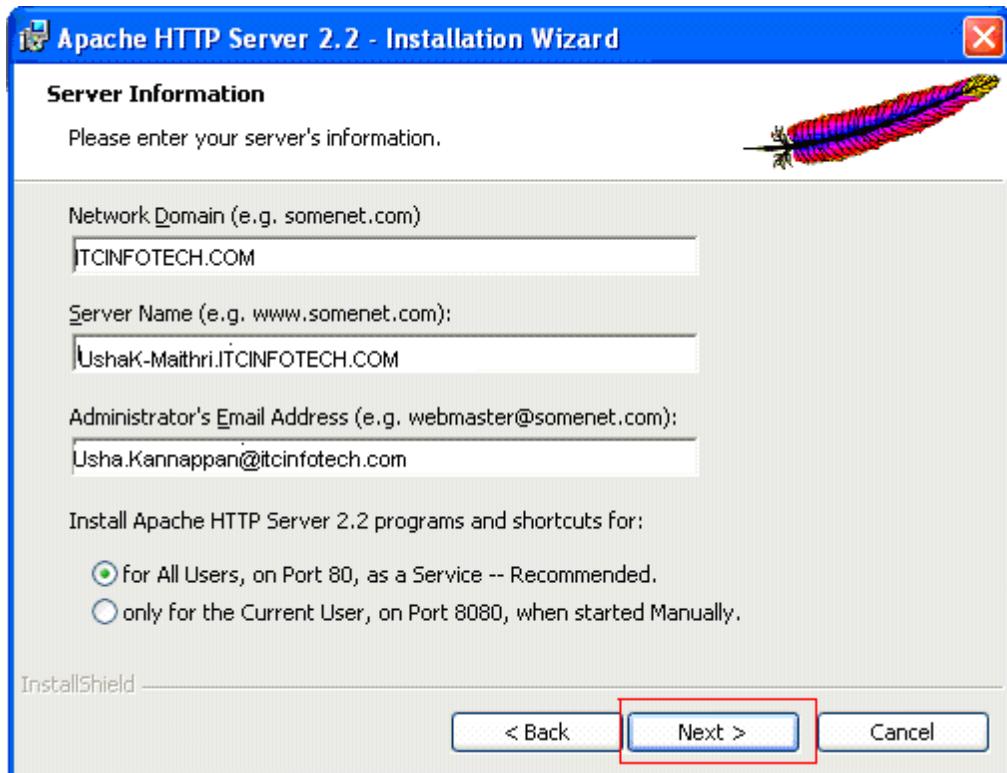
Provide the Server information details in the window.

**Domain Name :** ITCINFOTECH.COM

**Server Name :** The computer name

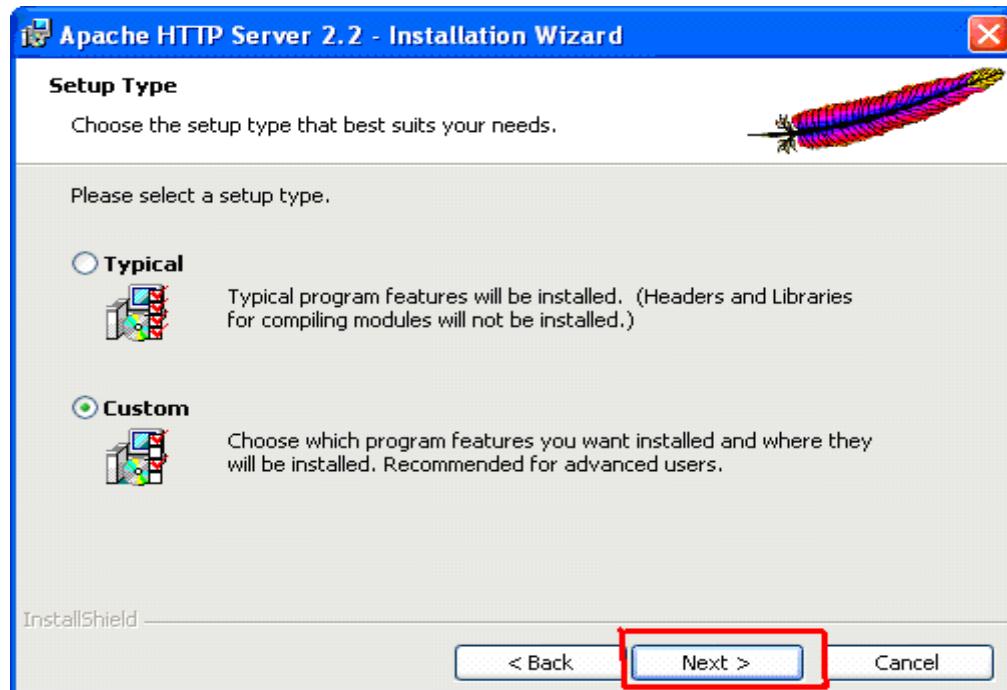
**Administrator's Email id:** Provide administrator email id. (If any error or problem in application you will get the information “asking the user to contact the administrator” For Ex: “**Please contact usha.kannappan@itcinfotech.com**”

## Selenium Tutorial(IDE, Core, RC)



### Step 5:

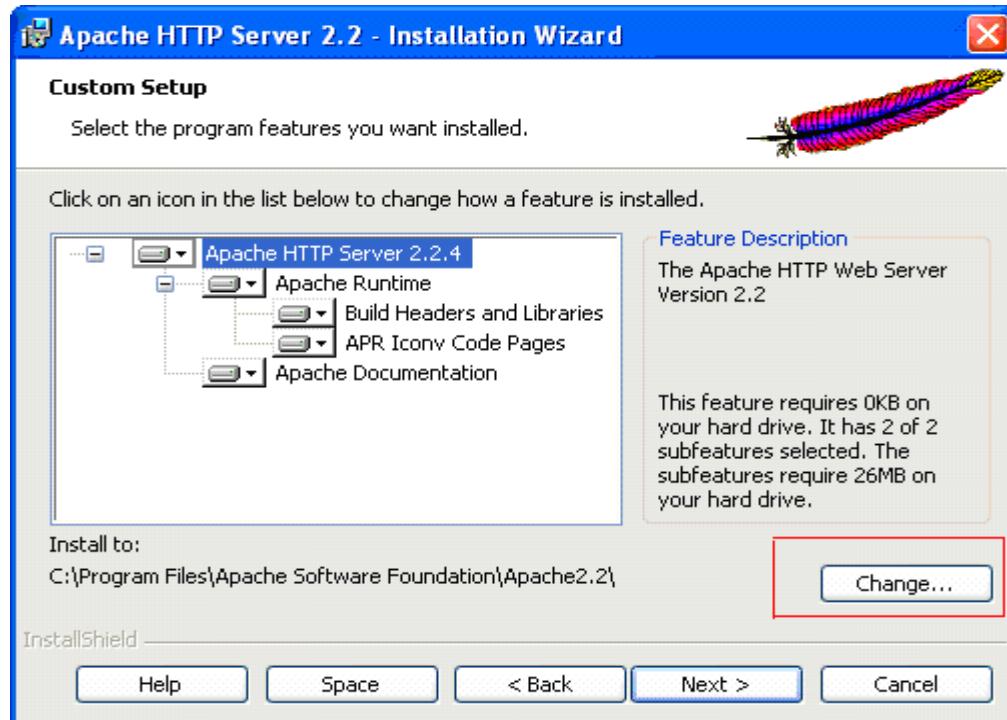
Choose "Custom" and press "Next".



## Selenium Tutorial(IDE, Core, RC)

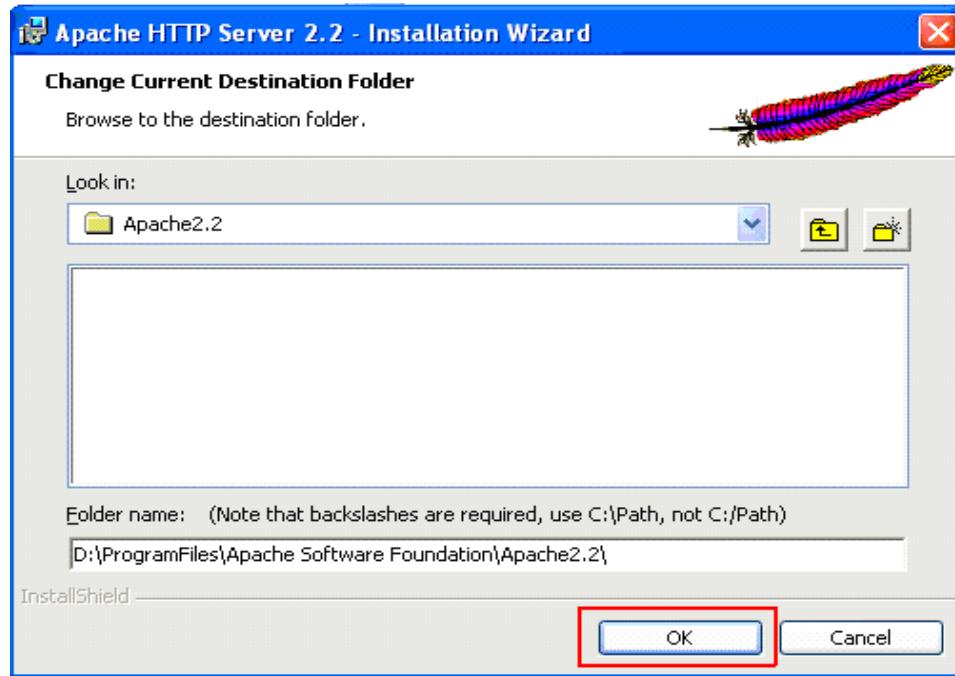
### Step 6:

On the following screen you will see the default path of Apache to installs. Click the "Change..." , and choose the **D:\ProgramFiles** folder (or whatever other name you chose) you made when you created the directory structure above. Click "Next".



### Step 7:

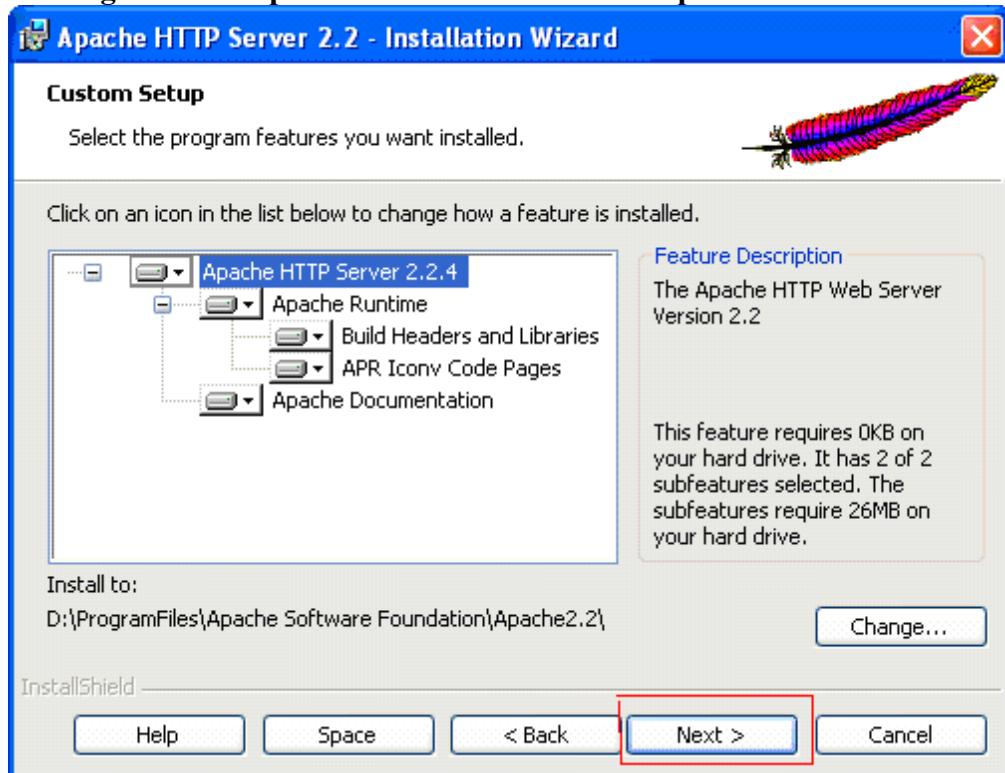
Continuation...



## Selenium Tutorial(IDE, Core, RC)

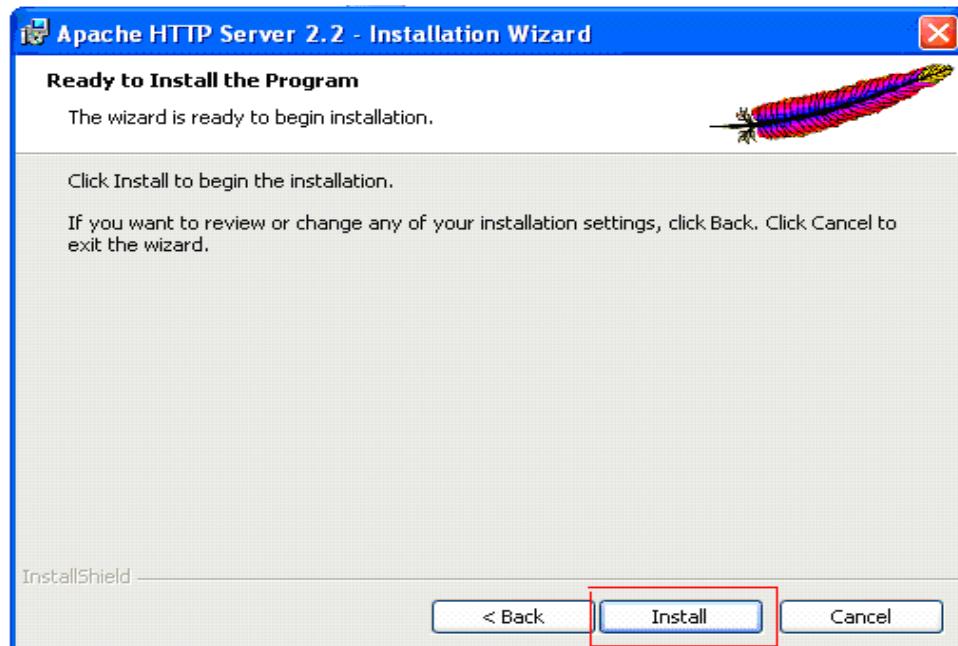
### Step 8:

Create a folder “Apache Software Foundation\Apache2.2”. The directory structure is like this:  
**D:\ProgramFiles\Apache Software Foundation\Apache2.2\**



### Step 9:

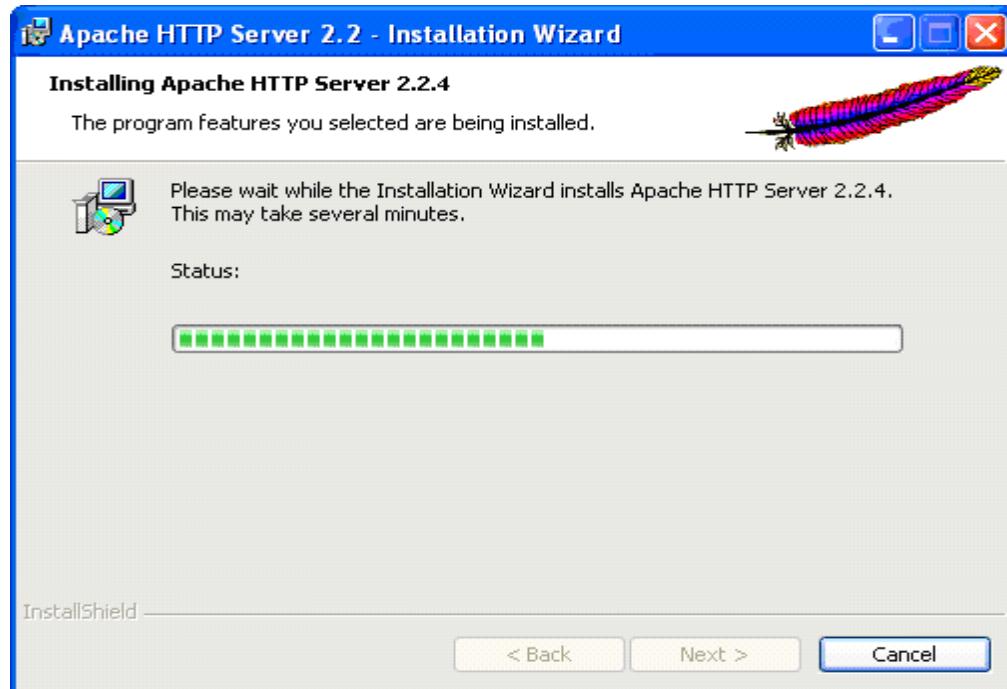
Click the “install” button and proceed further for installation.



