IBM Rational Quality Manager Integration with

Selenium

By

V.Niranjan

[vniranjan72@outlook.com](mailto:vniranjan72@outlook.com)

Table of Contents

[Introduction 3](#_Toc474693851)

[Downloading and Installing Selenium 3](#_Toc474693852)

[Steps to install the client driver for Java 4](#_Toc474693853)

[Configuring Eclipse to hold Selenium test source code 5](#_Toc474693854)

[Record a sample Selenium test 10](#_Toc474693855)

[Exporting the Selenium Test in to the Eclipse Project 12](#_Toc474693856)

[Run the Test from within IBM Rational Quality Manager 15](#_Toc474693857)

[Configuring the Selenium Adapter for IBM Rational Quality Manager 17](#_Toc474693858)

[Start the Selenium Adapter 21](#_Toc474693859)

[Create and Run the Selenium Test Script in IBM Rational Quality Manager 22](#_Toc474693860)

[Create a Test Script in IBM Rational Quality Manager 22](#_Toc474693861)

[Run the Automated Selenium Test from IBM Rational Quality Manager 24](#_Toc474693862)

[Summary 27](#_Toc474693863)

# Introduction

Selenium is a very popular open source tool for web application testing. By creating regression tests from Selenium and linking it to IBM Rational Quality Manager helps to maintain complete traceability from requirements to development and to testing.

This article will show you how to integrate IBM Rational Quality Manager with Selenium Junit tests. Only a sample test is created in this article from the integration point of view. The integration plays a very important role from a DEVOPS perspective of Continuous testing where the Selenium tests can run automatically after every build.

Much like my previous article on running and integrating IBM Rational Functional Tester scripts from IBM Rational Quality Manager you can also run Selenium tests by integrating with IBM Rational Quality Manager and return the test results back to RQM.

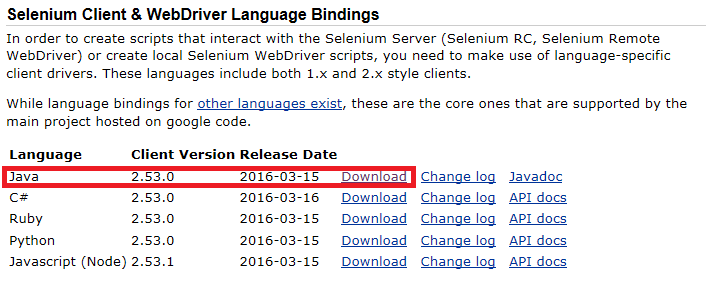
# Downloading and Installing Selenium

Selenium IDE is a Firefox plugin which will run from the Browser. To install the plugin there are many articles which guide you on the same. The following URL will guide you step by step for installing the plugin.

<http://www.guru99.com/install-selenuim-ide.html>

In order to create local Selenium WebDriver scripts, you need to make use of language-specific client drivers. Download the client specific driver for Java from

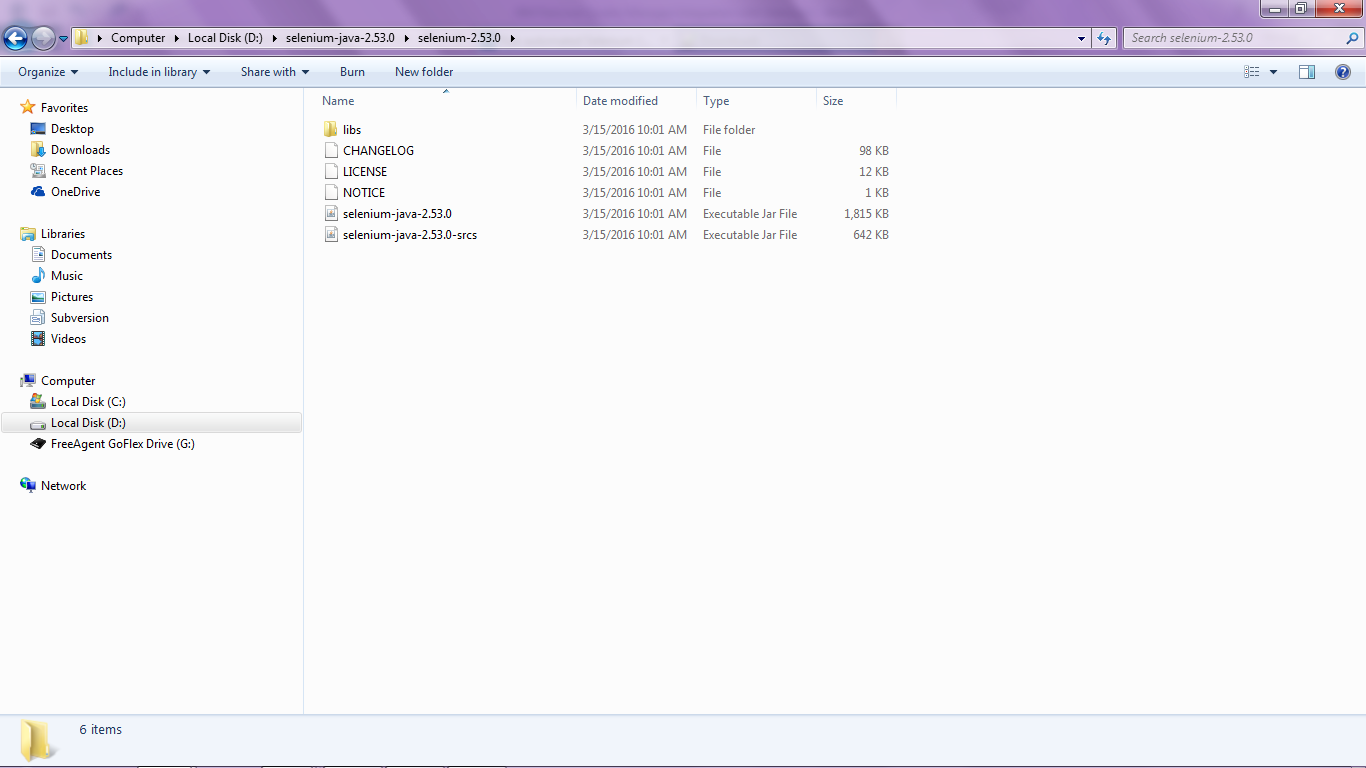
<http://docs.seleniumhq.org/download/>



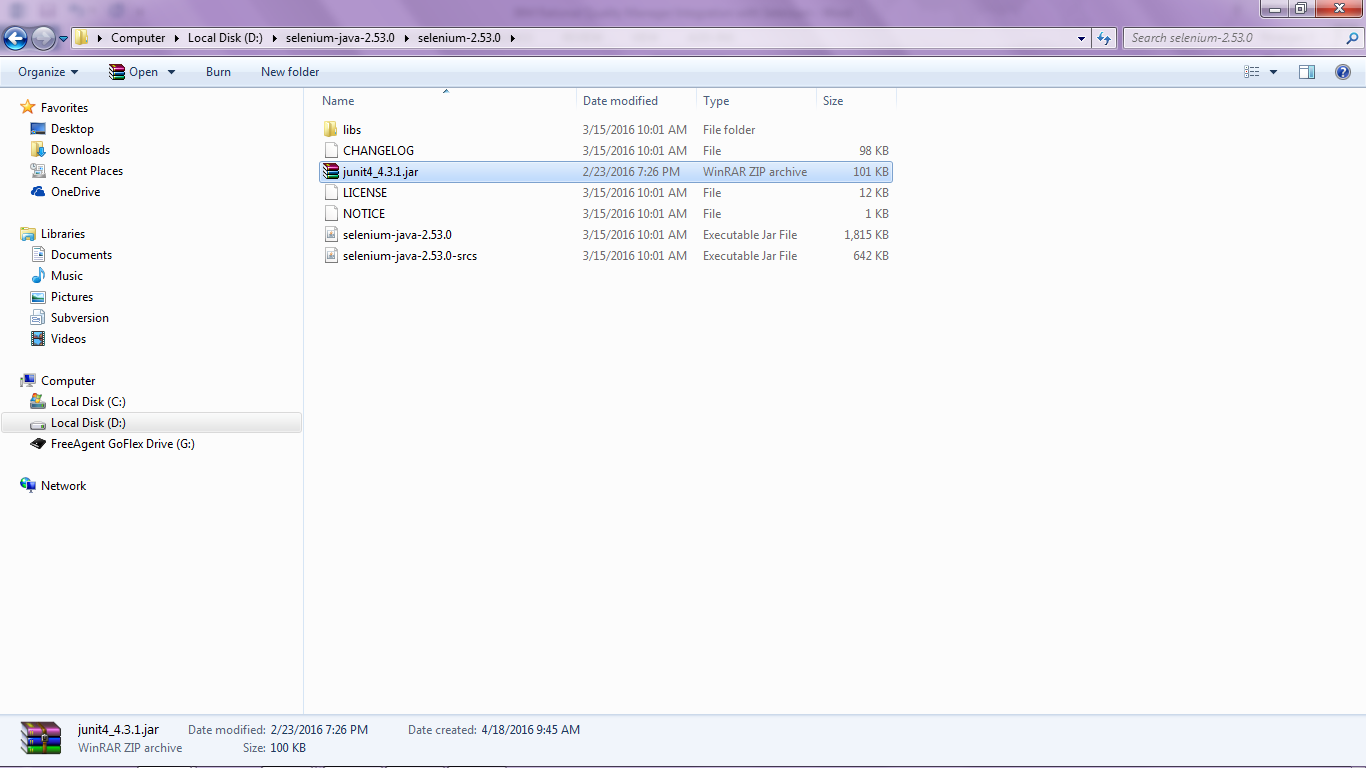
You could download any latest client specific driver for Java from the above URL.

