

# SOFTWARE AUTOMATION TESTING SECRETS REVEALED

## **(Revised Edition) -Part 1**

**Cucumber BDD, Selenium Web driver, Protractor, Selenium Grid, Appium, TestNG, Jenkins, UFT, RFT, Visual Studio, Excel VBA, SOAP, Selenium IDE based Automation Testing.**

NARAYANAN PALANI



Software Automation  
Testing Secrets Revealed

Publishing-in-support-of,

# **EDUCREATION PUBLISHING**

RZ 94, Sector - 6, Dwarka, New Delhi - 110075  
Shubham Vihar, Mangla, Bilaspur, Chhattisgarh - 495001

**Website:** *www.educreation.in*

---

Second Edition : 19 Oct'17  
First Edition : 12 Feb'16

### **© Copyright, Authors**

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted, in any form by any means, electronic, mechanical, magnetic, optical, chemical, manual, photocopying, recording or otherwise, without the prior written consent of its writer.

**ISBN:** 978-1-5457-0825-5

**Price:** INR 1015.00

The opinions/ contents expressed in this book are solely of the authors and do not represent the opinions/ standings/ thoughts of Educration or the Editors .The book is released by using the services of self-publishing house.

Printed in India

### Disclaimer

This is a work of author's personal automation experience summary and author do not represent any company or any form of organisation in this book. Names, tools, websites, characters, businesses, places, events and incidents are either the products of the author's imagination or used in a fictitious manner. Any resemblance to actual persons, living or dead, or actual events is purely coincidental.

Although the author and publisher have made every effort to ensure that the information in this book was correct at press time, the author and publisher do not assume and hereby disclaim any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from negligence, accident, or any other cause.

This book is not intended as a substitute for the official guides provided by tools such as selenium, uft and ranorex etc. The reader should regularly consult official pages of automation tools in matters relating to features and functionalities and particularly with respect to any code examples that may require to build automation frameworks.

The information in this book is meant to supplement, not replace, proper testing tools. Like any tools involving software engineering, (testing tools) poses regular changes and releases. The authors and publisher advise readers to take full responsibility for upgrading their knowledge by visiting official web pages of automation tools. Before practicing the skills described in this book, be sure that your testing tool is latest and downloaded from official websites of software testing tools

### Credits:

Selenium ©Apache 2.0 License

Cucumber © Cucumber Limited

Ranorex © Ranorex GmbH

SoapUI © 2017 SmartBear Software

Quick Test Professional, Unified Functional Testing© Copyright 2017 EntIT Software LLC  
Visual Studio ©Microsoft 2017

Jenkins.io ©the Creative Commons Attribution-ShareAlike 4.0 license

Rational Function Test(RFT) ©IBM

Protractor is a nodejs program © <http://www.protractortest.org>

# **Software Automation Testing Secrets Revealed**

*(Revised Edition) - Part 1*

*Cucumber BDD, Selenium*

*Webdriver, Protractor, Selenium Grid, Appium,  
TestNG, Jenkins, UFT, RFT, Visual Studio, Excel VBA,  
SOAP, Selenium IDE based Automation Testing*

## **Narayanan Palani**



**EDUCREATION PUBLISHING**

*(Since 2011)*

[www.educreation.in](http://www.educreation.in)

- Learn to write automation test scripts using Selenium Webdriver in java programming, javascript, C#, python and run in Cucumber BDD feature files.
- Conduct experiment to write protractor-based Cucumber BDD framework in javascript.
- Build TDD frameworks with the help of TestNG, Visual Studio, Jenkins, Excel VBA, Selenium, HP UFT (formerly QTP), Ranorex, RFT and other wide-ranged QA testing tools.
- Design first Appium scripts after setting up the framework.
- Build concurrent compatibility tests using Selenium Grid!
- Reap the benefits of well-built Selenium Webdriver automation framework!
- Apply powerful programming techniques in order to build well-established data-driven framework, keyword-driven framework, hybrid framework, Agile Continuous Integration Automation Framework.
- Exploit opportunities to design web service-based REST/SOAP automation frameworks and SQL automation to validate database!
- Utilize construction practices using Object Recognition technology of test automation tools and capture dynamic web elements.
- Debug problems in software development life cycle with a view to improve testing process using test estimation techniques, tool selection process.
- Raise quality automation scripts by understanding the machine-learning basics of Selenium automation framework properly.
- Resolve automation issues and introduce cucumber, SpecFlow-based behavior-driven development and test-driven development in functional testing.
- Repeated interview questions are explained with justifications for Cucumber BDD, Selenium IDE, Selenium Webdriver and Selenium Grid.

**Note: Part 2 has been sold separately at**  
<https://www.education.in/store/selenium-webdriver-software-automation-testing-secrets-revealed-part2-na-rayanan-palani.html>

This book has been dedicated to  
**Sree Raghavendra Swamy**

ବ୍ୟାକ୍‌ରୀତି-୫

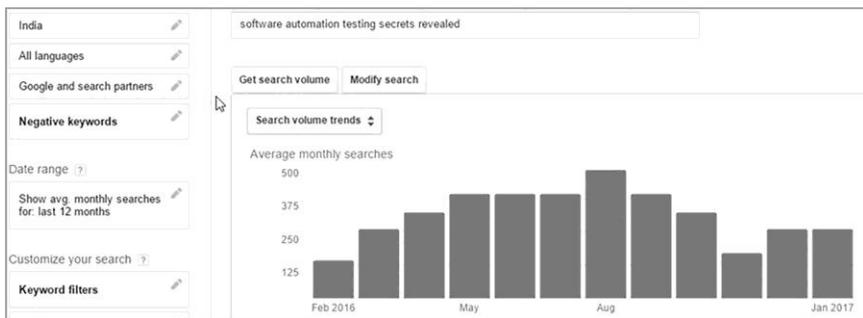


Pujyaya Raghavendraya Sathya Dharma Rathayacha  
Bajatham Kalpa Vrukshaya Namatham Kamadehnave

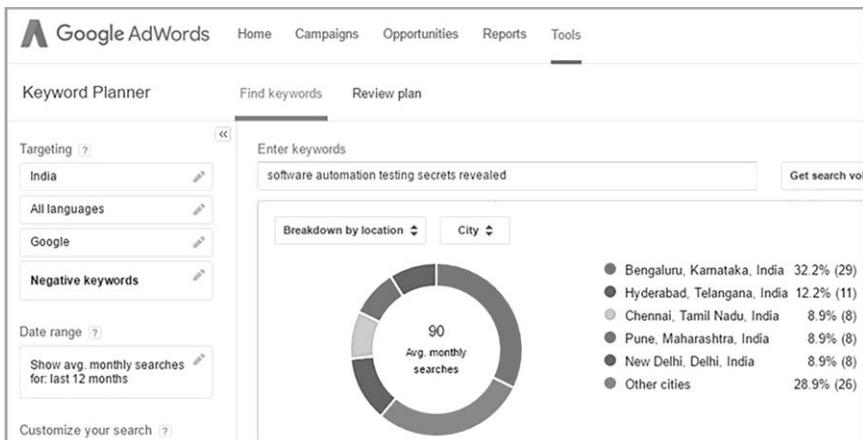


# Most Searched Computer Science Best Seller

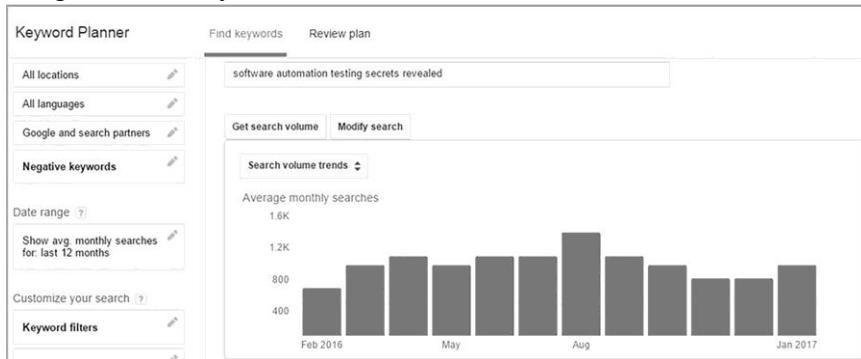
The first edition of the book, “Software Automation Testing Secrets Revealed”, was published on 12 Feb '16. It has been searched and referred across the globe and sustained to be the three times best seller in computer science books in India!



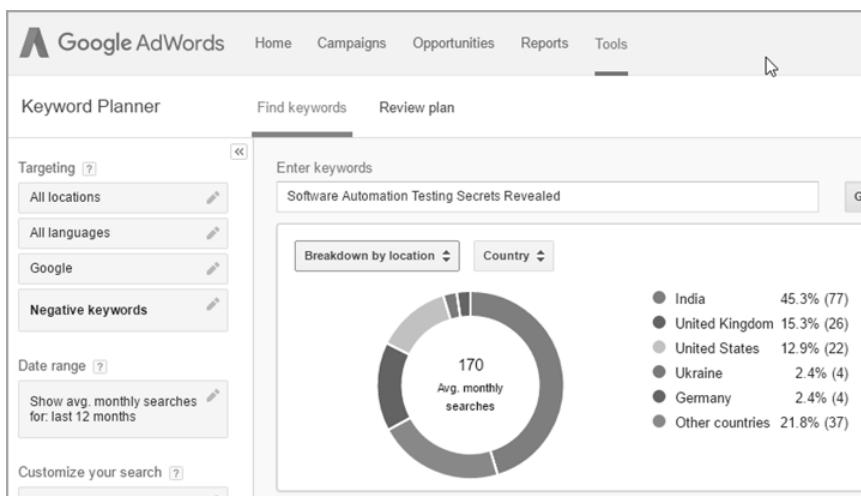
India wide Google Search-Google Keyword Planner Results for the period Feb '16–Jan '17.



Readers are located across the IT-based Indian cities, such as Bangalore and Hyderabad.



Worldwide Google Search-Google Keyword Planner Results for the period Feb '16–Jan '17.



# Global Reach of the Book

## CHARITY CONTRIBUTION

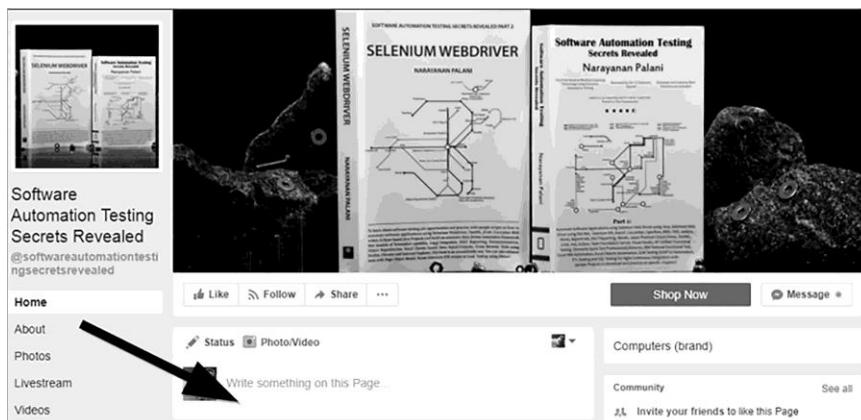
The first publication of this book has been ranked #1 in Computer Science Sales Ranks of Amazon India (as on 19 Aug '16) and successfully crossed more than 1,069 copies (from Feb '16 to Jun '17) of sales worldwide; hence, part of author royalty has been distributed for the following charity programs:

Serial no.	Amount (INR)	Type of charity	Schedule of event	Location	Transaction ID	Transaction date
1	1,019.55	Food distribution	27/03/2017	Srirangam, India	384621490621369	27/03/2017
2	2,000	Food distribution	11/04/2017	Alangudi, India	100001031007	28/03/2017
3	2,000	Food distribution	11/04/2017	Thirukarugavur, India	IRC6226851	27/02/2017
4	2,000	Food distribution	24/05/2017	Rameshwaram, India	100001024791	27/03/2017
5	2,400	Food distribution	11/06/2017	Thirukarugavur, India	IRD4121002	10/05/2017
6	2,000	Food distribution	18/09/2017	Rameshwaram, India	100002117914	23/07/2017

## **Review This Book And Get “Mobile Software Testing” Book Free**

Please rate this book with your respective review comments in the website from where the book has been purchased (like amazon). If the book is lent from some other sources, such as libraries, kindly rate the book in amazon or goodreads and also share the screenshot(of the website with your name,review and date visible) and email id in the following facebook page:

[www.facebook.com/softwareautomationtestingsecretsrevealed/](https://www.facebook.com/softwareautomationtestingsecretsrevealed/)



Selected posts(screenshots)of positive reviews will receive e-book of Mobile Software Testing for free of cost within 90 days from the date posted based on author's decision.



# **FAQ About the Book**

## **“Software Automation Testing Secrets Revealed” – Is this book mainly for Selenium?**

The first part of the book offers an introduction note on each automation tool-based testing taking account the respective viewpoints of the author itself; the succeeding part of the book predominantly deals with detailed explanation of Selenium Webdriver.

## **Why this book has been republished in the year 2017?**

On 12 Feb '16, the first edition of the book was released, and it received amazing sales record of more than thousand copies across the world within the first year of release. Moreover, it has been consistently ranked first on the computer science best sellers in India sales channels. In order to describe each testing tool with latest information, it is indispensable to revise the topics and provide latest details to readers. Thus, the republication has been planned during 2017 along with latest tool-based sections to facilitate engineers preparing for software testing job interviews and certifications.

## **Can a reader expect complete knowledge of automation testing in this book?**

This book is an author's diary on automation testing instructions based on his project experiences and knowledge regarding the automation testing tools. Clear direction will be provided by the habit of learning from a book, whereas complete knowledge is possible only when those learnings are put into practice, for example scripting the regular automation in automation projects.

## **Does this book contain clear split of tools and flow between every chapters?**

Not less than ten different automation tools have been discussed in this book which includes Cucumber BDD, Selenium, TestNG, Jenkins, Ant, Appium, UFT, RFT, Visual Studio, Excel VBA, SOAP, AutoIT etc. Hence, expecting a flow from one chapter to another chapter is impossible while programming languages may differ from each other at times. So this book can be preferred to refer as a guidance to build initial framework.

## **Why topics such as Jenkins and AutoIT are not explained in detail?**

Primary purpose of this book is to provide basic introduction of the frameworks with the help of different set of automation tools; hence, explaining every tool in detail in one book is inconceivable. So, Part 2 comes as an aid, which has been released to explain minutely the various aspects of Selenium Webdriver, and the descriptions of tools such as Jenkins, AutoIT, SOAP, HP UFT are not compiled within this book. Particularly, Selenium Grid is included only with few samples since it's difficult to cover all the tools in detail with multiple samples. If such tools need more scripting examples, there may be possibilities on Part 3 onwards (please search in Google for Part 3 and Part 4 availability).

## **Who are the target readers for this book?**

As this book is a fundamental introduction to test automation, it is primarily targeting freshers, non-IT professionals learning software testing to join IT organizations. It is also important to note that the book will not make any sense for experienced test automation professionals. Part 2 is highly suggested for engineers with few years of automation testing experience.

## **Where the readers can refer the code examples since there are very less or few code examples in the book?**

Since the target readers are primarily from graduation, freshers, Non-IT background and are novice to the software engineering roles, most of the scripts are accessed from GitHub and the details

provided in relevant sections of this book. So reading this book will offer justifications and explanations with limited amount of code examples only. The basic idea lies in getting readers' understanding about the topic using this book, referring the video to know how to script and the GitHub code base to use them to start writing first set of scripts in their automation testing learning experiences. Please refer "How to read this book" section for more details.

**I just have Part 1 of this book. Can I get enough learning on Selenium scripting?**

This book is not only meant for Selenium but also explains most of the major "most used" automation tools. Part 1 alone cannot serve the purpose of learning Selenium completely. So Part 2 is predominantly focused on Selenium Webdriver which is widely used as a web automation tool.

## **Praise for the Book**

This book has been reviewed by a number of technology professionals across the world. Some of their comments are listed below:

"This is one among the best books in the field of automation testing. This smartly demonstrates a nice blend of automation tools and its efficiency. A must read for testing professionals for enriching their proficiency and mastering the automation world."

Satyadip Das, Test manager, Lloyds Banking Group

"Those want to jump-start their carrier in automation testing. This book is a bible for them to guide and lead the life in a systematic manner."

**Arul Velan, Wipro technologies**

"Very much helpful for those who are new to automation testing, Detailed explanation with perfect practical coverage. What else you need to jump-start in the globe of software automation!"

**Altamash Khan, Senior Test Engineer,  
Wipro Technologies**

"This book provides practical insight into the world of software testing. Very concise and just filled with excellent information."

**Anku Jain, Senior Test Engineer,  
Wipro Technologies**

"Automation testing – doesn't only means that you save some time. It just more than saving the time, it gives a life to the system to function on its own."

**Sathyar Narayanan, Wipro Technologies**

"Future belongs to those, who believe in the beauty of their dreams. And this book will surely help you in guiding your way toward automation testing."

**Pratiksha Kadam, Senior Software Engineer,  
Wipro Technologies**

"In celebration of the release software automation testing, I appreciate your dedication and hard work. You deserve to be proud on your achievement. May your future efforts be equally successful and rewarding. Thanks for releasing software automation testing."

**Raman Sharda**

"Encyclopedia of all automation tools!"

**Raghavendra Mesta, Wipro Technologies**

"This will be definitely helpful for all the current automation test engineers and upcoming automation test engineers. All the best!"

**Greeni Hari, Wipro Technologies**

"Hope it is easily understandable by everyone."

**Jagan Kanniappan, Accenture**

"Great effort by the author! It is a step-by-step guide which aims at making sure anybody can do automation be it Selenium or any

other tool. I would say this book is a wonderful package available in market right now, just go for it."

**Shweta Bhat, Wipro Technologies**

"Book is really helpful in understanding core of automation testing, a great guide that is useful in planning test strategy and fit right into product management."

**Ishwar Singh, Consultant, Deloitte**

"Let the 'Software Automation Testing' be helpful to all the testing professionals to explore the unexpected."

**Neenu Ratheesh, Wipro Technologies**

"This book will be a great help for anyone to learn about *Test automation frameworks*, use as an effective reference guide to select and use various automation tools. This book is another solid contribution from Narayanan Palani to help anyone to develop and become an expert in automation testing."

**Libin Jose Mannala, Wipro Technologies**

"Automation testing is the current and future approach for the robotic systems. This book from such an experienced and thoughtful person Mr. Narayanan will sure help the budding techies to enhance as an expert."

**Arun Kumar, Technical Lead, Wipro Technologies**

"Automation testing is more than enough for a smart tester to sustain in smart world. This book enlightens the way to achieve it in the crazy labyrinth of test automation techniques."

**Sumayya Nej, Wipro Technologies**

"Time is power and money, and that's exactly why automation testing has an important role in the software industry. It is the solution to maximize the effectiveness, efficiency and coverage of software testing. Automation testing reduces the work force as well as the time for repetitive tests from days to hours with improved accuracy. Identification of the right automation tool is critical to ensure the success of the project."

**Neethu Joseph**

"Great book! This book is for people who want to know in depth of Selenium Webdriver concepts. And also the topics have been explained in detail which will be very useful for the people who want to start a new career or for cracking interviews. This is a must buy!"

**Dinesh Balaji,  
Test Automation Specialist, US BANK**

"Automated tests can run fast and frequently, which is cost effective for software products with a long maintenance life."

**Rahul Bhat**

It is a fantastic book! If your organization is interested in having a mature automated testing process, you need this book. I already recommended it to all friends on mine and got very good reply. I would recommend it to anyone who already has or planning to automate their testing practices. Thank you for the great work!

**Rohan Vakharia**

"Software Automation Testing' presents a test automation methodology which is practical, useful and complete, encompassing the state of the art of test automation as it stands right now.

If your organization is interested in having a mature automated testing process, you need this book.

It will guide you throughout the entire process from thinking about standards to implementing and maintaining them.

Also, it guides readers through each step in the planning, selection and implementation process to assure that automated software testing will be developed in a systematic manner. Bon Apprentissage!"

### **Anjana Nambron, Wipro Technologies**

"Automation testing has gained more prominence in recent times because of the software industry rapidly evolving where functionalities of a mobile app or a web app have to be tested and certified thoroughly within a short span of time. This book will help the young job seekers in understanding on the processes, methodologies adopted by different companies having a wide range of automation practices and frameworks."

### **Sriharsha Karanth**

"Keep up the good work, as always you do. It will help everyone to know when exactly automation testing can be used and how useful it will be by saving time ...."

### **Vijayalakshmi N Savanth, Wipro Technologies**

"I know you are dynamic and interesting blog writer. Nice to see you as a good author. You are brilliant person and we are expecting soon to read the revealing secrets of the Software Automation Testing ... your abstract tells – sample projects to download and practice on specific chapters! This would be helpful for advanced and all levels of testing professionals."

### **Vijeyasekaran Kbd**

"With growing trends of companies rapidly moving toward automation testing, having automation skills are no more considered just a 'good to have skill'. For IT professionals who

want to have a career in automation testing, this book is a great start to kick off as Narayanan Palani has explained in detail the fundamentals of automation. This book is a great guide to learn automation."

**Rajavarman Rajendran, Test Specialist, US BANK**

"This book is really helpful for people who want to grow in the testing domain. Everything is explained in very easy language. A book highly recommended for testers."

**Soma Mazumder, Wipro Technologies**

"Automation testing is very important in every project testing life cycle. Through automation, we can achieve more test cases in less time. So it is both cost effective and also helpful to achieve the stringent time lines. This book will be an eye opener for job seekers in the above field, to achieve their goals and also to optimize their performance."

**Sunit Roy, Consultant, Capgemini**

"This book will surely encompass all corners of testing a to zee .... It will be beneficial for all testers irrespective of what profile they are into ... people looking for transitioning from manual to automation will get the most ...."

**Shubham Joshi**

"It will be beneficial for all testers irrespective of what profile they are into ... people looking for transitioning from manual to automation will get the most ...."

**Karthik Manoharan, Senior Analyst,  
Wipro Technologies**

"Good"

**Krishnamurthy Chinnasamy,  
Finance Manager, Britannia Industries Limited**

"Very nice Narayanan."

**Kothandan Sivalingam,  
Quality Analyst, Dupont sustainable solutions L&D**

"Clear and easy to understand! Congrats"

**Kamalbharathi Murugesan,  
South Plainfield, New Jersey**

"I, as a professional automation tester, often fall in situations where we need to take call on what automation solution to employ for a project that would be most effective and efficient at the same time. But it is hard to take such decisions with limited insight and knowledge of the available tools for automated testing in the market today. The book here by Narayanan provides a panoramic view into these tools, thus empowering the reader to take more educated decisions to provide better automation test solutions."

**Sohag Satpati**

"Software automation testing – secrets revealed follows a job-oriented testing process that can make anyone industry ready. Keep up the good work & all the best!"

**Mitha SR Nair, BA, Wipro Technologies**



# About Author



**NARAYANAN PALANI**

**MS (SOFTWARE ENGINEERING) – BITS Pilani**  
**EXEC MBA** from K.J. Somaiya Institute  
of Management and Studies, Mumbai

## **EXPERIENCE SUMMARY**

Nine years of strong QA experience in automation, functional, performance and penetration testing

Exclusive agile test projects experience using Selenium, Cucumber, TestNG, HP UFT and Ranorex

Strong on test process, programming such as Java, JavaScript, VBScript, C#, Ruby, Python, Gherkin

Proven test automation capabilities in TDD and BDD – SpecFlow for integration and unit testing

Formulated high-quality master test plan, test strategy, automation PoC, estimates and test scripts

Efficaciously implemented SDLC, ISO and CMMI standards and worked on Agile model-based projects

Delivered high-quality test results and defects using HP QC, Rally and other test management tools

Experienced investment banking, trading and FX projects using Murex, Intelligent FX and Summit

Directed functional, automation, performance and support members to deliver high-quality results

## **CERTIFICATIONS**

International Software Testing Qualifications Board – CTAL Test Manager

International Software Testing Qualifications Board – CTFLCertified Scrum Master

PRINCE2 Project Management Foundations

Microsoft Certification Software Testing using Visual Studio

ITIL Foundation Certification

HP-accredited Integration Specialist in Application Security

IBM-certified Specialist in Rational Appscan

HP-accredited Integration Specialist in Quality Center v9 and QTP v9

IBM-certified Rational Specialist in Test Management and Rational Robot

IBM-certified Solution Designer in Rational Functional Test

IBM-certified Solution Designer in Rational Performance Test

IBM-certified Solution Designer in Rational Manual Tester

IBM Examinations in Requirements Gathering and Use Case 1 & 2

## **BACKGROUND**

Narayanan Palani is endorsed as exceptional talent within digital technology by Tech City UK. Being in a leading position of proven and renowned software testing industry, Narayanan Palani volunteered to share his technical knowledge through his best-selling book series “Software Automation Testing Secrets Revealed” which ranks top amongst the books on computer science across APAC, USA and UK. His well-designed GitHub open-source testing projects are accessed worldwide profusely and have been assisting job seekers to bag junior testing jobs successfully.

Being a leading test specialist and, at the same time, holding six awards, fifteen international certifications, eight published research papers and four books selling worldwide, he proceeded further to innovate and implement cutting-edge technologies as part of his recent roles. His IBM professional certificate profile<sup>i</sup> is viewed worldwide, and newcomers are being inspired to take interest in

software testing profession. His name is also enlisted for the nomination of the Testing Leader of the Year 2016<sup>ii</sup> and 2017.<sup>iii</sup>

As Narayanan is groomed from testing research background, he has been answering the questions from the people worldwide and is ranked no. 1 with more than 566,000 viewers in Quora. His more than 55 LinkedIn<sup>iv</sup> recommendations are strong evidences of his novel work as a testing leader, and the research papers such as Right Brain Testing using Gestalt psychology<sup>v</sup> is still referred by the research scholars for software testing R&D.

Apart from his spectacular personal records on software testing, he has trained and developed at least twenty-one non-IT/fresher candidates into software testing within a short period of one year, and surprisingly, most of them are showcasing their abilities as strong team members in testing projects and, thus, his ambition of generating employability among the job seekers has been materialized.

### **Linkedin Profile:**

<https://www.linkedin.com/in/narayananpalani/>

## **Foreword**



**Ponsailapathy Viswanathan  
Director-Delivery, Virtusa Polaris**

*Ponsailapathi Viswanathan “Pons” has over 22+ years of IT experience in Travel, Banking and Insurance industries, mainly focusing on software delivery and solution. He has played key role in building the testing practice at Europe for his organization and handled strategic roles in the design and set-up of TCoE engagements for global banks across Americas, Europe & Middle East.*

*He led the set-up of near-shore delivery center for his organization in Northern Ireland, UK to service European customers and successfully established global delivery teams which can operate in tri-coastal model for global banks.*

*Also built technology CoEs for mobile testing, security testing & nurtured test automation, performance engineering practices to develop innovative solutions for Banking & Insurance customers.*

### **Linkedin Profile:**

<https://www.linkedin.com/in/ponsailapathi-viswanathan-224a755/>

# **Foreword**

Information technology is one of the challenging and rewarding industry. As per NASSCOM & other market trends the global testing market is estimated to reach US \$50 Billion by 2020. QA enterprises and test automation engineers will find exciting opportunities in the area of big data, mobile and API testing. Test automation skill is going to be one of the most wanted skill in Y2017 and beyond.

Considering the industry demand, there is a need to upskill ourselves continuously to stay ahead of the game and reap the benefits of latest technology. Now the industry is looking for cost effective solutions, tools like Selenium will be the best fit to manage the client requirements both in terms of technical complexity and cost.

To pick-up the nuances of test automation quickly, Narayanan Palani has scripted this book very nicely with simple examples based on his experience.

As an automation architect, Narayanan Palani has successfully implemented robust test automation utilities for complex trading applications in banking space. His penchant for perfection and innovative approach has created a huge positive impact in test automation projects with excellent results in terms of ROI (Return on Investment). Test automation techniques orchestrated by him made a difference in test automation strategy and maintenance of the test scripts.

Most of the people, don't share their hard learnings or techniques to others easily. Hats off to Narayanan Palani for his initiative to publish his decade long experience in the Test automation to aspiring candidates. He has articulated the end to end test automation techniques effortlessly in line with industry best practices. The secrets revealed here will enable the reader to build the test automation suite with ease.

I take this opportunity to congratulate Narayanan Palani for his effort to spread the learning's with testing community! Looking forward to see more such products from him in near future!

**Happy reading!**

Ponsailapathy Viswanathan, Director-Delivery, VirtusaPolaris

# How to Read This Book?

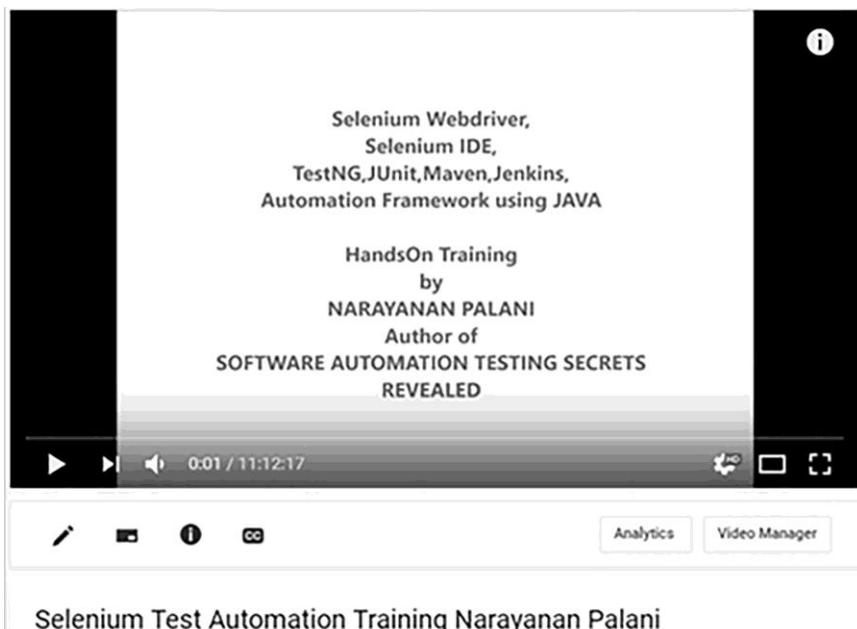
There are two parts associated with Selenium Webdriver reading, which are as follows:

Book1: Software Automation Testing Secrets Revealed

Book2: Software Automation Testing Secrets Revealed Part 2  
Selenium Webdriver

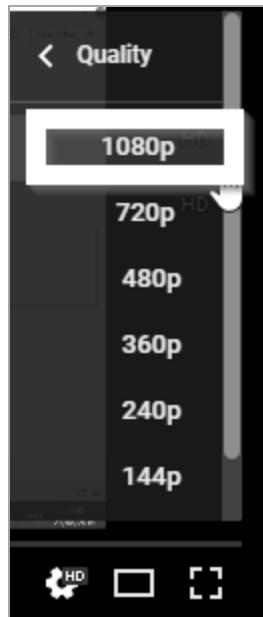
Video Support:

[https://www.youtube.com/watch?v=Ke\\_AQVxMhds](https://www.youtube.com/watch?v=Ke_AQVxMhds)



Both the books are supported by 11 hours 12 minutes training video available in YouTube.

**Note:** Change the *Settings* of *Quality* from 'Auto' to '1080p HD' for better view

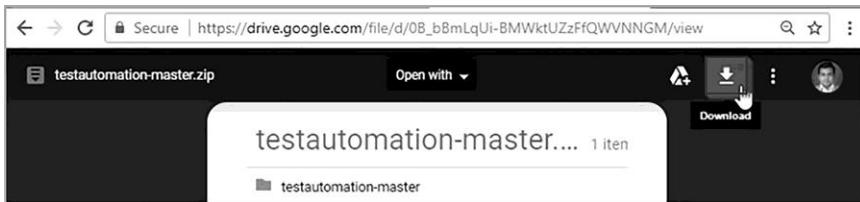


## Code support:

<http://github.com/narayananpalani/testautomation>

If readers are unable to download/clone the whole project from github, please download from:

<https://goo.gl/ayVfs4>



While reading this book, the technique of Pomodoro should be followed thoroughly for better understanding:

## Dos:

- Minutely focus your attention for 25 minutes without being interrupted on any chapter.

- Opt for a breather for 5 minutes to relax your mind. Perform Google search to analyze your area of difficulty and then jot them down on a paper (post these questions in quora.com or stackoverflow.com to get answered by experts).
- Write sample code and execute them to analyze results, exceptions or errors, if occurred any.
- Revise the same chapter at least 4 times in order to strengthen your concept as more concrete in its approach.
- A 15-minute break is advisable for refreshment and then should continue reading the next chapter.
- In case of meeting an alien programming language, in the course of reading, practicing Java programs using ‘Java BlackBook by Steven Holzner’ is duly suggested, thereafter, practicing scripts in command and be prompt to compile and execute.
- Save the practiced scripts and get them uploaded to Git and be an ‘open-source framework developer’ at [github.com/narayananpalani/testautomation](https://github.com/narayananpalani/testautomation).

### **Don’ts:**

- Never force yourself to complete the book overnight or within a week.
- Never plunge yourself into various programming languages until automation framework has been developed in that language.
- The task of reading the chapter and practicing the scripts should not be performed simultaneously; a comprehensive reading is must for a good conceptual understanding of the subject, so the reading should not be a futile attempt; ‘knowing’ and ‘doing’ are two different words.
- In case of meeting an unknown programming language in the course of reading, practicing Java programs using ‘Java BlackBook by Steven Holzner’ is advised, but practicing scripts in Eclipse or any IDE, while learning those fundamentals, is strictly prohibited, as command prompt gives quick learning in Java fundamentals (to analyze errors) rather than IDEs which help in building the scripts (whereas Eclipse can be used for Selenium-based learnings).

- The idea like programming language has to be learnt first in order to start learning Selenium scripting should not be presupposed; both can be learnt in parallel to understand how the script works for Selenium-based tests.

# Table of Contents

S. No.	Content	Page
1.	Introduction	1
2.	Test Automation Framework	4
3.	Selenium <ul style="list-style-type: none"><li>▪ Selenium versions and their capabilities</li><li>▪ FAQs about automation tests using Selenium tool</li></ul>	12
4.	Author's Perspective	18
5.	Cucumber BDD (Behavior-driven Development)	25
6.	How to Write First BDD Scripts in Cucumber, Selenium Using Java?	28
7.	Automation Framework Using Selenium Webdriver with DotNet	32
8.	Selenium Webdriver Tests in Python	41
9.	Selenium Test Automation Framework using Javascript	48
10.	Selenium Test Automation Framework using Java	51
11.	Selenium Webdriver Test Script Design Basics in JAVA	120
12.	Jason Phantom Ghost Driver Test Automation	125
13.	Maven Test Automation	130
14.	Schedule the Projects in Jenkins	150

15.	Selenium IDE	153
16.	Selenium Grid for Compatibility Tests	170
17.	Protractor End-to-End Automation Framework	194
18.	Automation Tool Selection-Know How	197
19.	Test Requirements	199
20.	Test Plans	199
21.	Test Scenarios	200
22.	Test Cases	200
23.	Agile Scrum-based Testing-Continuous Integration	203
24.	Test Estimation	206
25.	Automation Return on Investment	218
26.	Automation Test Scheduling	220
27.	Unified Functional Testing – Test Automation Basics	227
28.	UFT Test Automation Framework using Excel VBA	236
29.	Ranorex	240
30.	VB Scripting Excel Macro Automation	241
31.	How to Test Excel Sheet?	267
32.	IBM Rational Functional Test	269
33.	SOA Test Automation	278
34.	SOAP Messages Automation	280
35.	Data Warehouse Testing (DWH)	289
36.	Structured Query Language (SQL) for Software Testing	294

37.	Object Recognition	298
38.	Object Recognition Technology – automation testing	299
39.	Why to Learn Appium?	302
40.	Technical Round-based Interview Questions	366
41.	Selenium Interview Questions and Author's Views	370
	▪ Selenium IDE-based interview questions and author's views	375
	▪ Cucumber BDD-based interview questions and author's views	388
	▪ Selenium Webdriver 3.0-based interview questions and author's views	391
	▪ Selenium Grid 2 interview questions and author's views	401
42.	Selenium Cheat Sheet	406
43.	Book Chapter-based References	407

## Introduction

As an exclusive specialized industry, software testing is growing rapidly into automation, and the need of development experience and tools knowledge is enormous to sustain in testing profession! Tool-based testers are migrated to core automation testers over a period of time with various tool experiences. The agile-based projects brought the need for testers with strong development background; hence, automation testers are migrated to 'developer-in-test' roles to categories as a test specialist who is good at coding background.

This book brings the essence of automation basics in different topics varied from Selenium Webdriver to Excel VBA Test Automation – so that the target audience can be benefited with knowledge across the automation possibilities.

## How manual testers learn automation testing? 10 easy rules!

If you are decided to learn automation,

1. At first, choose right 'automation tool' and stick to it until learning the framework
2. Watch 'YouTube' related to the automation tool tutorials
3. Refer sample code-based 'Books' of the testing tool
4. Design your 'first script' in the automation tool with the help of sample scripts in [github.com](https://github.com)
5. 'Execute' and see how your automation script works!
6. Learn how you can actually design an 'automation framework' by yourself
7. Share your scripts in [github.com](https://github.com) and the script with fellow testers
8. Post your questions in forums (e.g. [qualitytesting.info](http://qualitytesting.info)) and get the valuable answers from experts
9. Document what you learnt, else you will forget over the years
10. Main point: 'teach' at least one person how to script first automation code!

## How to get benefited from this book?

Automation test frameworks are explained in the beginning of the chapters which will assist in understanding the types of frameworks used across the organizations.

Set of questions provided at the end of the chapter related to automation framework types – which are open questions and unanswered! The main reason is to keep these questions open and go through the rest of the chapters. So that the readers surely get the answer to these questions when the chapters are understood, and relevant open-source projects are practiced in parallel.

Note: These questions remain open in the absence of good practice. It is suggested to practice available projects online to move to the next chapters, which will help in building the skillset and get benefited maximum from the book in parallel.

Immediate chapters are discussing about how to select automation tools for the projects based on the nature of each tools.

When testers are contributing to automation test projects, the key factors differentiating them from others are: test estimation, test environments, cost estimation, time estimation, test schedule compression techniques, function point-based test estimation, Delphi test estimation technique, automation return on investment, automation test scheduling.

After discussing key foundations of automation tests, the main topics of automation tools are discussed such as Selenium IDE, Selenium Webdriver using Java/TestNG/Ant and Selenium Webdriver using Dot Net/Cucumber. Reason being, Selenium Webdriver using SpecFlow and C++ is an emerging field within BDD and TDD. This has been detailed in respective sections and moved toward other famous automation tools such as HP UFT, Excel VBA Automation and IBM RFT (rational functional test).

Since Cucumber is an emerging BDD tool in testing domain, this has been elaborated in an exclusive section. Along with Cucumber BDD, SOA tools such as SOAP UI also detailed to give basics on how to automate when the GUI is unavailable! This is irrelevant topic in this book to discuss specifically at sections of SOA,ETL domains – but the reason SOAP UI is discussed is mainly on the job market needs!

Knowledge in SOAP UI and ETL basics increases the possibilities to get good job profile in quick span of time. So these sections are briefly explained along with sample test strategy, test plan and test cases of functional testing.

Automation testing book explaining about functional testing is not new, and it is always beneficial to refer the foundations of functional/manual testing by sample documentations. Importantly, the basics of functional testing are provided through the sample test plan, test cases which will be helpful to build the fundamentals while practicing the automation framework.

## **Test Automation Framework**

*As a first step in test automation, it is important to learn how to build an automation framework; hence, this chapter explains from basic on how to build an open-source framework for testing.*

Set of practices, assumptions and concepts is used to design automation frameworks utilizing testing tools which are used for automation testing. The main benefit of automation framework is to provide better reusability of test scripts. This can be possible only if the automation framework is designed to help faster development of test scripts for new testing cycles, flexibility in adapting the product changes, comfort in training newcomers in order to understand the framework and especially the maintenance of automation framework once developed!

### **Test automation framework maintenance:**

Developing a test automation framework with open-source tools, such as Selenium, TestNG, Ant and Jenkins, to have web browser automation testing is simply an amazing idea! But this framework should support the maintenance of test data exclusively from excel sheets! Thus, it is important to extend the framework with the help of Read\_XLS.jar in order to read and write the excel from automation framework during test execution.

### **Types of automation test frameworks:**

Linear Scripting

Keyword-driven testing

Structured-driven testing

Data -driven testing

Hybrid automation framework

Agile Continuous Integration Automation Framework

### **Famous and widely used automation frameworks and test automation tools are as follows:**

Automation framework using Selenium Webdriver, JUnit, TestNG, Ant and Jenkins

Excel automation framework using HP UFT

Excel VBA Automation frameworks

Web Services (REST/SOAP) Automation Frameworks

## **Definition**

Set of assumptions, concepts, methods and workflows that constitute a work platform and best practice that provide complete support for automated automation framework testing and maintenance.

This includes script organization, documentation, coding standards &instructions, version control and unit test of the test scripts.

Key SFERMS Benefits – stable, flexible, extensible, reusable, maintainable, scalable.

Following are some of the key elements in which frameworks are assessed against:

## **Usability**

Automation framework usability can be described as how effectively new test analysts/SME (Subject Matter Expert) can use, learn or control the system. Some questionswhich are tobe asked to yourself to determine usability might be as follows:

Is there a UI metaphor (driver script) that I am using to help users adapt?

(for example, the ‘Eclipse’ is a metaphor for Selenium)

If a new test analyst takes responsibility over the test automation scripts, the amount of knowledge required on the subsystems and dependable software, such as Eclipse IDE (for Selenium Webdriver using JAVA), .Net (for Selenium Webdriver using C#) or any driver scripts which run their automation engine (such as UFT, Selenium) in the backend to produce results.

### **Best practice:**

It is always a best practice to use an exclusive usability

document for the actions such as Web Launch, User Login, Driver Script Launch; in addition to this document, library functions specification document should be designed to list the reusable libraries designed as part of the project-Sibbi Maruthu, Automation Test Program Manager

Are the most common operations streamlined to be performed quickly?

Actions, such as Web Launch, User Login, Driver Script Launch, are taken into consideration on how quick the automation framework is. Especially, Rational function test needs test configurations to be completed first in order to run the test scripts in test execution. Flexible tools, like Selenium IDE, look forward for assertions and assessors to be incorporated manually in the test script. On the basis of how quick these operations are performed in the test design, the overall test strategy gets affected on the test automation projects.

Can new test users quickly adapt to the automation framework without help? (Is it intuitive?)

If subject-matter experts learn the automation framework, it must be user friendly and straightforward in user interactions rather than code amendments which are complex for any new test users such as fresh test talents. What would be the ideal hands-on training required for the framework and how long it takes to master the tool – these are the key aspects of looking at it. While discussing on the training, it is also important to have a look at the test framework document on how extensive it has been documented. If the documentation is not in place, this could be one of the primary causes of automation framework failures over a period of time in agile testing projects.

Do validation and error messages make sense?

An excellent automation framework is judged on the basis of how effectively the errors are handled and the measures are taken into account in order to capture the maximum possible issues during test execution. Taking screenshots is not the only test deliverable

as part of automation. The match between expected and actual results has to be proved with the help of test inputs and outputs. Thus, validation check against the test results should result in the need derived from test requirements. If the test results don't match with the needs of testable requirements, though the automation tool runs the test scripts, it is unable to serve the purpose of test objective.

### **Maintainability (flexibility/testability)**

The definition of maintainability implies how brittle the code is to change. As a result, I tie the terms flexibility and testability into the overall maintainability of a project.

Does the entire team understand the code base or do knowledge islands exist? Is the code thoroughly regression tested?

Can modifications to the project be done timely?

#### **Best practice:**

Automation scripts, functional library, other test scripts can be externally controlled with different files which can be easy to maintain file model since any update to the functional library in future will affect specific files and rest of the framework remains reusable!

#### **Scalability**

Scalability is the ability for your program to gracefully meet the demand of stress caused by increased usage. In a nutshell, ensuring your program doesn't slow or bust while pounded by more users than you originally anticipated.

What is your current peak load that you can handle?

How many database records can be created until critical operations slow down?

Is the primary scaling strategy to “scale up” or to “scale out” – that is, to upgrade the nodes in a fixed topology or to add nodes?

## **Availability (or Reliability)**

How long the system is up and running and the mean time between failure (MTBF) is known as the availability of a program.

How long does the system need to run without failure?

What is the acceptable length of time for the system to be down?

Can down times be scheduled?

## **Extensibility**

Are there points in the system where changes can be made with (or without) program changes?

Can the database schema flex to accommodate change?

Does the system allow inversion of control (IoC)?

Can end users extend the system (scripts, user-defined fields etc.)?

Can third-party developers leverage your system?

## **Security**

The measure of system's ability is to resist unauthorized attempts at usage or behavior modification, while still providing service to legitimate users.

Does the system need user or role-based security?

Does code access security need to occur?

What operations need to be secured?

How will users be administered?

## **Portability**

Portability is known to be the ability for your application to run on numerous platforms. This can include actual application hosting, viewing or data portability.

Can the data be migrated to other systems?

For web applications, which browser does your web app support?

On which operating system does your program run?

The above questions are kept open and unanswered purposefully. The readers have to keep these questions in mind while going through further chapters in test automation. So these questions will be helpful in identifying answers for your own projects at the particular test environment. The next chapter helps in understanding the types of famous automation tools and their classifications.

### **How an uncontrolled automation program can cause millions of money to QA departments in 2020 and beyond?**

Quality assurance evolved from Glenford Myers' theory (1979) to college text books and remained as one of the standard practices for the application life cycle with continuous focus toward test automation! Test engineers of the 90s evolved as SDETs(software development engineers in tests)in current generation, and test cases are getting converted as BDD (behavior-driven development) scenarios! Undoubtedly, it is happening across the IT industry and happening for good!

Not less than 60,000 monthly Google searches are performed on Selenium Webdriver (a famous test automation tool) across the globe;-this statistics proves the demand for automation-based SDETs and the QA job market:

At the same time, we are heading toward a dangerous future with endless automation code garbage!

// Example of an infinite loop (Deadlock) in QA Test Automation Script:

```
while not selenium.is_text_present(  
    'logged'):  
    pass
```

//Reference:<https://saucelabs.com/blog/selenium-tips-infinite-loops-take-forever>

Meanwhile, industry has been observing that the script migration is almost impossible when organizations move from tools, such as HP UFT (earlier called QTP) to latest tools (e.g. Selenium Webdriver); the one and only option is to clear the garbage and rewrite the code from scratch – in this way, functional QA teams move toward a fast-phased test automation programs. So, the focus

toward 'script maintenance' is almost unrealistic since the amount of tests to get automated is humongous!

While millions of (QTP)old scripts/tests are getting rewritten in Selenium for the next ten years, it is 'highly possible' that the automation tests become non-maintainable garbage of the next generation!

Hence, there is a strong possibility for most demanding new role in testing programs(by 2020 approx.) just to maintain automation code and to unit test the entire QA code base in order to maintain automation scripts called 'software maintenance engineer in tests' (SMETs).



## Selenium

Industry-wide best practice in test automation is to look for the possibility of Selenium implementation in web-based testing projects. Selenium has been evolved over a period of time since 2004, and it is available in four different types at the moment in 2015: Selenium IDE, Selenium Core, Selenium RC and Selenium Grid. Selenium IDE (integrated development environment) is available as an add-in for Firefox. Selenium Grid is used for NFT (non-functional tests) like performance tests and concurrent tests. Selenium Webdriver is widely used across the platforms as an open-source test automation tool, and organizations build automation framework on top of Selenium to make use of the tool for testing projects.

Selenium 3.0 released on 13 Oct 2016 as per the updates at:

<https://seleniumhq.wordpress.com/2016/10/13/selenium-3-0-out-now/>

### Selenium versions and their capabilities

	Selenium 1	Selenium 2	Selenium 3	Selenium 4/ Selenium 5
Release date	2004	2011	13Oct '16	Future release*
Tools	Selenium IDE,Selenium Core,Selenium RC,Selenium Server	Selenium Webdriver,Selenium IDE,Selenium RC,Selenium Core,Selenium Server,Appium[for mobile]	Selenium Webdriver,Selenium IDE,Appium[for mobile]	Selenium W3C,Appium[for mobile]
Advantages	Jason Huggins – Project Manager for Selenium Core	Webdriver introduced and bind tight with underlying web browser;Object oriented	Beginning of W3C compliance protocol	W3C webdriver server protocol is used

	Table-based runner	Firefox is connected with the help of Firefox extensions	W3C compliant webdriver server: gecko driver is used for Firefox	Internationaly agreed standards are followed and gecko driver will be used
	HTML Table-Selenium IDE	Internet Explorer's COM object collection used to communicate for tests	Selenium Core removed	Selenium Webdriver may be planned to be removed
	Three columns	Chrome,Opera extended to used as part of webdriver tests		Safari extended to used as part of webdriver tests
	<b>Selenium Core</b> designed by Jason for time and expenses system	Introduced Jason Wire Protocol representing commands for Firefox driver		Improved Jason Wire Protocol targeted in Selenium 4
	Since there were defects from Internet Explorer,Firefox, Opera for the Javascript application (time and expenses system), he decided to run the tests in javascript and drive the application	Selenium Core project integrated with webdriver project; hence, it has become <b>Selenium Webdriver</b>		Selenium 5: Target to enable html5 features,user facing APIs,friendly locators,more focus of webdriver API towards mobile and improved protocol
Disadvantages	Incredibly difficult to manage table-based scripting hence Selenium RC introduced	Challenge-not possible to analyze source code of IE and other browsers		

	<b>Selenium RC</b> scripts written in multiple languages and communicate to Selenium Server, and this server has internal communication(core protocol) to Selenium Core javascripts	Every browser follows different wire protocol-challenge for webdriver		
	Scripts running in javascript sandbox is a real challenge hence integrated with webdriver in next version(in Selenium 2.0)	Browsers have become more complicated in every version hence Selenium decided to use W3C for next version(in Selenium 3.0)		
Life span	<b>Selenium Core:</b> Limited [not used in Selenium 3.0 onwards]	<b>Selenium Webdriver:</b> Guaranteed [but no plans to use from Selenium 4.0 onwards]*as per the update from Simon on Sept'16 <b>Selenium IDE:</b> Guaranteed <b>Selenium RC:</b> Limited [since Selenium Core is removed in Selenium 3 onwards, scripts are to be migrated to Selenium Webdriver] <b>Appium:</b> Guaranteed[not yet migrated to Selenium 3.0 as	<b>Selenium Webdriver:</b> Limited [there are plans to remove from Selenium 4 onwards as per the web discussions-yet to be confirmed]	

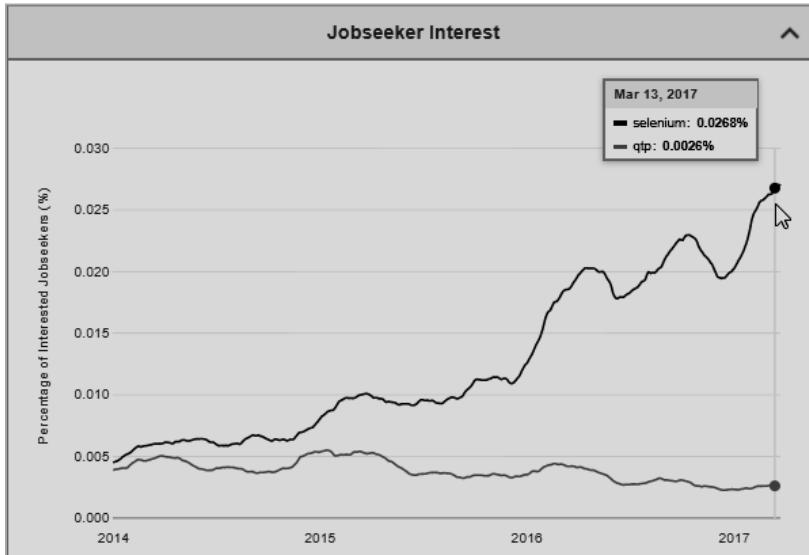
		on Sept2016]		
--	--	--------------	--	--

\*Features/functionalities of future releases may change and requesting readers to watch seleniumhq.org

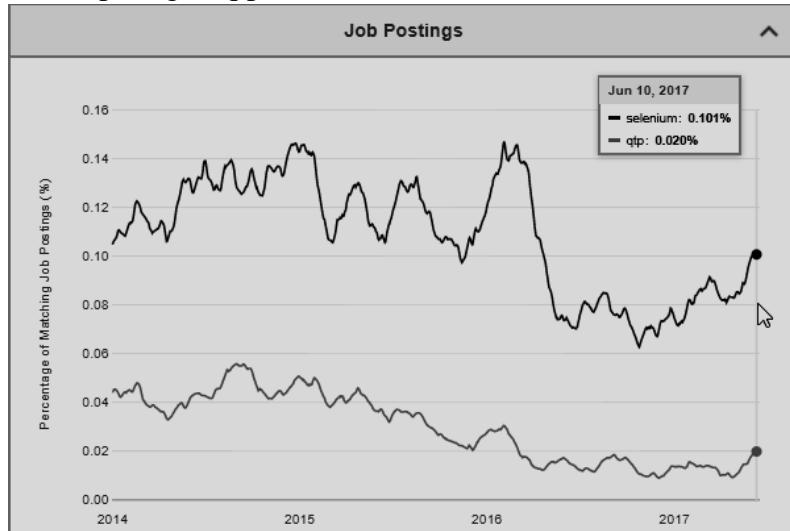
This table given above is derived on the basis of the inputs from  
[1] Simon Stewart (2016 May 26), Selenium 3 0 Retrieved from  
<https://www.youtube.com/watch?v=gS4ZqLpA9r4>  
[2] Simon Stewart (2016 September 14), Learn Using Selenium 3 0 Retrieved from <https://www.youtube.com/watch?v=bistojJPR98>  
[3] W3C (2016 September 14), World Wide Web Consortium from  
<https://www.w3.org/>  
[4] SeleniumHQ(2016 September 14), Selenium from  
<https://www.seleniumHQ.org/>  
[5] Simon Stewart (2017September 6), Selenium 3 0 Retrieved from  
<https://www.eventbrite.co.uk/e/asos-qa-meetup-the-future-of-selenium-by-simon-stewart-tickets-36864574936>

## Job Market Analysis

As per the job comparison between Selenium and qtp, there has been considerable amount of increment of job seekers' interest toward Selenium as of year 2017:



Similarly, job posts are increased rapidly for Selenium in recent years comparing to qtp:



Reference:

<https://www.indeed.com/jobtrends/q-selenium-q-qtp.html>

## **FAQs about automation tests using Selenium tool**

**Can Selenium be used for windows applications(example: automating desktop applications such as calculator etc.)?**

Appium has been designed to support desktop/windows applications. Refer:

<https://github.com/appium/appium-desktop/releases/>

Selenium is primarily used for web browser automation testing; Appium is used for mobile android,windows and iOS applications; In order to automate windows applications, following the open-source automation tools have been referred: 'Winium','Twin','White' and 'WinAppDriver'.

Winium:

<https://github.com/2gis/Winium>

WinAppDriver:

<https://github.com/Microsoft/WinAppDriver>

White:

<https://github.com/teststack/white>

Twin:

<https://github.com/JamesMGreene/twin>

## **How Firefox-based tests are executed in Selenium 3.x onwards?**

Gecko driver needed for Firefox tests when Selenium version 3.x are above has been used in framework(whereas there is no need of driver for Firefox tests in earlier versions). More details and configuration are provided in Part 2 of the book.

## **Which is the best tool alternative to Selenium?**

Protractor:

If the application is designed using AngularJS,it is most recommended to use Protractor end-to-end test framework. Refer the slides of protractor at:

<http://ramonvictor.github.io/protractor/slides/>

Tosca:

It is another automation framework using model-based test technique and risk-based test technique. Further details are furnished at:

<http://toolsqa.com/tosca/tosca-test-automation-suite/>

### **Author's Perspective**

This section shares the work experience, talks and useful blogs of author in relation to test automation to benefit readers on their job interviews.

### **Selenium vs QTP – which is better to learn for jobs?**

Both are mutually competent enough, and I would foresee more of Selenium in next ten years and less of QTP!

But when you go for interview, it doesn't mean which is the future, but which are in the market now!

So look experience on both!

Moreover, you should be good candidate to migrate scripts from QTP to Selenium, if required!

### **Which technology is better to learn – Hadoop or Selenium?**

Both are equally renowned but Selenium holds a strong future for the next ten years in the industry to become the most wanted tool!

Example: Sales force widely used CRM tool and Selenium automation, best supports such testing!

### **How is Selenium used in agile projects?**

Inspire automation is the target for agile projects, and Selenium is one of the most targeted tools in agile projects!

1. Framework constructed using Selenium during the initial sprint.
2. Selenium Page Objects Model used mostly in advanced disciplines.
3. Sprint planning consists of tasks related to Selenium scripting.
4. Sprint sizing evaluates automation efforts.
5. Retrospective discussion on what went wrong in automation scripts to improve in next sprint.
6. Cucumber-based feature files support Selenium scripts.

## **How do I get text for a read-only input field box using Selenium Webdriver?**

If it is a javascript-based framework using Selenium Webdriver IO, then

```
this.client  
.getText(selector);
```

Note: Selector is the object pass through locators.

If it is a Java-based Seleniumframework, then

```
driver.findElement(by.xpath(selector)).getAttribute("Value");
```

## **Why do organizations prefer to use Selenium for automated testing?**

The three primary reasons are:stability,cost and resources.

It's one of the few stable tools from open sources and also free, and majorityof engineers know Selenium in the market to employ!

## **How does the software QC and QA process work at your SaaS company?**

It works with the help of cloud tools such as JIRA for continuous integration,Jenkins for continuous deployment and open sources as such as Selenium for automation testing of SaaS products.

## **How do I explain about Jenkins in a Selenium project in an interview?**

Single line of answer: In CI/CD of agile projects,Selenium acts as tool for continuous testing and Jenkins acts as continuous integration tool for builds!

## **Do we have any tools or something to improve the rendering time of the homepage in Selenium automation?**

Simply use explicit wait or fluent wait to increase rendering time rather than using simple implicit fixed waits.

## **How do I use tab button functionality using Selenium?**

keys.TAB does it for you!

It is as simple as

1. WebElement webElement =  
driver.findElement(By.xpath("")); //You can use xpath, ID or name  
whatever you like in object
2. webElement.sendKeys(Keys.TAB);
3. webElement.sendKeys(Keys.ENTER);

## **What is the most reliable locator in Selenium Webdriver?**

Most unique xpath is most reliable!

If following types are still failing, I would suggest you to ask developers to add new object (tag) with description what you want (may be you can ask for qa-selector field like ID)!

Types:

- ID (Unique Locator):
- Example:driver.findElement(By.id("textbox"));
- Class
- Name
- Value
- Text (Link text, Partial Link text)
- XPATH (Relative X path)
- CSS (Fastest Locator)
- DOM

If none of the above works for you at times, use qa-selector as discussed in your script across!

so it will be like,

```
driver.findElement(By.qa-selector("textbox"));
```

## **How do I become a good tester in Selenium?**

A tester can be a good tester only when he/she finds most of the defects and helps the entire team with automation capability!

If you want to achieve that stage using Selenium frameworks, you need few more tool knowledge as I recommend:

If you are interested in this approach, it is the right time to choose right programming language for your career which is not limited to,

Java

.Net

PHP

Python

TCL/TK etc

Ruby javascript

Once you decide, for example you want to be a software developer in tests in Java (known as automation testers), you can start reading some great examples from part2 of this book.

### **Which language is demand now for Selenium: Python or Java?**

Demand is based on the job available in the market, and as per the current trend, Java is widely used and the most known with solved open-source solutions.

There is a sharp demand for javascript-based Selenium across companies such as CTS, Wipro and TCS.

Product companies wanted to develop Selenium scripts using Python and, similarly, some insurance companies are targeting Ruby.

Most media companies wanted C#-based Selenium framework when their application developed using GitHub projects.

If you really wanted to learn all the programming languages, it's an unfavorable for juniors because you need to figure out the flavors of them by some good books such as the part 2 of this book.

### **What is the scope of Cucumber automation?**

There is a good growth expected for Given, When and Then-based Cucumber Gherkin scripting since Dan North introduced it and, at present, most of the companies are adapting this approach as part of their agile projects!

### **What is the golden rule or ratio for automated testing in software development?**

**Golden Rule 1:**

Return on investment is always expected post 6 testing cycles!

**Golden Rule 2:**

It is not possible or worth to target 100% automation.

**Golden Rule 3:**

Once developed, automated scripts are not life proof (they have to be maintained)!

**Golden Rule 4:**

If automation framework is not documented in detail, it is going to be doomed in short time (no matter how strong the tool is!).

**Is Selenium Webdriver completely open source? How does the project get funded and stay active?**

It is one of the best questions of the year I would rate!

Reason being, Selenium is a free tool but the entire pack of test framework constructed using Selenium needs not be complete open source!

For example, framework can be based on Selenium, but if you have concurrent tests planned, and if you use SauceLabs for that, it is licensed!

In this way, project fund can be estimated only after the careful perusal of the complete test scope and automation test strategy!

**What could be the reply when my manager says "manual testing is superior to development because anybody can code"?**

You need to take the advantage of your expertise of coding in development and do not deny the opportunity at the same time; hence, you should propose combined offer of being software developer in tests (automation tester) to automate his/her applications then you will be treated unique and really valued as superior comparing to other developers and manual testers.

**What is the use of NUnit?**

NUnit acts as a test runner for the .Net-based projects (specifically the automation frameworks built in using Visual Studio or Team Foundation Servers).

Hence, it talks to VS and in turn VS gets the build running by using test automation engines calling application in the background.

NUnit has the capability of VS within their settings and you can learn further from .Net projects illustrated in this book.

### **How do I start working smartly as a QA engineer?**

You need charisma to be a QA engineer!

Following characteristics are chiefly important while growing from QA to test lead:

- Accuracy
- Fault finding
- Dedication
- Documentation skills
- Tools knowledge
- Reporting skills
- Defect life cycle knowledge
- Focus on test automation especially Selenium Webdriver

### **What do some senior-level QA software engineers advise to entry-level QA engineer?**

I have struggled through hard times as QA; hence, I think my voice would fit as answer here:

Recommended Path:Test Analyst > Module Lead >Test Lead > Test Manager > Head of QA

It depends upon how strong you are in terms of the skillset,tools and technologies you work to accumulate knowledge with relevant work experience over a period of time.

Example1: You can be a functional tester concentrating on telecommunication domain with strong domain knowledge such as LTE with more than 10 years of experience.

### **Automated testing: Is there any worldwide accepted certification for Selenium?**

There are Selenium certifications available for testers within India:  
<http://www.vskills.in/certification/Certified-Selenium-Professional>  
But it is not suggested(to opt just a tool certification) as you need a professional certification from Oracle for JAVA or from Microsoft

Visual Studio or Microsoft .Net certificate as evidence of your expertise in programming languages.

Reason:

If Selenium gets out of date in future, you can still use these certifications for the latest tool you use!

## Cucumber BDD (Behavior-driven Development)

*Programming language used in this section: Java Programming*

*More code examples of this section are available from github at:  
<https://github.com/narayananpalani/testautomation/tree/master/SACucumberTestProject>*

Cucumber (e.g. SpecFlow) is widely used automation framework to generate the test scenarios as feature files which are directly influenced by testable requirements.

### Example cucumber project

#### Requirement

Zoo website should provide the capability to check for animals' availability.

#### Feature file

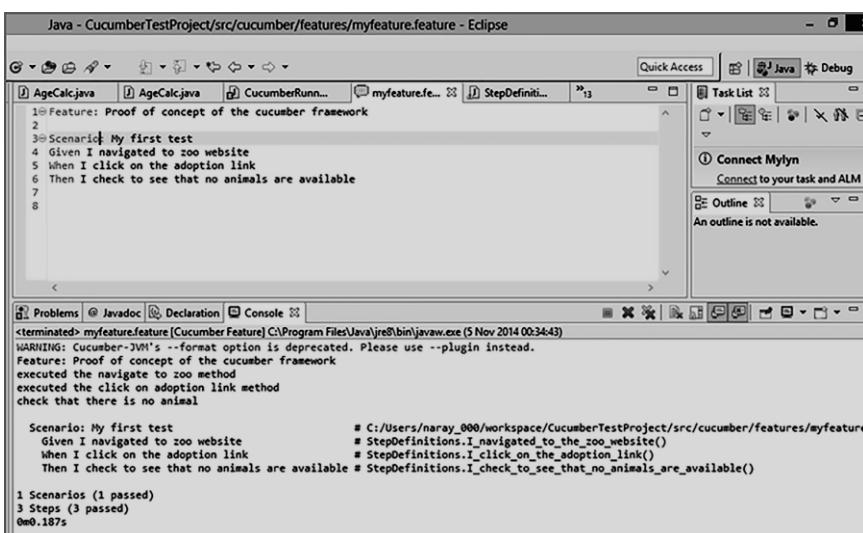
Feature: Proof of concept of the cucumber framework

Scenario: My first test

Given I navigated to zoo website

When I click on the link

Then I check to see that no animals are available



```
Java - CucumberTestProject/src/cucumber/features/myfeature.feature - Eclipse
Java Debug
Task List
Connect Mylyn
Outline
An outline is not available.

1 Feature: Proof of concept of the cucumber framework
2
3 Scenario: My first test
4 Given I navigated to zoo website
5 When I click on the adoption link
6 Then I check to see that no animals are available
7
8

<terminated> myfeature.feature [Cucumber Feature] C:\Program Files\Java\jre8\bin\javaw.exe (5 Nov 2014 0034:43)
WARNING: Cucumber-JVM's --format option is deprecated. Please use --plugin instead.
Feature: Proof of concept of the cucumber framework
executed the navigate to zoo method
executed the click on adoption link method
check that there is no animal

Scenario: My first test
  Given I navigated to zoo website          # C:/Users/naray_000/workspace/CucumberTestProject/src/cucumber/features/myfeature
  When I click on the adoption link        # StepDefinitions.I_navigated_to_the_zoo_website()
  Then I check to see that no animals are available # StepDefinitions.I_click_on_the_adoption_link()
                                                 # StepDefinitions.I_check_to_see_that_no_animals_are_available()

1 Scenarios (1 passed)
3 Steps (3 passed)
0m0.187s
```

## Step definitions

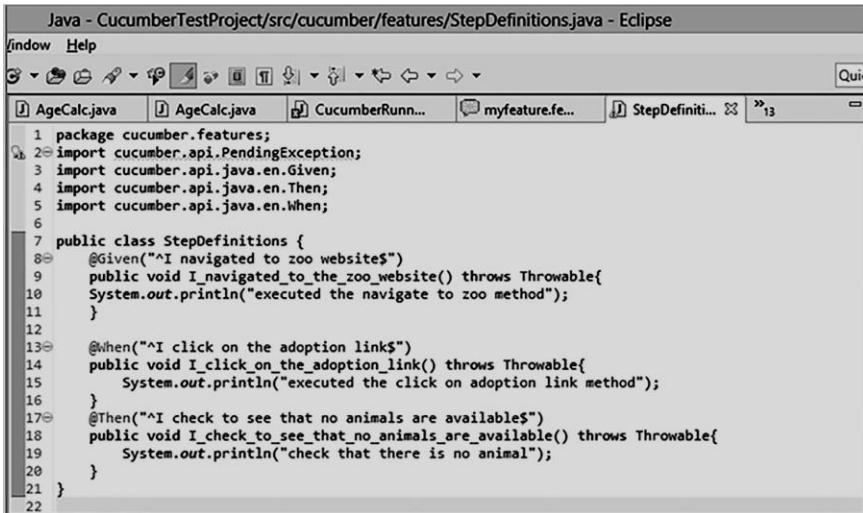
Each feature file, such as Given, When and Then, contains a step definition which is as follows:

```
package cucumber.features;
import cucumber.api.PendingException;
import cucumber.api.java.en.Given;
import cucumber.api.java.en.Then;
import cucumber.api.java.en.When;

public class StepDefinitions {
    @Given("^I navigated to zoo website$")
    public void I_navigated_to_the_zoo_website() throws Throwable{
        System.out.println("executed the navigate to zoo method");
    }

    @When("^I click on the adoption link$")
    public void I_click_on_the_adoption_link() throws Throwable{
        System.out.println("executed the click on adoption link method");
    }

    @Then("^I check to see that no animals are available$")
    public void I_check_to_see_that_no_animals_are_available()
        throws Throwable{
        System.out.println("check that there is no animal");
    }
}
```



The screenshot shows the Eclipse IDE interface with the title bar "Java - CucumberTestProject/src/cucumber/features/StepDefinitions.java - Eclipse". The code editor displays Java code for step definitions:

```
1 package cucumber.features;
2 import cucumber.api.PendingException;
3 import cucumber.api.java.en.Given;
4 import cucumber.api.java.en.Then;
5 import cucumber.api.java.en.When;
6
7 public class StepDefinitions {
8     @Given("^I navigated to zoo website$")
9     public void I_navigated_to_the_zoo_website() throws Throwable{
10         System.out.println("executed the navigate to zoo method");
11     }
12
13     @When("^I click on the adoption link$")
14     public void I_click_on_the_adoption_link() throws Throwable{
15         System.out.println("executed the click on adoption link method");
16     }
17     @Then("^I check to see that no animals are available$")
18     public void I_check_to_see_that_no_animals_are_available() throws Throwable{
19         System.out.println("check that there is no animal");
20     }
21 }
```

## Cucumber runner

This component aids in running the associated feature file with its step definition:

```
package cucumber;
import org.junit.runner.RunWith;
import cucumber.api.junit.*;
@RunWith(Cucumber.class)
@Cucumber.Option(
format = {"pretty", "json:target/cucumber.json"}
features = {"src/cucumber/"}
)
public class CucumberRunner {
```

```

Java - CucumberTestProject/src/cucumber/CucumberRunne
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer AgeCalc.java AgeCalc.java CucumberRun... myfeatu
CucumberTestProject
src
  cucumber
    CucumberRunner.java
  cucumber.features
    StepDefinitions.java
    myfeature.feature
JRE System Library [JavaSE-1.8]
Referenced Libraries
TestProject
  1 package cucumber;
  2 import org.junit.runner.RunWith;
  3 import cucumber.api.junit.*;
  4 @RunWith(Cucumber.class)
  5 @Cucumber.Option(
  6   format = {"pretty","json:target/cucumber.json"},
  7   features = {"src/cucumber/"}
  8 )
  9 public class CucumberRunner {
 10
 11 }
 12

```

For further reading and practice, download the projects at  
<https://github.com/narayananpalani/testautomation>

## How to Write First BDD Scripts in Cucumber,Selenium Using Java?

### Feature file using Cucumber

```

ExcelDataCh... SuiteOneBas... Param.prop... LaunchChrom... TestRun.java "bbcradio.f...
BBCRadioTestProject
src
  cucumber
    TestRun.java
    cucumber.features
      BBCradio.java
      BBCradio.feature
JRE System Library [JavaSE-1.8]
Referenced Libraries
BrowseDrivers
lib
CucumberTestProject
DataDriverFramework
src
  dataValidationPack
    ExcelDataCheckTest.java
  pageobjects
    LoginPageObjects.java
  1 Feature: Radio Nav Drawers
  2 As A user
  3 I WANT to see more content associated with navigation links
  4 So THAT I can easily find what I am looking for
  5 Background:
  6 Given BBC Radio is open in ChromeDriver
  7
  8 Scenario Outline: Opening and closing the drawers
  9 Given I can see the radio nav
 10 When I select <section> in the radio nav
 11 Then the <section> drawer is open
 12 When I select <section> in the radio nav again
 13 Then the <section> drawer is closed
 14 Then the <section> drawer is closed
 15
 16 Examples: of sections
 17 | section |
 18 | Stations |
 19 | Categories|
 20 | Schedules |

```

### Test run configuration for Cucumber

```

ExcelDataCh... SuiteOneBas... Param.prop... LaunchChrom... TestRun.java "bbcradio.f...
  1 package cucumber;
  2 //import cucumber.api.CucumberOptions;
  3 import org.junit.runner.RunWith;
  4 @RunWith(Cucumber.class)
  5 @Cucumber.Options(format = {"pretty","html:target/cucumber-html-report", "json:target/cucumber-report.json"},
  6   features = {"src/cucumber/testlandrover.feature"})
  7
  8
 10 public class TestRun {
 11
 12 }

```

# Step definitions in Java

Code:

```
package cucumber.features;
import java.io.File;
import java.io.IOException;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.By;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
//import org.openqa.selenium.android.AndroidDriver;
import org.openqa.selenium.chrome.ChromeDriver;
//import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.ie.InternetExplorerDriver;
import org.openqa.selenium.interactions.Actions;
//import org.openqa.selenium.iphone.IPhoneDriver;
import org.openqa.selenium.remote.DesiredCapabilities;

//import org.openqa.selenium.safari.SafariDriver;
import java.util.concurrent.TimeUnit;
//import com.sun.jna.platform.FileUtils;
//import cucumber.api.java.After;
import cucumber.api.java.Before;
import cucumber.api.java.en.Given;
import cucumber.api.java.en.Then;
```

```

import cucumber.api.java.en.When;
//import cucumber.api.java.en.And;
//import cucumber.runtime.PendingException;
public class bbcradio {
public WebDriver driver;
@Before
public void setUp() {
System.out.println("*****");
System.out.println("launching Chrome browser");
//System.setProperty("webdriver.ie.driver",
"E://AutomationProject//Training//LandroverTestProject//Browser
Drivers//IEDriverServer.exe");
System.setProperty("webdriver.chrome.driver",
"E:/AutomationProject/TrainingBBCRadioTestProject/BrowserDri
vers/ChromeDriver.exe");
}

@Given("^BBC Radio is open in ChromeDriver$")
public void BBC_Radio_is_open_in_ChromeDriver()
{
driver=new ChromeDriver();
driver.get("http://www.bbc.co.uk/radio");
}

@Given("^I can see the radio nav$")
public void I_can_see_the_radio_nav()
{
driver.findElement(By.className("radionav__stations-link
radionav-dropdown")).isEnabled();
driver.findElement(By.className("radionav__categories-link
radionav-dropdown")).isEnabled();
driver.findElement(By.className("radionav__schedules-link
radionav-dropdown")).isEnabled();
}

@Given("^When I select \"([^\"]*)\" in the radio nav$")

```

```
public void When_I_select_section_in_the_radio_nav(String Stations, String Categories, String Schedules)
{
    driver.findElement(By.linkText(Stations)).click();
    driver.findElement(By.linkText(Categories)).click();
    driver.findElement(By.linkText(Schedules)).click();
}
```

## Automation Framework Using Selenium Webdriver with DotNet

*Programming language used in this section: .Net Programming*

*More code examples of this section are available from github at:  
[https://github.com/narayananpalani/testautomation/tree/master/008AutomationProjects\\_Training/AutomationProject/Training/SVSF](https://github.com/narayananpalani/testautomation/tree/master/008AutomationProjects_Training/AutomationProject/Training/SVSF)*

DotNet-based Selenium Webdriver projects using C++ Programming are the most wanted skillset in IT testing domain. Especially, integrating the project from feature file (as scenarios), step definitions, Selenium objects and functions and then building the project to deploy and run in tools, such as NUnit/Gallio/MSTest, or alternatively scheduling it through Jenkins is something reputed as end-to-end automation implementation possibility at the moment in the industry.

Step-by-step approach!

### **Step1: Install visual studio**

Navigate to any best solution of Visual Studio (free version preferably for training) at:

<https://www.visualstudio.com/products/free-developer-offers-vs.aspx>

### **Step2:Install and configure SpecFlow**

First, select tools then 'Extensions and Updates' in Visual Studio and search online for SpecFlow (whatever the relevant latest version); Select the SpecFlow for Visual Studio 2013 (any latest version) extension to download and install. Restart Visual Studio to enable the extension (perform all necessary configurations required for BDD projects as per the SpecFlow chapter).

### **Step3: Selenium Webdriver for DotNet**

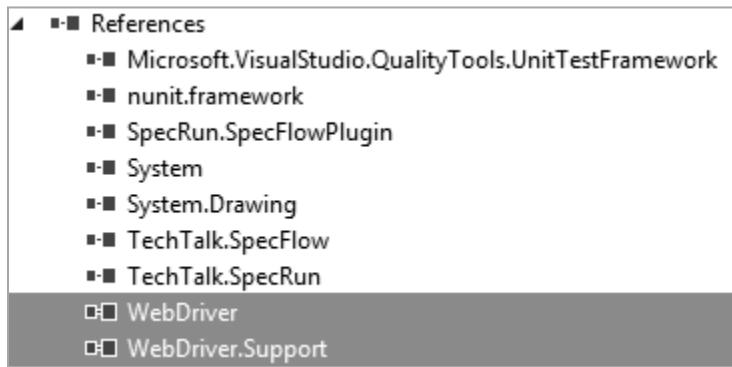
Install Selenium Webdriver through Package Manager Console following the steps given below:

Select tools and then Package Manager Console  
PM>

Command: PM> Install-Package Selenium.WebDriver

Make sure that the following references are updated in the project.

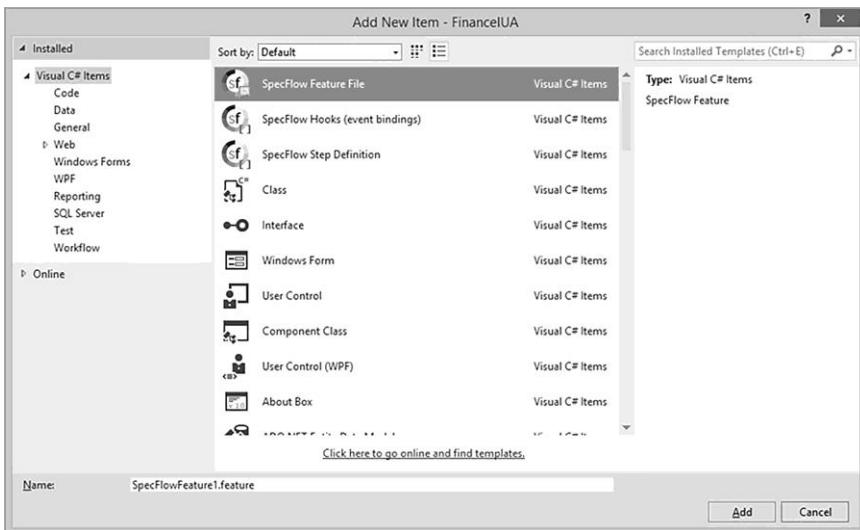
Selenium WebDriver API .NET Bindings and Selenium WebDriver .NET Bindings support classes:



Reference: <https://www.nuget.org/packages/Selenium.WebDriver>

#### Step4:Feature file

Right click on the project, select Add Item and choose SpecFlow Feature File and Add to open the new feature file:



It would have been added, and .cs file would have been associated by itself:



Write a scenario as per the example given below:

Feature: Currency Exchange Rate in ABC Bank

*In order to avoid silly mistakes*

*As a anonymous user*

*I want to check average INR exchange rate of banks in London*

### @RegressionTests

Scenario: average INR exchange rate of banks in London

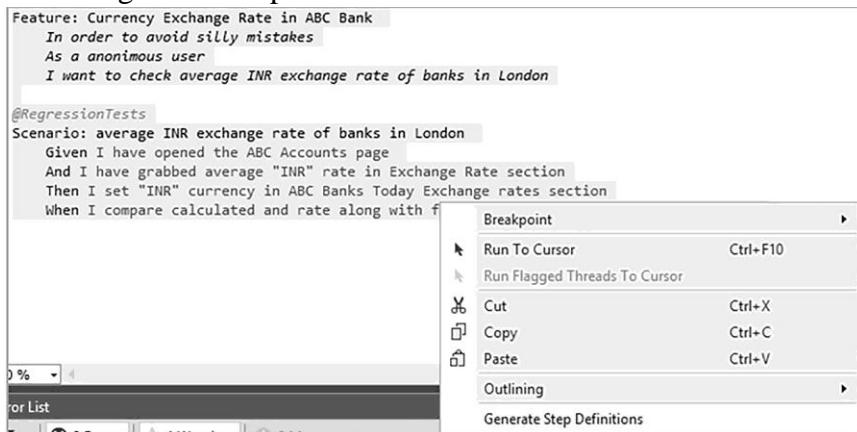
Given I have opened the ABC Accounts page

And I have grabbed average "INR" rate in Exchange Rate section

Then I set "INR" currency in ABC Banks Today Exchange rates section

When I compare calculated and rate along with fee values the result should be correct

Click on generate step definitions:



Copy to clipboard and paste them to different .cs file to edit further:

```
[Given(@"I have opened the ABC Accounts page")]
public void GivenIHaveOpenedTheABCAccountsPage()
```

```
{  
ScenarioContext.Current.Pending();  
}  
  
[Given(@ "I have grabbed average """(.*)"" rate in Exchange Rate  
section")]  
public void  
GivenIHaveGrabbedAverageRateInExchangeRateSection(string  
p0)  
{  
ScenarioContext.Current.Pending();  
}  
  
[Then(@ "I set """(.*)"" currency in ABC Banks Today Exchange  
rates section")]  
public void  
ThenISetCurrencyInABC Banks Today Exchange Rates Section(strin  
g p0)  
{  
ScenarioContext.Current.Pending();  
}  
  
[When(@ "I compare calculated and rate along with fee values the  
result should be correct")]  
public void  
WhenICompareCalculatedAndRateAlongWithFeeValuesTheResul  
tShouldBeCorrect()  
{  
ScenarioContext.Current.Pending();  
}
```

### **Another example:**

Develop the feature file with scenarios in Gherkin format then save it as a .feature file.

\*Refer Gherkin/SpecFlow sections

Example:

Feature: Currency Exchange Rate in ABC Bank

*In order to avoid silly mistakes*

*As a anonymous user*

*I want to check average USD exchange rate of banks in London*

*@RegressionTests*

Scenario: average USD exchange rate of banks in London

Given I have opened the ABC Finance page

And I have grabbed average "USD" rate in Average Rate section

Then I set "USD" currency in ABC Banks rates section

When I compare calculated and grabbed values the result should be equal to "4" decimal

## **Step5: Step definitions**

In the .cs file associated to your feature file, jot down the line-based script.

Right click on the feature file and select generate step definitions' and copy that to clipboard and paste to this .cs file and then edit the .cs files with valid Selenium functions.

Example:

```
using System;
using System.Collections;
using FinanceIUA.PageObjects;
using NUnit.Framework;
using OpenQA.Selenium;
//using OpenQA.Selenium.Firefox;
using OpenQA.Selenium.Chrome;
using TechTalk.SpecFlow;
```

----

----

```
[BeforeScenario()]
```

```
public void Setup()
```

```
{
```

```
//driver = new FirefoxDriver();
```

```
//This is the initiation of a chrome browser and the location where
the driver is:
```

```
driver = new ChromeDriver(@"E:\driver");
//driver = new ChromeDriver();
}
---
---

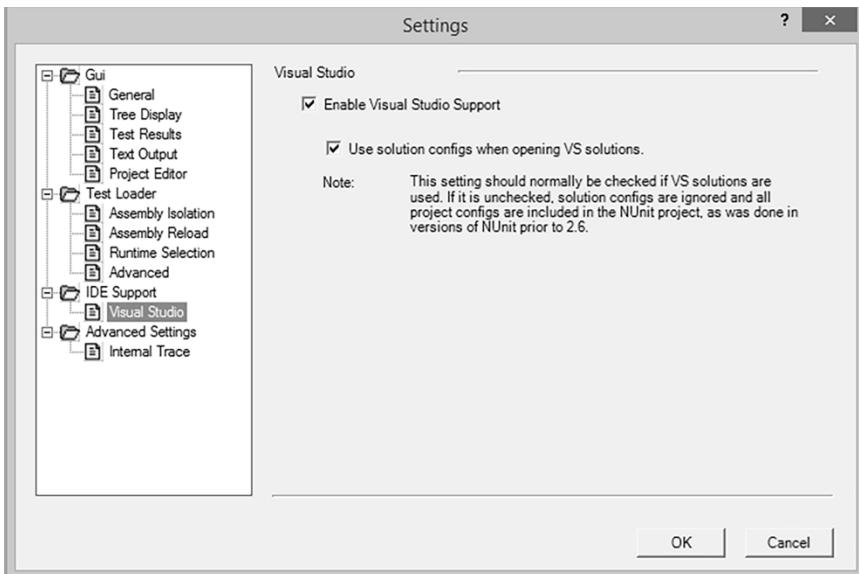
[Given(@"I have opened the ABC Finance page")]
public void GivenIHaveOpenedTheABCFinancePage()
{
    financePage = ABCFinancePage.NavigateTo(driver);
    driver.Manage().Timeouts().ImplicitlyWait(TimeSpan.FromSeconds(10));
    driver.Navigate().GoToUrl("http://www.abcbank.org/");
}
```

For further reading and practice, download the projects at  
<https://github.com/narayananpalani/testautomation>

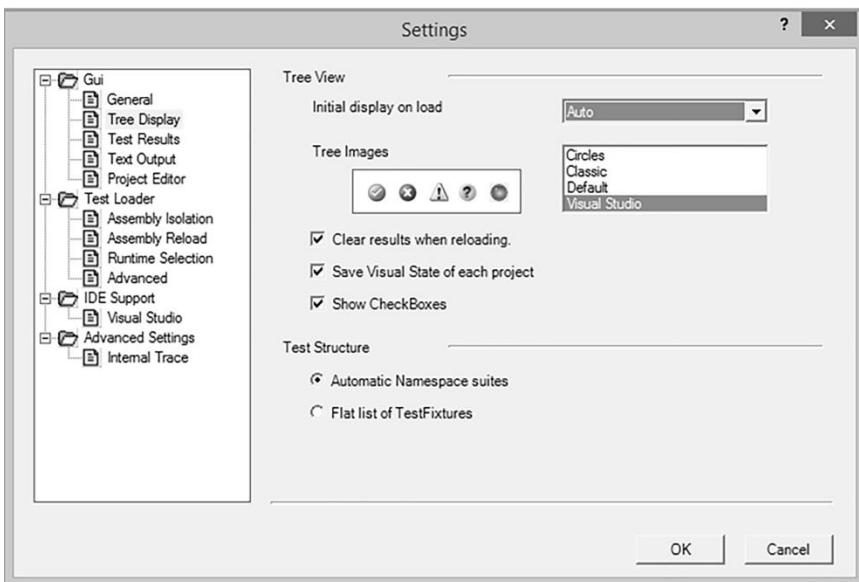
## **How to run the Visual Studio-based Selenium project in NUnit?**

NUnit gives the flexibility to run the Visual Studio-based projects with minor settings update which has been summarized below:

1. Install NUnit.
2. In Tools>Settings> select checkbox for Enable Visual Studio Support and use solution configs while opening VS solutions.



3. In tree display, select the checkbox for 'Show checkboxes' which will allow user to select the respective tests to run during test execution.



Once the tests are executed, click on Tools>Save Results as XML in order to save the test results.

## **ReportUnit**

This tool helps in developing an html-based report out of xml report of NUnit.

Download URL:<http://reportunit.relevantcodes.com/>

Once the tool has been downloaded, run the tool from command line with the help of following syntax:

reportunit [input-folder-path] [output-folder-path]

Assuming the xml files of test results are stored at C drive's result folder, you require the ReportUnit reports at reports folder; this can be performed following steps given below:

1. Install Reportunit
2. Go to command line where Reportunit is placed
3. Enter the commands:

reportunit "C:\Results" "C:\Reports"

The abovementioned steps enable to get the html-based reports within the folder reports-which can be shared with the stakeholders in html format.

## **How to set up the NUnit executable project in Jenkins?**

The steps are as follows:

1. Install Jenkins and open the URL localhost:8080
2. Once Jenkins is opened, click on Add New Item (new project)
3. Click Configure>Build>Add a build step>Execute Windows batch command

You can run the Eclipse project from command line given below:

[PATH]\bin\nunit-console.exe

[Path]\Selenium.Tests.dll /xml=nunit-result.xml

If the project is Visual Studio/TFS-based Sln project:

[PATH]\bin\nunit-console.exe

[Path]\SeleniumTests.sln /xml=nunit-result.xml

So add the respective command line to run the NUnit project command line instructions from Jenkin itself.

4. Post Build Actions section> Test Report XMLs >Type 'nunit-result.xml'
5. Post Build Actions section> Add your email id

## Selenium Webdriver Tests in Python

*Programming language used in this section: Python Programming*

In order to write automation scripts using python, tester should understand the ease of use and smooth test execution using python code base.

Sample Python Script to launch a web page in Firefox using Selenium:

```
from selenium.webdriver import Firefox  
driver = Firefox()  
driver.get('https://www.google.com/')
```

driver.quit()

### Prerequisites for python-based Selenium scripting

Click DOWNLOAD ZIP on the links of GITHUB below:

Entire slides (including the session slides shown today which are md extensions):

<https://github.com/davehunt/selenium-python-workshop>

Instructions:

<https://gist.github.com/davehunt/1f04963ca60c00b4490f04edf432f40f>

Login page used for sample scripts:

<https://the-internet.herokuapp.com/login>

To read the md files (slides available in GitHub links are md files; use the tool below to read them):

<http://markdownpad.com/>

Note:

Better use Python v2.x only as python 3.x holds different scripting standards.