## Selenium Tutorial(IDE, Core, RC)

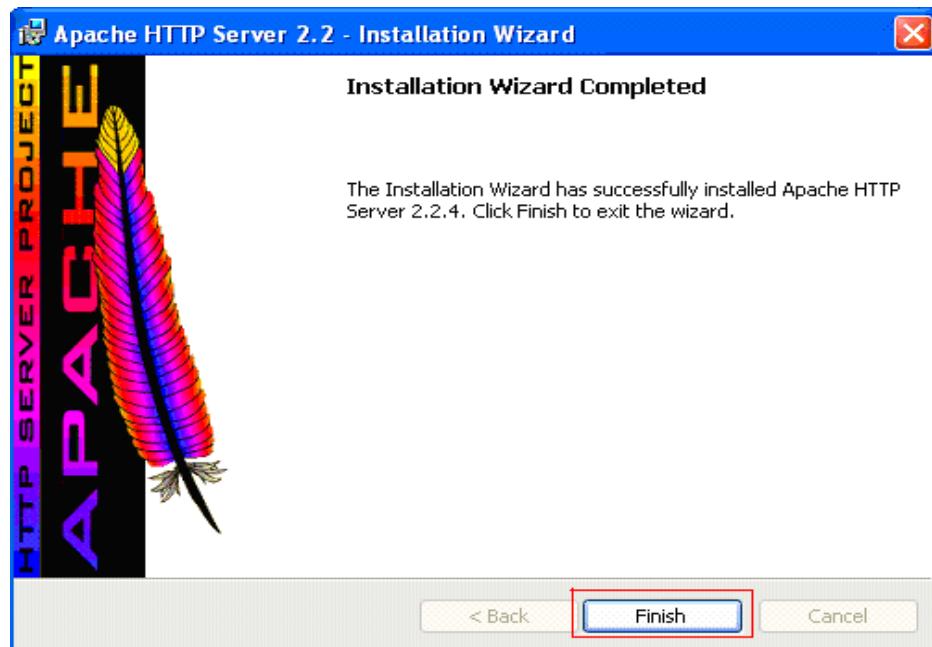
### Step 10:

You can see the status of **Installation status** in the Installation wizard.



### Step 11:

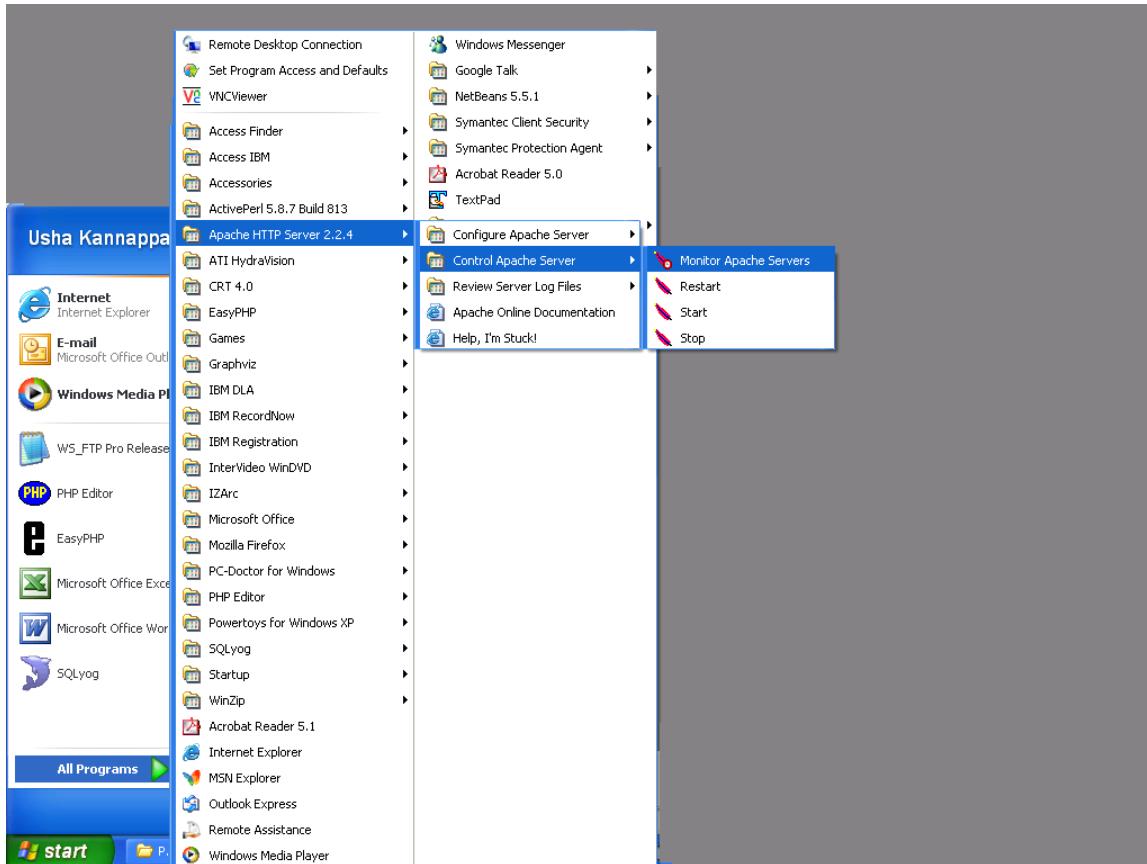
Once the installation has been completed, you will be getting the following window. Click **finish**.  
**Congratulation, apache 2.2.4 installation has been successfully finished.**



## Selenium Tutorial(IDE, Core, RC)

### Step 12:

Let's make sure it worked. Go to your programs folder in the Start menu and find the Apache HTTP Server folder. Go to the Control Apache Server folder, and start Apache

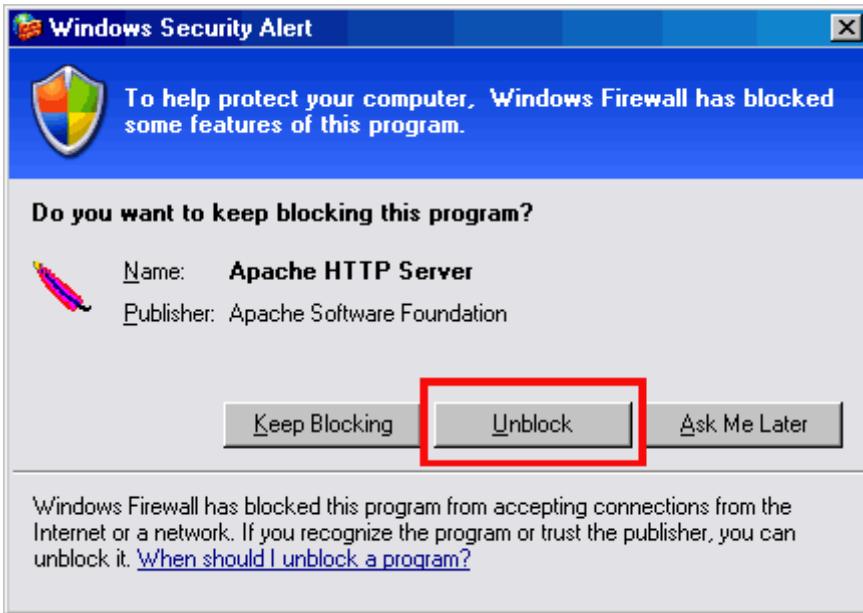


If you installed Apache as a service, you can also start it by double-clicking on the Apache Service

Monitor tray icon (which looks like: ) and pressing the "Start" button in the window that appears. If you did not install it as a service, you will see a blank command prompt window appear while Apache is running. This is normal.

The first time you run Apache, you will probably see a security warning pop up:

## Selenium Tutorial(IDE, Core, RC)



This is because windows has detected that Apache is opening a port on your computer, and it is checking with you about whether it should let Apache continue. Click the "**Unblock**" button to allow Apache to function.

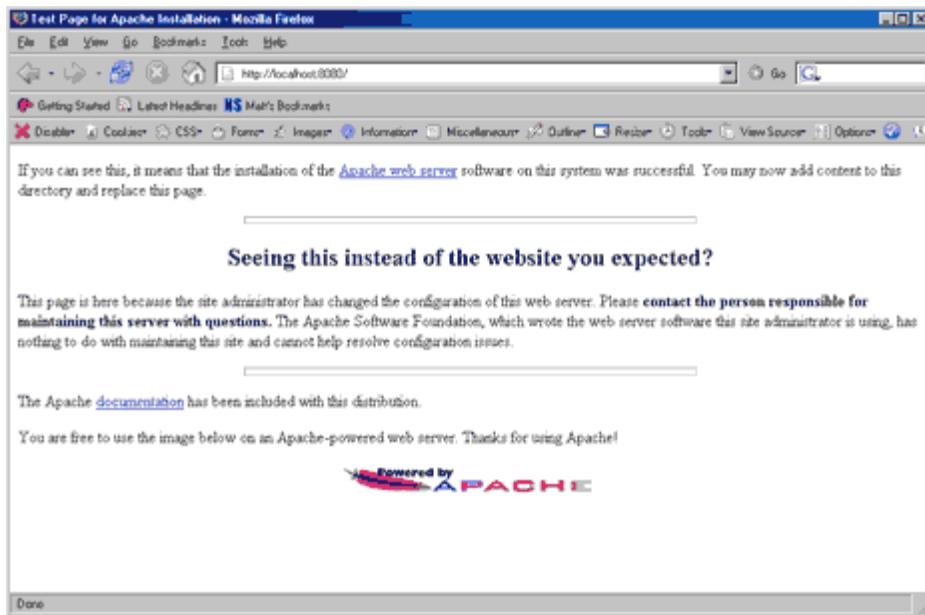
Now, let's open up a web browser and make sure Apache is running. Type in the name of your server (e.g. **local host, or your domain name or IP address**) in the location bar of the browser.

<http://localhost/>

or

<http://localhost:8080/>

## Selenium Tutorial(IDE, Core, RC)



If so, you're ready to move on. If not, something is wrong. It is hard to be sure what, but it might help to ask for assistance in the [ERT forum](#). Otherwise, your best bet is to uninstall Apache and try again.

Before proceeding to forum, check whether the following fields are having correct data and also these are the lines where user can modify.

**Listen 80**

**Port 80**

**ServerType** standalone

**ServerName** UshaK-Maithri-ITCINFOTECH.COM:80

**ServerAdmin** [Usha.Kannappan@ITCINFOTECH.com](mailto:Usha.Kannappan@ITCINFOTECH.com)

## **Selenium Tutorial(IDE, Core, RC)**

### **Download Selenium Core:**

Click the following link to download the Selenium Core. Try to download the latest version.

**Note:** The latest version always available at the top of the page.

**<http://selenium-core.openqa.org/download.jsp>**

## **Selenium Tutorial(IDE, Core, RC)**

## **Selenium Tutorial(IDE, Core, RC)**

## Selenium Tutorial(IDE, Core, RC)

The screenshot shows a Mozilla Firefox browser window with the title bar "Selenium Core: Download - Mozilla Firefox". The address bar contains the URL "http://selenium-core.openqa.org/download.jsp". The main content area displays the "Selenium-Core: Download" page from OpenQA.org. On the left is a sidebar with navigation links under categories like "Essential", "Evaluating", "Using It", and "Improving It". The right side lists several versions of Selenium Core with their details. A red box highlights the "Full release" link for Version 0.8.3, and a red arrow points from this link to a red-bordered callout box containing the text "Click here to download the latest version of Selenium Core".

**Selenium-Core: Download**

**Version 0.8.3**

- Date: 20-Sep-2007
- [Changelog](#)
- Downloads:
  - [Full release](#)

**Version 0.8.2**

- Date: 11-Dec-2006
- [Changelog](#)
- Downloads:
  - [Full release](#)

**Version 0.8.1**

- Date: 13-Nov-2006
- [Changelog](#)
- Downloads:
  - [Full release](#)

**Version 0.8.0**

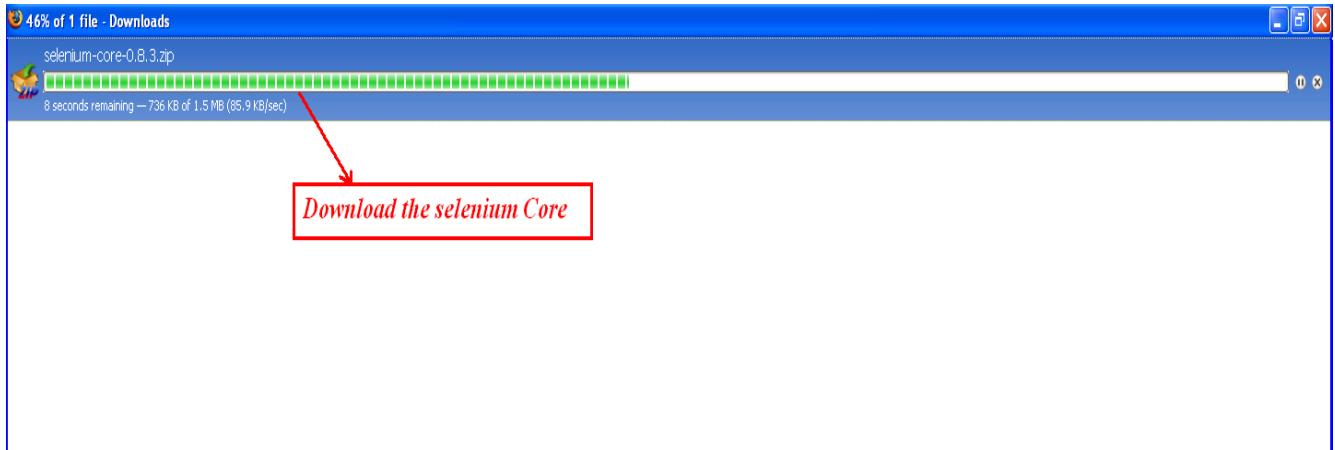
- Date: 20-Sep-2006
- [Changelog](#)
- Downloads:
  - [Full release](#)

**Version 0.7.1**

- Date: 3-Aug-2006
- [Changelog](#)
- Downloads:
  - [Full release](#)

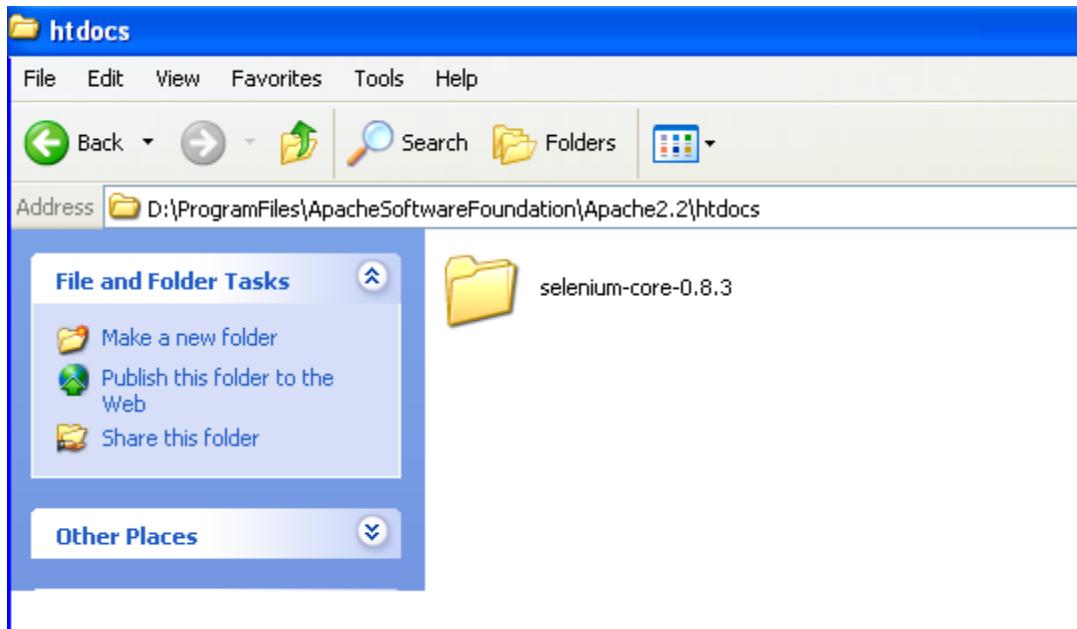
### Step 2: Downloading the selenium core