## Steps to install the client driver for Java

1. After the ZIP file above is downloaded extract it to a folder



1. Download the Junit version 4 Java Archive (JAR) file from **junit.org** and save the file in the above location just extracted. Here the file name is **junit4\_4.3.1jar**



1. Above steps need to be repeated on all machines running Selenium tests

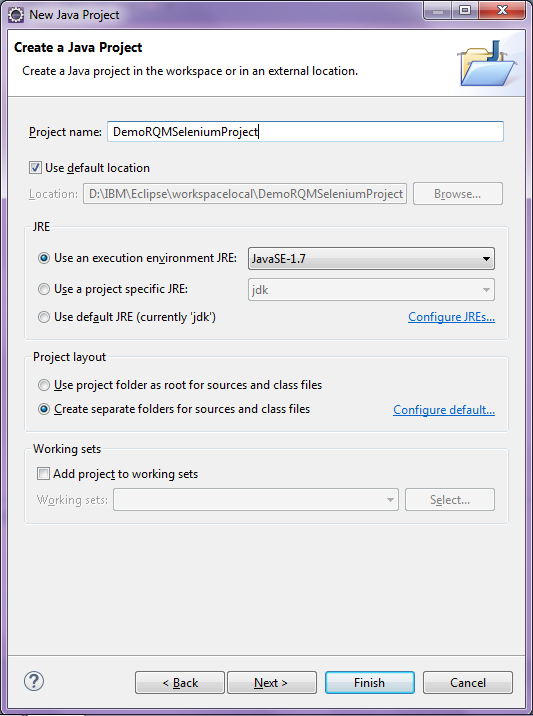
# Configuring Eclipse to hold Selenium test source code

The test source code which needs to be viewed or modified will need to be entered in to a java language based IDE. We will use Eclipse for modifying any selenium based tests.

One can download eclipse from <http://eclipse.org>

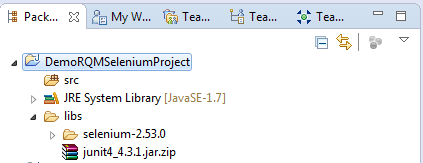
In this article I have used the Eclipse Luna version.

1. Create a new Java Project. Select **File🡪New🡪Java Project**
2. Enter a name such as **DemoRQMSeleniumProject**

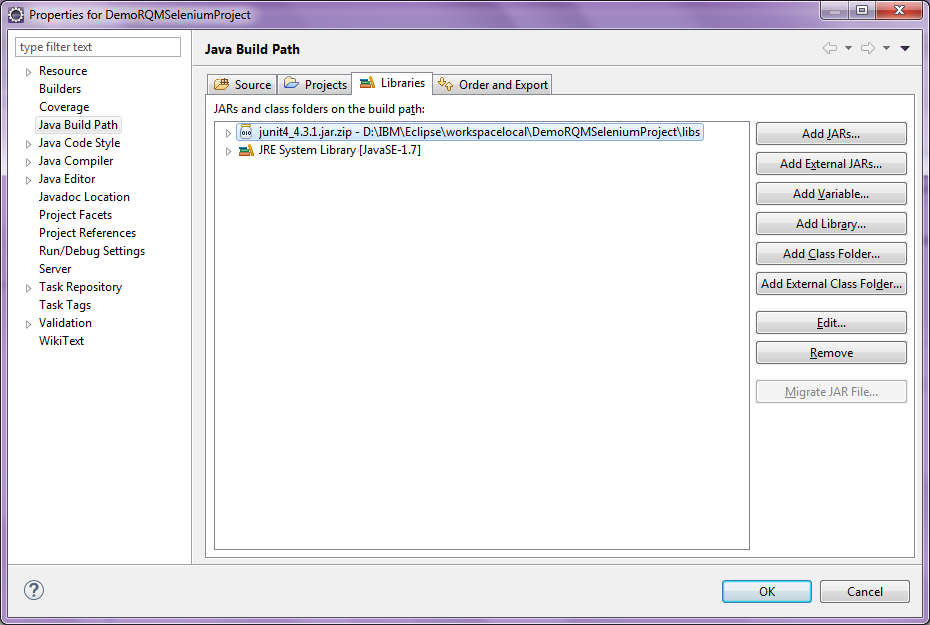


Click **Finish.** If asked to switch to Java Perspective then click on Ok.

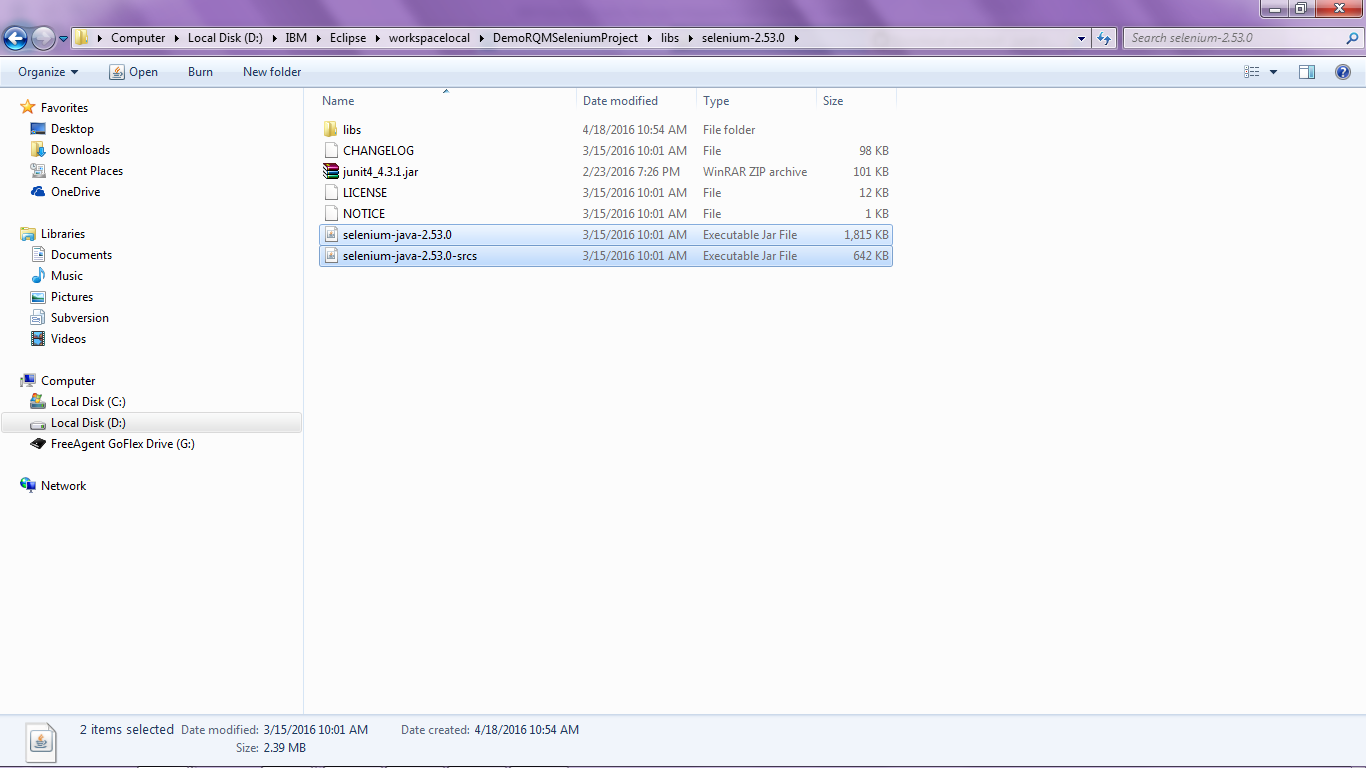
1. In the Package Explorer right click on the project and create a new folder called **libs**. Copy the **junit-4.jar** file and **Selenium-2.53.0** folder to the **libs** folder that is just created in the Eclipse project.

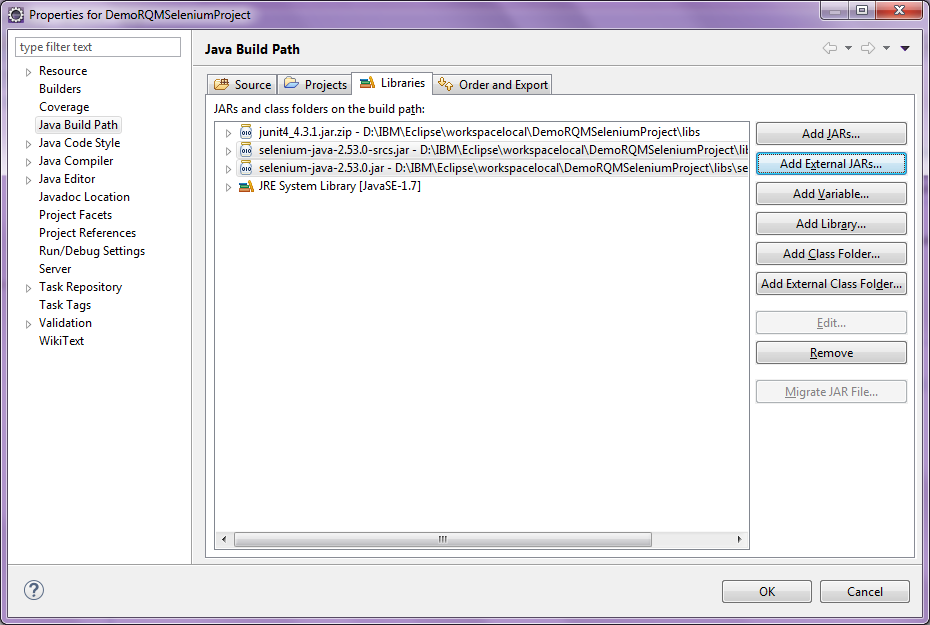


1. The JAR files will need to be added to the project class path, which will be used within the selenium tests. To do this right click on the **DemoRQMSeleniumProject** eclipse projectand select **Build Path > Configure Project Build Path**.
2. Go to the Libraries tab, and then click on **Add External JARs**. Choose the **junit4\_4.3.1.jar** file. Click OK.