### Python installation steps

Install python using Command Prompt Admin

**python-2.7.13.msi**

```
Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\>cd F:\Selenium using Python\Installable
C:\>f:
F:\Selenium using Python\Installable>python -m pip install selenium
F:\Selenium using Python\Installable>
```

How to get into python prompt from command prompt?

Type 'python'

```
Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\>cd F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions
C:\>f:
F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions>python
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> -
```

Install PyPOM - python:

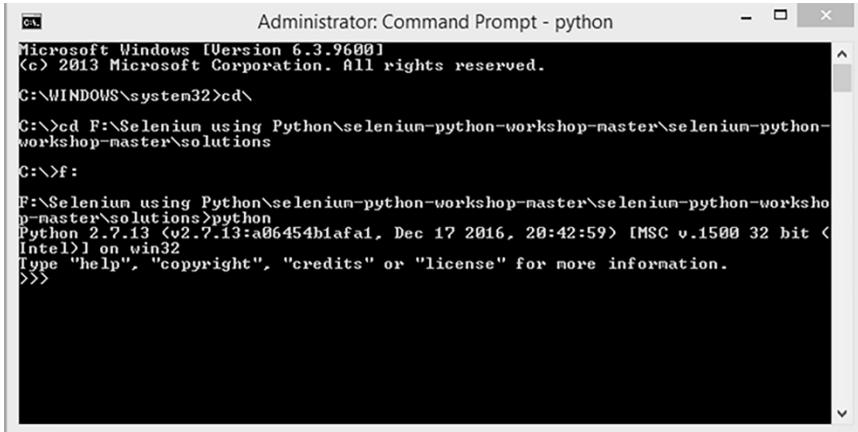
**pip install PyPOM**

```
Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\>cd F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions
C:\>f:
F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions>python
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> pip install PyPOM
```

Command Prompt shows right version when entered PYTHON:

Type 'python' to see version of the tool being installed



```
Administrator: Command Prompt - python
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

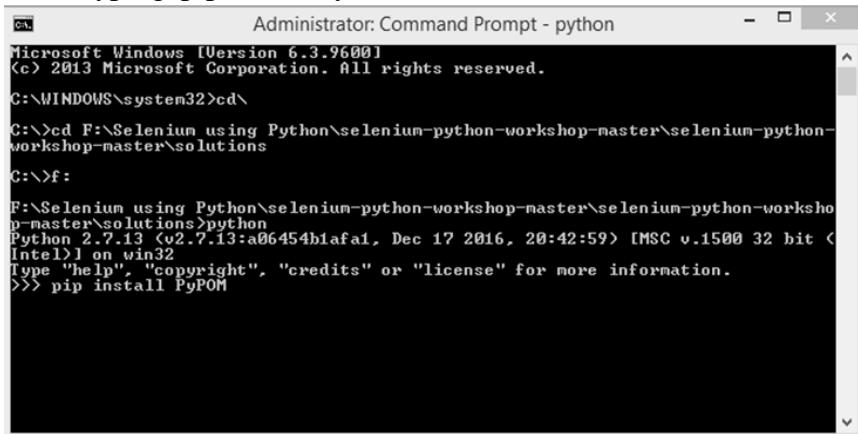
C:\>cd F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions

C:\>>f:

F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions>python
Python 2.7.13 (v2.7.13:a06454b1afaf, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

## Install PyPOM:

While typing 'pip install PyPOM' make sure its entered after '>>>'



```
Administrator: Command Prompt - python
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

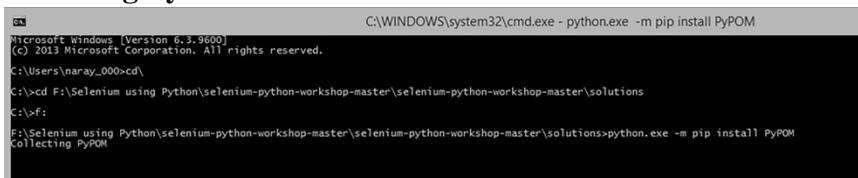
C:\>cd F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions

C:\>>f:

F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions>python
Python 2.7.13 (v2.7.13:a06454b1afaf, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> pip install PyPOM
```

Installation work from Command Prompt(tested) – so run cmd prompt and install PyPOM:

Alternative method is to type '**python.exe -m pip install PyPOM**  
**Collecting PyPOM**'



```
C:\WINDOWS\system32\cmd.exe - python.exe - m pip install PyPOM
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\naray_000>cd\
C:\>cd F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions
C:\>>f:
F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions>python.exe - m pip install PyPOM
Collecting PyPOM
```

## Install Selenium:

```
python.exe -m pip install pytest-selenium
```

```
C:\WINDOWS\system32\cmd.exe

[gn] Collecting PyPOM
  Downloading PyPOM-1.1.1-py2.py3-none-any.whl
[gn] Collecting zope.component (from PyPOM)
  Downloading zope.component-4.3.0.tar.gz (35kB)
  100% |################################| 35kB 506KB/s
[gn] Collecting selenium (from PyPOM)
  Downloading selenium-3.0.2-py2.py3-none-any.whl (915kB)
  100% |################################| 912kB 461kB/s
[gn] Collecting python-dateutil (from PyPOM)
  Downloading python_interface-4.3.3-cp37-win32.whl (135kB)
  100% |################################| 143kB 1.2MB/s
[gn] Requirement already satisfied: six (from PyPOM) in c:\python27\lib\site-packages (from zope.component->PyPOM)
[gn] Collecting zope.event (from zope.component->PyPOM)
  Downloading zope.event-4.2.0-py2-none-any.whl
Installing collected packages: zope.interface, zope.event, zope.component, selenium, PyPOM
  Running setup.py install for zope.component ... done
Successfully installed PyPOM-1.1.1 selenium-3.0.2 zope.component-4.3.0 zope.event-4.2.0 zope.interface-4.3.3
F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions>python.exe -m pip install pytest-selenium
```

## pytest-xdist:

```
python.exe -m pip install pytest-xdist
```

tox:

```
python.exe -m pip install tox
```

```
[GN]
 0. Downloading appipkg-1.4-py2.py3-none-any.whl
Requirement already satisfied: setuptools in c:\python27\lib\site-packages (from pytest>=2.4.2->pytest-xdist)
Requirement already satisfied: colorama; sys.platform == "win32" in c:\python27\lib\site-packages (from pytest>=2.4.2->pytest-xdist)
Installing collected packages: appipkg, execnet, pytest-xdist
  Running setup.py install for pytest-xdist ... done
Successfully installed appipkg-1.4 execnet-1.4.1 pytest-xdist-1.15.0

P: Selenium using Python:selenium-python-workshop-master\selenium-python-workshop-master\solutions>python.exe -m pip install tox
Collecting tox
 0.000s
    Downloading tox-2.6.0-py2.py3-none-any.whl (42kB)
  100% ##### [██████████] | 51KB 295KB/s
Requirement already satisfied: py>=1.4.17 in c:\python27\lib\site-packages (from tox)
Collecting pluggy<1.0,>=0.3.0 (From tox)
 0.000s
    Downloading pluggy-0.4-0-py2.py3-none-any.whl
Collecting virtualenv>=11.1.2; python_version >= "3.2" (from tox)
 0.000s
    Downloading virtualenv-11.1.2-py2.py3-none-any.whl (1.3MB 1.3MB)
  100% ##### [██████████] | 1.3MB 279KB/s
Installing collected packages: pluggy, virtualenv, tox
```

## Cookie cutter:

```
python.exe -m pip install cookiecutter
```

## Software Automation Testing Secrets Revealed

```
C:\WINDOWS\system32\cmd.exe - python.exe -m pip install cookieduster
Collecting pluggy==0.4.0-py2.py3-none-any.whl
  Downloading virtualenv==15.1.0-py2.py3-none-any.whl (1.8MB)
    100% [=====] | 1.8MB 279K/s
Installing collected packages: pluggy, virtualenv, tox
Successfully installed pluggy-0.4.0 tox-2.6.0 virtualenv-15.1.0
F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions>python.exe -m pip install cookieduster
Collecting cookieduster
  Downloading cookieduster-1.5.1-py2.py3-none-any.whl (1.8MB)
    100% [=====] | 1.8MB 279K/s
Collecting binaryornot>=0.2.0 (from cookieduster)
  Downloading binaryornot-0.4.0-py2.py3-none-any.whl (1.9MB)
    100% [=====] | 1.9MB 279K/s
Collecting click>=5.0 (from cookieduster)
  Downloading click-6.7-py2.py3-none-any.whl (71KB)
    100% [=====] | 71KB 1.2MB/s
```

Move to python prompt from cmd by entering python:

```
C:\WINDOWS\system32\cmd.exe - python
Collecting MarkupSafe==0.23 (from Jinja2>=2.7->cookieduster)
  Downloading MarkupSafe-0.23.tar.gz
Collecting python-dateutil (<from arrow->) Jinja2-time>=0.1.0->cookieduster)
  Downloading python_dateutil-2.6.0-py2.py3-none-any.whl (194KB)
    100% [=====] | 194KB 279K/s
Collecting six>=1.5 (from python-dateutil->arrow->Jinja2-time>=0.1.0->cookieduster)
  Downloading six-1.10.0-py2.py3-none-any.whl
Installing collected packages: chardet, binaryornot, whichcraft, click, poyo, future, six, python-dateutil, arrow, MarkupSafe, Jinja2, Jinja2-time, Running setup.py install for chardet ... done
Running setup.py install for future ... done
Running setup.py install for arrow ... done
Running setup.py install for MarkupSafe ... done
Successfully installed MarkupSafe-0.23 chardet-2.3.0 click-6.7 cookiecutter-1.5.1 Future-0.16.0 Jinja2-2.9.5 Jinja2-time-2.6.0 six-1.10.0 whichcraft-0.4.0
F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions>python
Python 2.7.13 (v2.7.13:6bd0d0d32138, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Running first test:

enter file name after >>> as **020\_our\_first\_test.py** for example:

```
C:\WINDOWS\system32\cmd.exe - python
Collecting six>=1.5 (from python-dateutil->arrow->Jinja2-time>=0.1.0->cookiecutter)
  Downloading six-1.10.0-py2.py3-none-any.whl
Installing collected packages: chardet, binaryornot, whichcraft, click, poyo, future, six, python-dateutil, arrow, MarkupSafe, Jinja2, Jinja2-time, Running setup.py install for chardet ... done
Running setup.py install for future ... done
Running setup.py install for arrow ... done
Running setup.py install for MarkupSafe ... done
Successfully installed MarkupSafe-0.23 chardet-2.3.0 click-6.7 cookiecutter-1.5.1 Future-0.16.0 Jinja2-2.9.5 Jinja2-time-2.6.0 six-1.10.0 whichcraft-0.4.0
F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions>python
Python 2.7.13 (v2.7.13:6bd0d0d32138, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 020_our_first_test.py
File <stdin>, line 1
  020_our_first_test.py
^
SyntaxError: invalid syntax
>>> execfile('F:\Selenium using Python\selenium-python-workshop-master\selenium-python-workshop-master\solutions\020_our_first_test.py')
```

## Capture object from the web page

Right click on the object and click on Inspect Element:

The screenshot shows a web browser window with a login form. The form has fields for 'Username' and 'Password', and a 'Login' button. A right-click context menu is open over the 'Login' button, with the 'Inspect Element' option highlighted. The developer tools show the HTML structure of the page, focusing on the button element.

Use `$()` syntax in console to double check whether the object captured is right or wrong:

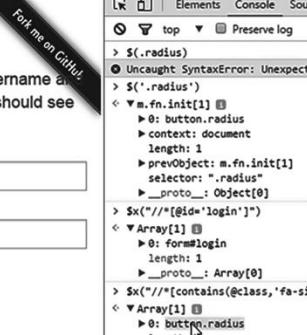
## Login Page

This is where you can log into the secure area. Enter `tomsmith` for the username and `SuperSecretPassword!` for the password. If the information is wrong you should see error messages.

Username

Password

 Login



## Firefox-based Selenium python test

If we wish to run a test in Firefox, sample script which works on Firefox is as follows:

```

from selenium.webdriver import Firefox
driver = Firefox()
driver.set_page_load_timeout(10)
driver.get('https://sameplewebsite')
driver.find_element_by_id('username').send_keys('XXXXXXXXXX')
driver.find_element_by_id('password').send_keys('XXXXXXXXXX')
driver.find_element_by_css_selector('#login button').click()
driver.get_screenshot_as_file("F:\\Selenium using
Python\\Screenshots\\loginpage.png")
driver.quit()

```

## Chrome-based Selenium python test

(download chrome driver

from <https://sites.google.com/a/chromium.org/chromedriver/downloads> and place it in python's folder):C:\Python27\chromedriver.exe

Script:

```

from selenium.webdriver import Chrome
driver = Chrome("C:\\Python27\\chromedriver.exe")
driver.set_page_load_timeout(10)
driver.get('https://sameplewebsite')

```

```
driver.find_element_by_id('username').send_keys('XXXXXXXXXX')
driver.find_element_by_id('password').send_keys('XXXXXXXXXX')
driver.find_element_by_css_selector('#login button').click()
driver.get_screenshot_as_file("F:\\Selenium using
Python\\Screenshots\\loginpage.png")
driver.quit()
```

If you really wish to learn how to run the tests without errors, watch the video suggested in the link given below:

<https://www.youtube.com/watch?v=zRrubJ13I8s>

## Selenium Test Automation Framework using Javascript

*Programming language used in this section: Java Script*

It is important to understand that the Selenium framework can be scripted using javascript which is different from the way Java-based scripts work!

The dissimilarities between Java and javascript-based scripting are explained as follows:

How Java-based Selenium scripts perform a web button click?

Example:

```
driver.find_element_by_css_selector('#login button').click()
```

How javascript-based Selenium scripts perform a web button click?

Example:

```
return driver.find_element_by_css_selector('#login button').click()
```

Refer the below website for available functions:

<http://webdriver.io/api.html>

### Feature file for the Javascript-based Selenium framework

@search

**Feature:** Searching for Selenium learning

As an internet user

In order to find out more about the Selenium books

I want to be able to search for information about the course of Selenium test automation

**Scenario:** Google search for Selenium books

When I perform search on Google website for "Selenium books"

Then I should see relevant results in Google Search Results

**Scenario:** Google search for course of Selenium test automation

When I perform search on Google website for "Selenium test automation testing training"

Then I should see relevant results in Google Search Results

### **Step definition for the javascript-based Selenium framework**

```
module.exports = function () {
this.When(/^I perform search on Google website for "([^\"]*)"$/, function (searchQuery) {
return helpers.loadPage('http://www.google.com').then(function() {
// Perform a google search using a function which is scripted in different file
return
page.googleSearch.performGoogleWebsiteSearch(searchQuery);
})
});
this.Then(/^I should see relevant results in Google Search Results$/, function () {
// driver wait to get the results for 20000 milliseconds
return driver.wait(until.elementsLocated(by.css('div.g')), 20000).then(function() {
// Use the object of search results css locator:
return driver.findElements(by.css('div.g'));
})
.then(function (elements) {
// verify this element has any children on itself
expect(elements.length).to.not.equal(0);
});
});
});
```

Now perform search in first-step definition needs a function to determine on how to perform a Google search on Google website!

### **Page object model within javascript-based Selenium framework**

This can be performed by introducing function within page object model:

```
module.exports = {
url: 'http://www.google.co.in',
elements: {
```

```
searchInput: by.name('q'),  
searchResultLink: by.css('div.g > h3 > a')  
,  
performGoogleWebsiteSearch: function (searchQuery) {  
var selector = page.googleSearch.elements.searchInput;  
// return a promise so the calling function knows the task has  
completed  
return driver.findElement(selector).sendKeys(searchQuery,  
Selenium.Key.ENTER);  
}  
};
```

Reference to the samples above:

<https://github.com/john-doherty/selenium-cucumber-js>

## Useful Git project on reusable step definitions

Most of the reusable, required and commonly used step definitions for the javascript-based Selenium framework have been maintained at:

<https://github.com/webdriverio/cucumber-boilerplate>

These step definitions can be referred at:

<https://github.com/webdriverio/cucumber-boilerplate/tree/master/src/steps>

Page object model-based functions can be referred at:

<https://github.com/webdriverio/cucumber-boilerplate/tree/master/src/support>

Aiming at making sense out of how the step definitions help your projects, try running some of the feature files at:

<https://github.com/webdriverio/cucumber-boilerplate/tree/master/src/features>

## Selenium Test Automation Framework using Java

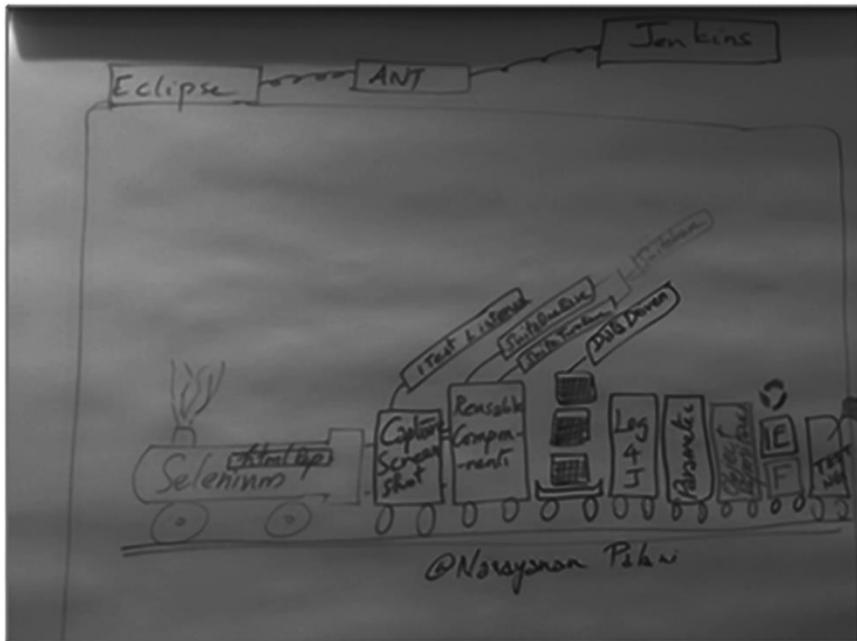
*Programming language used in this section: Java Programming*

*More code examples of this section are available from github at:  
[https://github.com/narayananpalani/testautomation/tree/master/SA  
F](https://github.com/narayananpalani/testautomation/tree/master/SAF)*

*This is one of the significant chapters in this entire book which explains the entire Selenium framework construction in step-by-step approach. Readers are requested to download the practice projects from the link provided at the end of the chapter and follow the instructions given in this chapter – so it will assist in practical experience of how to construct or design or structure a framework of Selenium from scratch!*

### Why Selenium is an important automation tool?

Until early 2006, there was no particular open-source tool (free tool) available in the market, which could prove to be a stable and trusted one to be implemented in testing projects. When Selenium occupied the market, it expanded rapidly and became the only open-source tool extensively used for web-based testing (in comparison to any other paid tools/licensed tools).



Automation framework is the combination of tools and their integration to benefit the test automation of particular application under test (AUT). While considering tools for the framework design, cost and time are two primary factors. Since Selenium Webdriver and tools like TestNG and Ant are freeware and stable releases from open-source community, this attracts to build an expert automation system that handles the software for multiple test releases.

This particular automation framework design requires additional expertise in understanding the test automation and how the maintenance work can be minimized by developing an intelligent system around the automation tools available.

Automation framework using Selenium Webdriver:

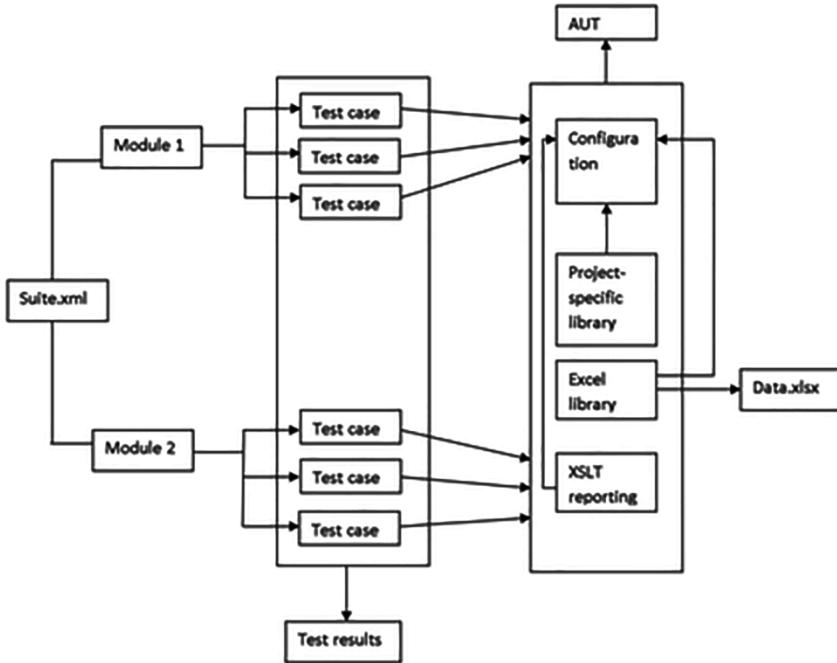
Prerequisite:

Download Eclipse from <http://eclipse.org/downloads/>

Note: Eclipse is famous for Java integrated development environment (IDE). Since Java is largely used for Selenium test automation, it is desirable to automate using Java programming as a scripting language (not 'javascript language').

High-level automation design:

### Selenium Webdriver for hybrid TestNG framework



### How to design the entire test automation framework with required toolset?

The entire project discussed in this section can be downloaded simply through the link given below:

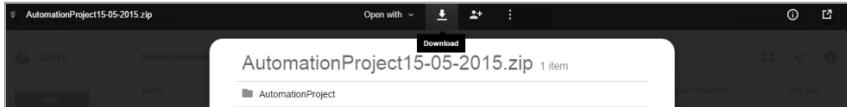
<https://drive.google.com/open?id=0B70x0BZZV5lzUXprYVZyVFRpaDg>

Alternative link for the same file:

<https://goo.gl/yZ5xfP>

Instructions to download: Click on the link-->Open in Chrome-->Click Download button ('click Download Anyway button' if asked)-->Extract after download-->Place the Automation Project Folder in C drive or D drive-->Once Placed, Install Eclipse as per

the instructions in this section and follow the steps noted below to access project.



## Install JAVA

Java programming is a famous and important programming language in testing industry, and the testers with Java knowledge are paid high in the job markets!

Please download Java from the following link:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

**Java SE Downloads**


  
→
DOWNLOAD ↓

Java Platform (JDK) 8u45

Product / File Description	File Size	Download
Linux x86	146.89 MB	jdk-8u45-linux-i586.rpm
Linux x86	166.88 MB	jdk-8u45-linux-i586.tar.gz
Linux x64	145.19 MB	jdk-8u45-linux-x64.rpm
Linux x64	165.24 MB	jdk-8u45-linux-x64.tar.gz
Mac OS X x64	221.98 MB	jdk-8u45-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	131.73 MB	jdk-8u45-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	92.9 MB	jdk-8u45-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	139.51 MB	jdk-8u45-solaris-x64.tar.Z
Solaris x64	95.88 MB	jdk-8u45-solaris-x64.tar.gz
Windows x86	175.98 MB	jdk-8u45-windows-i586.exe
<b>Windows x64</b>	<b>180.44 MB</b>	<b>jdk-8u45-windows-x64.exe</b>

## How to understand whether the Java is installed in the computer or not?

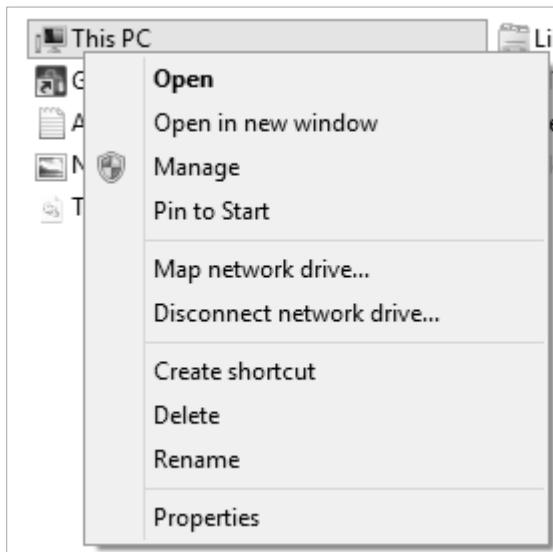
Once the exe file has been downloaded from the website, double click on the exe file and run the installation of the JAVA program

in the computer. Once the task is performed, kindly check the C:/ Drive's program files folder. This folder should have 'JAVA' folder with JDK and JRE sub-folders. As soon as the folders are displayed on the computer screen, it indicates that the Java installation is completed but the configuration has to be done to make sure that the JAVA files are successfully identified by the computer.

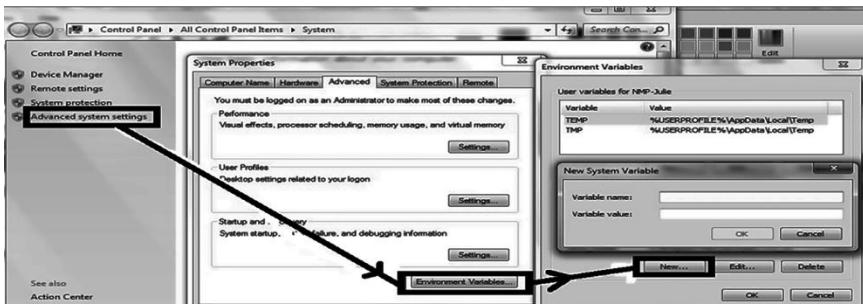
### **JAVA configuration**

Once Java has been installed successfully, JAVA\_HOME and PATH has to be set up in the system following the steps noted below:

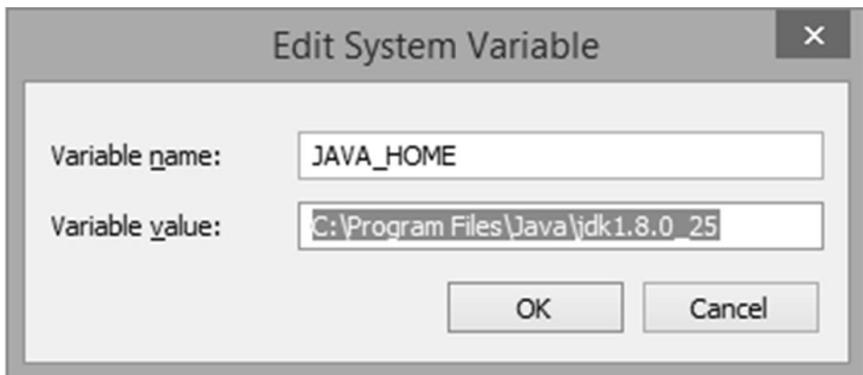
Right click on 'This PC' and select Properties:



Alternatively, navigate to Control Panel and select Advanced System Settings. In Advanced Tab, click on Environment Variables:



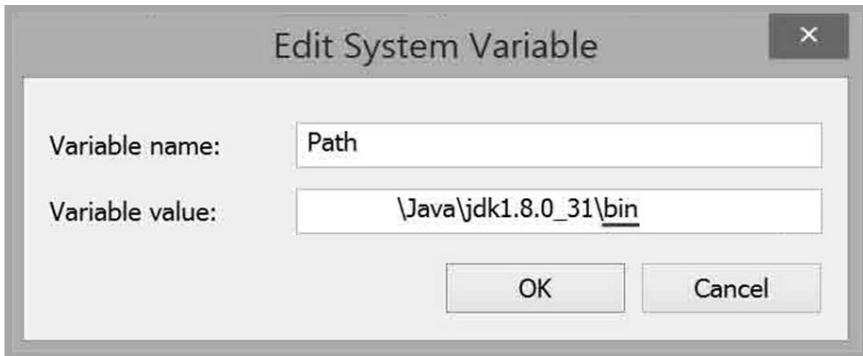
Click on New button and enter the location of JDK folder:



Note: The folder name differs according to your Java version. It is preferable to navigate to the location and copy the path as shown below:

Name	Date modified	Type	Size
bin	09/08/2014 01:13	File folder	
db	09/08/2014 01:13	File folder	
include	09/08/2014 01:13	File folder	
jre	09/08/2014 01:13	File folder	
lib	09/08/2014 01:14	File folder	
COPYRIGHT	16/06/2014 22:29	File	4 KB
javafx-src.zip	09/08/2014 01:13	Compressed (zipp...)	4,565 KB
LICENSE	09/08/2014 01:13	File	1 KB
README.html	09/08/2014 01:13	HTML Document	1 KB
release	09/08/2014 01:14	File	1 KB
src.zip	16/06/2014 22:29	Compressed (zipp...)	20,697 KB
THIRDPARTYLICENSEREADME.txt	09/08/2014 01:13	Text Document	175 KB
THIRDPARTYLICENSEREADME-JAVAFX.txt	09/08/2014 01:13	Text Document	108 KB

Similarly, click on the New button again to enter path as shown below:



Note: Make sure that you have mentioned the path including bin folder as above. You can take a copy of this path by just clicking on the bin folder as shown below:

	Name	Date modified	Type	Size
ces	bin	09/08/2014 01:13	File folder	
s	db	09/08/2014 01:13	File folder	
	include	09/08/2014 01:13	File folder	
	jre	09/08/2014 01:13	File folder	
	lib	09/08/2014 01:14	File folder	
	COPYRIGHT	16/06/2014 22:29	File	4 KB
	javafx-src.zip	09/08/2014 01:13	Compressed (zipp...)	4,565 KB
	LICENSE	09/08/2014 01:13	File	1 KB
	README.html	09/08/2014 01:13	HTML Document	1 KB
	release	09/08/2014 01:14	File	1 KB
	src.zip	16/06/2014 22:29	Compressed (zipp...)	20,697 KB
	THIRDPARTYLICENSEREADME.txt	09/08/2014 01:13	Text Document	175 KB
	THIRDPARTYLICENSEREADME-JAVAFX.txt	09/08/2014 01:13	Text Document	108 KB

## Why JAVA\_HOME and PATH has been updated in environment variable

Whenever the Java programs are compiled and executed in computer, required JAVA files have to be referred. So the computer is supposed to know about the location of the JAVA files and the respective PATH of bin folder. So the file location is provided through JAVA\_HOME, and bin folder location is provided through PATH set up. Once it is done, it is a good practice to restart computer and check the version of JAVA through command prompt. Simply running the command JAVA –VERSION will provide the Java version in command prompt which proves that the JAVA installation has been completed successfully.

Alternative way to set up JAVA\_HOME and PATH is through Command Prompt. This can be done by launching Command Prompt as follows.

Step1: Press Windows+R (to open RUN)

Step2: Type CMD, press enter

Step3: In Command Prompt, type 'SET JAVA\_HOME = c:/Program Files/Java'

Step4: Type 'SET PATH = %PATH%;%JAVA\_HOME%\bin'

Once JAVA installation and configuration has been completed, it is required to install Eclipse as it acts as a tool to write JAVA codes!

### **Launch Eclipse**

Download latest Eclipse version from <http://eclipse.org/downloads/>. Once downloaded, please install it and open Eclipse from the location it is downloaded by clicking on eclipse.exe as shown below:

## Software Automation Testing Secrets Revealed

Name	Date modified	Type	Size
configuration	04/05/2015 13:30	File folder	
dropins	25/09/2014 14:52	File folder	
features	15/11/2014 14:44	File folder	
p2	17/10/2014 22:24	File folder	
plugins	15/11/2014 14:44	File folder	
readme	17/10/2014 22:24	File folder	
.eclipseproduct	08/10/2014 19:28	ECLIPSEPRODUCT...	1 KB
artifacts.xml	15/11/2014 14:44	XML File	160 KB
<b>eclipse.exe</b>	<b>08/10/2014 19:28</b>	<b>Application</b>	<b>313 KB</b>
eclipse.ini	15/11/2014 14:44	Configuration sett...	1 KB
eclipsec.exe	08/10/2014 19:28	Application	26 KB
epl-v10.html	08/10/2014 19:28	HTML Document	13 KB
notice.html	08/10/2014 19:28	HTML Document	9 KB

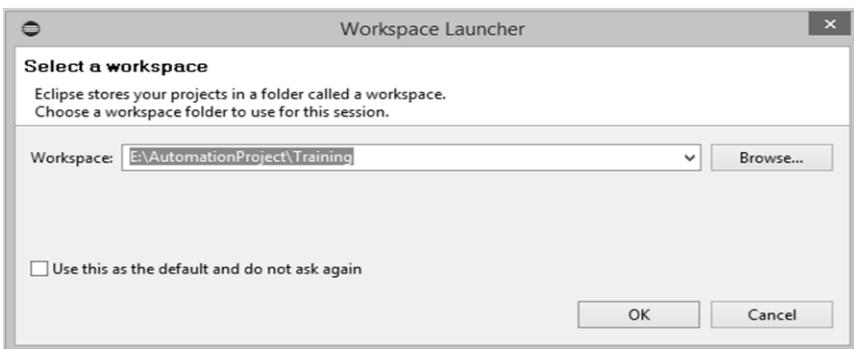
Name	Date modified	Type	Size
configuration	04/05/2015 13:30	File folder	
dropins	25/09/2014 14:52	File folder	
features	15/11/2014 14:44	File folder	
p2	17/10/2014 22:24	File folder	
plugins	15/11/2014 14:44	File folder	
readme	17/10/2014 22:24	File folder	
.eclipseproduct	08/10/2014 19:28	ECLIPSEPRODUCT...	1 KB
artifacts.xml	15/11/2014 14:44	XML File	160 KB
<b>eclipse.exe</b>	<b>08/10/2014 19:28</b>	<b>Application</b>	<b>313 KB</b>
eclipse.ini	15/11/2014 14:44	Configuration sett...	1 KB
eclipsec.exe	08/10/2014 19:28	Application	26 KB
epl-v10.html	08/10/2014 19:28	HTML Document	13 KB
notice.html	08/10/2014 19:28	HTML Document	9 KB

Figure: Eclipse exe file to launch eclipse

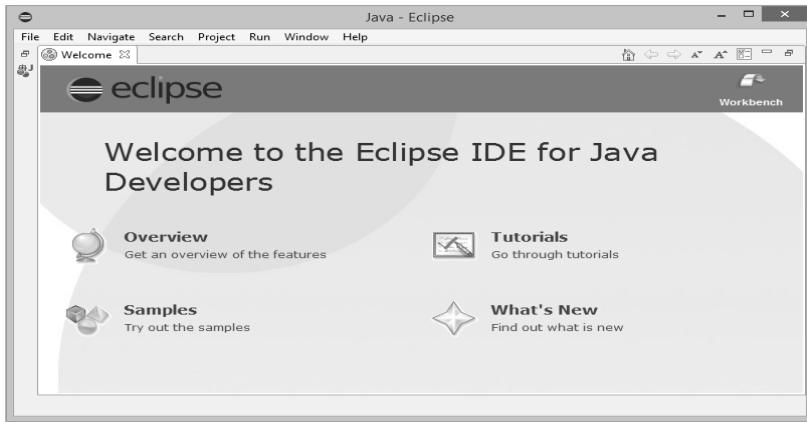
Verify that the Eclipse is getting launched successfully:



Set workspace for the Eclipse project as this location will be the primary location to save all the Java files that are going to be used for the automation framework:

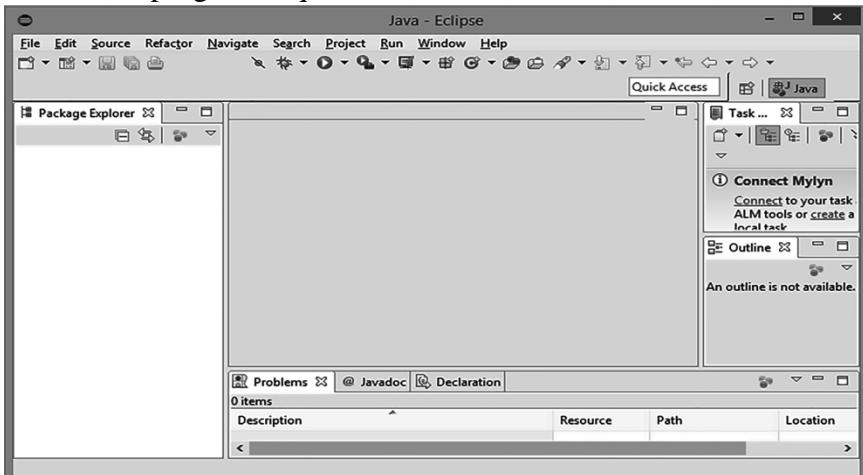


Once the workspace has been provided, the welcome page appears as shown in the picture below, and 'Workbench' button on the top right side of the window is to be clicked to view workbench and write the new programs:

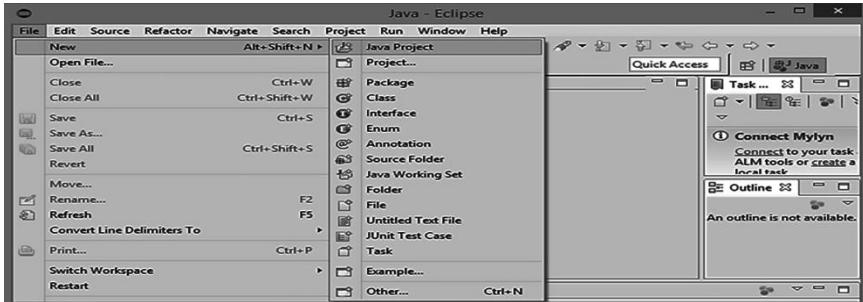


## Package explorer

This is the left panel where the projects are listed along with source code of the program required for the automation framework:

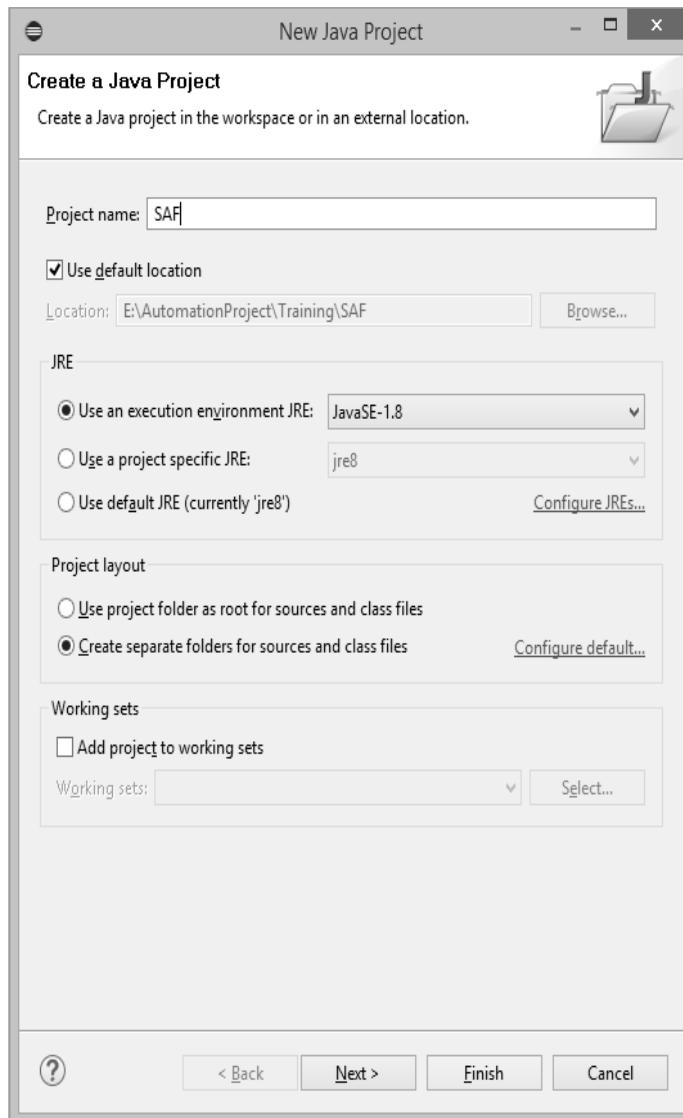


Create new project with the name SAF (Selenium Automation Framework):

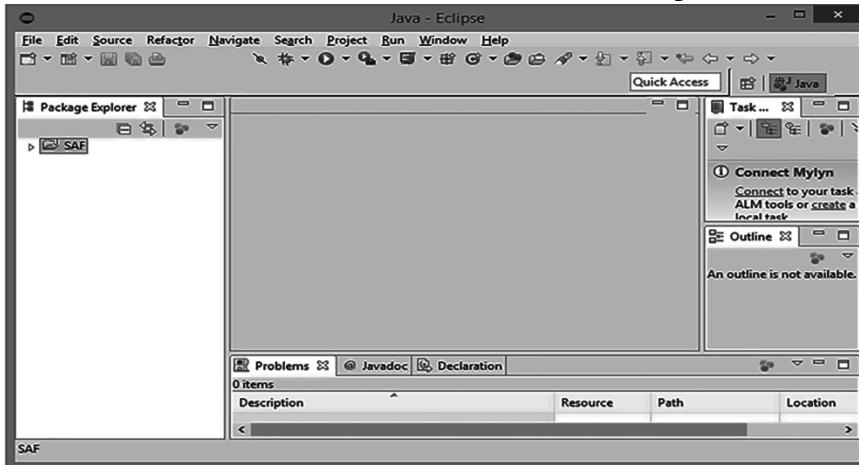


This SAF (Selenium automation framework) will act as a primary project for all the extensions and the test scripts used across the testing project.

## Software Automation Testing Secrets Revealed

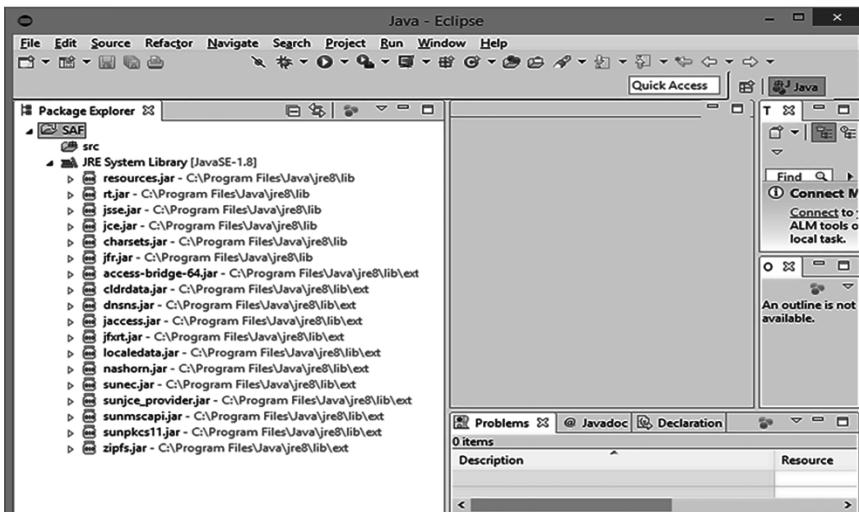


Once the folder is created, this can be viewed in Eclipse:



The new SAF project will be listed in Package Explorer as follows:

Note: JRE system library is a custom folder created for the test project automatically. If it is not created for available in different folder, it means either Eclipse would have been updated to the latest version or the folder creation is incorrect.



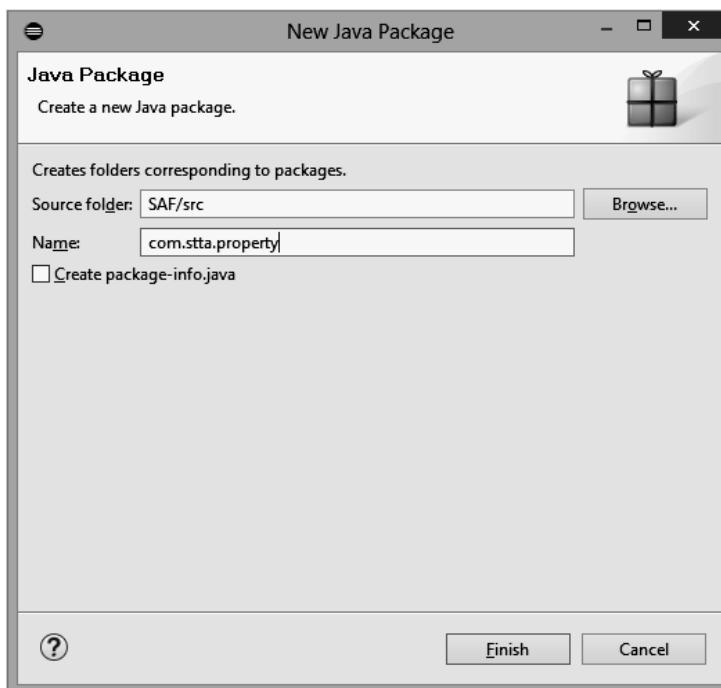
## Create Selenium Automation Framework packages

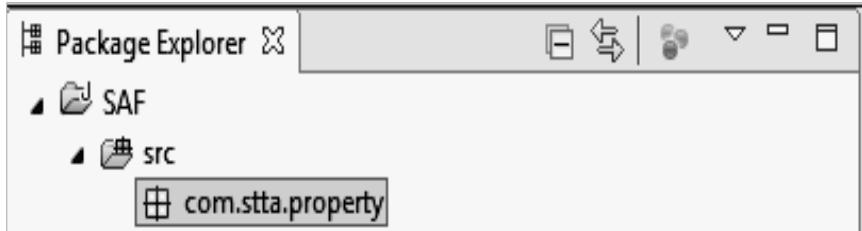
Right click on the SAF project, click on New and click on Package to create new packages which are the most wanted facilities of the automation framework.



### Create a package named com.stta.property

This package is required to have reusable properties, such as xpath property, and parameters, such as website URLs, browser details at central location. So we can insist the automation framework to run from particular URL in specific browser, just by updating the files in this package!

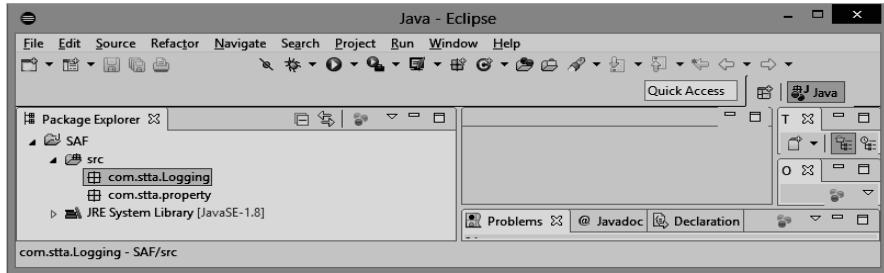




### Create package com.stta.Logging

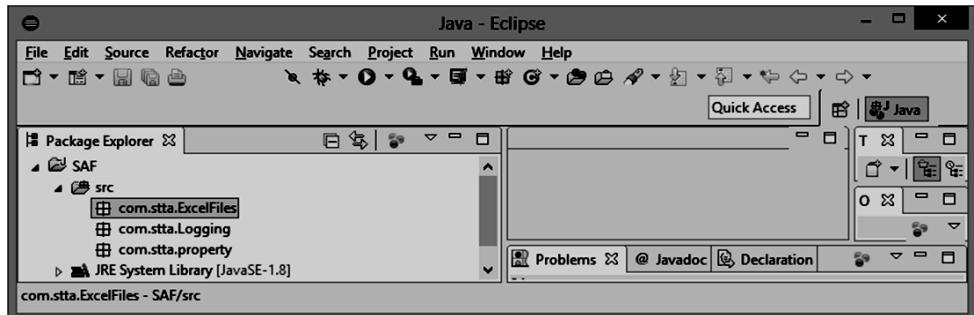
Similar to the package above, create another package and name it as com.stta.Logging.

This package is an important facility of any automation framework to capture the exceptions, failures and traces of test execution using logs.



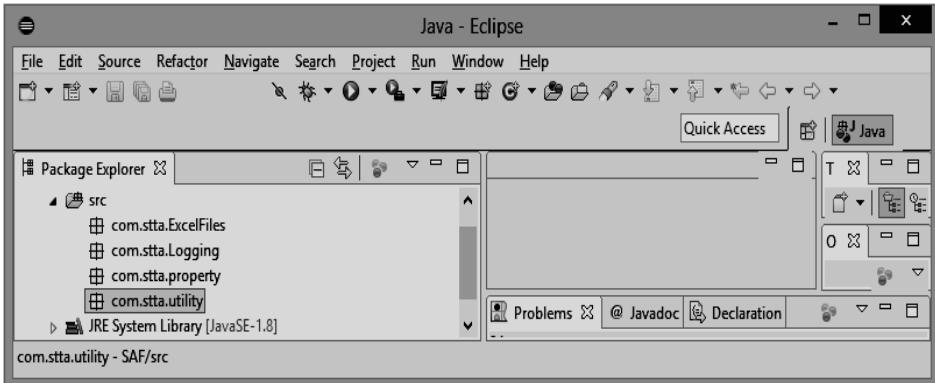
### Create package com.stta.ExcelFiles

All the related data sheets can be downloaded and kept under this package. This can be possible by clicking the right click on the package and pasting the required excel files to this package.

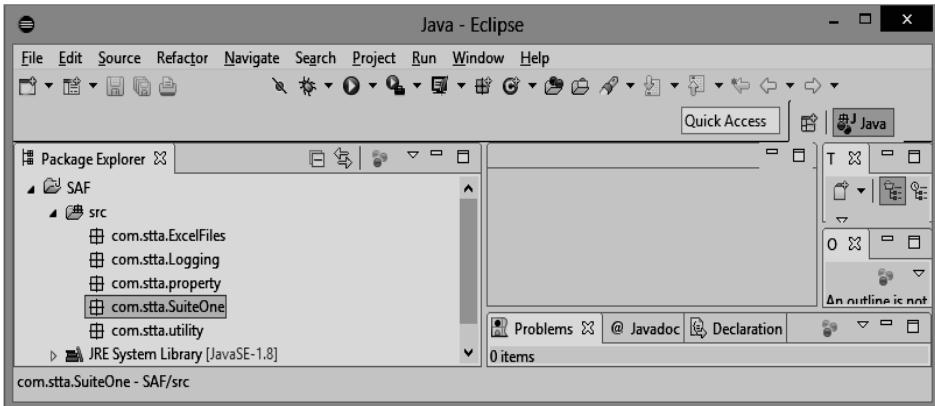


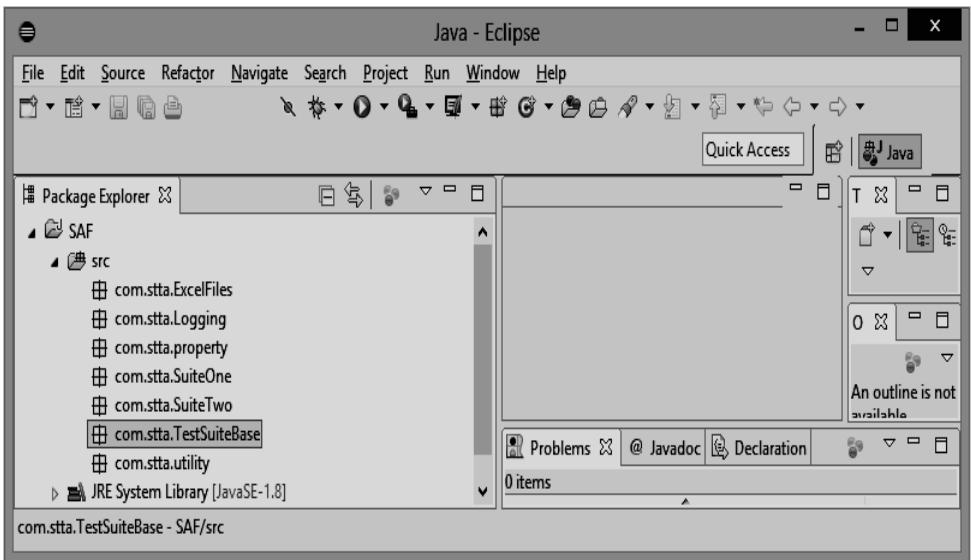
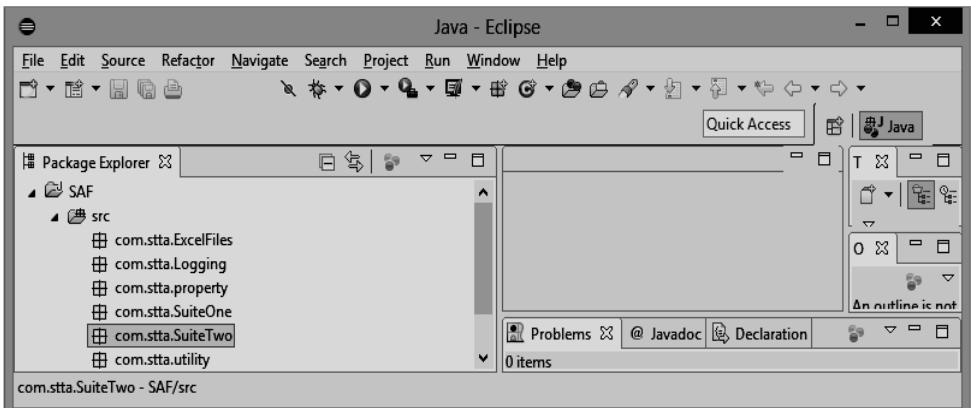
## Create package com.stta.utility

Reusable javascripts, reusable functions and other centralized file properties can be managed from this folder.



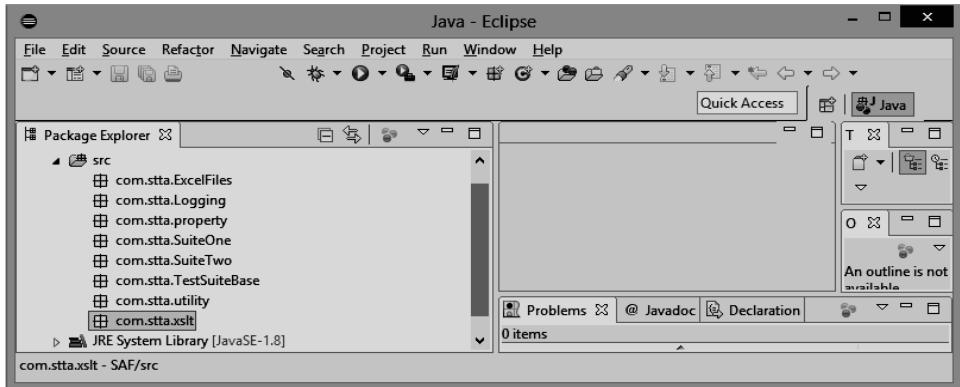
## Create test script-based package com.stta.SuiteOne, com.stta.SuiteTwo and com.stta.TestSuiteBase





## Create com.stta.xslt folder

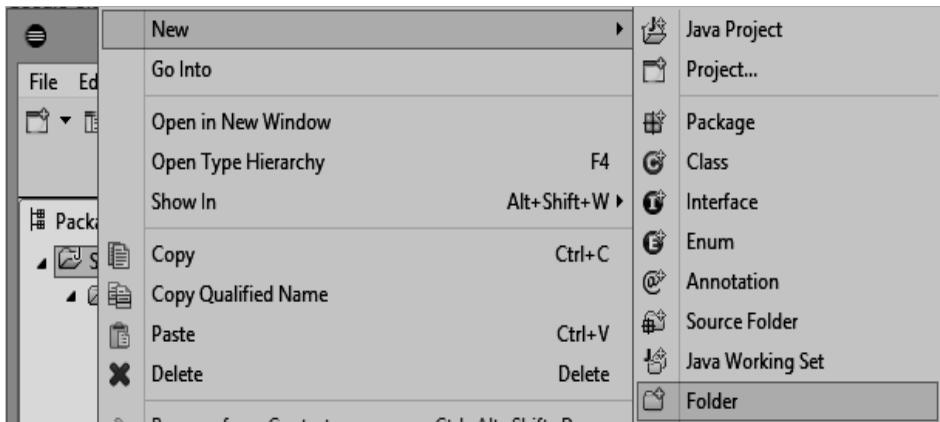
All the test report-ability logics can be saved under this package so that the latest versions and maintenance of the test reports are made simple.

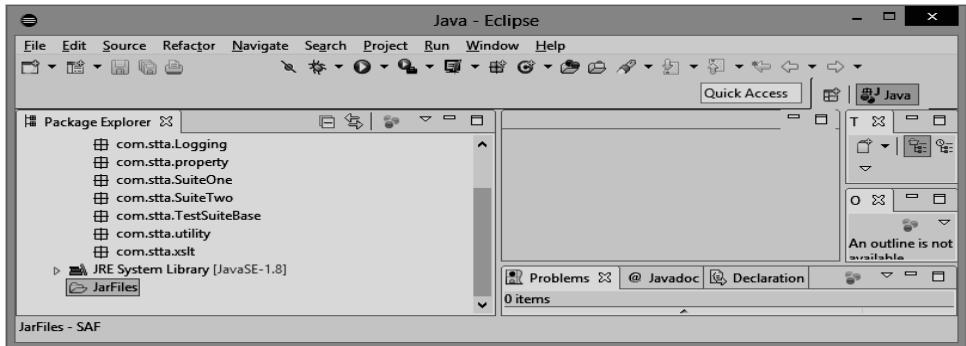


## Create a folder name JAR files to have all the relevant JAR files

When the automation framework is designed, it is a mixture of more than one extendable jar files downloaded from various source locations. Moreover, each jar file differs from their old versions due to latest defect fixes. So it is important to keep all the installed jar files of the automation framework in this particular folder.

Right click on the project, click on New and click on Folder:





Required downloadable jar files from various locations:

Apache POI API for excel data-driven framework:

Selenium doesn't support excel-based files at the latest releases, and it is possible to read and write in excel files using Apache POI API files!

Following are the files helpful in extending the excel read/write facility:

**poi-3.12-beta1-20150228.jar**

**poi-ooxml-3.12-beta1-20150228.jar**

**poi-ooxml-schemas-3.12-beta1-20150228.jar**

Inside 'ooxml-lib' folder:

**xmlbeans-2.6.0.jar**

**dom4j-1.6.1.jar**

Note on dom4j.jar file: If this file is not found inside ooxml-lib folder, it means that the file has not been provided for latest versions of Apache API. So it is available from poi-bin-3.10-FINAL-20140208.zip version. You can download this from <http://archive.apache.org/dist/poi/release/bin/>

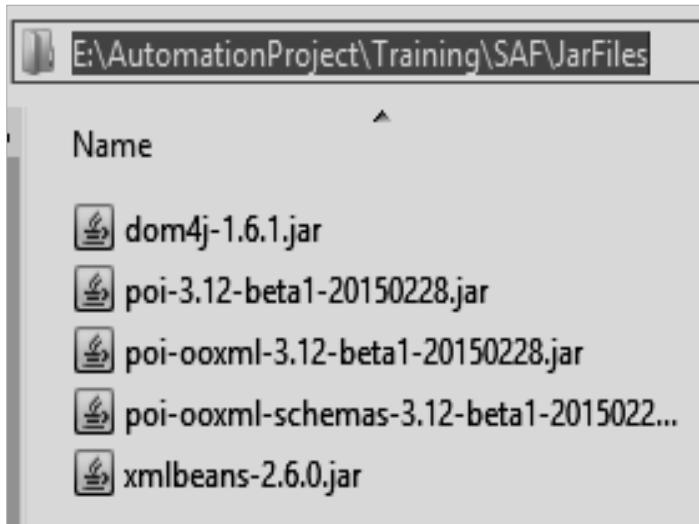
**Note:** Version of the files differson the basis of the time it has been downloaded.

Download the file from <http://poi.apache.org/download.html>

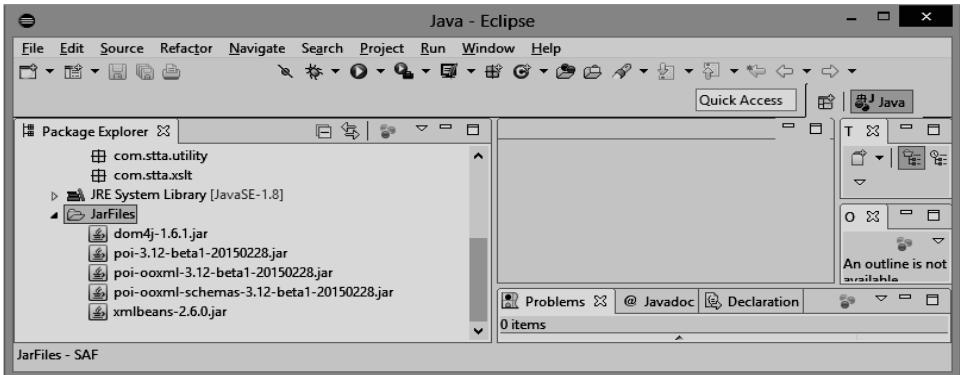
## Binary Distribution

- poi-bin-3.11-20141221.tar.gz (17MB, signed)  
MD5 checksum: 77259a19bfa74dccdc610e4ec3ea6535  
SHA1 checksum: h615193067011a8917f03a0c0e835fa31d3ec1d2
- poi-bin-3.11-20141221.zip (24MB, signed)  
MD5 checksum: 2D6211C5df1b164da26ced6c4d756dc  
SHA1 checksum: 0b14e26def396ad23ec83a177d97ce954d2f4f86

After downloading these files, place them into the JarFiles folder:



This indicates, you can view the JarFiles folder in Package Explorer after refreshing the folder:



### Download the jar file for logs extendibility:

Logging services such as execution time, test failure, success and exceptions are required to be captured during test execution. With the help of log4j file, it is possible to extend this facility in Selenium Automation Framework as the Selenium Webdriver doesn't have log-capturing technology in recent releases.

## Download Apache log4j 1.2.17

Apache log4j 1.2.17 is distributed under the Apache License, version 2.0 [View](#).

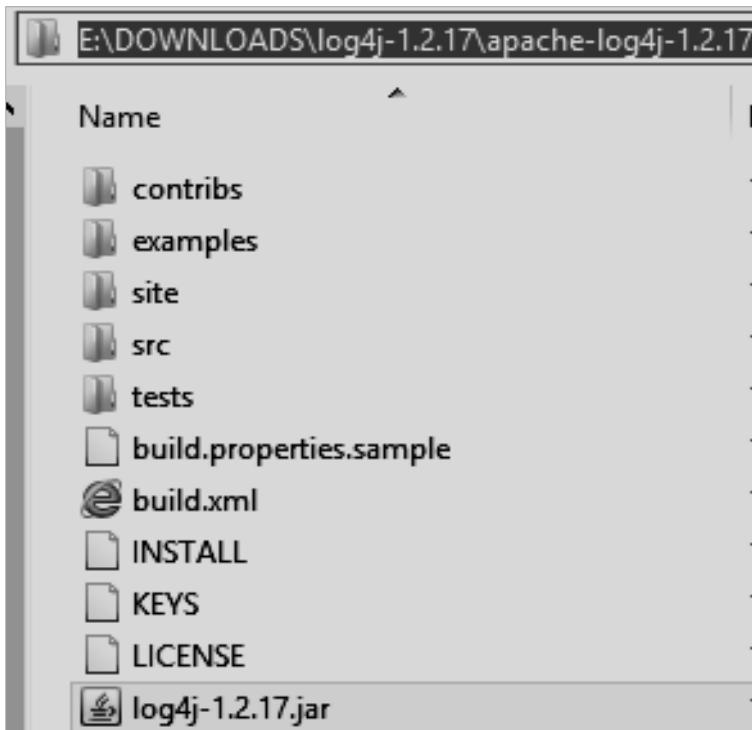
The link in the Mirrors column should display a list of available mirrors with a different browser. The checksum and signature are links to the originals on the mirror.

	Mirrors
Apache log4j 1.2.17 (tar.gz)	<a href="#">log4j-1.2.17.tar.gz</a>
Apache log4j 1.2.17 (zip)	<a href="#">log4j-1.2.17.zip</a>

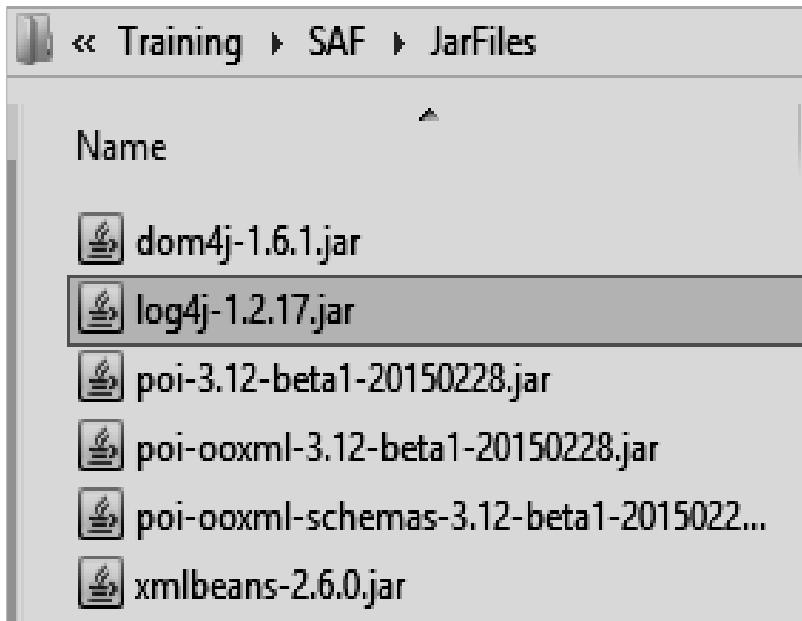
Download the file from

<http://logging.apache.org/log4j/1.2/download.html>

After extracting the file, please copy the file log4j from downloads:



Paste this file into JarFiles folder which is created for the Selenium Automation Framework:



### Main Selenium Webdriver-based jar files for Selenium Automation Framework

The very important test engine of the automation framework is the testing tool. Especially Selenium is made up of JAR files which evolve over a period of time to upgraded versions. So ensure that the latest version of file has been downloaded from the link given below:

Download the files from <http://docs.seleniumhq.org/download/>

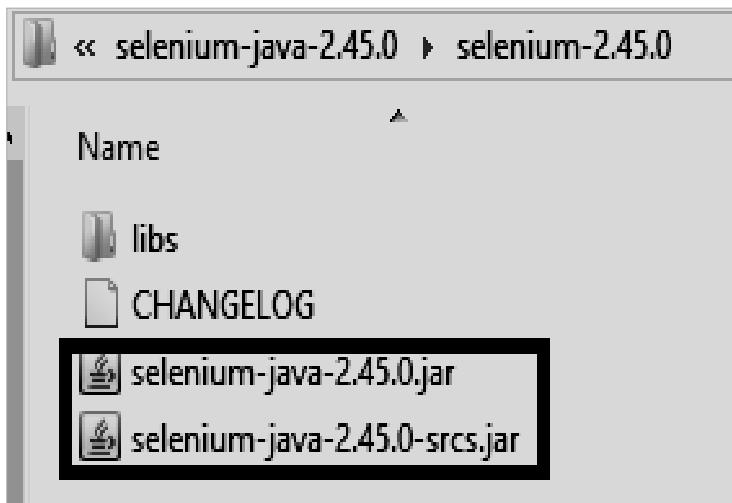
### Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Selenium RC, Selenium Remote WebDriver) or create local Selenium WebDriver script you need to make use of language-specific client drivers. These languages include both 1.x and 2.x style clients.

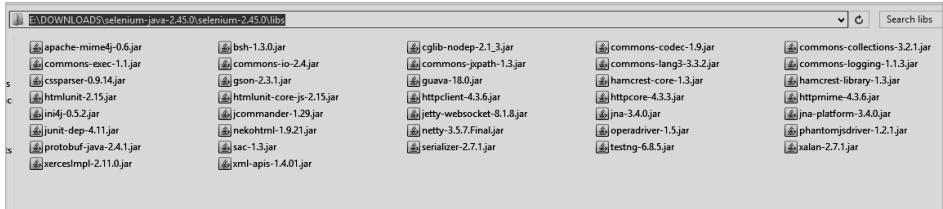
While language bindings for other languages exist, these are the core ones that are supported by the main project hosted on google code.

Language	Client Version	Release Date				
Java	2.45.0	2015-02-26	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">Javadoc</a>	
C#	2.45.0	2015-02-27	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>	
Ruby	2.45.0	2015-02-27	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>	
Python	2.45.0	2015-02-26	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>	
Javascript (Node)	2.45.0	2015-02-26	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>	

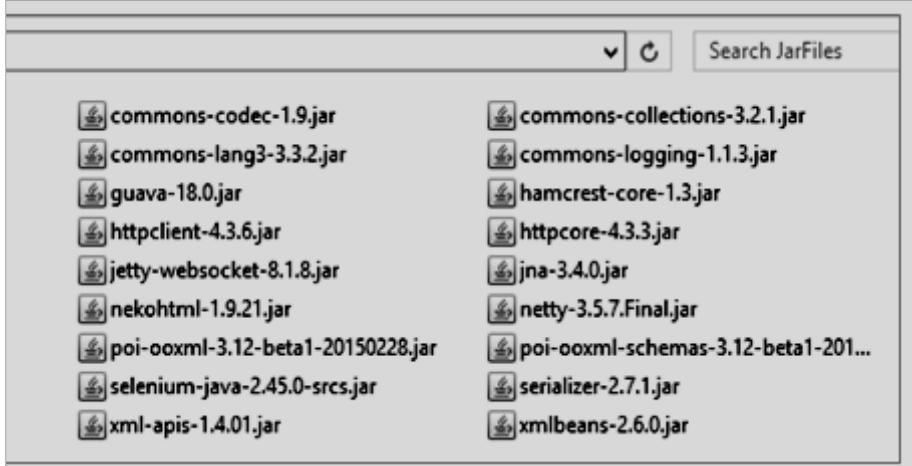
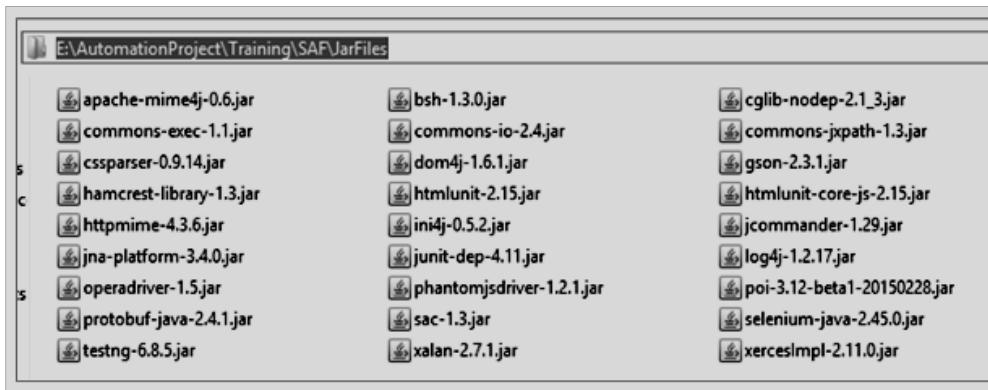
Once the files are downloaded, confirm that all the jar files from this file (including the jar files from sub-folders of this file) are pasted into JarFiles folder which is created for the automation framework.



It should be assured that the following files are included without missing any of them as each file plays a major role in the automation framework:



Once these files are pasted into JarFiles folder, this may look like the below structure:



## XSLT Report-based jar files

The basic need of an automation framework is the test report! If the test reports are attractive, it offers positive impression on the automation framework from project managers and test managers. So XSLT Report is one of the niche reporting methods, and extending XSLT to Selenium Automation Framework helps in getting the test reports to share it across the stakeholders!

Download the saxon jar file from

<http://mvnrepository.com/artifact/net.sf.saxon/saxon/8.7>

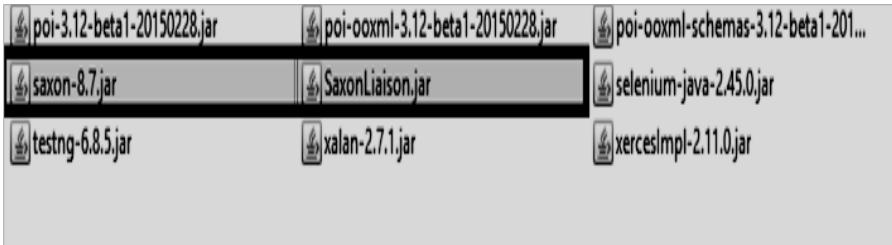
or from <https://goo.gl/9da8gC>

<b>Artifact</b>	<a href="#">Download ( JAR ) (3.2 MB)</a>
<b>POM File</b>	<a href="#">View</a>
<b>Date</b>	(Feb 23, 2006)
<b>HomePage</b>	<a href="http://saxon.sf.net">http://saxon.sf.net</a>

Alternatively, both the files can be downloaded from the link given below:

<https://github.com/prashanth-sams/testng-xslt-1.1.2/tree/master/lib>

Once these two files are downloaded, place them into the JarFiles folder:



To combine the test results in XSL file, it is required to have TestNG compatible xsl file in the automation framework.

Download the zip file from <https://github.com/prashanth-sams/testng-xslt-1.1.2> or from <https://goo.gl/7xSEPG>

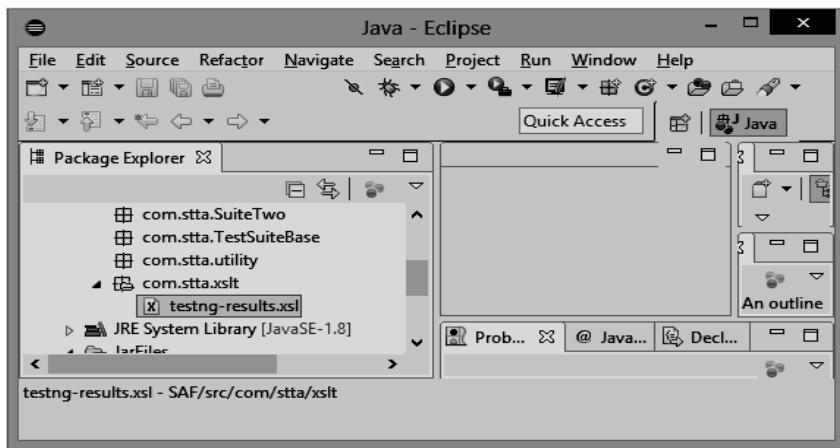
Once downloaded, navigate to the resources folder (testng-xslt-1.1.2-master\testng-xslt-1.1.2-master\src\main\resources).

Name	Date modified	Type	Size
testng-results.xls	11/05/2015 13:21	XSL File	58 KB

Copy the file testng-results.xsl and paste it to the package created for the XSLT Reports as below:

Name	Date modified	Type	Size
testng-results.xsl	11/05/2015 13:24	XSL File	58 KB

Once the file is located in this Package folder, this can be viewed in Package Explorer:



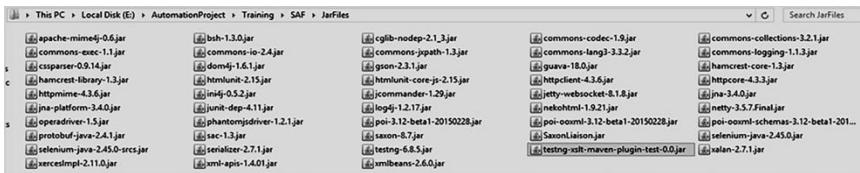
## TestNG plug-in to use the XSLT Reports

TestNG is used mainly to run the test scripts generated in Selenium. So it needs to be extended to XSLT reporting jar files using the plug-in. So the reports get the latest test execution status automatically. Download the plug-in from the below link and paste to the JarFiles folder as it is an important plug-in to capture the test results.

Link:

<https://drive.google.com/file/d/0B6vnknygMB3IdzF4X2taWFRRMVE/edit>

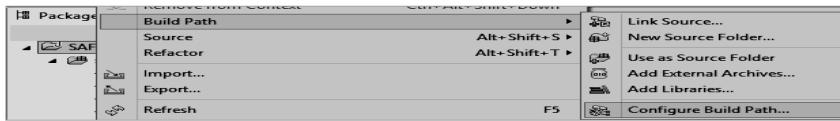
JarFiles folder structure after pasting the plug-in:



## Project Build Path configuration

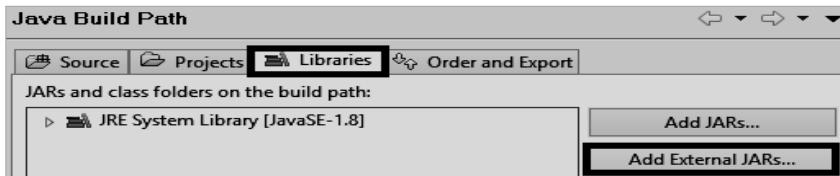
The need of placing all the relevant JAR files into one central location in automation framework is felt to access them in order to configure the project SAF.

Right click on the folder SAF, click on Build Path and click on Configure Build Path:



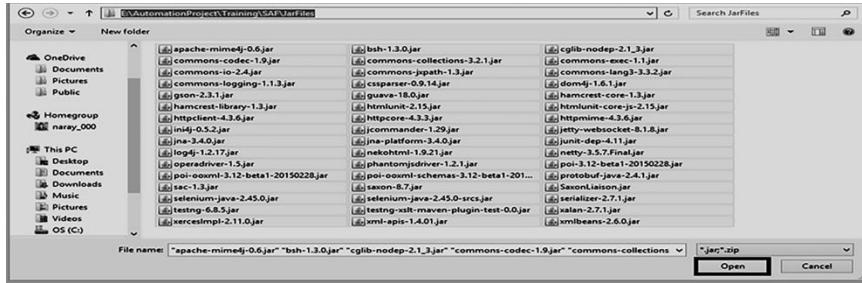
In Properties of SAF, click on the Libraries tab then on Add External JARs button:

**Note:** Ensure that the Libraries tab is clicked and then Add External JARs button is clicked from there.

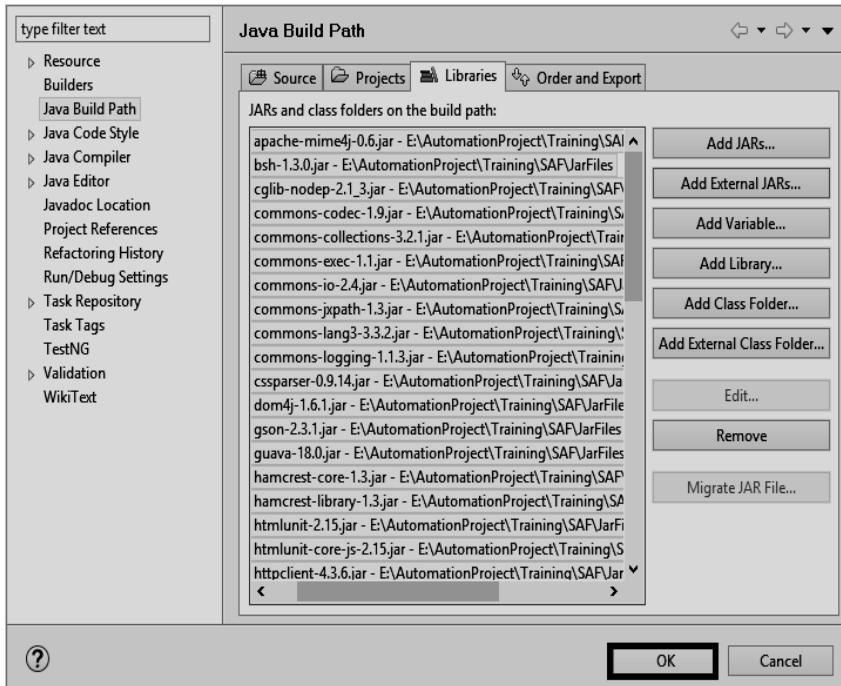


Select the path of the JarFiles folder and select all the JAR files in the folder and then click on Open button:

# Software Automation Testing Secrets Revealed

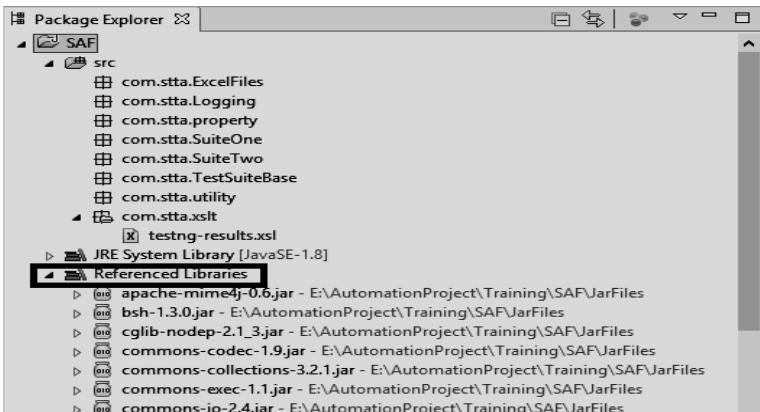


Once the files are selected, click on OK button and make sure that the files are added:



Once files are added, they can be found in the Referenced Libraries:

Note: Count the files from JarFiles folder and this Reference Library to confirm that all the JARs have been selected to the build path.



## Excel readability

Java program should be designed to interact to each column/row of test data to read and write the data as part of test execution. This can be done by using Read\_XLS.java file.

Download the file from the below link and place it in the utility folder:

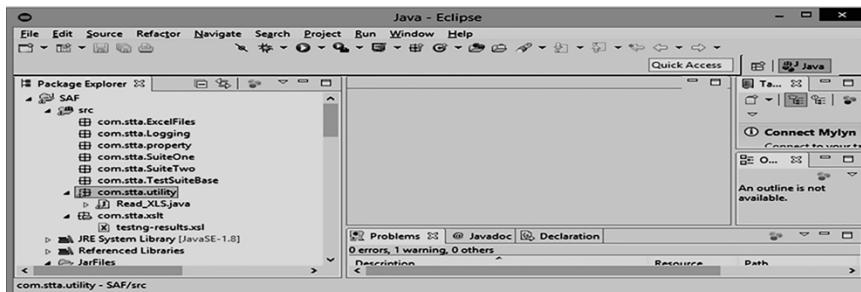
<https://drive.google.com/file/d/0B6vnknygMB3INVFqYi03T2FrOXc/edit>

Alternative Link: <https://goo.gl/MC66oD>

Utility folder:

E:\AutomationProject\Training\SAF\src\com\stta\utility			
Name	Date modified	Type	Size
Read_XLS.java	11/05/2015 14:17	JAVA File	7 KB

Once it is placed in Utility folder, this can be found at Package Explorer:



## Suite Utility

It is a best practice to have the test scripts in different test cases and linked to respective test suites. Hence, each test suite can be executed with the help of TestNG.

In order to have this flexibility, Suite Utility should be extended from a JAVA file:

Download file from

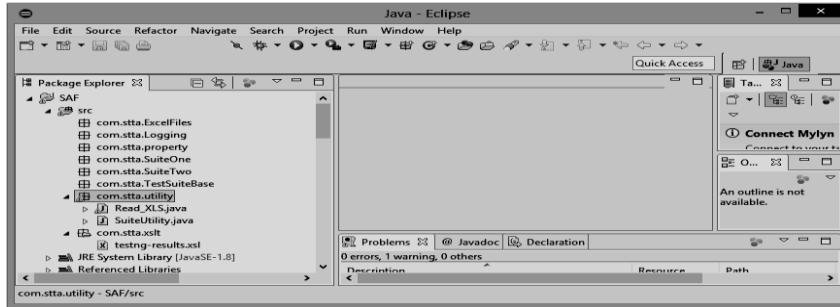
<https://drive.google.com/file/d/0B6vnknygMB3ISGdURIYwU20xWms/edit>

Alternative Link: <https://goo.gl/z39j5u>

Paste the utility Java file in Utility folder:

Name	Date modified
Read_XLS.java	11/05/2016
SuiteUtility.java	11/05/2016

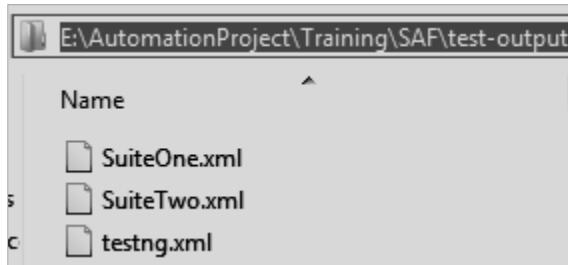
This can be found in Package Explorer:



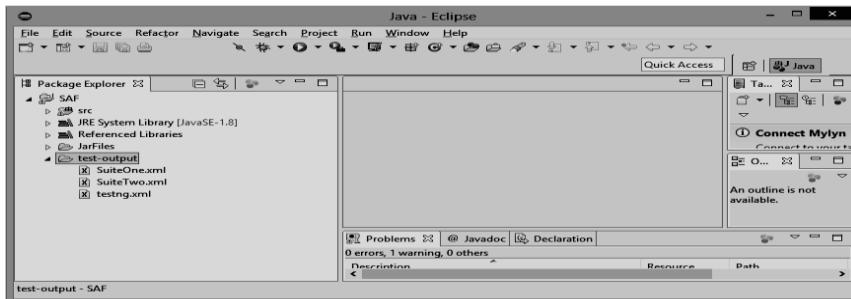
## Test execution through TestNG

Create exclusive test suites for each set of test cases and have the central testng.xml to execute required test scripts.

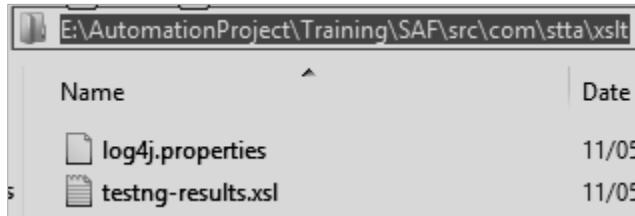
To facilitate this execution strategy, download the TestNG files and place them in test-output folder:



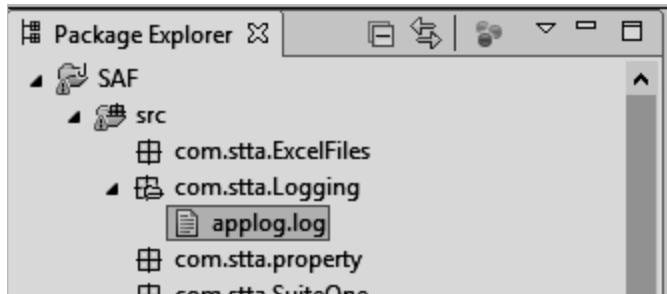
These TestNG files can be viewed from Package Explorer:



Place the files for logs and TestNG-supported xsl files in xslt folder:



Create a file with name applog.log under com.stta.Logging



Log4J file basically captures the records that are executed currently by TestNG and update it to applog.log in the following format:

2015-05-17 10:27:07,739 – rootLogger – INFO – All Excel Files Initialised successfully.

2015-05-17 10:27:07,747 – rootLogger – INFO – Param.properties file loaded successfully.

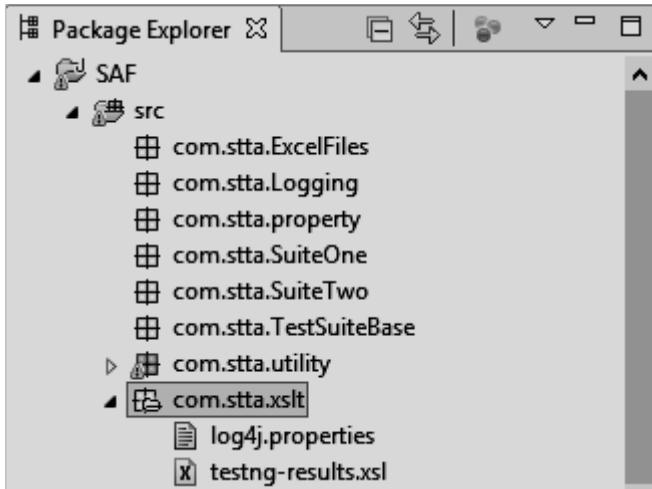
2015-05-17 10:27:07,747 – rootLogger – INFO – Objects.properties file loaded successfully.

2015-05-17 10:27:07,747 – rootLogger – INFO – Execution started for SuiteOneBase.