## Selenium Tutorial(IDE, Core, RC)



Step 3:

Unzip the selenium-core0.8.3 then copy the extracted one to the root directory of Apache2.2.4("D:\ProgramFiles\ApacheSoftwareFoundation\apache2.2.4"\Selenium-core0.8.3)

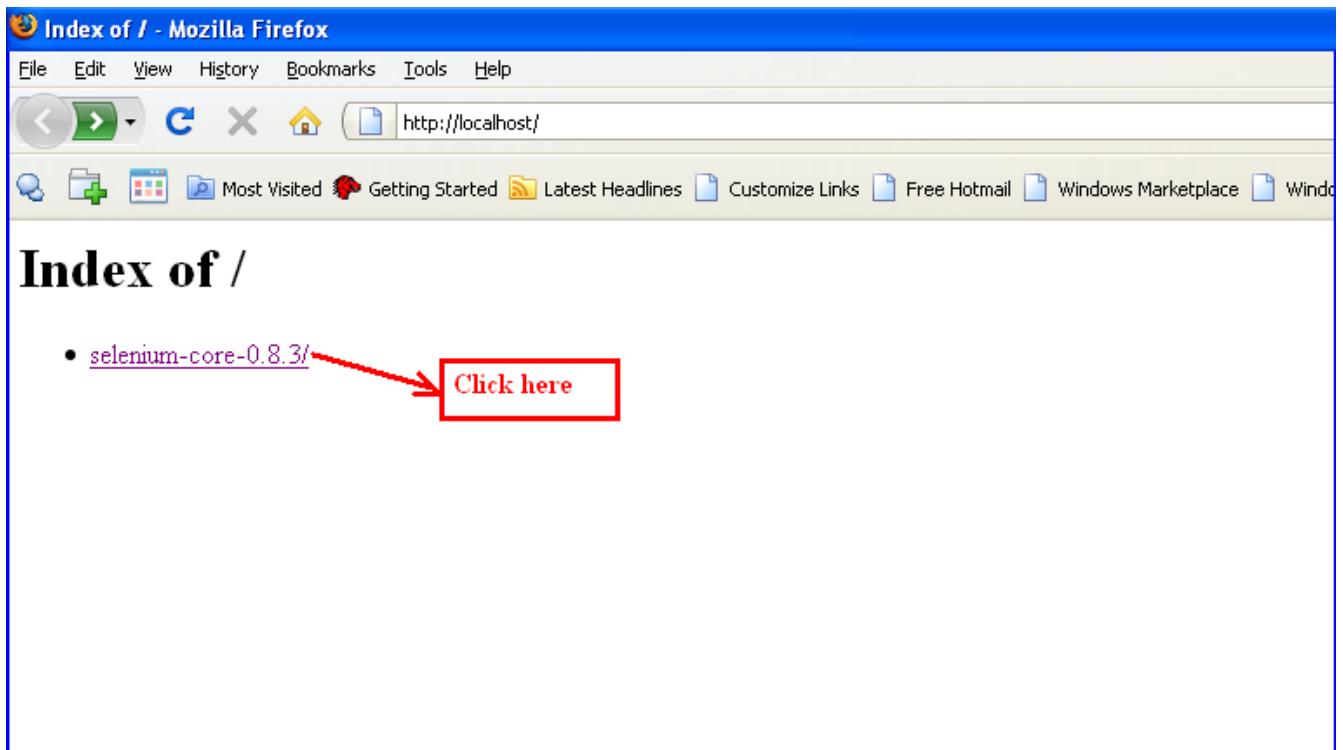


Now , open the Internet Explorer (any browser such as firefox , netscape, etc...), type the following

<http://localhost/>

## Selenium Tutorial(IDE, Core, RC)

User can able to see the Selenium-core folder available in the webpage. Click it.



Now the page will be having the Selenium core related information. By default, selenium core comes with an example. Using selenium core, user can do Acceptance and Unit testing. If you click the link such as Selenium Test runner, then you will move the selenium Testing home page.

## Selenium Tutorial(IDE, Core, RC)

The screenshot shows the Mozilla Firefox browser window with the title "Selenium starting points - Mozilla Firefox". The address bar displays "http://localhost/selenium-core-0.8.3/". The page content is as follows:

**Selenium**

[Selenium TestRunner](#) - Select a test suite to run in Selenium

**Acceptance tests:** These test-suites demonstrate/exercise the functionality of Selenium.

- [Selenium TestSuite](#) - functional tests for Selenium. This suite of tests should pass in any supported browser.
- [Error Checking TestSuite](#) - tests for the error verification commands
- [Dogfood TestSuite](#) - test Selenium with Selenium

**Unit-tests:** Use JsUnit to test Selenium internals.

- [Selenium BrowserBot unit-tests](#)
- [Selenium TestRunner UI unit-tests](#)
- [JsMock unit-tests](#)

By default, Selenium Core is having sample testcases for Acceptance & Unit tests

The screenshot shows the Mozilla Firefox browser window with the title "Selenium starting points - Mozilla Firefox". The address bar displays "http://localhost/selenium-core-0.8.3/". The page content is as follows:

**Selenium**

[Selenium TestRunner](#) Select a test suite to run in Selenium

**Acceptance tests:** These test-suites demonstrate/exercise the functionality of Selenium.

- [Selenium TestSuite](#) - functional tests for Selenium. This suite of tests should pass in any supported browser.
- [Error Checking TestSuite](#) - tests for the error verification commands
- [Dogfood TestSuite](#) - test Selenium with Selenium

**Unit-tests:** Use JsUnit to test Selenium internals.

- [Selenium BrowserBot unit-tests](#)
- [Selenium TestRunner UI unit-tests](#)
- [JsMock unit-tests](#)

Selenium Test Runner is having a collection of testcases called Testsuite. Click Selenium Test Runner.

## Selenium Tutorial(IDE, Core, RC)

The screenshot shows the Selenium Functional Test Runner v0.8.3 interface running in Mozilla Firefox. The window is divided into several sections:

- TEST SUITE**: A sidebar on the left containing a tree view of test suites, with "example test suite" expanded.
- CURRENT TEST**: The main area displaying a table of test steps:

TestSearch		
open	/	
type	id=qsearch__in	dotnet
submit	search	
waitForPageToLoad	2000	
verifyTextPresent	fit_framework_implementation_differences	
- Selenium TestRunner**: A panel on the right with controls for executing tests (Fast to Slow), highlighting elements, and a summary of test results (Elapsed: 00:03, 1 RUN, 1 passed, 0 failed, 0 skipped, 0 incomplete). It also includes links for View DOM and Show Log.
- PAGE UNDER TEST**: The bottom section showing search results for "dotnet". It includes a navigation sidebar with links like Home, Table of contents, Download FitNesse, Get Help, DbFit, WebTest, and Recent Changes. The main content area displays the search trace, results, and matching page names.

User will be getting the following page. The page has been classified into Test suite, Current test, control panel.

### Test suite :

The testsuite and individual test will be available in this location

### Control panel:

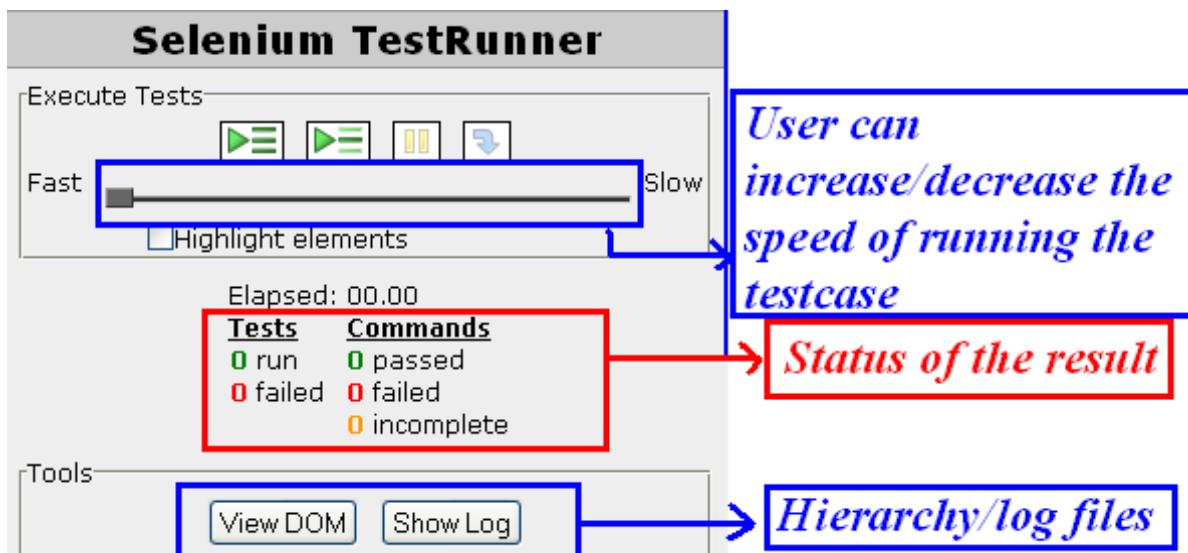
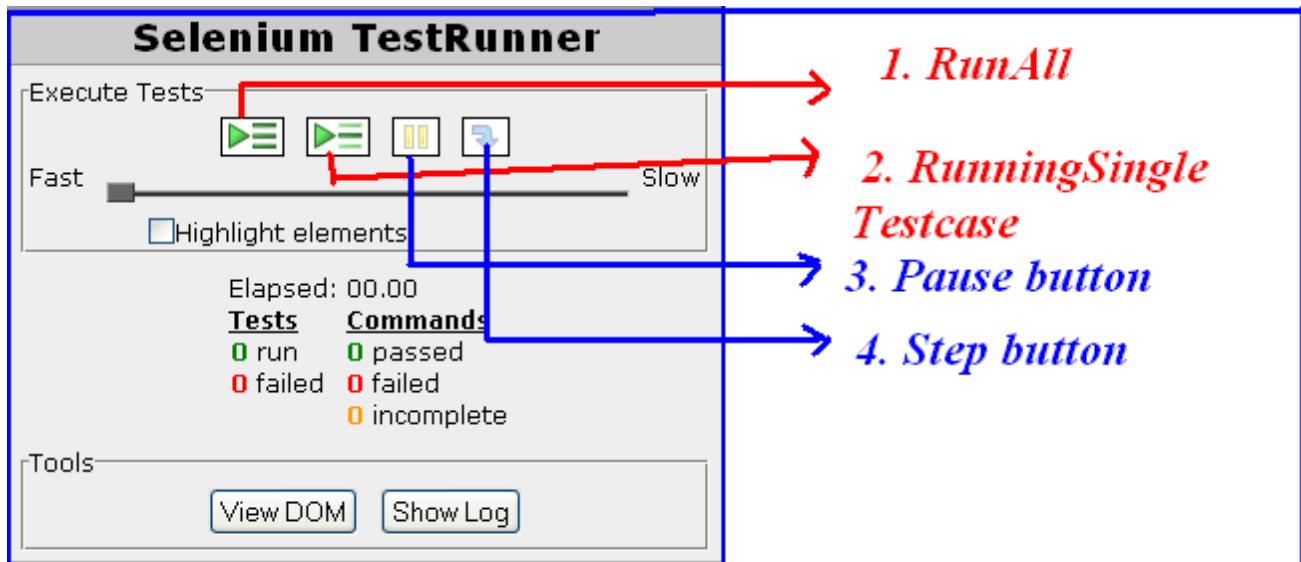
There are many options available in control panel. They are Runall , Run individual testcases, step in , pause, log file, dom viewer etc... Once the testcases has been run, test summary will be available for the user.

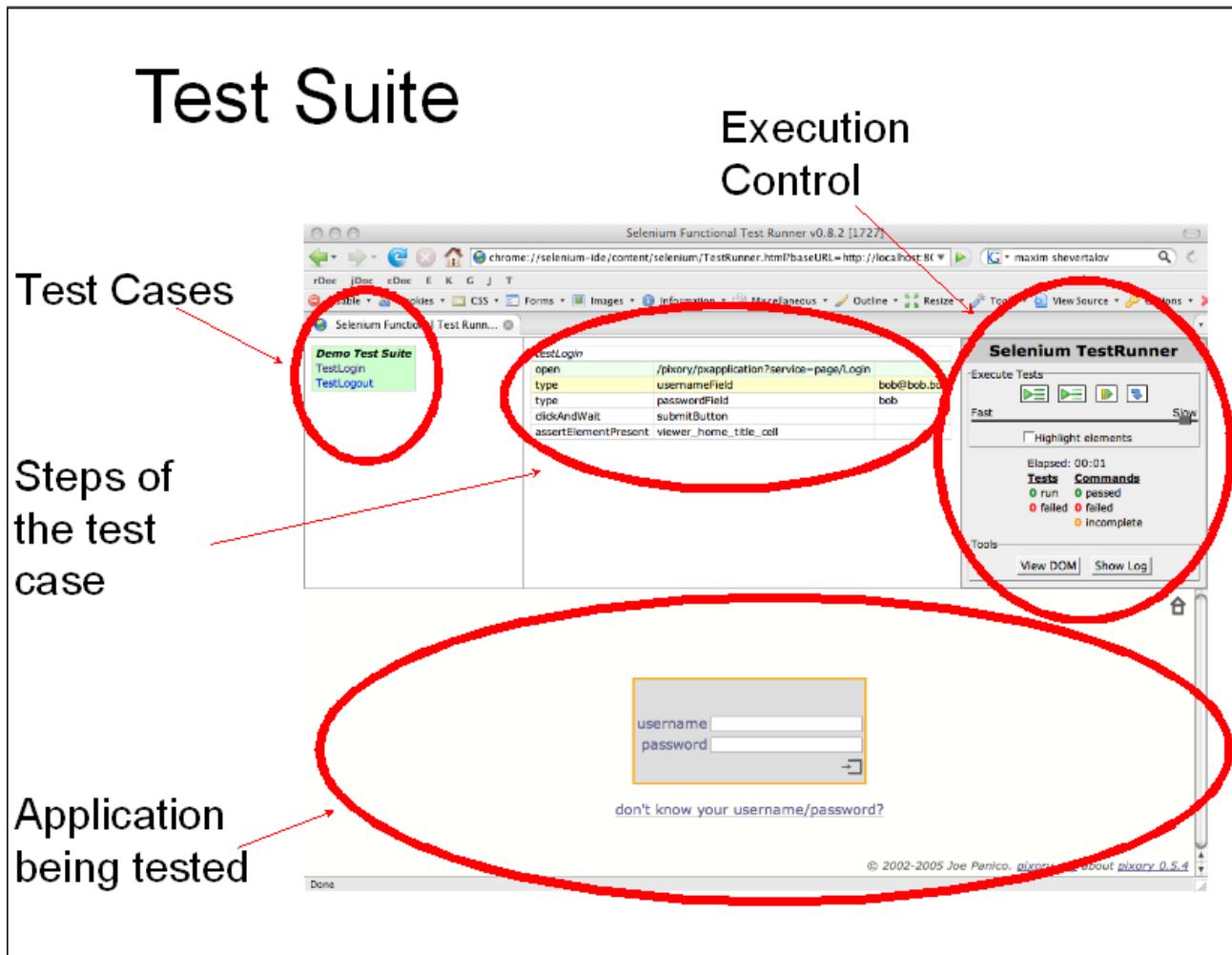
### Current Test:

In this location, user can view the steps of the test cases. When you start running, each line of the test cases will be executed and the execution line will be in yellow color , the passed and the failed test cases will automatically get the green and red color.

## Selenium Tutorial(IDE, Core, RC)

The application under test





I hope the user learnt the overview of the Selenium Core. Now let's start using the selenium Core with some practical examples.