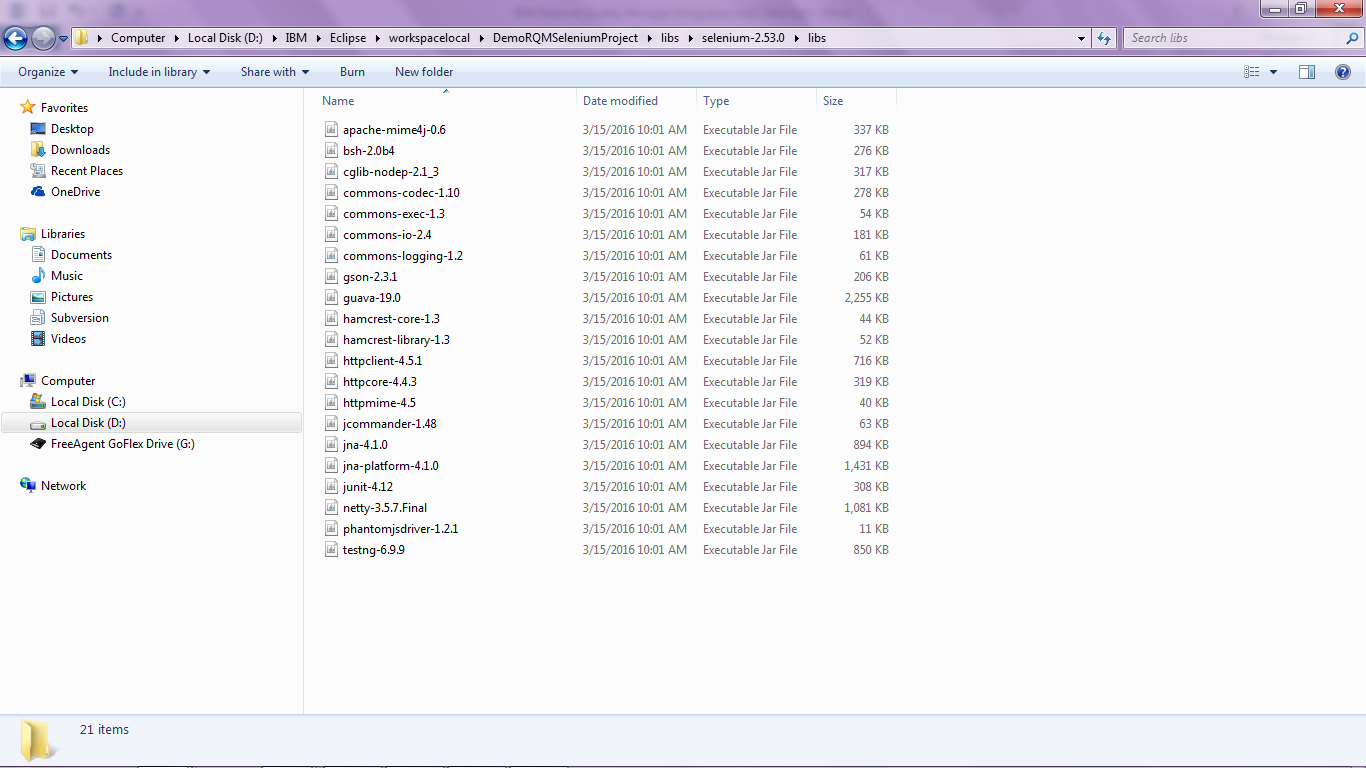


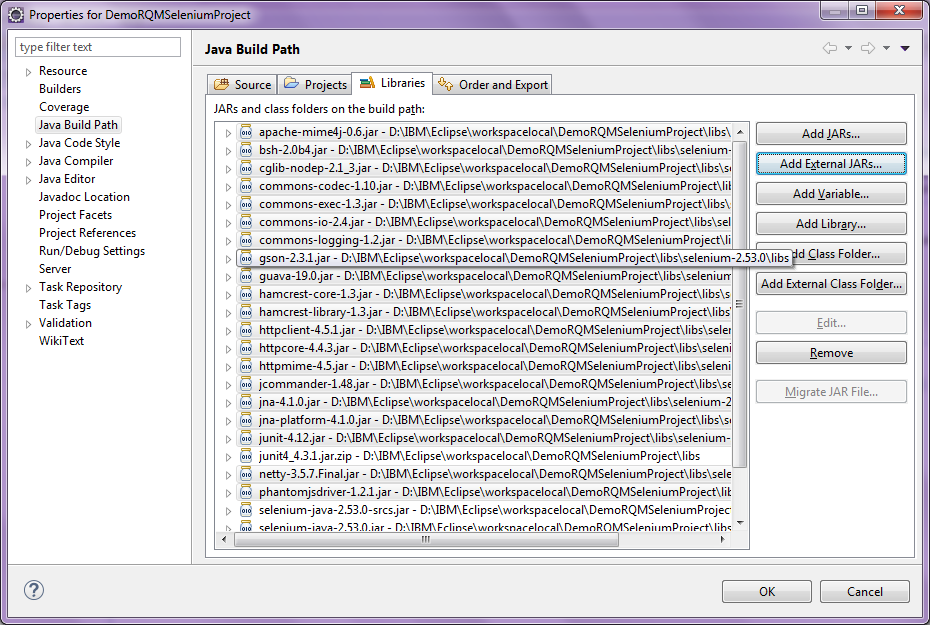
1. Also add the other 2 JAR files from the Selenium-2.53.0 directory (refer to section [“Steps to install Client Driver”](#_Steps_to_install) for this directory) to the following folder in Eclipse project referenced @ D:\IBM\Eclipse\workspacelocal\DemoRQMSeleniumProject\libs\selenium-2.53.0





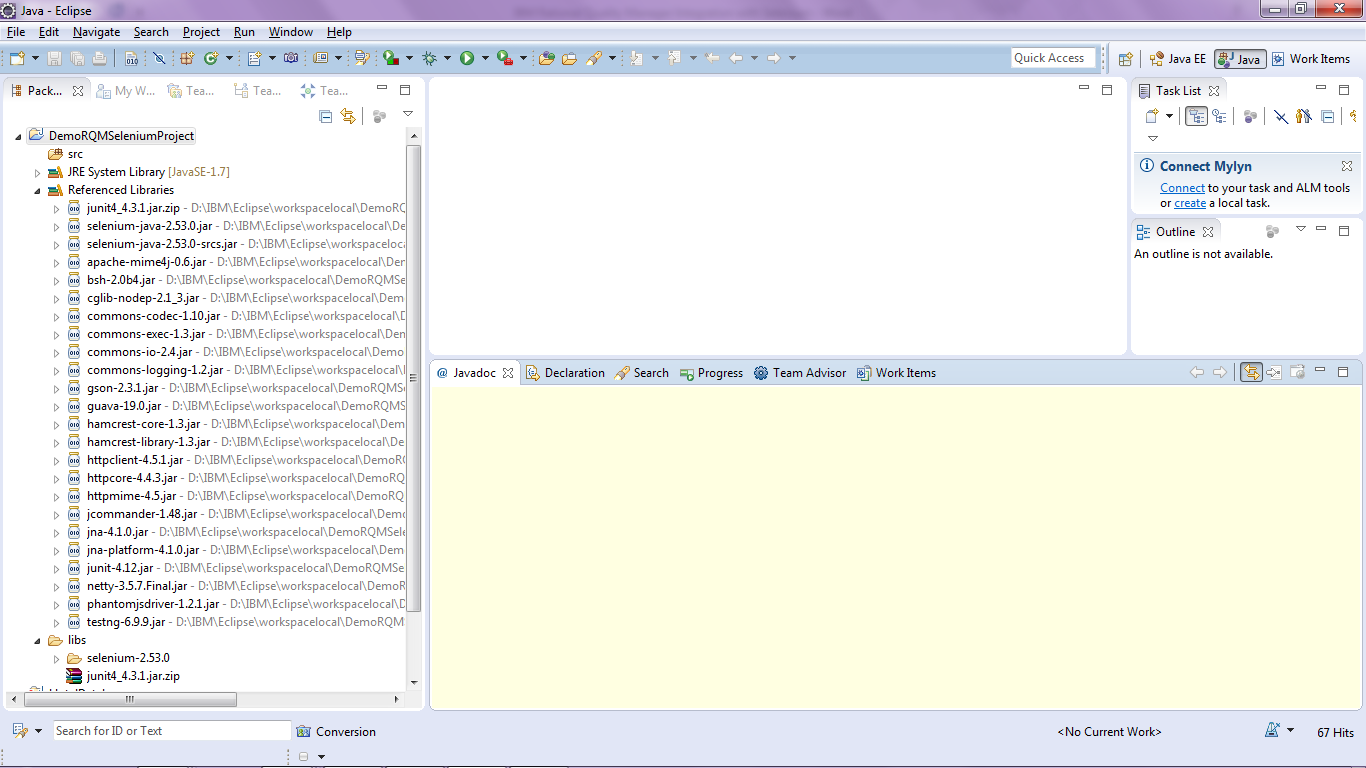
1. Also add all the JAR files from the Selenium-2.53.0/libs folder (refer to section [“Steps to install Client Driver”](#_Steps_to_install) for this directory) to the following folder in Eclipse project referenced @ D:\IBM\Eclipse\workspacelocal\DemoRQMSeleniumProject\libs\selenium-2.53.0\libs