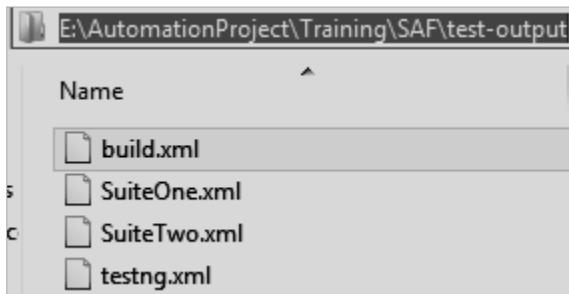
Once the execution is completed, these logs can be captured from the following folder alternatively:

E:\AutomationProject\Training\SAF\src\com\stta\Logging	
Name	Date modified
applog.log	11/05/2015 15:55

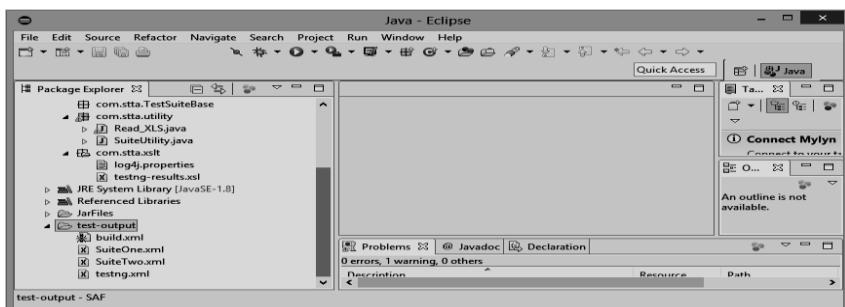
Once the log4j and TestNG result files are saved in xslt folder, it can be viewed in Eclipse:



Place the executable TestNG xml files in new folder 'test-output' under SAF:

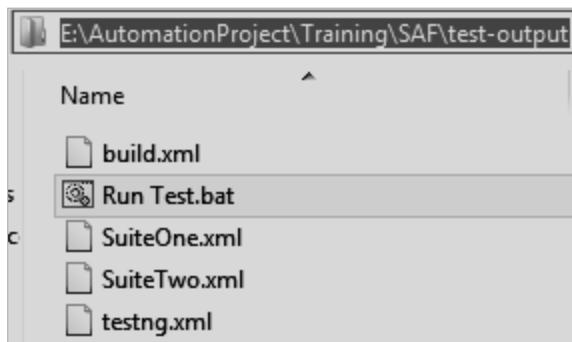


Once the TestNG files are added to the folder, this can be viewed as follows:



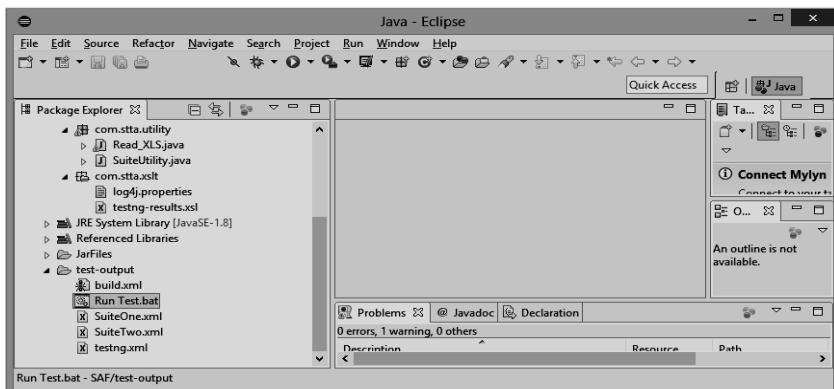
## Run Test.bat for Ant build

Once the tests are built in the automation framework, they can be combined using Ant and run it from command prompt. So this build (ant) can be used in Jenkins.



## Run Test Bat File

To have such compatibility with test execution, it has to be controlled by servers or the central locations like Jenkins where the bat file, shown below, can be located and timed to have a build at particular test execution time:



## Property file

This file has to be updated with parameters which are required for the test automation project:

Name	Date modified	Type	Size
Param.properties	11/05/2015 16:43	PROPERTIES FILE	1 KB

Each browser required for Selenium framework has to be downloaded and listed here for the reference(except Firefox as it is an in-built feature):

Name	Date modified
chromedriver.exe	11/05/2015 16:43
IEDriverServer.exe	11/05/2015 16:43

Each object (xpaths preferably) can be listed by object.properties file for ease of use. This is followed just to have multiple object files and to replace them on the basis of the test environment where the test execution is planned:

Name	Date modified	Type	Size
Objects.properties	11/05/2015 16:56	PROPERTIES FILE	1 KB
Param.properties	11/05/2015 16:54	PROPERTIES FILE	1 KB

Customized screenshot utilities can be added to the projected and extended through TestNG in order to capture the test evidences:

Name	Date modified
Read_XLS.java	11/05/2015 14:17
ScreenshotUtility.java	11/05/2015 17:03
SuiteUtility.java	11/05/2015 14:43

## Excel files for the test data of each iteration

TestSuiteList.xls centrally controls the subset of the tests like SuiteOne,SuiteTwo etc.

E:\AutomationProject\Training\SAF\src\com\stta\ExcelFiles		
Name		Date modified
SuiteOne.xls		11/05/2015 17:02

E:\AutomationProject\Training\SAF\src\com\stta\ExcelFiles		
Name		Date modified
SuiteOne.xls		11/05/2015 17:02
SuiteTwo.xls		11/05/2015 17:09

E:\AutomationProject\Training\SAF\src\com\stta\ExcelFiles			
Name	Date modified	Type	Size
SuiteOne.xls	11/05/2015 17:02	Microsoft Office E...	18 KB
SuiteTwo.xls	11/05/2015 17:09	Microsoft Office E...	18 KB
TestSuiteList.xls	11/05/2015 17:13	Microsoft Office E...	18 KB

## Running the project from Ant

Navigate to the project folder in command prompt and enter ant as command to know whether the Ant is installed in the system or not:

```
C:\WINDOWS\system32\cmd.exe
E:\AutomationProject\Training\SAF>ant
'ant' is not recognized as an internal or external command,
operable program or batch file.
```

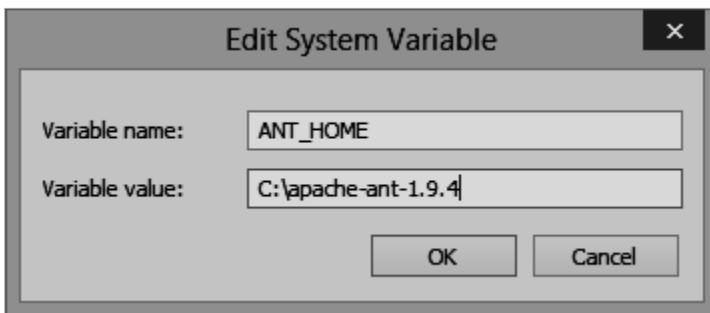
If Ant is not installed, simply type 'Install Ant' in Google to find the steps:

The screenshot shows a search results page with the query "install ant". The results include a list of steps:

1. JAVA\_HOME. Make sure JDK is installed, and JAVA\_HOME is configured as Windows environment variable. ...
2. Download Apache Ant. Visit Apache Ant official website, download the Ant binary zip file, for example : apache-ant-1.9.4-bin.zip , unzip it to the folder you want to store Apache Ant. ...
3. Add ANT\_HOME. ...
4. Update PATH. ...
5. Verification.

Once file has been downloaded from <http://ant.apache.org/antlibs/bindownload.cgi>, this can be extracted from c:/ drive in the system. Once the file has been extracted from C:/ drive, update the ANT\_HOME and PATH in environment variables (similar to JAVA\_HOME and PATH which have already been explained earlier).

### Ant Home





Once ANT\_HOME and PATH has been set, this can be verified by entering the command 'ant -version':

```
C:\WINDOWS\system32\cmd.exe
C:\apache-ant-1.9.4\bin>ant -version
Unable to locate tools.jar. Expected to find it in C:\Program Files\Java\jre8\lib\rt\tools.jar
Apache Ant(TM) version 1.9.4 compiled on April 29 2014
C:\apache-ant-1.9.4\bin>_
```

If the system displays Ant versions and compilation date, it means that the installation is completed and just that the configuration needs to be updated for the automation framework.

Navigate to test-output folder of automation framework and check for the build using the query 'ant':

```
C:\WINDOWS\system32\cmd.exe
E:\AutomationProject\Training\SAF\test-output>ant
Unable to locate tools.jar. Expected to find it in C:\Program Files\Java\jre8\lib\rt\tools.jar
Buildfile: E:\AutomationProject\Training\SAF\test-output\build.xml
usage:
[echo] [echo]           ant run will execute the test
[echo]
BUILD SUCCESSFUL
Total time: 0 seconds
```

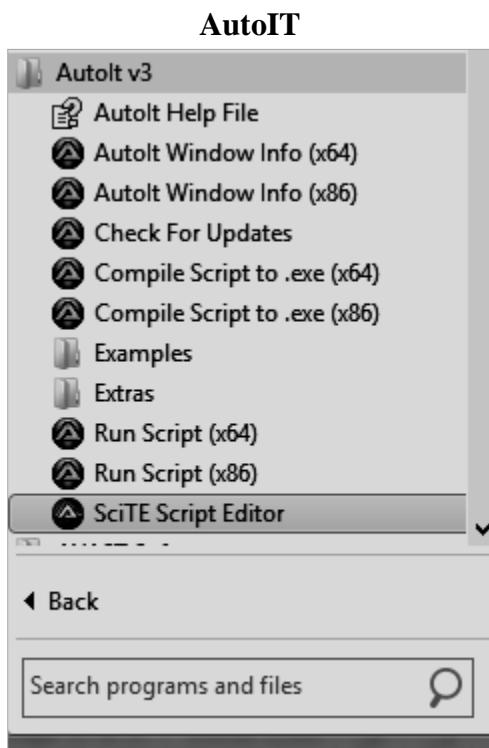
If the message – BUILD SUCCESSFUL displayed, the build is ready for the deployment!

Similar to the query above, ant clear query should result in BUILD SUCCESSFUL as the build has been placed in this folder for compilation:

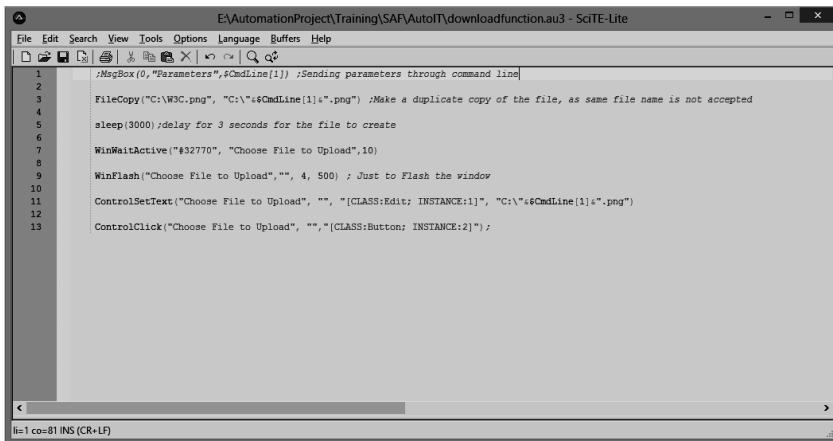


```
C:\WINDOWS\system32\cmd.exe
E:\AutomationProject\Training\SAF\test-output>ant clear
Unable to locate tools.jar. Expected to find it in C:\Program Files\Java\jre8\lib\tools.jar
Buildfile: E:\AutomationProject\Training\SAF\test-output\build.xml

clear:
BUILD SUCCESSFUL
Total time: 0 seconds
```

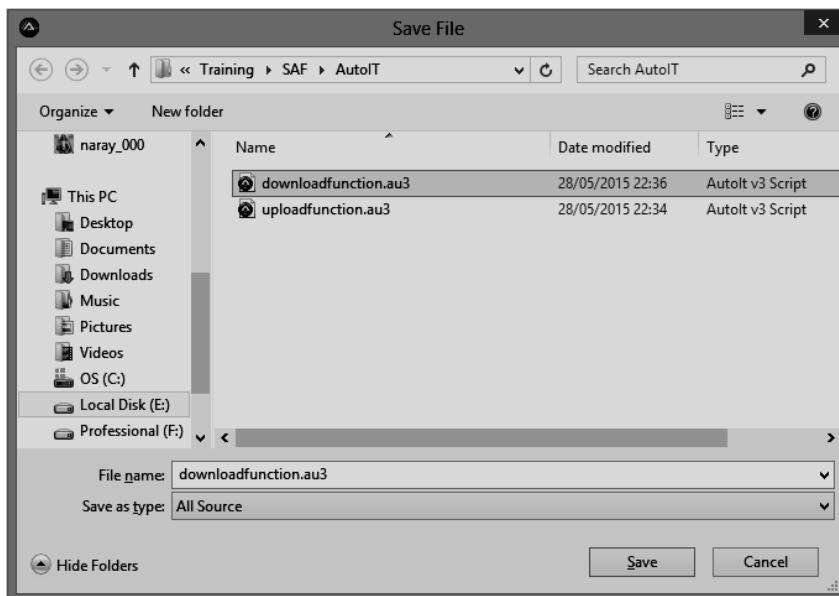


## Software Automation Testing Secrets Revealed



```
File Edit Search View Tools Options Language Buffers Help
1 ;MsgBox(0,"Parameters",$CmdLine[1]) ;Sending parameters through command line
2
3 FileCopy("C:\W3C.png", "C:\\" & $CmdLine[1] & ".png") ;Make a duplicate copy of the file, as same file name is not accepted
4
5 sleep(3000) ;delay for 3 seconds for the file to create
6
7 WinWaitActive("#$32770", "Choose File to Upload", 10)
8
9 WinFlash("Choose File to Upload", "", 4, 500) ; Just to Flash the window
10
11 ControlSetText("Choose File to Upload", "", "[CLASS>Edit: INSTANCE:1]", "C:\\" & $CmdLine[1] & ".png")
12
13 ControlClick("Choose File to Upload", "", "[CLASS:Button; INSTANCE:2]");
```

l=1 co=81 INS (CR+LF)



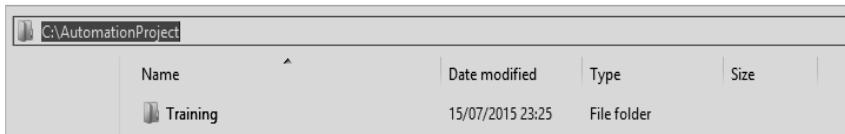
Related videos to understand on Selenium Automation Framework:

<https://www.youtube.com/user/seleniumcoaching>  
<https://www.youtube.com/watch?v=yRHyCgy6fnM>  
<https://www.youtube.com/watch?v=-eJ2cZXyJ0E>  
[https://www.youtube.com/watch?v=O\\_9lThgxBEI](https://www.youtube.com/watch?v=O_9lThgxBEI)

## Steps to use the existing or ongoing Selenium Automation project in Eclipse

### Step1:

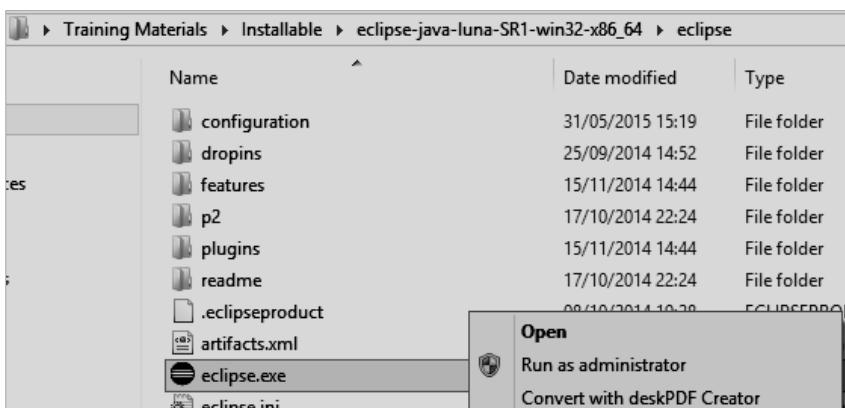
Place the AutomationFramework folder in C: drive like this:



### Step2:

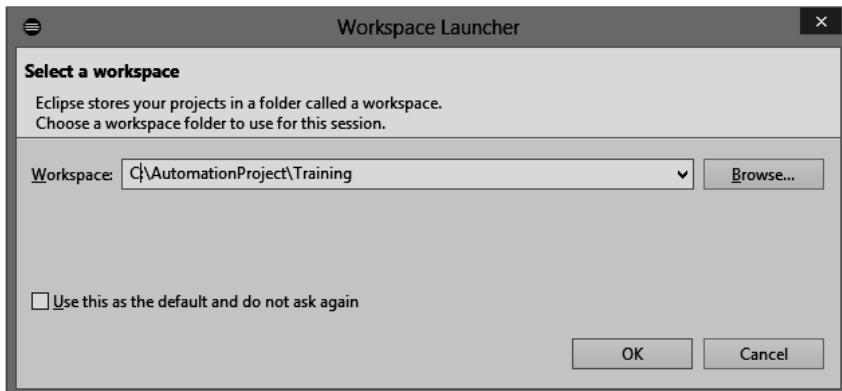
Launch Eclipse from the folder it has been installed (Unlike other software applications, Eclipse needed to be launched by exe every time to open and access whereas other software available in shortcuts or different links to access while launching them).

Open eclipse.exe (preferable Eclipse Luna):



### Step3:

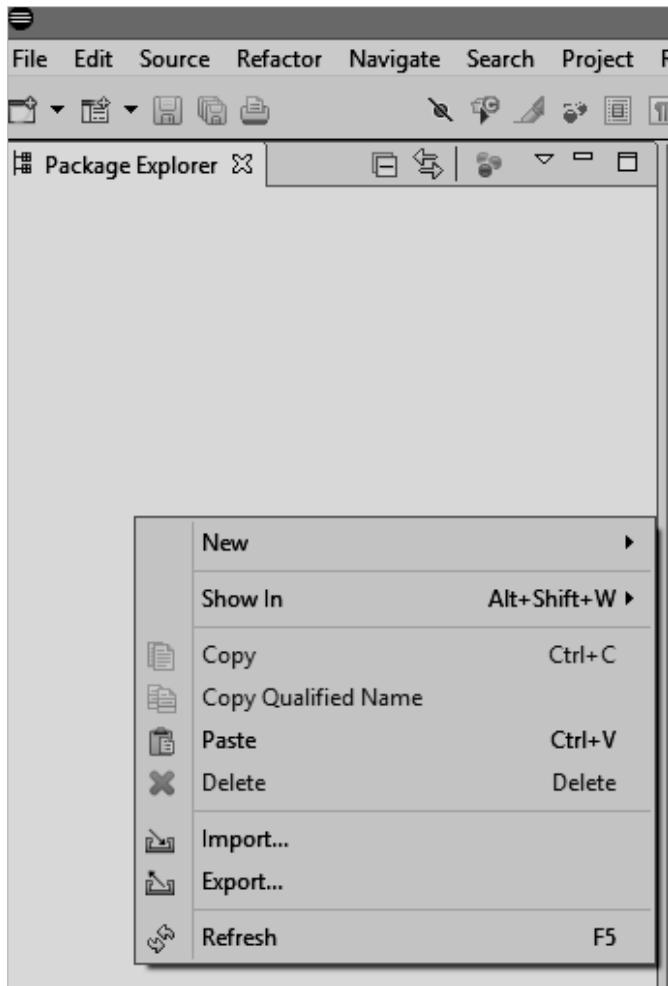
Choose the folder of the Automation Framework folder and click on OK button:



Eclipse Luna launches the screen:



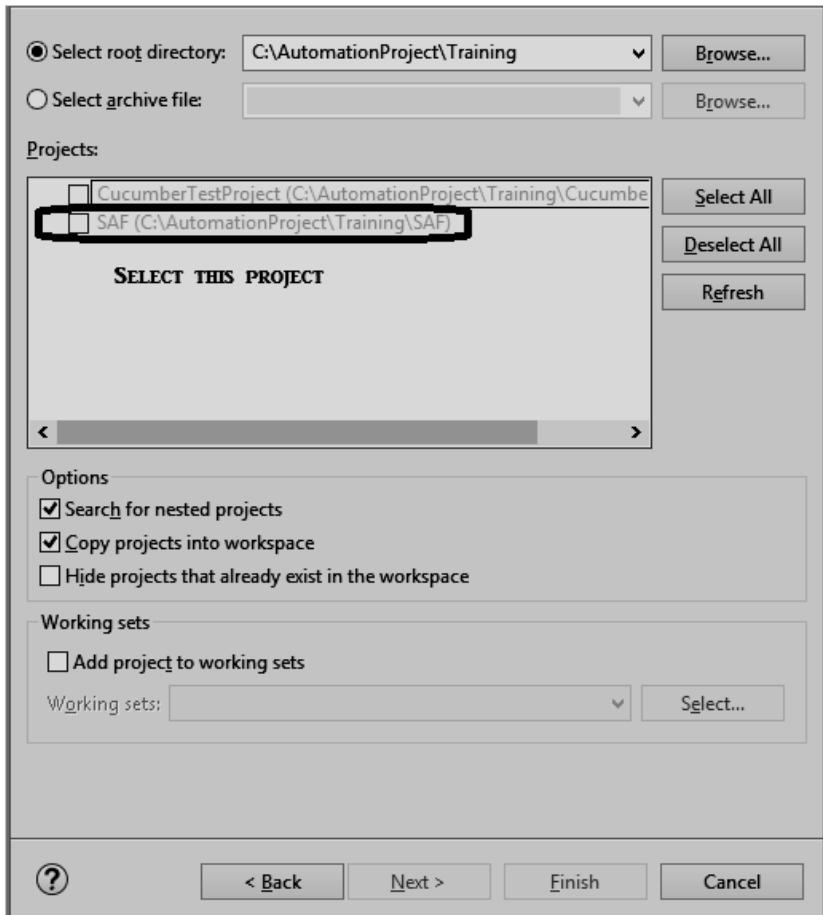
Once opened, click on Workbench button and navigate to the workbench, then right click on the Package Explorer and choose Import:



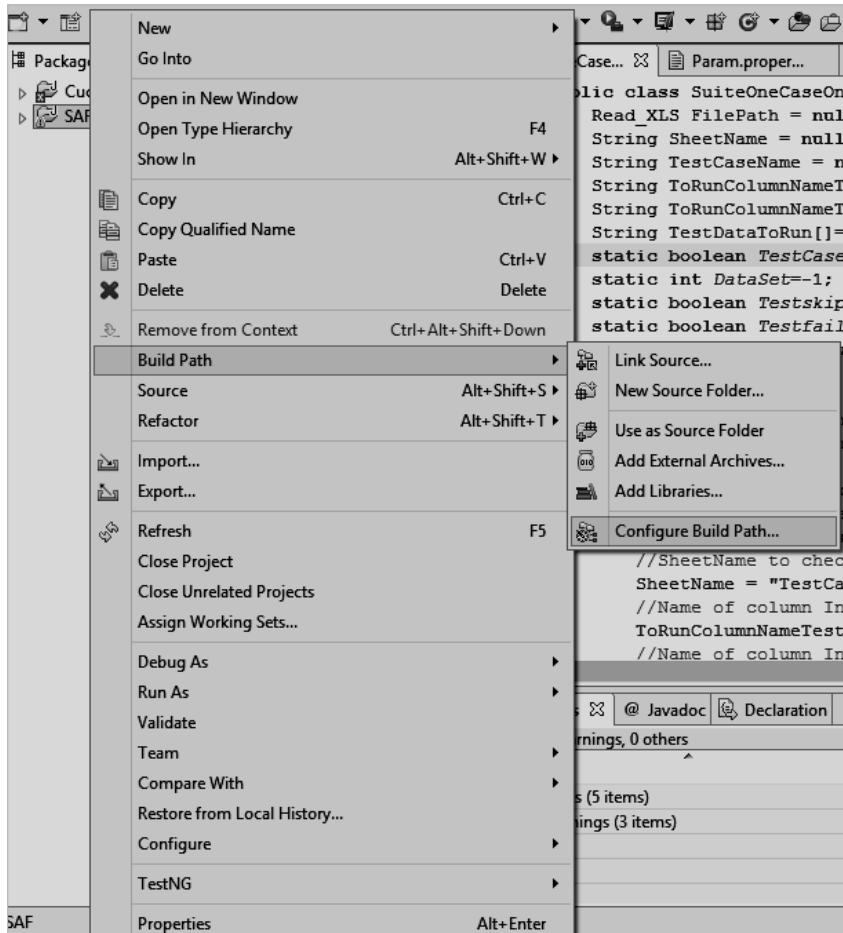
Choose Existing Project:



Click on the Browse button to choose the project and select both the checkboxes to search and copy the existing projects. Choose SAF project in the list and click on OK Button:



Once the SAF project is visible in Package Explorer, navigate to Configure Build Path as shown below:

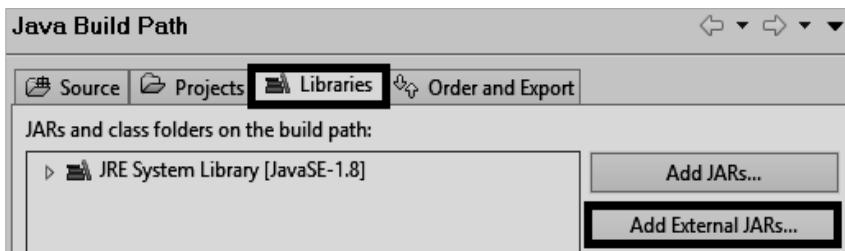


Right click on the folder SAF, click on Build Path and click on Configure Build Path:



In Properties of SAF, click on the Libraries tab and then click on Add External JARs button:

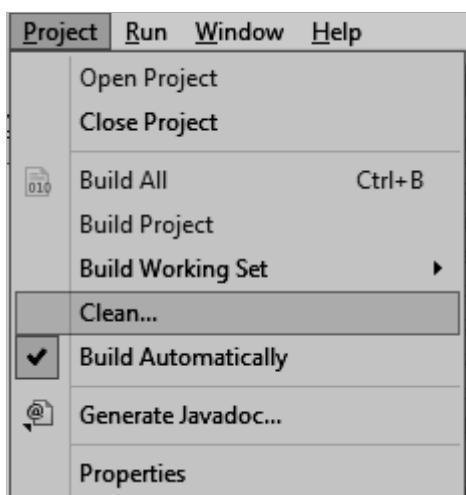
Note: Make sure that the Libraries tab is clicked and then Add External JARs button is clicked from there.



(Follow the same steps mentioned in this section to add all the external JARs.)

Final steps:

Click on Project--> Clean



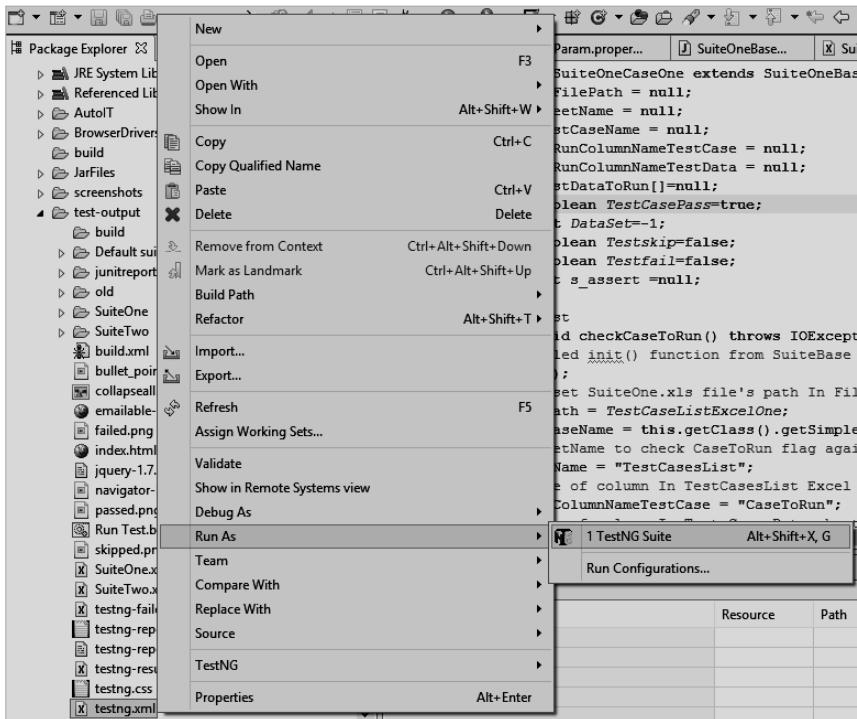
Click on Build Automatically or Build All (If it is enabled).

Now the project should be ready to execute.

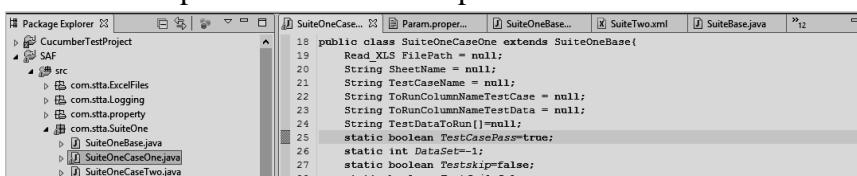
**Note:**

Where to start the execution: Step 'Test Execution through TestNG' in this chapter.

SAF-->testoutput folder-->testng.xml file-->Right click-->Run As-->TestNG Suite

**Where to start editing the scripts?**

Navigate to SAF-->src-->com.stta.SuiteOne-->The Java Files where the scripts are needed to be updated.



## **What should be the first step to review the code?**

Start looking at param.properties for the URL and SuiteOneCaseOne for the navigation of the webpage while testing.

### **References:**

What if there is an error while installing and configuring the Selenium project in Eclipse?

<http://stackoverflow.com/questions/2638016/why-no-projects-found-to-import>

How to setup?

<http://www.assertselenium.com/eclipse-2/how-to-setup-a-webdriver-project-in-eclipse/>

<http://stackoverflow.com/questions/18492668/how-to-execute-a-selenium-test-in-java>

### **Need certifications in Selenium?**

Useful links:

<http://www.vskills.in/certification/Testing/Certified-Selenium-Professional>

<https://www.quora.com/Is-there-any-certification-in-selenium>

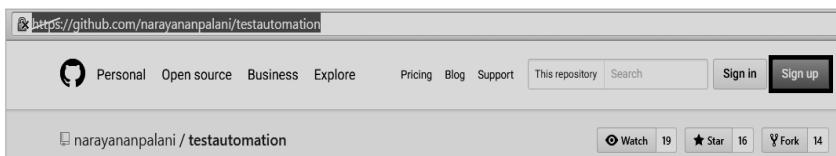
### **Download Selenium projects from GitHub**

In previous chapters, it has been explained on how to construct a TDD (test-driven development)-based hybrid Selenium framework. If one or few links is broken while downloading or if there is any problem in downloading one or many tools while performing installation, it is intricate to accomplish the framework design. In order to prevent such issues from designing the framework, entire developed framework has been made available from:

<http://github.com/narayananpalani/testautomation>

How to download and make use of such developed framework for scripting has explained in the following section:

Once page is opened, click on signup to register on GitHub Page:



Enter relevant details on GitHub registration to create user ID and password:

https://github.com/join?source=header-repo

**Step 1:** Set up a personal account    **Step 2:** Choose your plan    **Step 3:** Tailor your experience

**Create your personal account**

**Username**  
[Input Field]  
This will be your username — you can enter your organization's username next.

**Email Address**  
[Input Field]  
You will occasionally receive account related emails. We promise not to share your email with anyone.

**Password**  
[Input Field]  
Use at least one lowercase letter, one numeral, and seven characters.

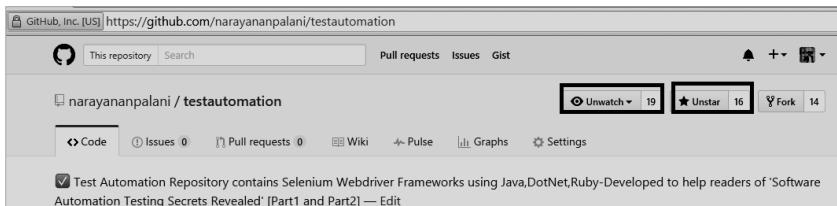
By clicking on "Create an account" below, you are agreeing to the Terms of Service and the Privacy Policy.

**Create an account**

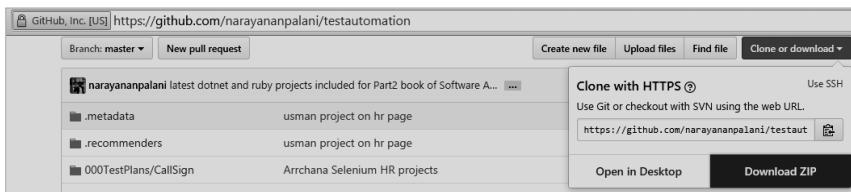
**You'll love GitHub**

- Unlimited collaborators
- Unlimited public repositories
- ✓ Great communication
- ✓ Frictionless development
- ✓ Open source community

Once logged onto GitHub Page, click on WATCH and STAR to make sure you are following the project updates; when new projects are added to the repository, 'watch'ed user IDs get notified on the changes:



Click on Clone or download and choose Download ZIP to download the entire Automation Frameworks:



Alternatively, this Git project can be cloned and performed Git Pull Action to update your local repository with up-to-date scripts. Once projects are downloaded, there are four tools which are required to access the automation frameworks and below, the section will help in downloading and configuring such tools.

## **Install Java, Eclipse, Cucumber, TestNG**

### **Java Installation**

#### **Install JAVA:**

Java programming is a famous and important programming language in the testing industry and testers equipped with Java knowledge are paid high in the job markets! Please download java from following link:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

How to understand whether the Java is installed in the computer or not?

Once the exe file has been downloaded from the website, double click on the exe file and then run the installation procedure of the JAVA program in the computer. Once it is completed, kindly check the C:/ Drive's Program Files folder. This folder should have

'JAVA' folder with JDK and JRE sub-folders. Once these folders are displayed in the computer, it means that the Java installation procedure is completed but the configuration is yet to be completed to ensure that the JAVA files are identified by the computer.

#### JAVA Configuration:

Once Java has been installed successfully, JAVA\_HOME and PATH has to be set up in the system as follows:  
Right click on 'This PC' and select Properties:

Alternatively, navigate to Control Panel and then select Advanced System Settings. In Advanced Tab, click on Environment Variables:

Click on New button and enter the location of JDK folder:

**Note:** Based on your Java version, the folder name will differ. It is better to navigate to the location and then copy the path as below:

Similarly, click on the New button again to enter the path as follows:

**Note:** Make sure that you have mentioned the path including the bin folder as above. You can take a copy of this path just by clicking on the bin folder as below:

Why JAVA\_HOME and PATH have been updated in the Environment Variable:

Whenever the java programs are compiled and executed in the computer, required JAVA files have to be referred during compilation and execution. So the computer needs to know where the JAVA files are located and what is the respective PATH of the bin folder? So the file location is provided through JAVA\_HOME and bin folder location is provided through PATH set up. Once it is completed, it is always a good practice to restart the computer and check the version of JAVA through the Command Prompt. Simply running the command JAVA-VERSION will provide the java version in the command prompt which proves that the JAVA installation has been completed successfully.

An alternative way to set up JAVA\_HOME and PATH is through the Command Prompt. This can be done by launching Command Prompt as follows.

Step1: Press Windows+R (to open RUN)

Step2: Type CMD,press Enter

Step3: In Command Prompt, type' SET JAVA\_HOME = c:/Program Files/Java'

Step4: Type 'SET PATH = %PATH%;%JAVA\_HOME%\bin'

Once JAVA installation and configuration have been completed, it is required to install Eclipse as it acts as a tool to write JAVA codes!

### **Eclipse:**

Download the latest eclipse version from

<http://eclipse.org/downloads/>

Once downloaded, please install it and open Eclipse from the location it is downloaded by clicking on eclipse.exe.

### **Cucumber:**

Install the Cucumber Plugin into Eclipse. This plugin enables certain features in Eclipse – e.g. a nice green cucumber icon for feature files. Also, when you double click on a feature file then it will open the feature in Eclipse rather than opening notepad. In Eclipse, go to help->install new software.

Paste the URL:<http://cucumber.github.com/cucumber-eclipse/update-site>

into the ‘Work with’ text box. Press enter once you have pasted the link. Note that if you are running behind a proxy server and you get a ‘HTTP Proxy Authentication Required’ error, you may need to contact a system administrator to set up your proxy server settings.

See<http://stackoverflow.com/questions/1940984/how-to-specify-eclipse-proxy-authentication-credentials>

Click the Cucumber Eclipse Plugin checkbox and then hit next, next, accept the license agreement and finish.

## TestNG:

Follow steps:

1. Click on Help > Install New software
2. Add the link <http://beust.com/eclipse> in Work with text box
3. TestNG feature will be shown, expand the tree node and then verify the version.
4. Tick the check box of TestNG
5. Click on Next button and then the Finish button.

Steps to use the existing or ongoing Selenium automation project in Eclipse:

## Step1:

Place the AutomationFramework folder in C: Drive like this:

C:\AutomationProject			
	Name	Date modified	Type
	Training	15/07/2015 23:25	File folder

## Step2:

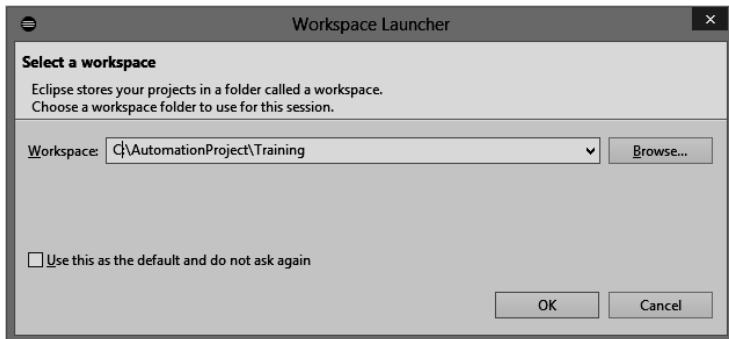
Launch Eclipse from the folder where it has been installed (Unlike other software applications, eclipse needs to be launched by exe, every time to open and access, whereas other software available in the shortcuts or different links to access while launching them).

Open `eclipse.exe` (preferable Eclipse Luna):

C:\Training Materials\Installable\eclipse-java-luna-SR1-win32-x86_64\eclipse			
	Name	Date modified	Type
	configuration	31/05/2015 15:19	File folder
	dropins	25/09/2014 14:52	File folder
	features	15/11/2014 14:44	File folder
	p2	17/10/2014 22:24	File folder
	plugins	15/11/2014 14:44	File folder
	readme	17/10/2014 22:24	File folder
	.eclipseproduct	08/10/2014 10:20	File
	artifacts.xml		File
	<b>eclipse.exe</b>		<b>File</b>
	eclipse.ini		File

### Step3:

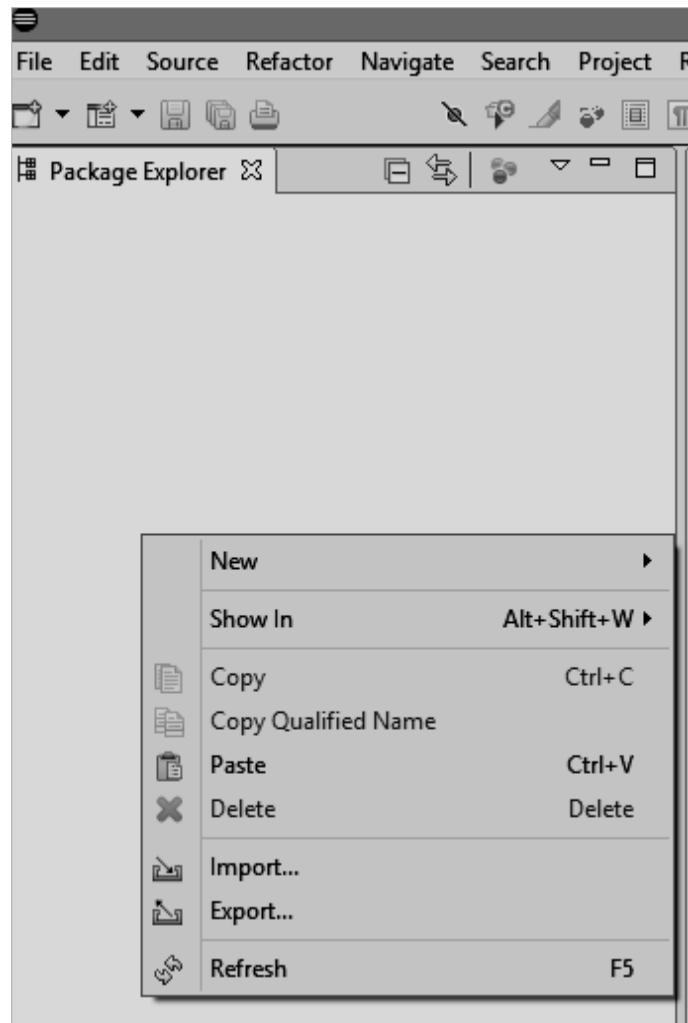
Choose the folder from the automation framework folder and click on the OK button:



Eclipse Luna launches the screen:



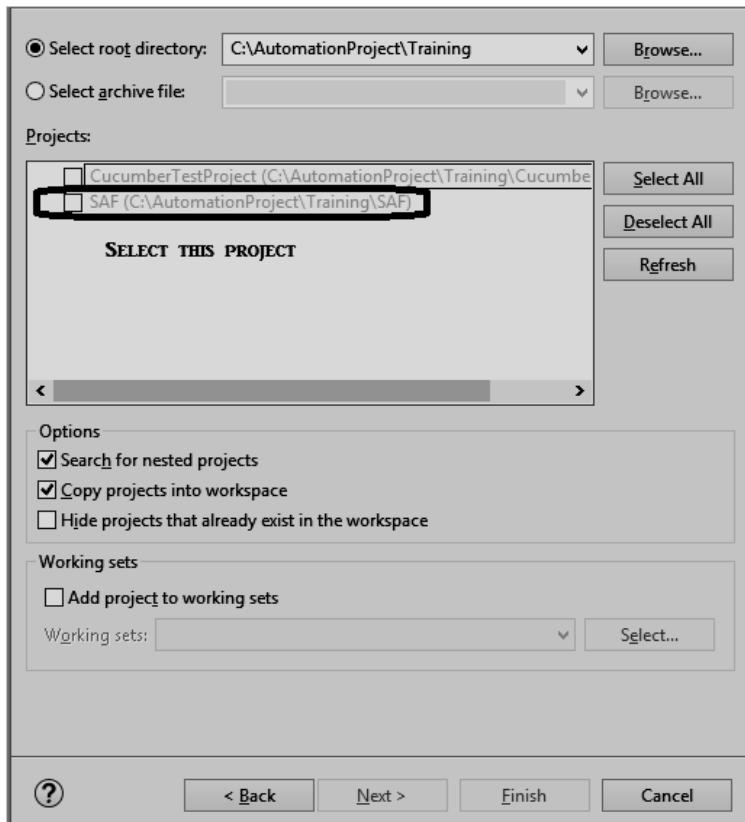
Once opened, click on Workbench button and then navigate to the workbench, then right click on the package explorer and then choose Import:



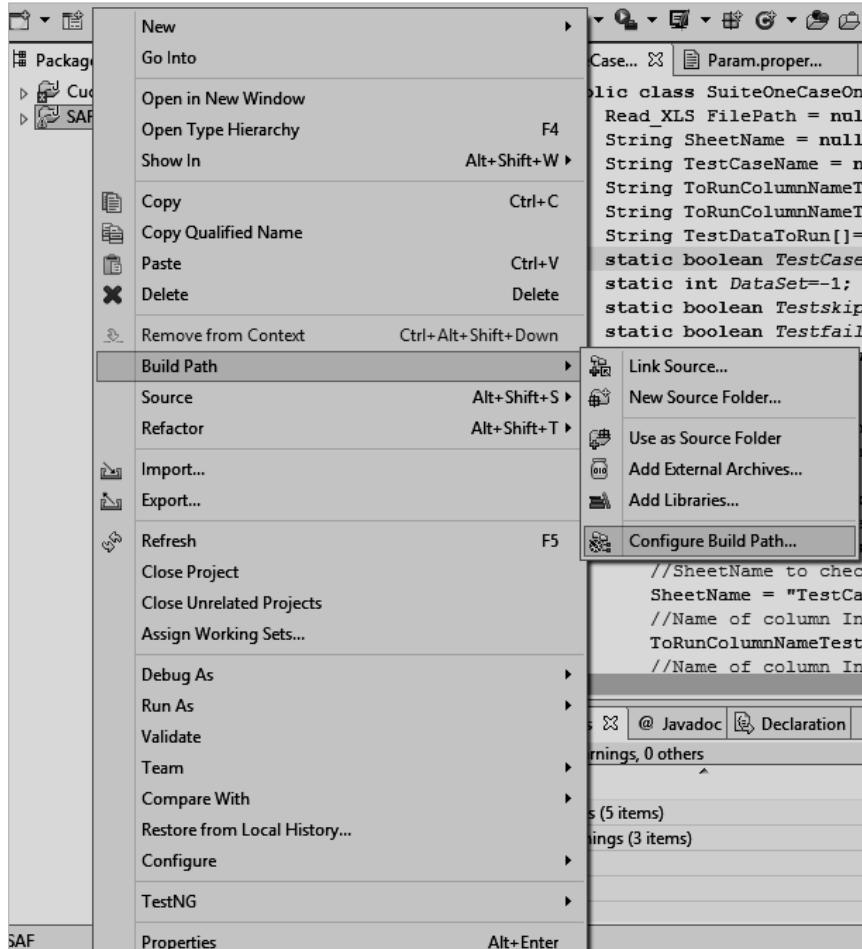
Choose the Existing Project:



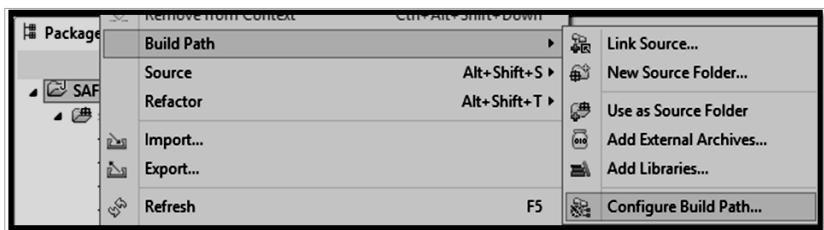
Click on the Browse button to choose the project and then select both the checkboxes to search and copy the existing projects. Choose SAF project in the list and then click on the OK button:



Once the SAF project is visible in Package Explorer, navigate to Configure Build Path as shown below:

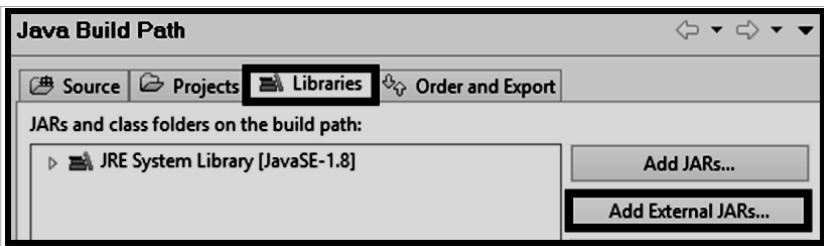


Right click on the folder SAF, then click on Build Path and then click on the Configure Build Path as follows:



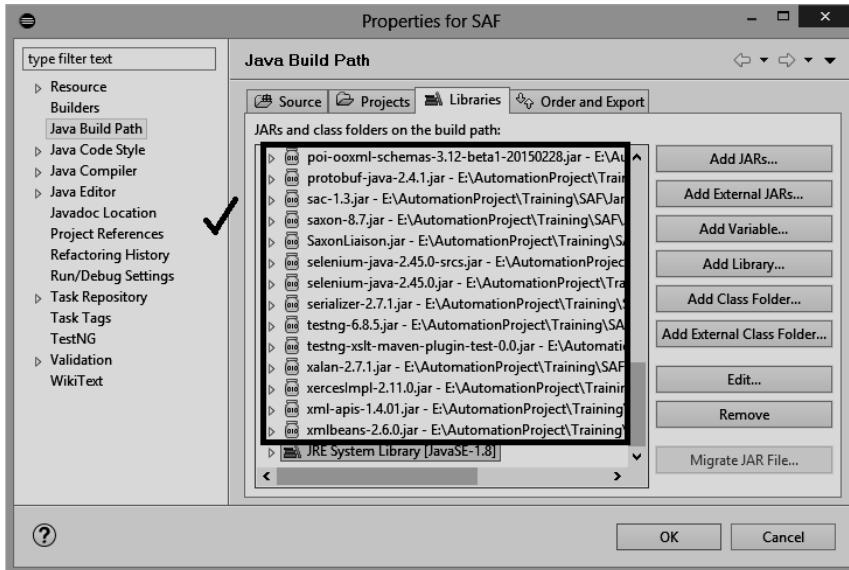
In Properties of SAF, click on the Libraries tab, then click on Add External JARs button as follows:

Note: Make sure that the Libraries tab is clicked and then Add External JARs button is clicked from there.

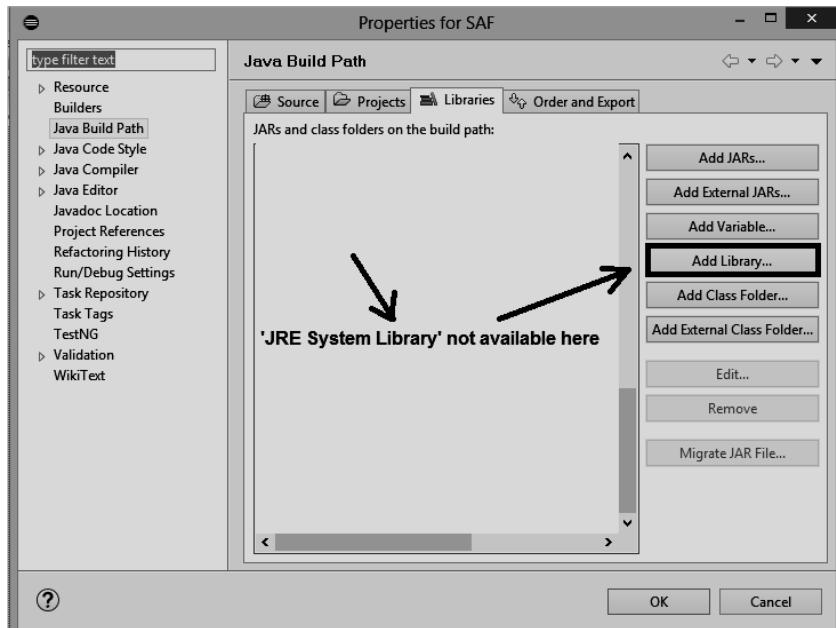


(Follow the same steps which are mentioned in this section to add all the external JARs)

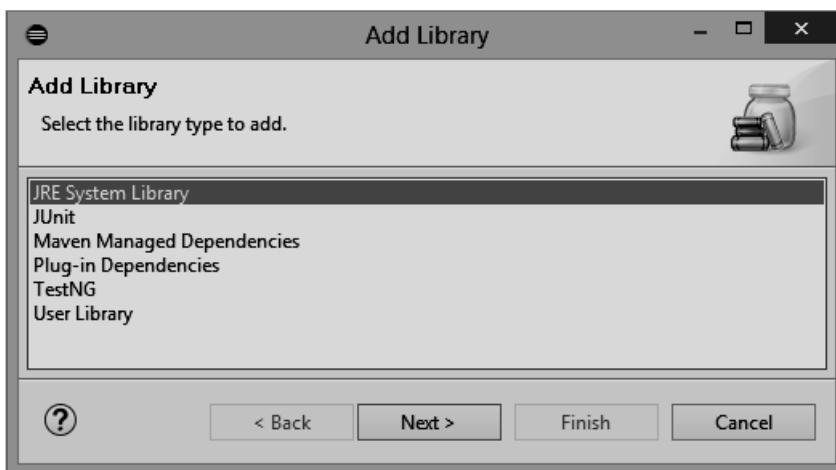
Please select JAR files from the JarFiles folder of the project and then click Open to list the jar files within 'Libraries'. If there are existing JAR files with incorrect path or highlighted with errors, remove them and add fresh files from JarFiles folder.



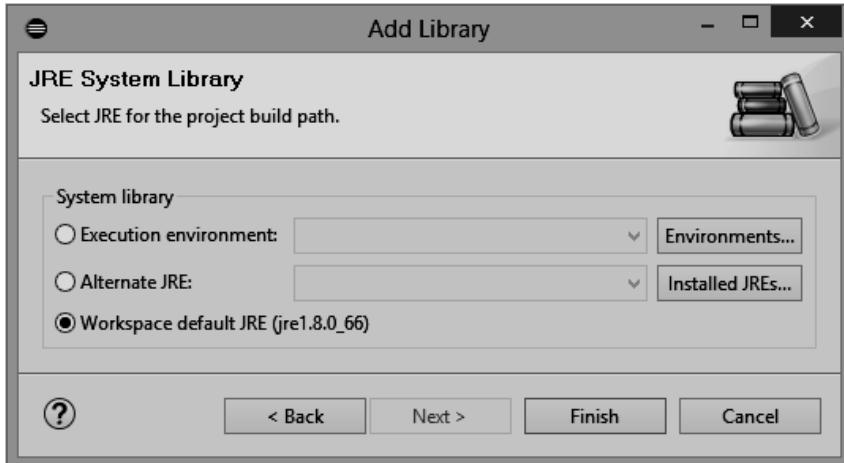
Similarly, if JRE System Library exists, that is good to run the test in java-based framework; if JRE System Library doesn't appear in the Libraries above, click on 'Add Libraries' in the Libraries section, and then choose the default JRE (respective version display as third check box) and then click Next and Finish buttons to display JRE System Library.



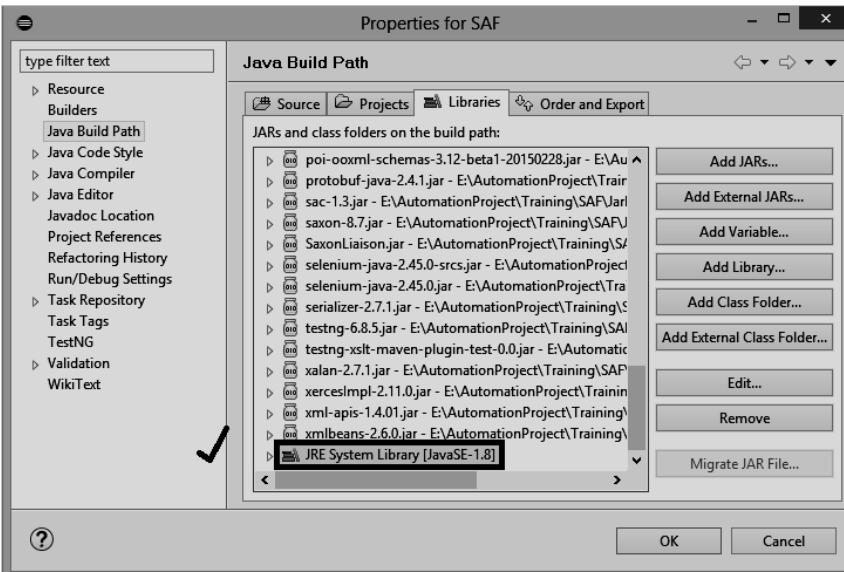
Select the JRE System Library, and then click on the Next button:



Click on the Workspace default JRE or any other relevant option of Java installation details (as per your target machine) and then click on Finish:

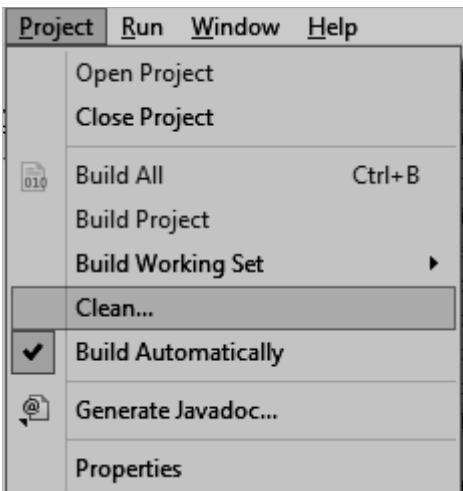


Once the JRE System Library is added without any errors, it will be listed in the Libraries section of the Build Path; right click on Java project-->Build Patch-->Configure Build Path-->Libraries to validate:



## Final Steps:

Click on Project--> Clean



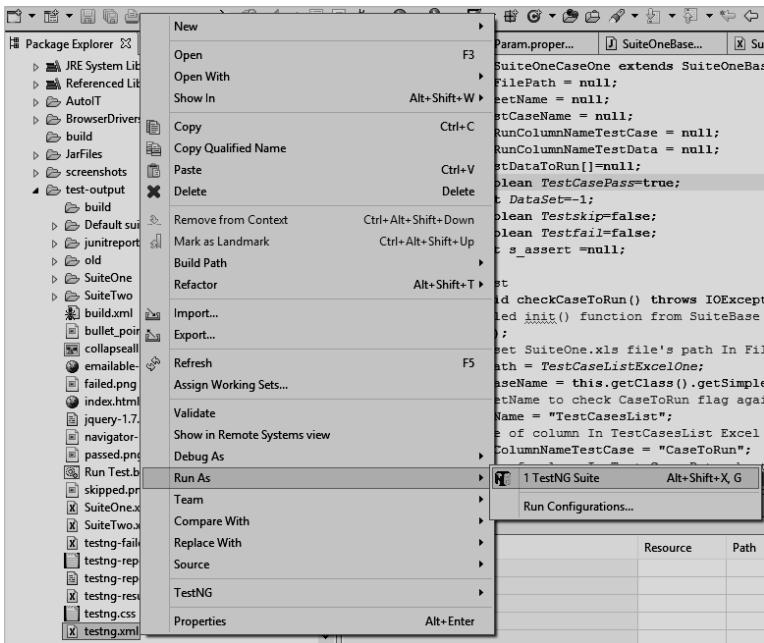
Click on Build Automatically or Build All (If it is enabled)

Now the project should be ready for execution.

### Note:

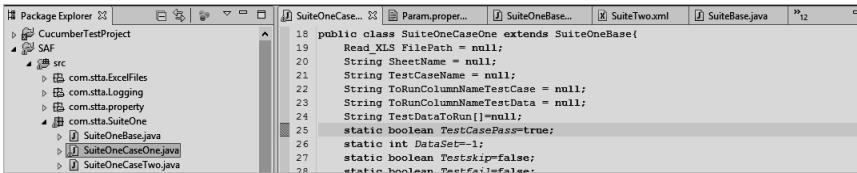
Where to start the execution?: Step 'Test Execution through TestNG' in this chapter

SAF-->testoutput folder-->testng.xml file-->right click-->Run As-->TestNG Suite



Where to start the editing of the scripts?

Navigate to SAF-->src-->com.stta.SuiteOne-->The Java Files where the scripts need to be updated.



What should be the first step to review the code?

Start looking at param.properties for the URL and SuiteOneCaseOne for the navigation of the webpage while testing.

## References:

What if there is an error while installing and configuring the Selenium project in Eclipse?

<http://stackoverflow.com/questions/2638016/why-no-projects-found-to-import>

How to setup?

<http://www.assertselenium.com/eclipse-2/how-to-setup-a-webdriver-project-in-eclipse/>

<http://stackoverflow.com/questions/18492668/how-to-execute-a-selenium-test-in-java>

## Selenium Webdriver Test Script Design Basics in JAVA

Basic Java programming knowledge is required in terms of coding the test scripts and maintaining the automation framework. In the first step, the procedure to initiate a web-based application inside a class of java file is described as follows:

```

1 package com.stta.SuiteOne;
2 import java.io.IOException;
3 import org.openqa.selenium.By;
4 import org.testng.SkipException;
5 import org.testng.annotations.AfterMethod;
6 import org.testng.annotations.AfterTest;
7 import org.testng.annotations.BeforeTest;
8 import org.testng.annotations.DataProvider;
9 import org.testng.annotations.Test;
10 import org.testng.asserts.SoftAssert;
11 import com.stta.utility.Read_XLS;
12 import com.stta.utility.SuiteUtility;
13
14 public class SuiteOneCaseOne extends SuiteOneBase{
15     @Test(dataProvider="SuiteOneCaseOneData")
16     public void SuiteOneCaseOneTest() {
17         // Enter Test Scripts Here
18     }
19

```

Every test script has to be started with a package name (row 1) followed by a set of imports that is required for the particular class in the program. For example, Selenium is required to perform automation test, so org.openqa.selenium.By is required to be mapped in the import. This is to navigate to org\openqa\Selenium\ to refer relevant files required for the automation. Similar to Selenium, TestNG (Automation Framework Tool)-related imports has to be mentioned as part of the imports.

As per the generic nature of the Java program, the class has to be defined by the class name (note that the file name and the class name have to be the same).

Refer to the section (Row 17) where the test scripts have to be written in order to invoke the application.

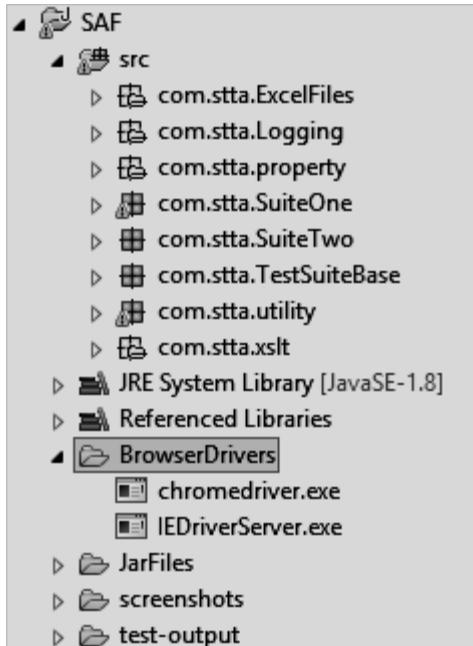
## Launch Webdriver

```
Step1: Sample scripts on web driver initiation  
//To Initialize Firefox web browser  
WebDriver driver = new FirefoxDriver();  
//To set time out 10 seconds.  
driver.manage().timeouts().implicitlyWait(10,  
TimeUnit.SECONDS);  
//To maximize the browser.  
driver.manage().window().maximize();  
//To navigate to URL.  
driver.get("http://www.abctest.com");  
//To close the browser.  
driver.close();
```

The above script will launch the respective web page and the close immediately. This script is sufficient if the browser requirement to test is only Mozilla Firefox. But most of the test projects are in need of a cross browser testing. It means that the script written in Selenium has to be run in either Internet Explorer or Firefox or any other given browser as per the requirement of the application.

Selenium supports Firefox, Chrome and Internet Explorer widely and it is easy to run the scripts using cross browsers in Selenium framework.

As an initial implementation, browser drivers have to be placed in BrowserDrivers folder in order to use the respective driver based on the browser selection:



Firefox-based driver is not required to be provided as an external driver (similar to chrome and IE) as Firefox is supported by Selenium Webdriver by default.

#### Prerequisite1 for Step2:

Create a Param.properties file and then update the web URL in the properties file. This will facilitate the script to launch the URL whenever required and it is a one-time activity to change the URL when there is a URL change and amendment.

	<pre> 1 2 3 #Web Application URL 4 siteURL=http://www.abctest.com 5 6 7 8 #set Mozilla or Chrome or IE 9 testBrowser=Mozilla </pre>
--	---

Whenever the URL undergoes changes for project needs, changes in every part of the scripts generated is not required. Just the

param.properties has to be updated with the right URL and the script is ready!

Similarly, the testBrowser represents the browser centrally from param.properties. If the entire script has to be run via Mozilla, this testBrowser has to be updated before the test Run.

Next step is to create reusable functions (classes) to handle the browsers as given below:

### Prerequisite1 for Step2 (Refer SuiteBase.Java file):

When the browser is already opened in the system, the script has to pick it up automatically for the test execution. This has to be facilitated with the help of the scripts as given below:

```
59     public void loadWebBrowser(){
60         //Check If any previous webDriver browser Instance Is exist then run new test In that existing webDriver browser Instance.
61         if(Param.getProperty("testBrowser").equalsIgnoreCase("Mozilla") && ExistingmozillaBrowser!=null){
62             driver = ExistingmozillaBrowser;
63             return;
64         }else if(Param.getProperty("testBrowser").equalsIgnoreCase("chrome") && ExistingchromeBrowser!=null){
65             driver = ExistingchromeBrowser;
66             return;
67         }else if(Param.getProperty("testBrowser").equalsIgnoreCase("IE") && ExistingIEBrowser!=null){
68             driver = ExistingIEBrowser;
69             return;
70         }
    }
```

Similarly, if no other driver exists in the system, automation framework should launch the browser based on the selection at param.properties as given below:

```
73     if(Param.getProperty("testBrowser").equalsIgnoreCase("Mozilla")){
74         //To Load Mozilla driver Instance.
75         driver = new FirefoxDriver();
76         ExistingmozillaBrowser=driver;
77         Add_Log.info("Firefox Driver Instance loaded successfully.");
78     }
79     else if(Param.getProperty("testBrowser").equalsIgnoreCase("Chrome")){
80         //To Load Chrome driver Instance.
81         System.setProperty("webdriver.chrome.driver", System.getProperty("user.dir")+"//BrowserDrivers//chromedriver.exe");
82         driver = new ChromeDriver();
83         ExistingchromeBrowser=driver;
84         Add_Log.info("Chrome Driver Instance loaded successfully.");
85     }
86     else if(Param.getProperty("testBrowser").equalsIgnoreCase("IE")){
87         //To Load IE driver Instance.
88         System.setProperty("IEDriverServer.driver", System.getProperty("user.dir")+"//BrowserDrivers//IEDriverServer.exe");
89         driver = new InternetExplorerDriver();
90         ExistingIEBrowser=driver;
91         Add_Log.info("IE Driver Instance loaded successfully.");
92     }
93
94     driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);
95     driver.manage().window().maximize();
96 }
```

### Step2(Reference: SuiteOneCaseOne.java file):

Same driver initiation can be simplified by using reusable components such as loadbrowser.

Now the scripts required for the execution of the web driver launch at any automation script (like SuiteOneCaseOne.java) contains just two lines of code:

```
loadWebBrowser();
driver.get(Param.getProperty("siteURL"));
```

## Close the Webdriver

When no Webdriver exists before launching of the automation framework, then the script should launch a new browser,test the application and finally close the browser. In case there is an existing browser, it simply has to be reused and it should not get closed at the end.

```
98     public void closeWebBrowser() {
99         driver.close();
100        //null browser Instance when close.
101        ExistingchromeBrowser=null;
102        ExistingmozillaBrowser=null;
103        ExistingIEBrowser=null;
104    }
```

Read the data from the excel sheets for the data driven framework by locating the path of excel files as given below.

```
32     public void init() throws IOException{
33         //To Initialize logger service.
34         Add_Log = Logger.getLogger("rootLogger");
35         |
36         //Please change file's path strings bellow If you have stored them at location other than bellow.
37         //Initializing Test Suite List(TestSuiteList.xls) File Path Using Constructor Of Read_XLS Utility Class.
38         TestSuiteListExcel = new Read_XLS(System.getProperty("user.dir")+"\\zsg\\com\\gtsa\\ExcelFiles\\TestSuiteList.xls");
39         //Initializing Test Suite One(SuiteOne.xls) File Path Using Constructor Of Read_XLS Utility Class.
40         TestCaseListExcelOne = new Read_XLS(System.getProperty("user.dir")+"\\zsg\\com\\gtsa\\ExcelFiles\\SuiteOne.xls");
41         //Initializing Test Suite Two(SuiteTwo.xls) File Path Using Constructor Of Read_XLS Utility Class.
42         TestCaseListExcelTwo = new Read_XLS(System.getProperty("user.dir")+"\\zsg\\com\\gtsa\\ExcelFiles\\SuiteTwo.xls");
```

The main instruction for any Selenium-based automation test is that the excel sheets should be closed completely before running the tests. The excel sheets which are opened at run time lead to errors in writing the data back to the excel sheets.

For further reading and practice, download the projects at  
<https://github.com/narayananpalani/testautomation>

## Jason Phantom Ghost Driver Test Automation

*Programming language used in this section: Java Programming*

*More code examples of this section are available from github at:*

*<https://github.com/narayananpalani/testautomation/tree/master/WDDF>*

PhantomJS is not a test framework but it facilitates the best test runner to run our tests! Headless testing of web applications (which means testing without using any web browser!) is possible with the help of Jason Phantom Ghost Driver.

### Headless browser testing

This is the latest technology and the fastest among the test automations, where as this is not performed by any GUI-based web browsers. Since the tests are running in the background, they run in rapid speed with Selenium. This headless browser is a pure java solution which is designed based on Rhino Automation Engine and is not dependent on any browser or operating system! Tests run in the background and provide results to the tester which is faster than IE/Firefox/Chrome-based tests.

### Install PhantomJS

Download Jason Phantom through  
<http://phantomjs.org/download.html>

Learn <ul style="list-style-type: none"><li>o <a href="#">Quick Start</a></li><li>o <a href="#">Headless Testing</a></li><li>o <a href="#">Screen Capture</a></li></ul>	<p><b>Windows</b></p> <p>Download <a href="#">phantomjs-2.0.0-windows.zip</a> (19.4 MB) and extract (unzip) the content.</p>
---	--

Once installed, clicking on the exe file inside the bin folder displays this command prompt:



Download the JAR (phantomjsdriver-1.2.1.jar) to use in Eclipse-based projects at

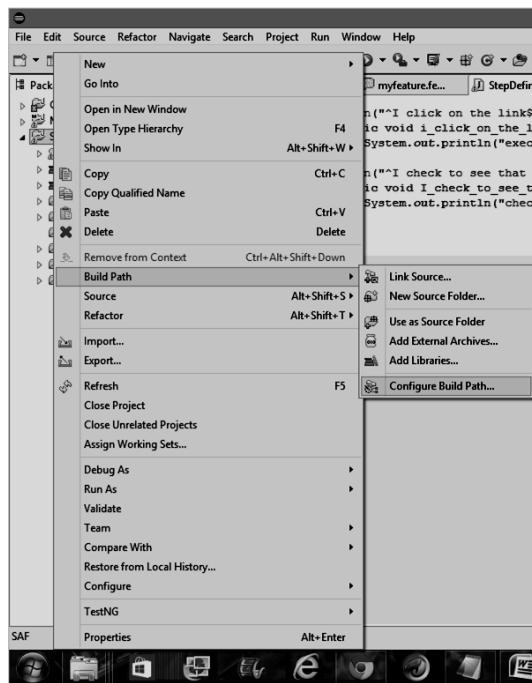
<https://drive.google.com/a/raghava.co.uk/file/d/0B6vnknygMB3IczZQVGQzNTloZ2c/view>

Alternatively, this JAR file can be downloaded from the Installable folder at <https://github.com/narayananpalani/testautomation>

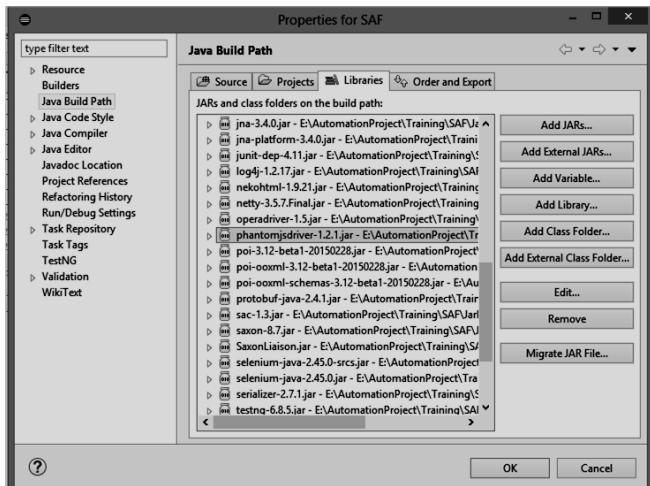
Once the JAR file is downloaded, add it to the build path of the project.

How to add in the build path?

Right click on the project and Build Path>Configure Build Path>



Select Libraries>Add External Jars>Select phantomjsdriver-1.2.1.jar. Once added, it looks like the screenshot as shown below:



Once the phantomJs has been configured, go to your project java files and then create a java file phantomjs.java and then paste the below script (and make sure that the package name is edited in the first line based on the package name in your project).

```
package com.stta.SuiteOne;
```

```
import java.io.File;
import java.io.IOException;
import java.util.concurrent.TimeUnit;

import org.apache.commons.io.FileUtils;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.phantomjs.PhantomJSDriver;
import org.openqa.selenium.phantomjs.PhantomJSDriverService;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;
```

```

publicclass phantomjs {
    WebDriver driver;

    @BeforeTest
    publicvoid setup() throws Exception {
        //Set phantomjs.exe executable file path using DesiredCapabilities.
        DesiredCapabilities capability = new DesiredCapabilities();
        //Make sure that the path of exe file has been provided here with /
        for  //every folder like the below:
            capability.setCapability(PhantomJSDriverService.PHANTOMJS_
EXECUTABLE_PATH_PROPERTY,
            "C:/Users/naray_000/Desktop/Training
Materials/Installable/phantomjs-2.0.0-windows/phantomjs-2.0.0-
windows/bin/phantomjs.exe");
        driver = new
            PhantomJSDriver(capability);driver.manage().timeouts().implicitly
Wait(15, TimeUnit.SECONDS);
    }

    @Test
    publicvoid phantomTest() throws IOException{
        driver.get("http://www.reed.co.uk/");
        driver.findElement(By.xpath("id('topSignInLink')")).click();
        //Update your registered user name
        driver.findElement(By.xpath("id('topSignInEmail')")).sendKeys("t
est@test.com");
        //Update your registered password
        driver.findElement(By.xpath("id('topSignInPassword')")).sendKeys(
"abcd1234");
        driver.findElement(By.xpath("id('topSignInPanel')/div[4]/div/span/
input")).click();
        driver.findElement(By.xpath("id('keywords')")).sendKeys("Softwa
re Tester");
        driver.findElement(By.xpath("id('location')")).sendKeys("cv313qd
");
    }
}

```

```
driver.findElement(By.xpath("id('homepageSearchButton')")).click();
();
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript("javascript:window.scrollBy(250,350)");
driver.findElement(By.xpath("id('jobSection27977887')/div/header
/div/h3/a")).click();

driver.findElement(By.xpath("id('applyBtn')")).click();
driver.findElement(By.xpath("id('submitBtn')")).click();
//To capture page screenshot and save In D: drive.
File scrFile =
((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
FileUtils.copyFile(scrFile, new File("C:\\Test.jpeg"),true);
}
}
```

Once the JAVA file is added to the project, right click on the java file and Run As, TestNG suite. Since it is a headless browser test, it runs in the background and provides the results once completed!

## Maven Test Automation

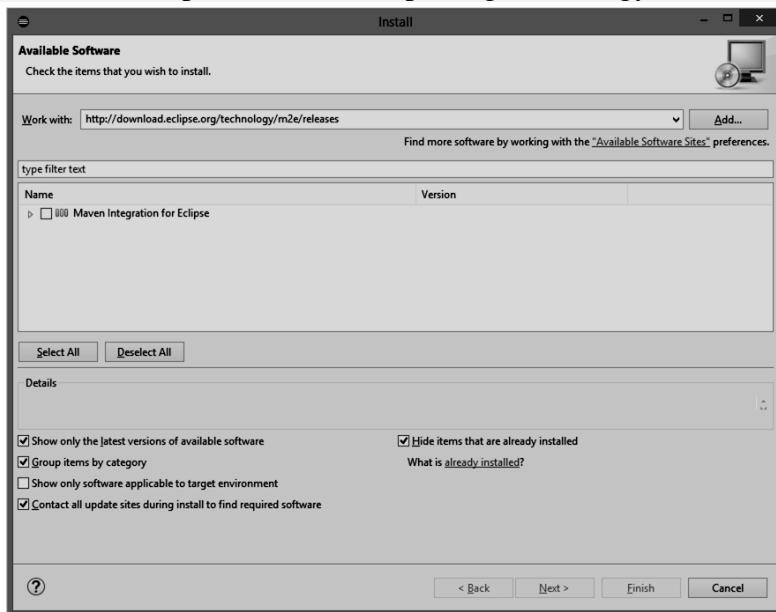
*Programming language used in this section: Java Programming  
 More code examples of this section are available from github at:  
<https://github.com/narayananpalani/testautomation/tree/master/014Maven%20Job%20Portal%20Project%20by%20Jeyabala/MavenProject>*

Apache Maven helps to build the projects and manage the java-based projects as Process Object Model (POM) of Maven is widely used across the IT industry for test automation. Primarily, it is an excellent build automation tool!

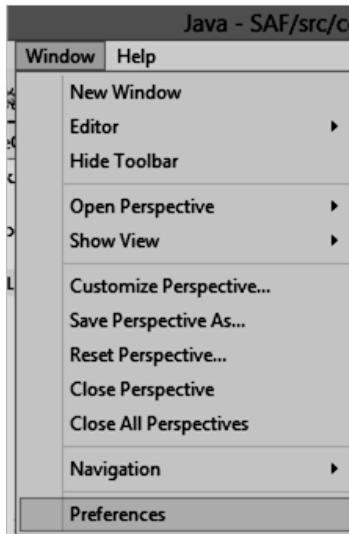
### How to install Maven from Eclipse IDE?

It can be installed in the Eclipse IDE via the Help tab>Install New Software>

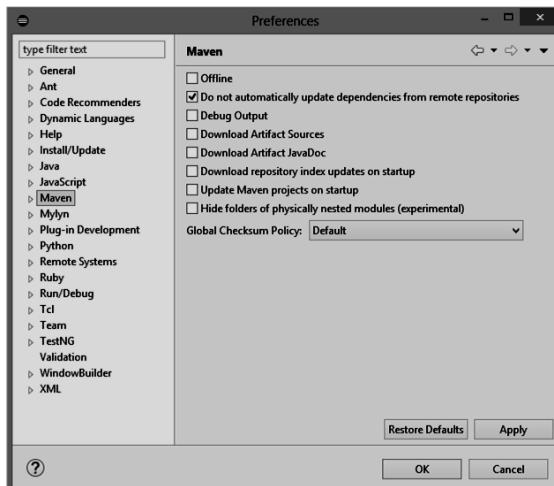
Enter URL as <http://download.eclipse.org/technology/m2e/releases>



Once it has been installed at the Eclipse, it can be verified by the Window tab>Preferences>



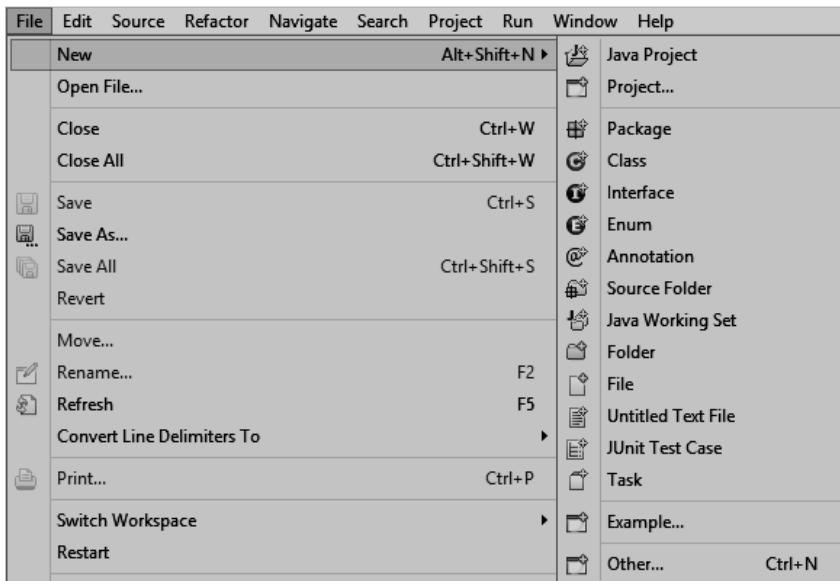
Display Maven in the list as follows:



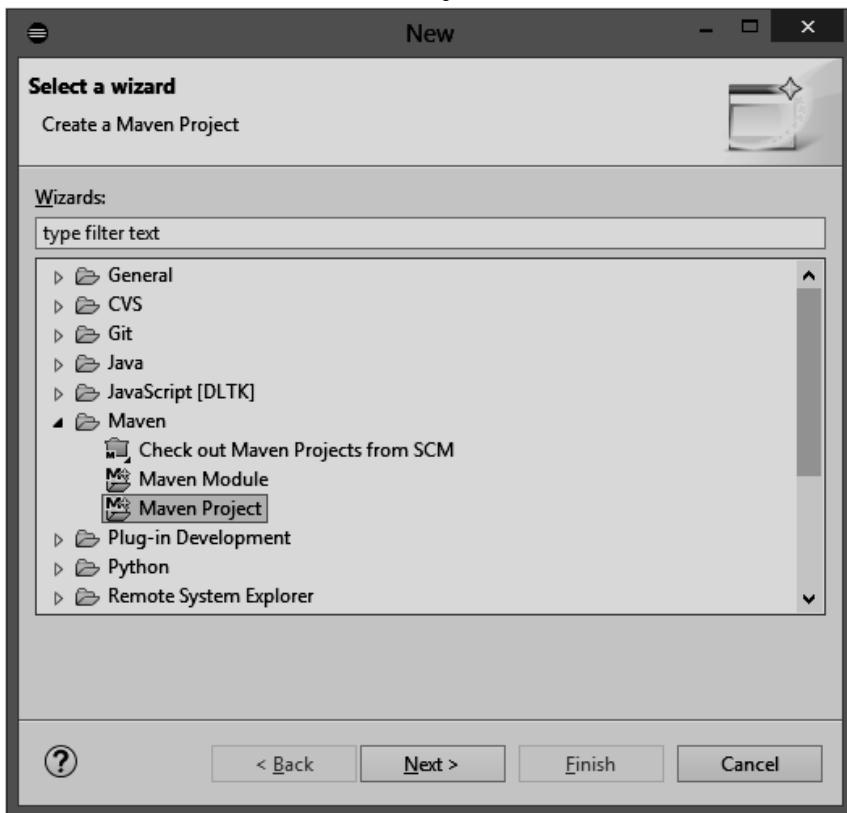
Press OK and then go ahead to create a new maven project for your automation project!