Steps:

1. open the Internet explorer
2. type <http://localhost/>
3. choose the seleniumcore-0.8.3
4. Now user will be getting the selenium list of testing type
5. choose the first option “Selenium TestRunner”
6. user will be getting selenium functional testrunner page
7. LHS, test suite is available and the default values of it is “.. tests/TestSuite.html”, below it there is a “GO” button. Click the “GO” button
8. Now user can see, list of Testsuite. Choosing any of the testsuite , corresponding test will be available in the middle tab.
9. First time you are doing testing in selenium core , so I prefer user to choose RunAll tests button.

## Selenium Tutorial(IDE, Core, RC)

10. Now wait and watch. (yellow is for executing test cases , red is for failed testcases, green is for passed testcases)
11. After 3 minutes, the entire test suite run and user will be getting the status of the ran testsuite.

The screenshot shows the Selenium Functional Test Runner interface in Microsoft Internet Explorer. The top navigation bar includes File, Edit, View, Favorites, Tools, and Help. The address bar shows the URL: http://localhost:selenium-core-0.8.3/core/TestRunner.html?test=..%2Ftests%2FTestSuite.html&resultsUrl=..%2FpostResults. A message indicates a pop-up was blocked. The main content area displays two tables: one for 'Test Failures' and another for 'Error Checking Test Suite'. The 'Test Failures' table lists various test cases with their status (e.g., failed, passed). The 'Error Checking Test Suite' table lists errors with their descriptions. On the right, there are two 'Selenium TestRunner' panels showing execution statistics: 75 runs, 601 passed, 2 failed, and 1 incomplete. The bottom section shows a sample application under test with a purple border, containing links to 'Click here for next page' and 'Click here for next page via absolute link'.

Now the user ran the available scripts. But as a user we should know how to add our scripts in the LHS testsuite and trying to execute it.

Before adding scripts we should know certain things. First of all Selenium core requires the application under test should be in the same pc. We can't able to do other webtesting. For example: if we try to do testing, we will be getting "Permission denied". So we need the application to be in the server.

So I couldn't able to give a sample application. But I will give an overall idea of how to add your testsuite in Testsuite window.

## Selenium Tutorial(IDE, Core, RC)

Let's learn how to add our testsuite in the LHS window.

The default testsuites are available in the following location,

**“D:\ProgramFiles\ApacheSoftwareFoundation\Apache2.2\htdocs\selenium-core-0.8.3\tests”**

Open the tests folder, search for a “**TestSuite.html**” file. Open the file(TestSuite.html) either in wordpad or notepad.

End of the html coding, you can able to see the following code,

```
<tr><td><a href="../dogfood/TestFailures.html">DogfoodTestFailures</a></td></tr>
```

The meaning of the above code is, open a folder called “**dogfood**” and search for TestFailures.html When the user click “**DogfoodTestFailures**”, it will try to open the file TestFailures.html from the folder dogfood.

Now we will give our own name for the sample testsuite. For example: “**MyScript**”

Add the following line next to the “**DogfoodTestFailures**”

```
<tr><td><a href=".sample/TestFailures.html">MyScript</a></td></tr>
```

```
<tr><td><a href="../dogfood/TestRunSuccessfullTests.html">DogfoodTestRunSuccessfullTests</a></td></tr>
<tr><td><a href="../dogfood/TestRunFailedTests.html">DogfoodTestRunFailedTests</a></td></tr>
<tr><td><a href="../dogfood/TestPauseAndResume.html">DogfoodTestPauseAndResume</a></td></tr>
<tr><td><a href="../dogfood/TestBreakPoint.html">DogfoodTestBreakPoint</a></td></tr>
<tr><td><a href="../dogfood/TestFailures.html">DogfoodTestFailures</a></td></tr>
<tr><td><a href=".sample/TestFailures.html">Myscript</a></td></tr>
```

```
</tbody>
</table>
```

Now create a folder called “sample” in the following location,

**“D:\ProgramFiles\ApacheSoftwareFoundation\Apache2.2\htdocs\selenium-core-0.8.3\tests\sample”**

Right now we don't have our own application need to test. So we will take the provided sample application “dogfood”. Copy all the content (folder and files) from the **D:\ProgramFiles\ApacheSoftwareFoundation\Apache2.2\htdocs\selenium-core-0.8.3\tests\dogfood** and paste it in **sample** folder.

Now we are ready with the application need to be tested.

## **Selenium Tutorial(IDE, Core, RC)**

Step 1: Open the Internet Explorer – type <http://localhost/>

Step 2: choose “**selenium-core-0.8.3**”

Step 3: Select the “**Selenium Test Runner**”

Step 4: In LHS(Testsuite) window, you can able to see the text box having “**../tests/TestSuite.html**”

Step 5: Click the “**Go**”button

Step 6: Now you can able to see list of “**Testsuite**”, at the end “**MyScript**” the one which we created in the Testsuite.html file. Select the testsuite.

Step 7: Click the Run button in the control panel

Step 8: The testcases will be executed and the result will be displayed in the control panel.

## Selenium Tutorial(IDE, Core, RC)

**Test Suite Selection:** Step1: choose the testsuite

**Test Failures:**

Method	Message
setTimeOut	120000
open	./core/TestRunner.html?test=..%2ftests%2fTestSuite.html&resultsUrl=..%2fpostResults
waitForEval	window.htmlTestRunner.testCaseLoaded true
Run First Test Case	
click	runSeleniumTest
waitForText	testRuns 1
verifyAttribute	failure1@class regex:status_failed
verifyAttribute	failure2@class regex:status_failed
verifyAttribute	failure3@class regex:status_failed
verifyAttribute	failure4@class regex:status_failed
verifyAttribute	failure5@class regex:status_failed
verifyAttribute	failure6@class regex:status_failed

**Selenium TestRunner (Top Right):**

- Execute Tests: Run button (highlighted)
- Elapsed: 00:01
- Tests: 1 run, 6 passed, 0 failed, 0 incomplete

**Selenium TestRunner (Bottom Right):**

- Execute Tests: Run button
- Elapsed: 00:00
- Tests: 1 run, 6 passed, 1 failed, 1 incomplete

**Sample Application Under Test (AUT):**

- Click here for next page
- Click here for next page via absolute link
- link to other link

User cannot do testing on cross-site. If the user prefers to use Seleniumcore then they should have the application under test in the same machine. If you wants to do cross site testing or web application then you have to go for SeleniumIDE or Selenium Remote control.

Now we will end up the seleniumcore here. Now let's proceed with the SeleniumRemote control.

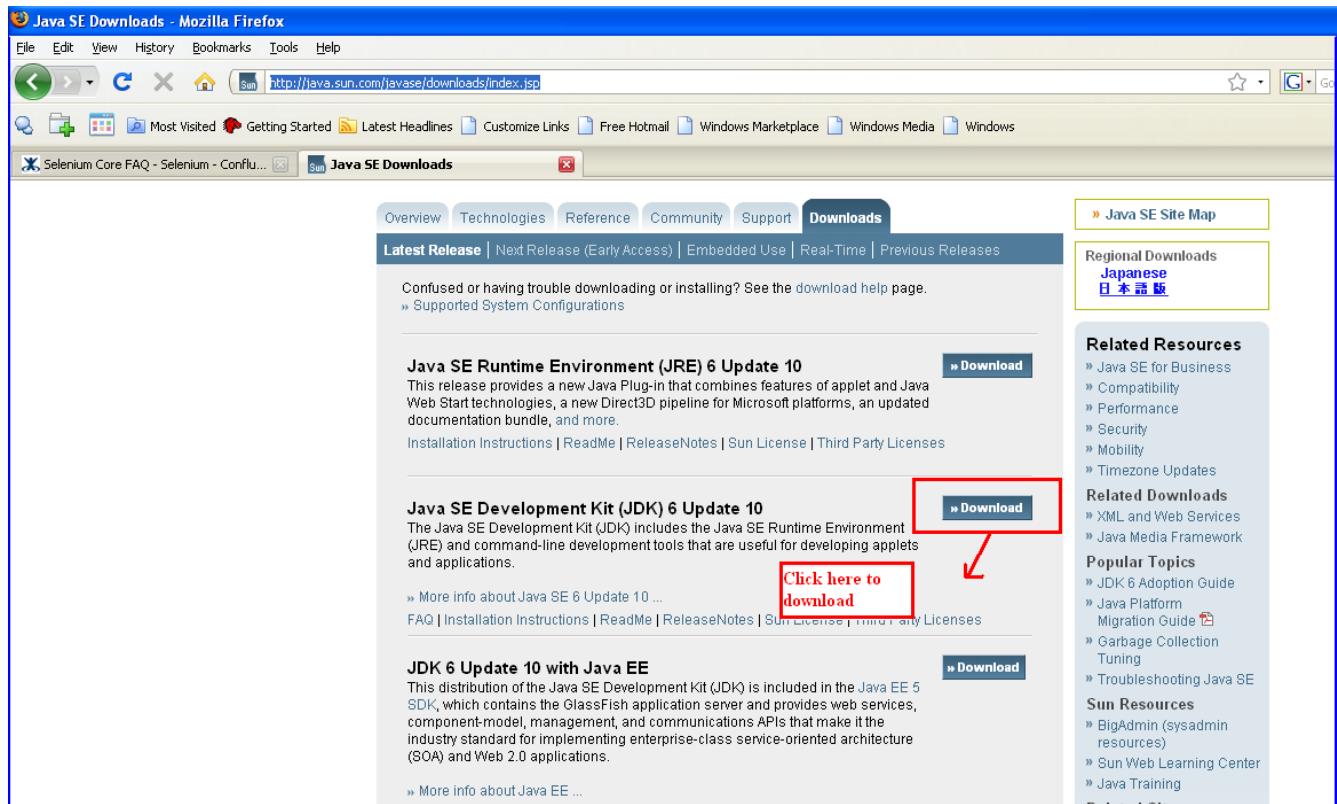
## Selenium Tutorial(IDE, Core, RC)

### Selenium Remote Control:

Installing JDK latest version:

Step 1: Go to the following link, try to download the latest version of JDK(right now 6.0 is the latest version but in my screenshot , I used the older version , so get confused with it)

<http://java.sun.com/javase/downloads/index.jsp>



## Selenium Tutorial(IDE, Core, RC)

Step 2: Accept the license agreement appearing in the next page of download.

**Note:** Before moving to this page, user has to choose the platform and language option.

**Platform:** Windows

**Language:** Multilanguage

J2SE(TM) Development Kit 5.0 Update 4  
For easier, more reliable downloads, try Sun Download Manager.

- Solaris 64-bit requires users to first install 32-bit.
- Information on [picking the right format to download](#)
- Installation instructions:
  - English
  - Japanese
- For Windows, choose "Windows Online Installation" for the quickest download and installation on a machine connected to the Internet. The default download size is **40.9MB**, which is the maximum download. The size may decrease if features are deselected, to a minimum of **18.2MB**.

NOTE: This page offers files for different platforms - please be sure to download the proper file(s) for your platform.  
We highly recommend using Sun Download Manager (SDM), as it lets you pause, resume, and restart your download while ensuring a successful download experience. Just select the files you want to download, then click the "Download Selected with Sun Download Manager" button to automatically install and start SDM. Alternately, click directly on the links in the file list to download through your browser.  
For any download problems or questions, please see the [Download Center FAQ](#).  
How long will the download take?

**Accept the License Agreement**

**Required:** You must accept the license agreement to download this software.  
 [Accept License Agreement](#) | [Review License Agreement](#)  
 [Decline License Agreement](#)

**Click the offline checkbox**

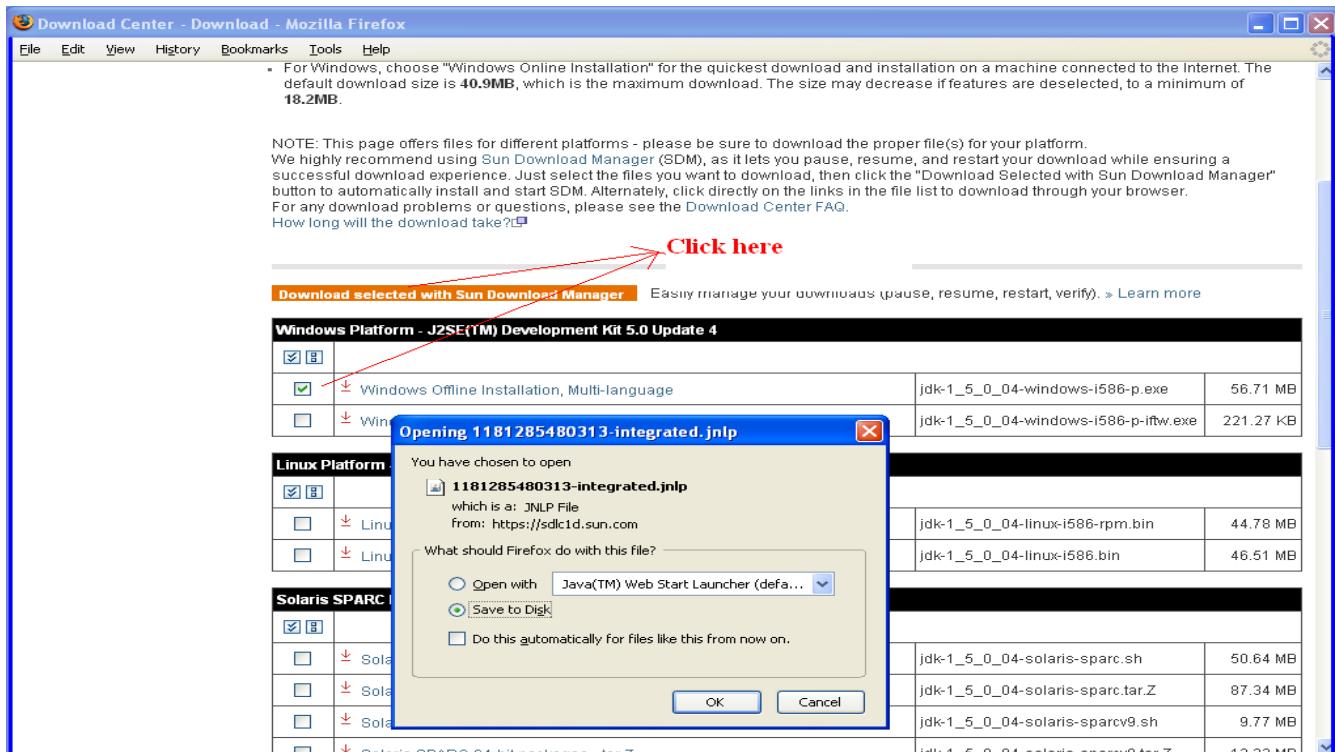
**Download selected with Sun Download Manager** Easily manage your downloads (pause, resume, restart, verify). > Learn more

Windows Platform - J2SE(TM) Development Kit 5.0 Update 4			
<input checked="" type="checkbox"/>		Windows Offline Installation, Multi-language	jdk-1_5_0_04-windows-i586-p.exe 56.71 MB
<input type="checkbox"/>		Windows Online Installation, Multi-language	jdk-1_5_0_04-windows-i586-p-iftw.exe 221.27 KB

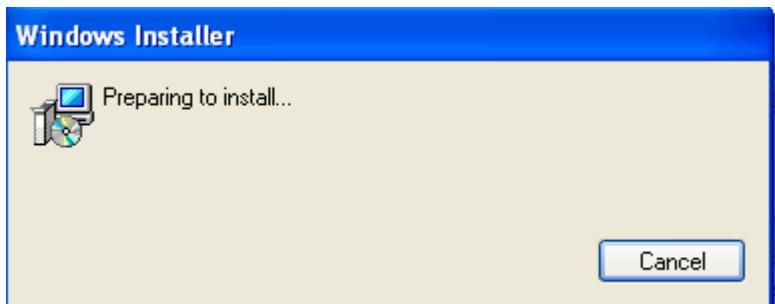
Waiting for sdc1d.sun.com...

Step 3: Choose the offline installation , save dialog box will open. Save the executable file in the desired location.