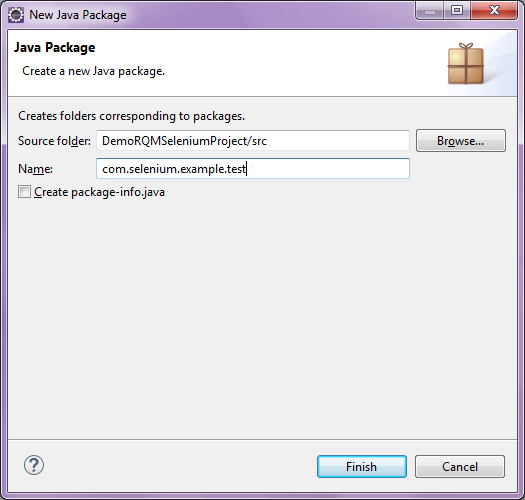


Click OK.

The final screen looks like the following in Eclipse.



1. As a developer you would normally create a package in any eclipse project to hold source code. In this case we will create a package to hold Selenium tests. Call it like **com.selenium.sample.test.** Right click on the **DemoRQMSeleniumProject** project and create a package.

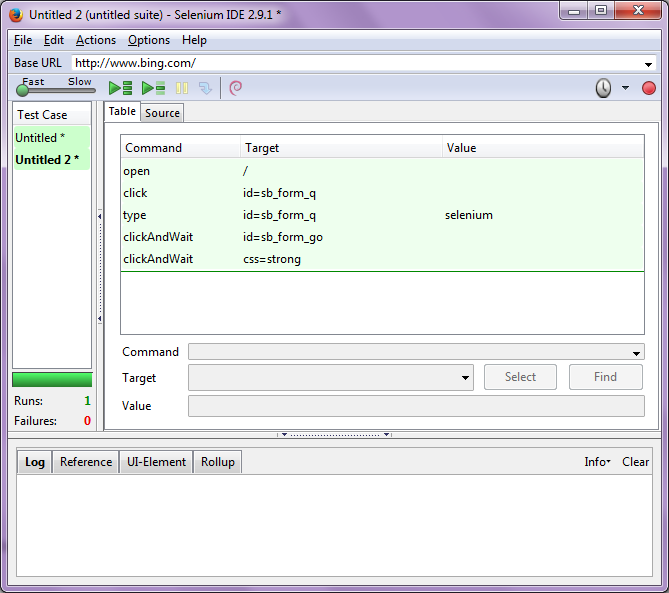


# Record a sample Selenium test

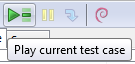
We will now record a sample Selenium test and import the same in to the project **DemoRQMSeleniumProject** created above. We will run a test to search a keyword ‘Selenium’ on [www.bing.com](http://www.bing.com)

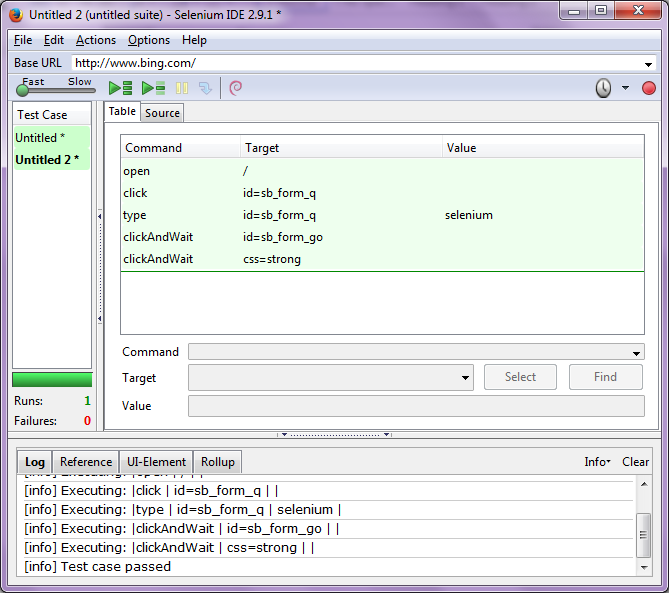
1. To open the Selenium in Firefox select **Web Developer > Selenium IDE**. Shortcut command **Ctrl+Alt+S** can also be used to open the IDE.
2. In the window that comes up, set the Base URL to **www.bing.com** as shown below and make sure that the red **Record** button is in record mode.
3. Open the URL **www.bing.com** in a new Firefox browser window. In the web page that has loaded, enter a search term such as **Selenium** and then click on the Search button.
4. Perform any activities in the search that is displayed.
5. Once finished return back to the Selenium IDE window which is opened as in step 1, and stop the recording by clicking the red **Record** button again.

The resulting Selenium IDE window should now look as shown in the figure below.



1. To playback the test just recorded click on **Play current test case** icon in the Selenium IDE

. 

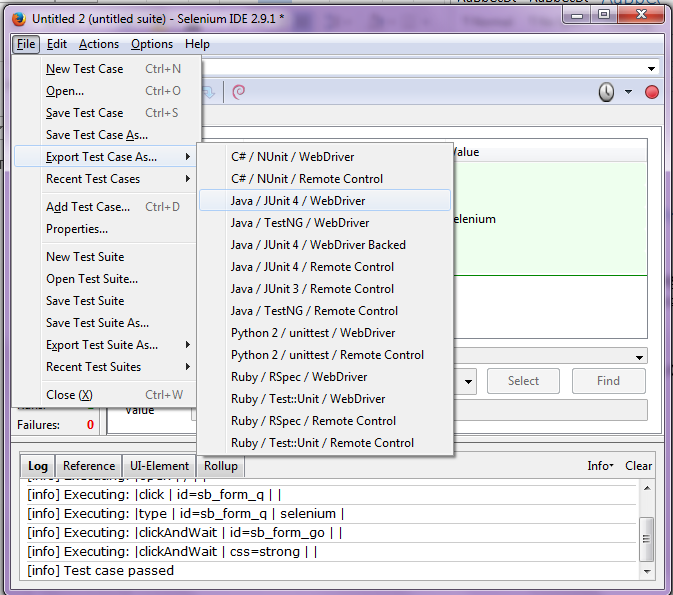


In the Log below you should see a message shown as Test Case Passed.

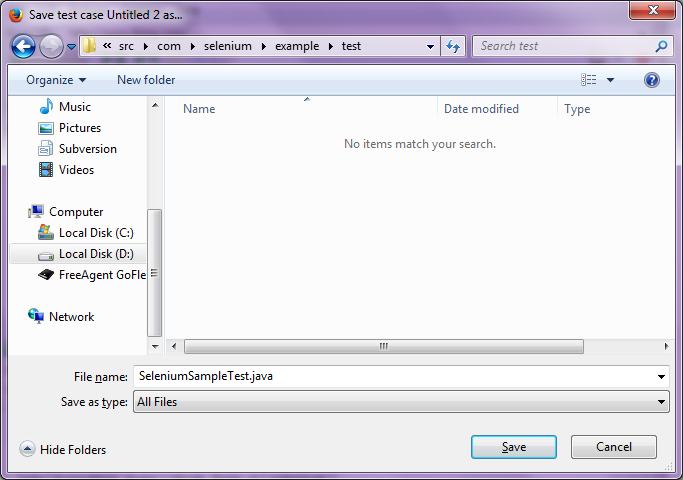
# Exporting the Selenium Test in to the Eclipse Project

To execute the selenium test just created from IBM Rational Quality Manager the test script just created must be in Java. The Selenium IDE has an export functionality which can export the test script in Java format required to execute from IBM RQM