## Create Maven project

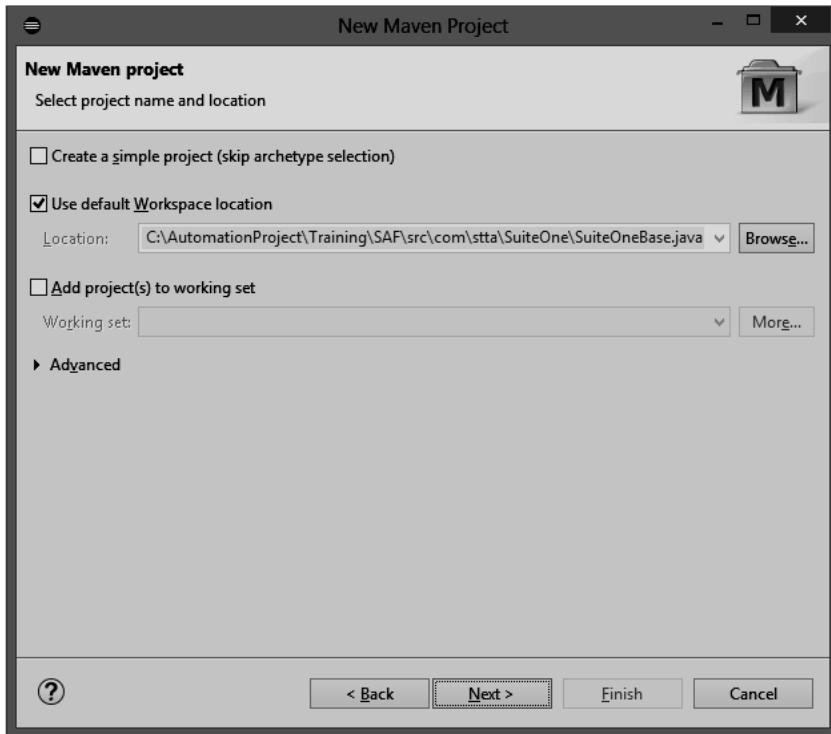
New project can be created by File>New>Other



Select a Maven Folder>Maven Project as follows:

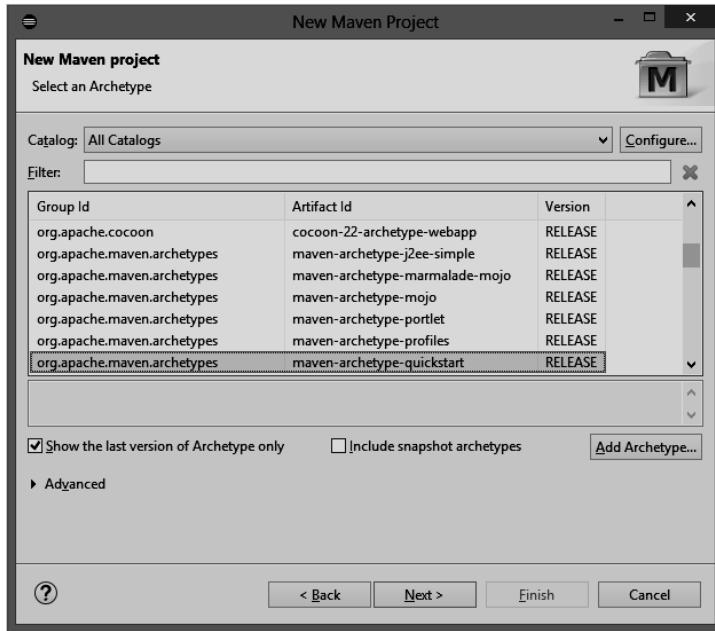


Click on Next in the box as shown below:

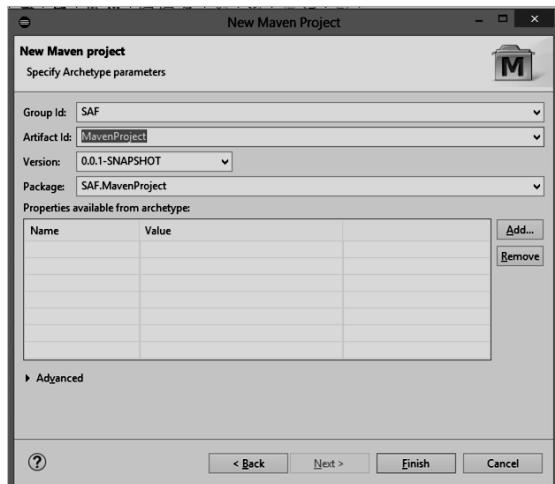


## Software Automation Testing Secrets Revealed

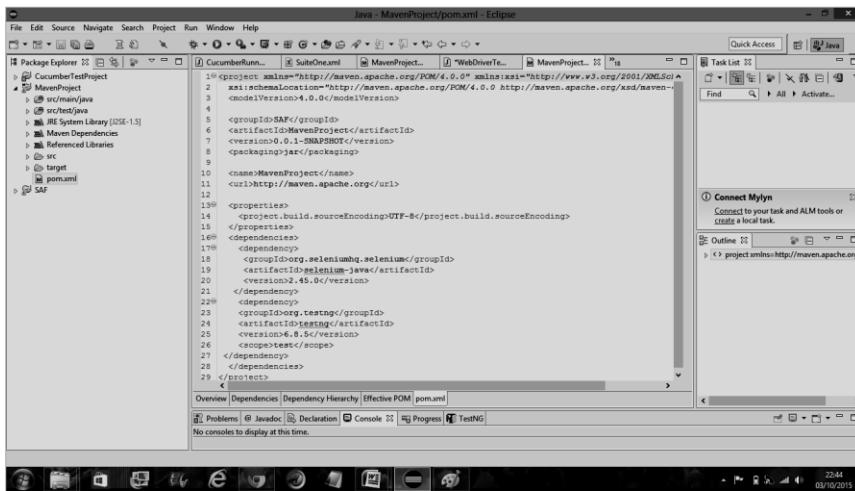
Select QuickStart Artifact ID as shown below and then click on Next:



Enter the required project details as shown below:



Select POM.xml and then navigate to the script to update the Selenium and the TestNG versions which are required for the project:



## Sample POM file

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>SAF</groupId>
  <artifactId>MavenProject</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>MavenProject</name>
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>

```

```

</properties>
<dependencies>
<dependency>
<groupId>org.seleniumhq.selenium</groupId>
<artifactId>selenium-java</artifactId>
<version>2.45.0</version>
</dependency>
<dependency>
<groupId>org.testng</groupId>
<artifactId>testng</artifactId>
<version>6.8.5</version>
<scope>test</scope>
</dependency>
</dependencies>
</project>

```

## Add external jars of Selenium and testNG

Main Selenium Webdriver-based Jar files for Selenium Automation Framework:

The very important test engine of the automation framework is the testing tool. Especially Selenium, which is made up of JAR files, that evolves over a period of time to upgraded versions. So make sure that the latest version of the file has been downloaded from the link given below.

Download the files from <http://docs.seleniumhq.org/download/>

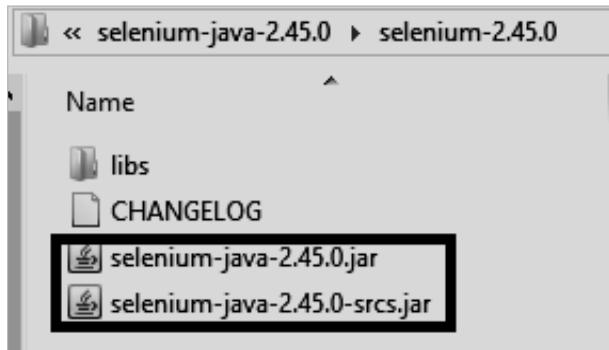
### Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Selenium RC, Selenium Remote Webdriver) or create local Selenium WebDriver script you need to make use of language-specific client drivers. These languages include both 1.x and 2.x style clients.

While language bindings for [other languages exist](#), these are the core ones that are supported by the main project hosted on google code.

Language	Client Version	Release Date	Download	Change log	Javadoc
Java	2.45.0	2015-02-26	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">Javadoc</a>
C#	2.45.0	2015-02-27	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Ruby	2.45.0	2015-02-27	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Python	2.45.0	2015-02-26	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Javascript (Node)	2.45.0	2015-02-26	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>

Once the files are downloaded, make sure that all the jar files from this file (including the jar files from the sub-folders of this file) are pasted into the JarFiles folder which is created for the automation framework.



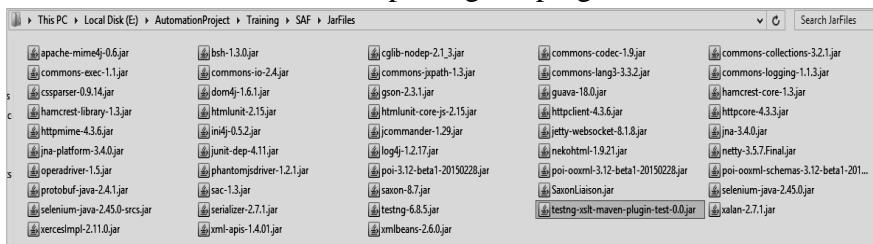
### TestNG Plugin to use the XSLT Reports:

TestNG is mainly used to run the test scripts which are generated in Selenium. So, this needs to be extended to XSLT reporting jar files using the plugin. So, the reports get the latest test execution status automatically. Download the plugin from the link given below and then paste it to the JarFiles folder as it is an important plugin to capture the test results.

Link:

<https://drive.google.com/file/d/0B6vnknygMB3IdzF4X2taWFRRMVE/edit>

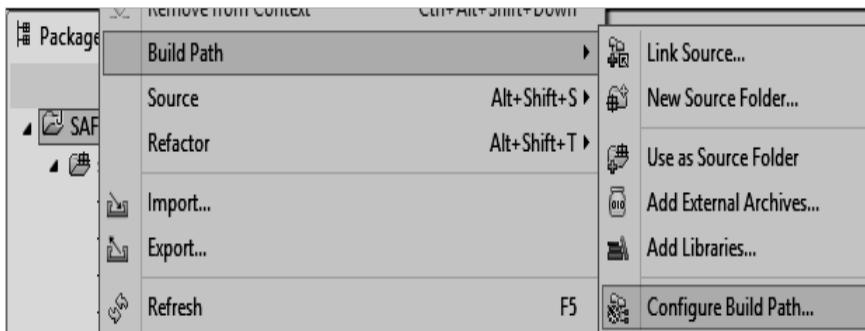
### JarFiles Folder structure after pasting the plugin:



## Project build path configuration for Maven

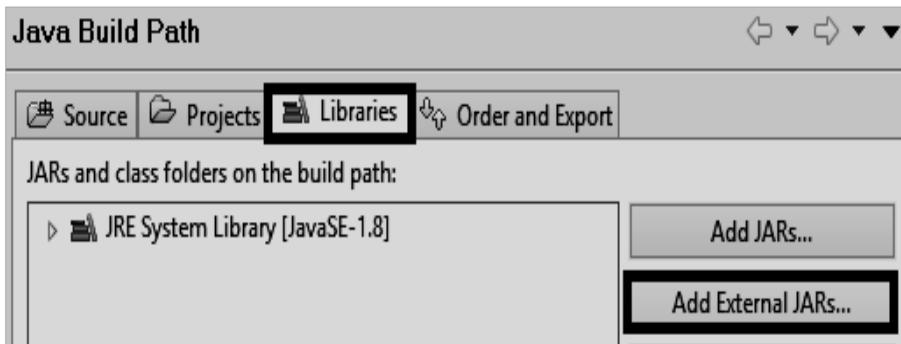
The need of placing all the relevant JAR files into one central location in the automation framework is to access them to configure the project MavenProject.

Right click on the folder MavenProject, click on Build Path and then click on Configure Build Path as follows:

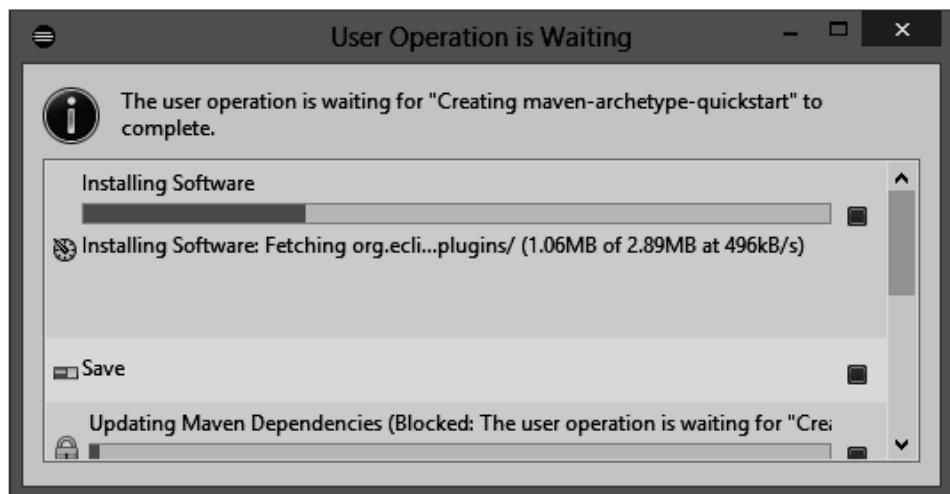
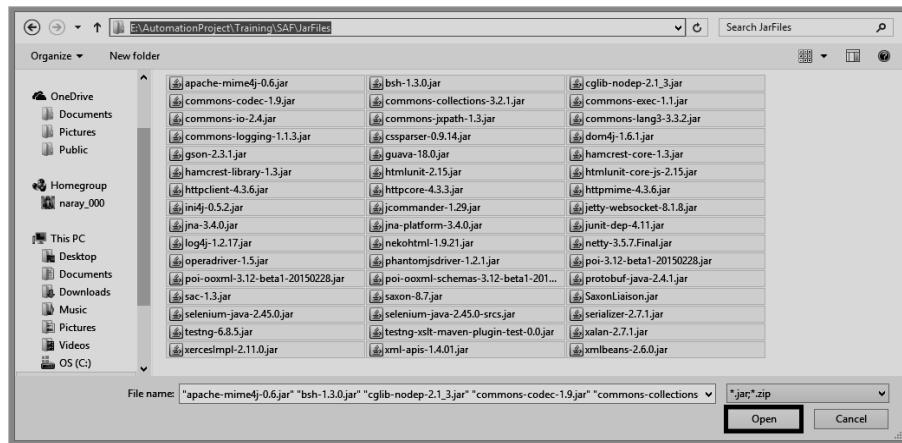


In the Properties of Maven Project, click on the Libraries tab, then click on the Add External JARs button as follows:

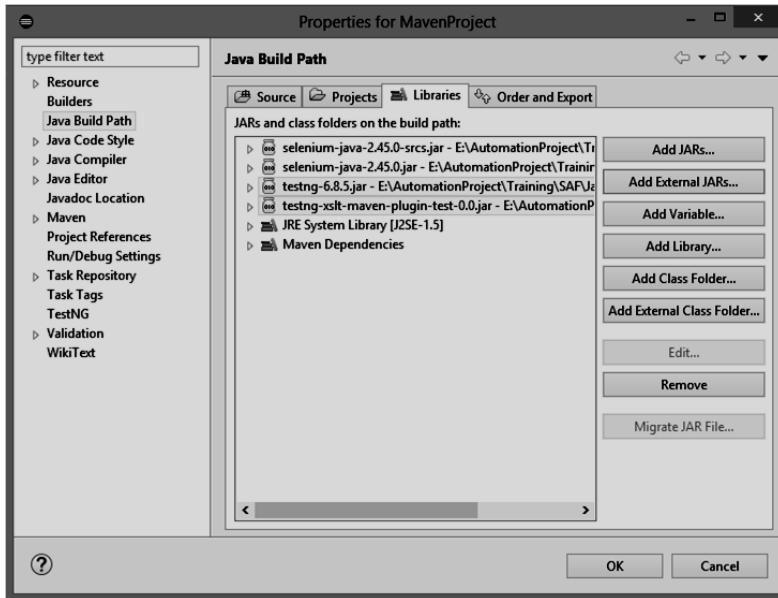
**Note:** Make sure that the Libraries tab is clicked and then the Add External JARs button is clicked from there.



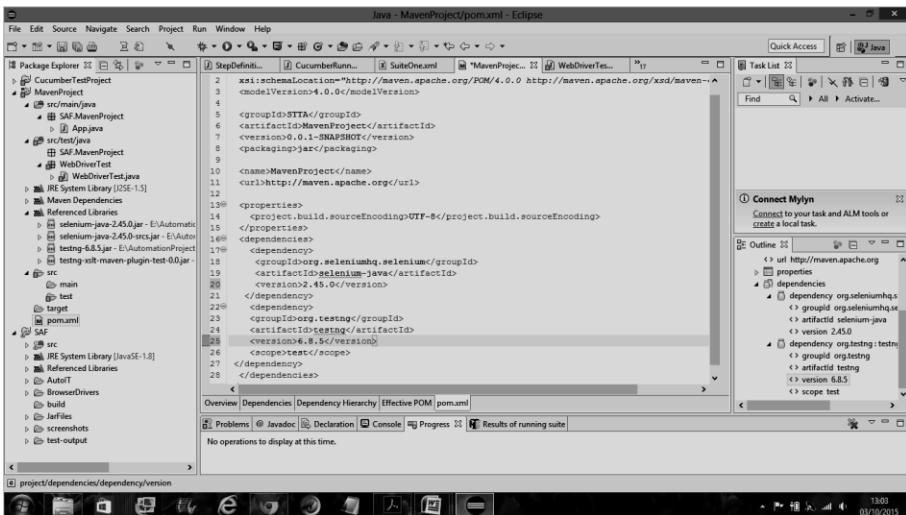
Select the path of the JarFiles (relevant to Selenium and TestNG) folder and then select all the JAR files in the folder and then click on the Open button:



## Software Automation Testing Secrets Revealed

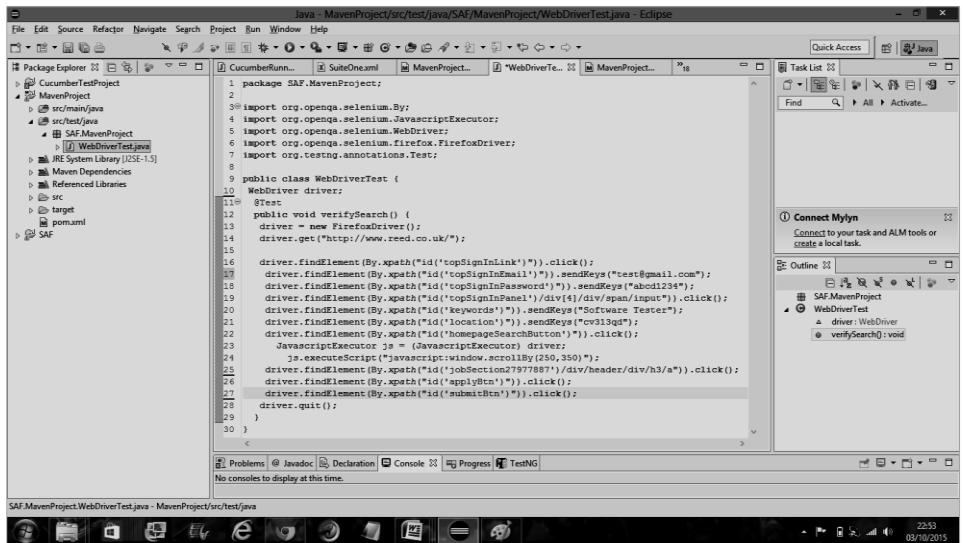


Make sure that the added external jar's version and the pom.xml's version of Selenium and TestNG resemble the same as per the script shown below:



Delete the existing app.java project within src/test/java

Add the script as shown below as a WebDriverTest.java:



## Script for Maven-based Selenium Webdriver project

/\*This is a sample project which works on a job portal and is meant for the job applicants; to login and to apply for a job, it is required that the applicant registers for a valid account and then update the email id and password as shown in the script below: \*/

**package** SAF.MavenProject;

```

import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.annotations.Test;
    
```

**public class** WebDriverTest {

    WebDriver driver;

    @Test

**public void** verifySearch() {

        driver = **new** FirefoxDriver();

        driver.get("http://www.reed.co.uk/");

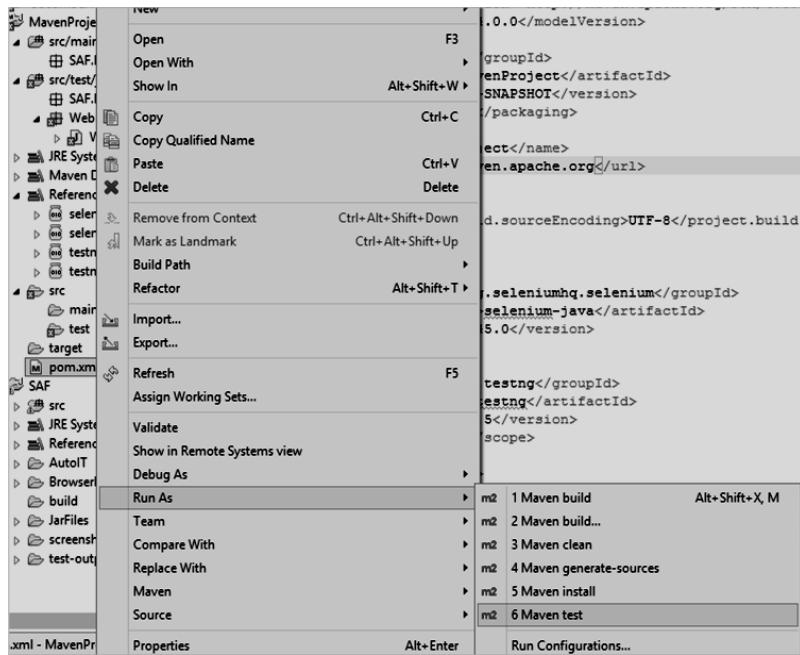
        driver.findElement(By.xpath("//a[@id='topSignInLink']")).click();

```
driver.findElement(By.xpath("id('topSignInEmail')")).sendKeys("test@gmail.com");
driver.findElement(By.xpath("id('topSignInPassword')")).sendKeys("abcd1234");
driver.findElement(By.xpath("id('topSignInPanel')/div[4]/div/span/input")).click();
driver.findElement(By.xpath("id('keywords')")).sendKeys("Software Tester");
driver.findElement(By.xpath("id('location')")).sendKeys("cv313qd");
driver.findElement(By.xpath("id('homepageSearchButton')")).click();
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript("javascript:window.scrollBy(250,350)");

driver.findElement(By.xpath("id('jobSection27977887')/div/header/div/h3/a")).click();
driver.findElement(By.xpath("id('applyBtn')")).click();
driver.findElement(By.xpath("id('submitBtn')")).click();
driver.quit();
}
}
```

## Run Maven Test

Once the script is added and the amendments are made, Run the test by right clicking on the pom.xml and then select Maven Test:



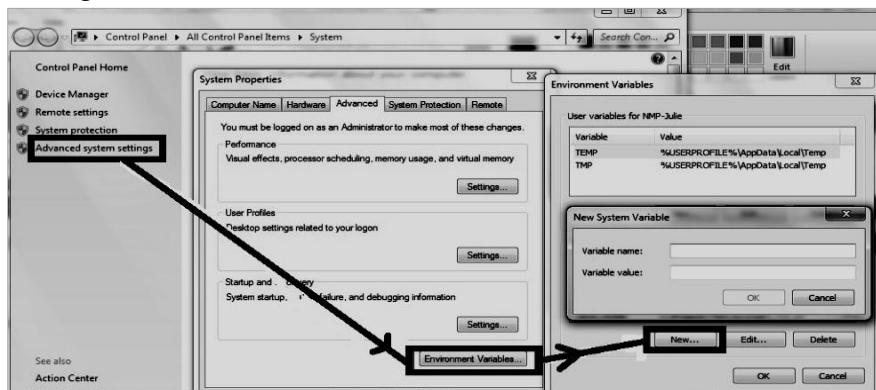
## How to run the Maven Project from the Command Prompt?

Download Maven from the link given below:

<http://maven.apache.org/download.cgi>

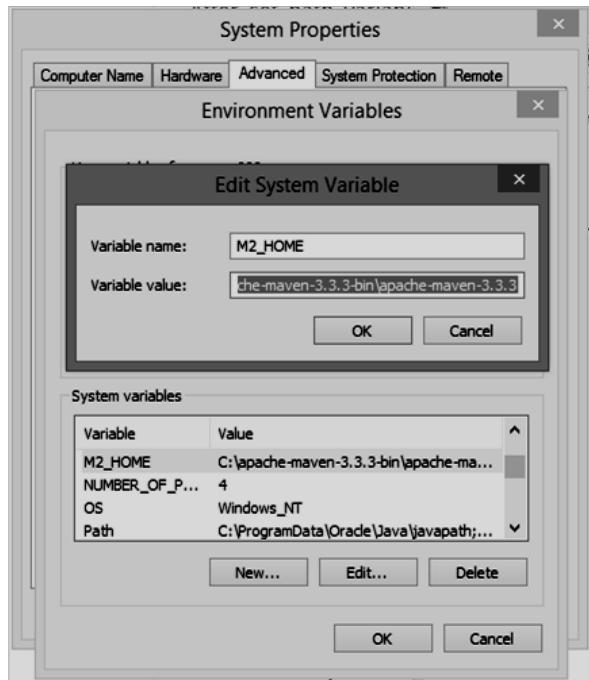
Set Environment Variables:

Navigate to Control Panel and then select Advanced System Settings. In the Advanced Tab, click on Environment Variables:

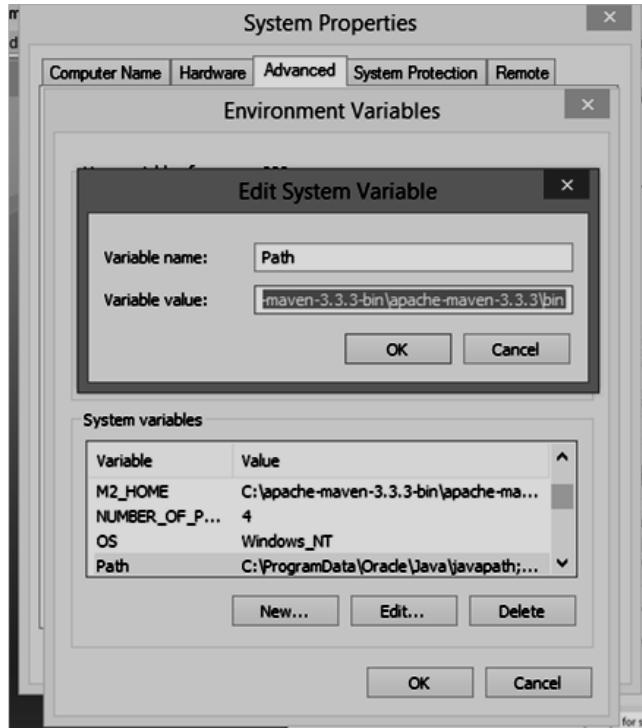


Click on New button and then enter the location of the JDK folder:

Update the M2\_HOME in the Environment Variables:



Set the path as the bin file location of the Maven folder:



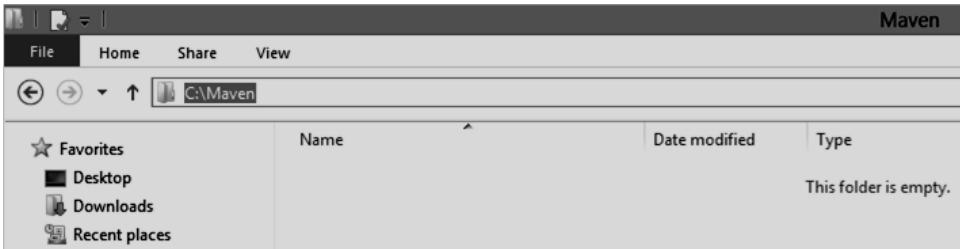
## How to check the Maven Project has been installed?

Type **mvn -version** from the command prompt as shown below:

```
C:\WINDOWS>
C:\WINDOWS>Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\naray_000>mvn -version
Apache Maven 3.3.3 (7994120775791599e205a5524ec3e0dfe41d4a06; 2015-04-22T12:57:37+01:00)
Maven home: C:\apache-maven-3.3.3-bin\apache-maven-3.3.3
Java version: 1.8.0_45, vendor: Oracle Corporation
Java home: C:\Program Files\Java\jdk1.8.0_45\jre
Default locale: en_GB, platform encoding: Cp1252
OS name: "windows 8.1", version: "6.3", arch: "amd64", family: "dos"
```

Create a folder within the C: drive or any other valid drive:



Navigate to the folder and then type mvn archetype:generate

```
C:\>cd Maven  
The system cannot find the path specified.  
C:\>cd Maven  
C:\Maven>mvn archetype:generate
```

Press Enter for all the numbers asked:

```
C:\WINDOWS\system32\cmd.exe - mvn archetype:generate

[INFO] Adding Spec2 AND Scalacheck AND JUnit.
[INFO] Adding ScalaCheck AND JUnit.
[INFO] only; adds JUnit ONLY.
[INFO] version: 2.10.x, DEFAULT: 2.11.x.
[INFO] (Note: Archetype versions are automatically
[INFO] aligned with the selected project's versions.
[INFO] Scala versions may not be compatible with selected test library versions. Adjust as necessary.)
[INFO] note > pro.savant.circumflexwebapp-archetype (-)
[INFO] note > ru.yandex.cocaine-archetype (-)
[INFO] note > ru.circumflexircumflex-archetype (-)
[INFO] note > ru.ruizv.android.archetypes:release (-)
[INFO] note > ru.stqa.seleniumwebdriver-avaarchetype (Archetype for a Maven project intended to develop tests with Selenium WebDriver and JUnit/TestNG)
[INFO] note > ru.stqa.seleniumwebdriver-junit-archetype (Archetype for a Maven project intended to develop tests with Selenium WebDriver and JUnit)
[INFO] note > ru.yandex.cocaine-client-archetype (Archetype for creating a Cocaine client)
[INFO] note > ru.yandex.cocaine-worker-archetype (Archetype for creating a basic worker for Cocaine Application Engine)
[INFO] note > se.vriongen.java.maven.archetypes:java-jaxb-archetype (-)
[INFO] note > se.walkerclough:maven-archetype (Quickstart for developers wanting to integrate the GHF Maven Plugin)
[INFO] note > tk-skunk/quickstarts-maven-archetype (A simple Maven archetype for Clojure)
[INFO] note > tr.com.luciddesk:kite-archetype (A Maven Archetype that allows users to create a Fresh Kite project)
[INFO] note > uk.ac.acrdg.resilience4j-resilience4j-webapp-archetype (-)
[INFO] note > us.fatehi:hchemarwar-archetype-maven7-archetype (A basic Java EEE7 Maven archetype)
[INFO] note > us.fatehi:hchemarwar-archetype-maven7-command (-)
[INFO] note > us.fatehi:hchemarwar-archetype-plugin-maven-command (-)
[INFO] note > us.fatehi:hchemarwar-archetype-plugin-quickstart (-)
[INFO] note > us.fatehi:hchemarwar-archetype-plugin-lint (-)
[INFO] number or apply Filter (format: [group:id]:artifactId, case sensitive contains): 63:
[INFO] org.apache.maven.archetypes:maven-archetype-quickstart:version:
[INFO] alpha-1
[INFO] alpha-2
[INFO] alpha-3
[INFO] alpha-4

[INFO] number: 6:
[INFO] value for property 'groupId': :
```

Enter com.demopack as the group id:

a number: 6:  
value for property 'groupId': : com.demopack

Artifact ID: Enter as MavenProject

```
number: 6:  
value for property 'groupId': : com.demopack  
value for property 'artifactId': : MavenProject
```

## Build Success:

```
Choose a number: 6:  
Define value for property 'groupId': : com.demopack  
Define value for property 'artifactId': : MavenProject  
Define value for property 'version': 1.0-SNAPSHOT: :  
Define value for property 'package': com.demopack: :  
Confirm properties configuration:  
groupId: com.demopack  
artifactId: MavenProject  
version: 1.0-SNAPSHOT  
package: com.demopack  
Y: :  
[INFO] -----  
[INFO] Using following parameters for creating project from Old (1.x) Archetype: m  
[INFO] -----  
[INFO] Parameter: basedir, Value: C:\Maven  
[INFO] Parameter: package, Value: com.demopack  
[INFO] Parameter: groupId, Value: com.demopack  
[INFO] Parameter: artifactId, Value: MavenProject  
[INFO] Parameter: packageName, Value: com.demopack  
[INFO] Parameter: version, Value: 1.0-SNAPSHOT  
[INFO] project created from Old (1.x) Archetype in dir: C:\Maven\MavenProject  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 06:18 min  
[INFO] Finished at: 2015-10-03T16:07:24+01:00  
[INFO] Final Memory: 13M/68M  
[INFO] -----  
C:\Maven>
```

Once the build is a success, verify that the src folder and the pom.xml gets created within the MavenProject and then go ahead and edit the script as described earlier in the Eclipse steps.

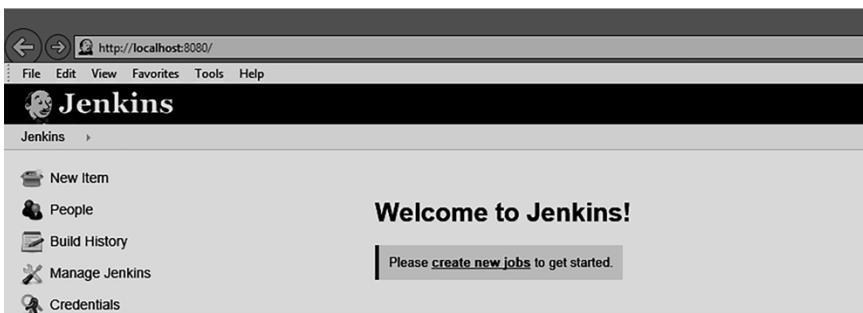
## How to add the project into the scheduled tasks of Jenkins and why?

When the automation tests are long running by themselves and providing the test results, it is always an advantage with respect to time as it is utilized in the best possible ways. Many projects target on the evening test executions as the test runs overnight and provides the test results by the morning.

## How to set up the Eclipse executable project in Jenkins?

Steps:

1. Install Jenkins and open the URL localhost:8080



2. Once Jenkins is opened, click on the Add New Item (new project)

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links: New Item, People, Build History, Manage Jenkins, and Credentials. Below the sidebar, there are two collapsed sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle). On the right, there's a configuration panel for creating a new item. It has a 'Item name' field containing 'TestProject'. Below it are four project type options: 'Freestyle project' (radio button), 'Maven project' (radio button), 'External Job' (radio button), and 'Multi-configuration project' (radio button, which is selected). A tooltip for 'Freestyle project' says: 'This is the central feature of J' and 'other than software build.' A tooltip for 'Multi-configuration project' says: 'This type of job allows you to' and 'Jenkins as a dashboard of yo'.

3. Click Configure>Build>Add a build step>Execute Windows batch command

The screenshot shows a 'Add build step' dropdown menu. The 'Execute Windows batch command' option is highlighted with a dashed border. Other options listed are 'Execute shell', 'Invoke Ant', and 'Invoke top-level Maven targets'. At the bottom of the menu are 'Save' and 'Apply' buttons.

You can run the Eclipse project from the command line as shown below:

```
C: \eclipse\clipsec -nosplash -application  
org.eclipse.jdt.core.aptBuild -data %WORKSPACE%
```

So, add the respective command line to run the Eclipse project command line instructions from Jenkin itself.

#### 4. Post Build Actions section> Add your email id

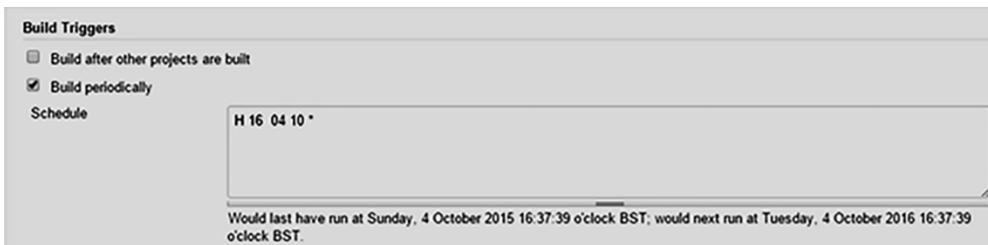
#### Schedule the Projects in Jenkins

Automation projects can be scheduled in Jenkins with specific timings! This can be done by adding the Crone Expression to the build.

#### Crone Expression:

String comprised of five fields with whitespace

For Eg: H 16 04 10 \* which means 16:4PM; 04: 04th Day;10:October



## Selenium IDE

*Programming language used in this section: Selenese*

*More code examples of this section are available from github at:*

*<https://github.com/narayananpalani/testautomation/tree/master/003SeleniumIDE-TestProject>*

Firefox extension which allows record/play testing paradigm:

Selenium IDE acts as a recording tool to capture the objects' properties of the application under test and then replay it back whenever required. The most important feature of the automation is to build a framework around the testing automation tool. This can be possible with the Selenium Webdriver as IDE works only for Firefox. So, if the requirement arises to perform test across browsers then the test analyst has to choose the right test automation tool for the purpose.

Automates commands, but asserts must be entered by hand:

Selenium IDE has three types of commands called 'Actions', 'Assessors' and 'Assertions'. But these three types can only be entered manually into IDE and the record option doesn't automatically add them into the script. So, the test analyst should be aware of the various types of commands that are used under these three types.

Creates the simplest possible Locator:

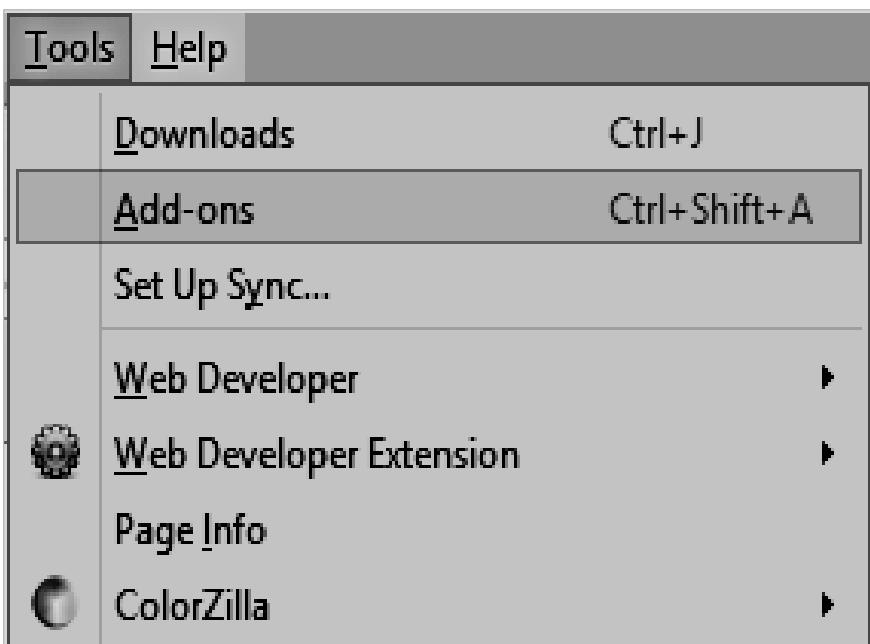
When recording the object properties, Selenium IDE picks the simplest object property as reference. This leads to failure of the script for the later cycles as the properties may change time to time. It is always advisable to change the property to any of the unique properties such as ID or xpath in such a way that the property will not change across the test cycles.

Based on Selenese:

Selenium IDE commands are written in JAVASCRIPT and HTML. So, these scripts are commonly known as Selenese.

## How to install Selenium IDE?

Install Firefox and then click on the Tools button and the Add-ons button:



In the search box, search for Selenium IDE and then download. Alternatively, navigate to <http://www.seleniumhq.org/download/> and then download it from the link given below:

**Selenium IDE**

Selenium IDE is a Firefox plugin which records and plays back user interactions to either create simple scripts or assist in exploratory testing or WebDriver scripts, though they tend to be somewhat brittle as a sort of Page Object-y structure for any kind of resiliency.

[Download latest released version 2.9.0 released on 09/Mar/2015](#)  
[install some plugins.](#)

[Download previous version 2.8.0 released on 29/Sep/2014.](#)

## Basic knowledge of Selenium IDE-Know How

Step1: Test Case Pane

Step2: Toolbar

Step3: Menu Bar

Step4: Log/Reference/UI-Element/Rollup Pane



Test Case Pane:

The Selenese script is displayed in the test case pane.

It contains two tabs:

One for displaying the command in Selenese (source) and their parameters in a readable “table” format, so it is easy to insert the assessors and the assertions as part of the script.

Toolbar: The toolbar of Selenium IDE contains buttons for controlling the test execution of your test cases, including the step feature.

Menu Bar:

File Menu: The File menu allows you to create, open and save test case and test suite files.

Edit Menu: The Edit menu allows you to copy, paste, delete, undo and select all operations for editing the commands in your test case.

Options Menu: The Options menu enables the changing of the settings. You can set the timeout value for certain commands, add user-defined user extensions to the base set of Selenium commands

and specify the format (language) used when saving your test cases.

#### Help Menu:

#### Recording and Run settings

When Selenium-IDE is first opened, the record button is ON by default.

During recording, Selenium-IDE will automatically insert commands into your test case based on your actions.

#### Remember Base URL MODE – Using Base URL to Run Test Cases in Different Domains

Record Absolute recording mode – Run Test Cases in Particular Domain

#### Running Test Cases

Run a Test Case – Click the Run button to run the currently displayed test case.

Run a Test Suite – Click the Run All button to run all the test cases in the currently loaded test suite.

Stop and Start – The Pause button can be used to stop the test case while it is in the running mode. The icon of this button then changes to indicate the Resume button. To continue, click Resume.

Stop in the Middle – You can set a breakpoint in the test case to make it to stop on a particular command. This is useful for debugging your test case. To set a breakpoint, select a command, right-click and from the context menu select Toggle Breakpoint.

Start from the Middle – You can tell the IDE to begin running from a specific command in the middle of the test case. This also is used for debugging. To set a startpoint, select a command, right-click and then from the context menu select Set/Clear Start Point.

Run Any Single Command – Double-click any single command to run it by itself. This is useful when writing a single command. It lets you immediately test a command you are constructing, when you are not sure if it is correct. You can double-click it to see if it runs correctly. This is also available from the context menu.

Sample script to use Wait, Assert and Assertion commands:

TC#1		
Open	http://www.abctest.com	
Click	id=abc-search	
Type	id=abc-search	Tool efficient
Click	//form[@id='abc-search-form']/div/span/i	
pause	3000	
assertTitle	abctest.com   imagination at work	
assertTextPresent	ABCTEST is translating the Industrial element	
verifyText	css=h2	Learn about ABCTEST's industry solutions that are powering the Industry

Sample script to use screenshot command:

Screenshottestcase		
open	/	
click	id=search	
type	id=search	SAP
clickAndWait	css=button.btn.btn-search	
captureEntirePageScreenshot	E:\Selenium IDE\googlesearch.jpeg	

## Sample Selenium IDE Automation Scripts on Selenium Web Page

*Download the projects at*

*<https://github.com/narayananpalani/testautomation> and refer the samples here*

Test Case using verifyTextPresent, assertTextNotPresent:

01_Selenium_tabs		
Open	/projects/	
clickAndWait	link=Selenium Projects	
verifyText	link=Selenium WebDriver	Selenium WebDriver
assertText	link=Selenium IDE	Selenium IDE
clickAndWait	link=Download	
verifyTextPresent		Selenium Standalone Server
assertTitle	Downloads	
clickAndWait	link=Documentation	
verifyText	css=div.ads > h3	Donate to Selenium
assertTextNotPresent	mnnfbmddfmgmjfg	hndshsdfhdsfmjds
clickAndWait	link=Support	
verifyTitle	Getting Help	Getting Help
assertText	css=h2	User Group
clickAndWait	link=Support	
clickAndWait	link=Support	
clickAndWait	link=About	
verifyTitle	About Selenium	About Selenium
assertTitle	About Selenium	About Selenium

Test Case using assertText:

selenium_projects1		
Open	/projects/	
clickAndWait	link=Selenium Projects	
verifyTextPresent		Selenium Projects
assertText	css=strong	Selenium is a suite

		of tools
clickAndWait	link=Selenium WebDriver	
verifyTitle	Selenium WebDriver	Selenium WebDriver
assertTitle	Selenium WebDriver	Selenium WebDriver

Test Case assertAlertNotPresent, verifyTextNotPresent, assertTitle

selenium_projects2		
Open	/projects/webdriver/	
verifyTextPresent		Selenium is a suite of tools to automate web browsers across many platforms.
assertTextPresent		Donate to Selenium
clickAndWait	link=Selenium Projects	
verifyTextNotPresent	dmbmjbmSn	Djfgjhfkhnks
assertAlertNotPresent	kdfkjdsb	Jhdfjhdb
clickAndWait	link=Selenium Grid	
verifyText	css=#mainContent > p	With the release of Selenium 2.0, the Selenium Server now has built-in Grid functionality. To see the how to configure this, please see the wiki.
assertTitle	Selenium Grid	Selenium Grid

## Test Case using verifyTable, assertTable

selenium_projects3		
Open	/projects/grid/	
clickAndWait	link=Selenium Projects	
clickAndWait	link=Selenium IDE	
verifyTable	css=table.0.0	Selenium IDE is an integrated development environment for Selenium scripts. It is implemented as a Firefox extension, and allows you to record, edit and debug tests.
assertTable	css=table.0.0	Selenium IDE is an integrated development environment for Selenium scripts.

## Test Case using verifyElementPresent

selenium_projects4		
Open	/projects/	
clickAndWait	link=Selenium Projects	
clickAndWait	link=Selenium Remote Control	
verifyElementPresent	//img[@alt='rc arch diagram']	Internet Explorer
verifyElementPresent	//img[@alt='rc arch diagram']	Firefox
verifyElementPresent	//img[@alt='rc arch diagram']	safari

Test Case using verifyElementPresent, verifyText, assertElementPresent, assertText, verifyElementPresent, verifyTable:

selenium_projects5		
open	/projects/	
clickAndWait	link=Selenium Projects	
clickAndWait	link=Download	
verifyElement Present	//div[@id='mainContent']/p	
verifyText	link=Latest Releases	Latest Releases
assertElement Present	//div[@id='mainContent']/div[14]/div[2]/table/tbody/tr/td/a[3]	
assertText	//div[@id='mainContent']/table/tbody/tr[4]/td[4]/a	Download
verifyElement Present	//div[@id='side']/div/div/a/img	
verifyTable	//div[@id='mainContent']/table[2].4.1	

Test Case using assertText, verifyText, assertValue, verifyValue:

selenium_projects6		
open	/projects/	
clickAndWait	link=Selenium Projects	
clickAndWait	link=Download	
clickAndWait	link=Documentation	
verifyText	css=a.reference.internal	Note to the Reader - Docs Being Revised for Selenium 2.0!
verifyElement Present	link=Introduction	
verifyText	link=Test Automation for Web Applications	Test Automation for Web Applications
verifyText	link=exact:To Automate or Not to Automate?	exact:To Automate or Not

		to Automate?
assertText	link=Introducing Selenium	Introducing Selenium
verifyElement Present	link=Brief History of The Selenium Project	
assertElement Present	link=Selenium's Tool Suite	
verifyText	link=Choosing Your Selenium Tool	Choosing Your Selenium Tool
assertText	link=Supported Browsers and Platforms	Supported Browsers and Platforms
verifyText	link=Flexibility and Extensibility	Flexibility and Extensibility
assertElement Present	link=exact:What's in this Book?	
verifyText	link=The Documentation Team–Authors Past and Present	The Documentation Team–Authors Past and Present
assertElement Present	link=Selenium-IDE	
verifyText	xpath=(//a[contains(text(),'Introduction')])[2]	Introduction
verifyTitle	Selenium Documentation – Selenium Documentation	
assertElement Present	link=Installing the IDE	
verifyElement Present	link=Installing the IDE	
assertElement Present	link=Opening the IDE	
verifyText	link=IDE Features	IDE Features
verifyElement Present	link=Building Test Cases	
assertText	link=Using Base URL to Run Test Cases in Different Domains	Using Base URL to Run Test Cases in Different

		Domains
verifyText	link=Using Base URL to Run Test Cases in Different Domains	Using Base URL to Run Test Cases in Different Domains
assertElement Present	link=Selenium Commands – “Selenese”	
assertElement Present	link=Script Syntax	
verifyText	link=Test Suites	Test Suites
assertText	link=Commonly Used Selenium Commands	Commonly Used Selenium Commands
verifyText	link=Verifying Page Elements	Verifying Page Elements
assertText	link=exact: Assertion or Verification?	exact: Assertion or Verification?
verifyElement Present	link=Locating Elements	
verifyText	link=Matching Text Patterns	Matching Text Patterns
assertElement Present	link=The “AndWait” Commands	
verifyElement Present	link=The waitFor Commands in AJAX applications	
verifyText	link=Sequence of Evaluation and Flow Control	Sequence of Evaluation and Flow Control
assertElement Present	link=Store Commands and Selenium Variables	
verifyText	link=Store Commands and Selenium Variables	Store Commands and Selenium Variables
assertText	link=JavaScript and Selenese Parameters	JavaScript and Selenese Parameters
verifyText	link=echo - The Selenese Print Command	echo - The Selenese Print

		Command
assertText	link=Alerts, Popups and Multiple Windows	Alerts, Popups, and Multiple Windows
verifyText	link=Debugging	Debugging
assertText	link=Writing a Test Suite	Writing a Test Suite
verifyText	link=User Extensions	User Extensions
assertText	link=Format	Format
verifyText	link=Executing Selenium-IDE Tests on Different Browsers	Executing Selenium-IDE Tests on Different Browsers
assertText	link=Troubleshooting	Troubleshooting
verifyTitle	Selenium Documentation – Selenium Documentation	
verifyElement Present	link=Introducing WebDriver	
assertText	link=exact:How Does WebDriver ‘Drive’ the Browser Compared to Selenium-RC?	exact:How Does WebDriver ‘Drive’ the Browser Compared to Selenium-RC?
verifyText	link=WebDriver and the Selenium-Server	WebDriver and the Selenium-Server
assertElement Present	link=Setting Up a Selenium-WebDriver Project	
verifyText	link=Migrating from Selenium 1.0	Migrating from Selenium 1.0
assertElement Present	link=Introducing the Selenium-WebDriver API by Example	
verifyElement Present	link=Selenium-WebDriver API Commands and Operations	
assertElement Present	link=Driver Specifics and Trade-offs	

verifyText	link=Selenium-WebDriver's Drivers	Selenium-WebDriver's Drivers
assertText	link=Alternative Back-Ends: Mixing WebDriver and RC Technologies	Alternative Back-Ends: Mixing WebDriver and RC Technologies
verifyText	link=Running Standalone Selenium Server for use with RemoteDrivers	Running Standalone Selenium Server for use with RemoteDrivers
assertText	link=Additional Resources	Additional Resources
verifyText	link=Next Steps	Next Steps
verifyTitle	Selenium Documentation – Selenium Documentation	
verifyText	link=Explicit and Implicit Waits	Explicit and Implicit Waits
assertElement Present	link=RemoteWebDriver	
verifyText	link=AdvancedUserInteraction s	AdvancedUserInt eractions
assertText	link=Browser Startup Manipulation	Browser Startup Manipulation
verifyText	link=HTML5	HTML5
assertText	link=Parallelizing Your Test Runs	Parallelizing Your Test Runs
verifyTitle	Selenium Documentation – Selenium Documentation	
verifyElement Present	xpath=(//a[contains(text(),'Introduction')])[3]	
assertText	link=How Selenium RC Works	How Selenium RC Works
verifyText	link=Installation	Installation
assertText	link=From Selenese to a Program	From Selenese to a Program
verifyText	link=Programming Your Test	Programming

		Your Test
assertText	link=Learning the API	Learning the API
verifyText	link=Reporting Results	Reporting Results
assertText	link=Adding Some Spice to Your Tests	Adding Some Spice to Your Tests
assertText	link=Server Options	Server Options
verifyText	link=Specifying the Path to a Specific Browser	Specifying the Path to a Specific Browser
assertText	link=Selenium RC Architecture	Selenium RC Architecture
verifyText	link=Handling HTTPS and Security Popups	Handling HTTPS and Security Popups
assertText	link=Supporting Additional Browsers and Browser Configurations	Supporting Additional Browsers and Browser Configurations
assertText	link=Troubleshooting Common Problems	Troubleshooting Common Problems
assertText	link=Test Design Considerations	Test Design Considerations
verifyText	link=Introducing Test Design	Introducing Test Design
assertText	link=Types of Tests	Types of Tests
assertText	link=Validating Results	Validating Results
verifyText	link=Location Strategies	Location Strategies
assertText	link=Wrapping Selenium Calls	Wrapping Selenium Calls
assertText	link=UI Mapping	UI Mapping
verifyText	link=Page Object Design Pattern	Page Object Design Pattern
verifyText	link>Data Driven Testing	Data Driven Testing

assertElement Present	link=Database Validation	
verifyTitle	Selenium Documentation – Selenium Documentation	
verifyText	link=Quick Start	Quick Start
assertText	link=exact:What is Selenium-Grid?	exact:What is Selenium-Grid?
assertElement Present	link=When to Use It	
verifyText	link=Selenium-Grid 2.0	Selenium-Grid 2.0
assertText	link=Selenium-Grid 1.0	Selenium-Grid 1.0
verifyText	link=How Selenium-Grid Works–With a Hub and Nodes	How Selenium-Grid Works–With a Hub and Nodes
verifyTitle	Selenium Documentation – Selenium Documentation	
verifyText	link=Configuring Selenium-Grid	Configuring Selenium-Grid
assertText	link=Hub Configuration	Hub Configuration
verifyText	link=Node Configuration	Node Configuration
verifyText	link=Timing Parameters	Timing Parameters
verifyText	link=Customizing the Grid	Customizing the Grid
assertText	link=Getting Command-Line Help	Getting Command-Line Help
assertText	link=Common Errors	Common Errors
verifyTitle	Selenium Documentation – Selenium Documentation	
verifyText	link=Actions	Actions
verifyText	link=Accessors/Assertions	Accessors/Assertions
assertText	link=Locator Strategies	Locator Strategies
verifyElement	link=Using User-Extensions	

Present	With Selenium-IDE	
assertElement Present	link=Using User-Extensions With Selenium RC	
verifyElement Present	link=.NET client driver configuration	
assertText	link=Importing Sel2.0 Project into Eclipse using Maven	Importing Sel2.0 Project into Eclipse using Maven
verifyText	link=Importing Sel2.0 Project into IntelliJ Using Maven	Importing Sel2.0 Project into IntelliJ Using Maven
assertText	link=Migrating From Selenium RC to Selenium WebDriver	Migrating From Selenium RC to Selenium WebDriver
verifyText	link=How to Migrate to Selenium WebDriver	How to Migrate to Selenium WebDriver
assertText	link=Why Migrate to WebDriver	Why Migrate to WebDriver
verifyText	link=Before Starting	Before Starting
assertText	link=Getting Started	Getting Started
verifyElement Present	xpath=(//a[contains(text(),'Nex t Steps')])[2]	
assertText	link=Common Problems	Common Problems
verifyText	css=#mainContent > div.related > ul > li > a	Selenium Documentation
assertText	css=#mainContent > div.related > ul > li.right > a[title="Note to the Reader - Docs Being Revised for Selenium 2.0!"]	Next
verifyElement Present	id=footerLogo	
selectWindow	Null	

verifyElement Present	link=Selenium Documentation	
assertText	link=next	Next
verifyText	link=Note to the Reader - Docs Being Revised for Selenium 2.0!	Note to the Reader - Docs Being Revised for Selenium 2.0!
assertText	css=em	Docs Being Revised for Selenium 2.0
verifyElement Present	name=submit	
verifyText	name=submit	
assertElement Present	css=a > img	
assertValue	css=li > input[type="image"]	Java
verifyValue	//input[@value='csharp']	Csharp
assertText	//input[@value='python']	
verifyText	//input[@value='ruby']	
assertValue	//input[@value='php']	Php
verifyValue	//input[@value='perl']	Perl
assertValue	//input[@value='javascript']	Javascript

**Note:**

Selenium IDE is not working on Firefox version 55 onwards (as of 9th Aug'2017)

**Reference:**

<https://seleniumhq.wordpress.com/2017/08/09/firefox-55-and-selenium-ide/>

## Selenium Grid for Compatibility Tests

*Programming language used in this section: Java Programming  
Version: Selenium Grid version 2.0*

Concurrent tests are possible in Selenium Grid using existing Selenium Webdriver scripts as a concurrent test execution tool! It acts as a hub with multiple nodes to run tests in the same or in a different test environment in multiple browsers across different operating systems!

Project on GitHub:

<https://github.com/SeleniumHQ/selenium/wiki/Grid2>

Why such tools are required for testing?

When a retail company develops the shopping website to attract online buyers, it has to get tested in the target systems which imitate the customers using their systems; if customers are using Internet Explorer, Chrome, Safari, Firefox to access the website in operating systems such as Windows 7, Windows 2008 and Macbook. This cannot be done in the Selenium Webdriver's single test! It has to be reconfigured to run every time using a particular browser and a particular operating system whereas all these tests can run in separate nodes of Selenium Grid in a single go! That is the primary advantage of using Selenium Grid!

## Benefits of Selenium Grid

Fast Test Execution Time: Concurrent tests run in the given configuration parameters

Easy to perform Compatibility Testing: Cross Browser, Cross Operating System

Download the Selenium Server from SeleniumHQ download page:  
<http://www.seleniumhq.org/download/>

**Selenium Standalone Server**

The Selenium Server is needed in order to run Selenium 3.X tests. Selenium 3.X is no longer capable of running it through emulation and the WebDriver API.

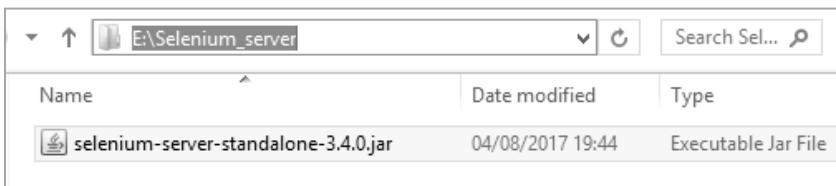
[Download version 3.4.0](#)

To run Selenium tests exported from IDEs

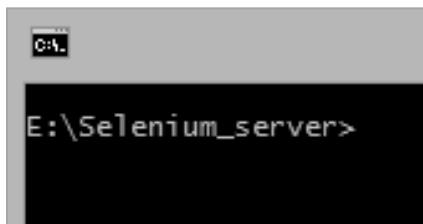
To use the Selenium Server in a Grid configuration

**The Internet Explorer Driver Server**

Once downloaded, store it in the local machine:



Navigate to the folder where it has been downloaded in the command prompt:



Once navigated to the folder level in the command prompt, start the server:

```
java -jar selenium-server-standalone-3.4.0.jar -port 4444 -role hub
```

**Note:** The version of Selenium server changes from time to time and when downloaded, note the version that has been downloaded and use the right file name in the command given above.

### Command Explanation:

```
java -jar:Launching a jar file (which is the compressed java package
selenium-server-standalone-3.4.0.jar :Name of the selenium server downloaded
-port 4444 :Port where Selenium listening to the nodes
-role hub:Act as a hub to execute the tests concurrent
```

If the following message is displayed without any error, it can be understood that the Selenium Grid Hub is up and is running fine:



```
C:\WINDOWS\system32\cmd.exe - java -jar selenium-server-standalone-3.4.0.jar -port 4444 -role hub
E:\Selenium_server>java -jar selenium-server-standalone-3.4.0.jar -port 4444 -role hub
19:55:49.213 INFO - Selenium build info: version: '3.4.0', revision:
19:55:49.225 INFO - Launching Selenium Grid hub
2017-08-04 19:55:59.431:INFO::main: Logging initialized @13242ms to o
19:55:59.483 INFO - Will listen on 4444
2017-08-04 19:55:59.615:INFO:osjs.Server:main: jetty-9.4.3.v20170317
2017-08-04 19:55:59.833:INFO:osjs.session:main: DefaultSessionIdManager
2017-08-04 19:55:59.834:INFO:osjs.session:main: No SessionScavenger s
2017-08-04 19:55:59.841:INFO:osjs.session:main: Scavenging every 600m
2017-08-04 19:55:59.885:INFO:osjs.ContextHandler:main: Started 'j
2017-08-04 19:56:00.017:INFO:osjs.AbstractConnector:main: Started 'S
2017-08-04 19:56:00.018:INFO:osjs.Server:main: Started @1389ms
19:56:00.020 INFO - Nodes should register to http://192.168.5.4444/
19:56:00.020 INFO - Selenium Grid hub is up and running
```

Press Ctrl+C to end the session of the server and ensure that the server is closing down.

Launch the website given below to see the status of the Selenium Grid:

<http://localhost:4444/grid/console>



Configuration for the Selenium Grid using multiple browsers:

Download chrome driver from the following website:

<https://sites.google.com/a/chromium.org/chromedriver/downloads>

Download IE Driver from the following website:

<http://docs.seleniumhq.org/download/>

A screenshot of a web browser window showing the "The Internet Explorer Driver Server" page from the Selenium documentation. The page contains instructions for downloading the driver and links to the "CHANGELOG" and "IE" sections.

Once downloaded, place both the IE and the Chrome Driver in a separate folder as shown below:

Name	Date modified	Type	Size
chromedriver.exe	23/01/2016 14:53	Application	4,895 KB
IEDriverServer.exe	23/01/2016 15:51	Application	3,476 KB

Use the command given below to launch the Selenium grid (if the Selenium grid is already running as hub with the previous command please keep it running and do not close. Use a separate command prompt opened to type the command given below in the same folder hierarchy of the Selenium server):

```
java -jar selenium-server-standalone-3.4.0.jar -role node -Dwebdriver.ie.driver="E:/BrowserDrivers/IEDriverServer.exe" -Dwebdriver.chrome.driver="E:/BrowserDrivers/chromedriver.exe" -hub http://localhost:4444/grid/register -port 5566
```

## Error Handling:

If you receive a message such as this error, then try an alternative command format:

```
E:\Selenium_server>java -jar selenium-server-standalone-3.4.0.jar -role node -Dwebdrivers/chromedriver.exe -hub http://localhost:4444/grid/register -port 5566
Exception in thread "main" com.beust.jcommander.ParameterException: Unknown option
    at com.beust.jcommander.JCommander.parseValues(JCommander.java:742)
    at com.beust.jcommander.JCommander.parse(JCommander.java:282)
    at com.beust.jcommander.JCommander.parse(JCommander.java:265)
    at com.beust.jcommander.JCommander.<init>(JCommander.java:210)
    at org.openqa.grid.selenium.GridLauncherV3$3.setConfiguration(GridLauncherV3$3.java:75)
    at org.openqa.grid.selenium.GridLauncherV3.buildLauncher(GridLauncherV3.java:75)
    at org.openqa.grid.selenium.GridLauncherV3.main(GridLauncherV3.java:75)
```

Command:

```
java -
Dwebdriver.ie.driver="E:/BrowserDrivers/IEDriverServer.exe" -
Dwebdriver.chrome.driver="E:/BrowserDrivers/chromedriver.exe" -
-jar selenium-server-standalone-3.0.4.jar -role node -hub
http://localhost:4444/grid/register -port 5566
```

If this code is also not working, then there are issues related to Win7 of X64 in the Selenium Grid, hence remove the configuration items of the browser and then run as shown below:

```
java -jar selenium-server-standalone-3.0.4.jar -role node -hub
http://localhost:4444/grid/register -port 5566
```

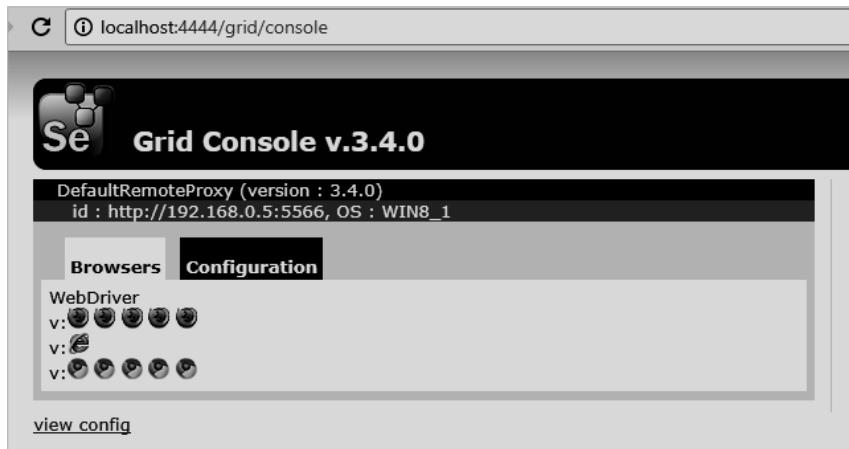
Further reference to these type of issues tracked under:

```
https://github.com/SeleniumHQ/selenium/issues/3170
```

When the command gets accepted, that shows a node confirmation of the registration as shown below:

```
Selenium build info: version: '3.4.0', revision: 'unknown'
Launching a Selenium Grid node
.012:INFO:main: Logging initialized @2370ms to org.seleniumhq.jetty9.util.log.StderrLog
Driver class not found: com.opera.core.systems.OperaDriver
Driver provider com.opera.core.systems.OperaDriver registration is skipped:
w instances on this machine.
Driver class not found: com.opera.core.systems.OperaDriver
Driver provider com.opera.core.systems.OperaDriver is not registered
Driver provider org.openqa.selenium.safari.SafariDriver registration is skipped:
ilities Capabilities {[{browserName=safari, version=, platform=MAC}]} does not match the current platform WIN8_1
.194:INFO:osjs.Server:main: jetty-9.4.3.v20170317
.247:INFO:osjs.ContextHandler:main: Started o.s.j.s.ServletContextHandler@1f0ff111{/null,AVAILABLE}
.297:INFO:osjs.AbstractConnector:main: Started ServerConnector@5223eSee{HTTP/1.1,[http/1.1]}{0.0.0.0:5566}
.299:INFO:osjs.Server:main: Started @2657ms
Selenium Grid node is up and ready to register to the hub
Starting auto registration thread. Will try to register every 5000 ms.
Registering the node to the hub: http://localhost:4444/grid/register
The node is registered to the hub and ready to use
SessionCleaner initialized with insideBrowserTimeout 0 and clientGoneTimeout 1800000 polling every 180000
```

Refresh the web page to see the browsers in the Selenium Grid:



How to write the first code in the Selenium Webdriver and run in the Selenium Grid?

Prerequisites:

Refer Selenium Framework (Hybrid) using Java (and TestNG) in this book to build the framework and use this code sample in the java-based Selenium test and run in the Selenium Grid (by running the test in TestNG).

Code Sample:

```
//Import section of the code written in Java programming
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeTest;
```

```
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;
import org.openqa.selenium.By;
import static org.testng.Assert.fail;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.Capabilities;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.RemoteWebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
//Class name of the program.File has to be saved in GridConcurrentTest.java to work in the framework with the help of Selenium and TestNG:
public class GridConcurrentTest {
    WebDriver driver = null;
    private StringBuffer verificationErrors = new StringBuffer();
    // Transfer the details of the platform and driver and URL to launch
    @Parameters({ "strPlatform", "driver", "url" })
    @BeforeTest(alwaysRun = true)
    public void setup(String strPlatform, RemoteWebDriver driver,
        String url) throws MalformedURLException {
        DesiredCapabilities bwsrSettings = new DesiredCapabilities();

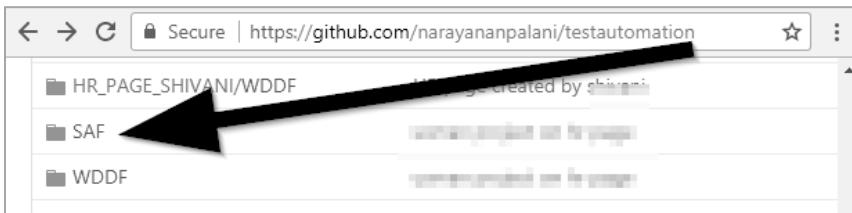
        bwsrSettings.setPlatform(org.openqa.selenium.Platform.WINDOWS);
        if (driver.equalsIgnoreCase("Internet Explorer"))
            bwsrSettings = DesiredCapabilities.internetExplorer();
        if (driver.equalsIgnoreCase("Firefox"))
            bwsrSettings = DesiredCapabilities.firefox();
        if (driver.equalsIgnoreCase("chrome"))
            bwsrSettings = DesiredCapabilities.chrome();
```

```
// This section launches the driver within the grid node by talking  
to hub of Selenium Grid.  
  
driver = new RemoteWebDriver(new  
URL("http://localhost:4444/wd/hub"),bwsrSettings);  
driver.manage().timeouts().implicitlyWait(30,  
TimeUnit.SECONDS);  
driver.manage().window().maximize();  
driver.get(url);  
}  
  
// Simple test method to execute.  
@Test(description = "Sample Grid Test")  
public void testCalc() throws InterruptedException {  
driver.navigate().to("www.webdriverinselenium.com");  
  
@AfterTest  
public void afterTest() {  
driver.quit();  
}  
}
```

**Note:**

Refer the code base at GitHub to build this same code inside the following framework:

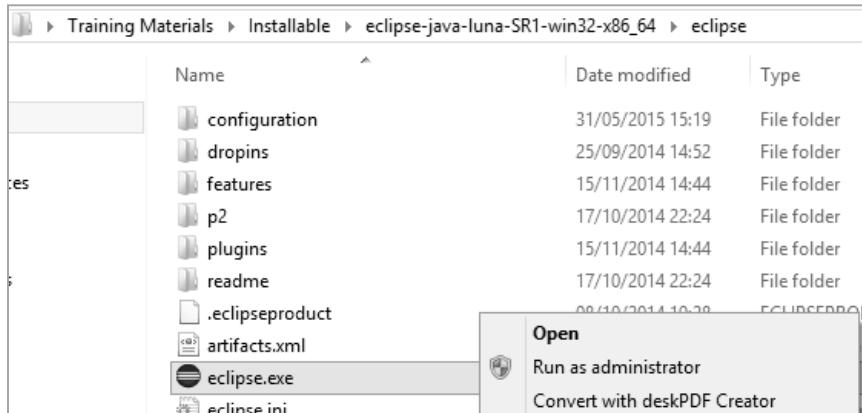
<https://github.com/narayananpalani/testautomation/tree/master/SAF>



Step by step enhancements of the Selenium Grid scripts to the hybrid Selenium Framework:

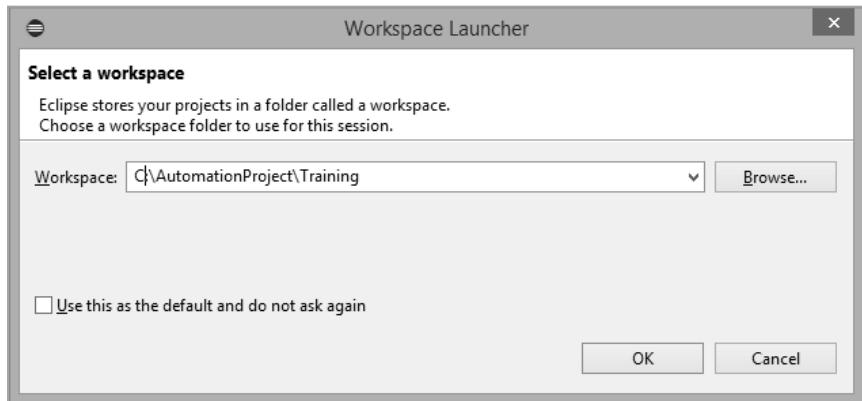
Launch Eclipse from the folder where it has been installed (Unlike other software applications, eclipse needs to be launched by exe every time to open and access whereas other software available in the shortcuts or different links to access while launching them).

Open `eclipse.exe` (preferable Eclipse Luna):



### Step3:

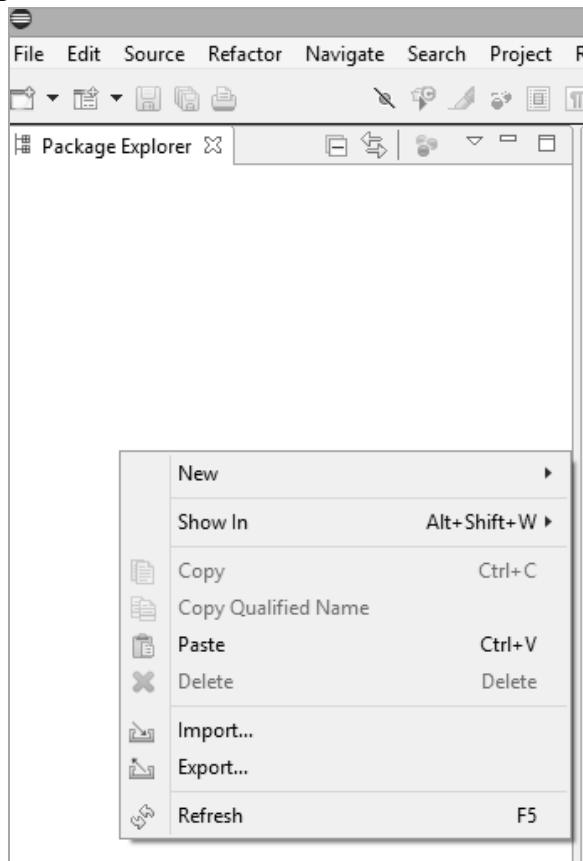
Choose the folder of the automation framework folder and then click on the OK button:



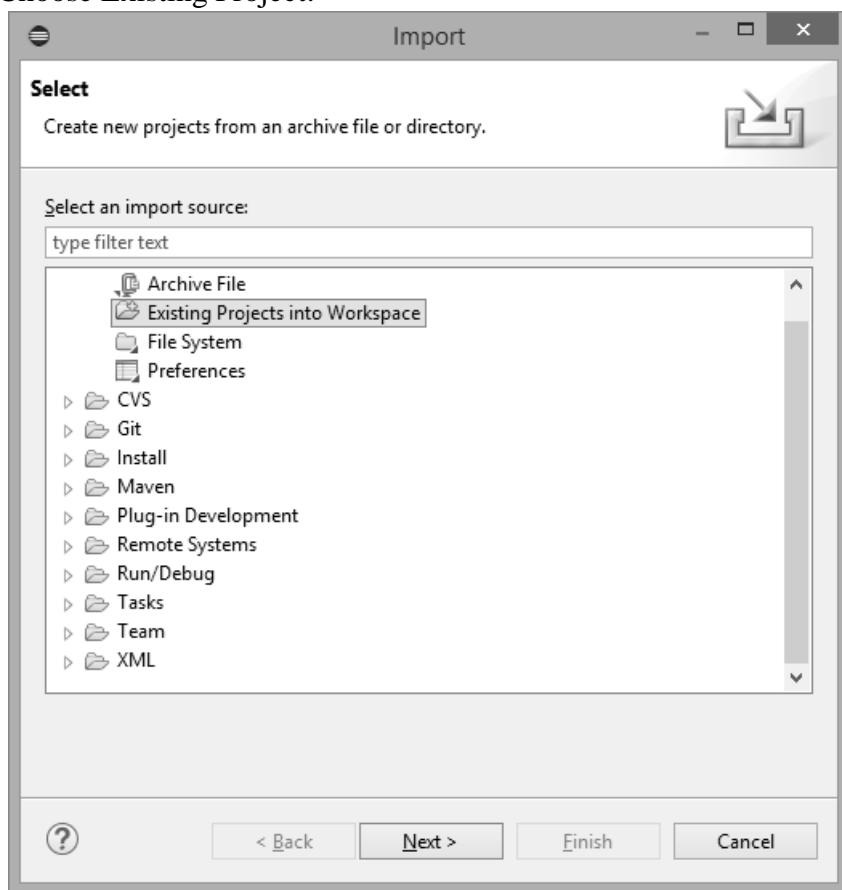
Eclipse Luna launches on the screen:



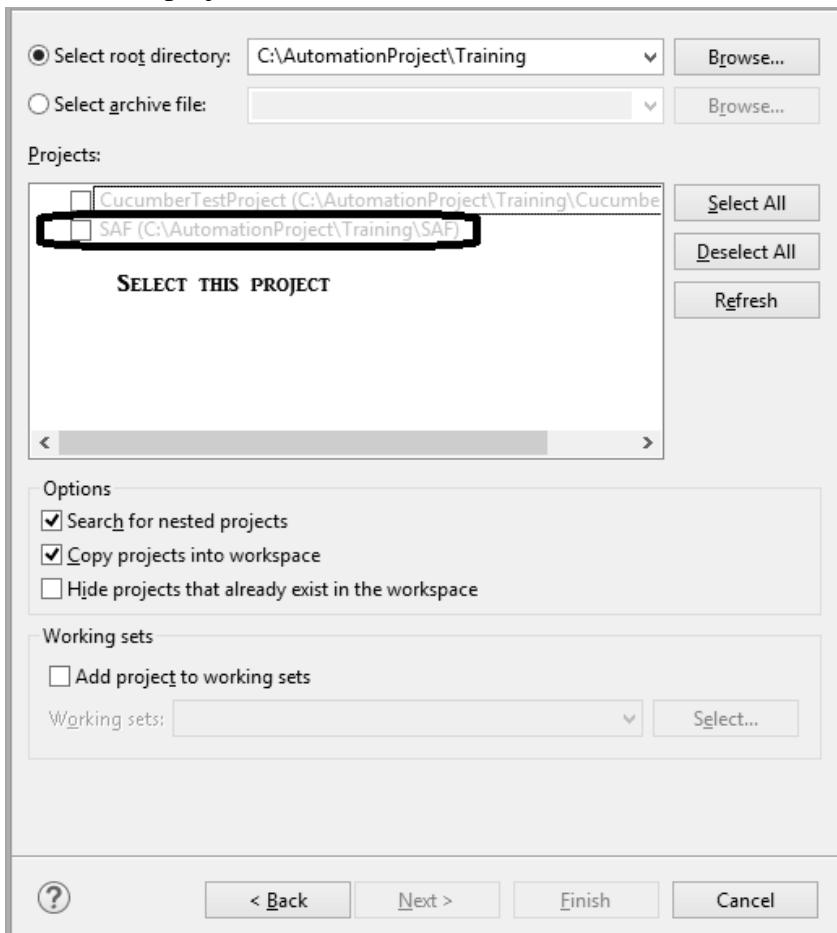
Once opened, click on the Workbench button and then navigate to the workbench. Right click on the package explorer and then choose Import:



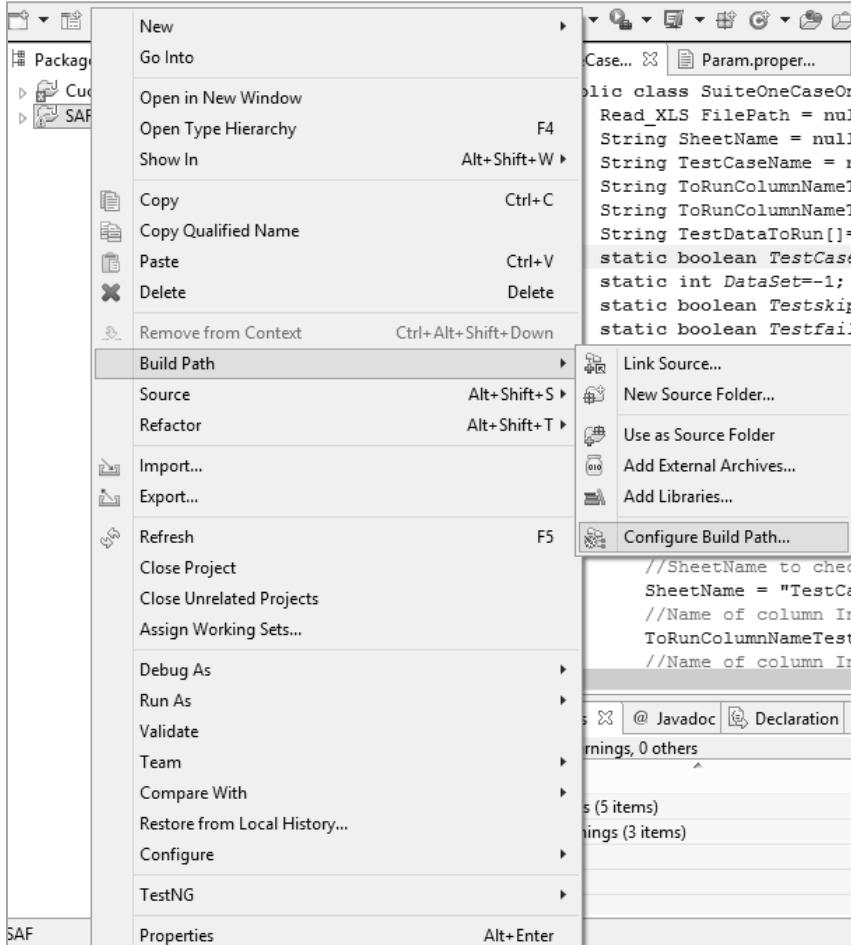
Choose Existing Project:



Click on the Browse button to choose the project and then select both the checkboxes to search and copy the existing projects. Choose SAF project in the list and then click on the OK Button:



Once the SAF project is visible in Package Explorer, navigate to Configure Build Path as shown below:

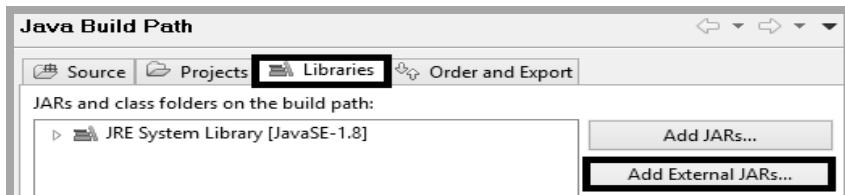


Right click on the folder SAF, click on Build Path and then click on Configure Build Path as follows:



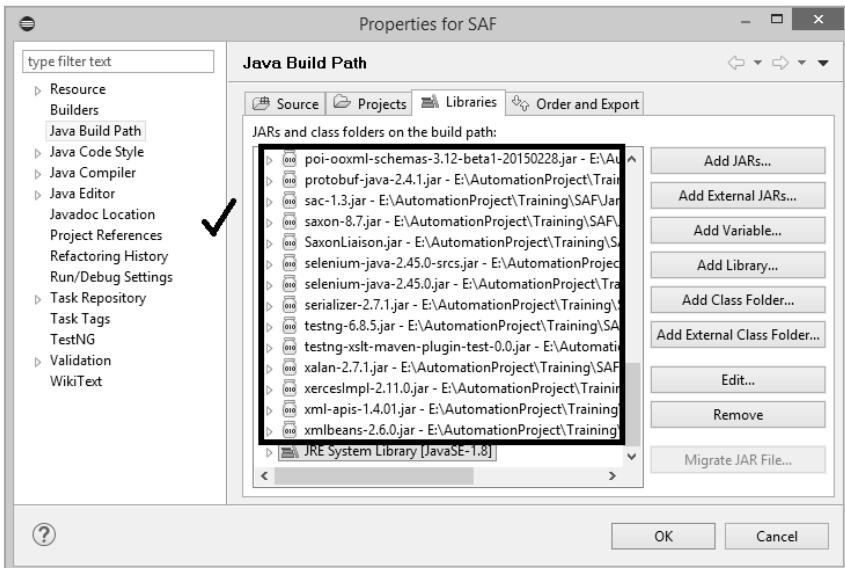
In the Properties of SAF, click on the Libraries tab, click on Add External JARs button as shown:

Note: Make sure that the Libraries tab is clicked and then the Add External JARs button is clicked from there.

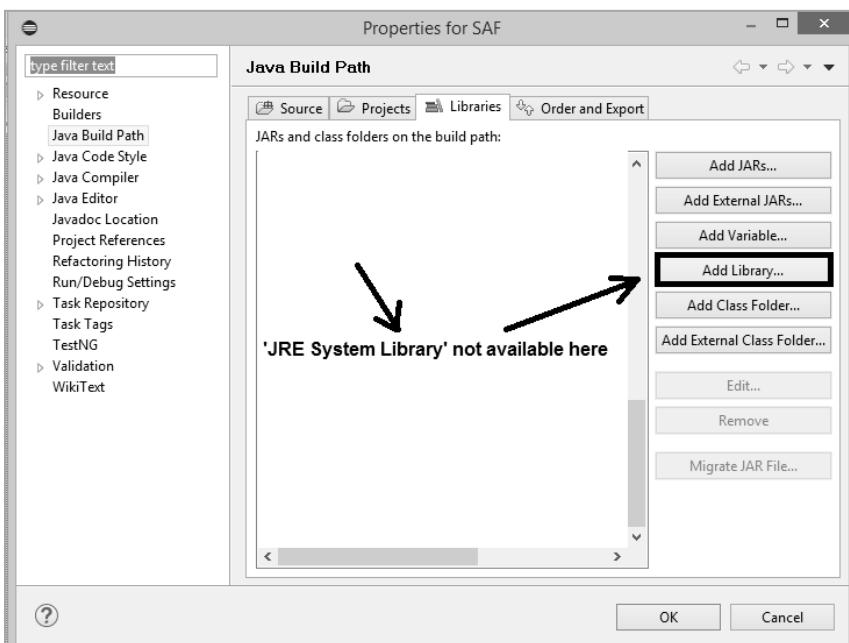


(Follow the same steps which are mentioned in this section to add all the external JARs)

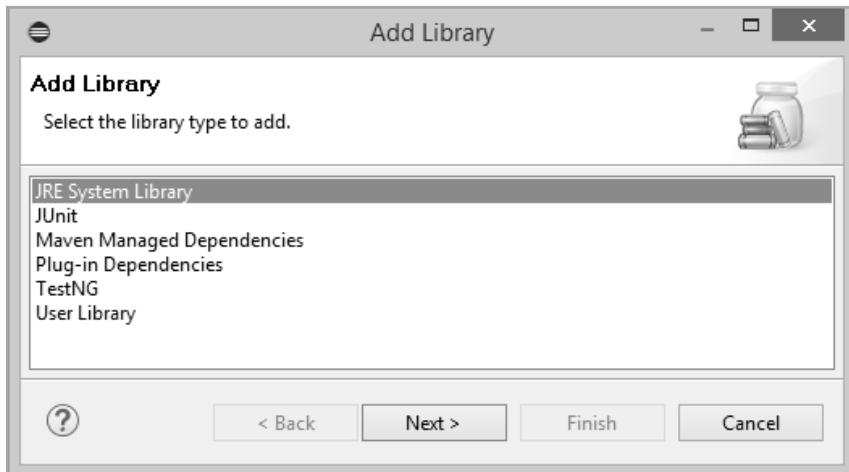
Please select the JAR files from the JarFiles folder of the project and then click Open to list the jar files within 'Libraries'. If there are existing JAR files with incorrect path or highlighted with errors, remove them and add fresh files from JarFiles folder.



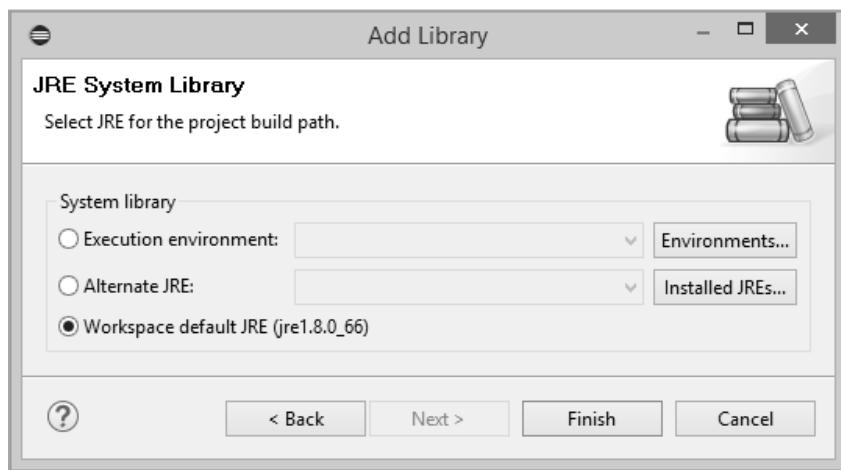
Similarly, if JRE System Library exists, that is good to run the test in java-based framework; if JRE System Library doesn't appear in the Libraries above, click on the 'Add Libraries' in the Libraries section, and choose the default JRE (respective version display as third check box) and click Next and Finish buttons to display JRE System Library.



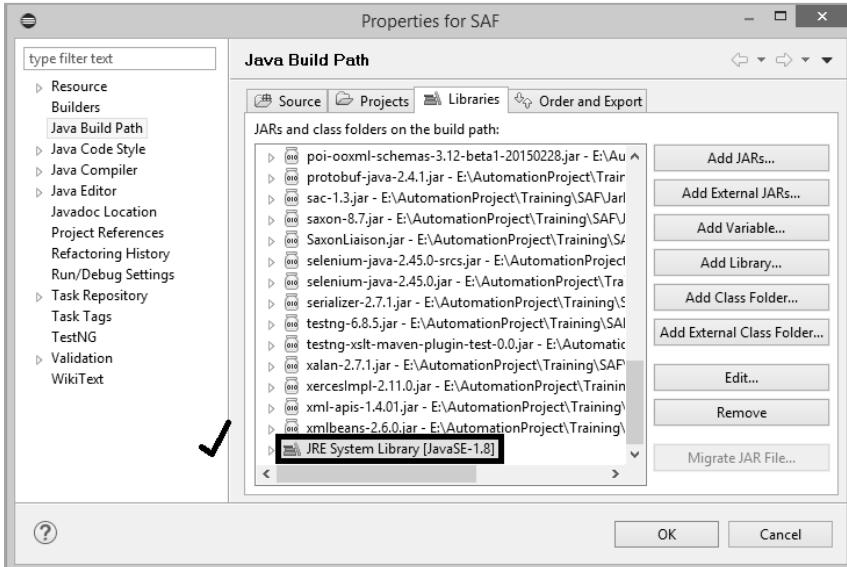
Select JRE System Library and then click the Next button:



Click on the Workspace default JRE or any other relevant option of Java installation details (as per your target machine) and then click on Finish:

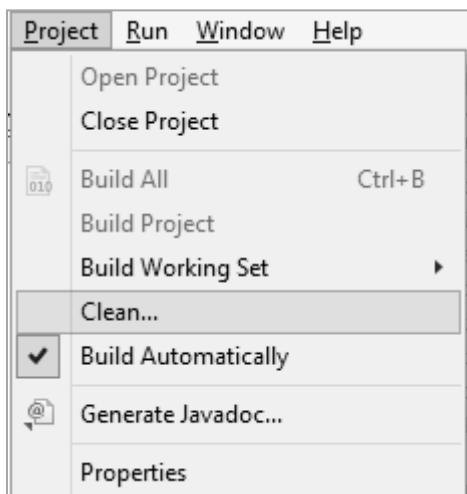


Once JRE System Library is added without any errors, it will be listed in the Libraries section of the Build Path; right click on the Java project-->Build Patch-->Configure Build Path-->Libraries to validate:

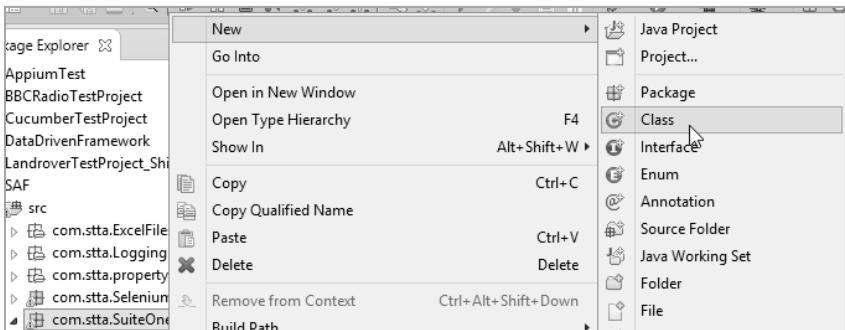


## Final Steps:

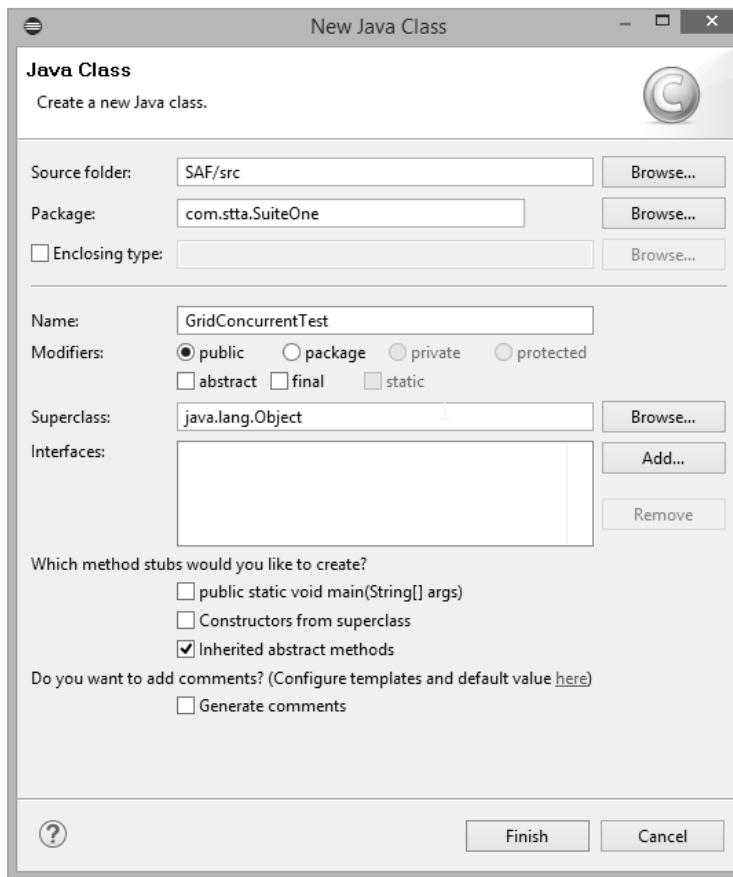
Click on Project--> Clean



Right click on a package to add a java file for Selenium Grid Tests:



Enter the file name as shown below:



Once the file has been created, paste the sample code inside as shown below:

All imports go next to the package name:

The screenshot shows a Java code editor with several tabs at the top: LaunchChrom..., TestRun.java, bbcradio.fe..., bbcradio.java, LoginPage, and GridConcurrentTest.java. The code in the main window is as follows:

```

1 package com.stta.SuiteOne;
2 //Import section of the code written in Java programming
3 import org.testng.annotations.AfterTest;
4 import org.testng.annotations.BeforeTest;
5 import org.testng.annotations.Parameters;
6 import org.testng.annotations.Test;
7 import org.openqa.selenium.By;
8 import static org.testng.Assert.fail;
9 import java.net.MalformedURLException;
10 import java.net.URL;
11 import java.util.concurrent.TimeUnit;
12 import org.openqa.selenium.Capabilities;
13 import org.openqa.selenium.WebDriver;
14 import org.openqa.selenium.remote.DesiredCapabilities;
15 import org.openqa.selenium.remote.RemoteWebDriver;
16 import org.openqa.selenium.support.ui.ExpectedConditions;
17 import org.openqa.selenium.support.ui.WebDriverWait;
18
19 public class GridConcurrentTest {
20
21 }
22

```

All Java programming code samples go inside the class file:

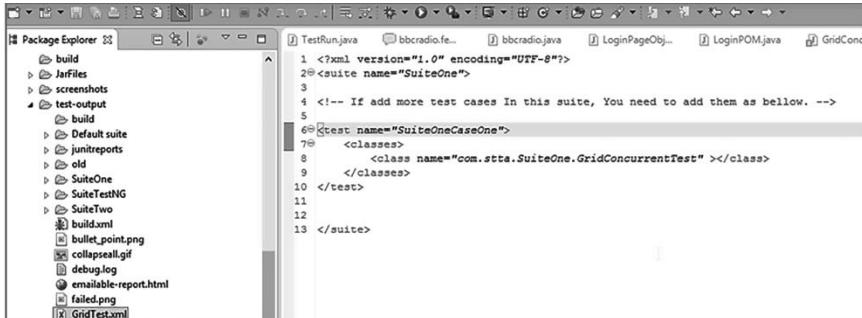
```

public class GridConcurrentTest {
    WebDriver driver = null;
    private StringBuffer verificationErrors = new StringBuffer();
    // Transfer the details of the platform and driver and URL to launch
    @Parameters({ "strPlatform", "driver", "url" })
    @BeforeTest(alwaysRun = true)
    public void setup(String strPlatform, RemoteWebDriver driver, String url) throws MalformedURLException {
        DesiredCapabilities bwsrSettings = new DesiredCapabilities();
        bwsrSettings.setPlatform(org.openqa.selenium.Platform.WINDOWS);
        if (driver.equalsIgnoreCase("Internet Explorer"))
            bwsrSettings = DesiredCapabilities.internetExplorer();
        if (driver.equalsIgnoreCase("Firefox"))
            bwsrSettings = DesiredCapabilities.firefox();
        if (driver.equalsIgnoreCase("Chrome"))
            bwsrSettings = DesiredCapabilities.chrome();

        // This section launches the driver within the grid node by talking to hub of Selenium Grid.
        driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), bwsrSettings);
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        driver.manage().window().maximize();
        driver.get(url);
    }
    // Simple test method to execute.
    @Test(description = "Sample Grid Test")
    public void testCalc() throws InterruptedException {
        driver.navigate().to("www.webdriverinselenium.com");
    }
    @AfterTest
    public void afterTest() {
        driver.quit();
    }
}

```

Now write a TestNG file to get the configurations for Test Execution:



Code Sample for TestNG xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<suite name="SuiteOne">
<!-- Selenium Grid Tests -->
<test name="SuiteOneCaseOne">
<classes>
<class name="com.stta.SuiteOne.GridConcurrentTest" ></class>
</classes>
</test>
</suite>
```

Configure multiple browser and platform combinations in TestNG file to run using Selenium Grid:

```
<suite name=" SuiteOne" verbose="1" thread-count="4"
parallel="tests">
<tests>
<!--Test Execution Parameters-->
<test name="Windows+Safari Test" >
<parameters>
<parameter name="platform" value="Windows" />
<parameter name="browser" value="Safari " />
<parameter name="url" value="http://webdriverinselenium.com" />
</parameters>
```

```
<classes>
<class name="com.stta.SuiteOne.GridConcurrentTest" />
</classes>
</test>
<test name="Windows+InternetExplorer Test" >
<parameters>
<parameter name="platform" value="Windows" />
<parameter name="browser" value="InternetExplorer " />
<parameter name="url" value="http://webdriverinselenium.com"
/>
</parameters>
<classes>
<class name="com.stta.SuiteOne.GridConcurrentTest" />
</classes>
</test>
<test name="Windows+Firefox Test" >
<parameters>
<parameter name="platform" value="Windows" />
<parameter name="browser" value="Firefox" />
<parameter name="url" value="http://webdriverinselenium.com"
/>
</parameters>
<classes>
<class name="com.stta.SuiteOne.GridConcurrentTest" />
</classes>
</test>
<test name="Windows+chrome Test" >
<parameters>
<parameter name="platform" value="Windows" />
<parameter name="browser" value="chrome" />
<parameter name="url" value="http://webdriverinselenium.com"
/>
</parameters>
<classes>
<class name="com.stta.SuiteOne.GridConcurrentTest" />
```

```
</classes>
</test>
</tests>
</suite>
```

Execute the Selenium Grid test by following the option in TestNG:



## Protractor End-to-End Automation Framework

*Programming language used in this section: Java Script*

One of the best tools in the market to compete with Selenium Webdriver is Protractor (javascript programming)! It is primarily designed for AngularJS based applications and it acts like a abstraction layer around Selenium Webdriver to talk to AngularJS applications. Since it is a Node.js-based javascript framework and acts as a wrapper on top of Selenium, it works as a protractor with additional capability when running against AngularJS based applications and at the same time it works as a normal Selenium test for any web application other than AngularJS! So, the tests are compatible and fast enough to run and provide the test results!