## Selenium Tutorial(IDE, Core, RC)

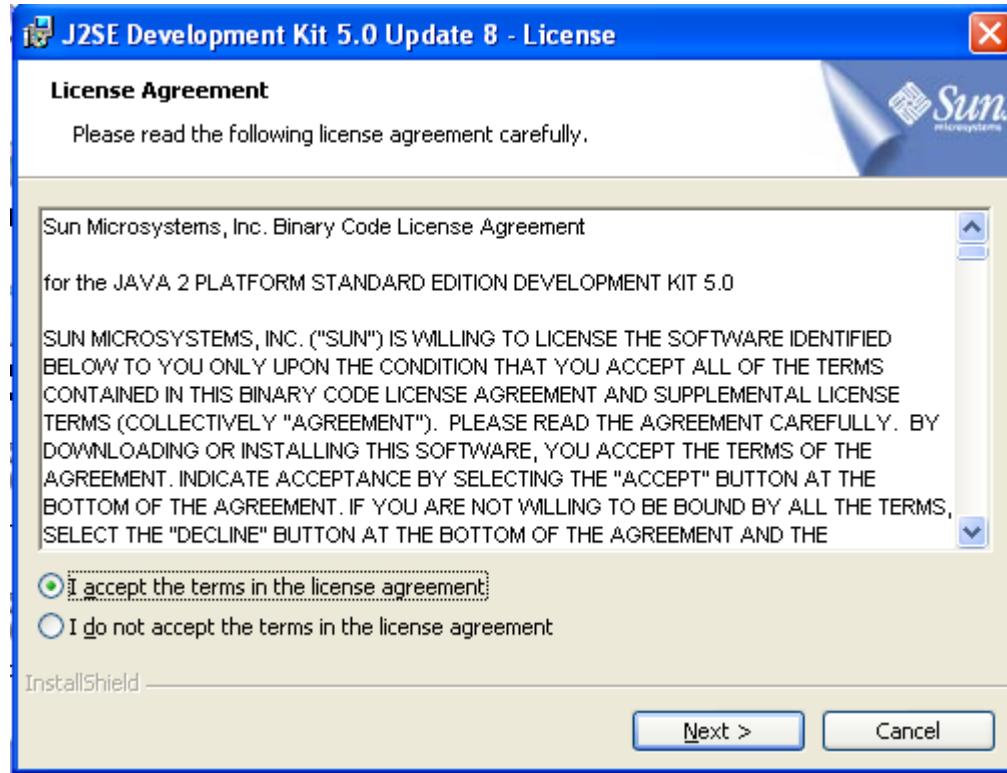


**Step 1:** Run the installer. Don't click cancel

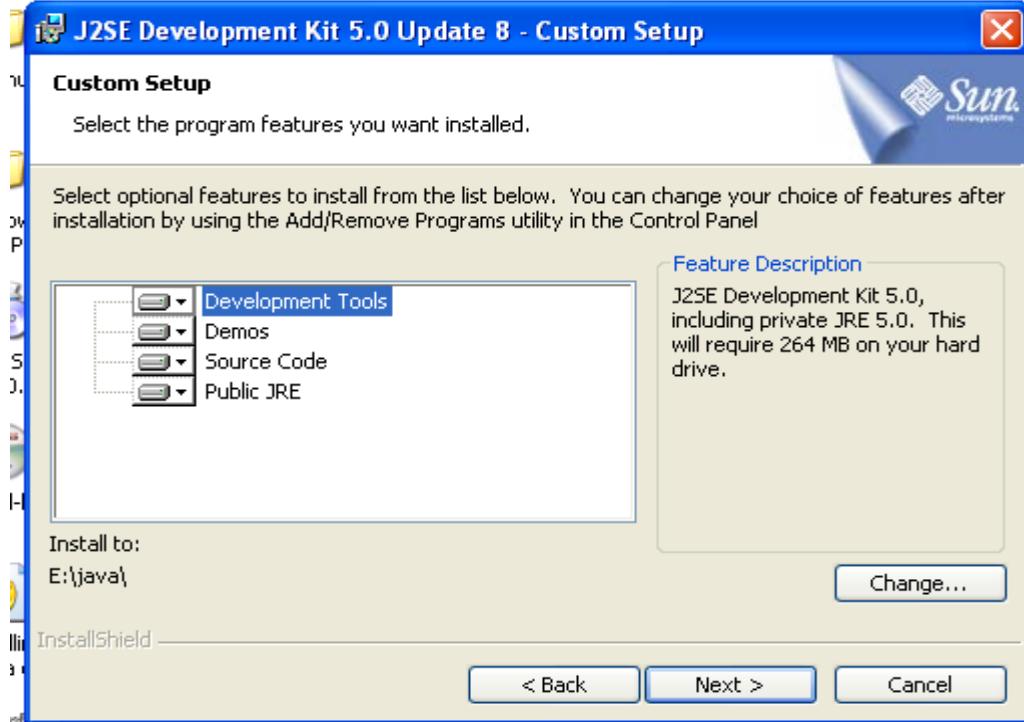


## Selenium Tutorial(IDE, Core, RC)

**Step 2:** Accept the License Agreement and click the next button

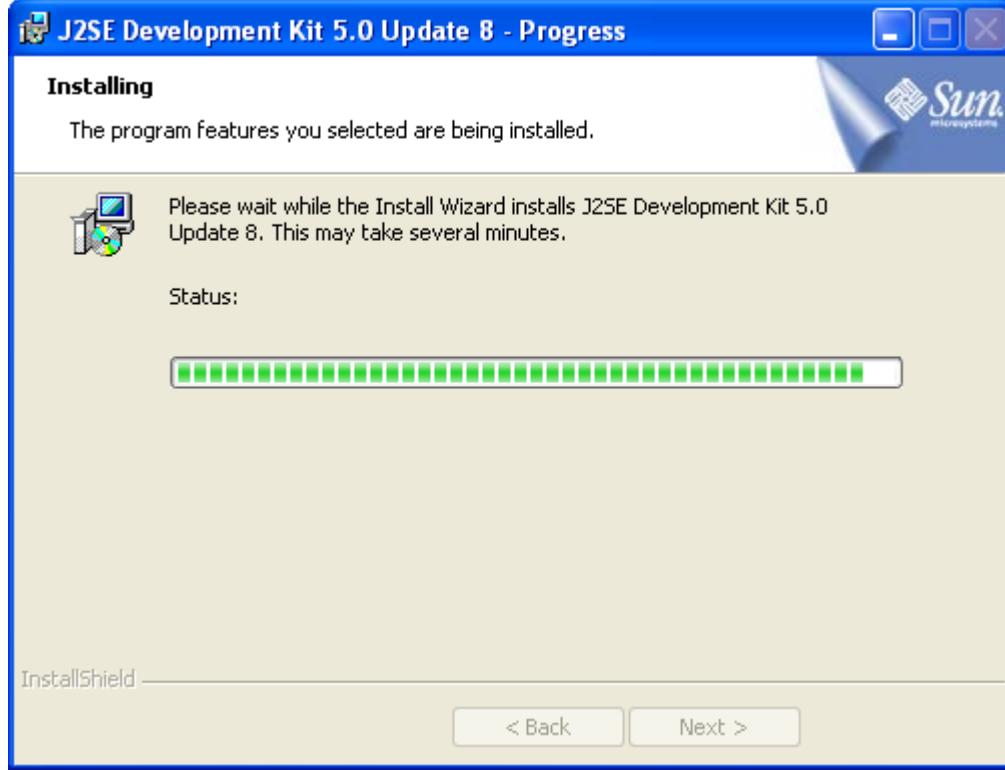


**Step 3:** Choose a location for installation. click the next button.

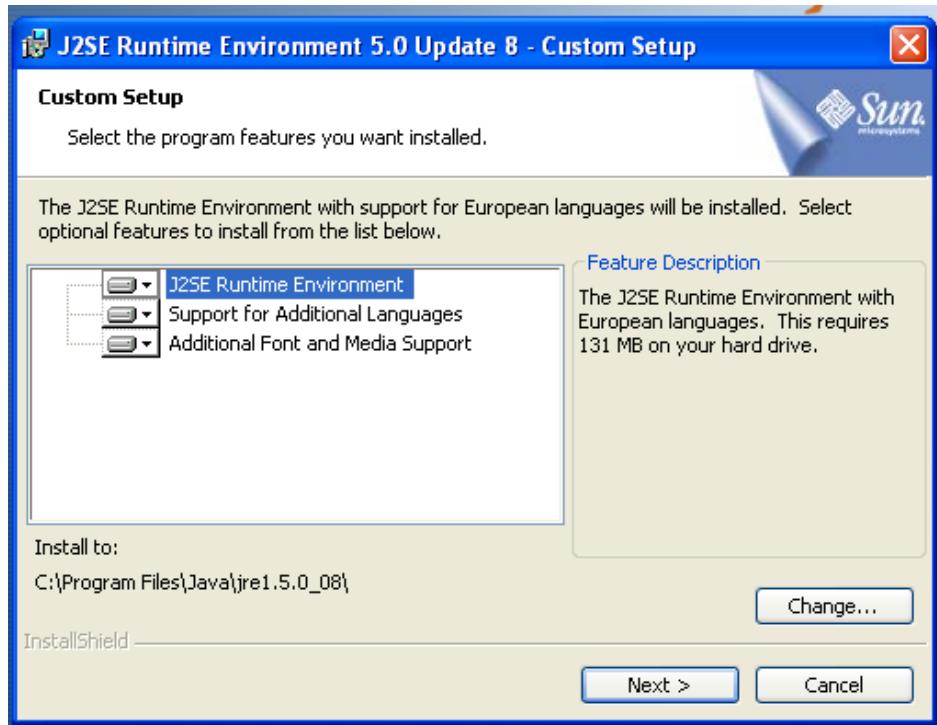


## Selenium Tutorial(IDE, Core, RC)

### Step 4: Installation are being in Progress

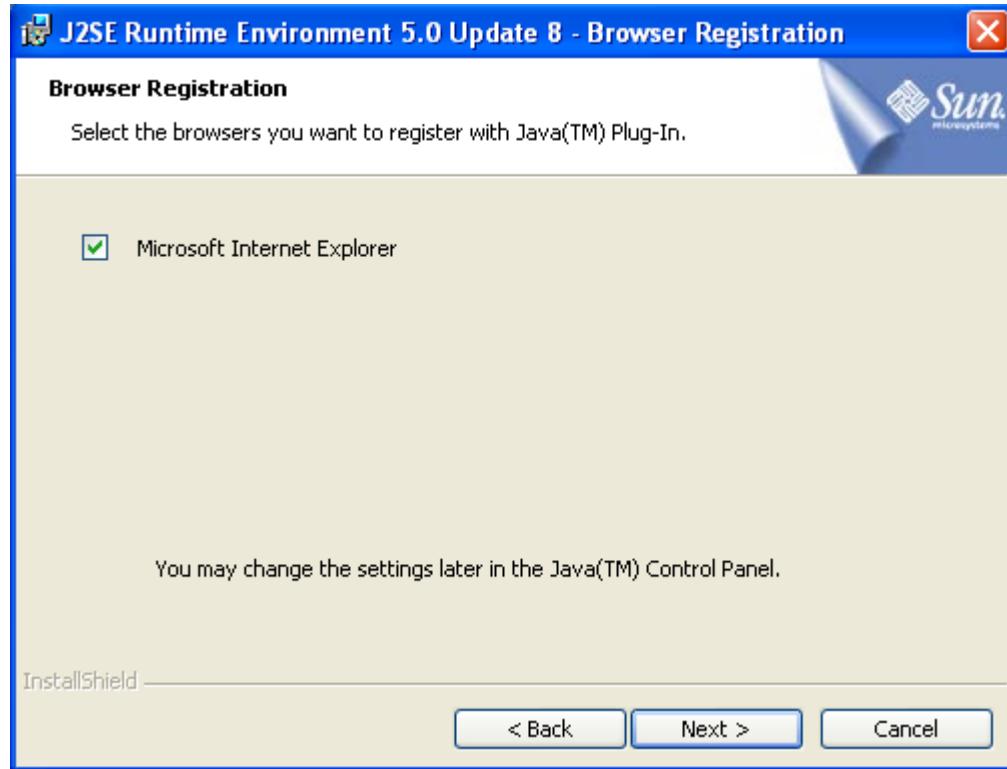


### Step 5: Run Time Environment – Custom Setup. Click the next button to continue.

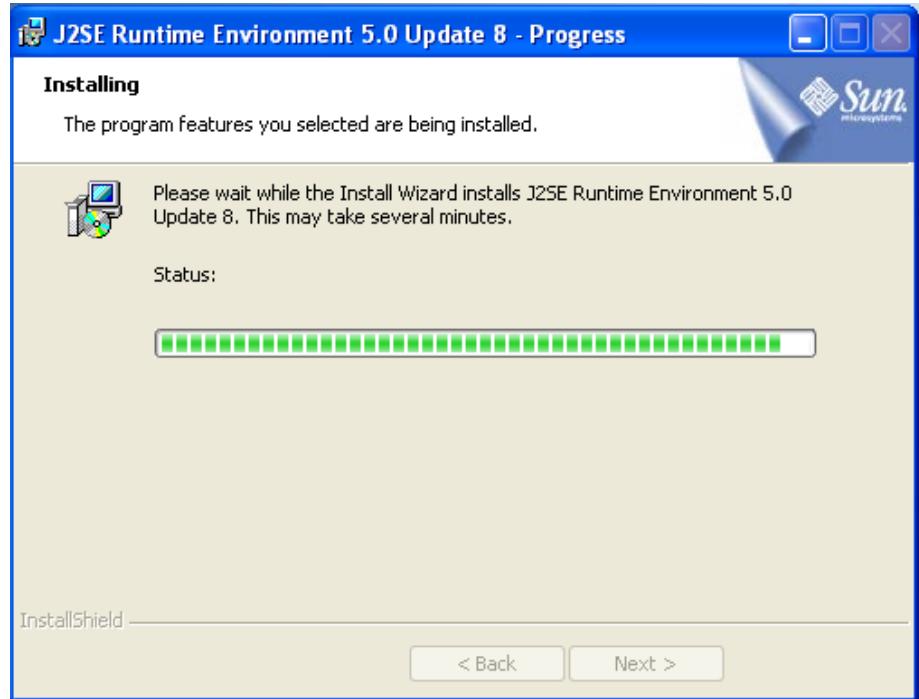


## Selenium Tutorial(IDE, Core, RC)

Step 6: Browser Registration. Click the next button



Step 7 : Installation is in Progress



## Selenium Tutorial(IDE, Core, RC)

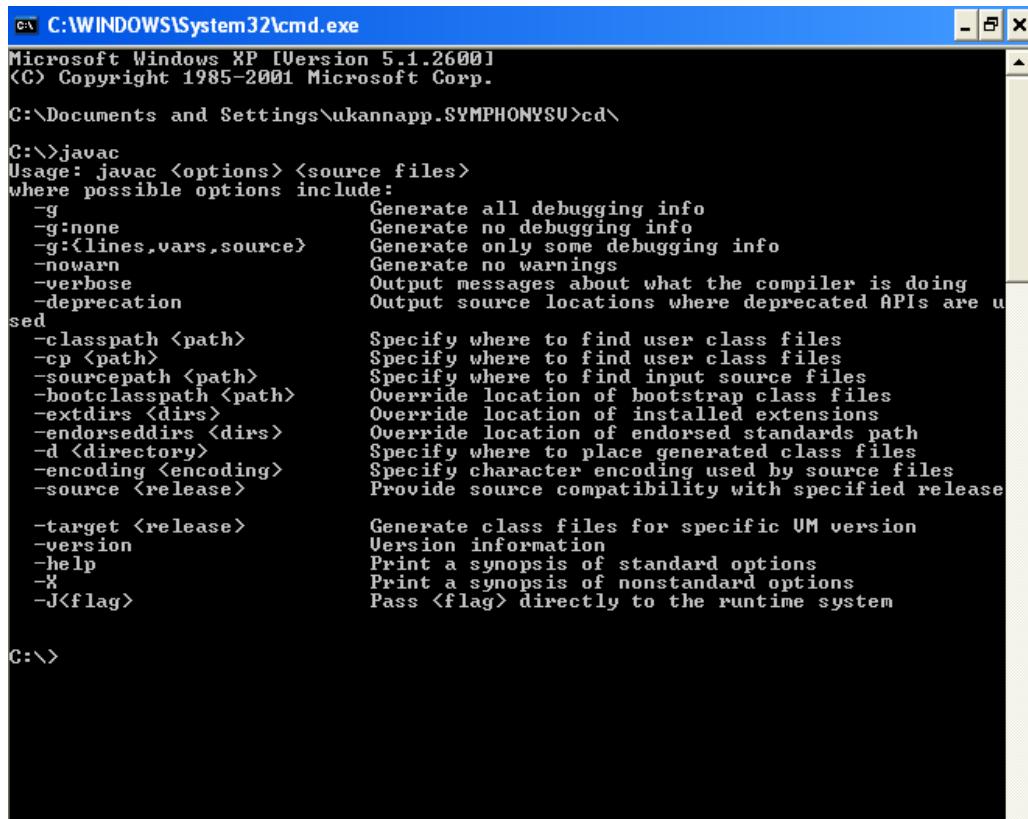
[Step 8](#): Successfully Installed



## Selenium Tutorial(IDE, Core, RC)

### Checking the java installation:

Go to command prompt and type javac. If it successfully installed, you will be getting the information about the java.