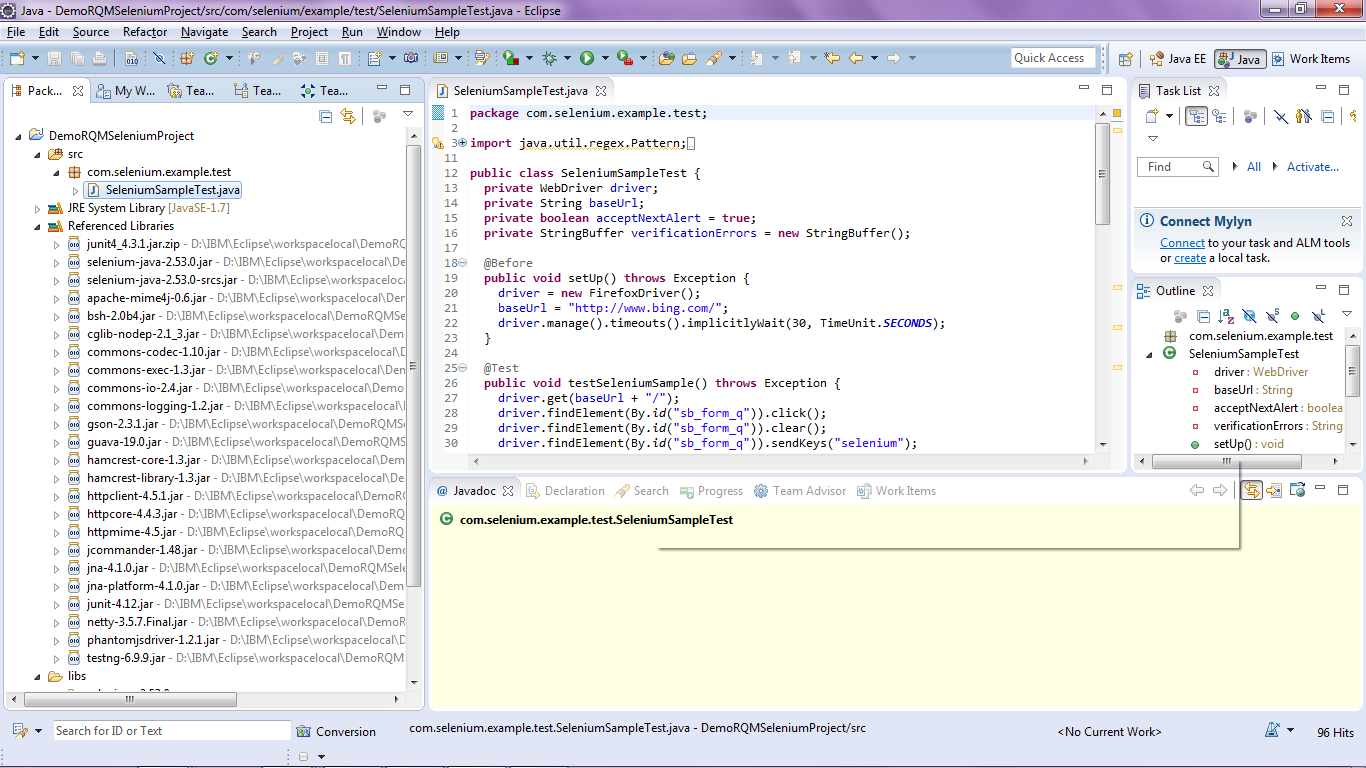
1. From the Selenium IDE which is open from the previous playback select **File > Export Test Case As > Java/JUnit4/WebDriver**



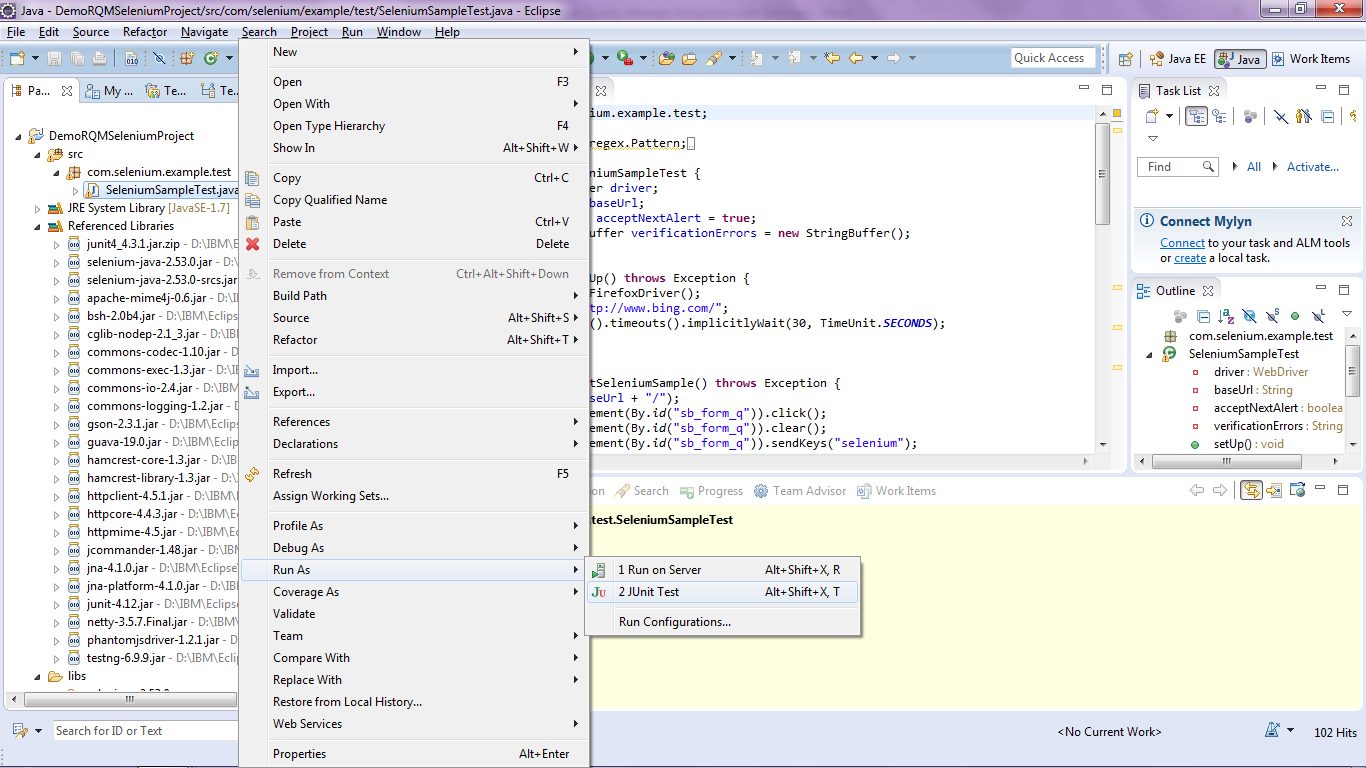
1. Give a name like **SeleniumSampleTest.java** and save it as a java file in the package created above which was **‘com.selenium.example.test’.** The Eclipse project location in this case would be D:\IBM\Eclipse\workspacelocal\DemoRQMSeleniumProject\src\com\selenium\example\test



1. View the Selenium Java test file exported in Eclipse. In case you are not able to view it refresh the Java project and you should be able to see the Java test file **SeleniumSampleTest.java** in the package **com.selenium.example.test**.



Run the file as JUNIT Test. Right click on the SeleniumSampleTest.Java file and select **Run As 🡪 Junit Test** as shown below.

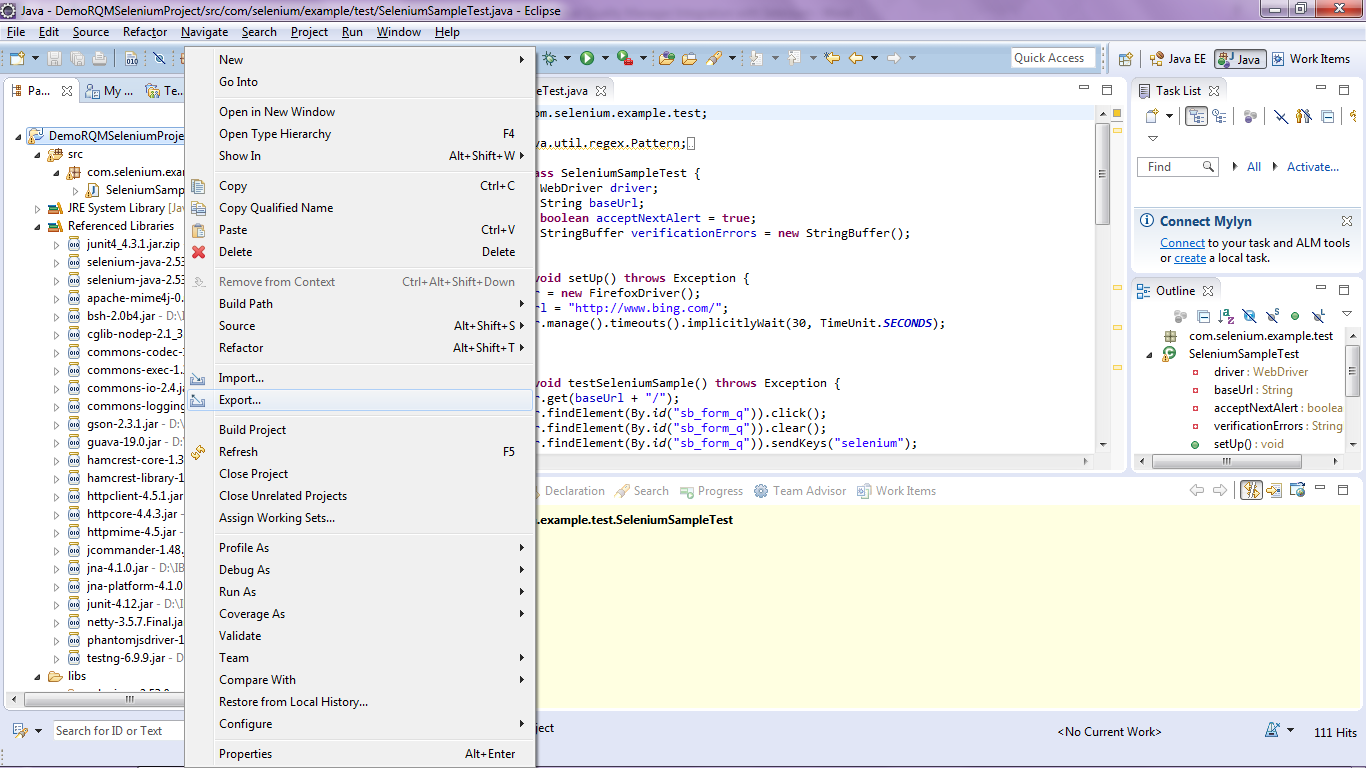


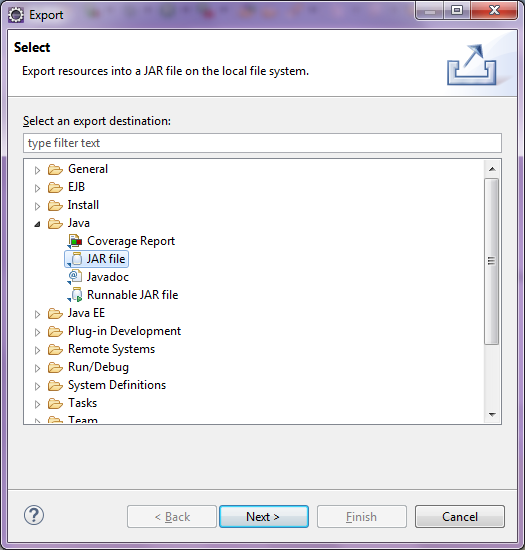
Verify playback works and same steps which you performed during recording worked.