Protractor-based framework has been integrated with Jasmine and provides the feasibility to extend the tests from unit level to end to end level BDDs within the framework!

The primary reason why protractor is super good is because some of the object elements of angularJS applications are not included in Selenium and supported by protractor:

ng-repeater

ng-controller

ng-model

Example:

```
<div ng-app="ShoppingApp" ng-controller="myView">
```

Protractor API acts as a wrapper around Webdriver and extends the support for some of the wonderful locator strategies such as:

By.binding

By.repeater

By.textarea

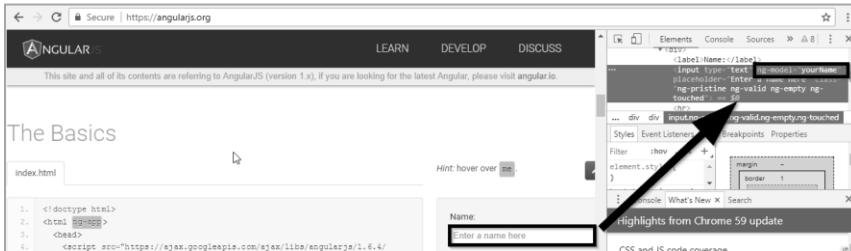
By.model

Webelement.all

WaitforAngular

Example:

Navigate to angularjs.org website and right click on the 'Name' text box and then press Inspect Element (to identify the object properties or xpath elements).



The entire section's outer HTML looks as shown below:

```
<input type="text" ng-model="yourName" placeholder="Enter a name here" class="ng-pristine ng-valid ng-empty ng-touched">
```

This ng-model object within the angularJS application has been supported by Protractor and it can be identified to enter the Name in the code sample of AngularJS given below:

```
describe('angularjs webpage enter name', function() {
it('should add a name', function() {
browser.get('https://angularjs.org');
element(by.model('yourName')).sendKeys('Sample AngularJS
Test');
});
```

Let us understand how to construct a Cucumber BDD based Protractor framework:

Feature: Test AngularJS Home Page

Scenario Outline: Enter your name in AngularJS Homepage

Given I open angularJS.org

When I type <Name> in the Name text box

Then I must be able to see the <Name> display

Examples:

|Name|

Tester
Expert

The configuration file for Cucumber to talk to Protractor based Step Definitions:

```
exports.config = {
  specs: ['protractortest/functionaltest/features/*.feature' ],
  framework: 'cucumber',
  cucumberOpts: {
    require: ['features/step_definitions/*_steps.js',
    'features/support/*.js'],
    format: 'pretty'
  },
  capabilities: { browserName: 'firefox' }
};
```

Step Definitions:

```
this.Given(/^I open angularJS.org$/, function (callback)
{
  browser.get('https://angularjs.org');
});

this.When(/^I type \"([^\"]*)\" in the Name text box$/, function
(nametext,callback)
{
  element(by.model('yourName')).sendKeys(nametext);
});

this.Then(/^I must be able to see the \"([^\"]*)\" display$/, function
(nametext,callback)
{
  var welcoming = element(by.binding(nametext));
  expect(welcoming.getText()).toEqual('Hello '+nametext);
});
```

## Automation Tool Selection-Know How

Important tools in the test automation are always compared to find out the best among them – so the right tool is picked every time before automating the test projects. As an example, Selenium and UFT are provided here with the comparison elements:

FEATURES	SELENIUM (IDE/WEB DRIVER/GRID)	UNIFIED FUNCTIONAL TESTING
<b>License</b>	Open source and free of cost	Licensed and very Expensive.
<b>Framework</b>	Integrate with Selenium, Eclipse, Maven, Ant, Jenkins, Hudson, TestNG, JUnit, SVN, XSLT Reports	Integrate with HP ALM (Application Life Cycle Management)
<b>Continuous Integration</b>	Possible through Jenkins, Hudson, Cruise Control	Possible through HP ALM, Jenkins
<b>Script Creation Time</b>	High	Low
<b>Image-Based Tests</b>	Need Extensions like SIKULI	Possible
<b>Application support</b>	Web applications only.	Client server applications (like built in TCL/TK and PowerBuilder)
<b>Browser support</b>	Supports IE, Firefox, Opera, Safari, etc.	Supports only IE and Firefox.
<b>Operating System/Platform</b>	Windows PC, MAC, UNIX platforms.	Windows Platform only.

<b>Object Oriented Language support and Scalability</b>	Supports Java, .Net, Perl, PHP, Python, C# and Ruby.	Supports VBScript or JavaScript.
<b>Usage</b>	Needs a quite Expertise.	Easy to learn.
<b>Technical support</b>	No official technical support.	Good technical support via phone, mail, web forum.
<b>Test Development Environment</b>	Test scripts can be developed in various IDEs like Eclipse, Visual Studio, TFS (Team Foundation Server), Net beans, etc.	Developed only in QTP.
<b>Support for File Upload</b>	Not available	Supports all kinds of File upload
<b>Database Applications</b>	Not so convenient.	Very well with Database applications.
<b>Report Generation</b>	Integration with Jenkins can give good reporting.	Quality Center has built-in dashboards.

Reference: " Comparative Study of Automated Testing Tools: Selenium and Quick Test Professional." by S.Rajeevan, B.Sathiyan

## Test Requirements

*Earlier sections talked about the framework and the comparison between automation tools. The next step is to understand how the framework has to be constructed technically. But strong fundamentals on test requirements, test plans, scenarios and test cases are required in order to proceed on automation essentials. So, this chapter basically explains the testing basics as a prerequisite.*

Requirements are any statement describing the functionality that is expected of the system. It is always needed in order to be able to design the test and it is also important to understand that it can be introduced at any given level. The more details in the requirement, results in easier test scenarios. Although a set of requirements can be provided, it can never be complete or testable as required.

An example of a requirement is that a customer should be able to place orders online or cancel online orders. This would be classified as a high level requirement. In order to deal with this requirement, we have to go through the concept of scenarios.

The requirements should always be needed to design a test. The statement describes any behavior or any functionality that someone expects from the system. Requirements can come in different levels of details .The more the requirements for us the easier it is to test because we know that the requirements are never complete and will never be testable.

**Requirements** are illustrations that describe what is expected of the system, it is always important that before you move to the next step of software testing, you need to understand what is expected of the system in the requirements document.

Please refer the Sample Test Requirement section to understand how testable requirements look and how to derive the scenarios from the requirements.

## Test Plans

Master Test Plan contains a set of important project details on testing activities such as Summary, Test Items, Environment, References, Comprehensive Assessment, Identification of Intermittent issues, Summary of Results, Evaluation, Limitations, Summary of Activities and Approvals.

Refer the sample test plan section for detailed section specific examples.

## Test Scenarios

Sometimes the requirements may not be elaborate or complete; this could simply be because, for every requirement, there may be several or multiple **scenarios** expected to give similar results, these attributed to conditions that may happen along the line of production.

The system needs to be tested on these multiple scenarios in order to increase the chances of having a more reliable and efficient system free from errors, which is the ultimate target of the project as well as the software testing team. This also helps in having a much more predictable system.

## Test Cases

*More Test Case Examples for this section:*

<https://github.com/narayananpalani/testautomation/tree/master/001FunctionalTestingProject-TrainingInstitute>

In order to make sure that the scenarios are tested from one scenario to another scenario in a systematic way, the team needs to come up with the **test cases**, where all the scenarios will be tested individually. Each case will be tested according to what has been specified in the requirement document e.g. the conditions associated with a particular test case such as a precondition and postcondition which is the expected outcome.

These groups of test cases with similar requirements are bundled up in one to create a **test set**. A test set is a group of test cases that require the same steps to be executed. And they are executed by creating **a test script**; these can be done for both manual and automated testing.

Formula for test case: **Input + precondition -> Postconditions + output**

Test cases are the means by which we test the scenarios. For each scenario, we have a number of test cases. Each test case consists of

a set of inputs that are going to be fed into the system and precondition are all the conditions that exist before a test case.

**For example:**

Preconditions: all facts about the input e.g. card used, order has not been canceled, card is still valid etc.

Input =order number

Postconditions: output = confirmation message e.g. order has been canceled. As a result of order cancelation, card has been refunded with the correct amount.

Refer to the sample test cases section for the detailed test cases with each column and with example test cases.

# THE BEST THINGS TO LEARN SELENIUM

## for high paid job opportunities

In the midst of a huge job opportunities, learning simple selenium webdriver scripts doesn't hurt your current job and it may land you in a nice job offer from top computer organization!

### LOCATORS

If you haven't been learnt how xpath works in identifying objects in automation tests, learning locators on Selenium WebDriver is the right step to start learning xpath mechanism.

### PAGE OBJECT MODEL

It's easy to take the simplest way of designing the selenium scripts using page objects and design the hierarchy of java files which facilitate an easy maintainence.

### DRIVERS

Cross browser testing has been carried out by choosing different browser drivers and it is important to know how to refer them in selenium scripts.

### SCRIPTING TECHNIQUES

Your scripts execute test cases and running them automatically through build tools such as maven or ant or jenkins need expertise in clever scripting.

### EXCEPTION HANDLING

With open source tools usage in automation frameworks, there are millions of ways of handling exceptions. Knowing such exception handling come by experience.

### FRAMEWORK DESIGN

The testing teams need best automation tools to build within a framework of features required to capture screenshots, logs, data input and outputs. Knowing and designing such frameworks need strong scripting expertise.

## Agile Scrum-based Testing-Continuous Integration

*Basic sections of this book explained about automation framework, basics on test requirements etc. But the main reason to learn automation is to implement in most challenging life cycles such as Agile-based Continuous Integration. So the following section drives through the fundamentals on Scrum-based Agile Testing.*

Scrum is described as an iterative and incremental agile software development framework for managing product development. Scrum focuses on delivering the requirement on time as well as responding to emerging requirements promptly.

- Scrum makes everything visible and allows clients to experience each part of the development.
- It also helps in keeping track of the project development.
- Helps to keep up with time to avoid extra costs that come with additional development time.
- Scrum allows you to see the problems in good time in order to make necessary adjustments before it is too late.

### Why do we need Scrum?

When we are planning to develop a software we don't know what is going to happen in a few weeks time and so it becomes difficult to keep track of the project progress. Delays in the project development burdens the project with additional cost as you will need additional development time. This will upset the client.

Scrum helps to keep track of the project progress and allows the team to know exactly what's going on and where adjustments are required to finish the project on time. Similarly, Automation Frameworks such as BDD are built from the beginning when the backlogs are constructed in Step1 as given below.

### Step 1 - Create a Product backlog

The backlog contains all the features that the client would like to have including their dream system and imagination, but that does not mean everything will be included in the project. These features got converted to Cucumber-based .feature files in automation frameworks (still the automation scripts are not ready - so the scripts status in Step1 would be pending).

## **Step 2 – Estimate and prioritize**

Once the product backlog is ready, product owner estimates the timescale for the amount of work required for each feature providing the total amount of work involved for the release backlog.

The product owner then prioritizes the features, the most important one gets placed at the top of the list and the less important one goes at the bottom of the list. He then picks the features that will be on the release and creates a release backlog.

## **Step3-Sprint**

Sprint is a short duration milestone that allows the team to prepare a chunk of the project to get in to ship ready state. Sprint is generally ranged between 3 and 30 days depending on the product release backlog. A project usually includes several sprints.

At the beginning of the Sprint, a team will have Sprint Planning Meeting where the product owner and the team decide what will be done. They select the features which occupy the highest priority from the release backlog.

Sprint Planning Meeting has two parts:

- 1) Team and product owner to decide which features to develop
- 2) The team plans out the sprint

The selected backlog features are assigned to the sprint backlog and to different team members. Each team member in automation framework assigned to list of .feature files in BDD (refer BDD section) design their automation scripts to get ready.

Every day the team meets for short meeting which is called Daily Scrum. Each team member answers three questions during the scrum meeting.

- 1) What have you done on this project since the last daily scrum meeting?
- 2) What do you plan to do on this project between now and the next scrum meeting?
- 3) What stands in your way to meet your commitments to this sprint and this project?

The purpose of the daily scrum is to discuss the progress of the sprint and address any issues that might cause delays in the project. In the sprint, the team must complete the feature/task that was defined for that sprint. Bugs that are related to the features on the backlog should also be fixed.

At the completion each of sprint, the team arranges Sprint Review meeting. In this meeting, the team presents what was developed during this sprint to the product owner and other stakeholders.

This meeting helps to decide what the team should do next. This also helps the client as the client can see the progress of the project and can provide a feedback. This prevents the risk of developing features that the client didn't ask for and also in case of a delay in the development process, the sprint will not be completed on time. This means the project is running behind the schedule and something needs to be done.

Therefore, it is important to monitor the progress of each sprint with burn down chart. This is the best way to monitor the progress and it gives the best visibility of the project. Burn down chart gives day by day progress report and remaining work on the sprint.

Scrum is a simple and effective way to have control over your development process. It ensures that things go according to plan.

## Test Estimation

*Earlier chapter discussed about Sprint-based Agile Testing. This chapter will help in understanding how important is the estimation for automation projects. Due to incorrect estimation, large automation projects led to expensive failures in the history. So this chapter illustrates the details about how to estimate any test automation assignment.*

Projections of Testing Project based on the testing process, data, extracts are basically derived from test estimation using various test estimation techniques. Time and resources required for testing and the number of estimations required for businesses have been calculated before initiating the testing activities.

Test effort, duration, infrastructure, resources, required expertise, tools and need for test environments are estimated and analyzed as part of test estimation.

### Test effort

Amount of time required to write the test scenarios, review of test scenarios, write test cases, test case review, test data preparation, test execution time, defect tracking time, backup time on each test execution cycle due to environment issues, number of test environment and systems required.

### Duration

Schedule for Test Design (Scenario, Test Cases, Test Data, Test Script Preparation on Test Automation), Schedule for Smoke Test, Schedule for Test Execution, Schedule for Test Reports and Sign-off.

### Resources

Number of test resources required for functional (manual, automation) and non functional tests (performance, penetration, usability tests).

### Tools

Required tools for test management, test case management, test configuration management, test automation, defect reporting tool, test report tool.

## **Test Environments**

Smoke Test Needs (Environment Shakedown Tests/Environment Readiness Test), Number of test environments required, Number of systems required in each test environment and their code versions.

## **Cost Estimation**

A cost estimate is the approximation of the cost of a testing program, project or operation such as change request testing. It has a single total value along with a set of component values as part of testing assignment.

Famous cost estimation techniques in the testing projects are classified as top down method, bottom up method, analogues and parametric test estimation.

### **Top-Down Test Estimation**

Efforts are estimated at task level such as cost of test case preparation, test execution etc. Individual work items as part of the testing project have been taken into consideration. This estimation starts from main modules and stepping into downstream of the components and estimate the cost involved in each component.

This means, calculation of the test project duration and the cost by comparing it with the set of the existing projects which were successful.

It is a fast and easy approach compared to other estimation models and the only negative factor is that the bottleneck cases are not taken into consideration such as test environment downtime, poor quality of test design etc.

### **Bottom-Up Test Estimation**

Each low level components of testing project are classified and taken into consideration along with Step 1. When the test estimation is completed for each component, the sum of whole test estimation is calculated for the entire project. Each test resource availability, entry and exit time between the test cycles are also taken into consideration. If a test specialist is involved in the test design of cycle 1 and the test execution of cycle 3, then respective amount of duration is included as part of the estimation. So, the

resource identification, task dependencies between the resources, details on when the task has to be completed and what is the dependency with other test cycles are mapped with details for each test cycle.

This method is a best approach only when the resources are available from the beginning of project initiation or at least at the time of test estimation. Also, test plan should be ready to refer for bottom up test estimation as the plan derives the tasks on each test cycle and test items. So, it is difficult to perform bottom up test estimation in the early stages of the STLC (Software Testing Life Cycle).

### **Analogues Test Estimation**

When the project details are not completely available and the dynamic projects such as continuous integration oriented agile projects are estimated with the help of expert judgment and existing project statistics. In analogues estimation technique, the actual duration and effort vary drastically as the initial estimation is derived with the help of judgments on key factors such as resource availability, test cycle duration etc. Often, it has been proved that the estimation derived from this technique is less accurate. But it is one of the fastest techniques to provide test estimations. It is not suitable for Agile-based sprint cycles as most of the sprint projects are dynamic and this has to be estimated accurately to avoid the pitfalls and failures. This is a proven successful technique for change requests and any testing for small needs.

### **Parametric Test Estimation**

Scalable cost per unit is estimated as part of parametric test estimation based on the statistical data available from the earlier projects.

Cost per line of code for automation project and Cost per hour of automation test execution are some of the key calculations in this method. This is the most relevant test estimation technique for test automation as the LOC (Line of Code) Estimation is highly recommended.

## Time Estimation

Duration of test cycles, test resources availability and the time required to complete the test activities are calculated as part of test time estimation process.

CPM (Critical Path Method), CCM (Critical Chain Method), Resource Leveling, PERT (Program Evaluation and Review Techniques) and Monte Carlo Simulation are some of the famous techniques used as part of the test estimation methods for testing projects.

### Critical Path Method (CPM)

The critical path calculates the longest path of the activities to the end of the testing project. So, the highly critical path where the earliest and the shortest time in each activity can start and finish without extending the project is highly important. The visibility into the critical path allows the test managers to prioritize the activities and take appropriate actions to meet the schedule.

### Critical Chain Method (CCM)

The fundamental idea of CCM is CPM with the key difference in resource limitation. By adding resource limitation to the calculation, this estimation differs from Critical Path Method. If the test resource is extended in Sprint 1 of the Agile Project and not available in Sprint 2 and available only for Sprint 3, this has to be taken as a constraint while calculating the CPM. Adding duration buffers by introducing alternative test resource to testing project schedules to protect the targeted finish date from slippage is a key factor while calculating the time required for the test cycle.

### Resource Leveling on CPM

Changing and amending the test schedule in order to mitigate the risks and changes over the test project is part of Resource Leveling for testing project estimates. Resolving over-allocation of the resources and conflicts over test cycles on test resources are adjusted and amended in order to meet the deadlines of the test projects.

## **Test schedule compression techniques**

As a result of test schedule analysis, test teams may identify a need to compress the test schedule. Schedule compression shortens the test project duration in order to meet the schedule deadlines without reducing the project scope for testing the application.

### **Crashing Compression Technique**

Crashing involves either adding test resources or increasing work (e.g. daily test execution) hours (overtime, weekends) to shorten task duration. Shorter test durations typically result in higher task costs, so project teams must determine, prior to crashing, whether the total cost savings is enough to justify the higher costs for the allocated testing project. Crashing almost always requires cost increases because it usually necessitates new tasks and activities such as test design, execution and reporting. Crashing is a controversial technique because adding test resources can increase project complexity or risk and may ultimately have a negative impact on the test schedule. Crashing does not involve reducing test project scope or eliminating project tasks mentioned in the test plan.

### **Fast Tracking Compression Technique**

Fast tracking is a schedule compression technique in which test project phases or activities usually conducted sequentially are performed in parallel to reduce the overall test duration. If possible, sprints run parallel in order to reduce the total duration of the agile testing project. Extra care must be taken to make sure that parallel work does not create additional work or increase risk in the testing activities such as defect reporting. Fast tracking frequently results in increased complexities in test dependencies, so additional test project controls must be implemented to ensure ongoing and accurate insight into test schedule performance.

### **Program evaluation and review techniques (PERT)**

Probabilistic time estimation is followed to calculate the estimate in order to get the realistic test estimation. The most pessimistic

duration, most likely duration and most optimistic duration are taken into consideration to derive at probabilistic test estimation.

$$\text{PERT} : (P + 4M + O) / 6$$

$$\text{Standard Deviation of a testing task using PERT} : (P - O) / 6$$

$$\text{Variation of a testing task using PERT} : ((P - O) / 6)^2$$

P: The most pessimistic duration

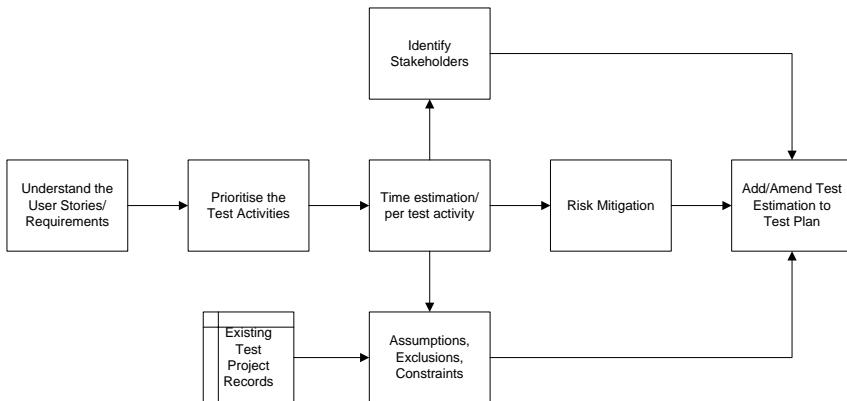
M: The most likely duration

O: The most optimistic duration

## Monte Carlo Simulation

Set of computational algorithms used to get random sampling data to obtain numerical results of test estimation. So, this technique is used mostly using software applications.

## Test estimation flow chart



## Effective Test Estimation Techniques for Test Automation

### Function point-based test estimation

Units of measurement in this method are known as Function Points (FP) and this is calculated for the size of the software functionality.

Following five types are taken as Function Points:

External Inputs: Automation Test Data Sheets

External Outputs: Excel Sheets and Test Outputs to extract test records

External Inquires: SQL Queries/Set of actions requested to validate the test

Internal Logical Files: Logs, screenshot, build and other logical files of automation framework

External Logical Files: Driver Scripts (to interact with automation test engines)

Development of Automation Framework – Function Point Calculation:

Use the following formula to calculate the development project function point count.

$$DFP = (UFP + CFP) * VAF$$

Where:

DFP is the development of Automation Framework function point count

UFP is the unadjusted function point count

CFP is the function points added by the conversion unadjusted function point count

VAF is the value adjustment factor

Enhancement of Automation Framework Function Point Calculation:

Use the following formula to calculate the size for enhancement projects.

$$EFP = [(ADD + CHGA + CFP) * VAFA] + (DEL * VAFB)$$

Where:

EFP is the enhancement of Automation Framework function point count.

ADD is the unadjusted function point count of those functions that were added by the enhancement project.

CHGA is the unadjusted function point count of those functions that were modified by the enhancement of Automation

Framework. This number reflects the functions after the modifications.

CFP is the function point count added by the conversion.

VAFA is the value adjustment factor of the application after the enhancement of Automation Framework.

DEL is the unadjusted function point count of those functions that were deleted by the enhancement of Automation Framework.

VAFB is the value adjustment factor of the application before the enhancement of Automation Framework.

### **Case study on automation framework FP analysis:**

Using the following table for function point weightings:

Factors	Weights		
	Simple	Average	Complex
Number of user inputs(Number of Excel Sheets for user inputs)	3	4	6
Number of user outputs (Test extracts as Excel Files)	4	5	7
Number of user inquiries (SQL Scripts used to extract search results in Automation Framework)	3	4	6
Number of files (Application Under Test)	7	10	15
Number of external interfaces (Driver scripts used to interact with different systems and automation engine)	5	7	10

A system being developed has the following characteristics:

Number of user inputs	10 (simple)
Number of user outputs	7 (simple)
Number of user inquiries	3 (average)
Number of files	6 (average)
Number of external interfaces	1 (complex)

Solution:

Basic Count	Function Types	Weight	Total
10 simple	inputs	10 * 3	30
7 simple	Outputs	7 * 4	28
3 average	Inquiries	3 * 4	12
6 average	Logical files	6 * 10	60
1 complex	Interface	1 * 10	10
		<b>Unadjusted Total</b>	<b>140</b>

### Constructive Cost Model (COCOMO)

Effort and schedule are estimated by analyzing line of code in this COCOMO model.

Case Study on a Sample Test Automation Framework:

COCOMO models include 3 automation framework development types:

Organic: relatively small automation teams develop familiar types of framework in an in-house environment. Most personnel have previous experience on similar automation tools (Effort: PM=2.4 (KDSI)<sup>1.05</sup>; Schedule: TD=2.5 (PM)<sup>0.38</sup>)

Embedded: the automation project may require new tool and technology, unfamiliar algorithms, or an innovative new method for solving the problem of test automation (Effort: PM=3.0 (KDSI)<sup>1.12</sup> Schedule: TD=2.5 (PM)<sup>0.35</sup>)

Semi-detached: having a mixture between organic and embedded types (Effort: PM=3.6 (KDSI)<sup>1.20</sup> Schedule: TD=2.5 (PM)<sup>0.32</sup>)

Where,

PM = person-month (man-month)

KDSI = delivered source instructions, in thousands

TD = number of months estimated for framework development

## COCOMO calculation

Assuming an organic type project with an estimated size = 132,000 lines of code.

Effort:  $PM = 2.4(128)^{1.05} = 392$  person-months

Productivity:  $132,000 \text{ DSU} / 392 \text{ PM} = 336 \text{ DSU/PM}$

Schedule:  $TD = TD = 2.5 (\text{PM})^{0.38} = 24$  months

Avg. Staffing:  $392 \text{ PM} / 24 \text{ months} = 16 \text{ FSP}$

Where,

FSP = full-time-equivalent staff person

## Collective test estimation technique

Automation Leads discuss with stakeholders of the project and get the single outcome as test estimation. Project Managers, Business Analysts, Developers are usually involved in this discussion to contribute inputs to the test estimation.

Sample Test Estimation on Automation Framework Development:

Number	Action Items	Estimated Hours
	<b>Environment Readiness</b>	
1	Test Environment Setup to perform Automation Test	3
2	Automation Test Engine Installation	2
3	Network Connectivity Check	1
	<b>Data Gathering Stage</b>	
4	Object Data Collection	20
5	Understanding the Functionality	30
6	Automation Test Scenario/Test Case Review	2
	<b>Tools Analysis Stage</b>	
7	Object Recognition	3
8	Tool Usability Comparison	5
9	Features Testability	5
10	Tool Support-Framework Compatibility	5

	Analysis	
	<b>Sample Framework Design Stage</b>	
11	Library Architecture Design	20
12	Data Mapping Design	20
13	Driver Script Design	20
14	Utility Script Design	20
	<b>Total Automation Framework Development Hours</b>	<b>156</b>

### Delphi test estimation technique

Test Manager acts as a moderator and collects the estimations from experts confidentially.

Once the estimations are taken from each stakeholder of test automation, he or she takes average of the estimates and update the test plan as automation test estimate.

Sample Delphi Test Estimation on Automation Framework Development:

Number	Action Items	(Average)
		Estimated Hours
	<b>Environment Readiness-From Environment Managers</b>	
1	Test Environment Setup to perform Automation Test	3
2	Automation Test Engine Installation	2
3	Network Connectivity Check	1
	<b>Data Gathering Stage-From Data Architect/Test Leads</b>	
4	Object Data Collection	20
5	Understanding the Functionality	30
6	Automation Test Scenario/Test Case Review	2
	<b>Tools Analysis Stage-From</b>	

<b>Automation Lead</b>		
7	Object Recognition	3
8	Tool Usability Comparison	5
9	Features Testability	5
10	Tool Support-Framework Compatibility Analysis	5
	<b>Sample Framework Design Stage-From Automation Lead/Automation Test Analysts</b>	
11	Library Architecture Design	20
12	Data Mapping Design	20
13	Driver Script Design	20
14	Utility Script Design	20
	<b>Total Automation Framework Development Hours</b>	<b>156</b>

## **Automation Return on Investment**

### **Initial cost to automate (ICA)**

Cost of licenses required for automation tools and cost for developing automation scripts are known as ICA

E.g.: £1000 per test cycle

### **Number of Test Cycles (NTC)**

Total number of test cycles planned for test automation

E.g.: Six Sprint Cycles are planned as part of the Agile Project (NTC: 6)

### **Manual Test Cost Per Test Cycle (MTCPTC)**

Average hourly rates of manual test analysts\*total number of hours is calculated as MTCPTC

Along with the hourly rates, additional costs like virtual server/hardware costs and opportunity costs are included as part of MTCPTC

E.g.: £25\*350=£8750 for first test cycle

### **Automation Test Cost Per Test Cycle (ATCPTC)**

Average hourly rates of automation test analysts\*total number of hours is calculated as ATCPTC

Along with the hourly rates, additional costs like automation test machine/hardware costs and opportunity costs are included as part of ATCPTC

E.g.: £27.8\*350=£9750 for first test cycle

### **Maintenance Time of the Automation Framework for the next Test Cycle (MTAF)**

Approximately 5% to 25% are the average maintenance time for any well developed automation frameworks for long running test projects. This has to be calculated as part of MTAF.

E.g.: 10% (which is 35 hours) for the first test cycle

Multiply the difference between MTCPTC and ATCPTC and sum it over the number of test cycles (yield).

This calculation provides the execution cost difference for test cycles.

ROI = (Total benefit derived from test automation/Total Cost of Test Automation)

$$\text{ROI} = \{\text{NTC } [(\text{MTCPTC}-\text{ATCPTC})+\text{MTAF}]-\text{ICA}\}/\text{ICA}$$

### Calculations:

Note: Assuming that the efforts required for manual testing and automation testing are getting reduced from Sprint 3 onwards.

Test Cycle	Return on Investment
Sprint 1	$(1*((8750-9750)+35)-1000)/1000$
Sprint 2	$(2*((8750-9750)+30)-1000)/1000$
Sprint 3	$(3*((2750-3750)+35)-1000)/1000$
Sprint 4	$(4*((1750-750)+35)-1000)/1000$
Sprint 5	$(5*((1750-750)+35)-1000)/1000$
Sprint 6	$(6*((1750-750)+35)-1000)/1000$

### Final Positive ROI in Sprint 4:

Test Cycle	Return on Investment
Sprint 1	-1.965
Sprint 2	-2.94
Sprint 3	-3.895
Sprint 4	3.14
Sprint 5	4.175
Sprint 6	5.21

As per the ROI calculation, the return gets added to the project only when the automation is implemented successfully for the first three sprints and moved on to fourth sprint.

## Automation Test Scheduling

*This section explains on how to schedule the automation and functional activities for the test project. This will help in scheduling the projects later in tools such as Hudson or Jenkins.*

When the automation test pack is being designed, the most important decision is to plan the Test Scheduling of those Automated Test Scripts. The objective of test automation is to reduce the amount of time spent in Regression Testing. Following are some of the strategies which can be implemented into automation projects:

### Activity list

Set of Test Scripts can be listed as part of the activity list and executed based on the satisfied criteria like most of the test scripts should be scheduled based on the preconditions and postconditions. So, the continuation of the script execution can be maintained till the end.

#### Example1:

Given below in the list, Test 9A and 9B are parallel tests. So, they get completed in 3 hrs time whereas all the other tests can start their execution only when the previous tests are completed (not necessary that the precondition tests have to be passed).

Precondition	Activity List	Duration
Environment Set-up	Sanity Test	1 Hr
Sanity Test	Regression Tests 1	2Hrs
Regression Tests 1	Regression Tests 2	2Hrs
Regression Tests 2	Regression Tests 3	2Hrs
Regression Tests 3	Regression Tests 4	2Hrs
Regression Tests 4	Regression Tests 5	2Hrs
Regression Tests 5	Regression Tests 6	2Hrs
Regression Tests 6	Regression Tests 7	2Hrs
Regression Tests 7	Regression Tests 8	2Hrs
Regression Tests 8	Regression Tests 9A	3 Hrs
Regression Tests 8	Regression Tests 9B	

Regression Tests 9A & 9B	Regression Tests 10	1 Hr
--------------------------	---------------------	------

**Example2:**

Given below in the example, tests are listed based on the priorities where Critical is the top priority followed by High, Medium and the bottom priority is Low. When the project required only important tests to be executed, 2 Critical and 4 High priority tests can be executed in order to complete the test execution.

Precondition	Activity List	Priority
Environment Set-up	Sanity Test	Critical
Sanity Test	Regression Tests 1	High
Regression Tests 1	Regression Tests 2	Medium
Regression Tests 2	Regression Tests 3	Medium
Regression Tests 3	Regression Tests 4	High
Regression Tests 4	Regression Tests 5	Medium
Regression Tests 5	Regression Tests 6	Low
Regression Tests 6	Regression Tests 7	High
Regression Tests 7	Regression Tests 8	Medium
Regression Tests 8	Regression Tests 9A	High
Regression Tests 8	Regression Tests 9B	
Regression Tests 9A & 9B	Regression Tests 10	Critical

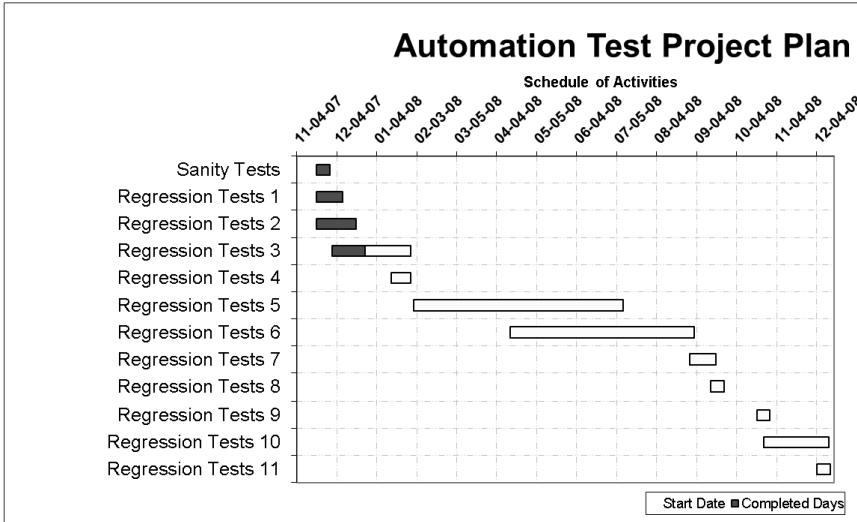
**Bar chart**

The best representation of automation test execution scheduling is possible through Bar Charts when multiple Automation Testers are involved in the test project.

**Example1**

The table below shows the number of days of the tests completed and the remaining number of days to test the application in a sample project.

<b>Task</b>	<b>Start Date</b>	<b>Completed Days</b>	<b>Remaining Days</b>	<b>Lead Contact</b>
Sanity Tests	11/19/07	10	0	Rajavarman
Regression Tests 1	11/19/07	20	0	Jagan
Regression Tests 2	11/19/07	30	0	Karthick
Regression Tests 3	12/01/07	25	35	Siva
Regression Tests 4	01/15/08	0	15	Robin H.
Regression Tests 5	02/01/08	0	160	Jessica S.
Regression Tests 6	04/15/08	0	140	Pete P.
Regression Tests 7	08/30/08	0	20	Robin H.
Regression Tests 8	09/15/08	0	10	Jessica S.
Regression Tests 9	10/20/08	0	10	Jessica S.
Regression Tests 10	10/25/08	0	50	Rick S.
Regression Tests 11	12/05/08	0	10	Tom W.



## Network diagram

List of Tasks of Automation Tests can be converted to network of activities which connects one to another to represent the complete automation tests in graphical representation.

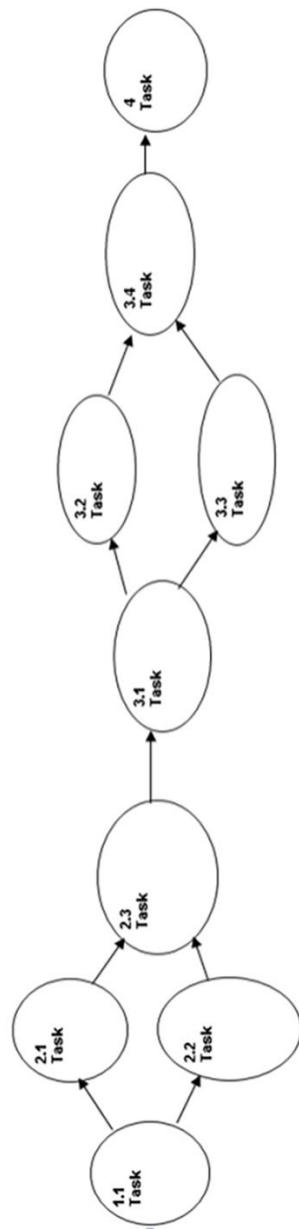
Example 1:

Assume the table given below is the list of activities involved in the automation test project. Each activity has to be updated with the effort estimation, start date, end date and the resource allocated to the activity.

	Project Title	Effort Estimate in XXX	Planned Start Date	Planned End Date	Resource
1	Activity				
1.1	Task:Sanity Test				
1.1.1	Sub Task: Environment Check				
1.1.2	Sub Task: Web Page Validation				
1.1.3	Sub Task: Operating			-	

	System Compatibility Test				
<b>2</b>	<b>Activity</b>				
<b>2.1</b>	<b>Task</b>				
2.1.1	Sub Task:Regression Tests 1				
2.1.2	Sub Task:Regression Tests 2				
2.1.3	Sub Task:Regression Tests 3				
2.1.4	Sub Task:Regression Tests 4				
<b>2.2</b>	<b>Task</b>				
2.2.1	Sub Task:Regression Tests 5A				
2.2.2	Sub Task:Regression Tests 5B				
<b>3</b>	<b>Activity</b>				
<b>3.1</b>	<b>Task</b>				
3.1.1	Sub Task:Regression Tests 6				
3.1.2	Sub Task:Regression Tests 7				
3.1.3	Sub Task:Regression Tests 8				

The Network Diagram given below represents the whole automation tests in single picture:



Once the network of automation tests has been designed, they can be listed further to calculate the time estimated for each tests in the table given below.

		Unit of Time	Unit of Time	Unit of Time	Unit of Time	Unit of Time	Unit of Time	Unit of Time	Unit of Time
<b>1</b>	<b>Activity</b>								
1.1	Task:								
<b>2</b>	<b>Activity</b>								
2.1	Task								
2.2	Task								
2.3	Task								
<b>3</b>	<b>Activity</b>								
3.1	Task								
3.2	Task				horizontal bar				
3.3	Task								
3.4	Task								
<b>4</b>	<b>Activity</b>								

## Unified Functional Testing – Test Automation Basics

*Programming language used in this section: VB Script*

<b>Tools</b>	Quick Test Professional (Unified Functional Testing)
<b>Vendor</b>	HP
<b>Description</b>	<p>Quick Test Professional (QTP) is an automated functional Graphical User Interface (GUI) testing tool that allows the automation of user actions on a <i>web</i> or <i>client</i>-based computer application.</p> <p>It is primarily used for <i>functional regression</i> test automation.</p> <p>QTP uses a scripting language built on top of <b>VBScript</b> to specify the test procedure, and to manipulate the objects and controls of the application under test.</p>
<b>Testing Process</b>	<p>The QuickTest testing process consists of 7 main phases:</p> <ol style="list-style-type: none"> <li>1. Create your test plan</li> <li>2. Recording a session on your application</li> <li>3. Enhancing your test</li> <li>4. Debugging your test</li> <li>5. Running your test on a new version of your application</li> <li>6. Analyzing the test results</li> <li>7. Reporting defects</li> </ol>
<b>Features &amp; Benefits</b>	<ol style="list-style-type: none"> <li>1. Key word driven testing</li> <li>2. Suitable for both client server and web-based application</li> <li>3. VB script as the script language</li> <li>4. Better error handling mechanism</li> <li>5. Excellent data driven testing features</li> </ol>

HP Unified Functional Testing (formerly Quick Test Professional) is a widely accepted tool for web-based testing and standalone applications testing. Though the web browser-based testing has huge competitions with open source tools such as Selenium Webdriver, JMeter and other tools, UFT has high value over the automation stability and maintainability.

Selenium Webdriver is a famous open source tool used by the industry and the only challenge with Selenium is that the test automation for Standalone applications are not possible (Until 2014). The future versions may extend the facilities from time to time. But HP UFT provides support to number of applications in the latest version.

Astra Quick Test is the primary tool developed which leads to develop Mercury QTP. Mercury introduced QTP during Nov 2002 and this has been changed as HP QTP in the year 2007. QTP is an object-based testing tool used for functional and regression testing. Since it follows keyword driven approach, it is flexible to build automation framework around the test engine.

GUI and web applications are supported by QTP and multilingual support is also available as one of the key feature of the tool!

Activex, Visual basic and Web are the default Add-ins to HP UFT whereas the application specific add-ins are available along with license of the tool based on the need.

VB Script is the programming language used in HP UFT and it is very important to understand how to script for UFT using VB Script, so this will reduce the amount of repeated scripts and increase the reusability over the period of time during test cycles.

## **Five phase of test automation using HP UFT**

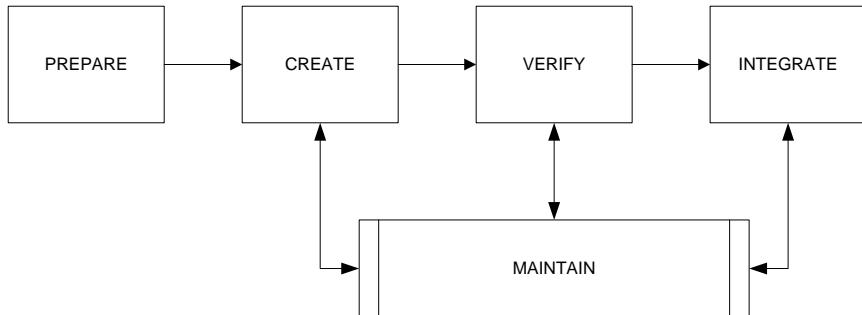
Prepare: Design the test approach and collect the details of testable items, data, requirements and test environment.

Create: Design test scripts.

Verify: Unit testing on the test scripts prepared.

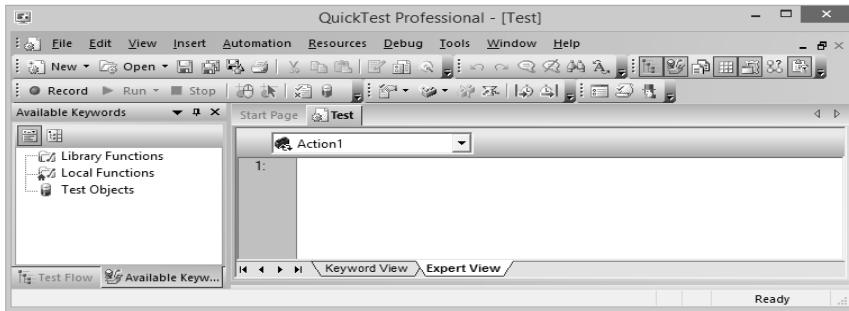
Integrate: Connect to the test management, framework and defect tracking system.

Maintain: Update/Amend/Facilitate the changes to the test scripts for new test cycles.

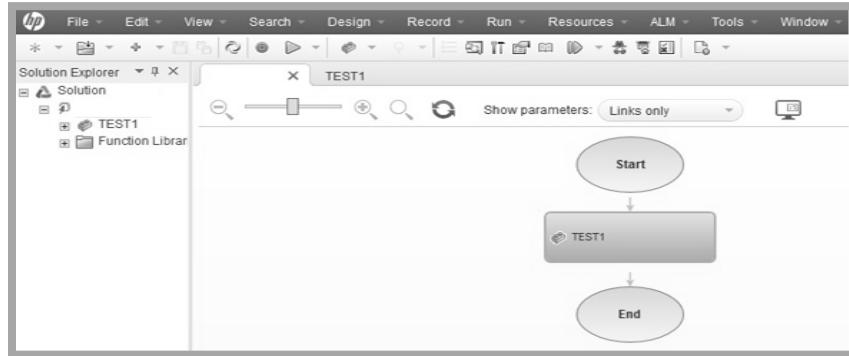


## User interface of HP UFT

Earlier versions of HP QTP 10



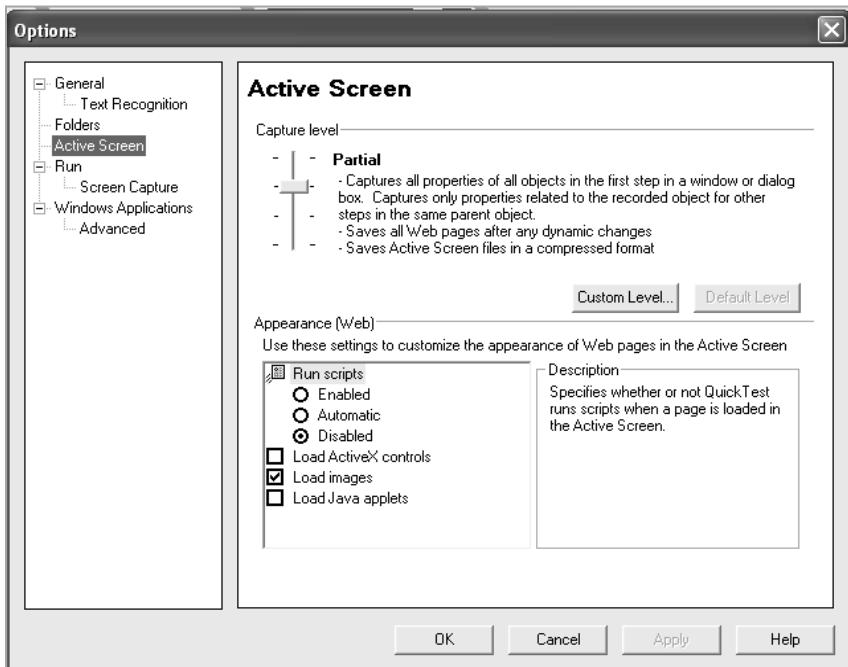
## User Interface in the latest version of HP UFT 11.0



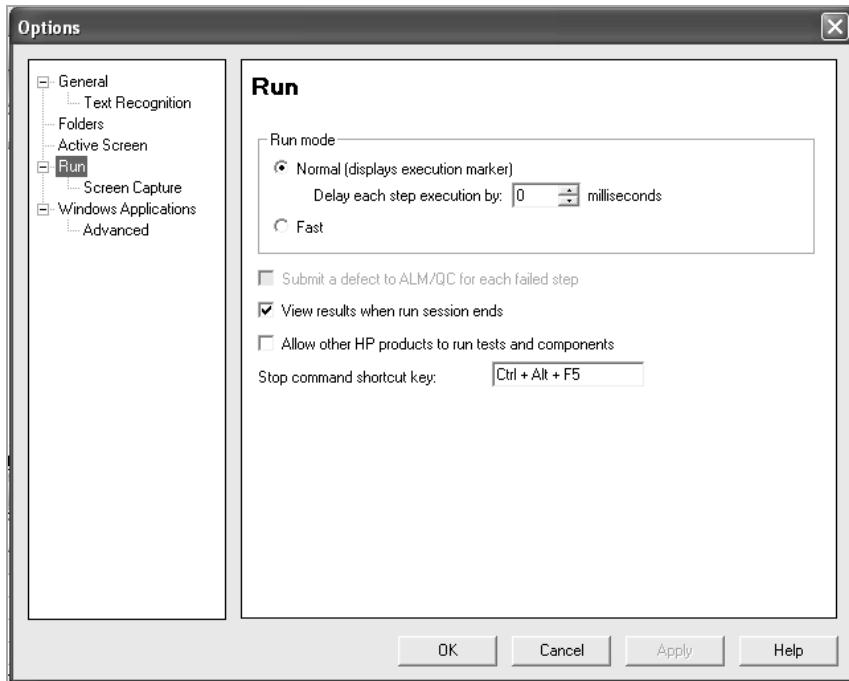
## Capture the level at Run Mode

Navigate to Tools>Options to set the object properties capturing level based on the tests performed in the application under test. If the tool captures most of the objects, it is easy to have a condensed

object repository with possible objects from the application. At the same time, this leads to delay in test execution when the objects are huge in size.

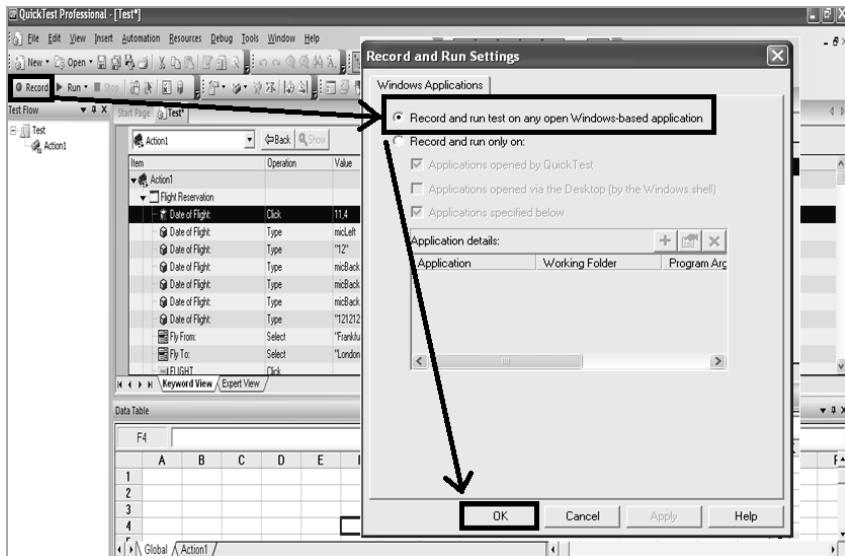


With HP UFT, it is possible to set up the delay between the steps by adding the delay time in the screen given below. When the responsiveness of the application is slow, it is recommended to have slow running tool to test and capture the evidence whereas the test will fail when it runs fast during the slow page navigations of certain web-based tests without any reason. So, it is always recommended to design the test and perform unit tests to make sure that the speed of the test has been controlled.



## Perform a basic test using Record and Playback

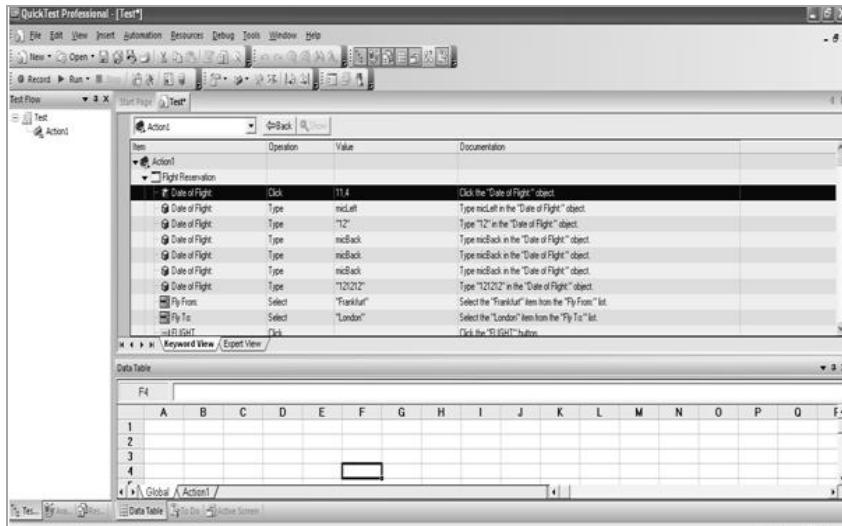
Click on Record and then select 'Record and run test on any open Windows based application', click on the OK button:



Record and Playback is not an efficient way to automate the tests whereas this feature is provided to understand how the automation tool has been designed and how it captures the test application properties by itself. The efficient way of performing automation test is to design the test scripts which are automatically captured during Record in this feature.

Perform a basic test using this feature by,

1. Click on the Record button
2. Navigate to the page where the automation is required and then perform the actions like click, edit, enter etc.
3. Once the test is completed, click on the Stop button and save the test with a valid name.



**Playback:** After recording the scripts, if the Run button is pressed, the entire test gets repeated by the automation tool to see the test results.

## How to launch the browser in HP UFT

System Utility command is widely used to launch the applications using HP UFT as given below:

`SystemUtil.Run ( FileName, Parameters, Path, Operation )`

Example:

Open Internet Explorer by providing path of the exe file:

`SystemUtil.Run "C://Program Files/Internet  
Explorer/IEXPLORE.EXE"`

Alternative way to open Internet Explorer:

`SystemUtil.Run "iexplore.exe", "", "C:\Program Files\Internet  
Explorer\"`

Open a web page along with Internet Explorer:

`SystemUtil.Run "iexplore.exe", "http://www.abctest.com"`

Open a notepad from HP UFT:

`SystemUtil.Run "notepad.exe"`

Open notepad file which is already created in the system:

```
SystemUtil.Run "notepad.exe", "D:\SampleFile.txt"
```

### **How to write a reusable Launch Browser Script:**

```
Public Function Web_LaunchURL (strAppURL,strBrowserType)
```

```
'Input value(s) are:
```

```
'String strAppURL
```

```
'Output Value(s) are:
```

```
SystemUtil.Run "iexplore",strAppURL,"","",""
```

```
Call Web_Execution_log (1, strAppURL&" is launched", "PASS")
```

```
End Function
```

Following are the regularly used QTP functions as part of the script:

### **Input Box Function**

Syntax:

```
InputBox(prompt,title,default,xpos, ypos, helpfile, context)
```

Example:

```
sName= InputBox("Enter Your Name", , "Type Here", , ,  
"help.hlp", 321321)
```

Sample script for the input box command:

```
Option Explicit
```

```
Dim sPrompt, sTitle, sHelpFile, sRes
```

```
Dim vDef
```

```
Dim nYPos, nXPos, nContext
```

```
sPrompt = "What is your name : sTitle = "Employee Data"
```

```
vDef = "Enter Here"
```

```
nXPos = 100 : nYPos = 100 : nContext = 1001
```

```
sHelpFile = "C:\WINDOWS\system32\winhelp.hlp"
```

```
sRes = InputBox(sPrompt, sTitle, vDef, nXPos, nYPos,  
sHelpFile, nContext)
```

### **Message Box Function**

Syntax:

MsgBox(prompt, buttons, title, helpfile, context)

MsgBox "message"

Example:

MsgBox "Hello World"

IF Loop:

Sample Test Script in IF loop for HP UFT:

This if loops checks the length of the sChars. If it is more than 0 then it checks the value if it belongs to the category between 97 and 122. If yes, it displays message box to enter upper case as first letter. Else it displays the words in ascending order as output.

```
Dim sChars  
Dim nCharCode  
sChars = "UnifiedFunctionalTesting"  
If Len(sChars) > 0 Then  
    nCharCode = Asc(sChars)  
    If nCharCode >= 97 And nCharCode <= 122 Then  
        MsgBox "The first char must be uppercase"  
    Else  
        MsgBox nCharCode  
    End If  
End If
```

Program to validate the test data in a VB Script Function:

sText="Welcome"

```
Function CheckText (sText)  
    Dim nChar  
    If Len(sText) > 0 Then  
        nChar = Asc(sText)  
        If nChar >= 65 And nChar <= 90 Then  
            CheckText = "The first character in the word is uppercase"  
        ElseIf iChar >= 97 And nChar <= 122 Then  
            CheckText = "The first character in the word is lowercase"  
        Else  
            CheckText = "The first character isn't alphabetical"
```

```

End If
Else
CheckText = "Please enter something for the sText"
End If
End Function

```

*For further reading and practice, download the projects at  
<https://github.com/narayananpalani/testautomation>*

## **UFT Test Automation Framework using Excel VBA**

Since HP UFT is technically coded with the help of VB Script programming language, it is possible only through automation experts to design and maintain the scripts for testing projects. To enhance the automation tool with handy test data-based excel sheets and run the test from excel-based documents, certain types of Excel VBA-based UFT automation frameworks are designed in order to run the tests from excel sheets. So the changes and amendments to the scripts can be controlled from excel files, and any business user can handle this by getting minimum instructions on how to handle the Excel VBA-based tool to run the automation tests in HP UFT.

### **Types of popular Excel VBA-based test automation frameworks**

- Keyword-driven framework
- Data-driven framework
- Table-driven framework

How keyword-driven frameworks are designed?

HP UFT has to be called from excel file using driver script in order to execute the required test scripts for the testing projects. Press Alt+F11 to open Visual Basic Editor of excel sheet and create the following subroutine:

```

Private Sub ExecuteExcel_Click()
Dim Aut 'As Application
On Error Resume Next
'This is the main script to call HP UFT from Excel sheet:
Set Aut = GetObject(, "QuickTest.Application")

```

```

If Err.Number <> 0 Then
Err.Clear
End If
Set Aut = CreateObject("QuickTest.Application")
If Err.Number <> 0 Then
'If HP UFT is not installed in the test system, this message should be displayed:
MsgBox "HP Unified Functional Test must be installed to run this Framework!"
Exit Sub
End If
On Error GoTo 0
Aut.Launch
Aut.Visible = True
Aut.Options.Run.RunMode = "Normal"
'Location of the test scripts are provided here:
Aut.Open "<Location of the UFT test script folder> ", True
Aut.Test.Run
Set Aut = Nothing ' Release the Application object
End Sub

```

Sample Excel VBA-based keyword-driven framework:

A	B	C	D	E	F
	Parent	Child	Description	Event	Value
Execute	Browser	NewTours		Launch	http://www.abctest.com
	Browser Page	WebEdit	name=userName	Set	sample
	Browser Page	WebEdit	name=passWord	Set	sample
	Browser Page	Image	name=login	Click	

Execute button has been assigned to macro: ExecuteExcel\_Click and test scripts are converted to each column from Column B to F. Any changes in the objects in near future, simple change over the excel sheet will be suffice in order to maintain this keyword-driven framework. Since the test scripts are listed from excel sheet and converted as a test executable script (.vbs) during test execution, HP UFT should be installed to run these tests.

## Sample table-driven framework using HP UFT and Selenium Webdriver

Test projects which need multiple test engines, such as UFT and Selenium, it is possible to develop the code to test from excel-based automation framework.

A	B	C	D
UFT Run	Selenium Run	Reports	
Test Case Name	Test Data File Name	Test Results	Reports
E:\TestScripts\Test1	E:\TestScripts\Test1.xls	PASS	SENT

Script to run HP UFT from Excel VBA:

```

Function UftRun()
Dim xl, bk, s, RowNo, qtpTest, qtpResults, qtp, qtpTestResults
Set qtp = CreateObject("QuickTest.Application")
qtp.Launch
qtp.Visible = True
'qtp.WindowState = "Maximized"
'qtp.Activateview = "ExpertView"
'app.Open "new text document", True
'app.test.Run , True
'Set xl = CreateObject("Excel.Application")
'Set bk = xl.Workbooks.Open("E:\AutomationFramework.xls")
Set s = ActiveWorkbook.Worksheets("Sheet1")
RowNo = 3
s.Cells(RowNo, 3).Value = ""
While Trim(LCase(s.Cells(RowNo, 1))) <> "end"
qtp.Open (s.Cells(RowNo, 1))
'Set s = Nothing
'qtp.Test.Settings.Resources.DataTablePath = s.Cells(RowNo, 2)
Set qtpResults = CreateObject("QuickTest.RunResultsOptions")
qtp.Test.Run qtpResults, True
qtpResults.ResultsLocation = "E:\Results\""
s.Cells(RowNo, 3).Value = "PASS"
RowNo = RowNo + 1
Wend
qtp.Quit

```

```
Set qtp = Nothing
'bk.Close
Set s = Nothing
Set bk = Nothing
Set xl = Nothing
'qtp.Test.Close
OpenFile "E:\TestScripts\Reports\TestReport.html"
AddNewCase 1, "Load Homepage", "Homepage Rendered OK",
"Homepage Rendered OK", "Pass"
AddNewCase 2, "Customer Search", "Customer Record Found",
"Record Not Found", "Fail"
AddNewCase 3, "Close Browser", "Browser Closed OK",
"Browser Closed OK", "Pass"
CloseFile
End Function
```

### Script to run Selenium from Excel VBA

```
Sub Selenium_Run()
Dim selenium As New SeleniumWrapper.WebDriver
selenium.Start "firefox", "http://www.google.com/"
selenium.Open "http://www.google.com"
selenium.Type "name=q", "Paris"
selenium.Click "name=btnG"
selenium.stop
End Sub
```

## Ranorex

*Programming language used in this section: C#*

In order to start the new test, perform the actions suggested below:

File>Click New Solution

New>Test Case

New>Recording solution

Click Record>Global Recording>Start

Note: Click on the web page, and after performing all the actions, click on Stop button.

Right click on the actions and select ‘view code’:

### Ranorex sample code in C#

This section is similar to any other C++ scripting using the different libraries required for the project:

```
// Click 'Save' button to open 'SaveDialog' after click action
repo.application.Buttons.ButtonSave.Click();
```

### Ranorex sample code to click on an image in C#

```
Bitmap bmp = Ranorex.Imaging.Load(@"..\..\sampleImage.bmp");
myRepo.WinFormsApp.Self.Click(MouseButtons.Right,bmp);
myRepo.WinFormsApp.Self.Click(new Location(bmp, new
Imaging.FindOptions(0.90)));
Imaging.FindOptions.Default.Similarity = 0.90;
myRepo.WinFormsApp.Self.Click(bmp);
Report.Success("Image is identified and clicked");
Similarity is checked upto 90 percent in the script above (which is
mentioned as 0.90).
```

## VB Scripting Excel Macro Automation

*Programming language used in this section: VB Scripting*

*More code examples of this section are available from github at:*

*<https://github.com/narayananpalani/testautomation/tree/master/007ExcelVBATestScripting>*

Visual Basic Scripting is a largely used scripting language for automation tools, such as HP Unified Functional Testing (formerly Quick Test Professional) and TestComplete (which supports five scripting languages including VB script!).

VB script is a light version of Microsoft's programming language Visual Basic and default scripting language for ASP (Active Server Pages). Projects that need custom automation frameworks due to unavailability of front-end systems or the projects where SOAP/REST messages have to be automated, excel-based automation frameworks are designed using VB script in order to overcome the challenges! This Excel Macro-based automation is the most important automation skillset in which most of the leading banks, telecommunication and public sector companies look forward to have technical specialist with Excel VBA (Macro) automation expertise. In this section, the introduction to excel macro-based VB script automation is discussed.

### Excel macro-based automation

Excel macros are designed with set of commands and functions that are required to execute and run the automation tests for the testing projects. MS Excel provides a built-in functionality known as Excel formulas (build in macros) which are available for scripting through macros.

	A	B	C	D	E	
1	2					
2	3					
3	4					
4	=sum					
5	(fx) SUM	Adds all the numbers in a range of cells				
6	(fx) SUMIF					
7	(fx) SUMIFS					
8	(fx) SUMPRODUCT					
9	(fx) SUMSQ					
10	(fx) SUMX2MY2					
11	(fx) SUMX2PY2					
	(fx) SUMXMY2					

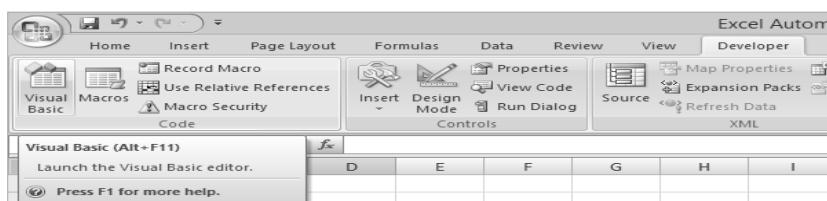
Type values in each cells and navigate to an empty cell, type =SUM and then drag to select the data to see the sum of the selected data. This is an example of a built-in macro of MS Excel.

### Creation of macro

Microsoft VB Editor is the environment used to create or edit macros and VB scripts. The complete automation script and the logic of automation framework can be updated at Microsoft VB Editor.

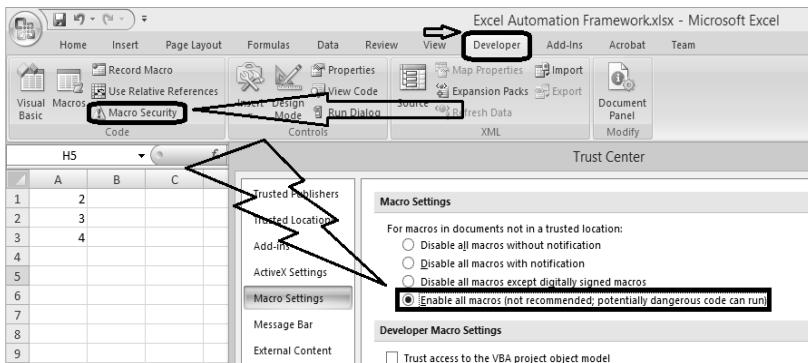
Following are the steps to be followed to create automation framework with subroutines/macros:

- Open an excel sheet (latest version)
- Press Alt+F11 (shortcut key to open MS VB Editor) or alternatively select Developer Tab, click on Visual Basic button.



## Enable excel macros

The scripts and subroutines can be saved in macros only when it is enabled at macro security. In order to enable it, user has to click on Developer, select Macro Security and choose the option of 'Enable all macros' as mentioned below:



## Sample script at Visual Basic Editor

Follow the steps below to create a macro:

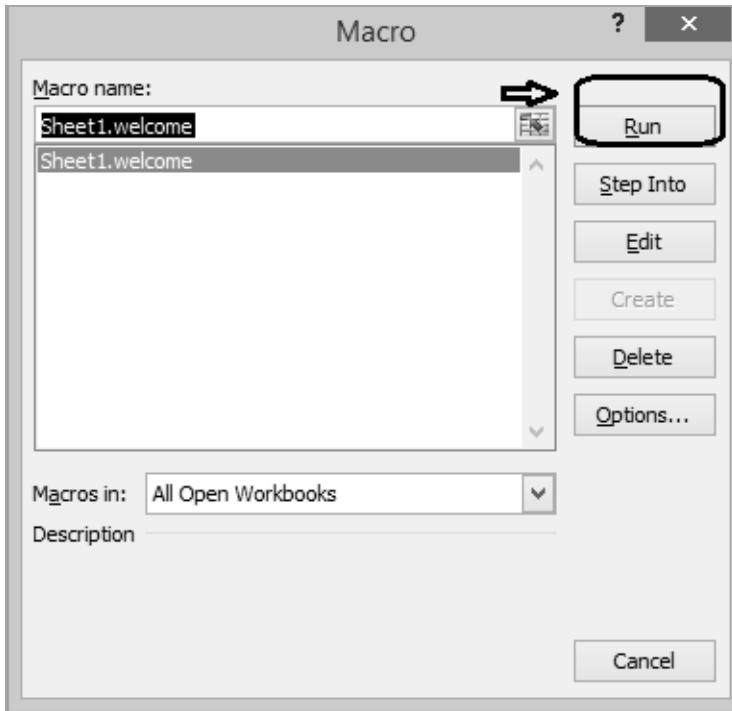
- 1) Open an excel file
- 2) Press Alt + F11
- 3) Verify Microsoft Script Editor is getting opened
- 4) Create a subroutine as:

```
Public Sub welcome()
    MsgBox "Welcome to Excel Macro Automation Framework"
End Sub
```

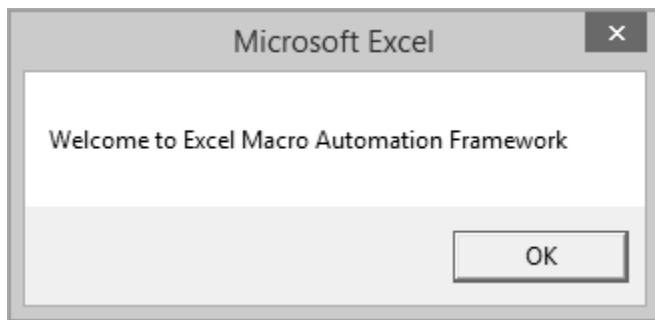
- 5) Save the excel file

Follow the below steps to execute the macro created above:

- 1) Open an excel file
- 2) Press Alt + F8
- 3) Verify macro is getting opened
- 4) Select the Macro name (which is saved earlier) and click on Run button
- 5) The message box should be displayed as designed in the script



Message box:



### Sample script to access the data from the excel sheet

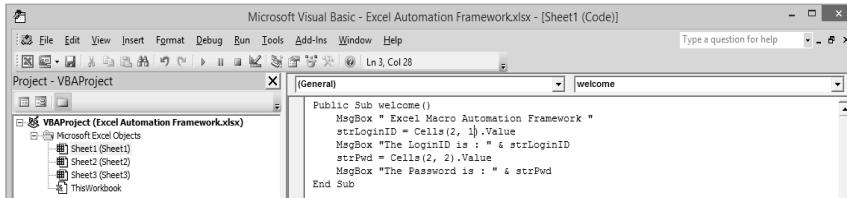
Often it is highly recommended to take the test data from excel sheet and utilize it as part of automation test execution. For example, if the set of HTTP requests has to be sent, the message can be designed by parameterizing the values in the columns of the

excel sheet. If hundred users have to log in to the particular page, the log-in functionality is common. So this can be scripted in excel macro, and the login credentials of hundred users can be listed in the Excel and referred while running the script.

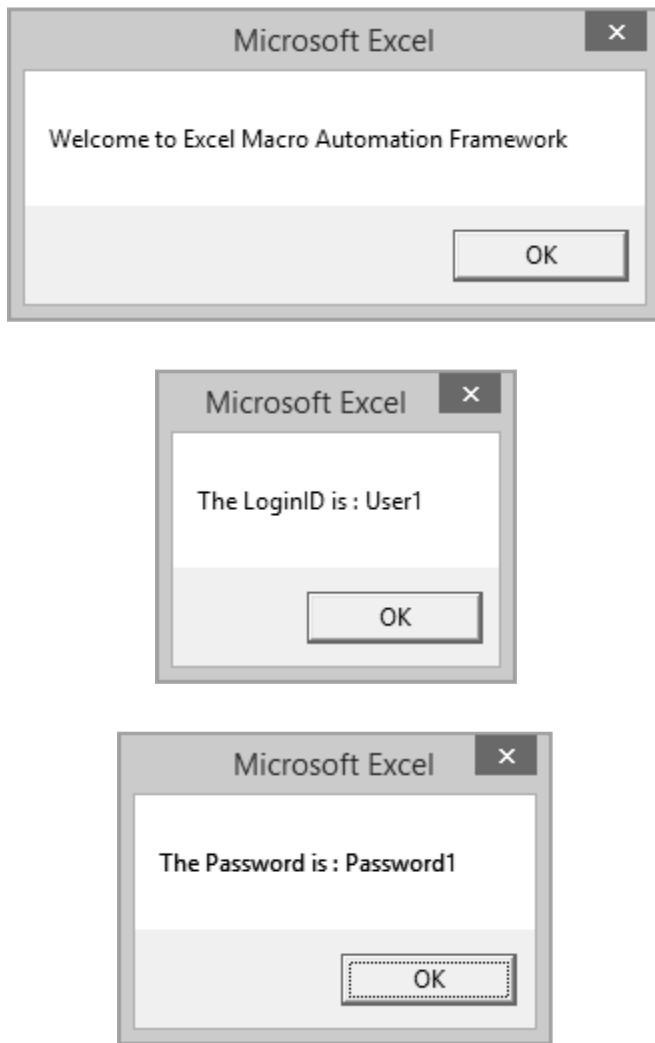
Access/read the text present in the excel file through a macro function and follow the steps given below:

- 1) Create a macro.
- 2) Be clear about the Row-number and Column-number from where the text is to be accessed.
- 3) Use the syntax: Cells (Row-index, Column-index).value.
- 4) The text captured by this step can be stored in a variable.
- 5) Then, it can be displayed through a MsgBox.

```
Public Sub welcome ()  
    MsgBox " Excel Macro Automation Framework "  
    strLoginID = Cells(2, 1).Value  
    MsgBox "The LoginID is : " & strLoginID  
    strPwd = Cells(2, 2).Value  
    MsgBox "The Password is : " & strPwd  
End Sub
```



Result:



Manipulation of the string through Excel macros enables the users to perform the following tasks:

- 1) Copy the text
- 2) Modify the text and/or
- 3) Delete the text

### Steps to perform any string manipulations

- 1) Obtain the text which is to be manipulated.
- 2) Decide the text operation like Append, delete or modify ... which is to be performed.
- 3) Decide whether the new-text is to be displayed in MsgBox or inside the excel file.

Script:

```
Public Sub welcome()
MsgBox " Excel Macro Automation Framework "
strLoginID = Cells(2, 1).Value
strPwd = Cells(2, 2).Value
Cells(2, 3).Value = strLoginID & "is the user id and " &
strLoginID & " is the password."
End Sub
```

Result:

	A	B	C
1	Login ID	Password	Result
2	User1	Password1	User1is the user id and User1 is the password.
3	User2	Password2	

### Example script for case-based conditional loop

Assume that the browser has to be selected for each test automation script and it has to be recognized by entering the number in the excel sheet.

The following script is designed for browser selection:

```
Public Sub welcome()
For Row = 2 To 6
Dim str1 As String
Dim n As Integer
Var = Cells(Row, 1).Value
```

```
'This is the sample of CASE statement'
```

```
Select Case Var
```

```

Case 1: result = "Chrome"
Case 2: result = "Firefox"
Case 3: result = "Internet Explorer"
Case 4: result = "Safari"
Case 5: result = "Opera"

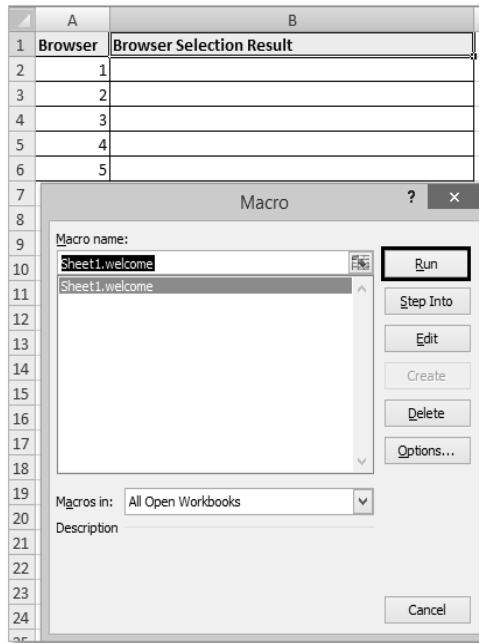
```

```

End Select
If Cells(Row, 1) <= 5 Then
Cells(Row, 2).Value = "The Browser selection is " & result
Else
MsgBox "Enter right number for the browser and continue"
End If
Next
End Sub

```

Press Alt+F8 to execute this script:



Valid result:

	A	B
1	Browser	Browser Selection Result
2	1	The Browser selection is Chrome
3	2	The Browser selection is Firefox
4	3	The Browser selection is Internet Explorer
5	4	The Browser selection is Safari
6	5	The Browser selection is Opera

Instead the value from 1 to 5, enter 7 in one of the rows and validate the result:

The screenshot shows a Microsoft Excel spreadsheet with data in columns A and B. Column A contains row numbers 1 through 6, and column B contains browser names corresponding to the row numbers. Row 7 is empty. A validation dialog box titled "Microsoft Excel" is overlaid on the spreadsheet. The dialog has the message "Enter right number for the browser and continue" and an "OK" button.

A	B
1	Browser
2	1 The Browser selection is Chrome
3	2 The Browser selection is Firefox
4	3 The Browser selection is Internet Explorer
5	4 The Browser selection is Safari
6	5 The Browser selection is Opera
7	
8	
9	

Commonly used VB script functions are listed for reference:

**String** – accepts a number and a character. Returns a string created with the character repeated the given number of times.

**Len** – returns the number of characters in a given string.

**InStr** – accepts two strings and returns whether the second is contained within the first or not.

**Left, Right** – accepts a string and a length and returns a substring of the given length from the beginning or the end of the original string. It also returns the original string if the given number is larger than the actual length of the string.

**Mid** – accepts a string, a starting point and a length and returns a substring of the given length from the starting point of the original string.

**Split** – accepts a string and a delimiter character and returns an array of substrings.

**Date** – returns the current system date. **Time** returns the current system time. **Now** returns the system date and time.

**DateAdd** – adds the specified number of years, months, weeks, days, hours, minutes or seconds to the given date.

**DateDiff** – returns the number of years, months, weeks, days, hours, minutes or seconds between the two given dates. The return value is negative if the first date is after the second date.

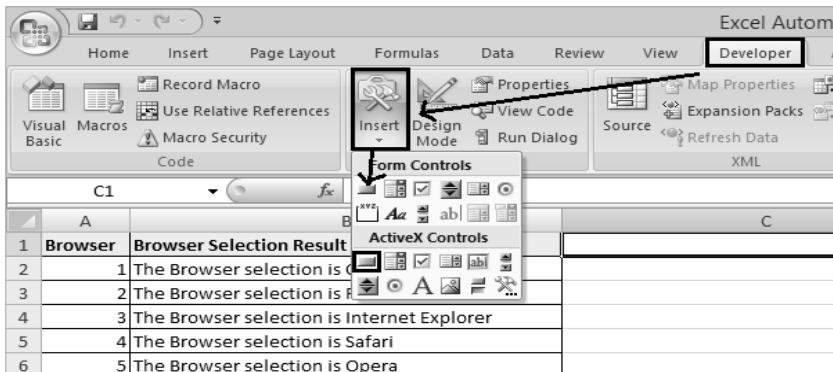
Day, Month, Year, Weekday – accepts a date and returns just the desired portion of the date.

## **GUI form control in Excel VBA**

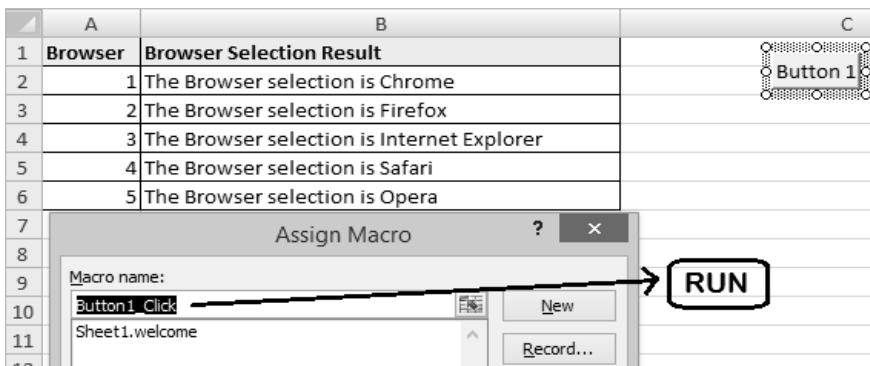
With the help of VB Form UI objects, Excel VBA automation frameworks are designed with available objects to utilize functions like RUN, COMPILE and other functions:

- 1) Text Box
- 2) Button
- 3) Label
- 4) ComboBox
- 5) ListBox
- 6) CheckBox
- 7) OptionBox (RadioButton)
- 8) ToggleButton
- 9) Frame
- 10) TabStrip
- 11) Multi-page
- 12) ScrollBar
- 13) Spin button
- 14) Image
- 15) RefEdit

These objects can be inserted by selecting Developer tab, click Insert button then choose the object:

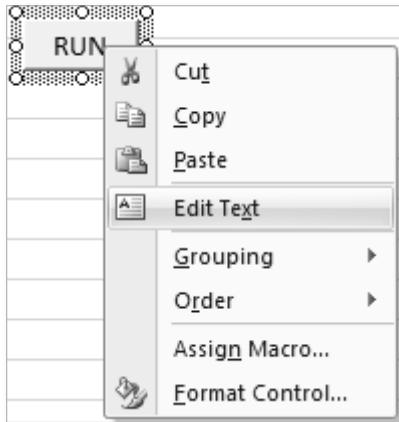


Choose the button and change the name to RUN:



How to edit the name?

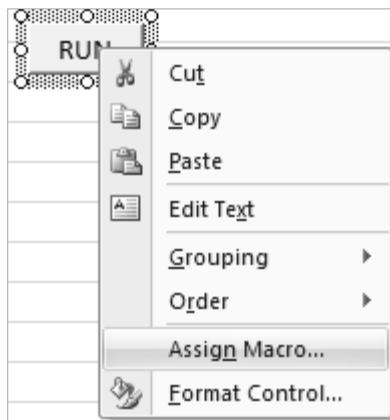
Right click on the button then click on Edit Text as shown below:

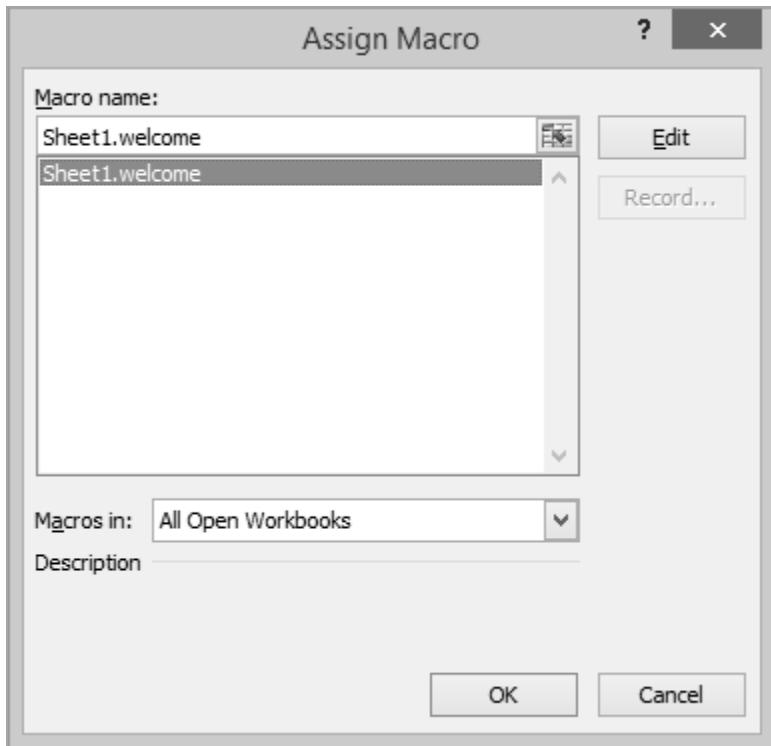


### Assign the script to the button

Once the script is assigned to the button, there is no need to execute every time by pressing Alt+F8. So all users' need is to click on the button RUN to execute!

This can be done by right click on the button, choose 'Assign Macro', select the subroutine developed for the automation framework and click on the button OK.