The screenshot shows a Windows XP Command Prompt window titled 'C:\WINDOWS\System32\cmd.exe'. The window displays the Java compiler's usage information. The text is as follows:

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

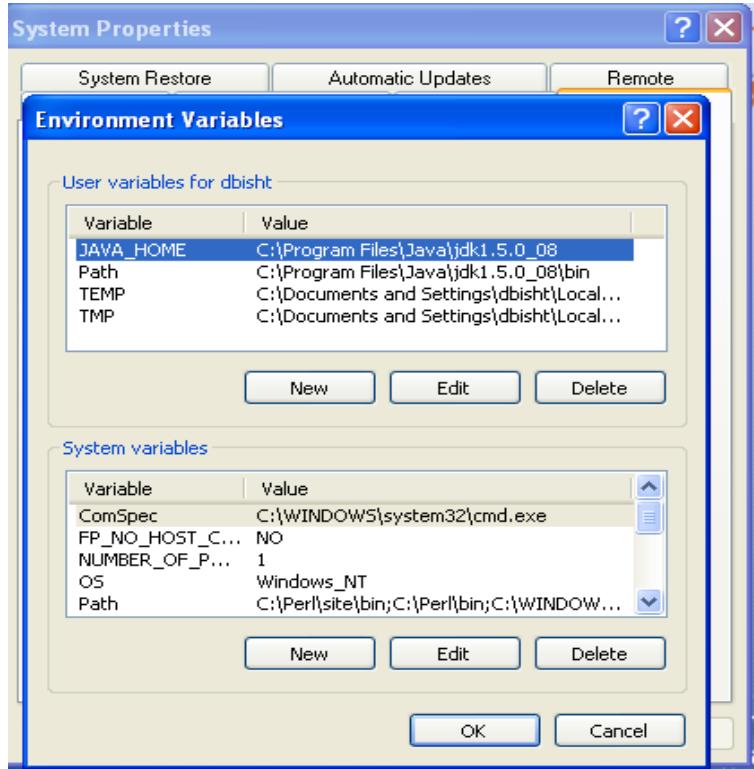
C:\Documents and Settings\ukannapp.SYMPHONYSU>cd\
C:>javac
Usage: javac <options> <source files>
where possible options include:
  -g                         Generate all debugging info
  -g:none                     Generate no debugging info
  -g:{lines,vars,source}       Generate only some debugging info
  -nowarn                     Generate no warnings
  -verbose                    Output messages about what the compiler is doing
  -deprecation               Output source locations where deprecated APIs are u
sed
  -classpath <path>          Specify where to find user class files
  -cp <path>                  Specify where to find user class files
  -sourcepath <path>          Specify where to find input source files
  -bootclasspath <path>       Override location of bootstrap class files
  -extdirs <dirs>            Override location of installed extensions
  -endorseddirs <dirs>        Override location of endorsed standards path
  -d <directory>             Specify where to place generated class files
  -encoding <encoding>        Specify character encoding used by source files
  -source <release>          Provide source compatibility with specified release

  -target <release>          Generate class files for specific VM version
  -version                   Version information
  -help                      Print a synopsis of standard options
  -X                        Print a synopsis of nonstandard options
  -J<flag>                  Pass <flag> directly to the runtime system

C:>
```

## Selenium Tutorial(IDE, Core, RC)

**Step 9:** See the java home and path in the system properties



If the Java class path is not set properly, then you need to set manually. Go to command prompt, type the following,

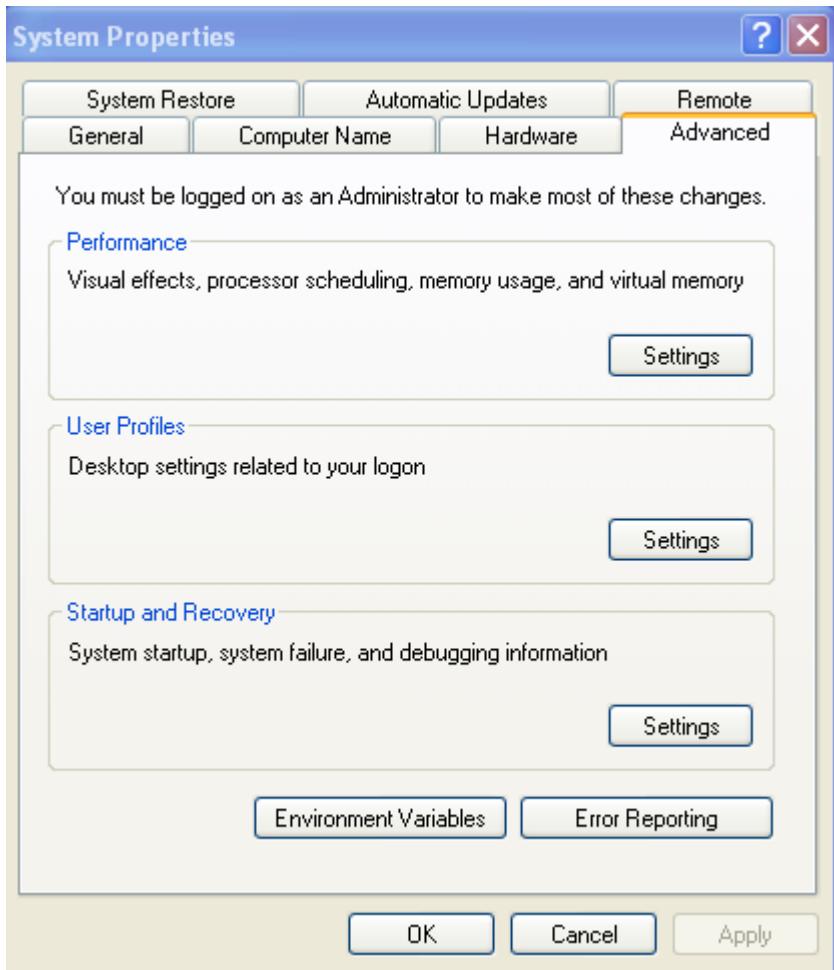
```
set JAVA_HOME=C:\Program Files\Java\jdk1.5.0_08  
set PATH=%PATH%;%JAVA_HOME%\bin
```

**or else**

**Right click on My Computer, click system properties,**

**5. go to Advanced**

## Selenium Tutorial(IDE, Core, RC)



### 6. Environment Variables

#### 7. Click new and set the path.

```
set JAVA_HOME=C:\jdk1.5.0_08  
set PATH=%PATH%;%JAVA_HOME%\bin
```

## Selenium Tutorial(IDE, Core, RC)

Download the selenium Remote control from the following link.

<http://selenium-rc.openqa.org/download.jsp>

The screenshot shows a Mozilla Firefox browser window with the title bar "Selenium RC: Download - Mozilla Firefox". The address bar contains the URL "http://selenium-rc.openqa.org/download.html". The main content area displays the "Selenium-RC: Download" page from the OpenQA website. On the left, there is a sidebar with navigation links for "Essential" (About, News, Changelog, Tutorial, FAQ), "Evaluating" (License, Download, Release Notes), and "Using It" (Experimental, Java, .NET, Perl, PHP, Python, ruby, Selenese, JavaScript, Server, Command Line Options, Developers Guide, Wiki, User Forums, Help with XPath, Self Help, Reporting). The main content area lists several versions of Selenium RC:

- Version 1.0 beta 1**
  - Date: 03/05/2008
  - [Changelog](#)
  - [Downloads](#)
    - [Major release](#)
- Version latest nightly**
  - Date: Every night - the actual date is in the name of the redirected file..
  - [latest nightly build - potentially unstable](#)
- Version 0.9.2**
  - Date: 09/20/2007
  - [Changelog](#)
  - [Downloads](#)
    - [Major release](#)
- Version 0.9.0**
  - Date: 11/14/2006
  - [Changelog](#)
  - [Downloads](#)
    - [Major release](#)
- Version 0.8.1**
  - Date: 06/05/2006
  - [Changelog](#)
  - [Downloads](#)
    - [Major release](#)
- Version 0.8.0**
  - Date: 06/02/2006

A red box highlights the "Major release" link under the Version 1.0 beta 1 section, and a red arrow points from this link to a red box containing the text "Click here to download the latest version of Selenium RC".

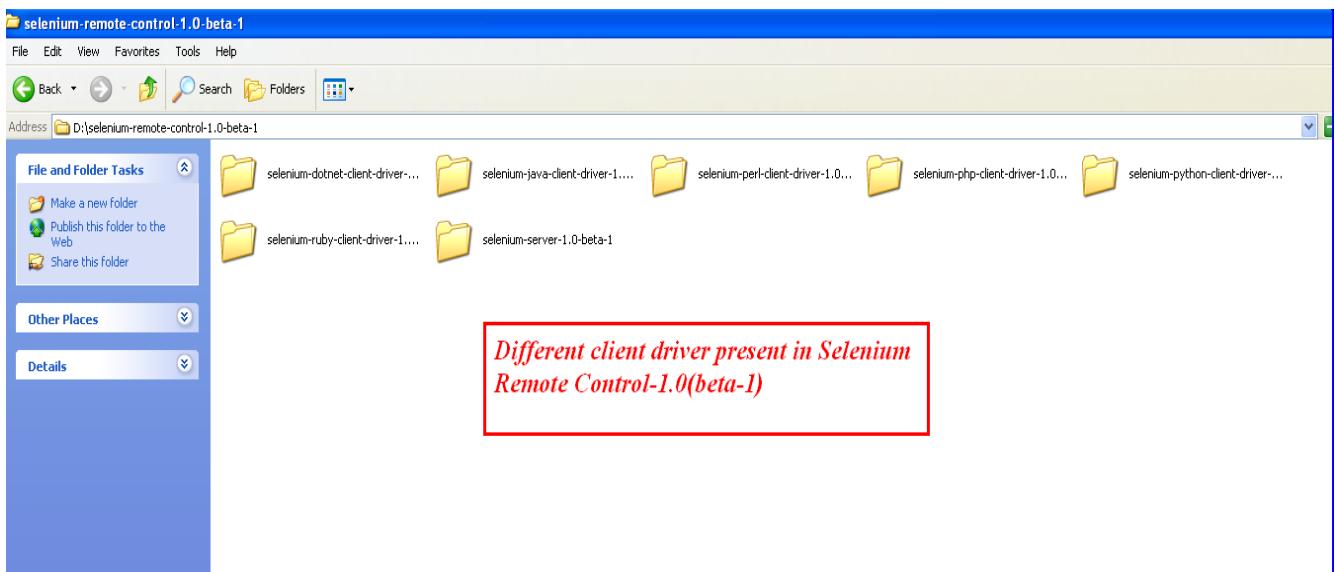
## **Selenium Tutorial(IDE, Core, RC)**

## **Selenium Tutorial(IDE, Core, RC)**

## Selenium Tutorial(IDE, Core, RC)



Extract the Selenium remote and paste in any directory(For convenience, I placed in D: drive). Now Open the selenium Remote control folder, you will be having the different client drivers.



## Selenium Tutorial(IDE, Core, RC)

Now that you have installed the Selenium files, you need to let windows know where to look if you run a selenium command. You do this by adding the Selenium directory to your environment path:

To add Selenium to your Windows environment take the following steps:

3. Click on **Start**
4. *Right Click on My Computer*
5. Click on **Properties**
6. Click on the **Advanced** tab.
7. Click on **Environment Variables**
8. On the upper half there should be a Variable named **PATH**
  1. If you located it, double click on it, and add this value at the end of what is already in “**Variable value**”:

`;C:\selenium-remote-control-0.9.2\selenium-server-0.9.2\selenium-server.jar`

**Make sure you replace -0.9.2 with the version you downloaded**

2. If you do not located it, do this:

1. Click on **New**
2. For **Variable name**: enter: PATH
3. For **Variable value**: enter:

`C:\selenium-remote-control-0.9.2\selenium-server-0.9.2\selenium-server.jar`

*(Make sure you replace -0.9.2 with the version you downloaded)*

4. Click on **OK**
9. Click on **OK**
10. Click on **OK**

Now you have finished setting up the environment, the next step is to run a few selenium commands.

## Selenium Tutorial(IDE, Core, RC)

Ok now we are going to run some basic selenium commands from the command line.

To start off lets first open the command line window:

Click on **Start**

Click on **Run...**

Type in **cmd**

Now that the command line window is open lets try some basic Selenium commands as mentioned on the Selenium tutorial page

Once you've got Java installed and ready to go, you can start the Selenium Server from the command line like this:

```
java -jar selenium-server.jar -interactive
```

That will start the Selenium Server and allow you to type commands in the command window. After a number of log messages, you should see the following message:

```
Entering interactive mode... type Selenium commands here  
(e.g: cmd=open&1=http://www.yahoo.com)
```

Let's begin by using Selenium Server to open up a browser. If you're on Windows and want to run your Selenium commands using Internet Explorer, try typing this:

```
cmd=getNewBrowserSession&1=*iexplore&2=http://www.google.com
```

If you want to use Firefox, try this:

```
cmd=getNewBrowserSession&1=*firefox&2=http://www.google.com
```

You're running the "getNewBrowserSession" command, using the browser of your choice (\*iexplore for Internet Explorer, \*firefox for Firefox, or \*opera for Opera), starting at www.google.com. Typing commands in this window automatically fires off HTTP web requests to the Selenium Server, requesting work to be done. (In non-interactive mode, you can use any automated tool you like to send these HTTP requests to the server, instead of typing them by hand.) When you press Enter, you'll see a message describing the request you just made:

```
--> Requesting http://localhost:4444/selenium-server/driver?  
cmd=getNewBrowserSession&1=*firefox&2=http://www.google.com
```

If all goes well, you should see a new browser window starting with the browser of your choice. Back in the Selenium Server command window, you should see the following message:

```
Got result: OK,260113 on session 260113
```

(If this doesn't happen, you may need to take a look at our fine [Troubleshooting](#) guide.)

## Selenium Tutorial(IDE, Core, RC)

The first part of this message "Got result: OK" says that your request for work was successful. The second part of this message, the number, is a Session ID. The Session ID will be different every time you run the "getNewBrowserSession" command from the command line.

Let's try some more commands; let's do a Google search! We'll start by opening up the Google website. Type this in the Selenium Server command window, replacing the Session ID below with the number you got when you ran "getNewBrowserSession":

```
cmd=open&1=http://www.google.com/webhp&sessionId=260113
```

When this command finishes, your browser window should reveal google.com in the lower frame. (The /webhp makes sure we stay on www.google.com; otherwise, we might get redirected to a country-specific Google server, e.g., [www.google.fr](http://www.google.fr).) Don't forget that you need to replace the Session ID above with your own personal Session ID, the one you got when you ran the "getNewBrowserSession" command.

Now that Google is open, let's try typing something in the search box. Type this in the Selenium Server command window:

```
cmd=type&1=q&2=hello world&sessionId=260113
```

Again, don't forget to replace the Session ID with your own Session ID. If all goes to plan, you should see "hello world" in the search box for your Google search.

Now, let's do a search!