# Run the Test from within IBM Rational Quality Manager

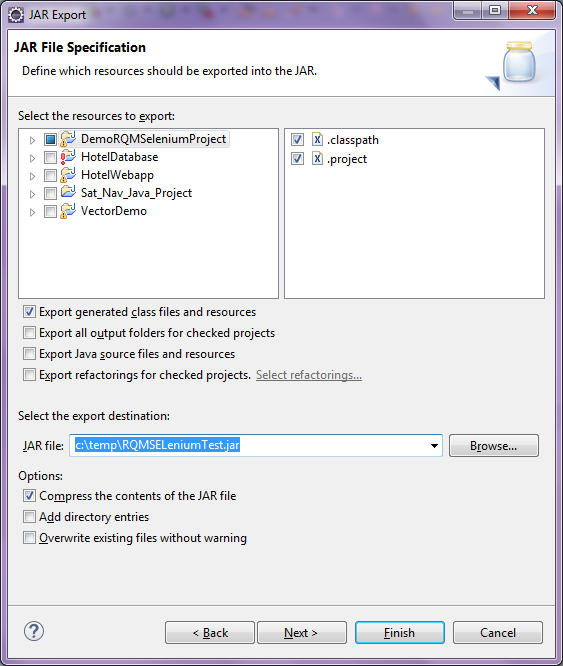
You have seen that the Selenium test can be run from within the Selenium IDE and as well as from Eclipse IDE. A test case in IBM Rational Quality Manager will need to be created and lined to the Selenium test script. For this integration to work Selenium test script will need to be made available in JAR file format.

1. Right Click on the eclipse **DemoRQMSeleniumProject** and select **Export.**
2. Expand Java and Select JAR file. Follow the steps as shown in the screenshots





1. Click **Next** and give a destination folder to store the JAR file.



Click **Finish** once done.

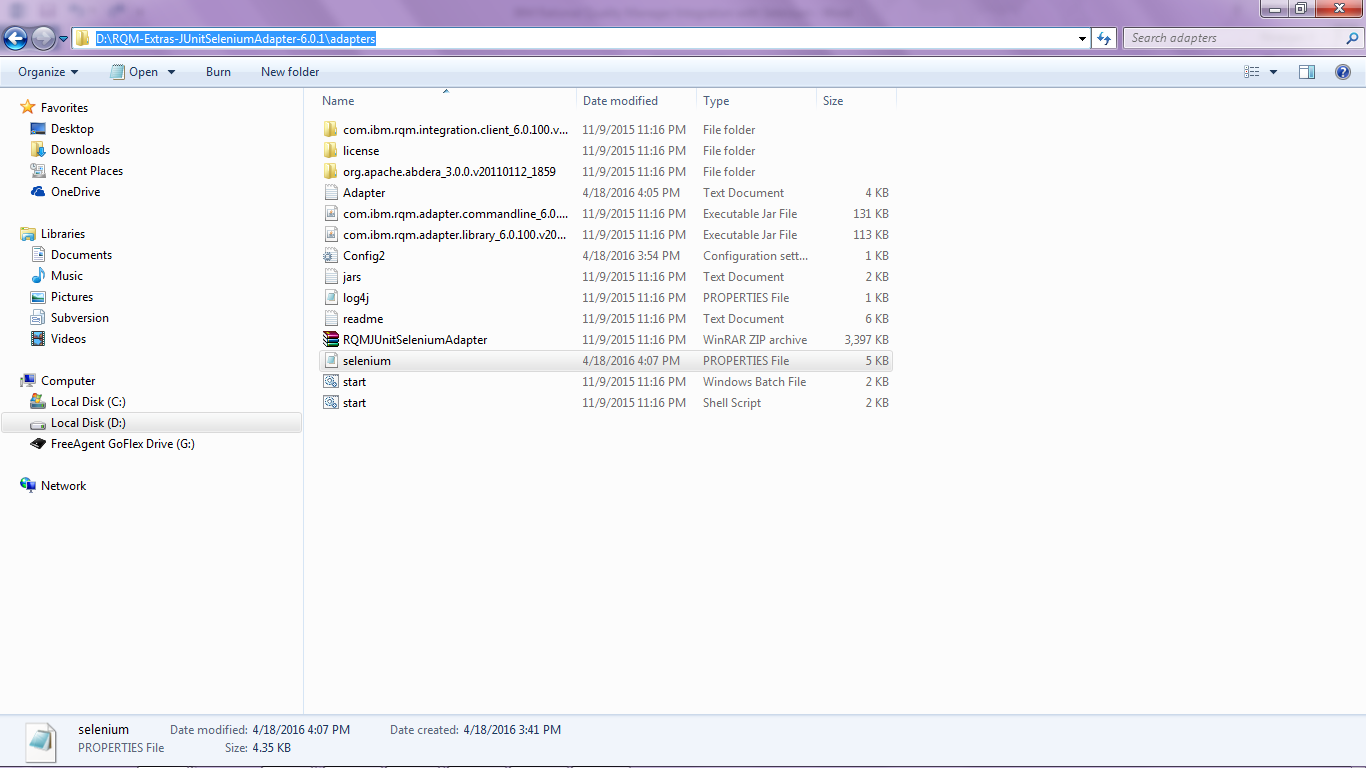
# Configuring the Selenium Adapter for IBM Rational Quality Manager

The Selenium adapter for IBM Rational Quality Manager is available for download from the following URL. The version of RQM used in this article is 6.0.1. If you use any other earlier version please check the download link for that version.

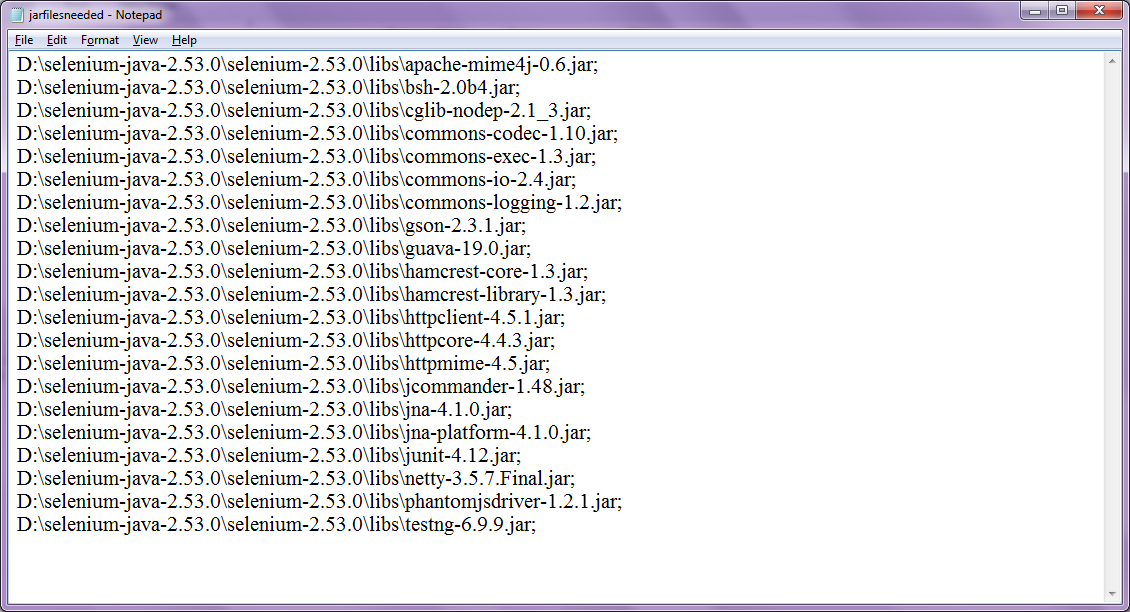
<https://jazz.net/downloads/rational-quality-manager/releases/6.0.1?p=allDownloads>

From the above link download the file **“Rational Quality Manager JUnit Selenium Adapter”.**

1. Unzip the Adapter ZIP file completely. There would be another ZIP file within it and that should be unzipped as well. Following are the contents of the file after complete UNZIP.



1. Create any text file containing the following paths from the libs directory of the client driver



1. Open the file and make sure the path has forward slash / instead of back slash \. The list of JAR’s should be shown as follows in a **SINGLE LINE**. Something similar to the following.