Once the subroutine is assigned, just click on the RUN button to get the results:

Step1: Click on RUN button

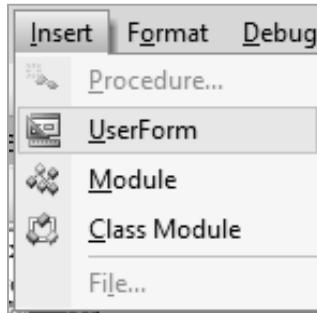
	A	B	C
1	Browser	Browser Selection Result	
2	1		RUN
3	2		
4	3		
5	4		
6	5		

## Step2: Validate the results

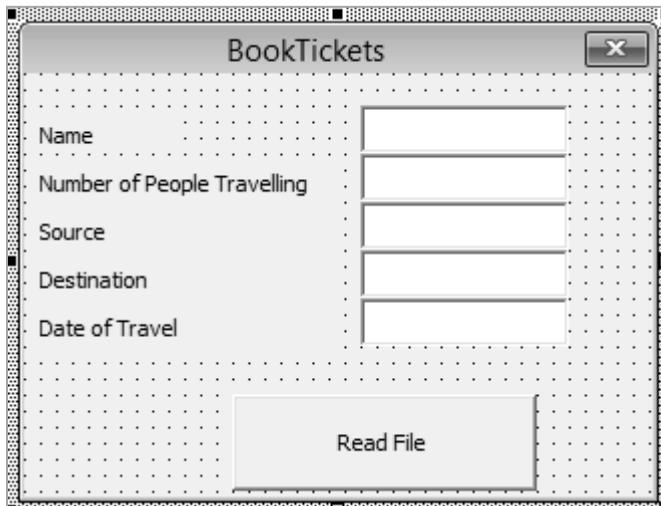
A	B	C
Browser	Browser Selection Result	RUN
1	The Browser selection is Chrome	
2	The Browser selection is Firefox	
3	The Browser selection is Internet Explorer	
4	The Browser selection is Safari	
5	The Browser selection is Opera	

## Control form in Excel VBA

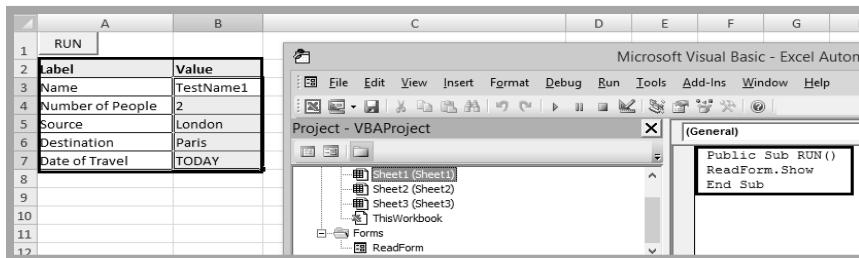
Click Insert and select UserForm in order to create a new form for data inputs:



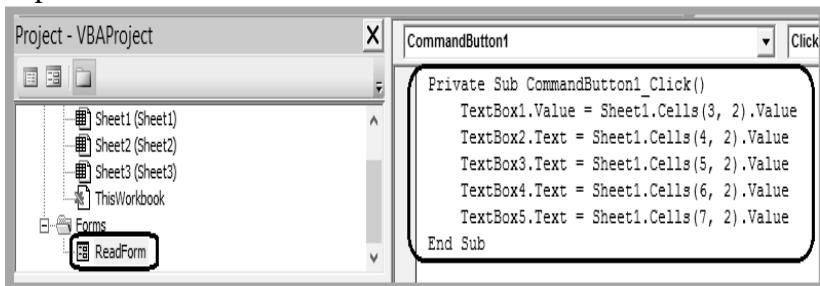
Create a form with following labels, text boxes and button:



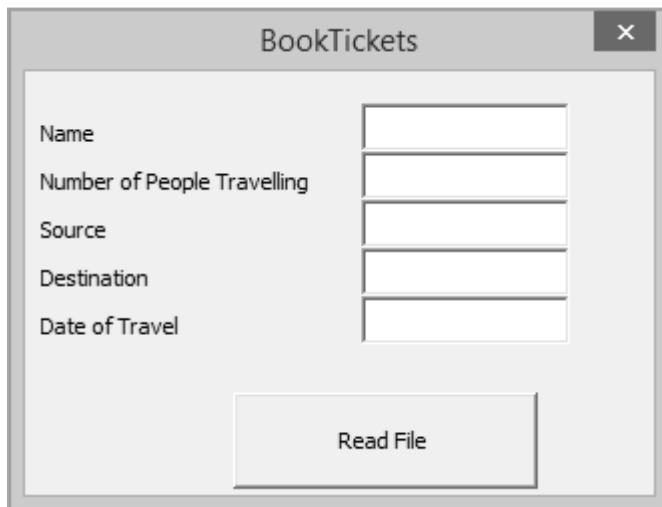
Write the script for sheet1:



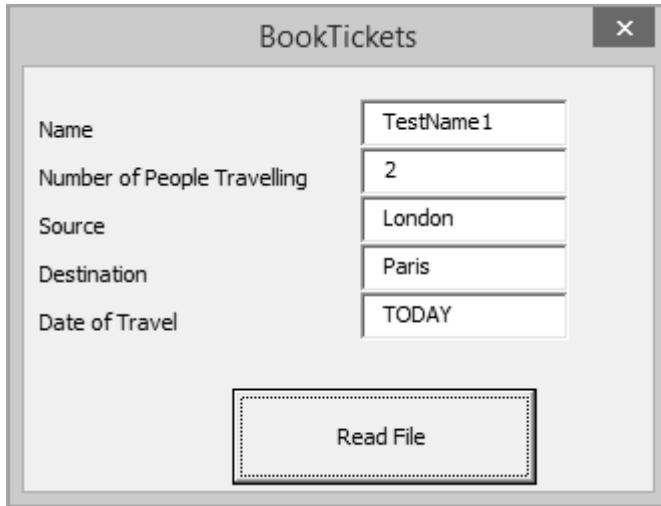
Script for the form:



Basically, clicking on the RUN button opens the form:



Click on the ReadFile button and see the data updated in the sheet:



Similarly, introduce a new button with the name Write File to write back to the same location in excel is possible through the below script:

```

Private Sub CommandButton1_Click()
    TextBox1.Text = Sheet1.Cells(3, 2).Value
    TextBox2.Text = Sheet1.Cells(4, 2).Value
    TextBox3.Text = Sheet1.Cells(5, 2).Value
    TextBox4.Text = Sheet1.Cells(6, 2).Value
    TextBox5.Text = Sheet1.Cells(7, 2).Value
End Sub

Private Sub CommandButton2_Click()
    txtName = TextBox1.Text
    intNumber = TextBox2.Text
    txtSource = TextBox3.Text
    txtDest = TextBox4.Text
    intDate = TextBox5.Text

    Sheet1.Cells(3, 2).Value = txtName
    Sheet1.Cells(4, 2).Value = intNumber
    Sheet1.Cells(5, 2).Value = txtSource
    Sheet1.Cells(6, 2).Value = txtDest
    Sheet1.Cells(7, 2).Value = intDate
End Sub

```

## Script:

In ReadForm:

```

Private Sub CommandButton1_Click()
    TextBox1.Text = Sheet1.Cells(3, 2).Value

```

```
TextBox2.Text = Sheet1.Cells(4, 2).Value  
TextBox3.Text = Sheet1.Cells(5, 2).Value  
TextBox4.Text = Sheet1.Cells(6, 2).Value  
TextBox5.Text = Sheet1.Cells(7, 2).Value  
End Sub
```

```
Private Sub CommandButton2_Click()  
txtName = TextBox1.Text  
intNumber = TextBox2.Text  
txtSource = TextBox3.Text  
txtDest = TextBox4.Text  
intDate = TextBox5.Text
```

```
Sheet1.Cells(3, 2).Value = txtName  
Sheet1.Cells(4, 2).Value = intNumber  
Sheet1.Cells(5, 2).Value = txtSource  
Sheet1.Cells(6, 2).Value = txtDest  
Sheet1.Cells(7, 2).Value = intDate  
End Sub
```

### In Sheet1:

```
Public Sub RUN()  
ReadForm.Show  
End Sub
```

### Nature of the automation frameworks in Excel VBA

Test data sheets are integrated with automation frameworks using Excel VBA, and the framework runs the test automation tool in the background by clicking on the RUN button or the similar button in the excel sheet. This is the most important technology in automation frameworks which is expected to grow large in the next couple of years.

Example 1: Capture the test data in Sheet1, click on Run button which leads to test execution through Selenium Webdriver.

Example 2: Capture the test data in Sheet1, click on Run button which leads to test execution through HP UFT.

Example 3: Capture the test data in Sheet1, click on Run button which leads to test execution through SOAP messages.

Sample project on Excel VBA data comparison:

Step1 – Create a tab 'Source' with data in three columns 'Employee ID,Name,Address' along with Compare button:

	A	B	C	D	E	F
1	Employee ID	Name	Address	<input type="button" value="Compare"/>		
2	1	James	London			
3	2	John	Paris			
4	3	Mark	Amsterdam			
5	4	Williams	California			
6	5					
7						

Step2 – Create a tab 'Target' with data in three columns 'Employee ID,Name,Address'.

## Software Automation Testing Secrets Revealed

The screenshot shows a Microsoft Excel window titled "Worksheet in DATABASE TESTING OVERVIEW...". The ribbon menu is visible at the top, and the formula bar shows "C6". The main area contains a table with the following data:

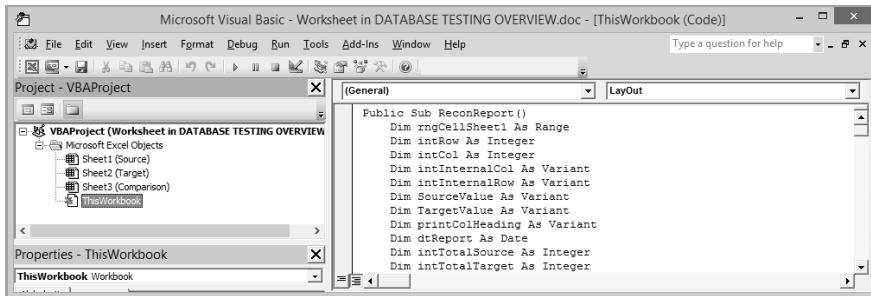
	A	B	C	D	E	F	G	H	I
1	Employee ID	Name	Address						
2		1							
3		2							
4		5	Mumbai						
5		5	Williams						
6		7							
7									

Step3 – Create a blank tab 'Comparison':

The screenshot shows a Microsoft Excel window titled "Worksheet in DATABASE TESTING OVERVIEW...". The ribbon menu is visible at the top, and the formula bar shows "A1". The main area contains a table with the following data:

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							

Step4 – Test Script in ThisWorkbook:



Write the sample scripts given below to understand the Excel VBA code:

```

Public Sub ReconReport()
    Dim rngCellSheet1 As Range
    Dim intRow As Integer
    Dim intCol As Integer
    Dim intInternalCol As Variant
    Dim intInternalRow As Variant
    Dim SourceValue As Variant
    Dim TargetValue As Variant
    Dim printColHeading As Variant
    Dim dtReport As Date
    Dim intTotalSource As Integer
    Dim intTotalTarget As Integer
    Dim strRange As Variant

    Worksheets(1).Name = "Source"
    Worksheets(2).Name = "Target"
    Worksheets(3).Name = "Comparison"

    dtReport = Now()

    Worksheets("Comparison").Range("A1:D1").MergeCells = True
    Worksheets("Comparison").Cells(1, 1).Font.ColorIndex = 5
    Worksheets("Comparison").Cells(1, 1).Font.Bold = True

```

```
Worksheets("Comparison").Cells(1, 1) = "Report Generated On : "
& dtReport
```

```
intRow = 3
```

```
intCol = 1
```

```
Call LayOut(intRow, intCol)
```

```
Worksheets("Comparison").Cells(intRow, intCol) = "DB"
```

```
Worksheets("Comparison").Cells(intRow, intCol).ColumnWidth =
8
```

```
Call LayOut(intRow, intCol + 1)
```

```
Worksheets("Comparison").Cells(intRow, intCol + 1) = "Column"
```

```
Call LayOut(intRow, intCol + 2)
```

```
Worksheets("Comparison").Cells(intRow, intCol + 2) = "Row"
```

```
Worksheets("Comparison").Cells(intRow, intCol +
2).HorizontalAlignment = xlRight
```

```
Call LayOut(intRow, intCol + 3)
```

```
Worksheets("Comparison").Cells(intRow, intCol + 3) = "Source
Value"
```

```
Worksheets("Comparison").Cells(intRow, intCol +
3).ColumnWidth = 15
```

```
Worksheets("Comparison").Cells(intRow, intCol +
3).HorizontalAlignment = xlRight
```

```
Call LayOut(intRow, intCol + 4)
```

```
Worksheets("Comparison").Cells(intRow, intCol + 4) = "Target
Value"
```

```
Worksheets("Comparison").Cells(intRow, intCol +
4).ColumnWidth = 15
```

```
Worksheets("Comparison").Cells(intRow, intCol +
4).HorizontalAlignment = xlRight
```

```
intRow = intRow + 1
```

```
Worksheets("Comparison").Cells(intRow, intCol) = "Source"
```

```

Worksheets("Comparison").Cells(intRow, intCol).Font.Bold =
True
intRow = intRow + 1

intTotalSource = 0
intInternalCol = 1
Do While (intInternalCol <=
Worksheets("Source").UsedRange.Columns.Count)
intInternalRow = 1
printColHeading = "N"
Do While (intInternalRow <=
Worksheets("Source").UsedRange.Columns(intInternalCol).Rows.
Count)
SourceValue = Worksheets("Source").Cells(intInternalRow,
intInternalCol)
TargetValue = Worksheets("Target").Cells(intInternalRow,
intInternalCol)
If Not SourceValue = TargetValue Then
If printColHeading = "N" Then
Worksheets("Comparison").Cells(intRow, intCol + 1) =
Worksheets("Target").Cells(1, intInternalCol)
printColHeading = "Y"
intRow = intRow + 1
End If
Worksheets("Comparison").Cells(intRow, intCol + 2) =
intInternalRow
Worksheets("Comparison").Cells(intRow, intCol + 3) =
SourceValue
Worksheets("Comparison").Cells(intRow, intCol + 4) =
TargetValue
intRow = intRow + 1
intTotalSource = intTotalSource + 1
End If
intInternalRow = intInternalRow + 1
Loop
intInternalCol = intInternalCol + 1

```

## Loop

```
intRow = intRow + 1
```

```
strRange = "A" & intRow & ":D" & intRow
```

```
Worksheets("Comparison").Range(strRange).MergeCells = True  
Worksheets("Comparison").Cells(intRow, 1).Font.ColorIndex = 5  
Worksheets("Comparison").Cells(intRow, 1).Font.Bold = True  
Worksheets("Comparison").Cells(intRow, 1) = "Total Number of  
Difference : " & intTotalSource
```

```
intTotalTarget = 0
```

```
intRow = intRow + 3
```

```
Worksheets("Comparison").Cells(intRow, intCol) = "Target"  
Worksheets("Comparison").Cells(intRow, intCol).Font.Bold =  
True
```

```
intRow = intRow + 1
```

```
intInternalCol = 1
```

```
Do While (intInternalCol <=
```

```
Worksheets("Target").UsedRange.Columns.Count)
```

```
intInternalRow = 1
```

```
printColHeading = "N"
```

```
Do While (intInternalRow <=
```

```
Worksheets("Target").UsedRange.Columns(intInternalCol).Rows.  
Count)
```

```
TargetValue = Worksheets("Target").Cells(intInternalRow,  
intInternalCol)
```

```
SourceValue = Worksheets("Source").Cells(intInternalRow,  
intInternalCol)
```

```
If Not SourceValue = TargetValue Then
```

```
If printColHeading = "N" Then
```

```
Worksheets("Comparison").Cells(intRow, intCol + 1) =  
Worksheets("Source").Cells(1, intInternalCol)
```

```
printColHeading = "Y"
```

```
intRow = intRow + 1
```

```

End If
Worksheets("Comparison").Cells(intRow, intCol + 2) =
intInternalRow
Worksheets("Comparison").Cells(intRow, intCol + 3) =
SourceValue
Worksheets("Comparison").Cells(intRow, intCol + 4) =
TargetValue
intRow = intRow + 1
intTotalTarget = intTotalTarget + 1
End If
intInternalRow = intInternalRow + 1
Loop
intInternalCol = intInternalCol + 1
Loop

intRow = intRow + 1

strRange = "A" & intRow & ":D" & intRow

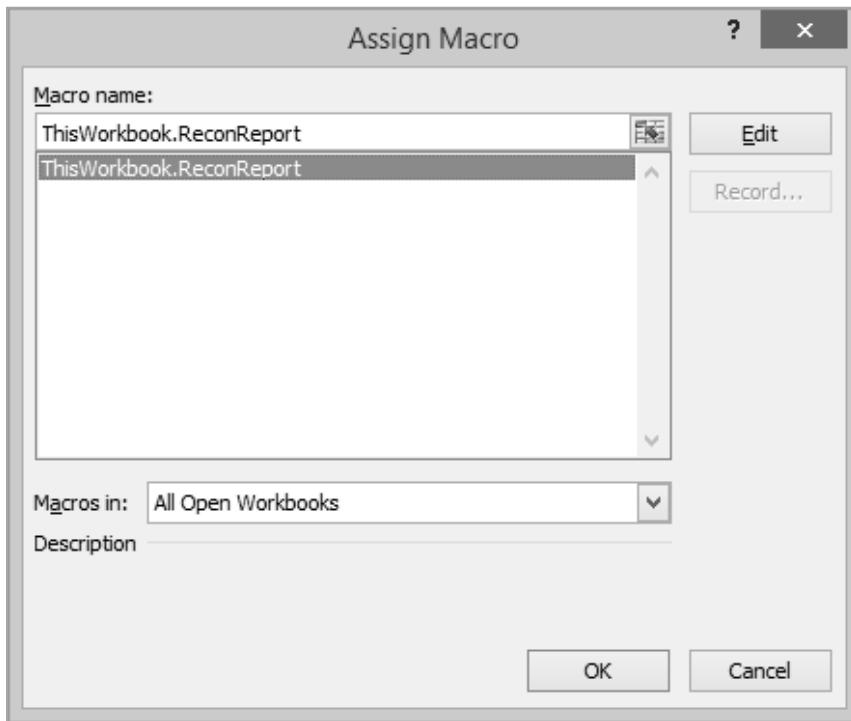
Worksheets("Comparison").Range(strRange).MergeCells = True
Worksheets("Comparison").Cells(intRow, 1).Font.ColorIndex = 5
Worksheets("Comparison").Cells(intRow, 1).Font.Bold = True
Worksheets("Comparison").Cells(intRow, 1) = "Total Number of
Difference : " & intTotalTarget

End Sub

Public Sub LayOut(ByVal row As Integer, ByVal col As Integer)
Worksheets("Comparison").Cells(row, col).Font.Bold = True
Worksheets("Comparison").Cells(row, col).Interior.ColorIndex = 3
Worksheets("Comparison").Cells(row, col).Font.ColorIndex = 0
Worksheets("Comparison").Cells(row, col).Font.Bold = True
End Sub

```

Step5 – Right click on the Compare button in Source Tab and assign following macro:



Step 6 – Test Execution – Click on Compare button and see the result in comparison tab:

	A	B	C	D	E
1	<b>Report Generated On : DATE AND TIME</b>				
2					
3	<b>DB</b>	<b>Column</b>	<b>Row</b>	<b>Source Value</b>	<b>Target Value</b>
4	<b>Source</b>				
5		Employee ID			
6			4	3	5
7			5	4	5
8			6	5	7
9		Name			
10			2	James	
11			3	John	
12			4	Mark	
13		Address			
14			2	London	
15			3	Paris	
16			4	Amsterdam	Mumbai
17			5	California	
18					
19	<b>Total Number of Difference : 10</b>				
20					
21					
22	<b>Target</b>				
23		Employee ID			
24			4	3	5
25			5	4	5
26			6	5	7
27		Name			
28			2	James	
29			3	John	
30			4	Mark	
31		Address			
32			2	London	
33			3	Paris	
34			4	Amsterdam	Mumbai
35			5	California	
36					
37	<b>Total Number of Difference : 10</b>				

## How to Test Excel Sheet?

*Programming language used in this section: VB Script*

In small and medium-level companies, the majority of manual tests are completed with the help of excel sheets for test cases and test execution. But testing the Excel itself as an application under test (AUT) is a different task, and it can be achieved with automation tools like HP UFT (formerly QTP).

Solution for the Excel Testing using HP UFT

In order to compare the excel files and its objects as part of HP UFT automation framework, the sample script is as follows:

We need two excel files to perform the test on this script at the folder C:\excefolder\

File names: *excelfile1.xls* and *excelfile2.xls*

Paste this script in HP UFT or Excel VBA macro to get worked:

Mismatch=0

**Set** myxl = **createobject**("excel.application")

*'To make Excel visible*

myxl.Visible = **True**

*'Open a workbook "excelfile1.xls"*

**Set** Workbook1= myxl.Workbooks.Open("C:\excefolder\excelfile1.xls")

*'Open a workbook "excelfile2.xls"*

**Set** Workbook2= myxl.Workbooks.Open("C:\excefolder\excelfile2.xls")

**Set** mysheet1=Workbook1.Worksheets("Sheet1")

**Set** mysheet2=Workbook2.Worksheets("Sheet1")

*'Compare two sheets cell by cell*

**For Each** cell **In** mysheet1.UsedRange

*'Highlights the cell if cell values not match*

**If** cell.Value <>mysheet2.Range(cell.Address).Value **Then**

*'Highlights the cell if cell values not match*

```
cell.Interior.ColorIndex = 3  
mismatch=1  
End If  
Next  
If Mismatch=0 Then  
Msgbox "No Mismach exists"  
End If  
'close the workbooks  
Workbook1.close  
Workbook2.close  
  
myxl.Quit  
set myxl=nothing
```

For further reading and practice, download the projects at  
<https://github.com/narayananpalani/testautomation>

## IBM Rational Functional Test

Programming language used in this section: Java Programming

RFT uses Java (primary) and VB scripting as programming language, and it consists of the following features:

- Projects
- Test Object Inspector
- Test Object Map
- Recording test scripts
- Replay test scripts
- Debug scripts
- Java scripting
- Create verification points – GUI, Bitmap, Menu
- Databases sample scripts
- Data pool the test cases
- Suite (Batch Run)

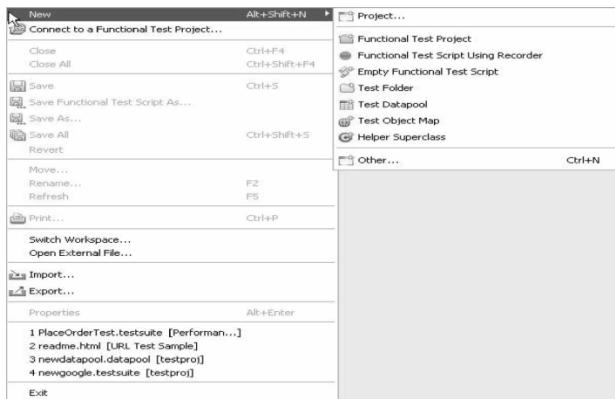
Following menus are available on top level:

- File
- Edit
- Navigate
- Search
- Project
- Script
- Configure
- Run
- Window
- Help

### How to create a project in IBM RFT?

Test project has to be created to automate the application and maintain the test scripts. Moreover, it is advised to have unique project name as per the project title to remember them during the test execution.

Click on File Menu, select New and click on Project ....



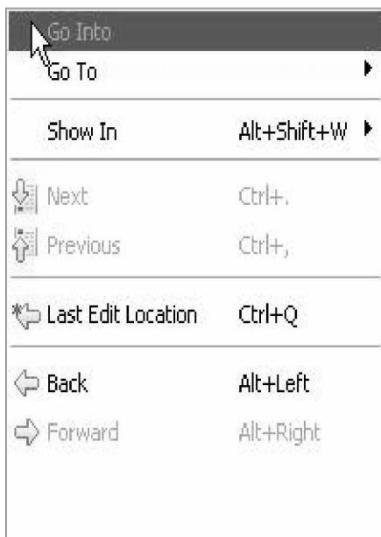
## How to delete a project?

Click on Edit and select Delete:



## How to navigate to different projects in IBM RFT?

Testing teams can align different projects in IBM RFT. If a user needs to navigate to different test project, click on Navigate menu and select Go to:



## How to search for the particular script?

Click on Search in menu and select Search:



## How to run the scripts in IBM RFT?

Click on Script in menu items and select Run button:



## How to create a test automation project using IBM RFT?

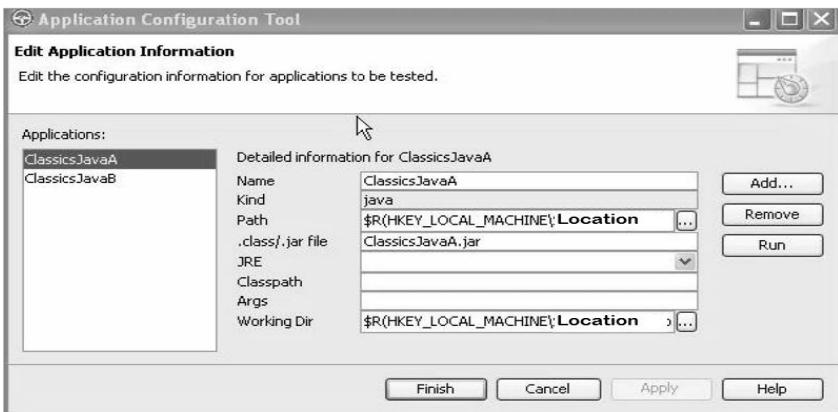
Step1: Test application needs to be configured in order to automate using IBM RFT and navigate to configure tab.

## How to configure the menu and applications in IBM RFT?

Click on Configure and select 'Configure Applications for Testing':

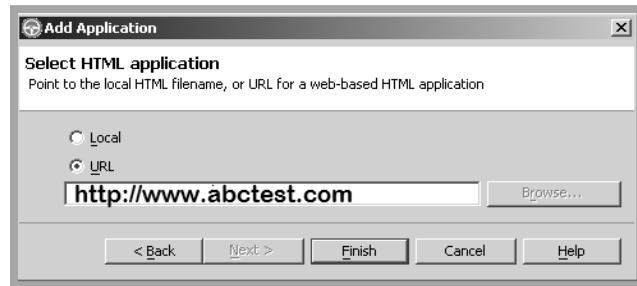
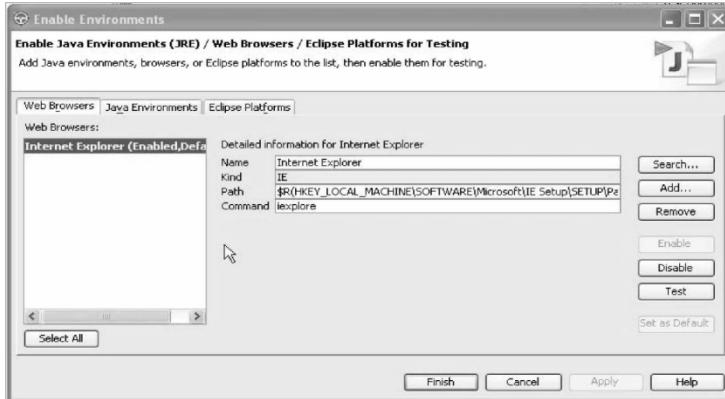


Set up the path of the test application:



If the test application is a web browser, this has to be configured in Configure>Enable Environment:

## Software Automation Testing Secrets Revealed



Step2: Test Object Inspector – Click on Run in menu and select Test Object Inspector:

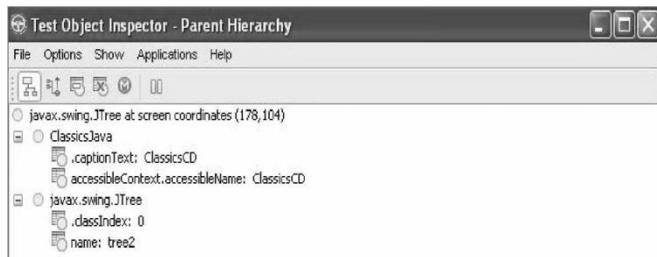


Test objects are viewed using this feature and information such as parent hierarchy, inheritance hierarchy, test object properties, non-value properties and method that can be used for those objects. Windows objects are also captured from Test Object Inspector!

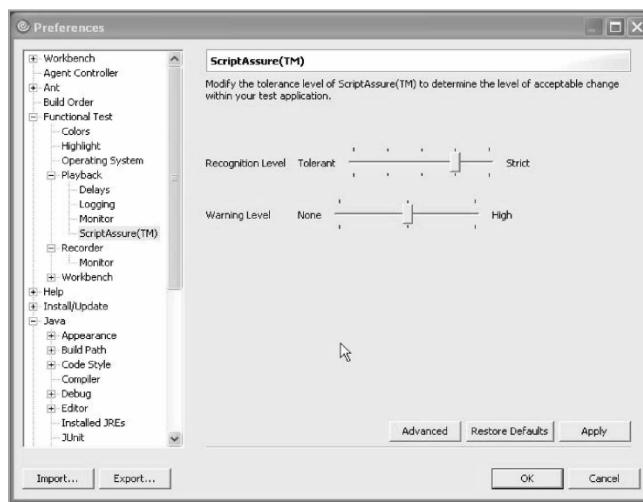
### Best practice:

Evaluate the tool by testing the sample application and analyzing the test object-capturing method in order to perform Proof of Concept for test automation readiness.

Parent hierarchy details are explored through Test Object Inspector:



Step 3: Preferences: Set the Recognition Level and Warning Level of objects on the basis of the observation from PoC (Proof of Concept):



#### Step 4; Record the script:

During the recording of test script, the script commands appeared in object.action() format for example object.select("item") or object.click(). All commands are based on GUI (graphical user interface) objects, and Test Object Map plays key role as a repository to all the test objects.

The commands used mostly are listed below for reference:  
edit box, list box, check button, radio button, push button, tab, grid, scroll bar, window, menu, toolbar, edit box

Click on Add Script using Recorder and record the script:



#### Types of common RFT commands

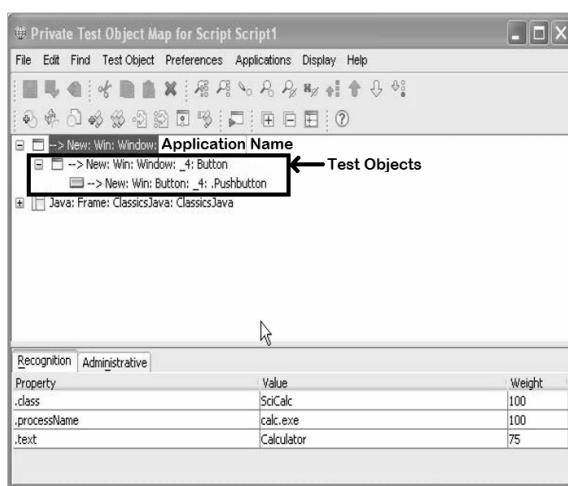
Action commands: These are commands that trigger the events such as click, edit, open for example object.click().

Get commands: These are commands returning the information or the property of the particular element such as object.getProperty("PropertyName").

Verification point-based commands: These are commands that verify object attributes such as objectVP.performtest().

Wait for existence-based commands: These are commands that wait for certain seconds until the appearance of the object and then move on to the next command line such as object.waitforexistance("Test Object").

**Step 5 – Identify how the objects are mapped for each test script:**  
 Once the test script is recorded, recognized mappings of logical to physical names of classes and objects in the application are stored in following format:



## Sample test script

```
/*
 * Description : Test Automation Sample Script
 */
import resources.order1Helper;
import com.rational.test.ft.*;
import com.rational.test.ft.object.interfaces.*;
import com.rational.test.ft.object.interfaces.SAP.*;
import com.rational.test.ft.object.interfaces.siebel.*;
import com.rational.test.ft.script.*;
import com.rational.test.ft.value.*;
import com.rational.test.ft.vp.*;
```

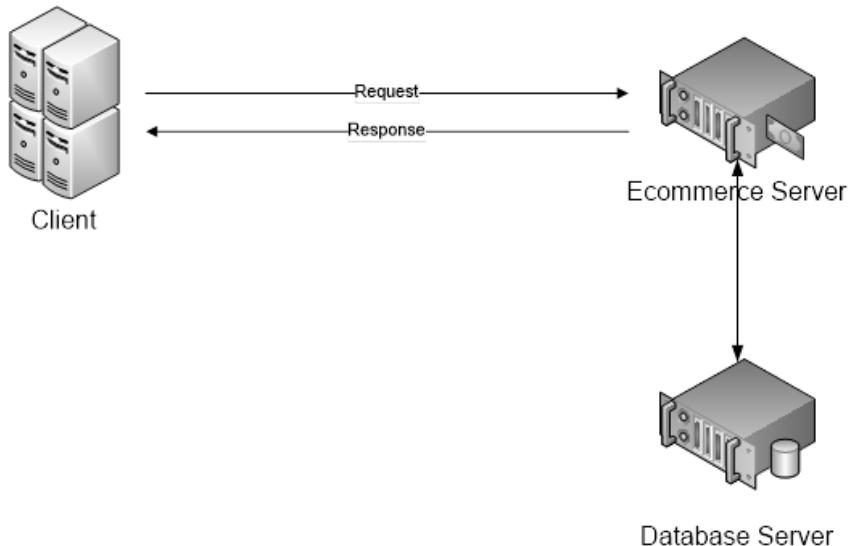
```
public void sampletestMain(Object[] args)
{
// Frame: Order Request System
jmb().click(atPath("Order"));
jmb().click(atPath("Order->Place New Order..."));
// Frame: Member Logon
newCustomer().click();
ok().click();
// Frame: Place an Order
cardNumberIncludeTheSpacesText().click(atPoint(24,11));
placeAnOrder().inputChars("Music iPod Buy");
nameText().click(atPoint(27,13));
placeAnOrder().inputChars("Sell");
phoneText().click(atPoint(17,12));
placeAnOrder().inputChars("Purchase");
placeOrder().click();
}
```

*For further reading and practice, download the projects at  
<https://github.com/narayananpalani/testautomation>*

## SOA Test Automation

Service-oriented architecture (SOA) testing is an evolving testing domain expected to grow rapidly in the next twenty years with multiple new implementations. SOA is an architecture overall for building the business applications based on the requirements. It consists of loosely coupled black box components orchestrated to provide a well-defined/established service by connecting the business processes into one umbrella.

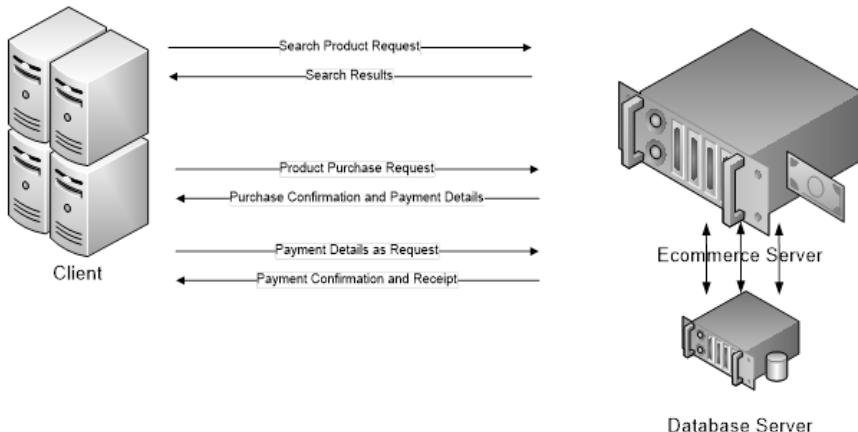
SOA is a collection of services that communicates within, performs data transfer and at the same time coordinates the activities together for the business requirements!



Typical request and response of an eCommerce server-client model is described above. If a customer is buying a product from eCommerce website, the client computer/device or mobile talks to the server of eCommerce system as a request and receives the response in order to process the purchase further.

## Purchasing of a product from eCommerce SOA – an example

eCommerce Product Purchase Requests/Responses



When customers purchase products over the eCommerce portals, they interact with eCommerce servers to get the response, and all the requests and responses are getting stored in database server in order to track the orders. Verifying requests and responses of these eCommerce transactions need a strong expertise over test automation on the requests push over the server and verification over the responses what customers get is typical testing example in SOA model.

### What is called 'service' in SOA?

Any repeatable task in business transaction is considered service. Request for an exchange rate over the foreign-exchange-transfer website which is a request and the ideal response is the exchange rate and fee for the particular amount of transfer.

### What is 'service orientation' in SOA?

While various services are providing similar process results, the better approach in the business is to integrate the related services together. For example, if eCommerce platform is integrated to bank payments and shipment systems, the entire workflow gets integrated in one login and the user can book the service, pay for

the service and use the shipment system till the end of the service which can be called the integration of the complete services.

### **What are the types in SOA testing?**

SOAP UI testing, XML testing and REST Protocol testing are some of the types of SOA testing at the moment.

### **SOAP Messages Automation**

Simple Object Access Protocol (SOAP) is a standard protocol specification for message exchange based on XML. XML messages are used for the communication between the web service and client.

SOAP defines the rules for communication, for example what are the tags that should be used in XML and their meaning. An advantage of SOAP – since HTTP requests are usually allowed through firewalls(will not be filtered by firewall), programs using SOAP to communicate can be sure that they can communicate with programs anywhere. WSDL refactoring allows you to automatically update your tests and simulations to be compliant with new versions of your WSDLs

Automation of SOAP messages is one of the key areas where the next-generation-test automation focusing on!

SOAP message has four parts – Envelope, Header, Body and Fault. Only Envelope and Body are mandatory in order to send the SOAP messages through the URL.

SOAP sender usually sends the message with format and data in respective tags. SOAP receiver receives the SOAP messages and processes it further. Hence, SOAP receiver processes and sends the response back, which will be received and validated.

### **Sample SOAP input message**

```
<soapenv:Envelope  
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:b="http://htng.org/2013A"  
xmlns:ns="http://www.opentravel.org/OTA/2003/05"  
xmlns:wsu="http://docs.oasis-
```

```
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
<soapenv:Header>
<To
    xmlns="http://schemas.xmlsoap.org/ws/2004/08/addressing">http
://localhost:63682/Service1.asmx</To>
<Action
    xmlns="http://schemas.xmlsoap.org/ws/2004/08/addressing">http
://localhost:63682/Service1.asmx?op=UpdateRoomStatus</Actio
n>
<ReplyTo
    xmlns="http://schemas.xmlsoap.org/ws/2004/08/addressing">
<Address>http://localhost:63682/Service1.asmx</Address>
</ReplyTo>
<MessageID
    xmlns="http://schemas.xmlsoap.org/ws/2004/08/addressing">uui
d:ceb2798e-76d9-433b-87de-2da0e96557d3</MessageID>
<Security xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
<UsernameTokenxmlns="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
    xmlns:ns15="http://schemas.xmlsoap.org/ws/2006/02/addressingi
dentity" xmlns:ns14="http://docs.oasis-open.org/ws-sx/ws-
secureconversation/200512"
    xmlns:ns13="http://www.w3.org/2003/05/soap-envelope"
    wsu:Id="uuid_f794ba10-e1a0-40ae-b184-909efbd1e2e9">
<Username
    xmlns="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">IGRUser</Username>
<Password
    xmlns="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
    Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">*****</Password>
</UsernameToken>
```

```
</Security>
</soapenv:Header>
<soapenv:Body>
<b:HTNG_HotelRoomStatusUpdateNotifRQ
EchoToken="Echo22222"TimeStamp="2014-08-17T09:30:47Z"
Version="1.0">
<b:POS>
<b:Source>
<b:RequestorID Type="10" ID_Context="SUPERHOTEL_PMS"
ID="PMS1">
<b:CompanyName>Superhotel</b:CompanyName>
</b:RequestorID>
</b:Source>
</b:POS>
<b:PropertyInfo ChainCode="HotelChain"
BrandCode="HotelBrand" HotelCode="HotelHotel"/>
<b:Room RoomID="EE12345">
<b:Devices>
<b:Device ID="2341ae3c" FriendlyName="MB-12345"
Description="Mini Bar" Class="InRoomRefreshmentCenter">
<b:PriorHealthStatus Value="ERROR" Reason="UNKNOWN"/>
<b:CurrentHealthStatus Value="OPERATIONAL"
Reason="POWER_CYCLED">Manual device reset
performed</b:CurrentHealthStatus>
</b:Device>
<b:Device ID="ef3278ea" FriendlyName="OS-12345"
Description="Ceiling Sensor"
Class="OccupancyDetection">
<b:PriorHealthStatus Value="ERROR"
Reason="LOW_BATTERY"/>
<b:CurrentHealthStatus Value="OPERATIONAL"
Reason="BATTERY_REPLACEMENT">Device
online.</b:CurrentHealthStatus>
</b:Device>
</b:Devices>
</b:Room>
```

```
</b:HTNG_HotelRoomStatusUpdateNotifRQ>
</soapenv:Body>
</soapenv:Envelope>
```

The above message is a sample SOAP input request message for room-booking service that is sent to respective SOAP receiver for the processing of the message. The next message is the output received after the process. The below response message has to be validated in order to ensure that the request and response are matching as expected.

## **SOAP automation using Excel VBA**

Programming language used in this section: VB Script

SOAP messages can be sent using Excel VBA (refer the section of Excel VBA automation to understand the Visual Basic Editor).

Update SERVER address with valid address for the URL it is getting used below.

Option Explicit

```
'Set Reference to Microsoft XML, v6.0
Sub login()
    Dim responseText As String
    Dim sURL As String
    Dim sEnv As String
    Dim xmlhttp As New MSXML2.XMLHTTP
    Dim xmlDoc As New DOMDocument
    Dim startPos, endPos, openTag, closeTag, startTagPos, sid As
String
    sURL = "[URL="http://<SERVER
ADDRESS>:8080/axis/services/USD_R11_WebService?wsdl"]ht
tp://< SERVER ADDRESS
>:8080/axis/services/USD_R11_WebService?wsdl[/URL]"
    sEnv = "<?xml version=""1.0"" encoding=""utf-8""?>"
    sEnv = sEnv & "<soap:Envelope
```

```

xmlns:xsi=""[URL="http://www.w3.org/2001/XMLSchema-
instance"]http://www.w3.org/2001/XMLSchema-
instance[/URL]"""
xmlns:xsd=""[URL="http://www.w3.org/2001/XMLSchema"]htt-
p://www.w3.org/2001/XMLSchema[/URL]"""
xmlns:soap=""[URL="http://schemas.xmlsoap.org/soap/envelope-
"]http://schemas.xmlsoap.org/soap/envelope[/URL]"">
    sEnv = sEnv & "<soap:Body>""
    sEnv = sEnv & "<login>
xmlns=""[URL="http://www.ca.com/UnicenterServicePlus/Servi-
ceDesk"]http://www.ca.com/UnicenterServicePlus/ServiceDesk[/
URL]"">
    sEnv = sEnv & "<username></username>""
    sEnv = sEnv & "<password></password>""
    sEnv = sEnv & "</login>""
    sEnv = sEnv & "</soap:Body>""
    sEnv = sEnv & "</soap:Envelope>""
With xmlhttp
    .Open "post", sURL, False
    .setRequestHeader "Host", "webservices.gama-system.com"
    .setRequestHeader "Content-Type", "text/xml; charset=utf-
8"
    .setRequestHeader "soapAction",
"[URL="http://www.ca.com/UnicenterServicePlus/ServiceDesk"]
http://www.ca.com/UnicenterServicePlus/ServiceDesk[/URL]"
    .setRequestHeader "Accept-encoding", "zip"
    .send sEnv
    xmlDoc.LoadXML .responseText
    responseText = .responseText
End With
openTag = "<loginReturn"
closeTag = "</loginReturn>"
startPos = InStr(1, responseText, openTag) + 10
endPos = InStr(1, responseText, closeTag)
startTagPos = InStr(startPos, responseText, ">") + 1
' Parse xml for returned value

```

```
sid = Mid(responseText, startTagPos, endPos - startTagPos)
'Call Next Web Service
getUser
End Sub
```

## Sample test project on SOAP Excel VBA automation

### Step1:

Capture the sample SOAP request:

Request:

```
POST
http://www.w3schools.com/webservices/tempconvert.asmx
HTTP/1.1
Host: www.w3schools.com
Content-Type: text/xml; charset=utf-8
Content-Length: 388
SOAPAction:
"http://www.w3schools.com/webservices/CelsiusToFahrenheit"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<CelsiusToFahrenheit
xmlns="http://www.w3schools.com/webservices/">
<Celsius>25</Celsius>
</CelsiusToFahrenheit>
</soap:Body>
</soap:Envelope>
```

### Response:

```
HTTP/1.1 200 OK
Cache-Control: private, max-age=0,public
```

Content-Type: text/xml; charset=utf-8  
Date: Thu, 04 Jun 2015 20:59:50 GMT  
Server: Microsoft-IIS/7.5  
X-AspNet-Version: 4.0.30319  
X-Powered-By: ASP.NET  
Content-Length: 408

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope  
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body  
><CelsiusToFahrenheitResponse  
xmlns="http://www.w3schools.com/webservices/"><CelsiusToFahr  
heitResult>77</CelsiusToFahrenheitResult></CelsiusToFahr  
heitResponse></soap:Body></soap:Envelope>
```

### Step2:

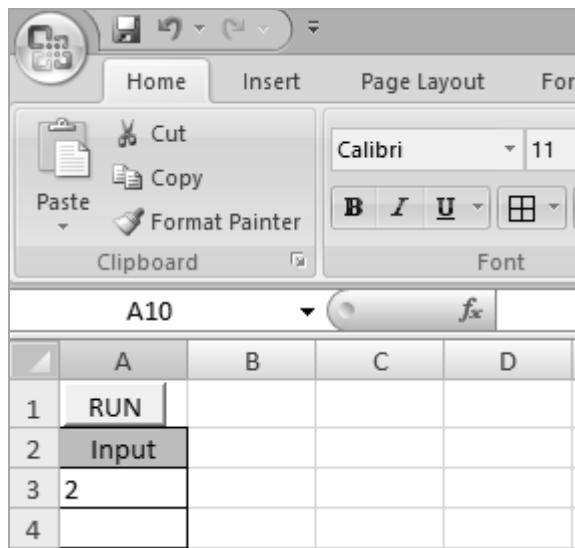
Analyze what needs to be parameterized?

Answer: Inputs can be parameterized. In the request above:

<Celsius>25</Celsius>

### Step3:

Create a new excel file, insert a button and update with Input column and input value:



For further reading and practice, download the projects at  
<https://github.com/narayananpalani/testautomation>

# Selenium Testing Best Practices

HERE ARE 5 WAYS TO IMPROVE YOUR AUTOMATED TEST CODE



## 1.) NEVER USE (IMPLICIT)WAIT OR SLEEP

High probability of delay when multi tests run using selenium with such wait or sleep code

## 2.) ALWAYS TRY FLUENT WAIT OR EXPLICIT WAIT IN ORDER TO LOAD THE OBJECT

Using fluent wait is always a best practice to check for object using throttle time



## 3.) CUSTOMIZE LOCATOR

Do some research and always try your own object properties created for the project for the purpose of automation tests rather than id or other locators

## 4.) AVOID COMMON LOCATOR

Avoid using common name or id as locator which runs a test successfully by checking first possible object found in javascript framework or fails in java framework when multiple object with same properties identified



## 5.) BE INFORMATIVE AND APPROACHABLE

Have detailed comments which includes date of code edit, owner of the piece of code, further reference and version of selenium used while developing the piece of code- details for every piece of function developed in complex team environment. Always mention whether the piece of code is compatible to particular browser driver

## **Data Warehouse Testing (DWH)**

DWH is one of the highly important and growing sectors of software testing industry. The expertise required in DWH/ETL (extract-transform-load) has been very critical for world leading testing projects.

### **Key advantages of any ETL tools**

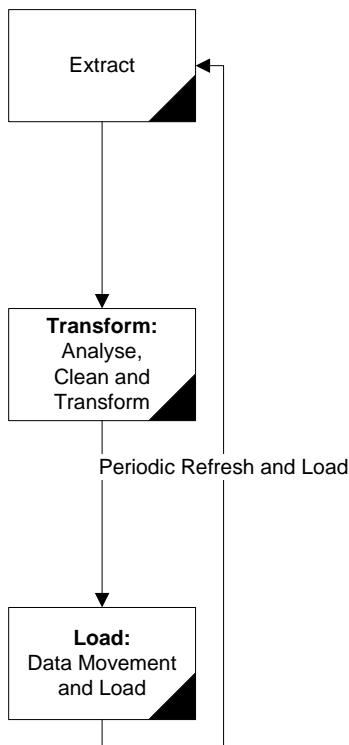
As per Ralph Kimball (author of The Data WarehouseETLToolkit), the three important areas to determine best ETL tools are as follows:

Timeliness of the tool: How quick the tool performs the ETL process!

Data volume: How much data it is capable of extract, transform and load to process the steps!

Response time: How quick it is to respond back with the results what is required when requested!

## Datawarehousing process flow



### Why data transformation is required?

Every business is growing with multiple business transactions which enable to have multiple databases to store the records. Each database has different codification and format requirements based on the business need.

When two businesses merge, it creates a disparity over the data representations automatically. So merger and acquisitions provide a gateway to handle ETL process in a structured and more advanced method

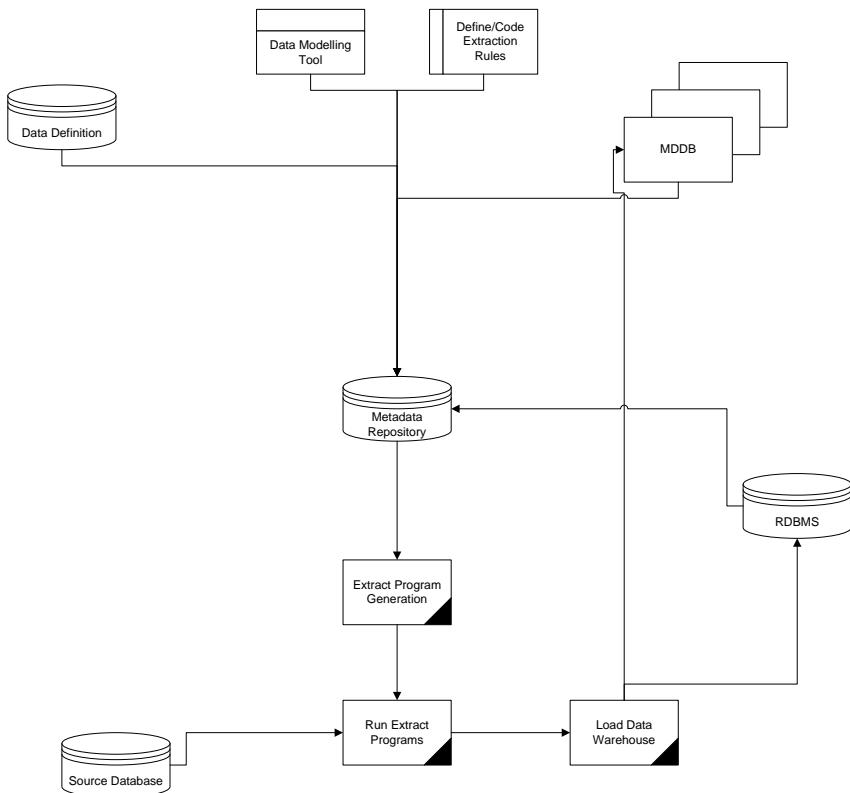
## Why the data has to be transformed?

Operational data of business is supposed to be converted into a consistent and business-oriented format that leads to the need of transformation. It means that the transformation computes the derived information, and summarization is carried out to aggregate and precompute the summaries provided.

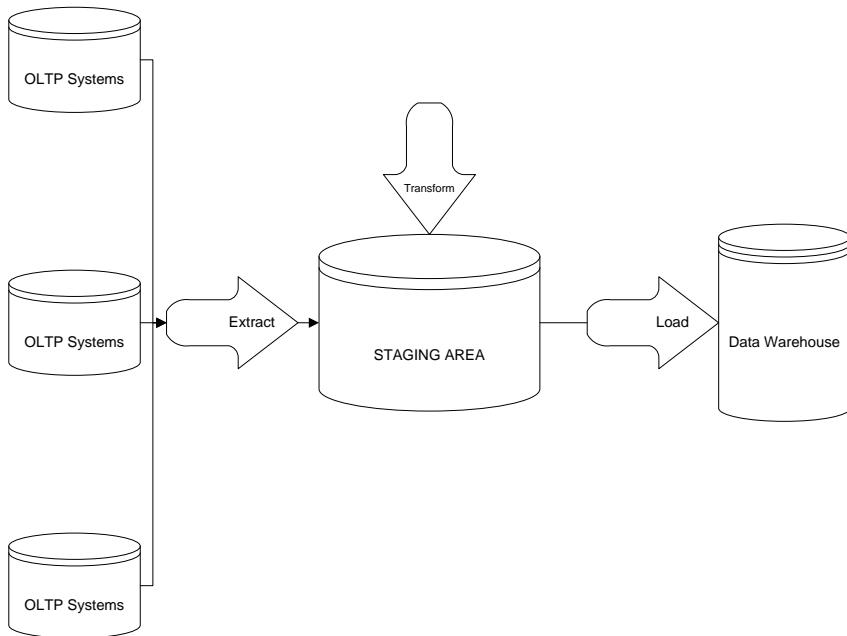
When the importance of load is to be known?

If multiple data formats are used in different database tables, it is discernible that the load process is tedious for maintaining the large database systems.

## The ETL process of DWH testing



## High-level ETL processing logic in OLTP systems to DWH flow



### How organizations choose among ETL products?

When big organizations change over a period of time, they adapt either generic solutions that are already built in to serve the market needs or they adapt a custom solution in which specifications are made on the basis of the business needs. In custom solutions, RDBMS staging tables and stored procedures are used and programming languages such as C,C++,Perl and Visual Basic are the most required skills to perform customization. Building code generators are the key tasks of those ETL activities. Either of the ways the testing of data transformation is compulsory in order to confirm that the ETL process is working as expected.

## **Why generic solutions are required by organizations?**

Auto-generated scripts are major advantage of generic solutions, and the limitation due to manual coding across project has been reduced when the solutions are implemented. Especially, if the business value has to be quantified in terms of how beneficial it is to implement any tool or solution is something depends on how beneficiary the tool is. With the help of Generic Solutions, functionality, reliability and viability are no longer considered the major or critical issues by organizations!

## **ETL tools**

Huge number of transformation rules with a GUI (graphical user interface) can be specified by ETL tools. These ETL tools can generate programs to transform data across the database. Multiple data sources can be controlled, and data redundancy can be addressed to the greater extent!

Metadata can be generated as output from ETL tools, and most of the tools run multi-threaded-environment-based low-cost servers to benefit the ease of tool usage!

## **ETL tool classification**

### **Generation 1: Code-generation-oriented products**

These are some of the famous first-generation tools: Prism, Passport, ETI-Extract, Copy Manager, SAS/Warehouse Administrator.

Source code is generated through ETL tools in first generation, and most of the projects customized the source code to executable code to utilize in projects. GUI interfaces are used to define the process and server or host runs the extract, transform and load process. Most of the tools generate the source code automatically for extract program, and it can be compiled, scheduled and run in batch mode. Intermediate files are also used, and most of the programs were single threaded in first generation with the capability of metadata generation to limited amount of the complete project.

Although the tools were mature and programmers were capable of using their Cobol, C and C++ knowledge to generate code, the training to those ETL tools were complex and expensive.

Especially, extract-related programs have to be compiled from source, and transformation-based code has to be customized manually for single execution as parallel execution was not supported in those days! So most of the metadata were manually generated with limited amount of autogeneration in place!

### **Generation 2: Engine-directed projects**

These are some of the famous first-generation tools: Data Mart Solution, DataStage, PowerCenter/Mart

Executable code can be generated when configuration and customization are set as per the project description, and this executable code can directly generate metadata as outputs with the aid of Generation 2 tools! All extraction, transformation and load runs can be performed in server machines, and data can be directly processed on server from source.

### **Structured Query Language (SQL) for Software Testing**

Functional testers are expected to possess good knowledge on SQL to get into testing projects as a basic requirement. This aims to ensure that the project database has been verified on the basis of the tests performed. In terms of DB verification, basic queries of SQL are highly recommended as a knowledge base for any entry-level testers.

For automation testers, advanced SQL-level understanding is highly recommended since the code has to interact with DB to extract the relevant test results and match them with front end results.

### **What is SQL?**

SQL is a structured query language for storing, manipulating and retrieving data that stored in relational database. The names of some relational databases are MySQL, MS Access, Oracle, Sybase and Informix.

## Enhance automation framework with DB verification capability

Let us take an example of implementing SQL-based tests in Selenium. Download the required jar file (mysql-connector-java-3.1.13-bin.jar) from the link below:

<http://www.java2s.com/Code/Jar/m/Downloadmysqlconnectorjava3114bingjar.htm>

Once downloaded, update to the Reference Libraries as mentioned in the earlier chapters on the automation framework development.

//As part of the class file, update the required imports as follows:

```
import org.testng.After;
import org.testng.Before;
import org.testng.Test;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
```

//This class file is explained in terms of adapting the DB verification within the selenium script

```
public class DBTesing {
    // Database URL has been assigned with DB_URL as a constant
    public static String DB_URL="jdbc:mysql://localhost:3306/user";
    // Connection object which is assigned to null initially.
    static Connection con = null;
    // Statement object is assigned as stmt
    private static Statement stmt;
    // Constant for DB Username
    public static String DATABASE_USER = "xxxxxx";
    // Constant for Database Password
    public static String DATABASE_PASSWORD = "xxxxxx";
    @Before
    public void setUp() throws Exception {
        try{
            // Database connection is getting established in this line
            String dbClass = "com.mysql.jdbc.Driver";
```

```
Class.forName(dbClass).newInstance();
// DB Connection is getting established here:
Connection con = DriverManager.getConnection(DB_URL,
DATABASE_USER, DATABASE_PASSWORD);
// Statement object to send the SQL statement to the Database
stmt = con.createStatement();
}
catch (Exception e)
{
e.printStackTrace();
}
}
@Test
public void test() {
try{
//Insert the SQL query within the Selenium Script here:
String query = "select * from table t where empid=101";
// Get the contents of t table from DB
ResultSet res = stmt.executeQuery(query);
// Print the result untill all the records are printed
// res.next() returns true if there is any next record else returns false
while (res.next())
{
System.out.print(res.getString(1));
System.out.print("\t" + res.getString(2));
System.out.print("\t" + res.getString(3));
System.out.println("\t" + res.getString(4));
}
}
catch(Exception e)
{
e.printStackTrace();
}
}
}
@Test
public void afterTest() {
```

```
public void tearDown() throws Exception {  
    // Close Database connection  
    if (con != null) {  
        con.close();  
    }  
}
```

## Object Recognition

*In earlier chapters, the topics are covered on how the tools are used along with sample document format for test plan, test cases and other test team document methods. In this chapter, an important topic has been discussed on how automation tools capturing the human usability using object properties recognition!*

When user manually operates the functions on the web pages or the applications, the same steps are automated using the object-capturing method of automation tools. This is a process of pattern recognition algorithm using feature-based or the property-based techniques by every automation tool.

### HP UFT (formerly HP QTP) – test object model

Objects of the applications are captured and used from object repository (.tsr files) in the ‘test object model’.

Two types of supported repository in QTP are: local and shared object repository. During the recognition of the objects (using Object Spy tool within QTP or descriptive programming manually updates the script with properties), QTP tries to use a ‘human’-like technology for identification. During test execution, the tool compares the stored object repository with actual properties available on the application. If the Run Time-Objects matches to the stored objects, it performs the actions and events instructed by the script, and it consequently completes the test execution.

### Ranorex – ready to use GUI objects

Objects of the applications are captured and utilized from Ranorex object repository to manage GUI/application-based objects and represent the file-based mapping information. Ranorex Spy is used for the identification of the object properties (like Xpath of the properties). The advantage of the tool is on ‘ready to use GUI objects’ as each Ranorex repository automatically generates a source code file in DotNet file in which all the elements of Ranorex adapters are located. Thus, the automation scripts are simplified by reusing these auto-generated scripts based on the objects.

## Selenium IDE – locators

Selenium IDE is an add-on to Firefox at the moment, and the following are the most common object locators within Selenium IDE: ID, ClassName, Name, TagName, LinkText, PartialLinkText, Xpath, CSS Selector, DOM. After capturing the scripts through record mode, clicking on Target listbox drop down, it provides the option of picking selective/right object properties for the test. Please refer the Selenium IDE section to fully comprehend with the help of the programs illustrated in examples.

## Selenium Webdriver – object properties

The advantage of Selenium Webdriver is to customize the script by creating the object repository in a separate properties file and list all the xpath properties of the GUI over the properties file location. The locators who frequently used the Industry best practice on Selenium Webdriver are namely ID, ClassName, Name, TagName, LinkText, PartialLinkText, Xpath, CSS Selector and DOM. Kindly refer Selenium Webdriver section to understand how to call the properties file within the automation script (class file).

## Object Recognition Technology – Automation Testing

*In earlier chapters, the topics are covered on how the tools are used along with sample document format for test plan, test cases and other test team document methods. The previous chapter talks about object recognition; in this chapter, the key challenges of automation testing in objection recognition are discussed.*

XPath of the objects is completely trusted to build the automation frameworks across the organizations. Especially if the automation framework is made up of Selenium or UFT (formerly QTP) or Ranorex or any other automation tool, the core logic of capturing the object is through XPath is considered to be the most common way of automation frameworks.

If the frameworks depend on XPath or the object properties, the need for changes over a period of time results in automation framework failures. This is one of the main reasons why most of the automation frameworks show Return on Investment only in long term! When the project exists for more than five years, it is

obvious that the objects travel across many changes. So the automation framework has to be maintained as per the changes made to the objects across the test projects.

### **Object property reference tables in an exclusive database**

To avoid rewriting the automation framework scripts and editing the xpath property details frequently, it is useful to list the entire properties and the object details in database that can be controlled by developers and testers for their testing purposes! For example, developers use xpath properties for their unit testing as part of BDD (behavior-driven development) and on the other hand testers use the properties for regression testing. Hence, the need of the common xpath referencing table is of right approach-so that the changes to the objects are getting notified by both the stakeholders by implementing this approach!

If the existing code in Selenium written in this model:

```
driver.findElement(By.name("j_namefield")).click();
```

It has to be upgraded to the below format to adapt this solution:

```
driver.findElement(By.name (select object_property from table  
Ref where test_name= 'Regression')).click();
```

When the database is connected to the framework in initial steps, the ongoing scripts can receive the respective line of data and extract the required object details from DB to perform the test automation. Whenever the object properties are getting changed, this can be updated to the database to keep the references up-to-date.

### **File upload method:**

Automation framework can display message to upload the object properties csv file or excel file in order to perform the automation execution for the latest object properties uploaded. In this way, any latest changes can be updated to the csv file by developers or testers and at the same time it can be used during the test execution.

## Software Automation Testing Secrets Revealed

THE ULTIMATE  
NEWBIE'S GUIDE  
TO SELENIUM  
TESTING

How to build your tests from scratch

CHOOSE A SELENIUM VERSION

To start a test, you have to find the best selenium version to suit your needs. Weigh the pros and cons of each before you settle on one. Selenium 2.53 and 3.53 are stable versions recently...

FIND COMPATIBLE BROWSER DRIVERS

Selenium versions until or before 3.0 do not need any additional driver for Firefox browser. If you choose to upgrade from earlier versions to 2+, choose 2.53 which runs Firefox tests using Marionette Driver

PICK A GECKO DRIVER

Choose a gecko driver to run Firefox tests if using Selenium 3+ versions. Selenium 3.53 is latest stable version. But it is advised to upgrade to 2.53 selenium with marionette driver first and then migrate to 3.53 selenium with gecko driver next! Firefox 38 and lower, use Selenium 2.46.0 Firefox < 47, use Selenium 2.53.1 Firefox >= 47, use Selenium 3.0.1

PICK AN IE DRIVER

Use IEDriver 2.42 for IE7 and IE6.  
Use the x64 IEDriver if you want to take screenshots (Screenshots truncated issue)  
Use the x32 IEDriver for your tests (Slow text-entry issue)

PICK A CHROME DRIVER

Chrome > 55, use chromedriver 2.25  
Chrome > 49, use chromedriver 2.22  
Chrome > 43, use chromedriver 2.20  
older versions, use chromedriver 2.10

## Why to Learn Appium?

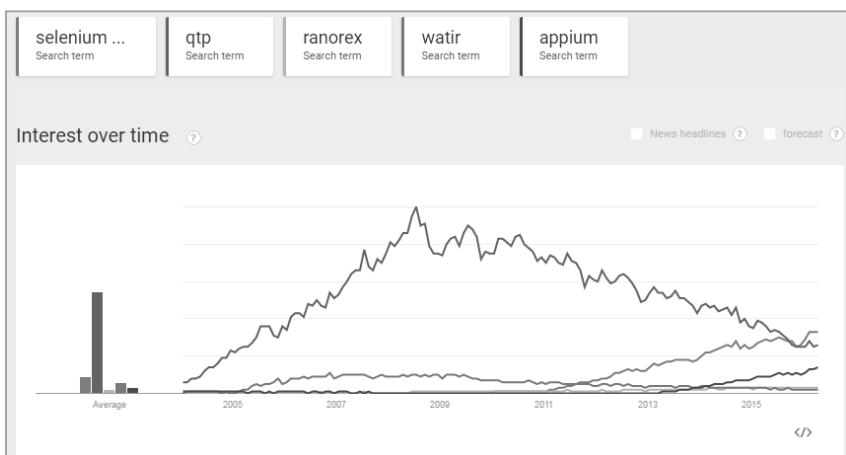
*Programming language used in this section: Java Programming*

*More code examples of this section are available from github at:*

*<https://github.com/narayananpalani/testautomation/tree/master/019AppiumTest>*

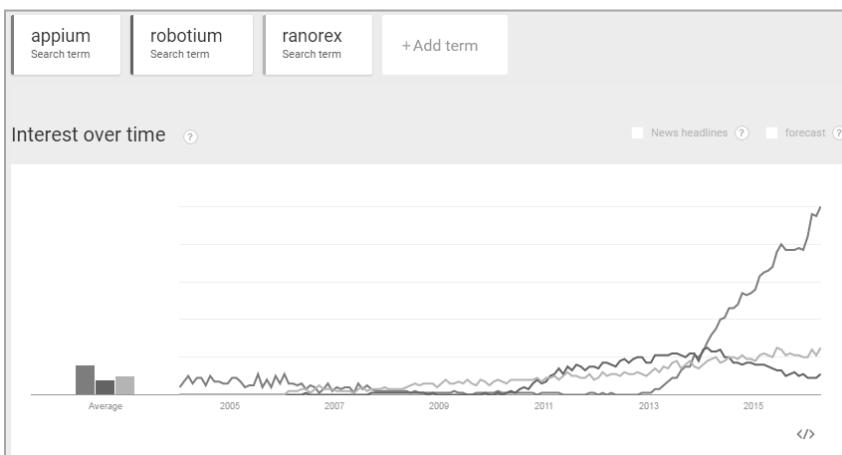
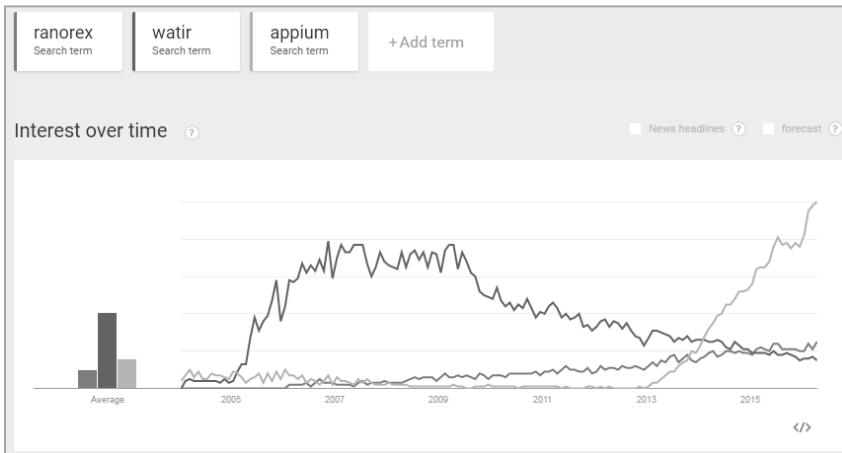
Software testing has been emerging as an exclusive software specialization, and it has been growing like never before! Test automation branched toward open-source tools and Selenium grown as one of the leading tools in test automation world. But how to compare the tools and observe which tool has a great future? Learning particular tool will give a niche job opportunity, but how to know which tool is the right tool to learn? Let us understand how people are searching particular automation tool across the globe!

### Test automation tool analysis:

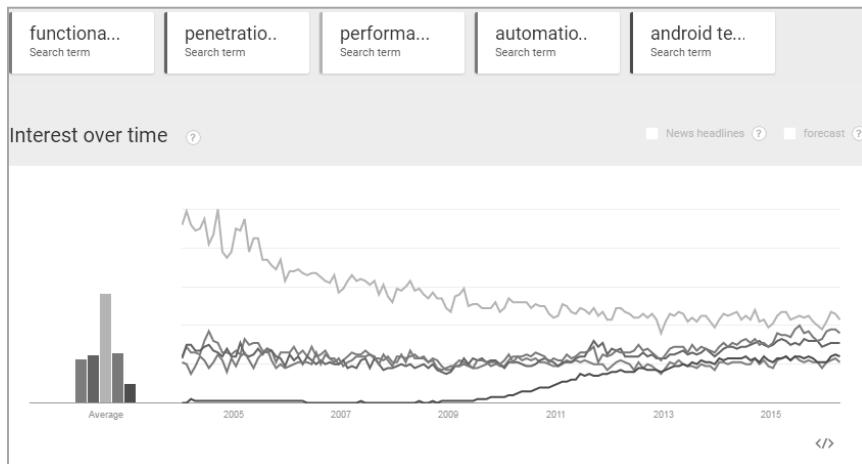


Selenium has been growing since 2011 and overtaken QTP in 2015 due to the hunt for Selenium in the market by a number of people ; hence, Selenium Webdriver is popular and competent enough to get implemented like QTP in test automation projects, whereas Appium has started its growth only since 2013. But it has been growing in a consistent speed.

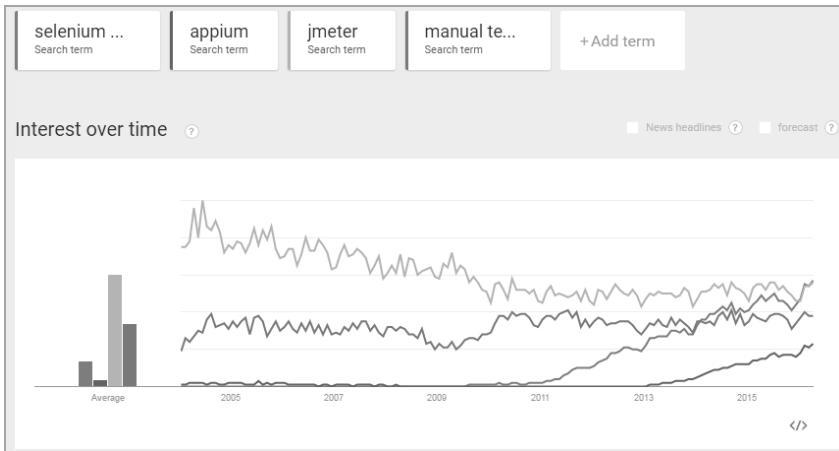
## Software Automation Testing Secrets Revealed



Let us concentrate on the three tools popular to mobile testing world-Ranorex,Watir and Appium. Needless to say as Selenium's new tool Appium is growing in a rapid speed from the year 2014! If we compare Appium, Robotium and Ranorex, there is a clear decline of Robotium over a period from 2014 onwards, and Ranorex is growing at a slower pace but in a steady phase. But Appium grew with a mind-blowing speed from 2014 onwards!



Functional, Penetration, Performance, Automation and Android (Mobile) testings are the separate practices within software testing domain; there is no doubt that performance testing was studied across the years and stabilized to be the most niche domain with popularity. But automation testing has been growing since 2006 in a steady phase! Among the divisions, mobile testing is the recently started area where lot of scope for growth and opportunities are expected in near future as the mobile testing and test automation practices of android devices are growing in a good speed since 2009!



If we carefully look at the Appium comparing to all other tools from each domain (Selenium from test automation, JMeter from Performance testing and Manual testing), it is a new tool and we can easily compare the popularity with the curve of Selenium that has been growing since 2011. Although Appium started to become popular since 2013, the growth level is similar and has a good span of expected growth for the next fifteen years. It may have been improved to different names or different API compatibility over a period but it has a strong growth without any doubt!

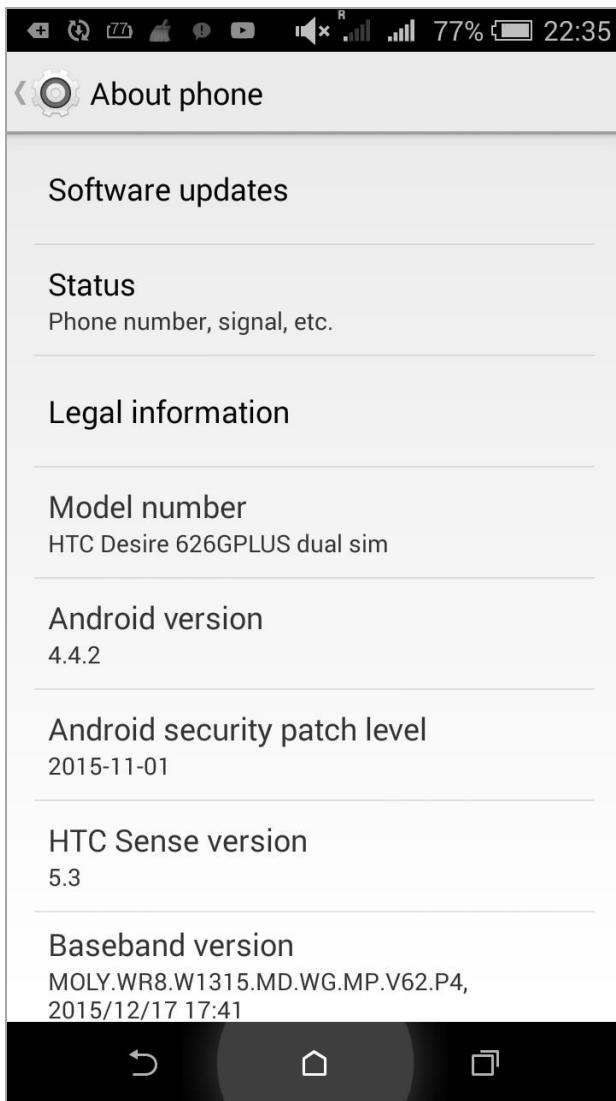
Henceforth learning Appium will lead to get niche job opportunities and lot of exposures to well-established technical projects in mobile domain!

### **Test setup installation and configuration**

#### **Android SDK download and installation**

When mobile devices are being tested from desktop or laptop devices, it is mandatory to download and install Android SDK to access the mobile-related files during the tests. Especially installing Android SDK takes minimum of three hours to maximum of two days when internet connection is slow. It is suggested to pick only relevant files to install during the installation (rather than selecting all android version files), so it is highly recommended to check the box-related current android version of mobile devices being tested. Navigate to 'Settings' and

'About Phone' section to find 'Android version'. Below screenshot is the sample from Android mobile version 4.4.2 (known as Android Kitkat version).



Once the mobile android version has been identified, go to PC or desktop or laptop where the automated scripts are targeted to be designed and searched for Android SDK in Google as below:

## Software Automation Testing Secrets Revealed

A screenshot of a Google search results page. The search query is "download android-sdk". The top result is a link to "Download Android SDK - Xamarin.com" with a small "Ad" label. Below it is another link to "Download Android Studio and SDK Tools | Android Studio" from developer.android.com. The page shows various search filters like All, Videos, News, Books, and More.

Alternatively navigate to below web address and click on 'Download options':

<https://developer.android.com/studio/index.html>

A screenshot of the official Android Studio download page at developer.android.com/studio/index.html. The page features a large image of a laptop displaying the Android Studio interface. On the right side, there is a navigation bar with several links: Features, Latest, Resources, Videos, and a highlighted "Download Options" button. A large black arrow points to the "Download Options" button.

Click on the android sdk zip file available in the section 'Windows' if tests are being written in a window-based desktop:

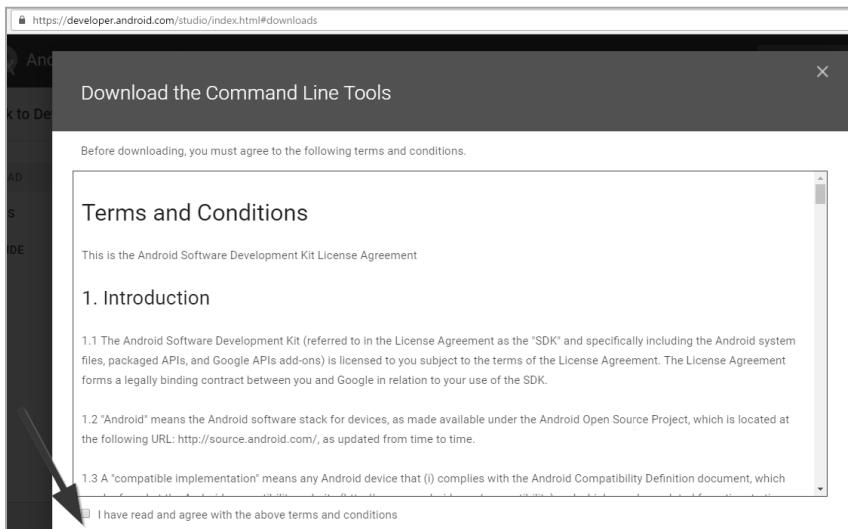
If you do not need Android Studio, you can download the basic Android command line tools.

Platform	SDK tools package	Size
Windows	installer_r24.4.1-windows.exe android-sdk_r24.4.1-windows.zip No installer	144 MB (151659917 bytes)  190 MB (199701062 bytes)
Mac OS X	android-sdk_r24.4.1-macosx.zip	98 MB (102781947 bytes)
Linux	android-sdk_r24.4.1-linux.tgz	311 MB (326412652 bytes)

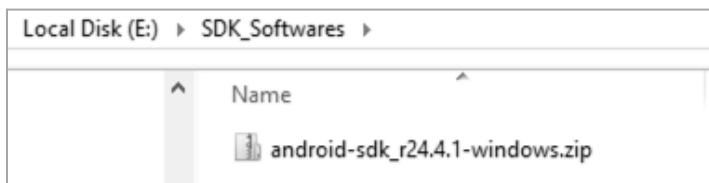
**Note:** Please don't click on Android Studio which is also available as a zip file under another windows section since Android SDK is more than enough, and it is not required to download entire android studio to automate the mobile applications. Only android-sdk\_r zip file is good enough to practice Appium-based tests.

Platform	Android Studio package	Size
Windows	android-studio-bundle-145.3360264-windows.exe Includes Android SDK (recommended) android-studio-ide-145.3360264-windows.exe No Android SDK	1641 MB (1721650280 bytes)  423 MB (444308960 bytes)
Mac OS X	android-studio-ide-145.3360264-mac.dmg No Android SDK, no installer	445 MB (467098338 bytes)
Linux	android-studio-ide-145.3360264-linux.zip	445 MB (466765476 bytes)

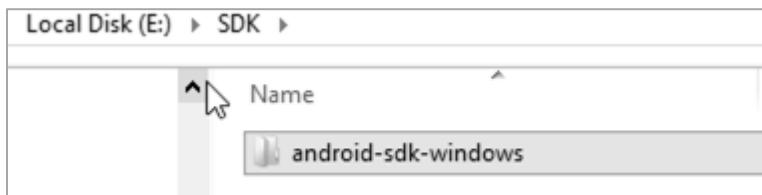
Once Android SDK zip file has been clicked, read the entire terms and conditions and click on OK to proceed further:



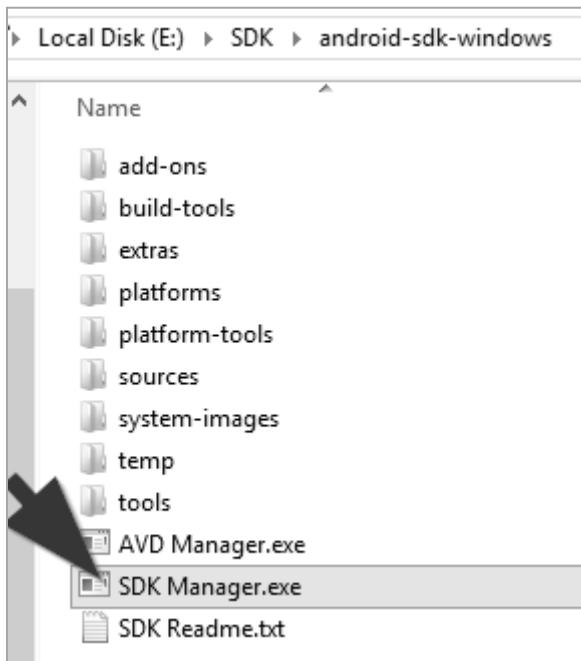
Once zip file has been downloaded to your target computer, have it saved in SDK folder within E drive:



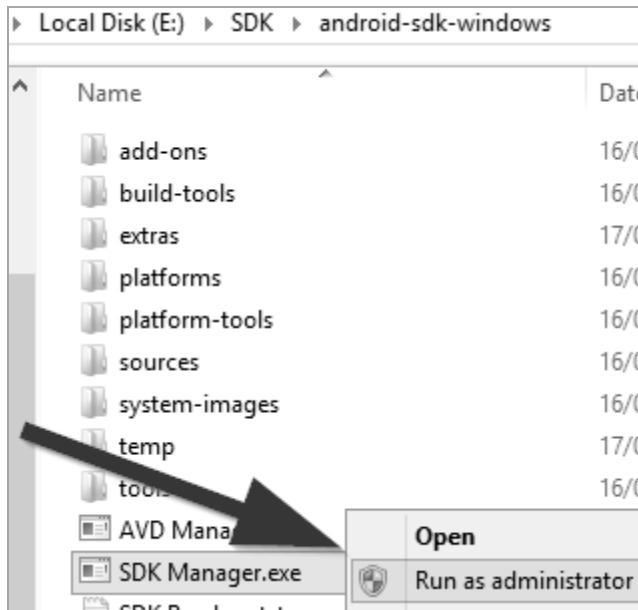
Extract the zip file and save under SDK folder within E: drive (for example). Alternatively, you can save in any drive available within your hard disk:



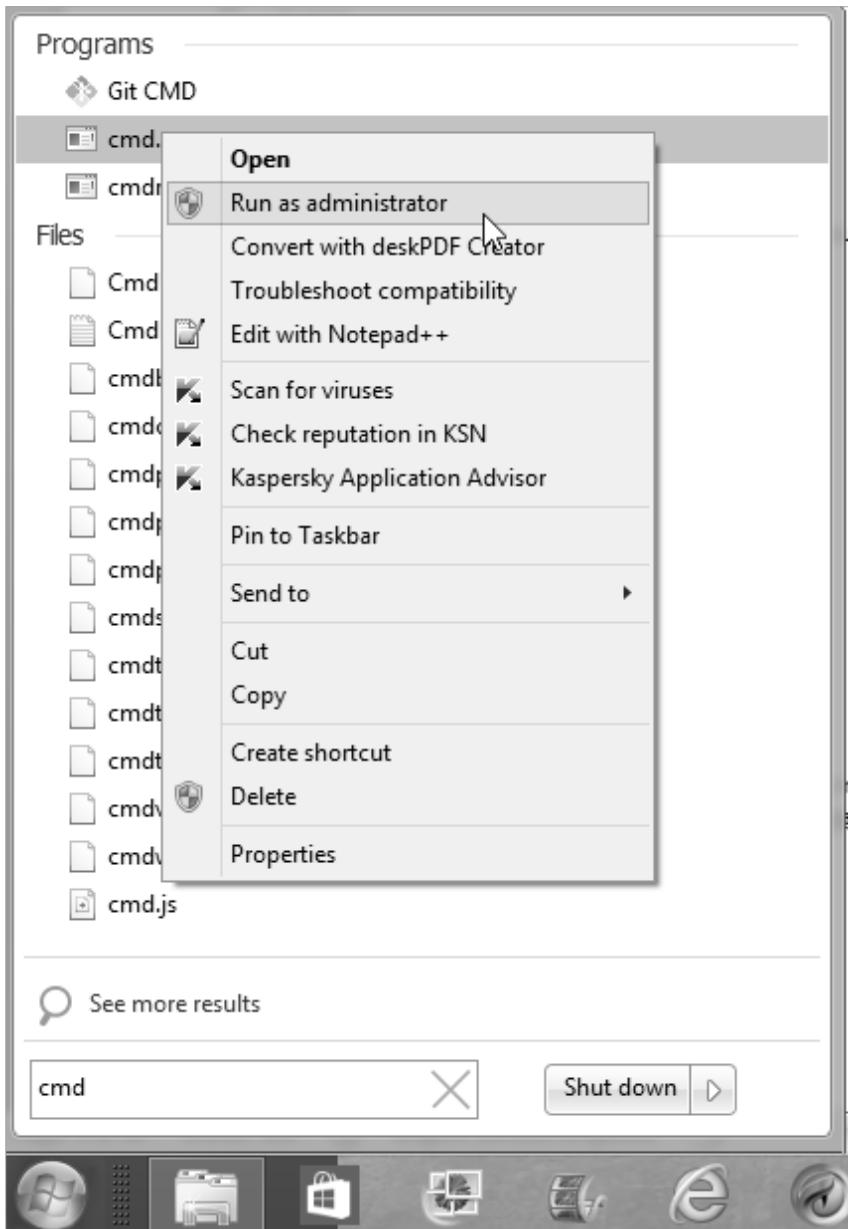
Once the file has been extracted, navigate to SDK Manager exe file and click on it to proceed the installation:



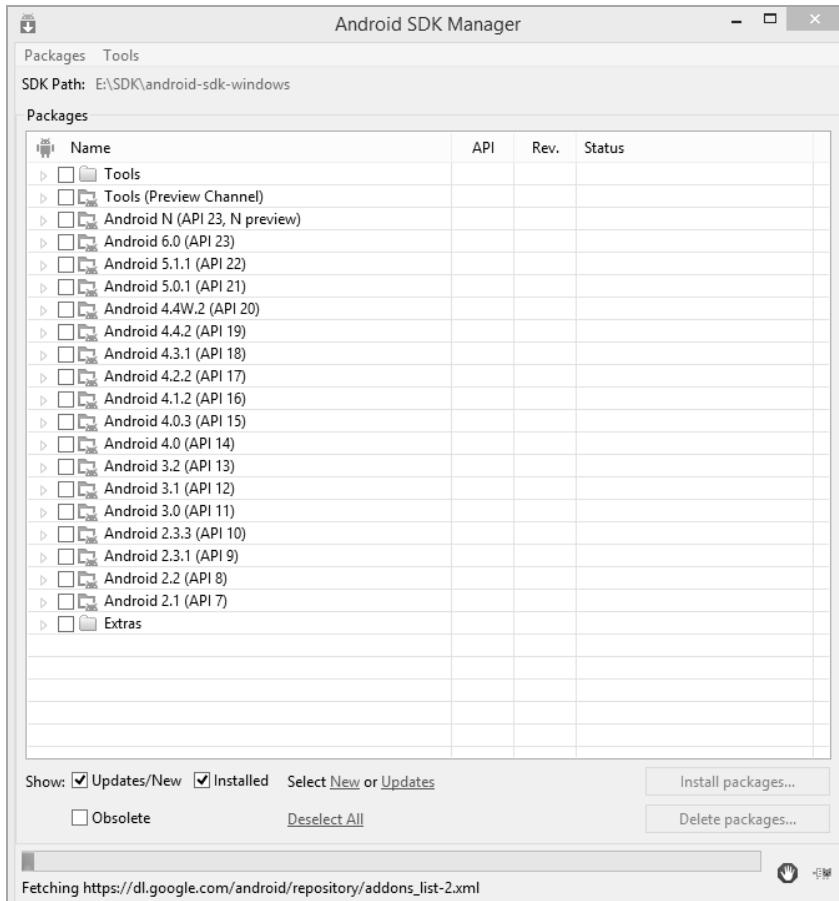
If SDK Manager is not getting opened, there may be user rights issue and please try top launch it using 'Run as Administrator' mode:



**Note:** If 'Run as Administrator' option is not available in your target machine, you may need to contact the administrator to give user permissions to your account. Alternatively, type 'cmd' in Programs Explorer to view command prompt icon and then right click on cmd.exe which provides an option to open command prompt in Administrator mode. Finally navigate to the Android SDK folder to launch SDK Manager exe from command prompt:



Once Android SDK Manager has been launched, please see the list of available android files for installation and the entire list will get upgraded whenever new version of android is being released:



Navigate to Settings in your phone, select About Phone or About Tab which is the last option and view Android version. For example: 4.2.2.

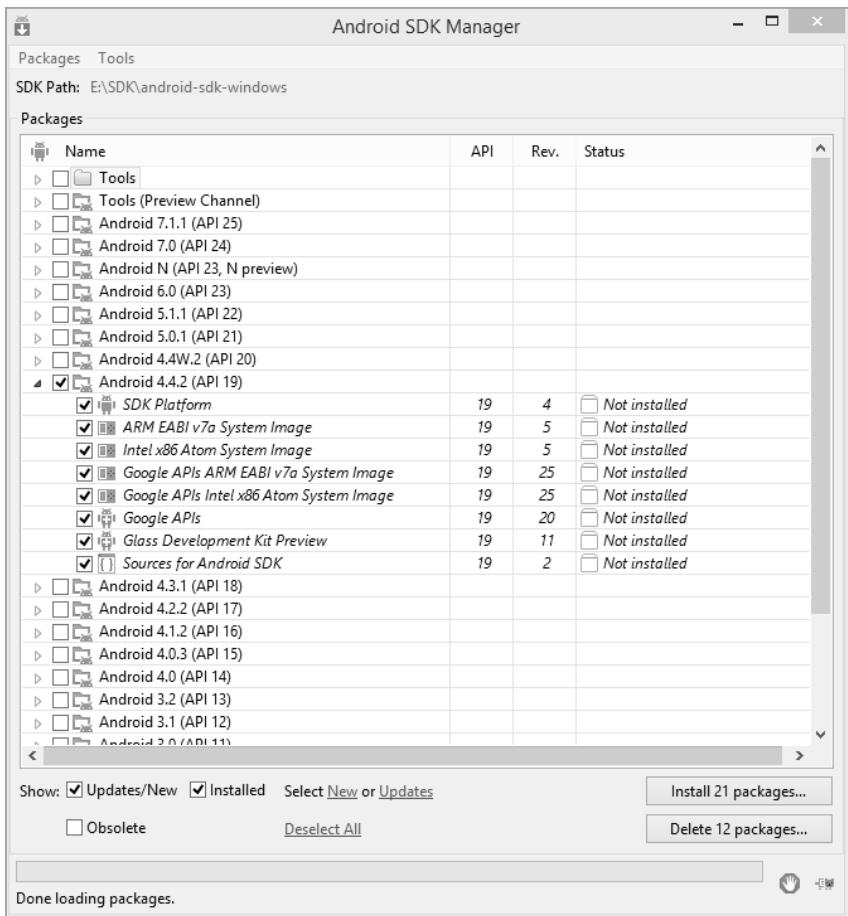
On the basis of the version, select the respective folder in the installation packages. If there is a doubt in Android version and API version, please read Android Version History Wikipedia.

(Ref: Wikipedia(26Nov2016), URL:

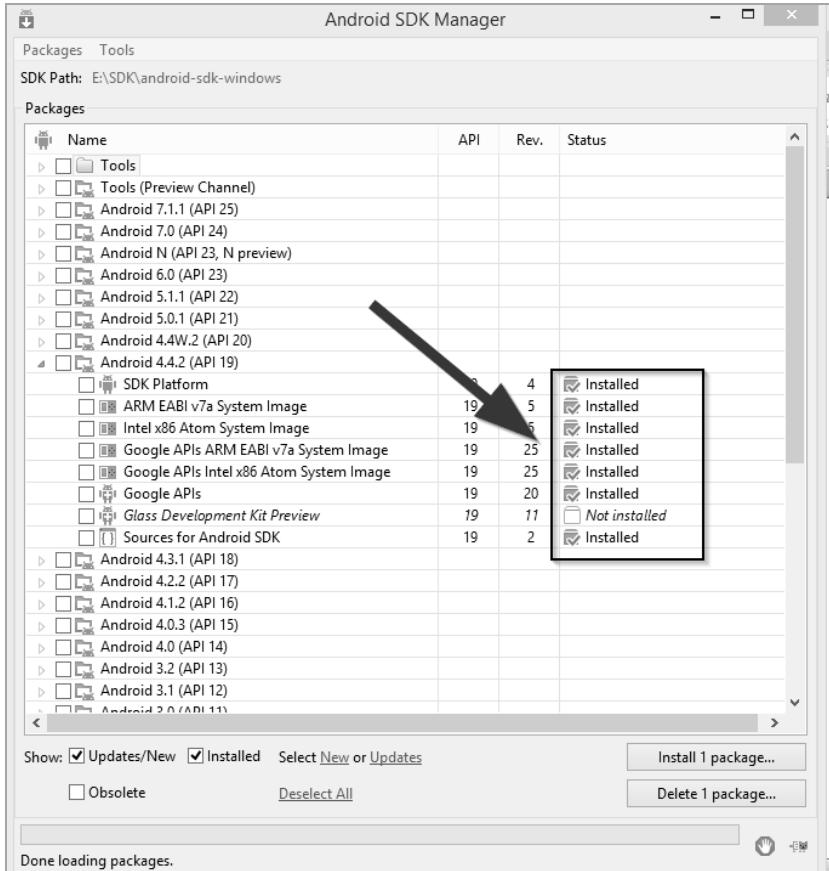
[https://en.wikipedia.org/wiki/Android\\_version\\_history](https://en.wikipedia.org/wiki/Android_version_history))

Note: It is not necessary that you need to have a mobile with android version of 4.4.2 or 4.2.2 only. It is just that the sample mobile is used to perform scripts and execution for the book. Any latest mobile with android or iOS can be used. As long as Android

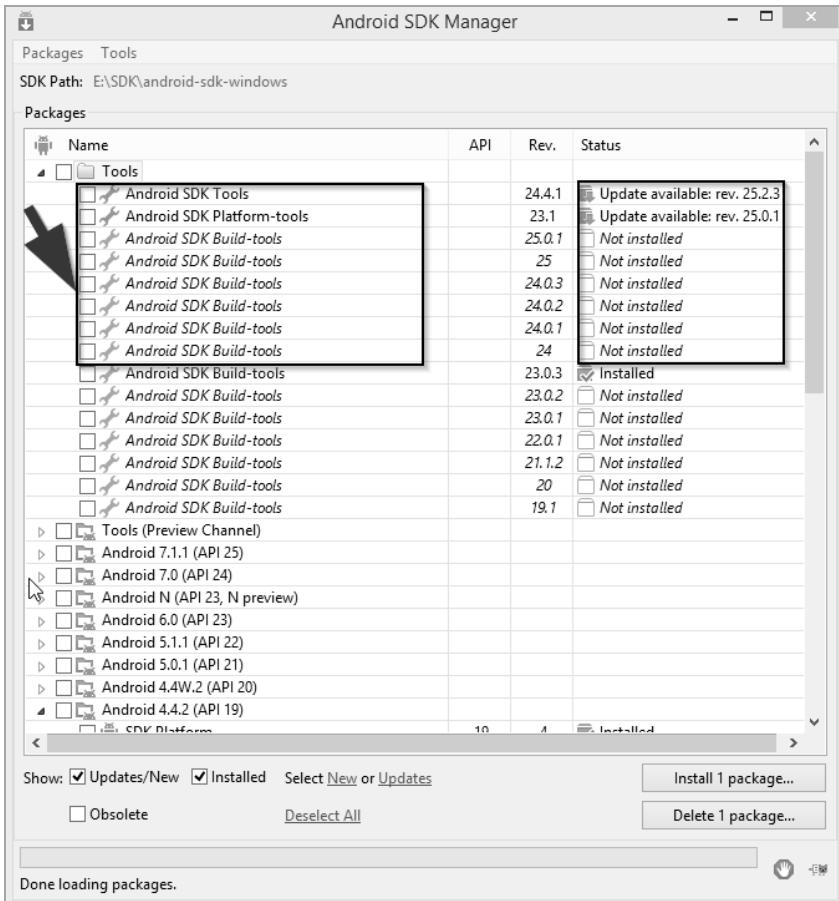
SDK API>=17 installed, it is possible to use Appium to test. When a mobile needs to be tested and it has android version below 4.2.X, make a note that Maven is required to support Selendroid which would be required to run Appium for android version below 4.2.X. So it is always advisable to visit 'System Setup' <sup>1</sup>section in appium.io website.



<sup>1</sup> Appium.io(11Dec2016), URL: <http://appium.io/slate/en/v1.0.0/?ruby#system-setup-android>

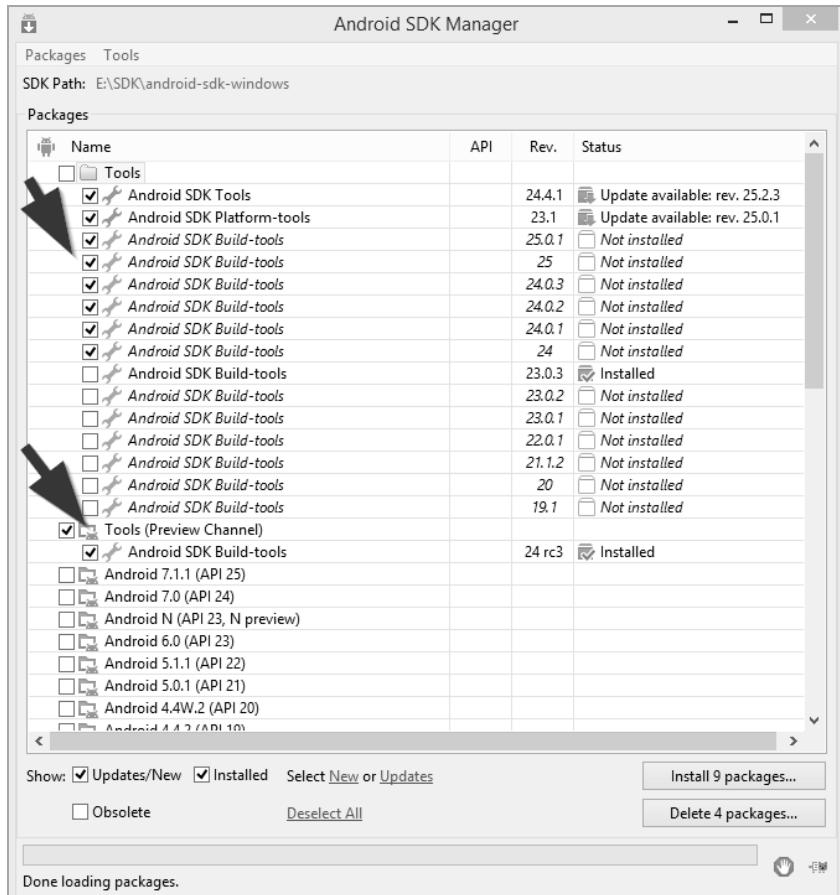


Click on Install \_ package button after selecting all the files from relevant Android folder. For example, if 4.4.2-based android device is used for testing, select all the files within the folder of android 4.4.2 before clicking on Install button. It is to be confirmed that the desktop or laptop is not switched off until installing the complete package and showing tick mark in the status of each file. Once android-version-based files are installed, expand the folder 'Tools' and select relevant build tools. In case of using 4.4.2 mobile, select all those build tools files available with version of your mobile API. If your mobile API is 24 for an android version of 7.0, you need the below (as an example):



Navigate to Tools (Preview Channel) folder and select the files in that folder to install it:

## Software Automation Testing Secrets Revealed



Once installed, following message pop-up appears to close and reopen the manager window:

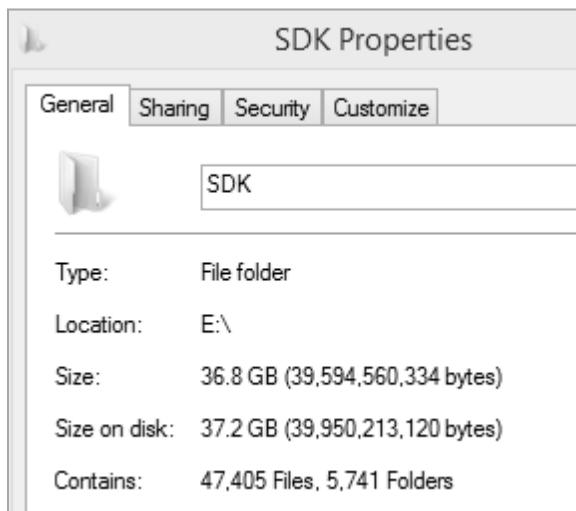


### Error handling during installation:

If there are issues in installation process, go to Tools and Options to enable proxy for corporate networks to retry downloads.

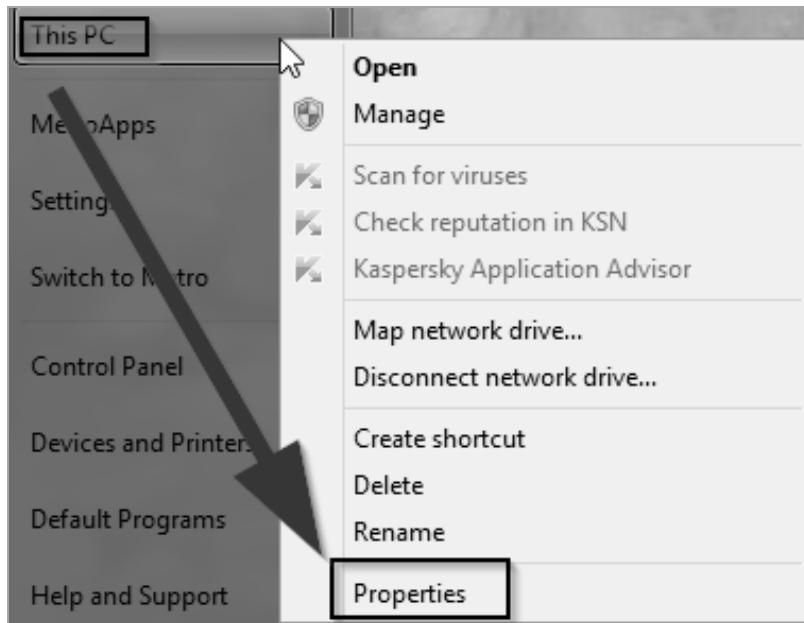
Disk space is another concern when huge files are getting downloaded. Kindly ensure that you have enough space to download.

As an example of downloading android files to install the relevant packages may take at least thirty six GB space in hard disk!

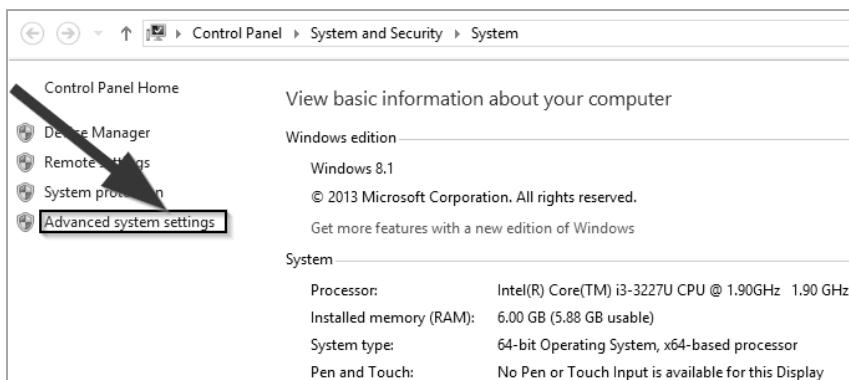


### Android environment variables configuration

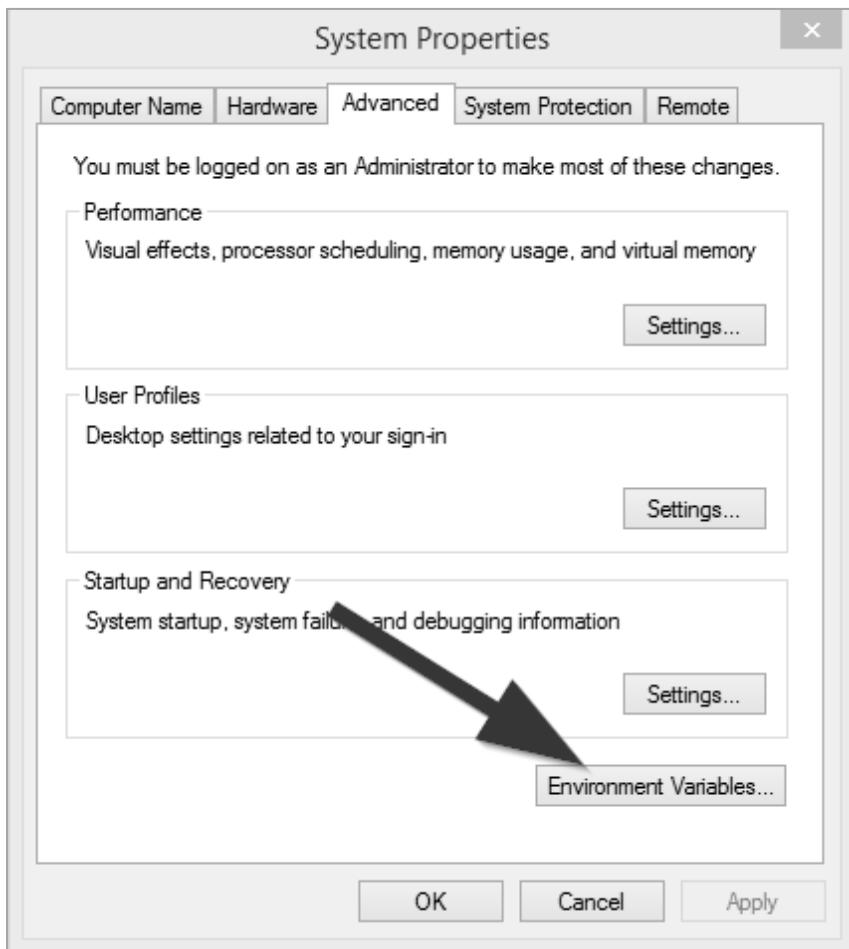
Once android package has been installed, it is mandatory to set the HOME and PATH of android as environment variables to get the scripts recognizing the path of android files. This can be performed by navigating to the environment variables through PC>Properties:



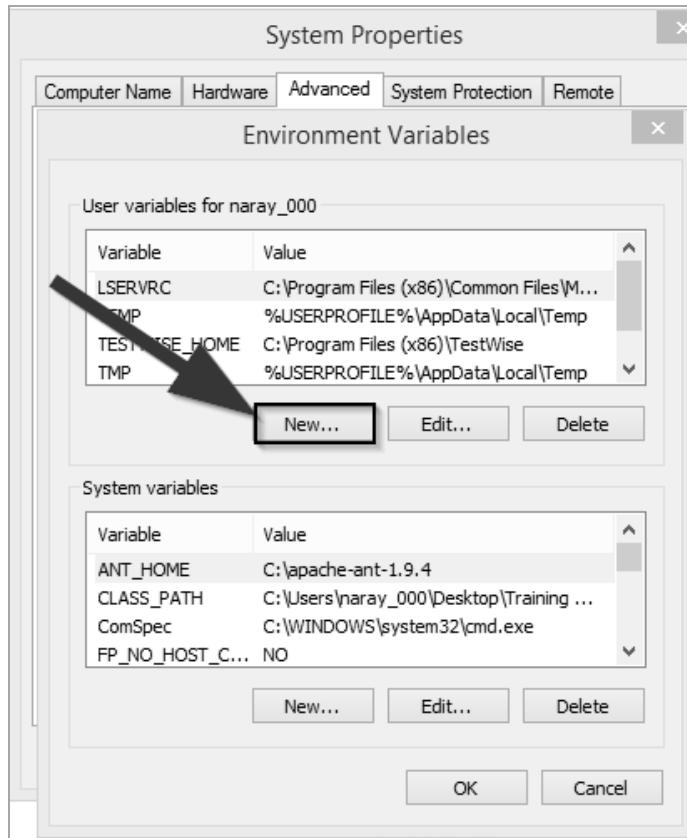
Click on Advanced System Settings:



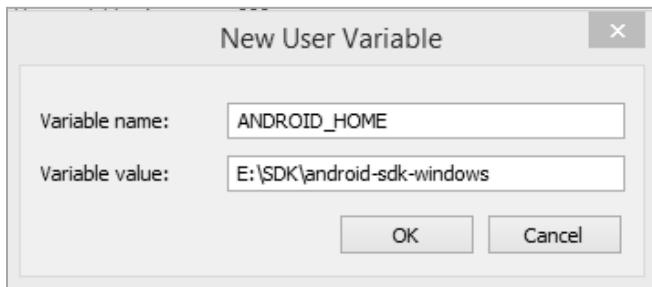
Click on Environment Variables:



Click on Advanced Tab and click New within the frame of User Variables:

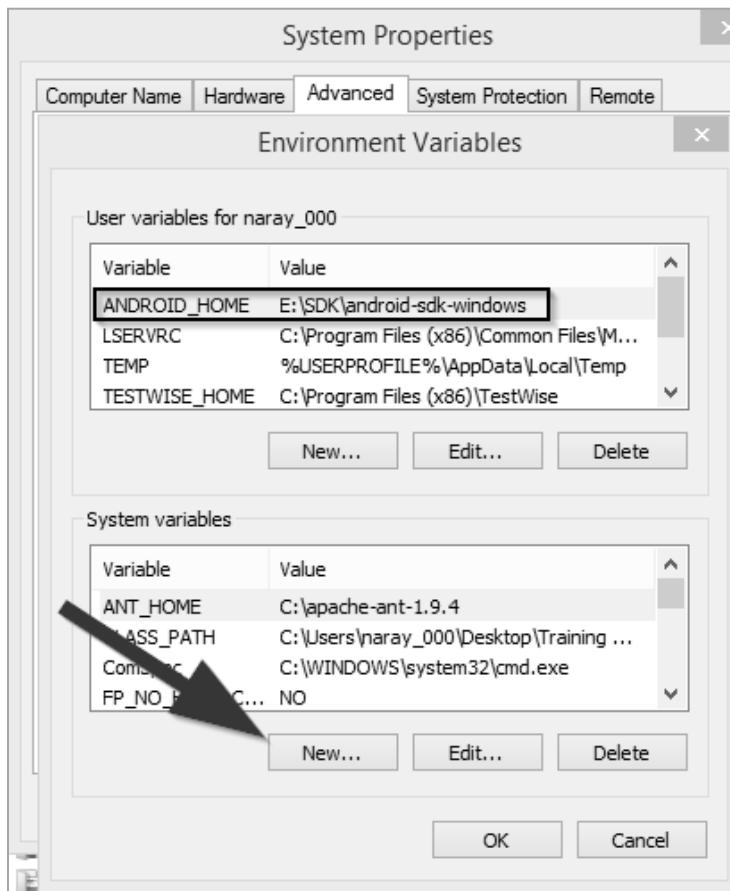


Enter ANDROID\_HOME as Variable Name and Variable Value as the path where android files are located within the system:

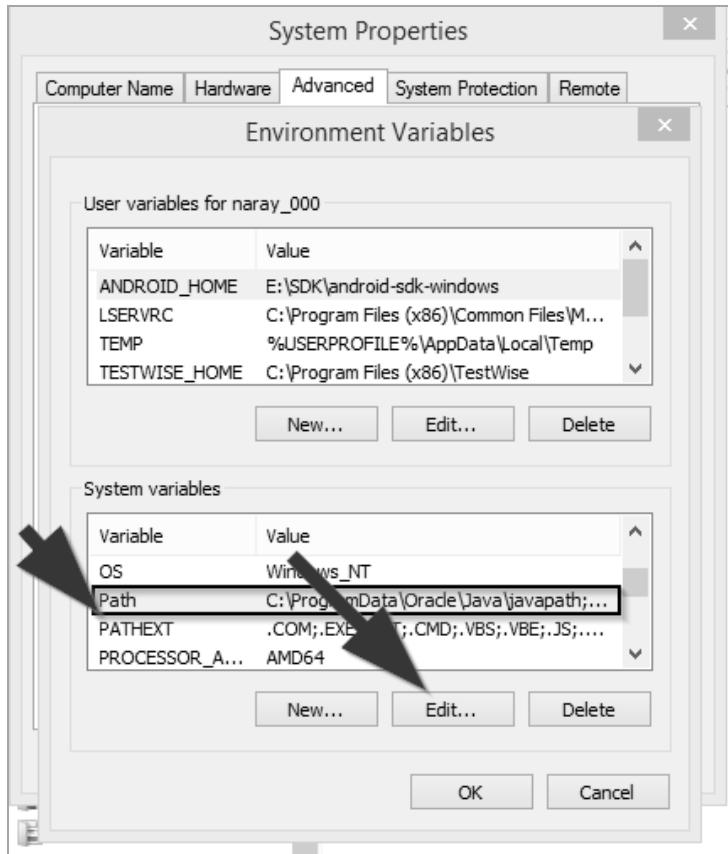


Once ANDROID\_HOME has been set, select 'Path' system variable, click on 'Edit' for Path within the frame of System

Variables. Note: It is not required to click on 'New' within System Variables:



Once 'Edit' has been clicked for 'Path' system variable, navigate to the end of the paths listed and type:



Now take the path of the android build-tools folder which is available within the installation package,paste it within the path and end with semicolon:

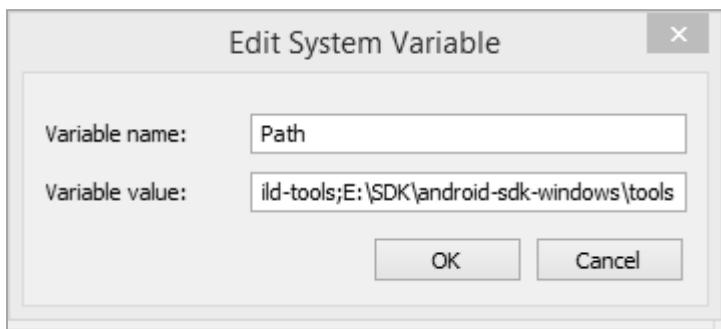


Similarly, copy the path of tools folder that is available within the android installation package and paste it in path of environment variable and end with semicolon:



Once both the paths are updated in PATH system variable, it should look like the below picture:

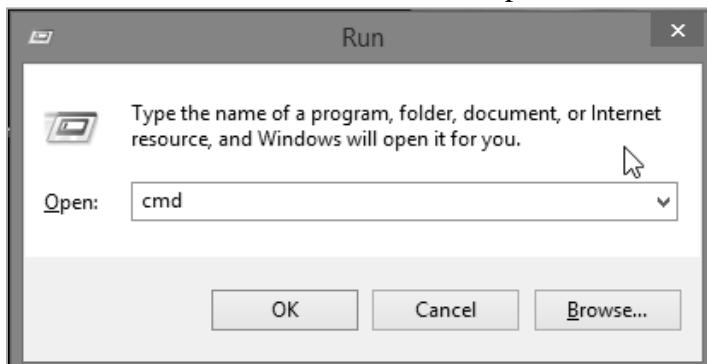
Note: Make sure semicolon separated each path and the remaining paths added earlier within this system variable are not deleted.



How to check if android has been installed properly and the path has been set correct?

Connect the mobile to your computer using relevant cable to recognize the mobile from computer and navigate to command prompt:

Note: Not all the cables help in recognizing the mobile within computer; hence, the best way is to use the branded cable of particular mobile's type. For example, if the mobile is HTC, use cable of HTC to connect from mobile to computer.



Once command prompt has been launched, type 'adb devices' and press enter:



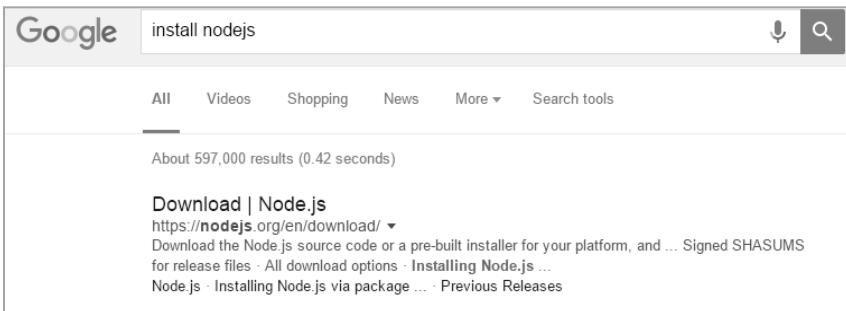
```
C:\WINDOWS\system32\cmd.exe
C:\Users\naray_000>adb devices
List of devices attached
CC591YR06487    offline