```
cmd=click&1=btnG&sessionId=260113
```

You should now see the results of your Google Search in your browser window. You can run dozens of Selenium commands to automate all manner of browsing tasks. For more information on particular commands, you can go look at the [Selenium Core](#) section at the Open QA website, or check out the reference materials available for any of our Client Drivers ([Java](#), [.NET](#), [Perl](#), [PHP](#), [Python](#) or [Ruby](#)).

The next thing we might want to do is read some information back out of the page... for example, let's retrieve the HTML title of the current page, like this:

```
cmd=getTitle&sessionId=260113
```

The browser will return the title of the HTML page we've loaded, like this:

```
Got result: OK, hello world - Google Search on session 260113
```

That's enough Interactive Mode for now; let's move on to writing some code! But before we go, let's stop the browser we started. Type this command in the Selenium Server command window:

```
cmd=testComplete&sessionId=260113
```

When this command finishes, your browser window should automatically close. To quit the Selenium Server, type "quit" or just press Ctrl-C.

## Selenium Tutorial(IDE, Core, RC)

When you look at the selenium Remote control page, they mentioned that you can do programming in any of the following language (php,perl, java, C#, python,Ruby). If you go to the following site, they gave scripts in different programming language.

<http://selenium-rc.openqa.org/tutorial.html>

Instead of knowing everything, let's understand with one programming language. I am assume that everyone would have learnt java in their college life and feel little comfortable in this language. So now we will take how to write our scripts in java and need to execute it.

Even if you don't know java , don't worry I will tell you how to get the scripts and need to execute it in available tool.

### Selenium Java programming with Eclipse:

#### The first step is to get Eclipse.

*Eclipse is a software platform comprising extensible application frameworks, tools and a runtime library for software development and management. It is written primarily in Java to provide software developers and administrators an integrated development environment (IDE).*

In other words Eclipse is a software that allows you to write and execute programs in many different languages. In our case we will be using Eclipse to write and execute Selenium via Java.

Best of all Eclipse is a free software! (~Thank you Open Source)

Let us begin...

#### Getting Eclipse:

Go to <http://www.eclipse.org/downloads/>

3. You will want to get **Eclipse Classic 3.4**
4. Click on “Windows” link to the right.



#### Eclipse Classic 3.4 (151 MB)

The classic Eclipse download: the Eclipse Platform, Java Development Tools, and Plug-in Development Environment, including source and both user and programmer documentation. Note: For other downloads see the [Eclipse Project download page](#).

[More...](#)

[Windows](#)  
[Mac OS X](#)  
[Linux 32bit](#)  
[Linux 64bit](#)

[Release notes](#) | [Other downloads](#) | [Documentation](#)

5. Now to download it, Click on the link that appears to the right of “Download from:”

## Selenium Tutorial(IDE, Core, RC)



**Download from:** [United States] Rochester Institute of Technology, Department of Computer Science ([http](http://))  
...or pick a mirror site below.

6. Save the file.

After it has finished downloading, you need to extract the files to your D: driver directory.

In windows you would take the following steps:

3. Right Click on the zip file you just downloaded
4. Click on **Extract All...**
5. Click **Next**
6. Click on **Browser...**
7. Click on **My Computer**
8. Click on **Local Disk (D:)**
9. Click on **OK**
10. Click on **Next**
11. Click on **Finish**

At this point Eclipse is installed and ready to be used

Note: Keep in mind Eclipse is an “Executable” program; there is no installation need other than extracting the files on to your hard drive.

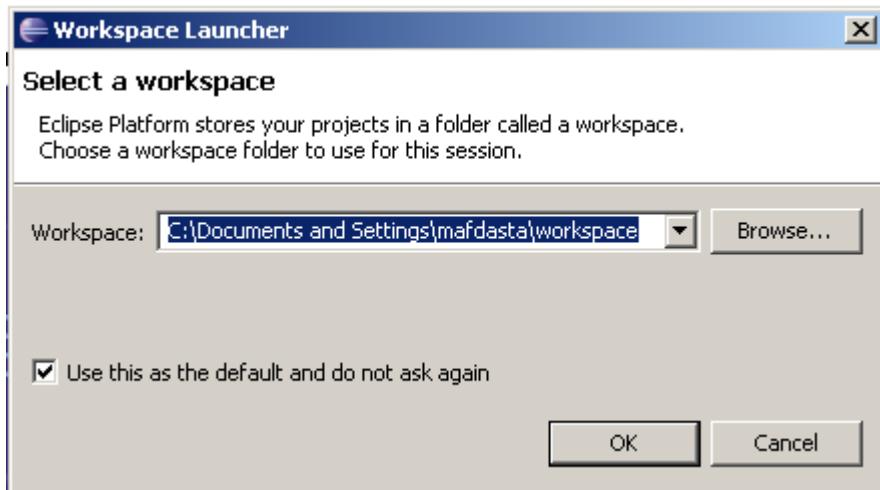
Now let us start up Eclipse...

Go to your C: driver and open up your **Eclipse** folder.

*Double Click on Eclipse* to run the program.

A Settings Window will open up letting you know that Eclipse saves your projects into a folder called “**workspace**.”

## Selenium Tutorial(IDE, Core, RC)

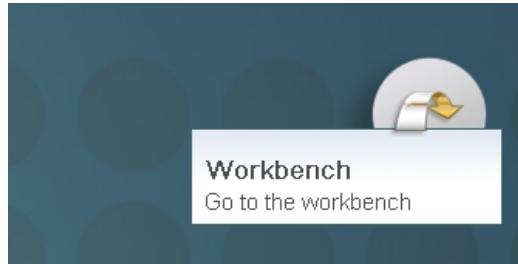


If you wish to change the location of this folder from it's default location, *Click on Browser...* and choose your desired location.

If you do not wish to be informed about this again, *Check the Box* next to "Use this as the default and do not ask me again."

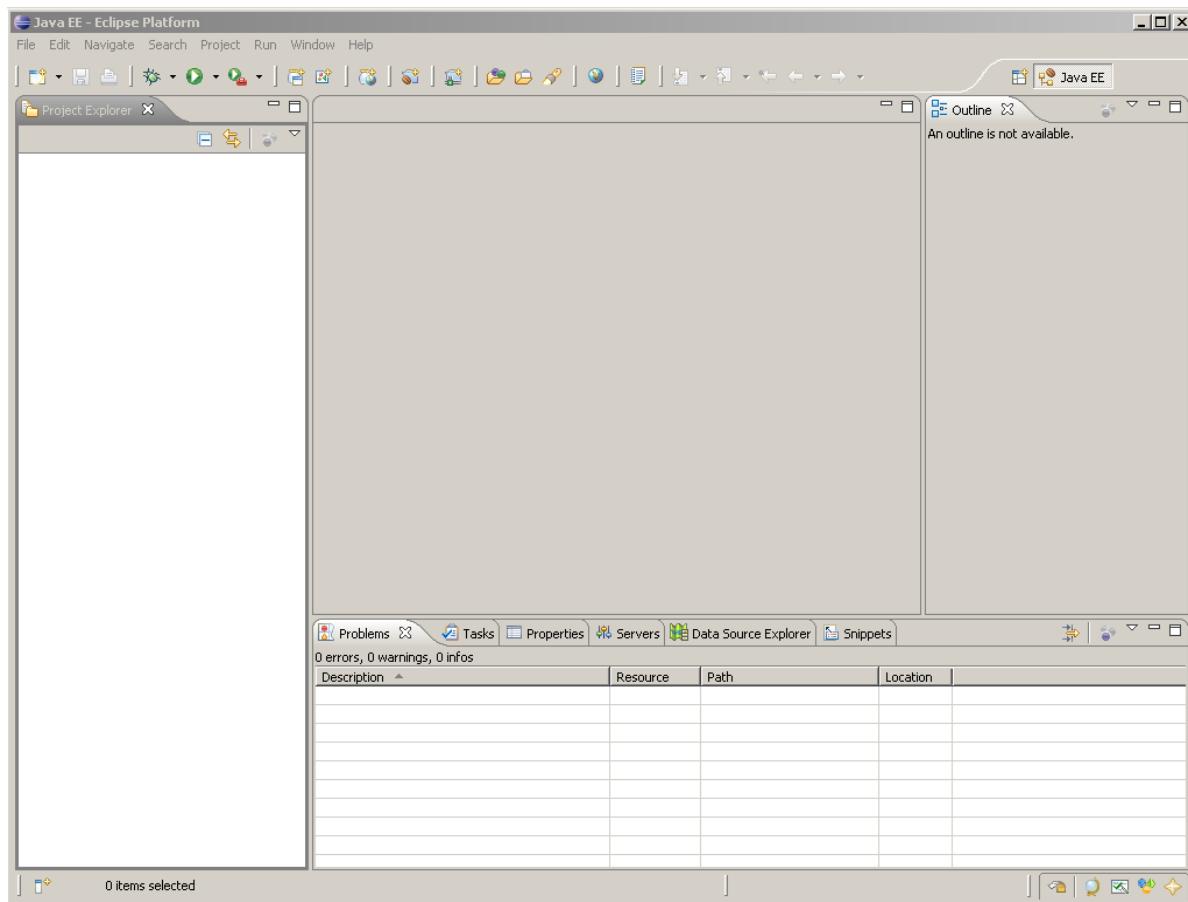
*Click on OK* to continue.

Once Eclipse has loaded you will have the option to get an overview of the application, find out what is knew, review come sample codes, go through some tutorials, or go to your workbench. In our case, we want to go to the workbench, so click on the icon with the title "Workbench."



This will bring you to the environment that you will be working in:

## Selenium Tutorial(IDE, Core, RC)



## Selenium Tutorial(IDE, Core, RC)

Our first step will be to create a **Project\***.

Go to **File**

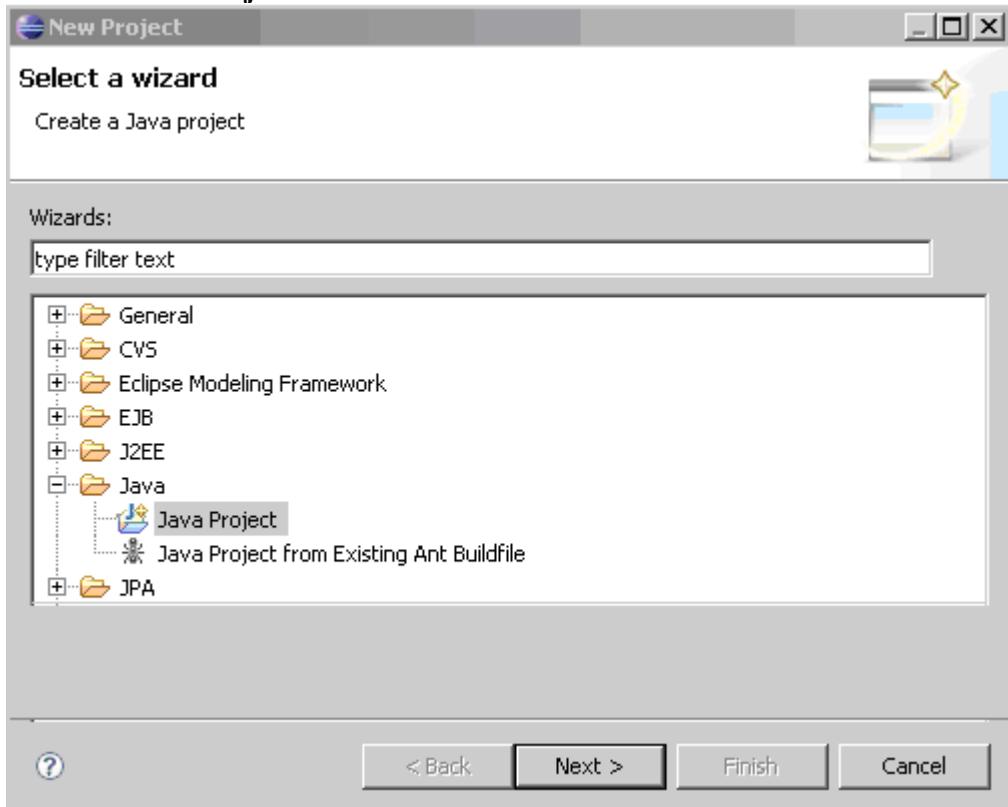
Click on **New**

Click on **Project...**

At this point a new window has opened, here you will tell Eclipse what type of Project you want to create.

Click on **Java**

Click on **Java Project**

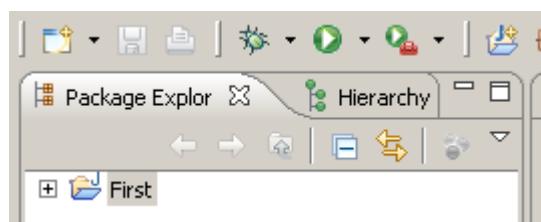


Click **Next**

On the top next to **Project name:** give your project the name “First”

Click **Finish**

Now you should be back at your work bench with your project’s folder visible to the left.

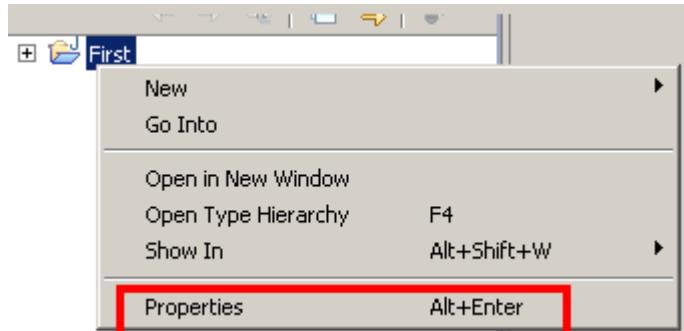


## Selenium Tutorial(IDE, Core, RC)

\* A **Project** is basically what will save property setting of your environment and create a folder that will contain all the files you will be creating/using to accomplish your programming tasks.

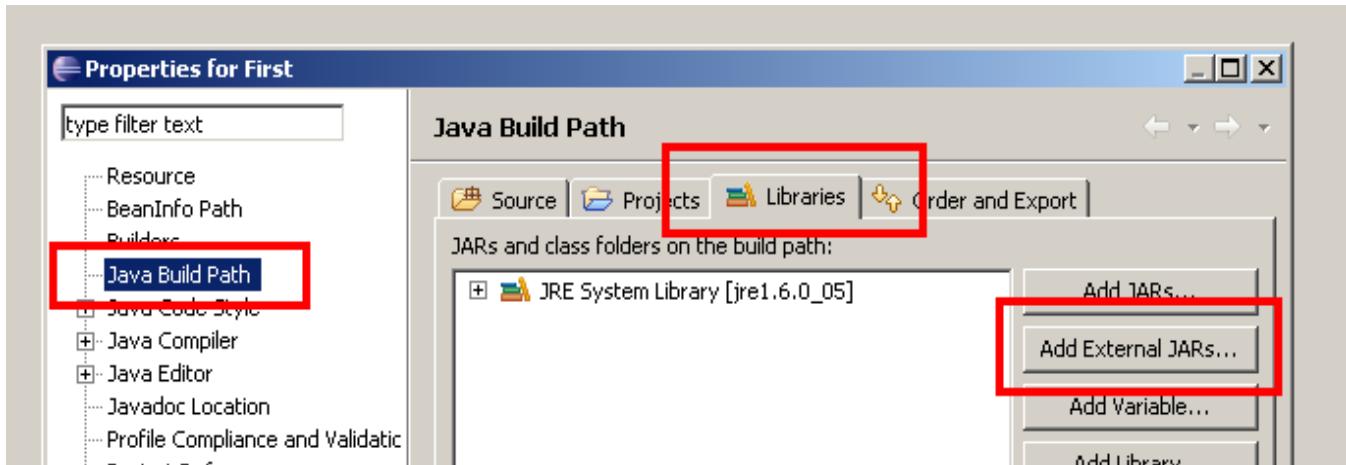
The next step in the process is to import two Selenium Libraries, without them, you would not be able to run Selenium.

5. Right Click on the “First” folder
6. Click on **Properties**



(Please note I crop the screenshots sometimes to a more appropriate size)

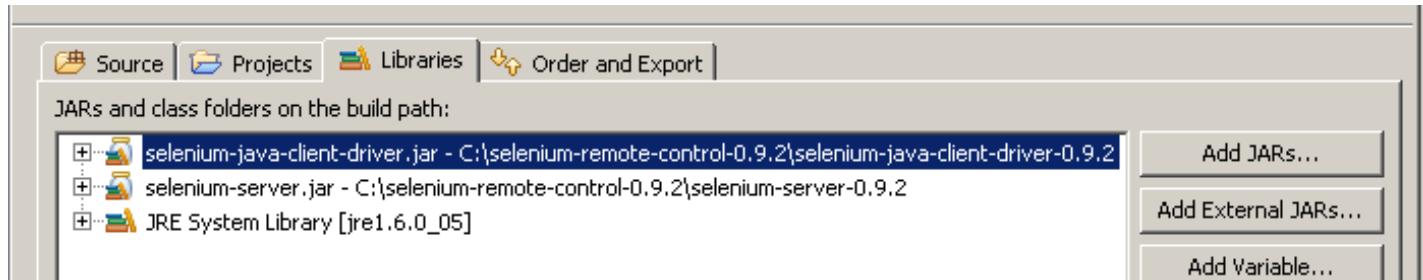
7. Click on **Java Build Path**
8. Click on the **Libraries** tab
9. Click on **Add External JARs...**



10. Find and select selenium-server.jar then Click **Open**  
(i.e. C:\selenium-remote-control-0.9.2\selenium-server-0.9.2\selenium-server.jar)
11. Click on **Add External JARs...** again
12. Find and select selenium-java-client-driver.jar then Click **Open**  
(i.e. C:\selenium-remote-control-0.9.2\selenium-java-client-driver-0.9.2\selenium-java-client-driver.jar)

## Selenium Tutorial(IDE, Core, RC)

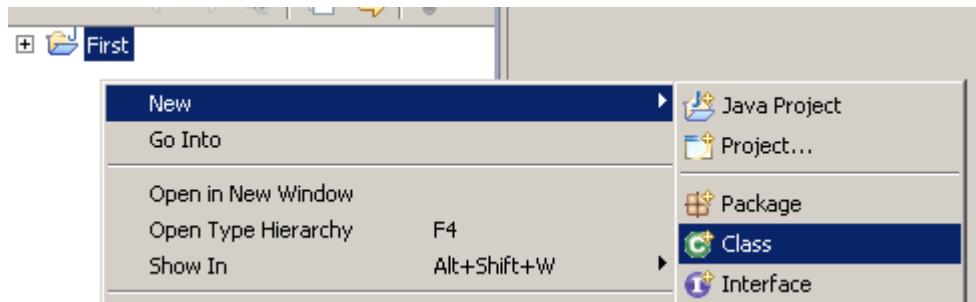
13. You should now see the two new JARs added to your Libraries list



14. Click **OK**

You have finished setting up your project to run selenium, now we create a class and write a simple Selenium program.

4. Right Click on First Folder
5. Click on **New**
6. Click on **Class**



At this point a new Window will open up

7. In the Name field give it the name MySelenium
8. Click **Ok**

A new sub folder was created called “src” (short for source), and in this subfolder your class file was created called “MySelenium.java”

Further more your class file is opened up in eclipse and ready to be populated by code.

## Selenium Tutorial(IDE, Core, RC)

Lets try to run a simple Selenium program that opens of a Cafe express in Internet explorer.

Suppose if the user doesn't know java , no worries. Use selenium IDE and record the scripts.

Go to selenium IDE-> options->format-> choose Java-Selenium RC. Now you will be getting the following scripts in java format.

Step1: Open the mozilla firefox , I assume your homepage of the Firefox is <http://www.google.com>

Step2: Go to tools-> click SeleniumIDE

Step3: Now SeleniumIDE will open, minimize the Selenium IDE

Step4: By default, seleniumIDE opens in Recording mode

Step 5: Type the cafeexpress home page url, <http://10.6.13.180/default.asp>

Step 6: login to the cafeexpress, for someuser it will show session expired. Click the relogin link and move to the home page.

Step 7: click few links and do signout.

Step 8: Now if you go to Selenium IDE → Options → format → select Java-SeleniumRC , copy the scripts and paste it in Eclipse.

But usually the scripts won't be that much perfect like the following. Edit accordingly.