Also note that the path for the following JAR from the client driver location directory is also added to the text file.

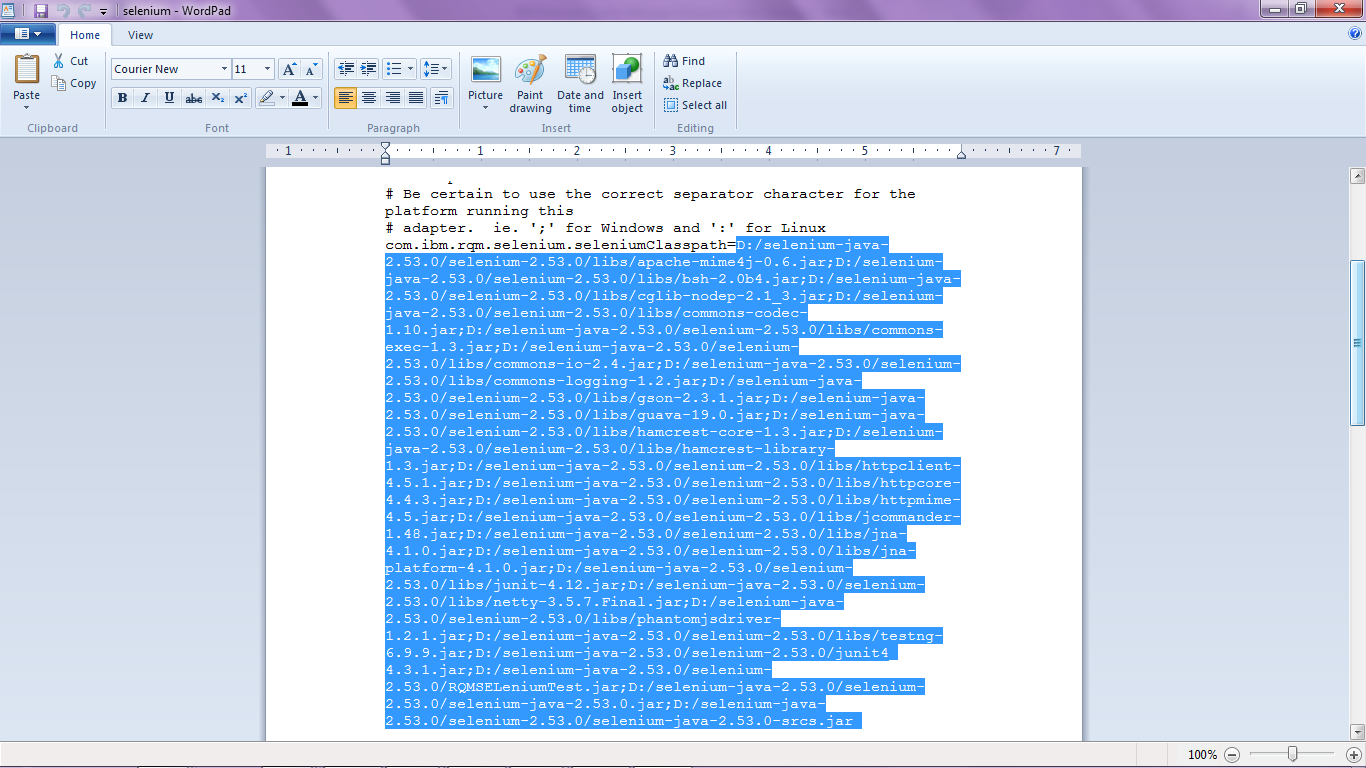
* + - selenium-java-2.53.0.jar
    - selenium-java-2.53.0-srcs.jar
    - junit4\_4.3.1.jar
    - RQMSELeniumTest.jar (which was exported from Eclipse)

The resulting file looks like as shown below

D:/selenium-java-2.53.0/selenium-2.53.0/libs/apache-mime4j-0.6.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/bsh-2.0b4.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/cglib-nodep-2.1\_3.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/commons-codec-1.10.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/commons-exec-1.3.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/commons-io-2.4.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/commons-logging-1.2.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/gson-2.3.1.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/guava-19.0.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/hamcrest-core-1.3.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/hamcrest-library-1.3.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/httpclient-4.5.1.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/httpcore-4.4.3.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/httpmime-4.5.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/jcommander-1.48.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/jna-4.1.0.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/jna-platform-4.1.0.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/junit-4.12.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/netty-3.5.7.Final.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/phantomjsdriver-1.2.1.jar;D:/selenium-java-2.53.0/selenium-2.53.0/libs/testng-6.9.9.jar;D:/selenium-java-2.53.0/selenium-2.53.0/junit4\_4.3.1.jar;D:/selenium-java-2.53.0/selenium-2.53.0/RQMSELeniumTest.jar;D:/selenium-java-2.53.0/selenium-2.53.0/selenium-java-2.53.0.jar;D:/selenium-java-2.53.0/selenium-2.53.0/selenium-java-2.53.0-srcs.jar

Locate and open the **selenium.properties** file that is stored in the **RQMJUnitSeleniumAdapter** folder. In this case it is **D:\RQM-Extras-JUnitSeleniumAdapter-6.0.1\adapters.** Remove # for this property. Paste in the list of file paths as shown in the above box from the above text file next to the property **com.ibm.rqm.selenium.seleniumClasspath**=

The **selenium.properties** file now looks as shown below.



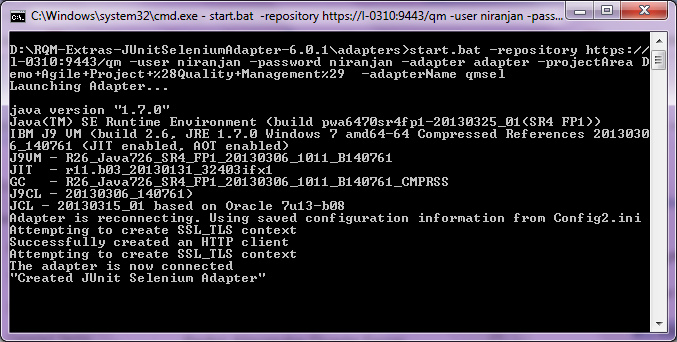
# Start the Selenium Adapter

* 1. Open a command prompt and navigate to the Adapter directory (In this case the directory would be D:\RQM-Extras-JUnitSeleniumAdapter-6.0.1\adapters)
  2. Run the following command to start the adapter

start.bat -repository https://rqmserver:port/qm -user userid –password password –adapter adapter -projectArea projectAreaName -adapterName adaptername

**Example of the above command:**

**D:\RQM-Extras-JUnitSeleniumAdapter-6.0.1\adapters>**start.bat -repository https://l-0310:9443/qm -user niranjan -password niranjan -adapter adapter -projectArea Demo+Agile+Project+%28Quality+Management%29 -adapterName qmsel



Make sure **Java.exe** is added to the PATH.

Note: In the above command to get the ProjectArea name you will need to provide Alias. In any browser enter the URL to get the alias.

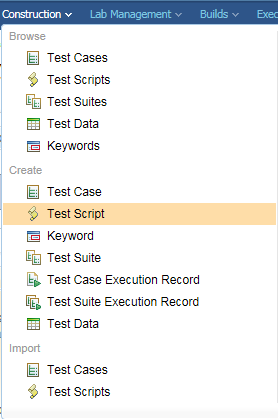
https://<Server>:9443/qm/service/com.ibm.rqm.integration.service.IIntegrationService/resources/projects

# Create and Run the Selenium Test Script in IBM Rational Quality Manager

The last step now is to create the Selenium Test Script within IBM Rational Quality Manager

## Create a Test Script in IBM Rational Quality Manager

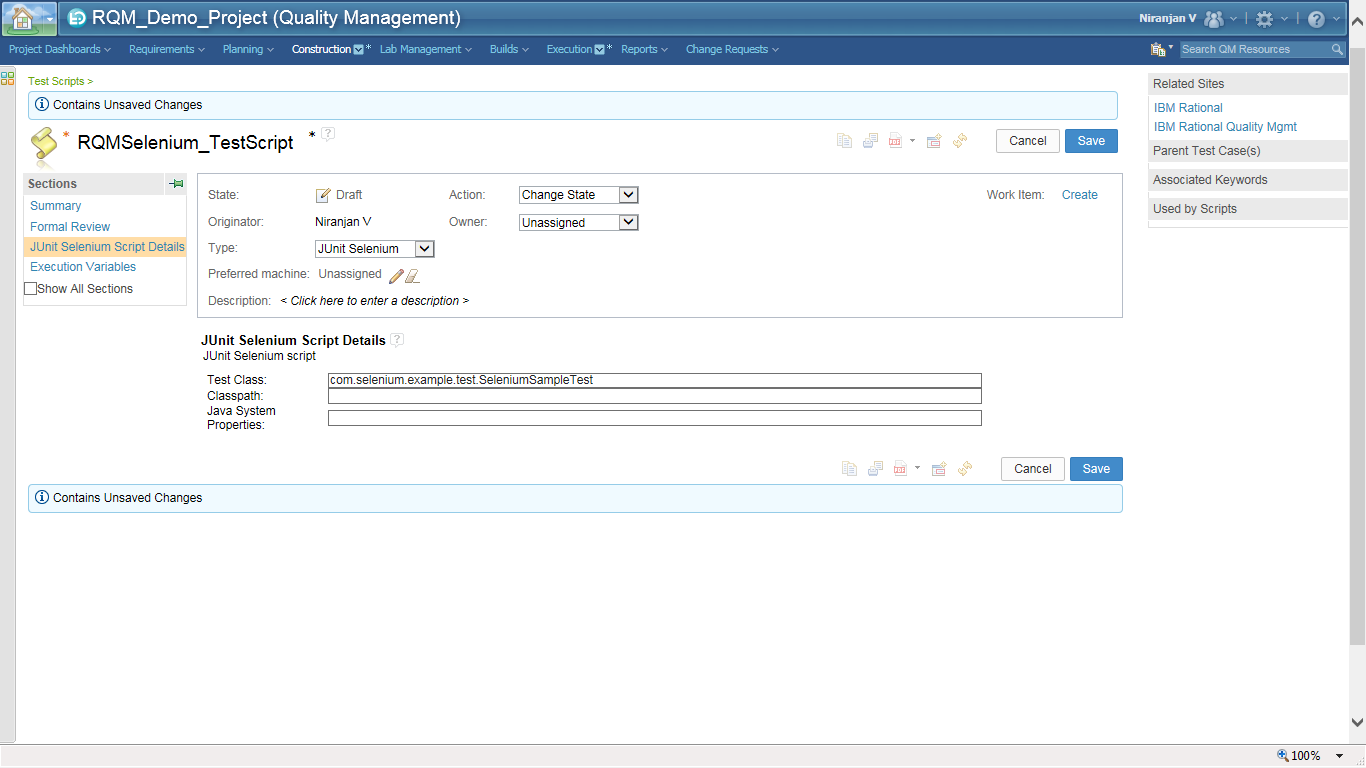
To create a Test Script in IBM Rational Quality Manager select **Construction🡪Create🡪 Test Script.**