C:\Users\naray_000>
```

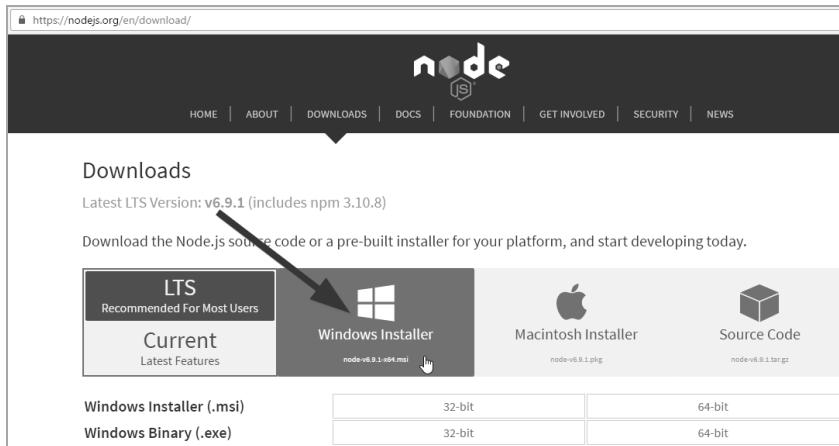
If device has been listed within the command prompt, it tells that the android installation is perfect and mobile device has been recognized. If the mobile has not been listed, ANDROID\_HOME and PATH in environment variable has to be rechecked.

## Install Node JS

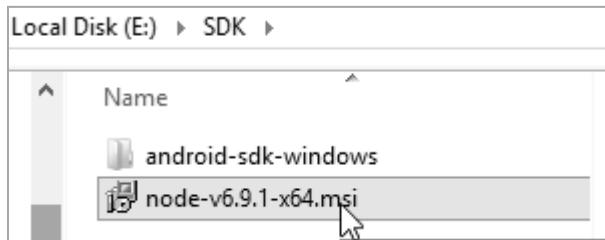
Navigate to Google and type Install Node.js or alternatively navigate to the website nodejs.org/en/download:



Make sure that 'Windows Installer' section has been clicked. If the web page has been redesigned over a period of time, navigate to the section where window-based installers are available if your computer is based on Windows operating system:

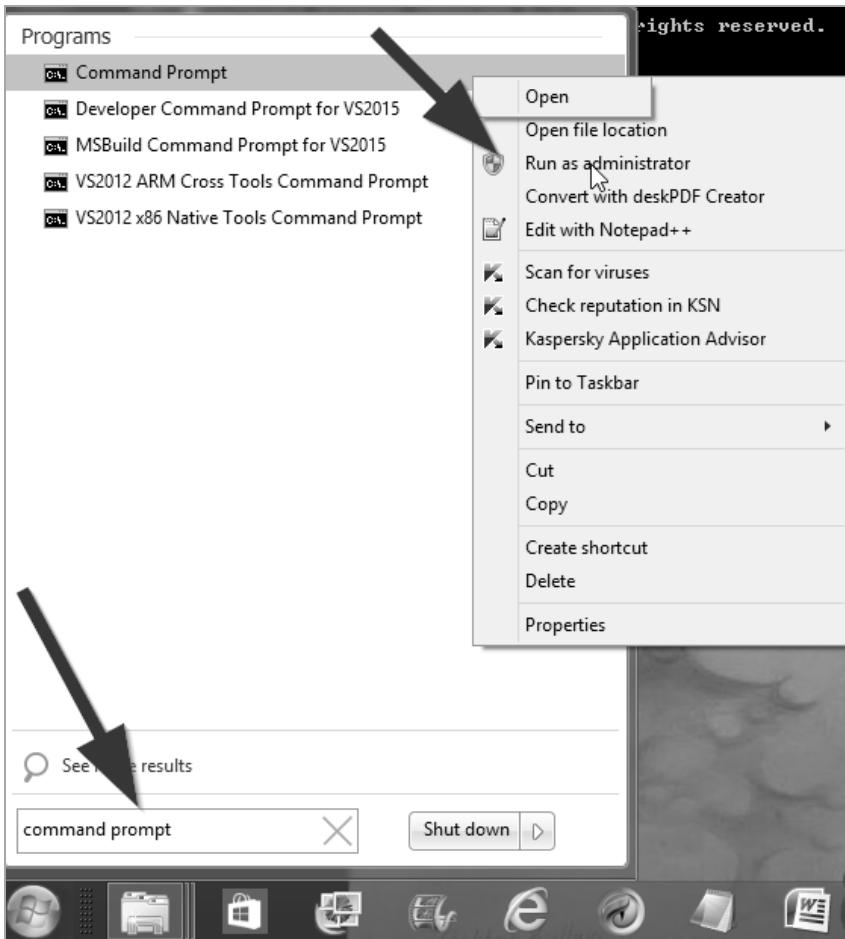


Once downloaded, node js file appears in local machine as the below msi file.



Not every computer user is configured with user permissions to launch msi files. If msi file is not opened after clicking on Open option in right click, select 'Run as administrator' to launch the installer. If such option is unavailable, navigate to command prompt as an administrator:

## Software Automation Testing Secrets Revealed



Navigate to the folder where nodeJS file has been downloaded:

```
Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd\

C:\>e:

E:\>cd E:\SDK

E:\SDK>dir
Volume in drive E has no label.
Volume Serial Number is D008-5FAB

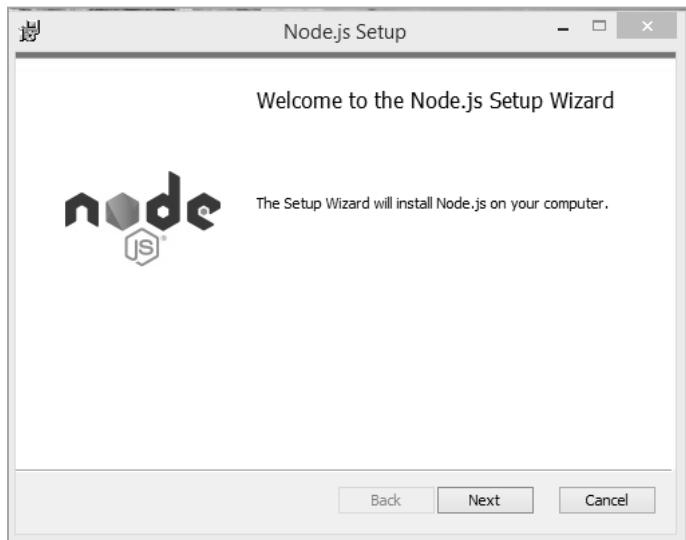
Directory of E:\SDK

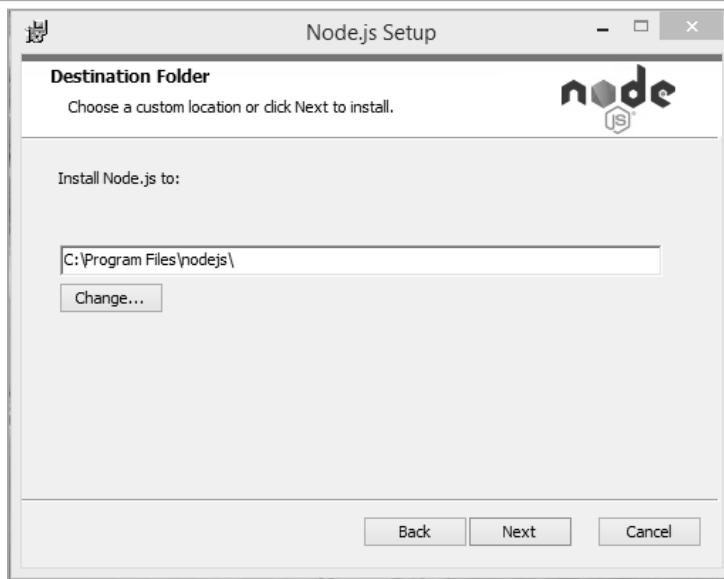
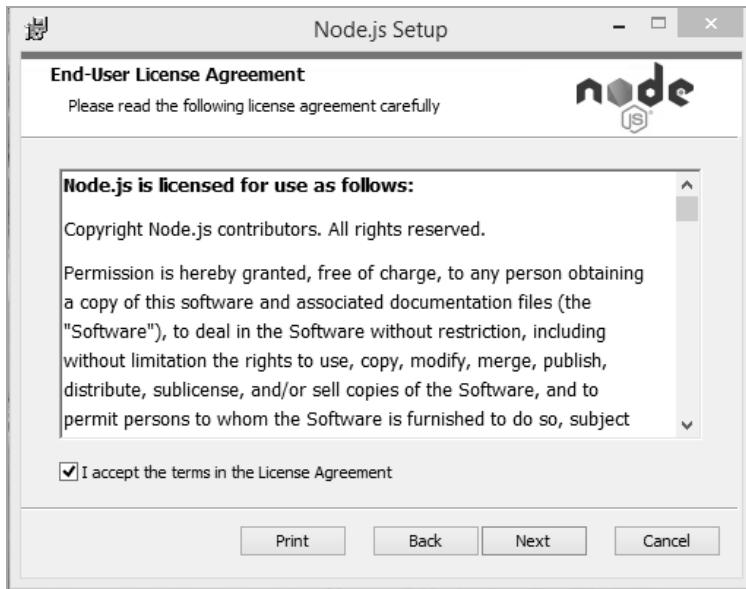
26/11/2016  17:38    <DIR>
26/11/2016  17:38    <DIR>          .
26/11/2016  05:40    <DIR>          ..
26/11/2016  17:00           13,762,560 android-sdk-windows
                           1 File(s)   13,762,560 bytes
                           3 Dir(s)  13,899,354,112 bytes free

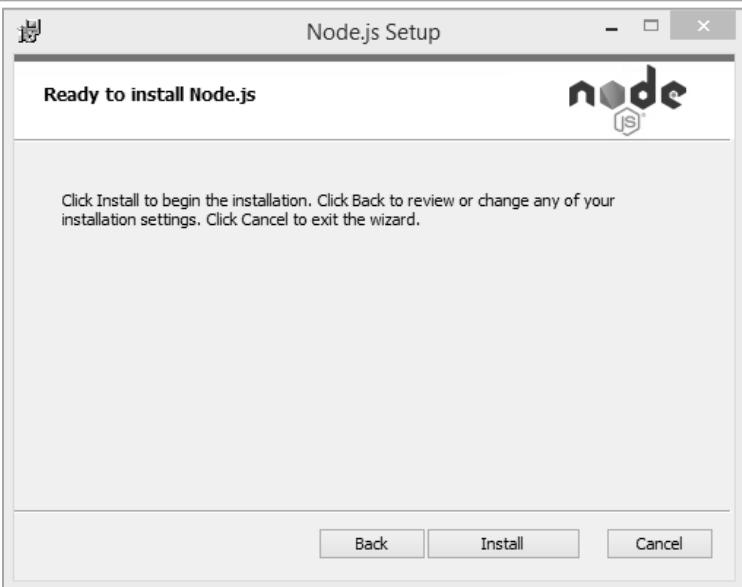
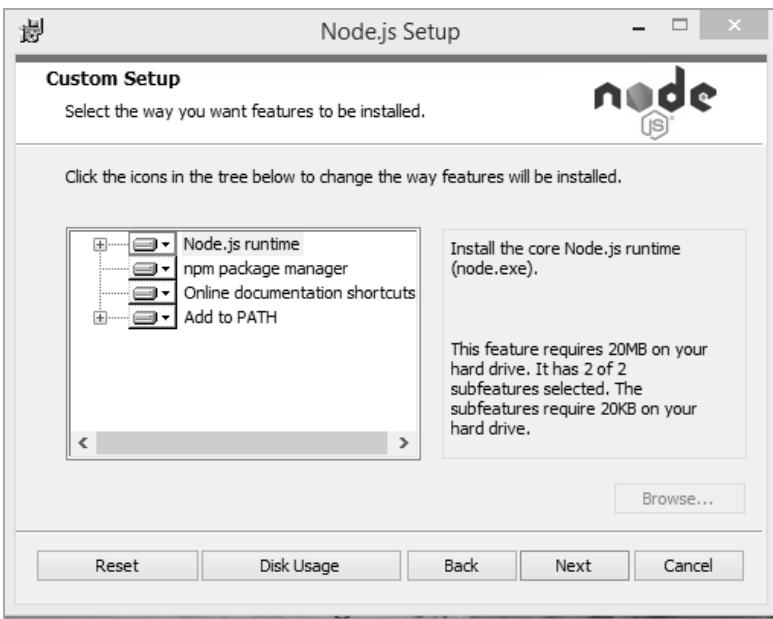
E:\SDK>node-v7.2.0-x64.msi

E:\SDK>
```

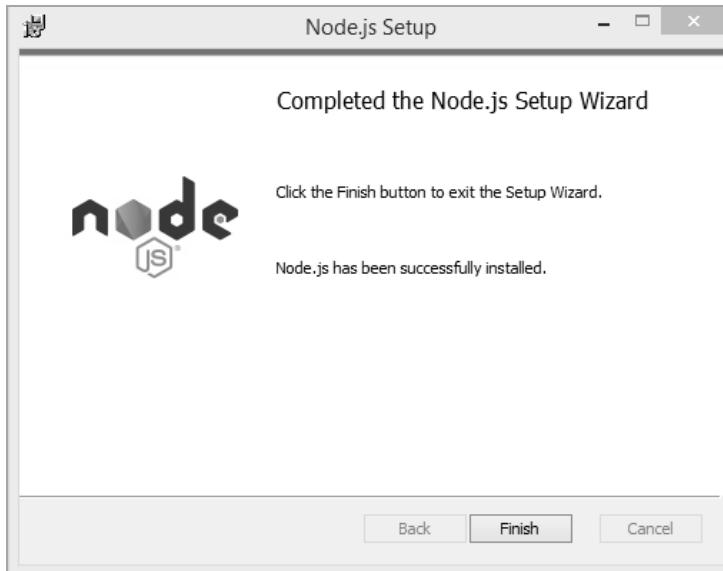
Enter the file name in command prompt and press enter to launch the installer. Once Installer opened, click on Next button on Node.js setup:



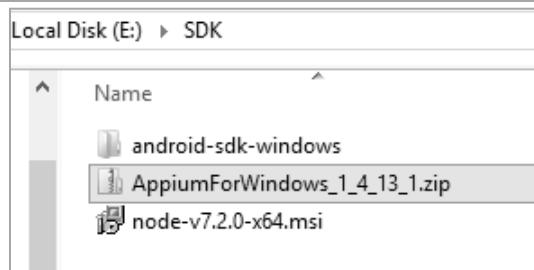


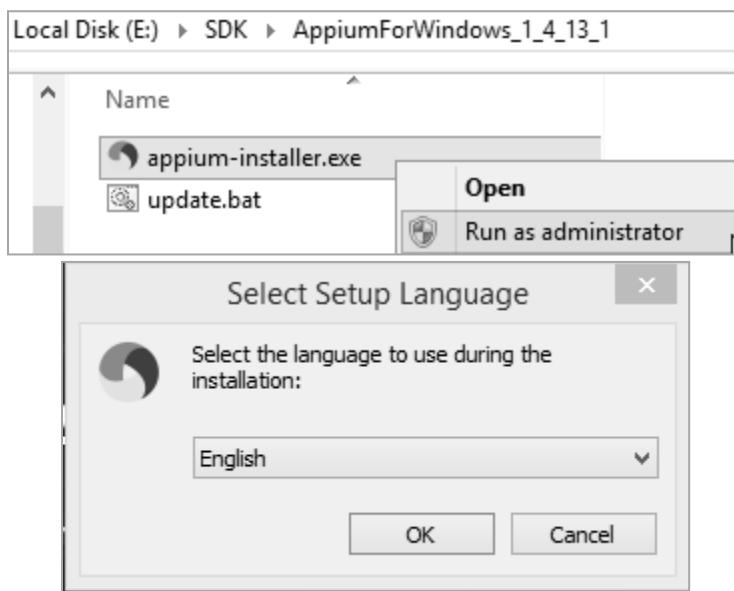
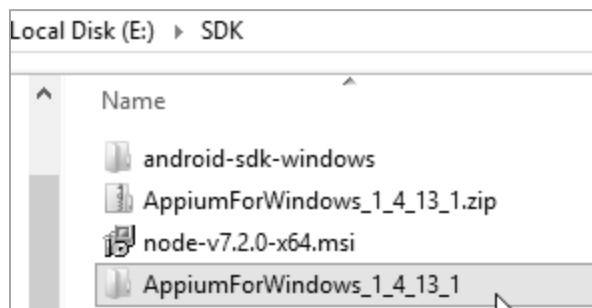
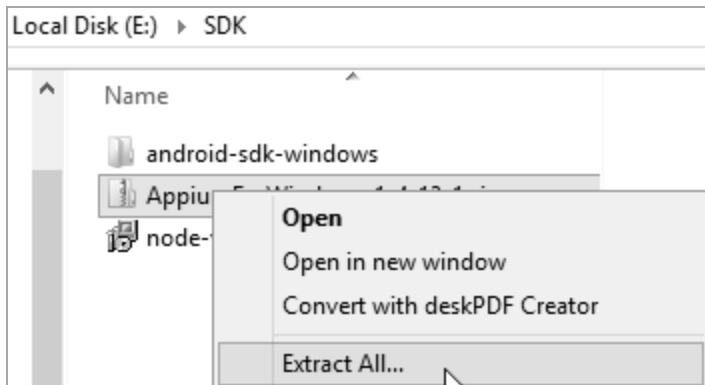


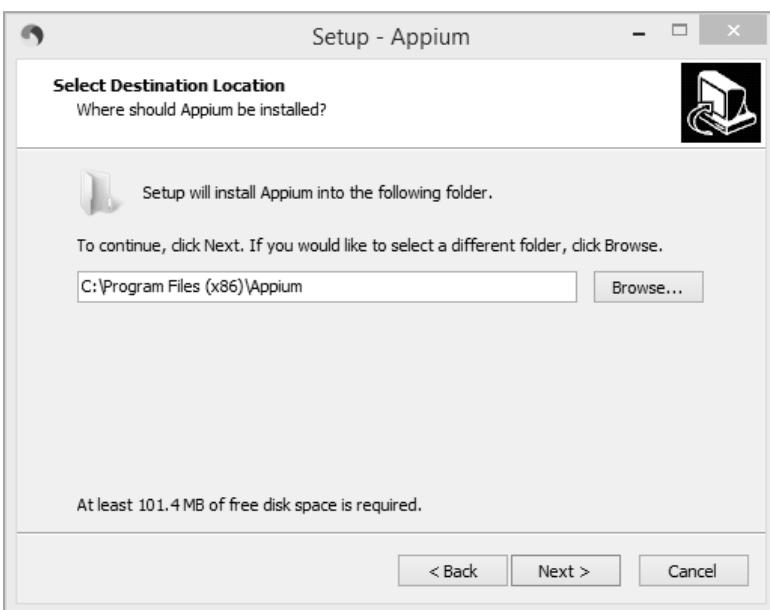
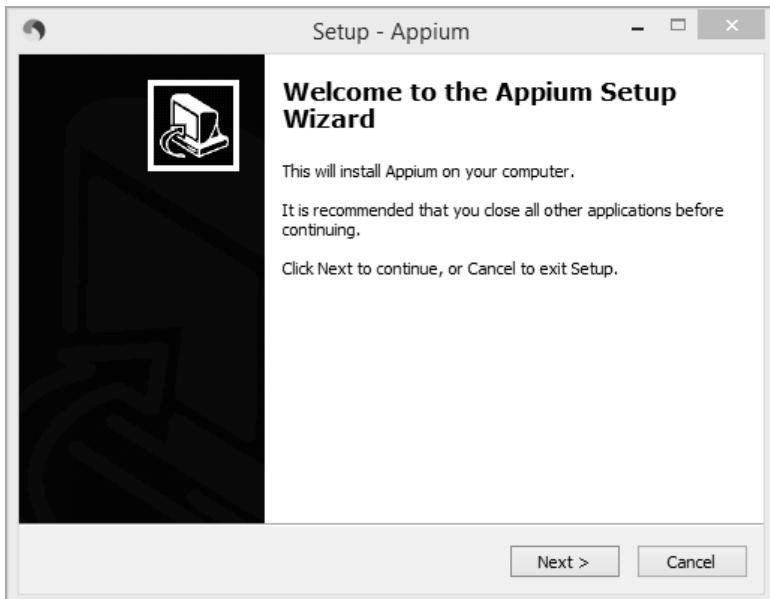
## Software Automation Testing Secrets Revealed

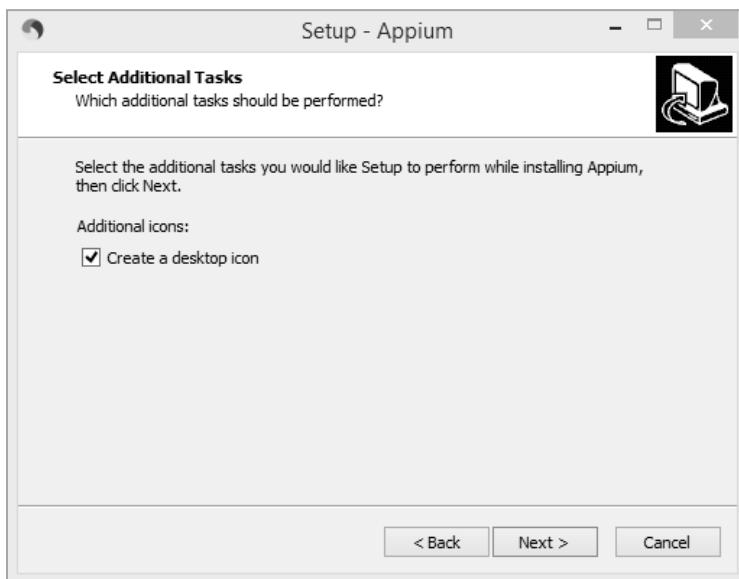
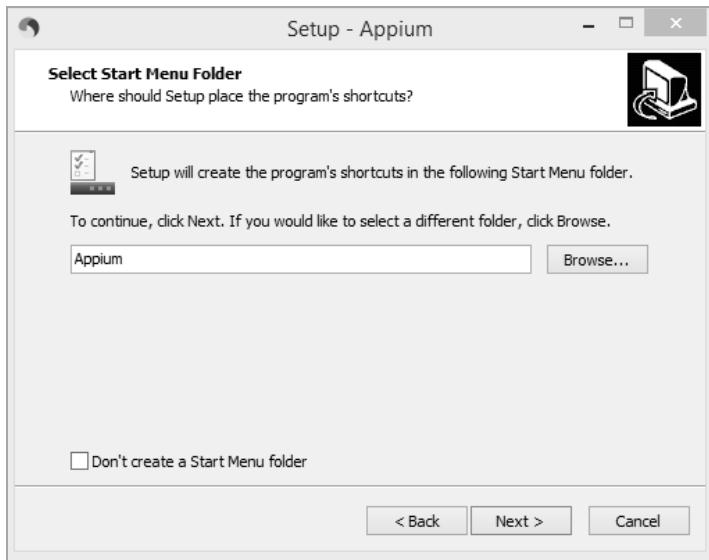


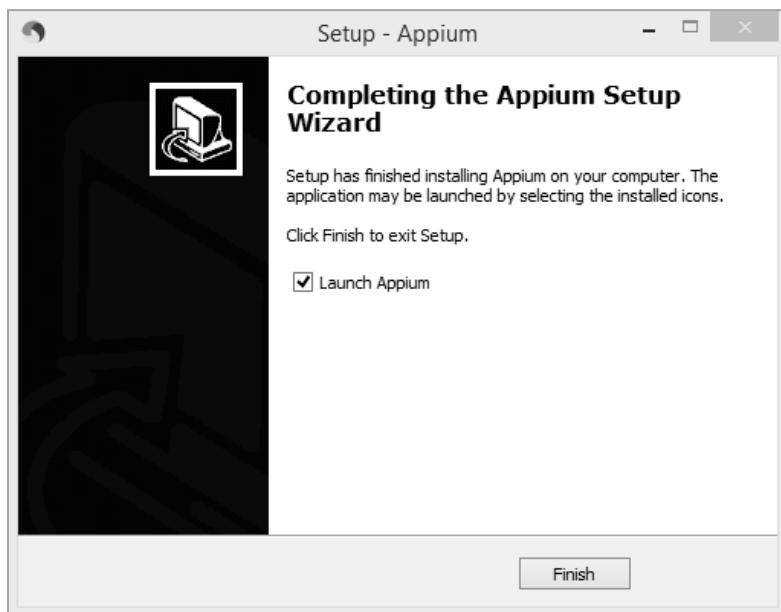
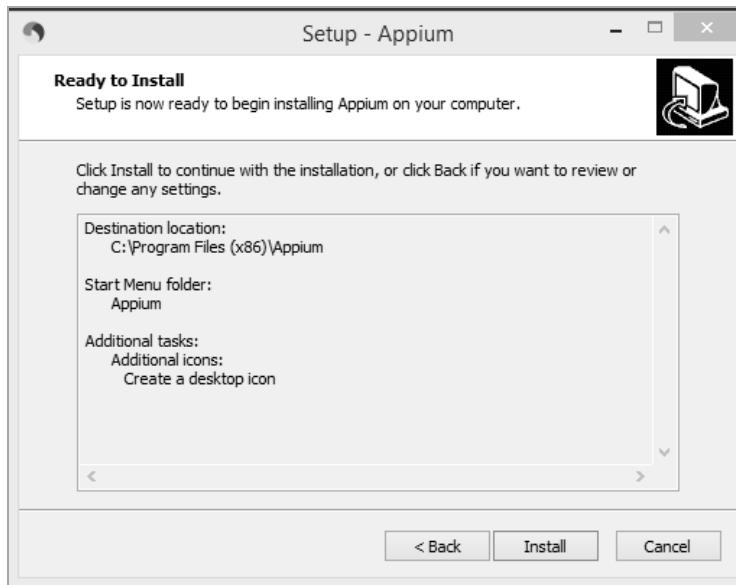
Once node setup completed, install Appium:

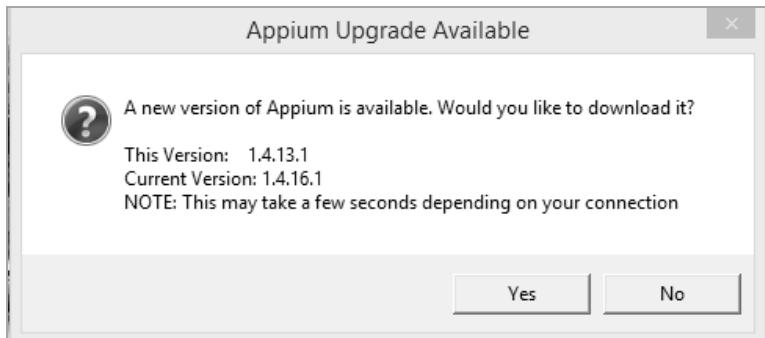










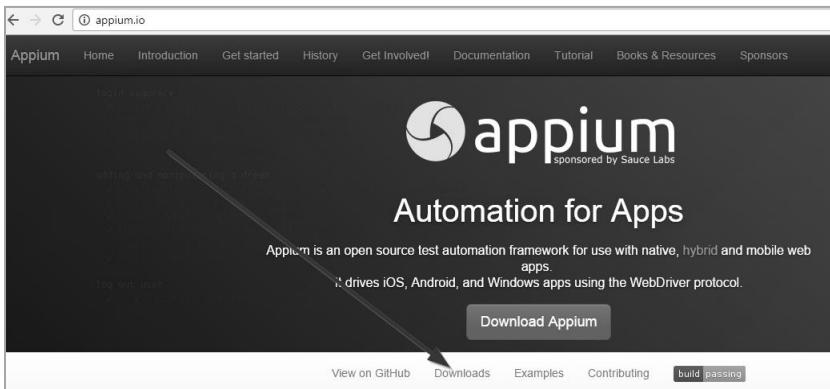


Launch Appium:



## Software Automation Testing Secrets Revealed

The next step is to download Appium client libraries:



The screenshot shows the 'Downloads' section of the Appium website. The main title is 'Downloads' with a subtitle 'Appium libraries & apps'. Below this, the heading 'Appium Client Libraries' is displayed. A large black arrow points from the left towards the list of supported languages. The list includes Ruby, Python, Java, JavaScript, PHP, C#, and RobotFramework. A cursor icon is positioned near the end of the language list.

# Downloads

## Appium libraries & apps

### Appium Client Libraries

Appium has language bindings for:

- Ruby
- Python
- Java
- JavaScript
- PHP
- C#
- RobotFramework

The screenshot shows the search results for 'io.appium.java-client' on The Central Repository. The search bar contains 'io.appium.java-client'. The results table has columns: GroupId, ArtifactId, Latest Version, Updated, and Download. One result is shown: 'io.appium.java-client' with version '4.1.2-all(28)' updated '21-Aug-2016'. The 'Download' link is 'dom-lax-javadoc-sources.jar'. A cursor icon is hovering over this download link.

https://search.maven.org/#search%7Cga%7C1%7Cg%3Aio.appium%20a%3Ajava-client

## The Central Repository

SEARCH | ADVANCED SEARCH | BROWSE | QUICK STATS

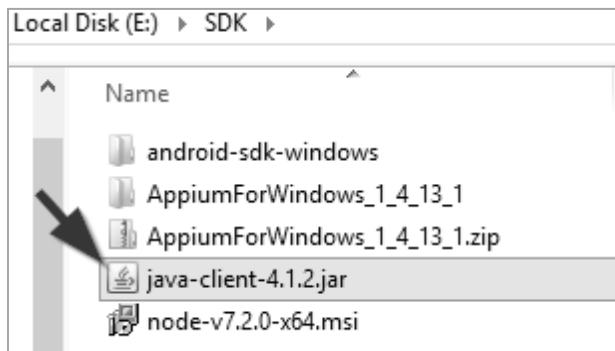
io.appium.java-client

About Central Advanced Search | API Guide | Help

Search Results < 1 > displaying 1 to 1 of 1

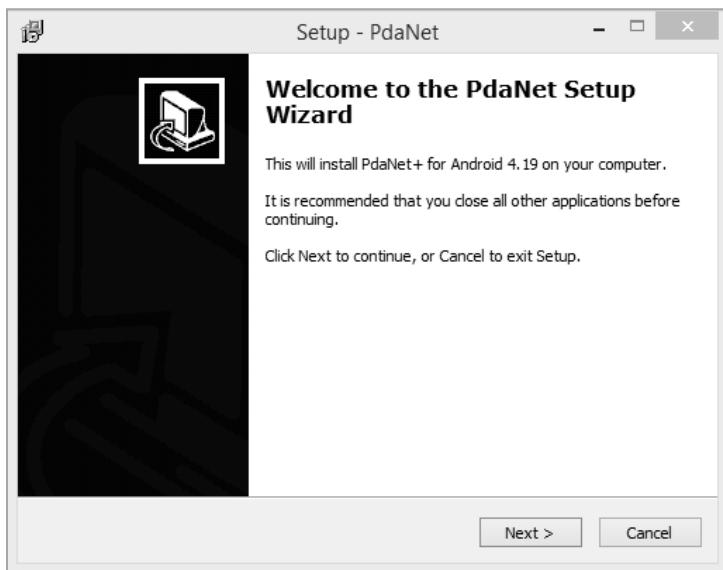
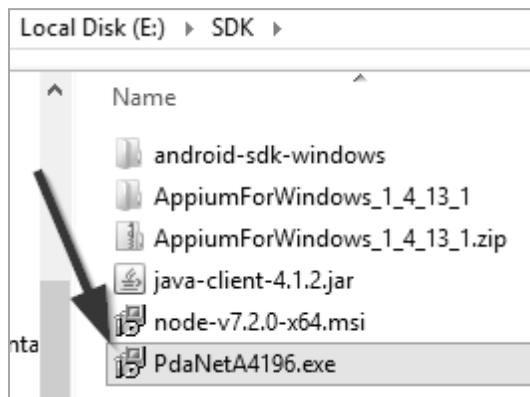
GroupId	ArtifactId	Latest Version	Updated	Download
io.appium	java-client	4.1.2-all(28)	21-Aug-2016	dom-lax-javadoc-sources.jar

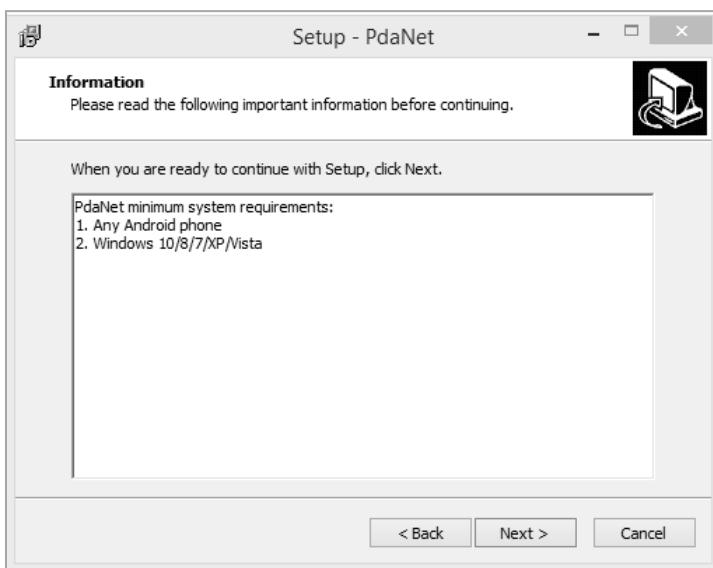
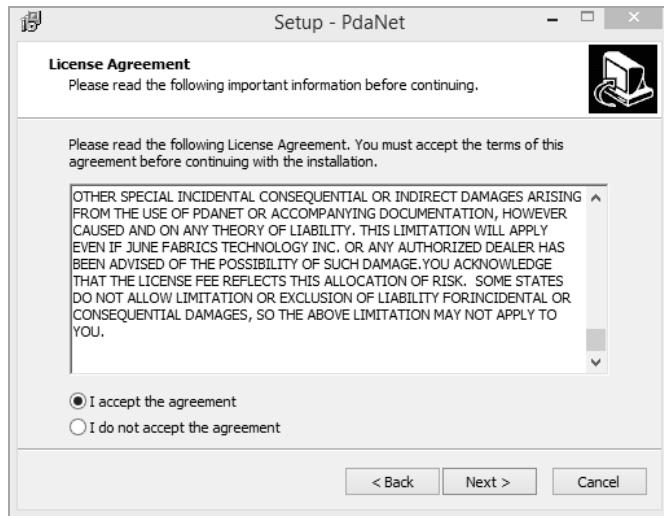
Ensure the availability of Java client inside SDK:

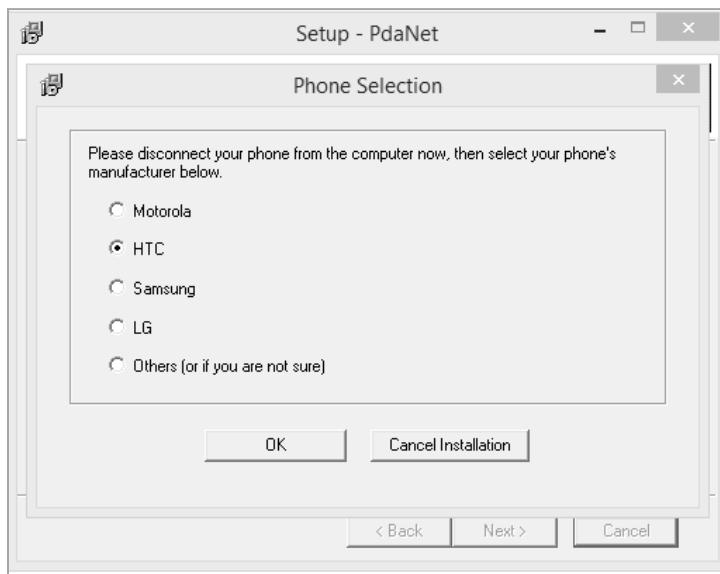
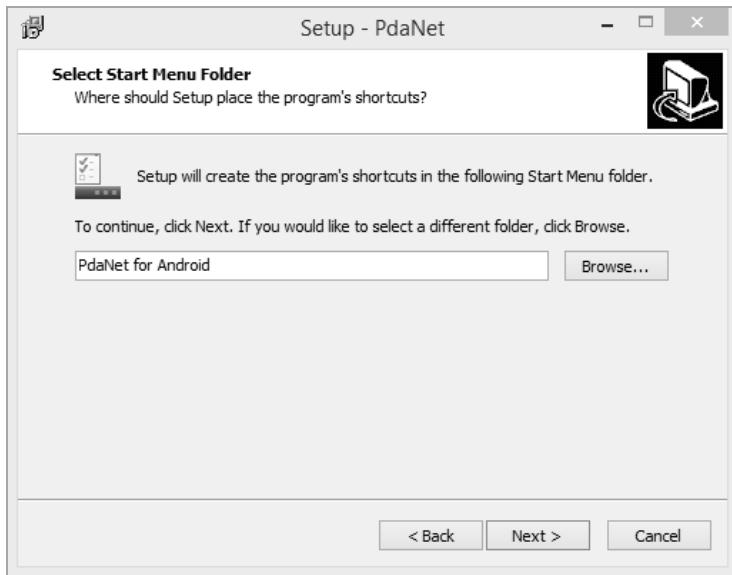


Install PdaNet tool for internet access:

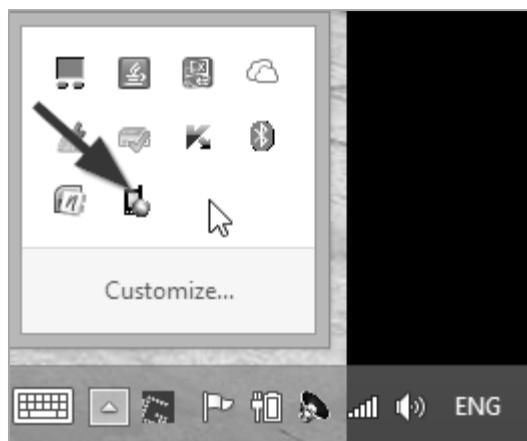
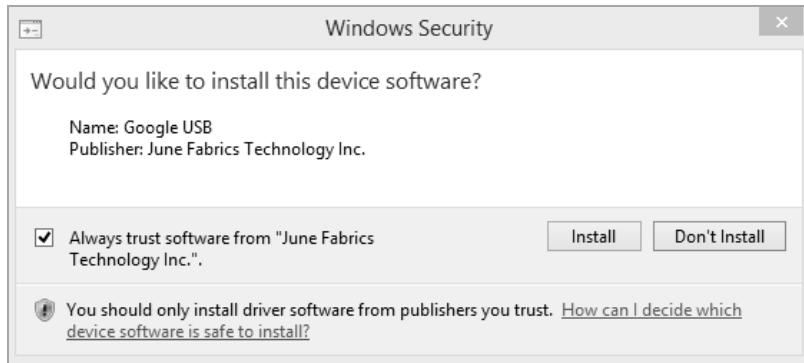
A screenshot of a web browser displaying the PdaNet+ website at [pdanet.co](http://pdanet.co). The page has a header with tabs for "Home", "FoxFi", "Help", and "Products". Below the header, there is a large Android robot icon with a globe and signal bars. To its right, the text "PdaNet + FoxFi" is displayed. The main content area contains several paragraphs of text about the product, followed by a section for "Version 4.0: PdaNet+ = PdaNet+FoxFi!". It also mentions "Version 3.02 improves performance and fixes multiple connection issues." and "Version 2.15 add supports for Mac USB Tether.". A link "Follow all PdaNet updates at [Twitter](#) (recommended)." is present. At the bottom, there is a "PdaNet+ for Android" section featuring a large Android robot icon, a "Download" button, and a "Purchase full version" button. A note at the bottom states: "Note: Free edition will interrupt your usage and requires you to turn back on."



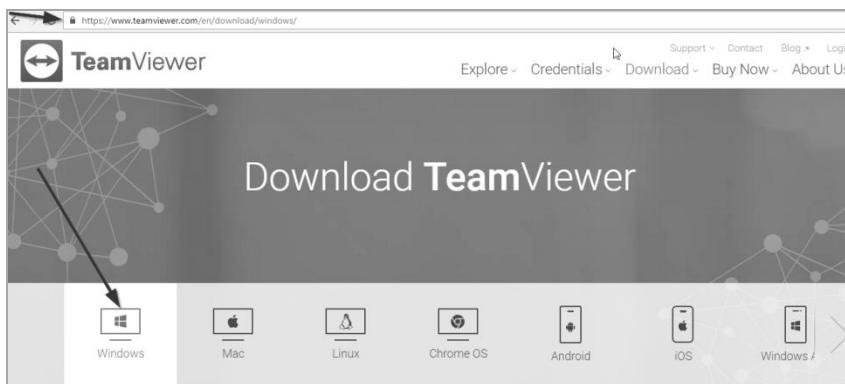


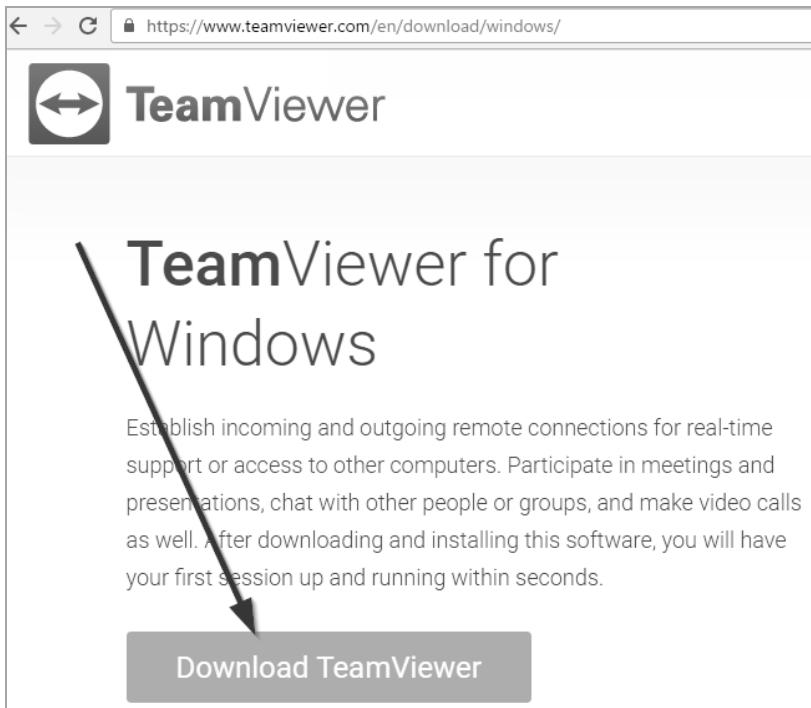


## Software Automation Testing Secrets Revealed



Download and install TeamViewer to access the mobile setup within windows:

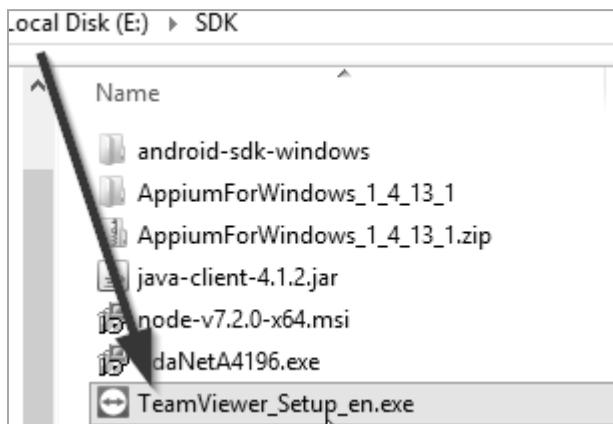




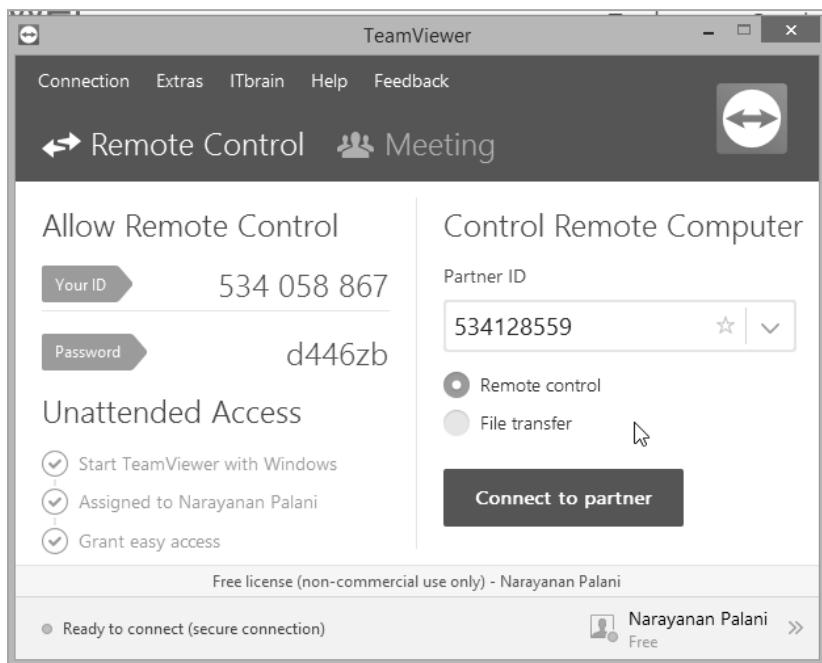
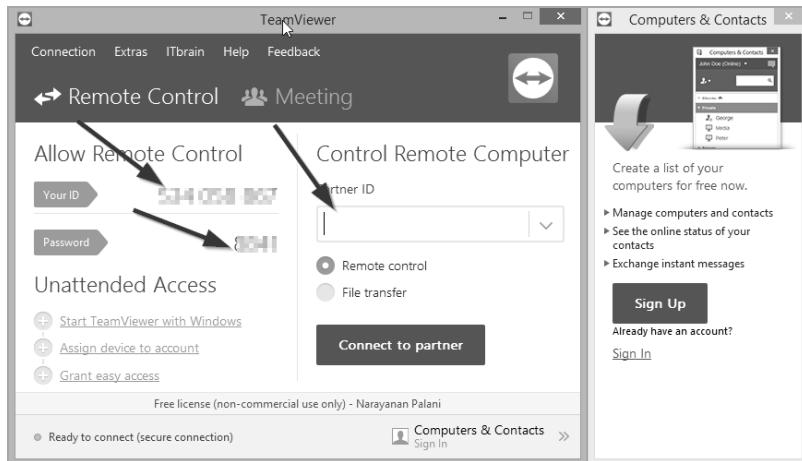
The screenshot shows a web browser window with the URL <https://www.teamviewer.com/en/download/windows/>. The page features the TeamViewer logo and a large title 'TeamViewer for Windows'. Below the title is a descriptive text block and a prominent 'Download TeamViewer' button.

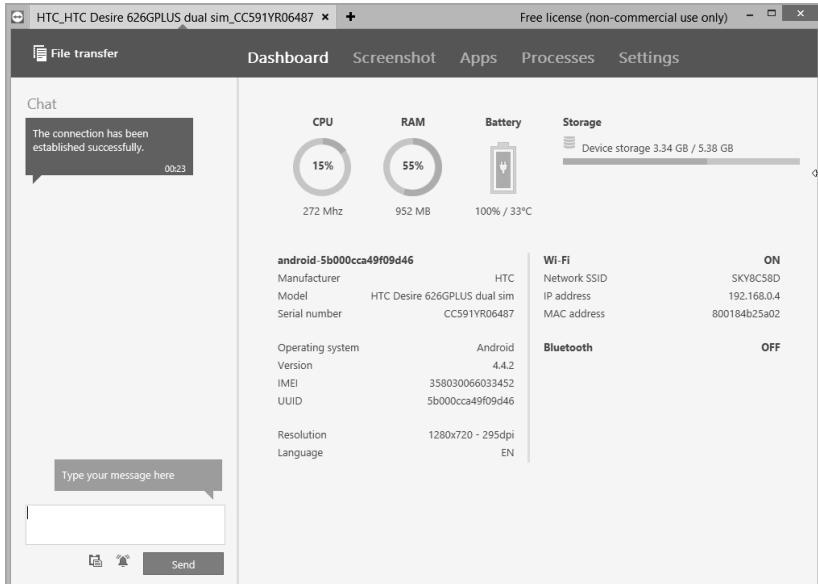
Establish incoming and outgoing remote connections for real-time support or access to other computers. Participate in meetings and presentations, chat with other people or groups, and make video calls as well. After downloading and installing this software, you will have your first session up and running within seconds.

Download TeamViewer



## Software Automation Testing Secrets Revealed





**Immediate assistance:**

**TeamViewer QuickSupport**

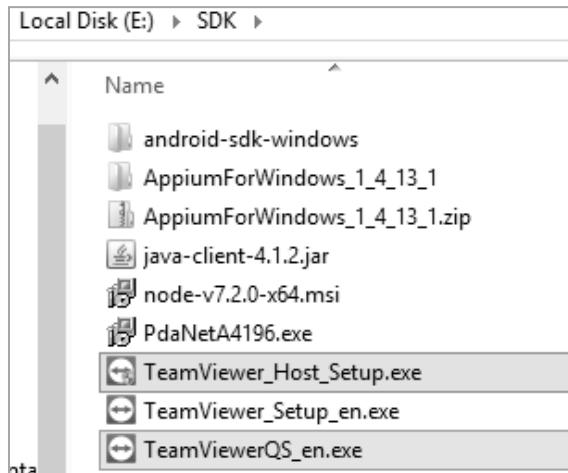
Optimized for instant support, this small customer module does not require installation or administrator rights – simply download, double click, and give the provided ID and password to your supporter.

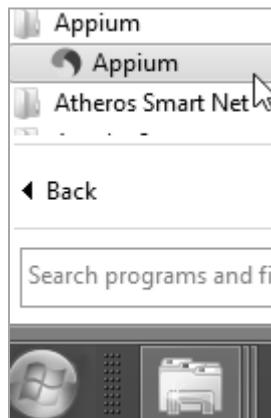
**Join remote control session**

## Unattended access: **TeamViewer Host**

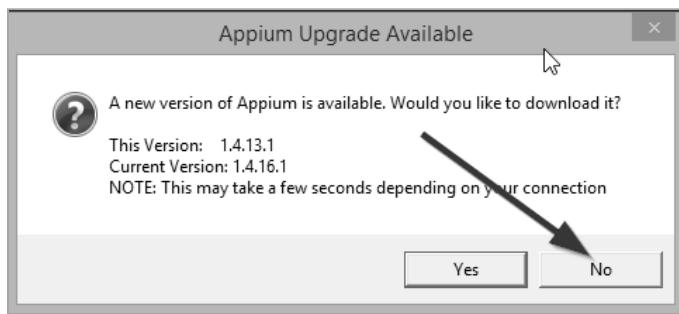
TeamViewer Host is used for 24/7 access to remote computers, which makes it an ideal solution for uses such as remote monitoring, server maintenance, or home-office access. Install TeamViewer Host on an unlimited number of computers and devices. As a licensed user, you have access to them all!

[Download Host](#)





Launch Appium(note that click on No when there is an upgrade available):

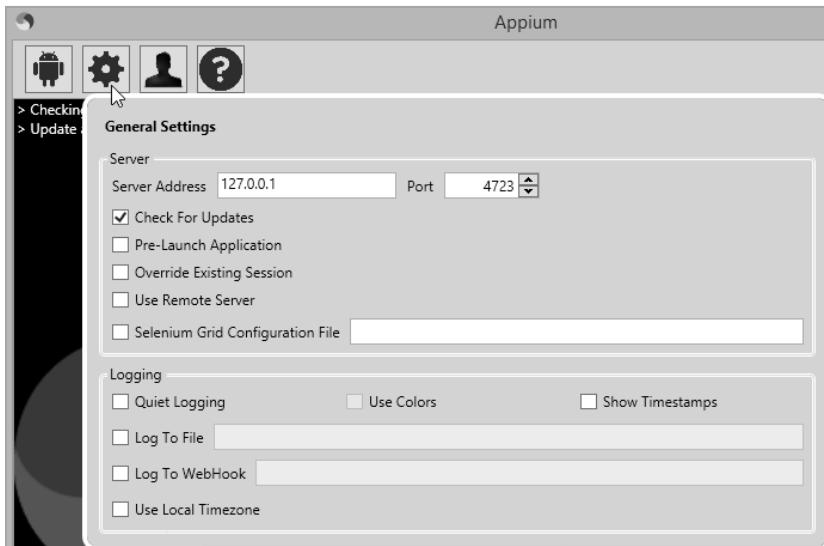


Click on General Settings:

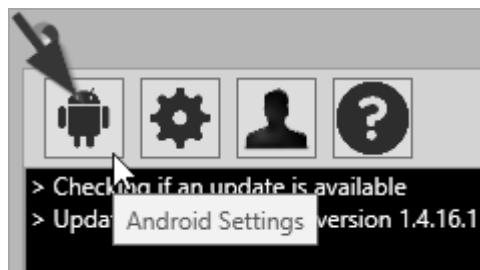


## Software Automation Testing Secrets Revealed

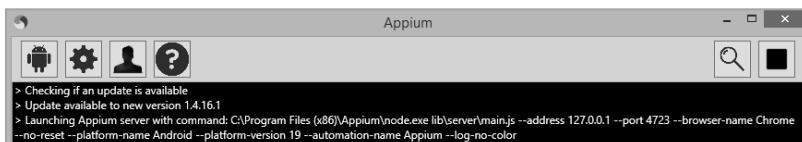
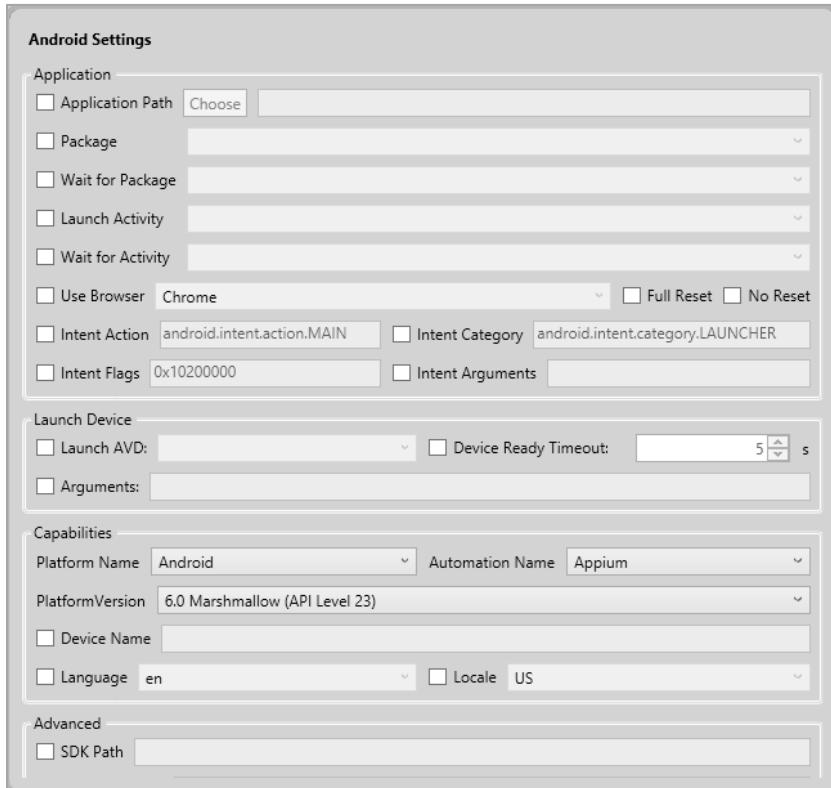
Update as shown below:



Click on Android Settings:



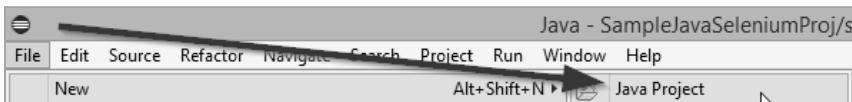
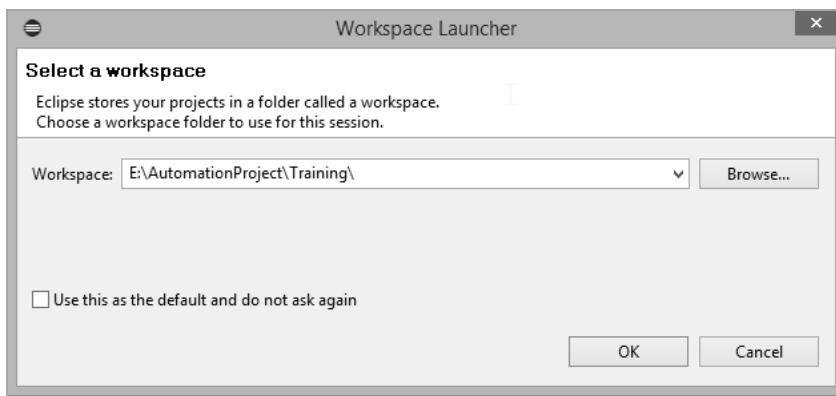
Update as shown below:

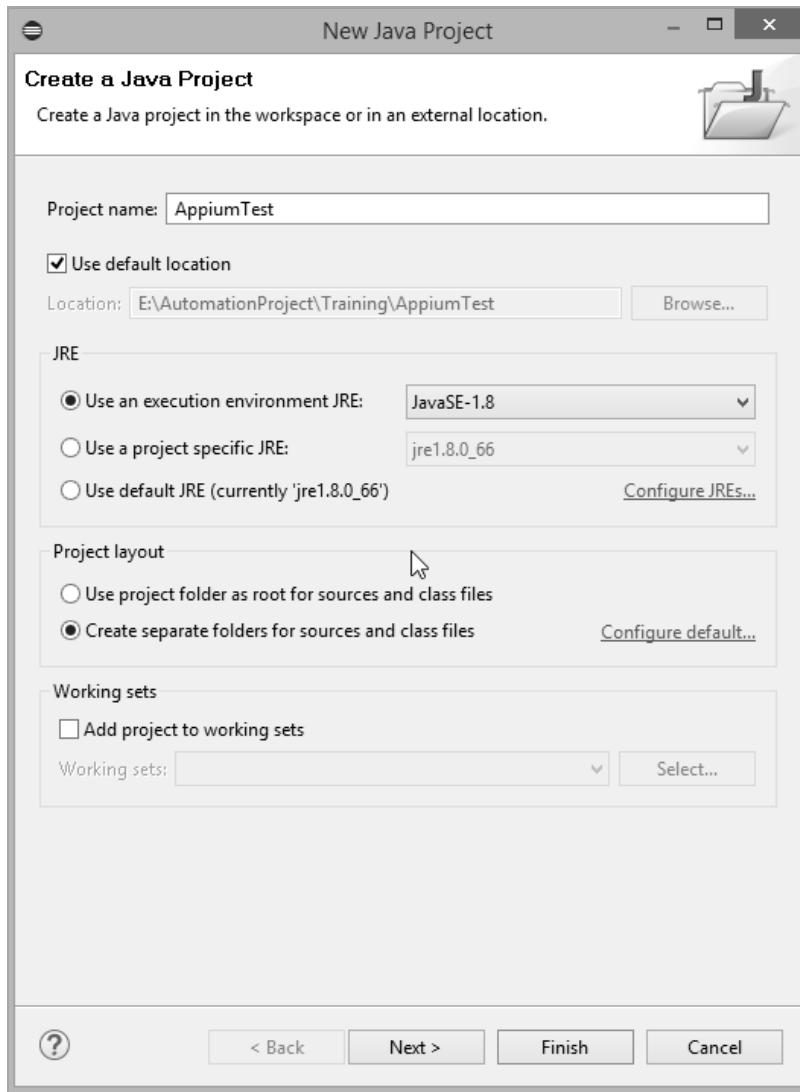


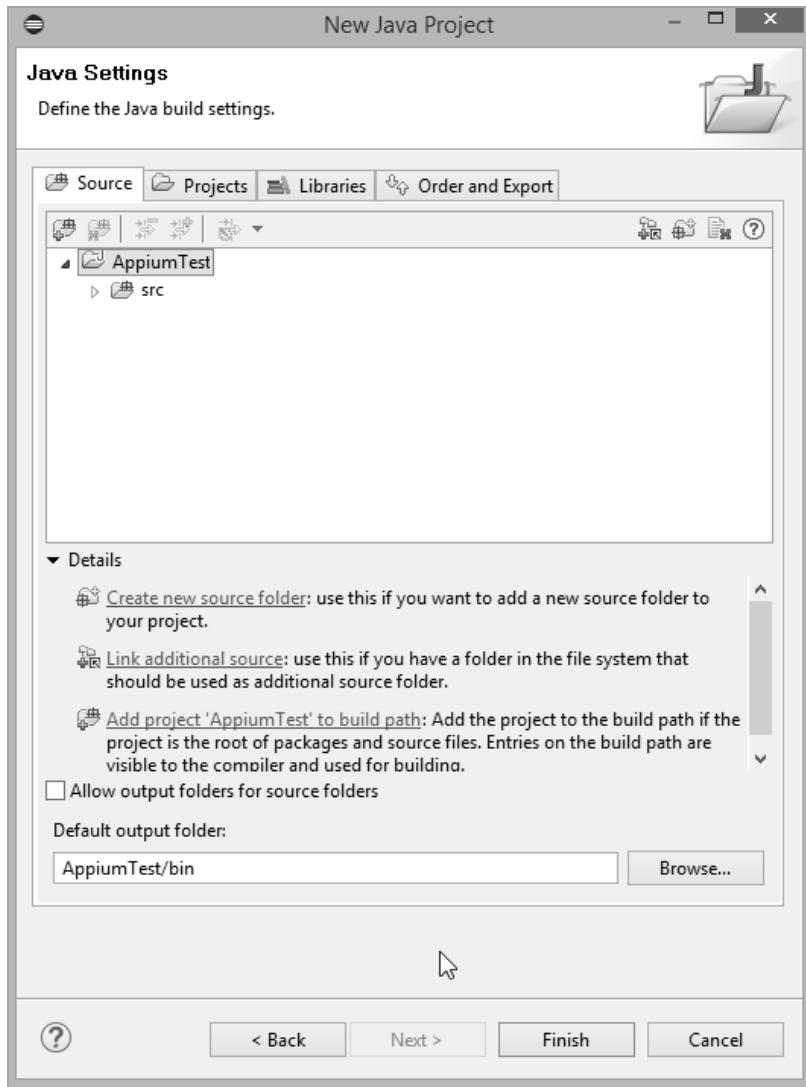
Launch Eclipse to load the IDE to script:

Name	Date modified	Type	Size
configuration	19/11/2016 17:40	File folder	
dropins	25/09/2014 14:52	File folder	
features	23/01/2016 13:20	File folder	
p2	17/10/2014 22:24	File folder	
plugins	23/01/2016 13:25	File folder	
readme	16/11/2015 19:59	File folder	
target	23/01/2016 21:27	File folder	
eclipseproduct	28/01/2015 10:08	ECLIPSEPRODUCT...	1 KB
ifacts.xml	23/01/2016 13:21	XML Document	185 KB
<b>eclipse.exe</b>	04/02/2015 09:07	Application	306 KB
eclipse.ini	16/11/2015 19:59	Configuration sett...	1 KB
epl-v10.html	28/01/2015 10:08	Chrome HTML Do...	13 KB
notice.html	28/01/2015 10:08	Chrome HTML Do...	9 KB

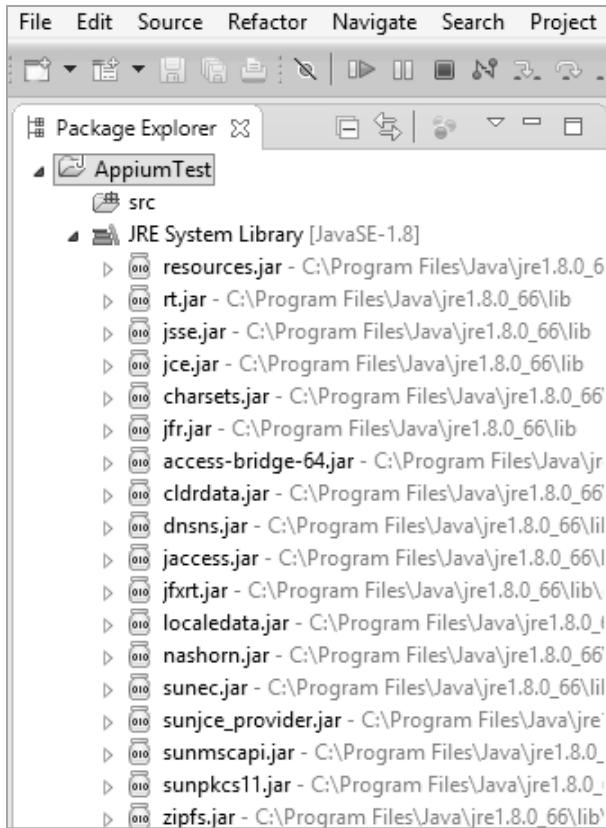
Choose or create a folder:







Create a project called Appium Test:



Download Selenium Server and place it inside the folder as per the details below:

[www.seleniumhq.org/download/](http://www.seleniumhq.org/download/)

**SeleniumHQ**  
Browser Automation

[edit this page](#) [search selenium:](#) [Go](#)

[Projects](#) [Download](#) [Documentation](#) [Support](#) [About](#)

**Downloads**

Below is where you can find the latest releases of all the Selenium components. You can also find a list of [previous releases](#), [source code](#), and additional information for [Maven users](#) (Maven is a popular Java build tool).

**Selenium Standalone Server**

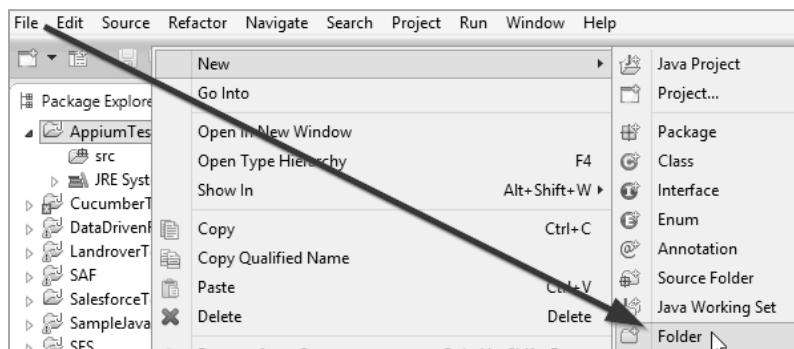
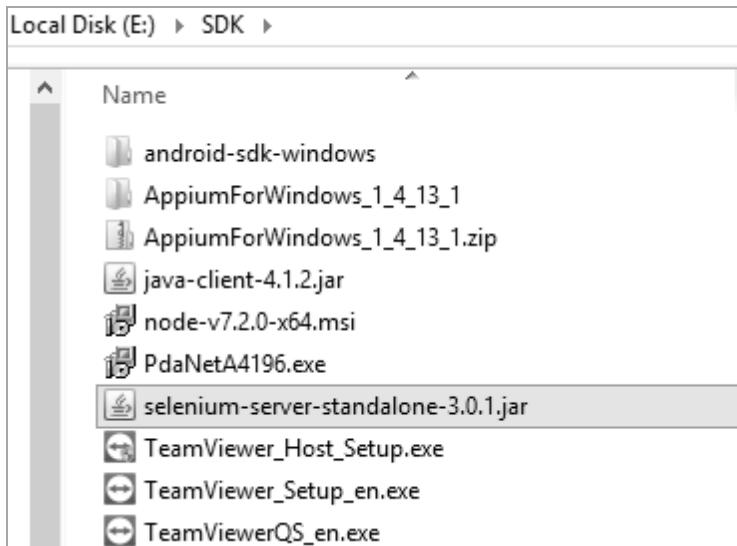
This Selenium Server is needed in order to run Remote Selenium WebDriver. Selenium 3.X is no longer capable of running Selenium RC directly, rather it does it through emulation and the WebDriver and Selenium Interface.

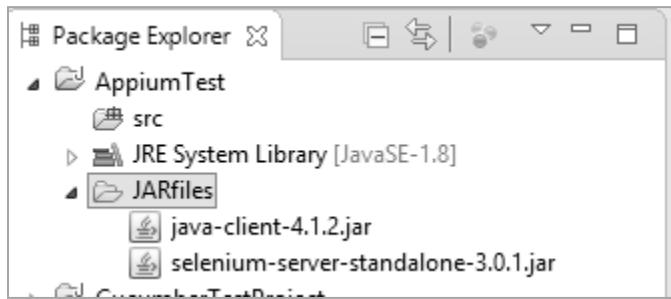
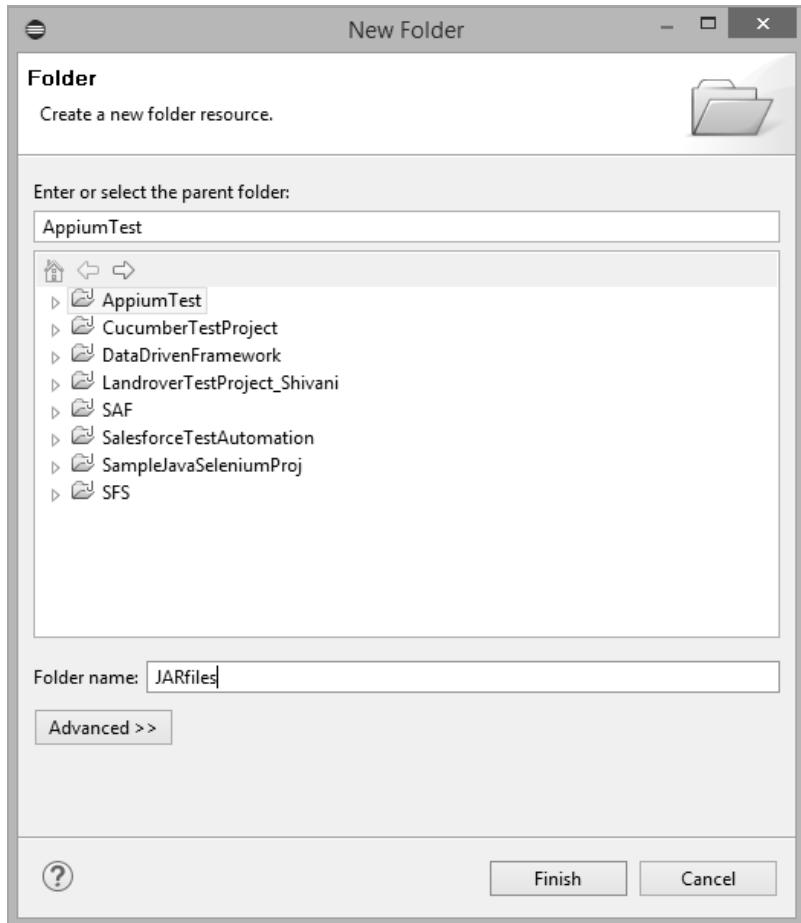
Download version [3.0.1](#)

To run Selenium tests exported from IDE, use the [Selenium Html Runner](#).

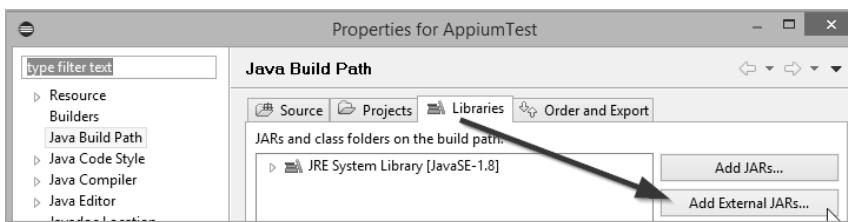
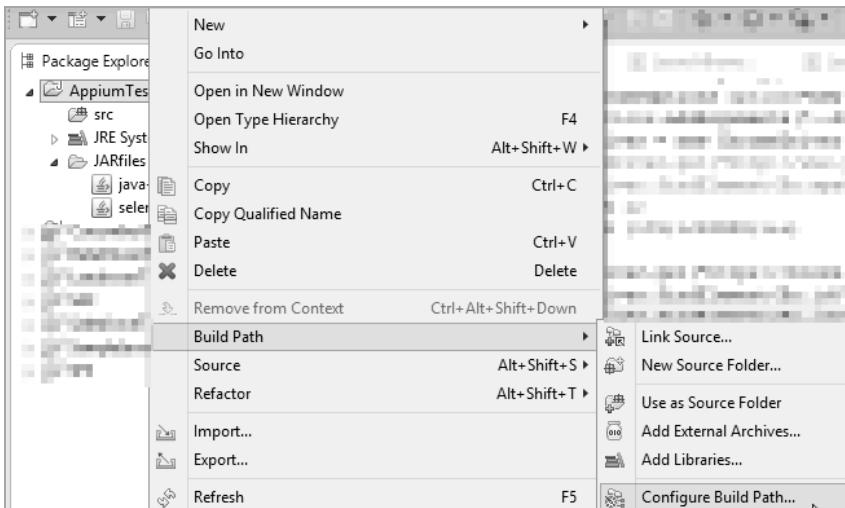
To use the Selenium Server in a Grid configuration [see the wiki page](#).

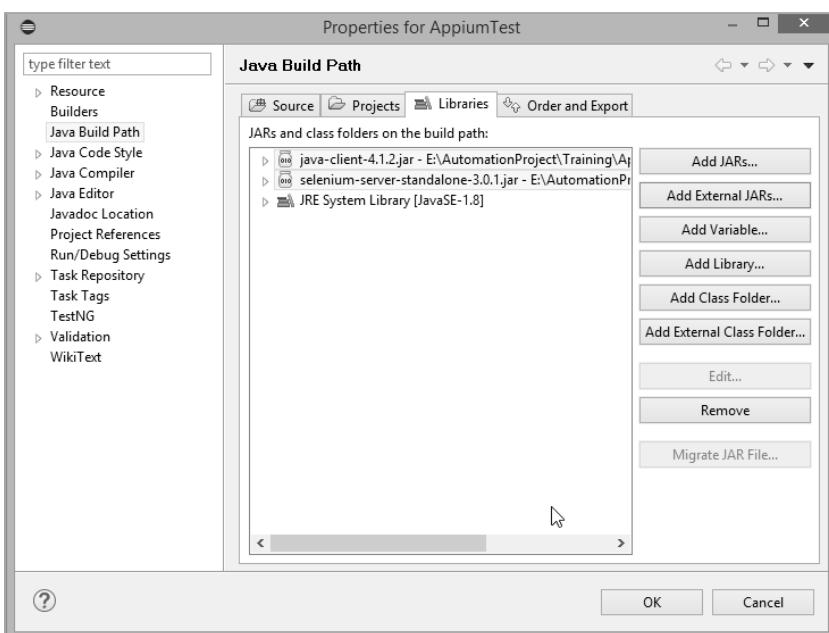
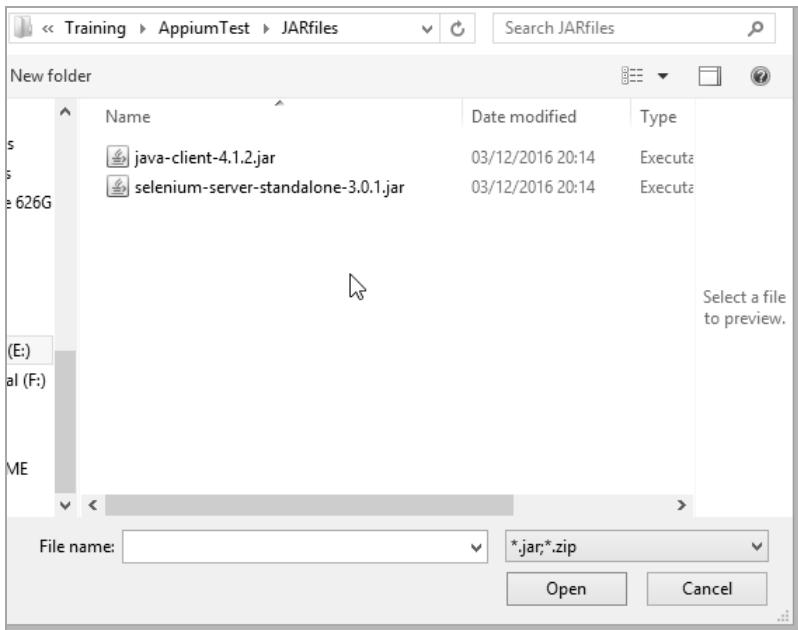
## Software Automation Testing Secrets Revealed

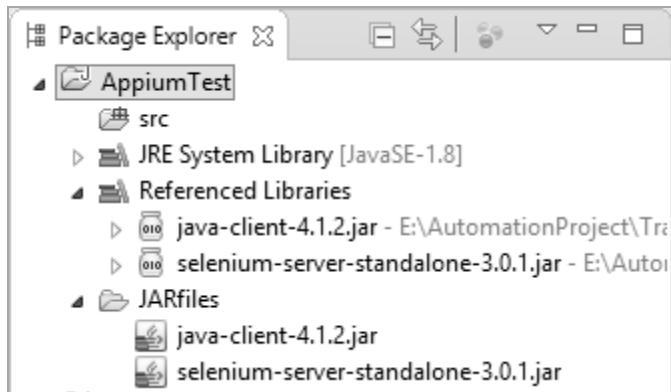




## Software Automation Testing Secrets Revealed

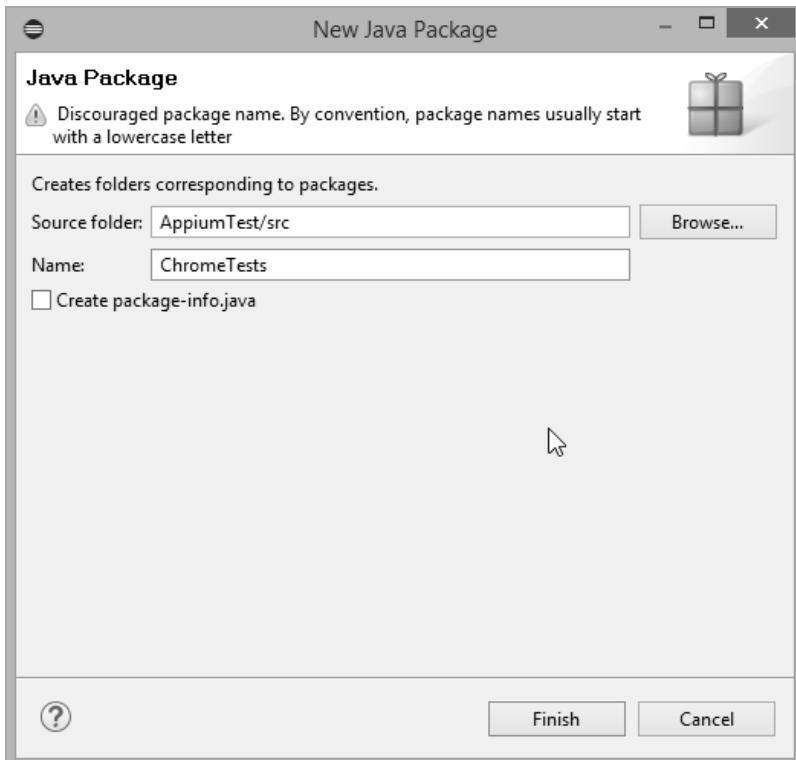




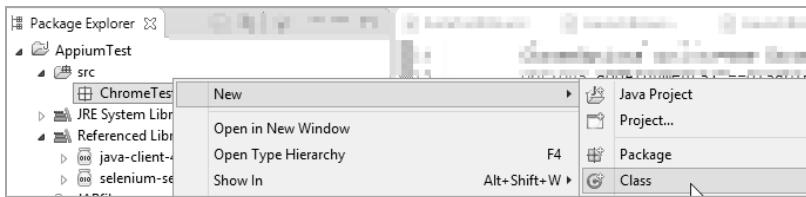


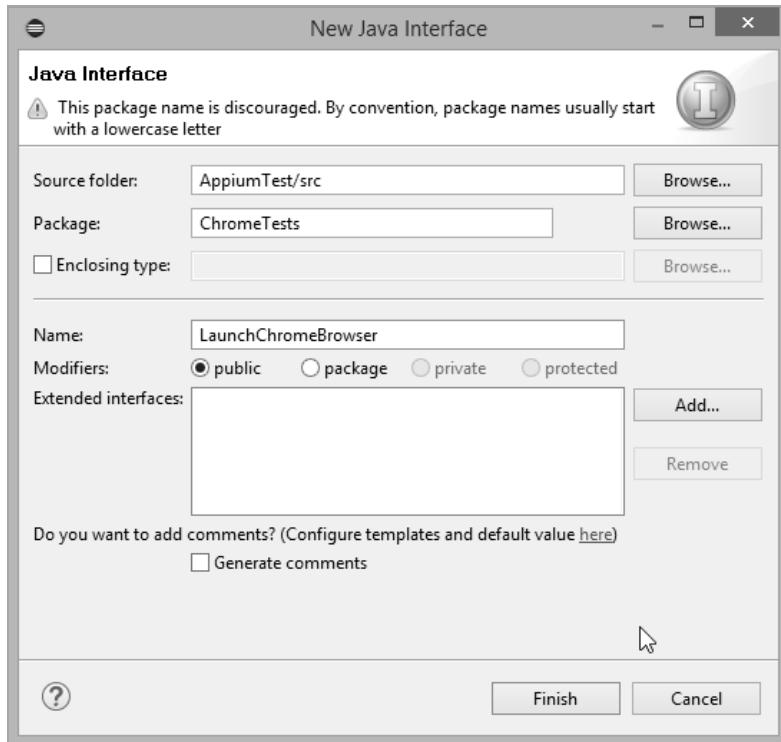
Click on the src to create a package:





Write your first class within the project:





## Write your first Appium script in the project

The screenshot shows the Java project structure and the generated code for the interface.

**Project Structure:**

- AppiumTest
- src
- ChromeTests
- LaunchChromeBrowser.java

**Generated Java Code:**