```
import com.thoughtworks.selenium.Selenium;
import com.thoughtworks.selenium.DefaultSelenium;

public class MySelenium {

    static Selenium browser;

    public static void main(String arg[])
    {
        browser = new DefaultSelenium("localhost",
        4444, "*iexplore", "http://10.6.13.180/default.asp");

        browser.start();
    }
}
```

## Selenium Tutorial(IDE, Core, RC)

```
browser.open("http://10.6.13.180/default.asp");
browser.open("/index.asp?message=expired");
browser.click("//table[@id='Table1']/tbody/tr/td/a/strong");
browser.waitForPageToLoad("30000");
browser.click("//table[@id='Table1']/tbody/tr[4]/td[3]/a/b");
browser.waitForPopUp("eprintwinclose", "30000");
browser.selectWindow("name=eprintwinclose");
browser.click("Submit3");
browser.close();

}
```

Supporting other tools:

Here we used eclipse for java and executed our scripts. There are many open source tools available in the market for supporting different programming language. We should know , which tool to use and how to use the tool.

Here I am mentioning few supporting tools for SeleniumRemote control. They are

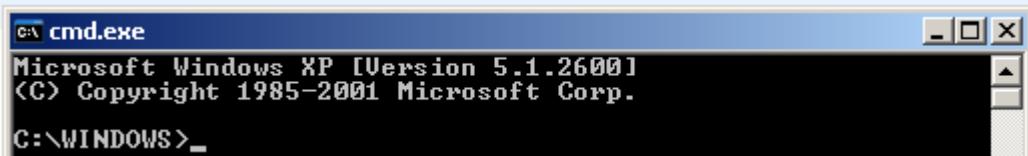
1. Bromine
2. Cubic Test
3. pyWinAuto
4. Frankenstein
5. Watir Recorder
6. WET
7. JTC

## Selenium Tutorial(IDE, Core, RC)

Save the file, and now before we run the program we need to start the Selenium Server.

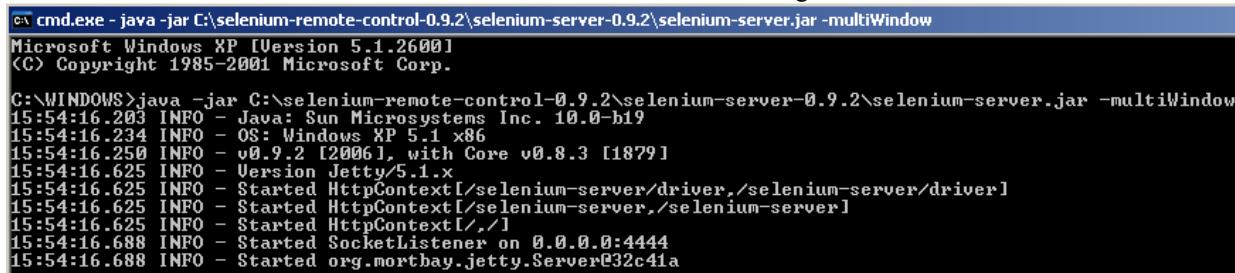
8. Click on **Start** (Windows Start menu)
9. Click **Run**
10. Type cmd

This will open up the command window



11. Type in **java -jar <location of your selenium-server.jar file> -multiWindow**  
(i.e. **java -jar C:\selenium-remote-control-0.9.2\selenium-server-0.9.2\selenium-server.jar -multiWindow**)

The Selenium Server should be running now:

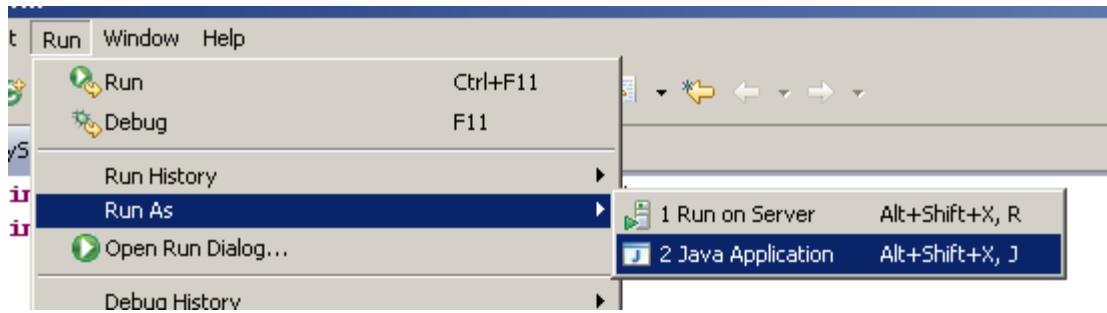


```
cmd.exe - java -jar C:\selenium-remote-control-0.9.2\selenium-server-0.9.2\selenium-server.jar -multiWindow
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS>java -jar C:\selenium-remote-control-0.9.2\selenium-server-0.9.2\selenium-server.jar -multiWindow
15:54:16.203 INFO - Java: Sun Microsystems Inc. 10.0-b19
15:54:16.234 INFO - OS: Windows XP 5.1 x86
15:54:16.250 INFO - v0.9.2 [2006], with Core v0.8.3 [1879]
15:54:16.625 INFO - Version Jetty/5.1.x
15:54:16.625 INFO - Started HttpContext[/selenium-server/driver,/selenium-server/driver]
15:54:16.625 INFO - Started HttpContext[/selenium-server,/selenium-server]
15:54:16.625 INFO - Started HttpContext[/,/]
15:54:16.688 INFO - Started SocketListener on 0.0.0.0:4444
15:54:16.688 INFO - Started org.mortbay.jetty.Server@32c41a
```

Now that Selenium Server is running in the background lets go back to Eclipse and run our program.

3. Click on **Run**
4. Click on **Run As**
5. Click on **2 Java Application**



And you are DONE! Congratulations you have written your first Java-Selenium program using Eclipse.

Similarly you can get the scripts for other programming language from SeleniumIDE and run the scripts in Selenium Remote control.

## Selenium Tutorial(IDE, Core, RC)

### Using Selenium RC With Multiple Users:

Additional information Selenium RC server on any port you wish, allowing multiple clients to connect at the same time.

To start the selenium RC server, you would typically run this command:

```
java -jar selenium-server.jar
```

Which will start it on the default port of 4444. If you want to start it on another port, simply add -port <portnumber> to the command like this:

```
java -jar selenium-server.jar -port 1234
```

This command would start the server on port 1234.

### **Great, now I can start the server on multiple ports, so what?**

So here is the good news for those of us in shared development environments. Say you have a windows virtual machine running selenium with your test browser installed. Things get complicated when multiple clients are testing at the same time. Tests can fail unexpectedly, developers begin fighting for time on the test server, and all hell breaks loose. By running multiple servers on different ports, we can avoid the third world war. Everyone can connect at the same time to their personal server, and we have peace and harmony in our testing environment.

**User can even create a .bat file which is having the following code. So that selenium RC automatically starts running in different port.**

```
start "Selenium 4444" /min java.exe -jar selenium-server.jar -port 4444
start "Selenium 4445" /min java.exe -jar selenium-server.jar -port 4445
start "Selenium 4446" /min java.exe -jar selenium-server.jar -port 4446
start "Selenium 4447" /min java.exe -jar selenium-server.jar -port 4447
```

With this in place, we now have 4 selenium RC servers running at all times. If one or more crashes for any reason, running the batch file again will start any that are missing. The selenium RC server will not start if its port is in use, so only the ones missing will run, and you'll have them all up and running again.

**Ok, what about the client side? I don't want to have to edit another file in my development environment.**

Suppose if you are using any programming language then you have to write your scripts in the concern language for execution.

## Selenium Tutorial(IDE, Core, RC)

For example:

If I am using php scripting language then I have to write my scripts in the code and need to open the selenium RC in different port.

Ex:

```
<?php
class SeleniumTestHelper
{
    public function setup()
    {
        $ports = array('user1' => 4444,
                      'user2' => 4445,
                      'user3' => 4446,
                      'user4' => 4447);
        $user = someFunctionThatDeterminesUser();
        define('BROWSER', '*firefox');
        define('URL', 'http://www.example.com');
        define('SERVER', '192.168.0.2');
        define('PORT', $ports[$user]);
        $this->selenium = new Testing_Selenium(BROWSER, URL, SERVER, PORT);
    }
}
?>
```

Congratulation !!! now the user comes to know how to run the scripts in different port.

## Selenium Tutorial(IDE, Core, RC)

### Platforms Supported by Selenium

#### Browsers

Browser	Selenium IDE	Selenium Remote Control	Selenium Core
<b>Firefox 3</b>	Record and playback tests	Start browser, run tests	Run tests
<b>Firefox 2</b>	Record and playback tests	Start browser, run tests	Run tests
<b>IE 8b1</b>	not supported	?	?
<b>IE 7</b>	not supported	Start browser, run tests	Run tests
<b>Safari 3</b>	not supported	Start browser, run tests	Run tests
<b>Safari 2</b>	not supported	Start browser, run tests	Run tests
<b>Opera 9</b>	not supported	Start browser, run tests	Run tests
<b>Opera 8</b>	not supported	Start browser, run tests	Run tests
<b>Others</b>	not supported	Partial support possible*	Run tests**

\* Selenium Remote Control server can start any executable, but depending on browser security settings, there may be technical limitations that would limit certain features.

\*\* Selenium Core is written in Javascript so it should work (in theory) on most browsers with decent Javascript support. Some operations are only possible via Selenium Remote Control because of browser security settings.

### Operating Systems

OS	Selenium IDE	Selenium Remote Control	Selenium Core
<b>Windows</b>	Works in Firefox 2+	Start browser, run tests	Run tests
<b>OS X</b>	Works in Firefox 2+	Start browser, run tests	Run tests
<b>Linux</b>	Works in Firefox 2+	Start browser, run tests	Run tests
<b>Solaris</b>	Works in Firefox 2+	Start browser, run tests	Run tests
<b>Others</b>	Should work in Firefox 2+	Start browser, run tests*	Run tests**

\* Selenium Remote Control server is written in Java, so it may run on other systems as well, as long as there's a supported web browser available.

\*\* Selenium Core is written in Javascript so it should work (in theory) on most browsers with decent Javascript support. Some operations are only possible via Selenium Remote Control because of browser security settings.

## Selenium Tutorial(IDE, Core, RC)

### Programming Languages

Programming languages are supported through Selenium Remote Control "drivers." These are libraries made for each language that expose commands from the [Selenium API](#) natively in the form of methods/functions.

Language	Selenium IDE	Selenium Remote Control	Selenium Core
C#	Generate code	Library ("driver") support	n/a
Java	Generate code	Library ("driver") support	n/a
Perl	Generate code	Library ("driver") support	n/a
PHP	Generate code	Library ("driver") support	n/a
Python	Generate code	Library ("driver") support	n/a
Ruby	Generate code	Library ("driver") support	n/a
Others	Generate custom code*	Commands via HTTP requests**	n/a

\* You can actually set up a custom code generator to output to any language. The templates are written in Javascript, but they just output text.

\*\* Any language that can make an HTTP call can pass commands to the Selenium Remote Control server. Languages listed here have a wrapper library made already.

### Testing Frameworks

Testing frameworks aren't required, but they can be helpful if you want to automate your tests.

Framework	Selenium IDE	Selenium Remote Control	Selenium Core
Bromine	Comes with template to add to IDE	Manipulate browser, check assertions via custom driver	Special support**
JUnit	Out-of-the-box code generation	Manipulate browser, check assertions via Java driver	n/a
NUnit	Out-of-the-box code generation	Manipulate browser, check assertions via .NET driver	n/a
RSpec (Ruby)	Custom code generation template*	Manipulate browser, check assertions via Ruby driver	n/a
Test::Unit (Ruby)	Out-of-the-box code generation	Manipulate browser, check assertions via Ruby driver	n/a
TestNG (Java)	Custom code generation template*	Manipulate browser, check assertions via Java driver	n/a
unittest (Python)	Out-of-the-box code generation	Manipulate browser, check assertions via Python driver	n/a
Others	Custom code generation template*	Manipulate browser, check assertions via HTTP requests***	n/a

\* Built-in code generation creates code specifically for the "out-of-the-box" frameworks, but you can modify the Javascript-based templates to output any kind of code you want.

\*\* Bromine is built specifically for Selenium and it directly supports both Selenium Core and Selenium

## **Selenium Tutorial(IDE, Core, RC)**

Remote Control. The other testing frameworks use Selenium Remote Control drivers for testing.

\*\*\* Using the appropriate driver (or HTTP commands), you can manipulate and test the browser using any testing framework by calling commands like "click" or "isElementPresent." Selenium Remote Control just deals with the commands and doesn't care about what actually calls those commands.

## **Conclusion:**

There are n no. of open source tools available in the market. Each tool has it's own advantages and disadvantages. When it comes to open source tool contribution is very less. Obviously when you compare the opensource tool with the licensed version, the disadvantages may seen visible. But if the tester who is using the tool knows the tool end to end then he/she can able to handle any sort of situation successfully. Similarly selenium is also having it's own advantage and disadvantages but still people are using frequently shows that the tool is having something unique and useful to the IT organization. So learn and explore the tool thought it's useful now or not. The tools success is entirely depends upon the person who is going to use the tool. Selenium actually reached certain level of success in the market which really shows not only the tools features but also the people who used it in the right situation.