```
1 package ChromeTests;
2
3 public interface LaunchChromeBrowser {
4
5 }
6
```

Describe the imports for the project dependencies:

```

1 package ChromeTests;
2 import java.net.MalformedURLException;
3
4 public interface LaunchChromeBrowser {
5
6 }
7

```

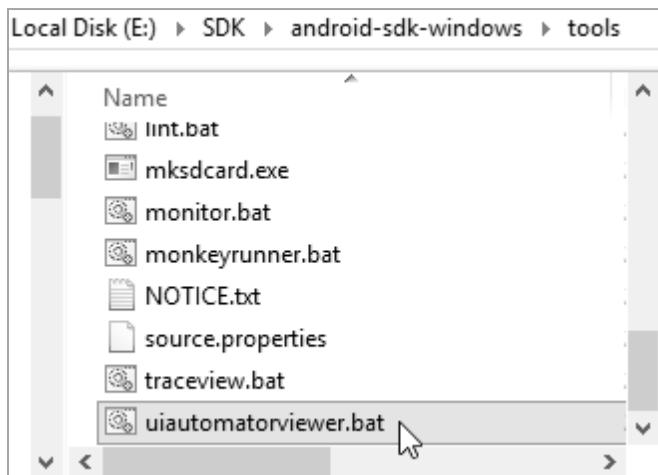
Mention @Test to call using test runners:

```

1 package ChromeTests;
2 import org.testng.annotations.Test;
3 import java.net.MalformedURLException;
4
5 public interface LaunchChromeBrowser {
6
7 @Test
8 public void browserTest() throws MalformedURLException{
9
10
11 }
12
13 }

```

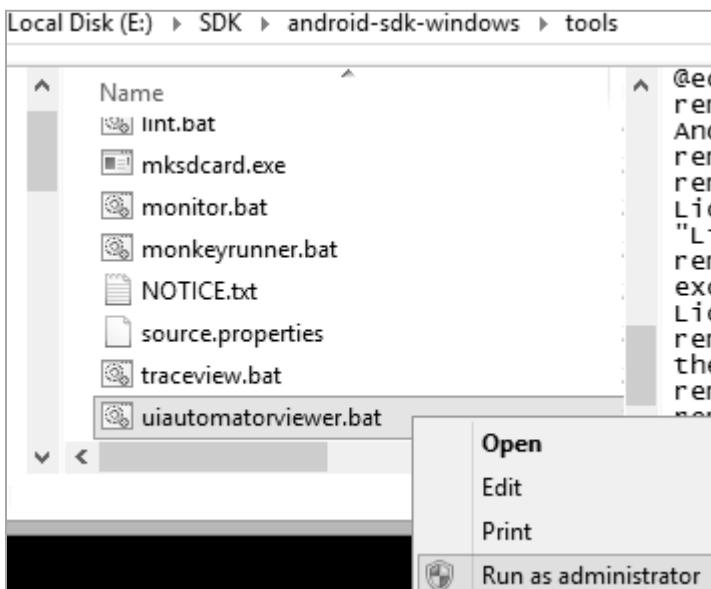
Click on the bat file below:



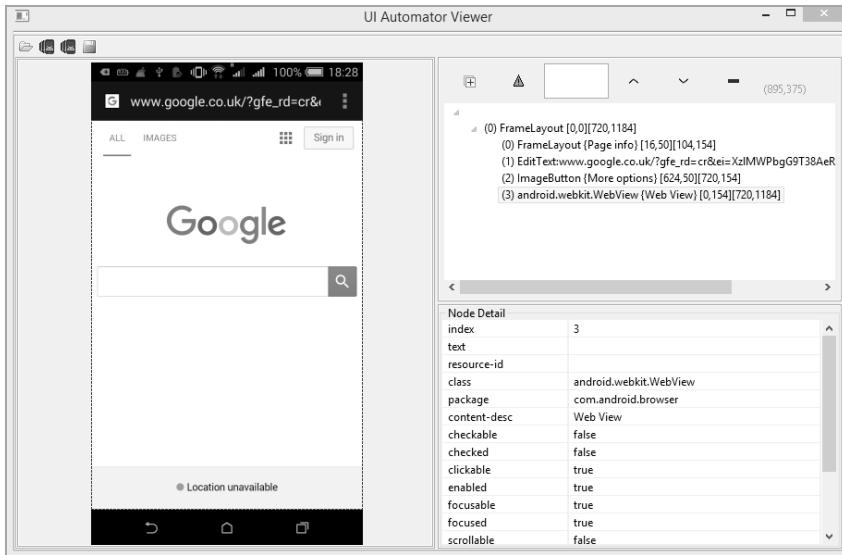
Check adb devices in command prompt to double check the device is listed:

```
C:\Users\naray_000>adb devices
List of devices attached
CC591YR06487    device
```

Run the bat file with admin privileges:



Now the mobile screen can be accessed from UI Automation Viewer:



This chapter is elaborately explained on how to setup a basic scripting base to start scripting on the Appium right from the scratch on any mobile applications.

## First Appium script

```
package ChromeTests;
import org.openqa.selenium.By;
import org.openqa.selenium.Platform;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.remote.BrowserType;
import org.openqa.selenium.remote.CapabilityType;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.RemoteWebDriver;
import io.appium.java_client.android.AndroidDriver;
import io.appium.java_client.remote.MobileCapabilityType;
import io.appium.java_client.remote.MobilePlatform;
import org.testng.annotations.Test;
import java.net.MalformedURLException;
import java.net.URL;
```

```
public class LaunchChromeBrowser {  
  
    @Test  
    public static void main(String[] args) throws  
    MalformedURLException, InterruptedException{  
        DesiredCapabilities  
        capabilities=DesiredCapabilities.android();  
        capabilities.setCapability(MobileCapabilityType.BROWSER_  
NAME,BrowserType.CHROME);  
        capabilities.setCapability(MobileCapabilityType.PLATFORM  
,Platform.ANDROID);  
        capabilities.setCapability(MobileCapabilityType.PLATFORM  
_NAME,"Android");  
        capabilities.setCapability(MobileCapabilityType.DEVICE_N  
AME,"Narayanan Palani");  
        //capabilities.setCapability("device","Android");  
        //capabilities.setCapability("app", "Chrome");  
        capabilities.setCapability(MobileCapabilityType.VERSION,  
"4.4.2");  
        //WebDriver driver = new RemoteWebDriver(new  
URL("http://127.0.0.1:4723/wd/hub"), capabilities);  
        //driver.wait(2000);  
        //driver.get("http://www.yahoo.com");  
        URL url=new URL("http://127.0.0.1:4723/wd/hub");  
        WebDriver driver=new AndroidDriver(url,capabilities);  
        driver.get("https://www.google.co.in/?gws_rd=ssl");  
        driver.findElement(By.id("lst-ib")).sendKeys("Selenium  
Webdriver");  
        driver.findElement(By.name("btnG")).click();  
        driver.findElement(By.name("hdtb-msb/[3]/a")).click();  
    }  
}
```

## Technical Round-based Interview Questions

Some of the latest interview questions faced by automation test engineers across Asia Pacific are listed below:

- Which automation framework model have you worked on?
- Have you worked on CMS (Content Management System)?
- What is RTM (Requirement Traceability Management)?
- How good are you in java programming which would help in test scripting?
- How good are you in Selenium?
- When do we go for automation?
- Where were you storing your automation scripts?
- Why do I need to modify the test script?
- How many automation scripts can you write per day?
- If there is scope of system testing phase (in parallel to test automation POC),-are you comfortable working in manual testing team currently?
- Is there any tracker-requirement assigned to handle?
- What are the different locators that you have used in Webdriver?
- Can you give example of absolute and relative xpaths?
- What is difference between close and quit in Webdriver?
- Webdriver is a class or interface?
- Can you give the class name which implemented Webdriver interface?
- Scenario: If there is an element which contains ID attribute, that ID is split by two parts by '\_' (underscore), first part is constant and second part is changing.
- Suppose if element is hidden, how do we get text of that element?
- Difference b/w implicit and explicit waits?
- How you deal dropdowns?
- Scenario: There is a link on webpage, click on that link a window will open and now that opened window contains 3 different frames and you have to do operation on different elements in 3 frames and then come back to window.
- What is timeout in grid?
- What is browsertimeout in grid?
- What do you mean by max instance and max session?

- Explain different annotations provided by TestNG.
- If I want to check particular exception in TestNG, how will you do it?
- Write a program to make a string to reverse and print.
- Extract only numbers in the given string.
- How good are you in Selenium?
- Have u built framework?
- Explain the framework used in your project?
- How you were passing multiple data to your test script?
- WAP to select multiple elements in the list andto extract the elements selected.
- Have you used synchronization?
- What is the difference between implicit and explicit wait?
- Draw the architecture of pom framework and its uses.
- Tell about OOPs concepts.
- Scenario: Suppose you have amount field which accepts 0 to 1000, then write manual and automation testcases for that field.
- What is the differencebetween while and do while when we use while and do while?
- Tell about the Automation process in your company.
- What is difference betweenMaven and testNG?
- What is testNG?
- What types have you done?
- Draw your project architecture.
- Tell about your project.
- What are the important classes in Webdriver?
- What is Sikuli and Robot class?
- What is AutoIT?
- What is difference betweenAutoIT and Sikuli?
- Which is best in Sikuli and Robot class, justify with reason.
- Scenario: Suppose there is a window popup which accepts username and password, I used AutoIT, when test is running I open notepad parallel; Question: Code of the username and password is entered in notepad or in window popup?
- Can we use AutoIT on other operating systems?

- Do you know about any third party reporting other than testNG?
- Who will do the system integration testing?
- Tell difference between sanity and smoke testing.
- What is garbage collector?
- Tell the bug life cycle process.
- Tell SDLC and STLC process.
- What is overloading and overriding with examples which you used in your project?
- What is Selenium grid?
- Can't we execute tests parallelly without Selenium grid?
- What is difference in testNG and Selenium grid?
- How do you execute tests in Selenium grid?
- How do you attach screenshot to test, which tool is used for screenshot attachment?
- Scenario: Suppose ten testcases in one scenario, that scenario executed in 5 times, so
- How will you generate reports for 5 times, will reports override? If override is there, then how?
- Will you take backup of previous test reports?
- How will you read csv file, write sample code?
- Will java able to read excel files and what is the java code for reading excel file?
- Which Excel files are you using in your company, xls or xlsx?
- Write code for reading xlsx.
- How will you read pdf files?
- What classes you use for reading pdf file?
- What is use of pdfStripper class?
- What is deferred state?
- What is duplicate state of bug?

MOST REPEATED QUESTIONS	
WHERE WERE YOU STORING YOUR AUTOMATION SCRIPTS?	WHICH AUTOMATION FRAMEWORK MODEL HAVE YOU WORKED ON?  Answers varies like BDD,TDD or Hybrid/Data Driven Framework
Possible to store in coding repository such as Git,SVN etc	
WHY DO I NEED TO MODIFY THE TEST SCRIPT?	HOW MANY AUTOMATION SCRIPTS CAN YOU WRITE PER DAY?  It depends upon how lengthy one user story is! If the story is short, it takes an hour to three to write complete scripts with review amendments incorporated...
When test data changes or the amendments to the user story or even the selenium version changes to latest, it is must to modify the scripts to run with latest user story and latest framework setup...	
HAVE U BUILT FRAMEWORK?	WHAT IS AUTOIT?  Image based test validations are not possible through selenium hence autoit has been used to validate the image files...
Built own automation framework using Selenium,TestNG,Cucumber,Ant,AutoIT,Maven,JUnit and Log4j with screenshot capture,excel reading capabilities	

## Selenium Interview Questions and Author's Views

Based on the years of experience, interview questions differ and the trend of asking in depth tool specific questions are to be watched on the basis of recent interview questions.

Some of the most repeated interview questions to Automation Testers are answered as follows:

- What is the tool and purpose of Selenium set of tools?

Selenium(IDE,Webdriver,Grid) is a famous set of open source browser automation tool to automate web-based projects with actions such as type, click and selection from a drop down of a web page etc. and events such as check the input box visible or invisible etc.

- How is Selenium different from commercial browser automation tools?

The major benefit of Selenium is to script on testers choice by programming languages such as Java, javascript, Ruby, Python or C#.

This facility is not available with licensed tools such as HP UFT (using VB Script),Ranorex (using C#) etc.

- What are the set of automation tools available with Selenium?

Selenium IDE

Selenium Webdriver

Selenium Grid

Selenium RC (Deprecated from 3.0 onwards)

- Which type of Selenium Tool should I use for my project?

Selenium Webdriver is widely used for automation testing on web pages.

Selenium IDE is not suggested for large projects, whereas if there are plans to run concurrent tests, Selenium Grid can be used.

- What is Selenium IDE?

It is a plugin in Firefox to record the steps of actions and replay them as an automation test. It works only on Firefox browser!

- Which language is used in Selenium IDE?

Selenese is the html-based programming language used in Selenium IDE.

- What is Selenium 1.0?

Selenium 1.0 /Selenium RC (Remote Control) are developed to extend the framework in multiple programming languages since Selenium IDE have been used only on firefox and there is a need for a stable open source tool which has been fulfilled by Selenium 1.0 release!

	<b>Selenium 1</b>
Release Date	<b>2004</b>
Tools	<b>Selenium IDE,Selenium Core,Selenium RC,Selenium Server</b>
Advantages	<p>Jason Huggins-Project Manager for Selenium Core</p> <p>Table-based Runner</p> <p><b>HTML Table-Selenium IDE</b></p> <p>Three columns</p> <p><b>Selenium Core'</b> Designed by Jason for Time and Expenses System</p> <p>Since there were defects from Internet Explorer,Firefox,Opera for the Javascript Application (Time and Expenses System), he decided to run the tests in javascript and drive the application</p>
Disadvantages	Incredibly difficult to manage table-based scripting hence Selenium RC introduced

	<b>Selenium RC</b> scripts written in multiple languages and communicate to Selenium Server and this Server has internal communication(core protocol) to Selenium Core Javascripts.
	Scripts running in Javascript Sandbox is a real challenge hence integrated with Webdriver in next version(in Selenium2.0)
Life Span	<b>Selenium Core:</b> Limited [Not used in Selenium 3.0 onwards]

- What is Selenium 2.0?

	<b>Selenium 2</b>
Release Date	<b>2011</b>
Tools	<b>Selenium Webdriver,Selenium IDE,Selenium RC,Selenium Core,Selenium Server,Appium[for mobile]</b>
Advantages	<p>Webdriver introduced and bind tight with underlying web browser;Object Oriented</p> <p>Firefox is connected with the help of Firefox extensions</p> <p>Internet Explorer's COM object collection used to communicate for tests</p> <p>Chrome,Opera extended to be used as part of Webdriver tests.</p> <p>Introduced Jason Wire Protocol representing commands for Firefox Driver</p> <p>Selenium Core Project integrated with Webdriver project hence it is become <b>Selenium Webdriver</b></p>
Disadvantages	Challenger not possible to analyze source code of IE and other browsers

	Every browser follow different wire protocol – challenge for Webdriver
	Browsers are become more complicated in every version hence Selenium decided to use W3C for next version(in selenium 3.0)
Life Span	<p><b>Selenium Webdriver:</b>Guaranteed [But no plans to use from Selenium 4.0 onwards]*as per the update from Simon on Sept'16</p> <p><b>Selenium IDE:</b> Guaranteed</p> <p><b>Selenium RC:</b> Limited [Since Selenium Core is removed in selenium 3 onwards, scripts are to be migrated to Selenium Webdriver]</p> <p><b>Appium:</b> Guaranteed[Not yet migrated to Selenium 3.0 as on Sept2016]</p>

- What is Selenium3?

	<b>Selenium 3</b>
Release Date	<b>Approximately around 13-Oct-16</b>
Tools	<b>Selenium Webdriver,Selenium IDE,Appium[for mobile]</b>
Advantages	Beginning of W3C compliance Protocol
	W3C compliant Webdriver server:Gecko Driver is used for Firefox
	Selenium Core removed
Life Span	<b>Selenium Webdriver:</b> Limited [There are plans to remove from selenium 4 onwards as per the web discussions, yet to be confirmed]

What are the element locators used in Selenium to capture objects?

Types:

ID (Unique Locator):

Example:

```
driver.findElement(By.id("textbox"));
```

Class

Name

Value

Text (Link text , Partial Link Text)

XPATH (Relative X path )

CSS (Fastest Locator)

DOM

Recommended object: Request developer of the project to introduce an attribute for object called data-automation and use it in the automation framework such as:

Example:

```
driver.findElement(By.xpath("//*[@data-automation='test-username']"));
```

- What is Selenium Grid?

Suppose the tests need to run in IE, Firefox, Chrome, Safari browsers across Windows 8,Windows 2000,Windows NT desktops, it can be configured using Selenium Grid with the same set of scripts developed for automation tests.

## Selenium IDE-based interview questions and author's views

What are two modes of views in Selenium IDE (not Webdriver)?

Way1:Side bar (View > Side bar > Selenium IDE)

Way2:Popup window (Tools > Selenium IDE).

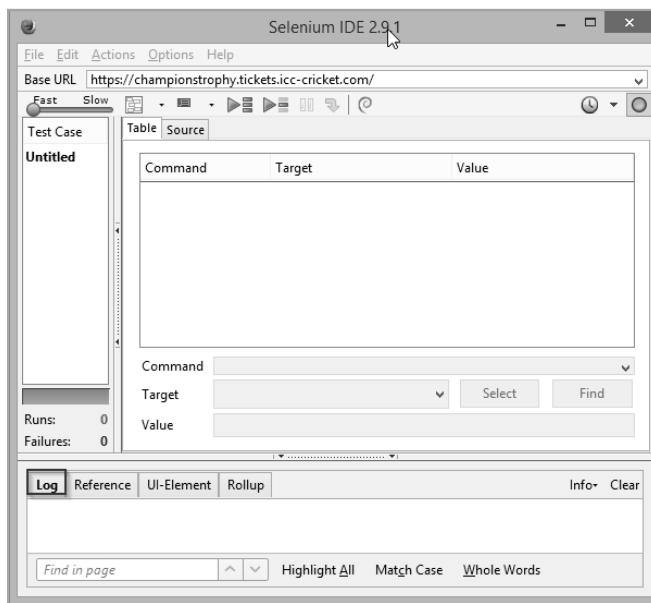
Note: In browser side bar it cannot record user operations in a popup window opened by application.

Can I control the speed and pause of test execution within Selenium IDE?

Functionality of a slider with Slow and Fast pointers is made to control the speed of test execution and it can be controlled before running the tests (not during the tests).

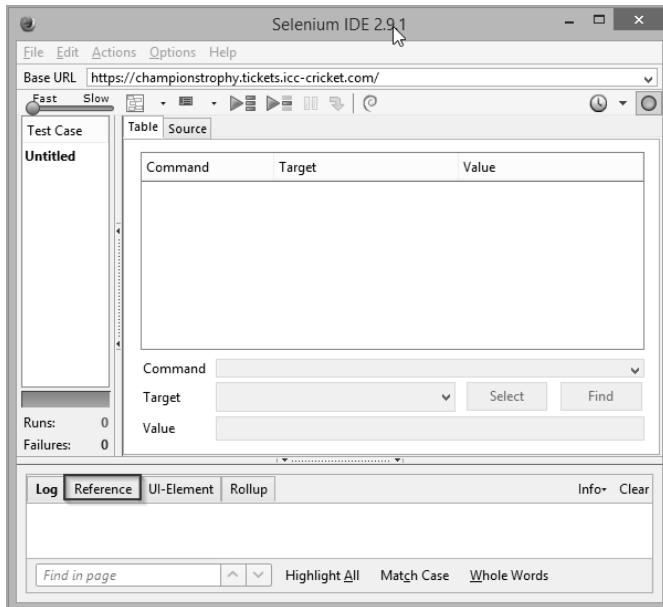
- Where do I see the results of Test Execution in Selenium IDE?

Log Window



- Where do I see the Commands Description in Selenium IDE?

Reference section

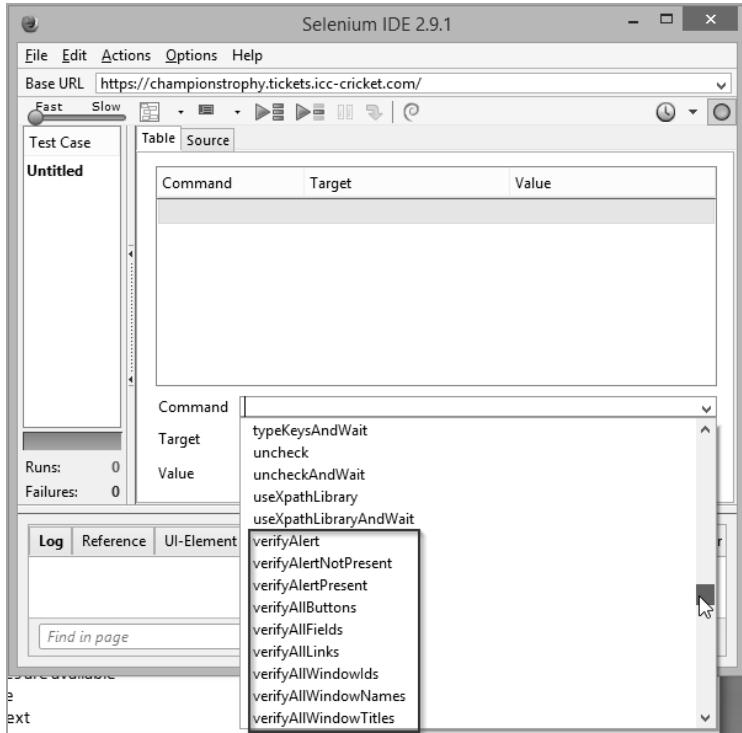


- Can I build entire test suite using Selenium IDE?

Record individual TCs, save them as html files and then group all of them in a Test (Test Suite) and save them as html as well.

- What verification points are available with Selenium?

Alert, Page Title, Links, Fields and most of the web elements can be verified and these commands are available within Command section:



Basically two types of checks in selenium – verification and assertion, what's the difference between them?

During the assertion, if the step failed, tests are getting stopped there, whereas if the verification steps failed, steps are not stopped and are run till the end (still the test is failed, but all the steps are completed)

- How do I amend tests in Selenium IDE?

Save the tests in html format after recording them. So it can be edited either in selenium IDE interface or in html (using notepad++ or other tools)

- What is the syntax of commands used in Selenium?

Command, Target and Value



- There are lots of Selenium Command, do I need to use all of them?

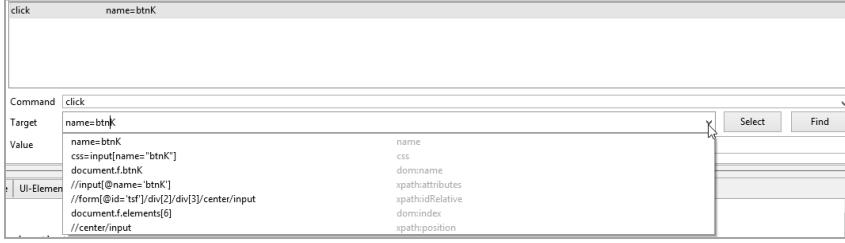
Though it is an opportunity to have loads of commands available within selenium IDE, it is advised to use only the needed commands.

- How to use html ID and name while using Selenium IDE?

Inspect Element:



Right click on any object – lets say Google search button and choose Inspect Element to see the name(btnK) and use it in Selenium IDE:



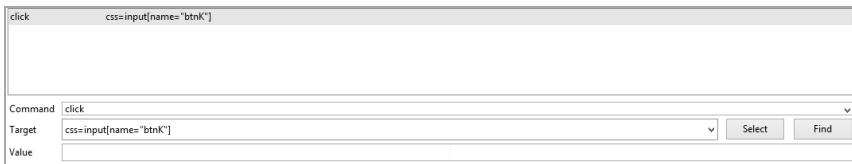
- What is XPath? When tester should choose to use XPath in Selenium IDE?

XPath is an option to easily navigate to the XML content of the page. In Google search button example, tester can use three types of xpaths as below:



- How to use CSS locator in Selenium IDE?

Google Search Button Example:



- There are ID, name, XPath, CSS locator and DOM. Hence, which one should I use?

If the name of object has been chosen and there are three objects with the same name in the page, test is going to fail in such cases! So finding and using a unique attribute of object matters! So it is suggested to use ID when no other object is unique or request developer to add new attribute such as test-object

- How to use javascript in Selenium IDE to add two numbers and count?

Use the below commands:

Command	Target	Value
store	60	Theory

<b>store</b>	40	Practical
<b>storeEval</b>	parseInt(storedVars['Theory']+parseInt(storedVars['Practical']))	addTotal
<b>echo</b>	<code> \${addTotal}</code>	

- Can I store result of an evaluation and use it later in my test?

In the above example, watch addTotal which is used in different row for display (echo).

- If I stored result of an evaluation; can I print it in IDE to check its value?

In the above example, echo prints addTotal.

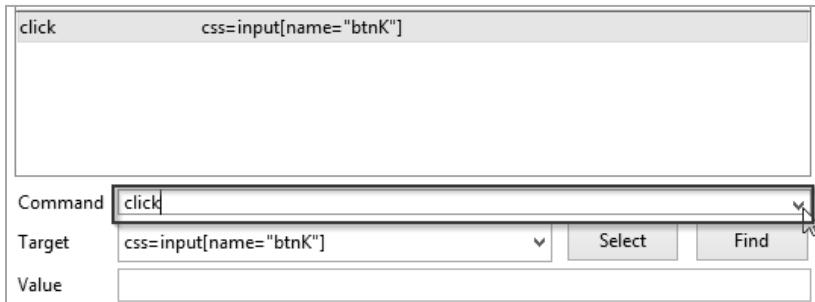
- Can I handle java script alert using Selenium?

assertAlert command has been used for this type of assertion.



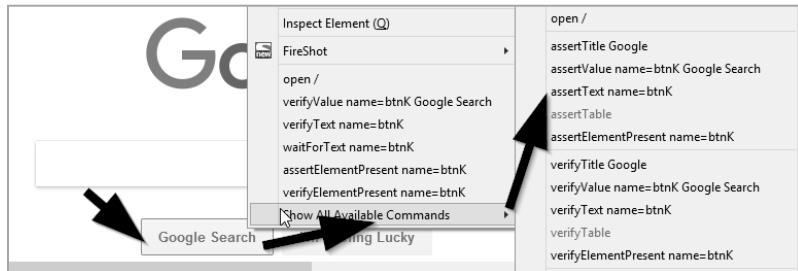
- Selenium has recorded my test using XPath, how do I change them to different locator?

Click on the dropdown in commands section:



- If my developer written custom attribute for the element locator, how do I use it?

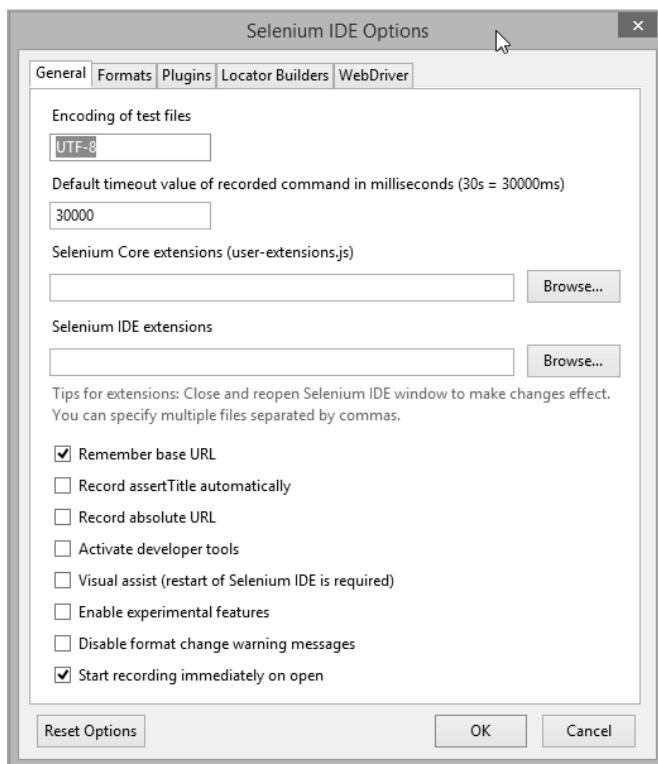
Have the selenium IDE on Record mode and right click on the object, choose “Show All Available Commands.”



Alternatively, use it in Target field in the right format.

- I have written one javascript extension; can I plug it in Selenium and use it?

Options window of Selenium IDE is used for this:



- How to convert Selenese-based Selenium IDE scripts to another language?

Format option within Selenium IDE is a method. Alternatively Selenium IDE recorded scripts can be used in JMeter (refer Part 2 of the book to know more).

- Execution options and tables-related tabs are disabled with Selenium IDE tests formatted to Java, why?

Change the Selenium IDE scripts to language of your choice but run them using Selenium 1.0 (note that many functionalities of Selenium IDE are not supported when the scripts are converted and used in different programming language).

- How to execute Selenium IDE-based Selenese tests in browsers other than Firefox?

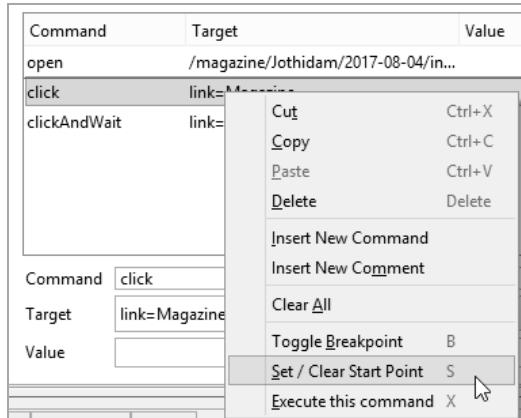
Use **htmlSuite** when starting Selenium Server and specifying different browser.

- If user add a command in the middle of other lines, how to test only that command in IDE of Selenium?

Double click on the specific command which will in turn run only particular step or rerun the test to validate.

- Can I set any command to start with?

Set start point mentioned below on the specific command:



Once it has been set, it shows a green symbol:

Command	Target	Value
open	/magazine/Jothidam/2017-08-04/in...	
▶ click	link=Magazine	
clickAndWait	link=Kumudam	

Are there any other add-ons or tools available outside Selenium IDE to help tests to identify element locators?

Some add-ons available from Firefox (but they work only in personal WiFi networks and blocked in organizations due to security reasons):

Xpath Checker for Firefox:



### Xpath finder for Firefox:



Xpath Helper for Chrome (but this would not help in running Selenium IDE tests since IDE meant for firefox only):



What is upcoming advancement in Selenium IDE?

Please refer [http://www.seleniumhq.org/docs/02\\_selenium\\_ide.jsp](http://www.seleniumhq.org/docs/02_selenium_ide.jsp)

What is the best option to insert loop-based commands in Selenium IDE?

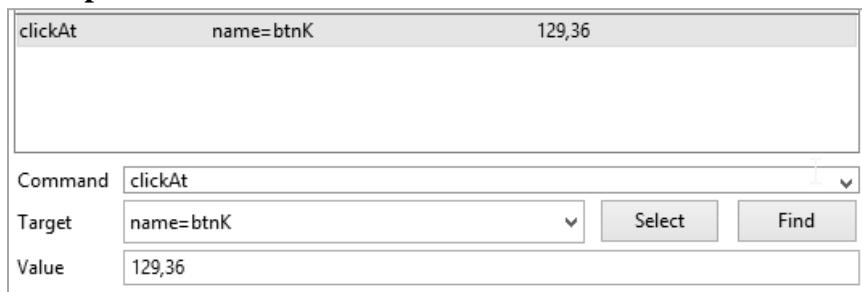
Use the extension from the reference website below to construct loops within Selenium IDE:

I want to click at specific part of my element by screen coordinate while using click command. Can I?

Following clickAt command helps in adding the coordinate x,y path to the script

`clickAt(locator, coordString)`

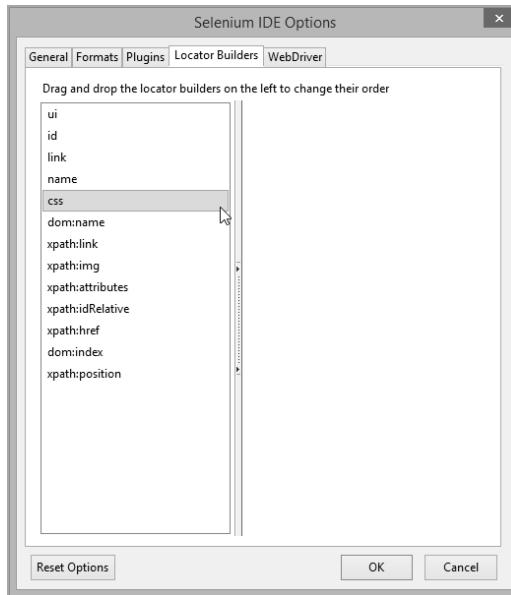
### Example:



How does the user verify dropdown values in Selenium IDE?  
`assertSelectOptions(command)`

How to get a data from html table cell using Selenium IDE?  
`storeTable(command)`

How to change Selenium IDE record and display to display css locator followed by other locators after recording the scripts? Is it worth or possible to give high priority to css in Selenium IDE?  
Navigate to Options >Options >Locator Builders and change the order to keep CSS on top:



How to handle dynamic alerts in Selenium IDE?

Use the command given below for OK Alerts:

Command	chooseOkOnNextConfirmation
---------	----------------------------

Use the command given below for click and wait:

Command	chooseOkOnNextConfirmationAndWait
---------	-----------------------------------

Use the command given below for CANCEL Alert:

Command	chooseCancelOnNextConfirmation
---------	--------------------------------

How to perform a right click on an element using Selenium IDE?  
contextMenu and contextMenuAt are the commands used frequently but they can't be added in recording time and it has to be added as a step manually in IDE interface.

## How to take screenshots using Selenium IDE?

Following are the available commands:

Command	<code>captureEntirePageScreenshot</code>
Target	<code>captureEntirePageScreenshot</code>
	<code>captureEntirePageScreenshotAndWait</code>

How to make Selenium IDE to wait or pause for sometime during the test?

Use the command and value given below:

Command	<code>pause</code>
Target	
Value	1000

If my web page takes more than 30 seconds to load, the URL launching fails in Selenium IDE, why?

If **Open** command used to launch the URL and it takes more than 30 seconds to load the page, test will fail hence use **setTimeout** command to set customized timeout for the page load

Why there are two types of commands such as type and typekeys?

Command	<code>type</code>
Target	<code>type</code>
Value	<code>typeAndWait</code>
	<code>typeKeys</code>
	<code>typeKeysAndWait</code>

First command types the entire word in one go, whereas the second (typeKeys) just types one letter at a time!

## Cucumber BDD-based interview questions and author's views

**Feature: Display Customer Details clearly**

**As a user**

**I want to see the customer profile details**

**So that I can read and understand about personal details of customer**

**Read the above piece of Cucumber BDD (behavior-driven development) code and say what's wrong?**

**Answer:** Aimless Feature File!

The above gherkin code is not aiming to one set of requirement. It has to be rewritten as:

**Feature: Identify a customer**

In order to provide solution to customer queries

As a customer service agent

I want to access relevant customer profile page

**Justification:**

In order to provide solution to customer queries (business goal)

As a customer service agent(context)

I want to access relevant customer profile page(capability)

Reference slides to learn more:

<https://www.slideshare.net/wakaleo/bdd-antipatterns-59940406>

**Read the piece of feature file below:**

**Given Landrover page is open in InternetExplorer**

**Respective step Definition in Java Programming:**

```
@Given("Landrover page is open in InternetExplorer")
public void Landrover_page_is_open_in_InternetExplorer()
{
    browser=new ChromeDriver();
    browser.get("http://www.landrover.co.uk/");
}
```

**What is wrong in the step definition above?**

Answer: The syntax of @Given is incorrect:

```
@Given("Landrover page is open in InternetExplorer")
public void Landrover_page_is_open_in_InternetExplorer()
{
    browser=new ChromeDriver();
    browser.get("http://www.landrover.co.uk/");
}
```

Expected Step Definition is:

```
@Given("^Landrover page is open in InternetExplorer$")
public void Landrover_page_is_open_in_InternetExplorer()
{
    browser=new ChromeDriver();
    browser.get("http://www.landrover.co.uk/");
}
```

**Read the below feature file to understand next question:**

**Feature: Identify a customer**

**In order to provide solution to customer queries**

**As a customer service agent**

**I want to access relevant customer profile page**

**Scenario Outline: Agent Login**

**Given I am a Registered Agent**

**When I login as <username> with password <pwd>**

**Then I should not allowed to login**

**And I should see an error message <error>**

**Examples:**

username	pwd	error
abcdefge		error
abcdefge	pwd	failure login
abcdefge	pwd	failure login

## Read the above piece of Cucumber BDD (behavior-driven development) code and say what's wrong?

Answer:

Its having UX concerns on the Examples (data table) and vague error messages and incorrect format of the data table (third data row missing '|') as below:

```

Feature: Identify a customer
In order to provide solution to customer queries
As a customer service agent
I want to access relevant customer profile page
Scenario Outline: Agent Login
Given I am a Registered Agent
When I login as <username> with password <pwd>
Then I should not allowed to login
And I shour see an error message <error>
Examples:
|username          |pwd           |error
|abcdefge         |              |error
|abcdefge         |pwd           |failure login
|abcdefge         |pwd           |failure login
|              |              |

```

It can be rewritten as the below:

```

Feature: Invalid Login of a registered customer service agent
In order to provide solution to customer queries
As a customer service agent
I want to access relevant customer profile page
Scenario Outline: Customer Service Agent Invalid Login
Given I am a Registered Customer Service Agent
When I login as <username> with password <pwd>
Then I should see an error message <error>

```

---

Examples:

username	pwd	error	
user1	password	please enter valid credentials	
user2	password	please click register if not registered yet	
user3	password	Password expired. Please reset the password	

## Selenium Webdriver 3.0-based interview questions and author's views

In this section, author answers to latest version of Selenium (3.0 onwards) only and Selenium RC/other versions less than 3 are not included

- What are the main components of Selenium Webdriver 3.0?

It consists of libraries to automate web-based applications to write code in Java, javascript, C#, Python, Ruby, Perl and PHP programming languages.

Understand the entire code base at GitHubpage:

<https://github.com/SeleniumHQ/selenium>

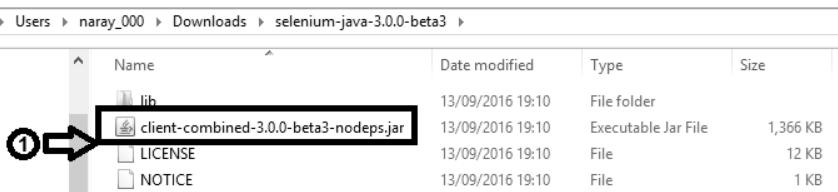
- Do I need Selenium server to run Selenium Webdriver 3.0 tests?

No. Selenium server was needed earlier to stand in between the application and the Selenium engine to inject the test instructions which is not needed in latest version.

I have downloaded Selenium Webdriver 3.0 and Client libraries, how do I start Selenium configuration to the java-based framework using Eclipse tool to script?

Navigate to <http://www.seleniumhq.org/download/> and go to the section 'Selenium Client & WebDriver Language Bindings' and click on download on the row where Java referenced.

Extract the zip file when downloading,use all the jar files referenced below to add as external jars:



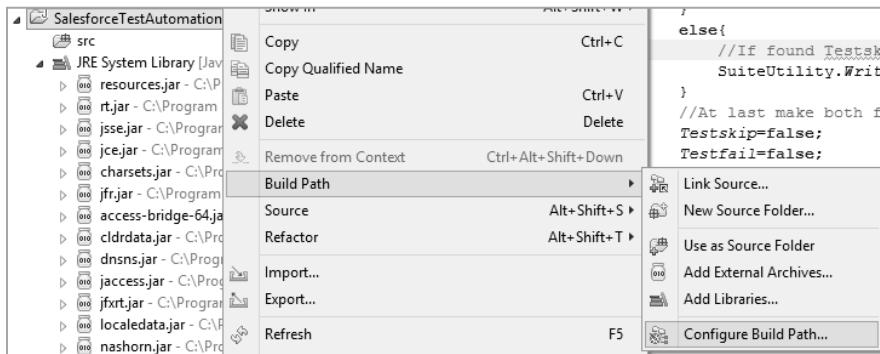
Users > naray_000 > Downloads > selenium-java-3.0.0-beta3 >				
	Name	Date modified	Type	Size
	lib	13/09/2016 19:10	File folder	
① →	client-combined-3.0.0-beta3-nodeps.jar	13/09/2016 19:10	Executable Jar File	1,366 KB
	LICENSE	13/09/2016 19:10	File	12 KB
	NOTICE	13/09/2016 19:10	File	1 KB



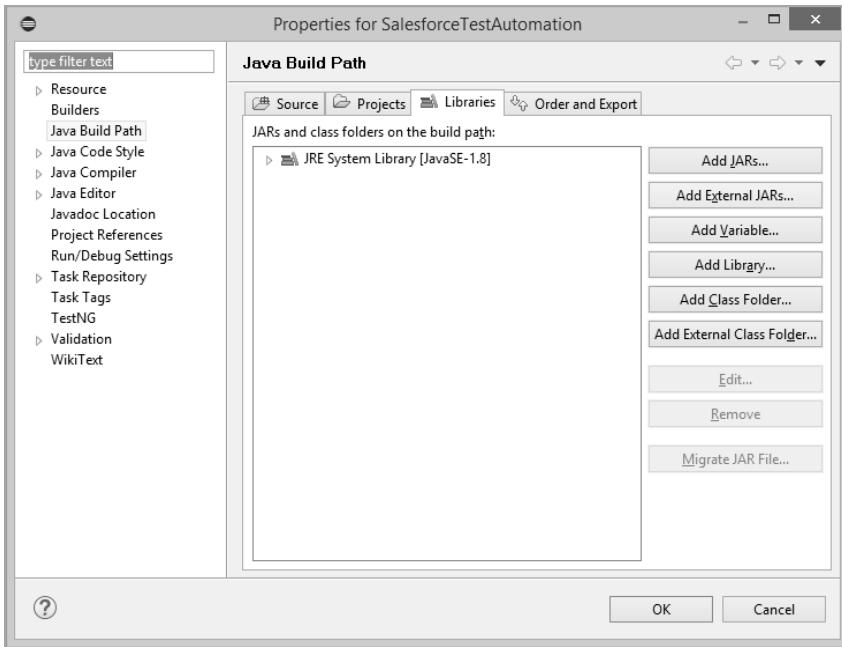
Name	Date modified	Type	Size
cglib-nodep-3.2.4.jar	13/09/2016 19:10	Executable Jar File	343 KB
commons-codec-1.10.jar	13/09/2016 19:10	Executable Jar File	278 KB
commons-exec-1.3.jar	13/09/2016 19:10	Executable Jar File	54 KB
commons-logging-1.2.jar	13/09/2016 19:10	Executable Jar File	61 KB
gson-2.3.1.jar	13/09/2016 19:10	Executable Jar File	206 KB
guava-19.0.jar	13/09/2016 19:10	Executable Jar File	2,255 KB
hamcrest-core-1.3.jar	13/09/2016 19:10	Executable Jar File	44 KB
hamcrest-library-1.3.jar	13/09/2016 19:10	Executable Jar File	52 KB
httpclient-4.5.2.jar	13/09/2016 19:10	Executable Jar File	720 KB
httpcore-4.4.4.jar	13/09/2016 19:10	Executable Jar File	320 KB
httpmime-4.5.2.jar	13/09/2016 19:10	Executable Jar File	41 KB
jna-4.1.0.jar	13/09/2016 19:10	Executable Jar File	894 KB
jna-platform-4.1.0.jar	13/09/2016 19:10	Executable Jar File	1,431 KB
junit-4.12.jar	13/09/2016 19:10	Executable Jar File	308 KB
netty-3.5.7.Final.jar	13/09/2016 19:10	Executable Jar File	1,081 KB
phantomjsdriver-1.3.0.jar	13/09/2016 19:10	Executable Jar File	12 KB

**Note:** All these jar files are to be included, not just one jar file displayed in Section1.

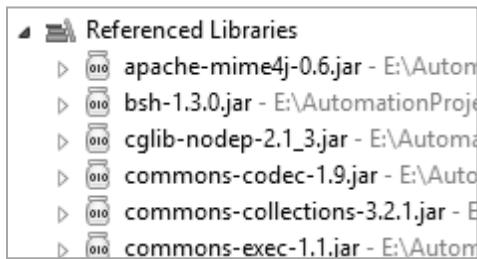
Navigate to Configure Build Path:



Click on Add External Jars:



Add relevant JAR files (refer all the jar files from Section 1 and 2 and include all of the here) which will be listed in the Project in Package Explorer under the section 'Referenced Libraries' once added:



- Is it possible to write Selenium Webdriver code using javascript and BDD in Linux Platform?

*Most of the Selenium Webdriver books explain about theory or framework with java/.net possibilities; but only few blogs are actually available about Linux-based Selenium Implementation; it is very important to know how an automation framework can be*

*design in operating systems other then Windows versions. This section describes more on javascript-based Selenium Framework to work on Chrome Browser and Linux platforms.*

Reference: <https://github.com/Matt-B/cucumber-js-selenium-webdriver-example>

Where you will find this file in framework: cucumber-js-selenium-webdriver-example-master\cucumber-js-selenium-webdriver-example-master\features

File Name: google.feature

The below scripts are written by understanding the core concepts given in the Git project in reference section above. So download the project from Git and replace feature file and step definitions with the scripts given in this section:

Name	Date modified	Type	Size
step_definitions	08/08/2016 19:54	File folder	
support	08/08/2016 19:54	File folder	
candidateverificationform.feature	08/08/2016 19:54	Gherkin Specificat...	1 KB

### **Feature File:**

Feature: Candidate Verification Form

As an internet user

In order to find out job in SAP

I want to be able to submit the candidate verification form

Scenario: Successful submission of Candidate Information

When I Launch HR Candidate Verification Form

Then I should enter details to submit the Form

## Step Definition File:

features > step_definitions	
Name	Date modified
 candidateverificationform-steps.js	08/08/2016 19:54

### Step Definitions in Javascript Programming Language:

```
'use strict';
var expect = require('chai').expect;
module.exports = function() {
this.World = require('../support/world.js').World;

this.When(/^I Launch HR Candidate Verification Form$/, function (searchQuery) {
this.driver.get('https://goo.gl/J4Wgrq');
});

this.Then(/^I should enter details to submit the Form$/, function () {
this.driver.findElement({ name: 'entry_1000000' })
.sendKeys("Albert Einstein");
this.driver.findElement({ name: 'entry_1000013' })
.sendKeys("albert@einstein.co.uk");
this.driver.findElement({ name: 'entry_1000014' })
.sendKeys("07867563478");
this.driver.findElement({ name: "group_1000001_1" })
.click();
this.driver.findElement({ name: 'entry_1000002' })
.sendKeys("2");
this.driver.findElement({ name: 'entry_1000003' })
.sendKeys("1.5");
this.driver.findElement({ name: 'entry_1000004' })
.sendKeys("1");
this.driver.findElement({ name: 'entry_1000005' })
```

```
.sendKeys("1");
this.driver.findElement({ name: 'ss-submit'})
.click();
});

};
```

### Explanation on the feature file:

Due the “When” condition following page getting launched in Chrome browser:

**HR Candidate Verification Form**

Overall 6 to 8 Years of Experience in SAP Basis, Netweaver, Kernel upgrade  
 -Experience in SAP Basis & N/W Administration  
 -Expertise in Installing & supporting diversified SAP databases includes Oracle  
 -Experience in performing Oracle Database upgrades  
 -Preference: Kernel upgrade - Update to latest support pack level for both ABAP and JAVA and add-ons

Company: Confidential  
 Location: Bangalore

\* Required

Full Name \*

Email ID \*

Contact Number \*

Total IT Experience \*

6 to 8 years of experience in IT  
 Less than 6 years of experience in IT  
 More than 8 years of experience in IT

Experience in SAP Basis

Experience in Netweaver

Experience in Kernel Upgrades

Current Company

**Submit**

Never submit passwords through Google Forms.

Powered by  
 Google Forms

This content is neither created nor endorsed by Google.  
[Report Abuse](#) · [Terms of Service](#) · [Additional Terms](#)

“Then” condition sends the user inputs to the page and click on submit button hence form is getting submitted.

Prerequisite for this project – Chrome Driver:

Chrome Driver can be installed from

<http://chromedriver.storage.googleapis.com/index.html>

Once installed, add the reference to the 'Path' in Linux box as follows (if the driver is placed in drivers folder within D drive),

```
export PATH=$PATH:~/D/Drivers/chromedriver
```

Validate if it is working as expected (by entering **chromedriver** in your terminal and validate the response like the below):

```
[naru@computer ~]$ chromedriver
```

*Starting ChromeDriver (v2.11) on port 9515*

*Only local connections are allowed.*

Press Ctrl+C to come out of the above message and navigate to the folder where the script is placed by entering cd <path> command;  
If the project folder is selenium-automation-framework-using-javascript, enter as:

```
cd selenium-automation-framework-using-javascript
```

Once navigated to the project folder, enter:

```
npm install
```

Install grunt (or gulp) which will be used for test running purpose (just like TestNG in Selenium Java projects)

### **Verify once grunt installed at node modules:**

```
node_modules/grunt-cli/bin/grunt
```

### **Use gruntfile.js to run the tests and produce results like the below:**

```
[naru@computer selenium-automation-framework-using-javascript]$ grunt
```

Running "jshint:files" (jshint) task

```
>> 4 files lint free.
```

Running "exec:run\_cucumber\_tests" (exec) task

Feature: Candidate Verification Form

As an internet user

In order to find out job in SAP

I want to be able to submit the candidate verification form

```
Scenario: Successful submission of Candidate Information      #
features/google.feature:6
When I Launch HR Candidate Verification Form      #
features/google.feature:7
Then I should enter details to submit the Form      #
features/google.feature:8
```

1 scenario (1 passed)

2 steps (2 passed)

Done, without errors.

- How do I launch the web page using Selenium Webdriver 3.0 in every possible programming language?

Use the Java code sample below to launch a web page:

```
driver.get('http://www.webdriverinselenium.com/');
```

Use the Javascript code sample below to launch a web page:

```
return driver.get('http://www.webdriverinselenium.com/');
```

Use the Ruby code sample below to launch a web page:

```
driver.navigate.to " http://www.webdriverinselenium.com/"
```

Use the Python code sample below to launch a web page:

```
driver.get("http://www.webdriverinselenium.com/")
```

Use the C# code sample below to launch a web page:

```
driver.Navigate().GoToUrl("http://www.webdriverinselenium.com/");
");
```

- Writing javascript-based framework of Selenium means writing some lines of javascript code within Java-based Selenium framework? I mean are they one and the same or different?

Both are totally different!

Writing a “Selenium framework using javascript” means, the project code base made out of javascript programming only and the tool used is called “webdriverio” [Selenium has been called webdriverio when used for javascript projects] and Node.js has been used in these kinds of javascript frameworks.

Writing a javascript piece of code within “Java-based Selenium Framework” means, some of the javascript capable functions of application such as image click actions (which are difficult when java programming used and easy when javascript programming being used) has been taken care like the below:

```
WebElement element =  
driver.findElement(By.id("PhotoImageClickable"));  
JavascriptExecutor js = (JavascriptExecutor)driver;  
js.executeScript("arguments[0].click()", element);
```

- My framework is not made of Java and I am coding in different programming language for Selenium. Do I still need to install Java in my computer?

Yes. Whenever the java programs compiled and executed in computer(such as Selenium), required JAVA files has to be referred during compilation and execution. So the computer need to know where the JAVA files located and what is the respective PATH of bin folder. So the file location is provided through JAVA\_HOME and bin folder location is provided through PATH set up. Once it is completed, it is a good practice to restart computer and check the version of JAVA through Command Prompt. Simply running the command JAVA-VERSION will provide the java version in command prompt which proves that the JAVA installation has been completed successfully. Alternative way to set up JAVA\_HOME and PATH is through Command Prompt. This can be done by launching Command Prompt as follows.

Step1: Press Windows+R (to open RUN)

Step2: Type CMD,Press Enter

Step3: In Command Prompt, Type“SET JAVA\_HOME = c:/Program Files/Java”

Step4: Type “SET PATH = %PATH%;%JAVA\_HOME%\bin”

- What are the most frequently used browsers for Selenium Tests in the IT industry?

Firefox, Chrome and Internet Explorer(IE)

Note:

IE 11 need custom configurations to get it worked. Please refer:  
<https://github.com/SeleniumHQ/selenium/wiki/InternetExplorerDriver>

- Which are the browsers I can run the Selenium Webdriver 3.0 tests?

Firefox [Need a Gecko Driver to run tests. Refer:

<https://github.com/mozilla/geckodriver/>]

Chrome [Refer:

<https://sites.google.com/a/chromium.org/chromedriver/>]

Internet Explorer

[Refer:<https://github.com/SeleniumHQ/selenium/wiki/InternetExplorerDriver>]

Opera

Microsoft Edge

Ghost

Safari

- How to use Recovery Scenarios in Selenium scripts similar to HP UFT (formerly QTP)?

It can be easily managed using exception handlers and looping mechanism of respective programming language used for Selenium Webdriver!

If Java used to construct the framework for login page, following sample try catch mechanism can be used to include recovery scenario within Selenium:

```
try
```

```
{
```

```
browser.findElement(By.name("loginID")).sendKeys("narayanan.  
p");
```

```
browser.findElement(By.name("pwd")).sendKeys("abcde");
browser.findElement(By.name("signinButton")).click();
}
catch (Exception errText)
{
errText.printStackTrace();
}
```

## Selenium Grid 2 interview questions and author's views

- **What can be done to reduce the selenium test execution time when the tests are to run against three browsers and five different operating systems?**

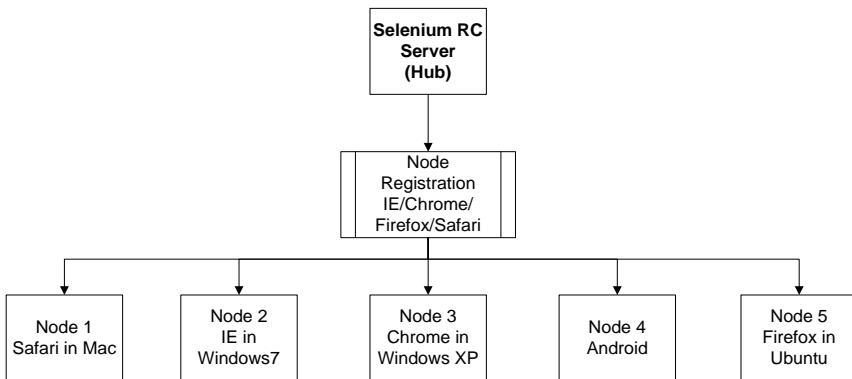
Selenium Grid is the best option to reduce execution time and increase concurrency of test execution across compatibility tests.

- **How does Selenium Grid perform concurrent tests?**

Selenium Grid runs in hub and node model using Selenium Remote Control Server. When server started for hub, nodes have to be registered with different browser combinations for the tests.

When test getting executed, hub (Selenium RC Server) look for open node to run the test in respective configuration and get such Selenium RC slave server to run the tests. Once test gets completed, automatically the node gets that allocated to next available test.

- Write a diagram to describe Selenium grid functionality?



- What jar file required to Selenium Grid?

Latest Selenium Standalone Server(selenium-server-standalone-3.x.x.jar) from:<http://www.seleniumhq.org/download/>

- How do I start Selenium Grid hub from ant tool?

Setup “ant” build configuration in the framework and when the configuration works in ant, it has to launch the hub of Selenium Grid as,

`ant launch-hub`

Once launched verify the webpage to check hub status

<http://localhost:4444/console>

- How do I start Selenium Grid node from ant tool?

Setup “ant” build configuration in the framework and when the configuration works in ant, it has to launch the node of Selenium Grid as,

`ant launch-remote-control`

Once launched verify the webpage to check number of nodes registered,

<http://localhost:4444/console>

- **How to get Selenium Grid slave customized to run in specific port?**

Use the command -Dport:

```
ant -Dport=8989 launch-remote-control
```

- **How to get the nodes running in virtual machines or different computers to talk to Hub in Selenium Grid?**

Follow the configuration startup while registering the node to hub in the beginning:

```
ant -Dport= -Dhost= -DhubURL= launch-remote-control
```

Details:

ant :Configured build tool

Dport: Port where the nodes setup to run

Dhost Hub which is started with Selenium RC

DhubURL= Respective hub URL such

as http://localhost:4444/console

- **How to tell to Selenium Grid on the test environment in commands?**

Use -Denvironment command:

```
ant -Denvironment="Chrome on Windows" launch-remote-control
```

- **How do I use particular desktop specific configuration in Selenium Grid test execution?**

Selenium grid uses a file “grid\_configuration.yml“ in which the specification of compatibility tests such as “IE on WinXP“ to be provided by tester. Once provided, specify the custom configuration on the target test execution machine within the test script:

```
Selenium = new DefaultSelenium("localhost", 4444, **'IE on  
WinXP'**, 'http://webdriverinselenium.com');
```

After this line of code, every Selenium instance runs on the node, it runs using the above configuration.

- **How Selenium Grid-based hub talks to slaves?**

“remoteControlPollingIntervalInSeconds” has been set before launching the hub in “grid\_configuration.yml” and based on such poll interval, hub try to ping slaves and see their status. Once the slaves are not connecting to the network requests, the hub disconnects the node and gets the allocation to next tests.

- **What happens if there are increased number of tests and how complex it is to handle in Selenium Grid to wait for every node of tests?**

“sessionMaxIdleTimeInSeconds” in “grid\_configuration.yml” file has to be revised to make it possible amount of Wait time in order to run tests in test environment.

- **If the hub of Selenium Grid goes down during the test execution at nodes, how to handle it to free the nodes from Selenium Grid hub?**

While setting up the test, specify the polling time of RC (node) to check the status of hub at – “DhubPollerIntervalInSeconds.”

```
ant -DhubPollerIntervalInSeconds= launch-hub
```

If polling time crosses and hub is unresponsive, then the node deregisters from the hub.

- **Can Selenium Grid used for NFT performance tests?**

It is possible to run Selenium Grid as a performance test when there are required hardware and software setup available within test environment. But there is no possibility of getting detailed reporting charts in this tool to display NFT reports such as throughput time analyzer, aggregator reports etc. So it is highly recommended to run the Selenium tests when JMeter is in recording mode to convert Selenium tests to JMeter-based performance tests and use them to setup NFT tests in JMeter (These steps are explained in Part2 of this book).

- **How to validate logs during the Selenium Grid tests?**

Hub logs: “log/hub.log”

Remote Control logs: “log/rc-\* .log” folder.

- **When Selenium Grid tests run in organizations, proxy setup blocks the tests and how to specify that in the code?**

Use the commands such as “http.proxyHost” abd “http.proxyPort” during the start of Selenium RC server:

Starting Hub:

```
ant -Dhttp.proxyHost= -Dhttp.proxyPort= launch-hub
```

Starting Slave:

```
ant -Dhttp.proxyHost= -Dhttp.proxyPort= launch-remote-control
```

- How Selenium Grid can be used in java based frameworks?

TestNG helps in running the tests of Selenium(Java framework tests) using Selenium Grid-based Hub and Nodes

- **What is the perfect test report used for Selenium Grid tests?**

*Cucumber Report* or *HTML Reports* are used in Javascript frameworks of Selenium where as *TestNG,XSLT Reports* are wide used for Java-based Selenium frameworks; similarly *ReportUnit-based reporting* is compatible for C#-based Selenium framework when they run using NUnit.

# Selenium Cheat Sheet

Driver Initialization		Selenium Locators	Selenium Exceptions
Firefox Ref v.47 Part2-Page 58	System.setProperty("webdriver.gecko.driver", "PathTo\\geckodriver.exe");	Locating by ID Refer Part2-Page37	'Server Exception : sessionid should not be null' Refer Part2-Page160
Chrome Refer Part2-Page 59	System.setProperty("WebDriver.chrome.driver", System.getProperty("user.dir")+"\\BrowserDrivers\\chromedriver.exe");	Locating by Name Refer Part2-Page37/38	Replace Chrome Driver with Latest Version
Internet Explorer Refer Part2-Page 59	WebDriver driver = new ChromeDriver();	Locating by Xpath Refer Part2-Page37/38	ERROR: No sessionid provided Refer Part2-Page160
	System.setProperty("Webdriver.ie.driver", "PathTo\\IEDriverServer.exe");	Locating by Hyperlinks by Link Text Refer Part2-Page38/39	UnreachableBrowserException Refer Part2-Page160 Registry Key Changes Recommended
	WebDriver driver = new InternetExplorerDriver();	Locating by DOM Refer Part2-Page39	Illegal state exception using Selenium 3 Refer Part2-Page161 Get latest Firefox Gecko Driver
	System.setProperty("WebDriver.ie.driver", "PathTo\\IEDriverServer.exe");	Locating by CSS Refer Part2-Page40-43	
	WebDriver driver = new InternetExplorerDriver();		



Annotations		Practice Scripts (cont)	Practice Scripts (cont)
TestNG @BeforeSuite @AfterSuite @BeforeTest @AfterTest @BeforeGroups @AfterGroups @BeforeClass @AfterClass @BeforeMethod @AfterMethod		(Mouse)Click Refer Part2-Page52	System.out.print(pageTitle);
JUnit @After @AfterClass @Before @BeforeClass @Ignore @Test		Compare TextRefer Part2-Page64	Implicit WaitRefer Part2-Pag e69
		Disable a FieldRefer Part2-Page64	driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
		Enable a FieldRefer Part2-Page64	driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
		ScreenshotRe fer Part2-Page64	Explicit WaitRefer Part2-Pag e70
		Print the Title of the PageRefer Part2-Page68	wait.until(ExpectedConditions.textToBePresentInElementLocated(By.xpath("//div[@id='<Object ID>'], "Enter Your Personal Details"));
			Fluent WaitRefer Part2-Pag e70
			.withTimeout(20, SECONDS)
			SleepR Refer Part2-Pag e70

## **Book Chapter- Based References**

- Narayanan Palani. (2017, August 02). Test Automation [Web log post]. Retrieved from <https://github.com/narayananpalani/testautomation>.
- Selenium. (2017, August 02). Selenium Official Website [Web log post]. Retrieved from <http://www.seleniumhq.org/>.
- Aravind. (2016, September 07). Software Testing Tutorials and Automation [Web log post]. Retrieved from <http://www.software-testing-tutorials-automation.com/p/selenium-webdriver.html>.
- Selenium. (2016, September 07). Selenium [Web log post]. Retrieved from <https://code.google.com/p/selenium/wiki/GettingStarted>.
- Specflow. (2016, September 07). Specflow [Web log post]. Retrieved from <https://github.com/techtalk/SpecFlow/wiki/Unit-test-providers>.
- Specflow.org. (2016, September 07). Specflow.org [Web log post]. Retrieved from <http://www.specflow.org/getting-started/#InstallSetup>.
- PhantomJS. (2016, September 07). PhantomJS [Web log post]. Retrieved from <http://phantomjs.org/download.html>.
- Stackoverflow. (2016, September 07). How Schedule Build in Jenkins [Web log post]. Retrieved from <http://stackoverflow.com/questions/7000251/how-schedule-build-in-jenkins>.
- RelevantCodes. (2016, September 07). RelevantCodes [Web log post]. Retrieved from <http://reportunit.relevantcodes.com/>.
- IBM. (2016, September 07). IBM Functional [Web log post]. Retrieved from <http://www-03.ibm.com/software/products/en/functional>.
- HP. (2016, September 07). HP UFT [Web log post]. Retrieved from <http://www8.hp.com/uk/en/software-solutions/unified-functional-automated-testing/>.
- Borland. (2016, September 07). Silk Test [Web log post]. Retrieved from <http://www.borland.com/en-GB/Products/Software-Testing/Automated-Testing/Silk-Test>.

- Groovy. (2016, September 07). Groovy [Web log post]. Retrieved from <http://www.groovy-lang.org/>.
  - RamonVictor. (2017, August 06). Protractor for AngularJS [Web log post]. Retrieved from <http://ramonvictor.github.io/protractor/slides/>.
  - Amit Tayade. (2017, August 06). Why Protractor? [Web log post]. Retrieved from [https://www.linkedin.com/pulse/why-protractor-amit-tayade?trk=v-feed&lipi=urn%3Ali%3Apage%3Ad\\_flagship3\\_search\\_srp\\_content%3BXI3Cb8BzO5Rzj5trB4csAA%3D%3D](https://www.linkedin.com/pulse/why-protractor-amit-tayade?trk=v-feed&lipi=urn%3Ali%3Apage%3Ad_flagship3_search_srp_content%3BXI3Cb8BzO5Rzj5trB4csAA%3D%3D).
  - Simon Stewart (2017September 6), Selenium 3 0 Retrieved from <https://www.eventbrite.co.uk/e/asos-qa-meetup-the-future-of-selenium-by-simon-stewart-tickets-36864574936>
- 

<sup>i</sup>NARAYANAN PALANI. (2016, September 14), Certified Member Profile[Web log post]. Retrieved from <http://www.ibm.com/certify/profile?palanin6>

<sup>ii</sup>UNICOM. (2016, September 14), India Testing Awards 2016[Web log post]. Retrieved from [http://www.unicomlearning.com/2016/India\\_Testing\\_Awards/index.html](http://www.unicomlearning.com/2016/India_Testing_Awards/index.html)

<sup>iii</sup>UNICOM. (2016, August 31), India Testing Awards 2017[Web log post]. Retrieved from

[http://www.unicomlearning.com/2017/India\\_Testing\\_Awards/](http://www.unicomlearning.com/2017/India_Testing_Awards/)

<sup>iv</sup>NARAYANAN PALANI. (2016, September 14), LinkedIn[Web log post]. Retrieved from <https://uk.linkedin.com/in/narayananpalani>

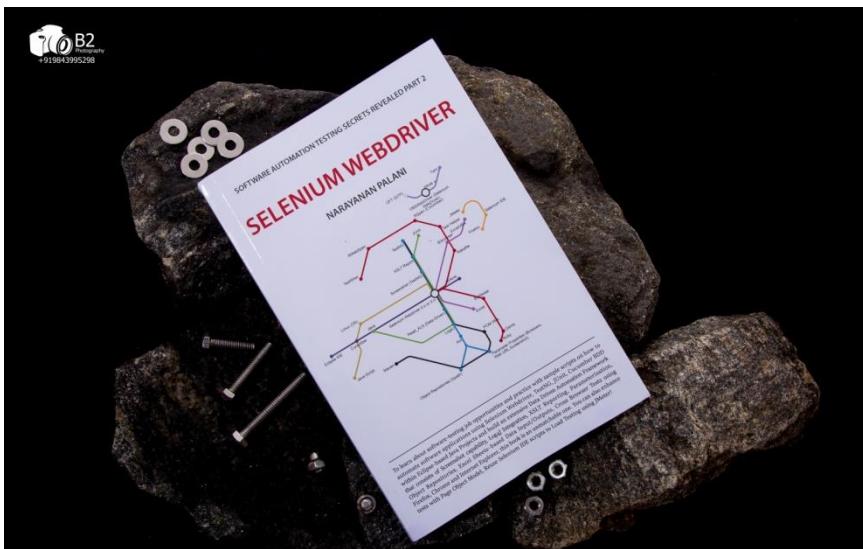
<sup>v</sup>DOCUMENTS.MX, (2016, September 14), ADCOM 2009 CONFERENCE PROCEEDINGS[Web log post]. Retrieved from <http://documents.mx/documents/adcom-2009-conference-proceedings.html>

# INDEX

activity list – 37, 200  
Agile Project - 1  
Analogues Test Estimation - 189  
assertElementPresent - 144  
assertions - 150  
asserts - 92, 108, 136  
assertText – 141, 144, 146  
assertTextNotPresent - 141, 382  
assessors – 5, 136, 138  
Automation Framework function point count - 193  
Automation Return on Investment 36, 198  
Automation Test Engine Installation - 196  
automation test project - 203  
Automation Test Scheduling – 37, 200  
Availability - 7  
BDD – 5, 33, 40  
Bottom-Up Test Estimation - 188  
C++ Programming - 28  
COCOMO – 36, 382  
COCOMO Calculation - 195  
Collective Test Estimation Technique 36, 196  
Cost Estimation - 187  
Cost of licenses - 198  
Crashing Compression Technique - 190  
Critical Chain Method - 190  
Critical Path Method (CPM) - 190  
Delphi Test Estimation Technique - 36, 197  
ETL tools 39, 270  
Excel sheet – 37, 221

Extensibility – 7, 145  
Fast Tracking Compression Technique - 191  
Five Phase of Test Automation – 37, 207  
functional regression - 206  
JAVA\_HOME – 49, 52, 357  
Key word driven testing - 206  
Macro Security - 222  
manual test analysts - 198  
Monte Carlo Simulation – 189, 192  
Network Diagram – 37, 203, 205  
NUnit - 21  
Parametric Test Estimation – 187, 189  
PERT - 36  
Portability - 8  
Priority - 201  
Program Evaluation and Review Techniques – 36, 181, 189  
Quick Test Professional - 180, 206  
Recording and Run - 138  
Regression Tests - 201  
Reliability 7, 270  
Resource Leveling on CPM - 190  
Saxon - 67  
Selenium IDE – 45, 39, 112, 141  
SOA - 38  
Stop and Start - 139  
Subroutines - 221  
System Utility command - 212  
Test Data Sheets – 192, 236  
Test effort - 187  
Test Environments - 187  
test execution scheduling - 202  
Test Schedule Compression Techniques – 36, 190  
Testing Process - 206  
Time Estimation - 189  
Toggle Breakpoint - 139

Top-Down Test Estimation - 188  
Total benefit derived from test automation - 199  
Total Cost of Test Automation - 199  
unadjusted function point count - 193  
UNIFIED FUNCTIONAL TESTING - 178  
Usability – 4, 196  
VB Script Function – 214, 228  
verifyElementPresent - 143  
verifyTable - 143  
Visual Basic Editor 37, 215, 222, 260  
XSLT reporting - 125, 69



**Readers can download sample eBook of Part2 at:**  
<https://www.educreation.in/store/sample/book309E.pdf>

**Readers can buy Part 2 at:**  
<https://www.educreation.in/store/selenium-webdriver-software-automation-testing-secrets-revealed-part2-narayanan-palani.html>

# **Software Automation Testing Secrets Revealed**

## **Revised Edition**

Part 1 Cucumber BDD, Selenium Webdriver,Protractor,Selenium Grid, Appium, TestNG,Jenkins,UFT,RFT,Visual Studio,Excel VBA,SOAP, Selenium IDE based Automation Testing

"I, as a professional automation tester, often fall in situations where we need to take call on what automation solution to employ for a project that would be most effective and efficient at the same time. But it is hard to take such decisions with limited insight and knowledge of the available tools for automated testing in the market today. The book here by Narayanan provides a panoramic view into these tools, thus empowering the reader to take more educated decisions to provide better automation test solutions."

**Sohag Satpati, Consultant, Capgemini**

"Automation testing is very important in every project testing life cycle. Through automation, we can achieve more test cases in less time. So it is both cost effective and also helpful to achieve the stringent time lines. This book will be an eye opener for job seekers in the above field, to achieve their goals and also to optimize their performance."

**Sunit Roy, Consultant, Capgemini**

"Automation testing is more than enough for a smart tester to sustain in smart world. This book enlightens the way to achieve it in the crazy labyrinth of test automation techniques."

**Sumayya Nej, Wipro Technologies**

"Great effort by the author! It is a step-by-step guide which aims at making sure anybody can do automation be it Selenium or any other tool. I would say this book is a wonderful package available in market right now, just go for it."

**Shweta Bhat, Wipro Technologies**



You may reach author at:

✉ narayananpalani@educreation.in

Also available as an eBook

**NON-FICTION**

ISBN 978-1-5457-0825-5



9 781545 708255 >



**EDUCREATION**

PUBLISHING (Delhi)  
www.educreation.in