****

From the Type drop down select **“Junit Selenium”**

In the Test Class enter: **com.selenium.example.test.SeleniumSampleTest**

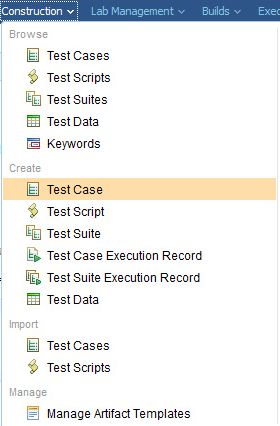
**The above was the Junit test created in Eclipse. This means that the specific Selenium test can run from the JAR file through the test script which is just created.**

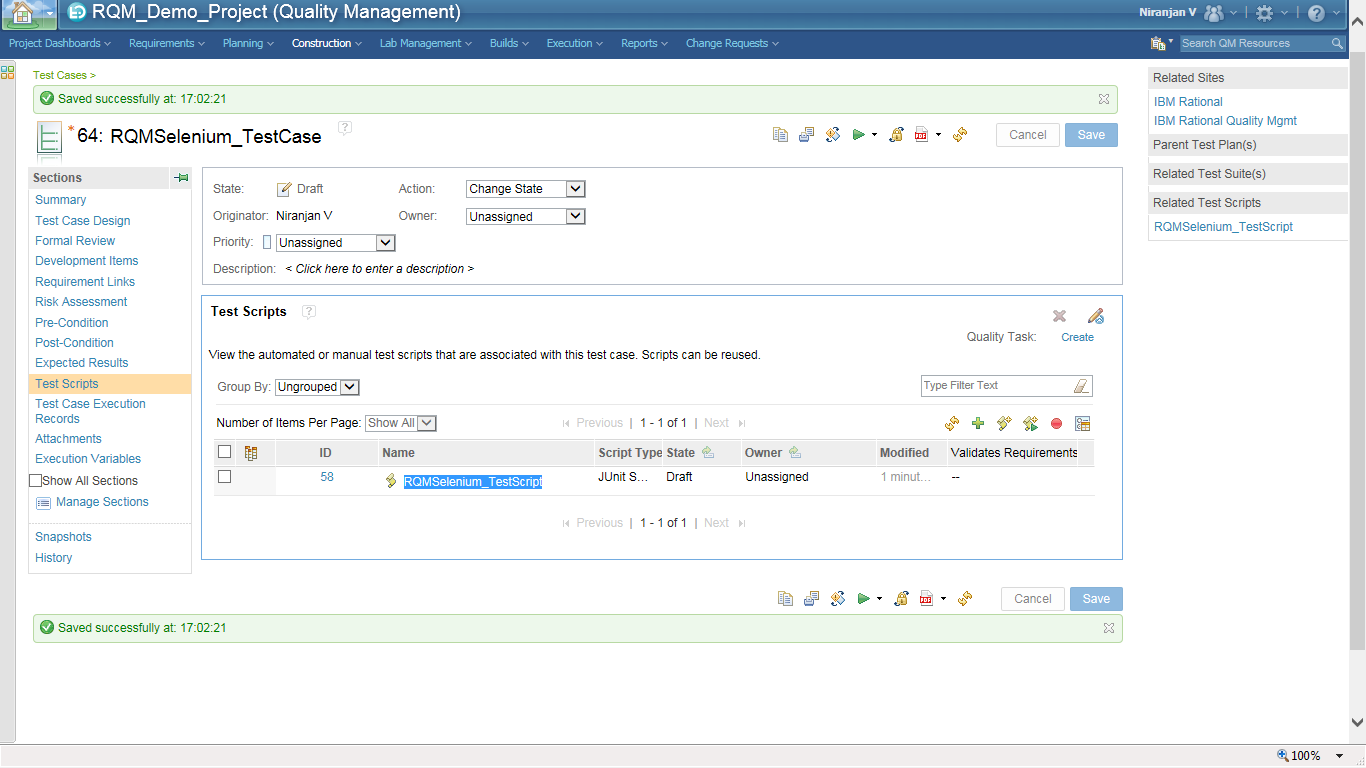


**Click on Save.**

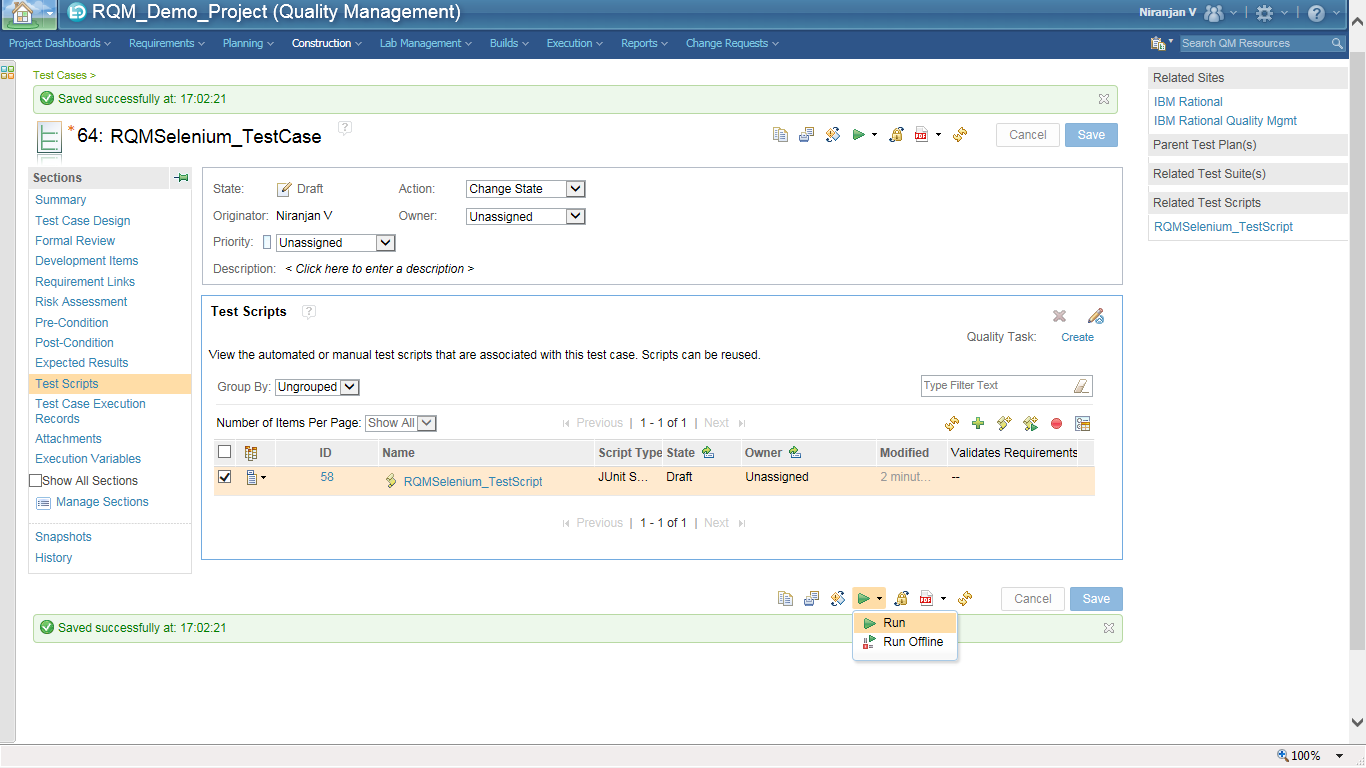
## Run the Automated Selenium Test from IBM Rational Quality Manager

To run the above Test Script we need to create a Test Case in RQM. Select **Construction🡪 Create 🡪 Test Case.** Once the Test Case is created add the above Test Script to the Test Case and execute the same.

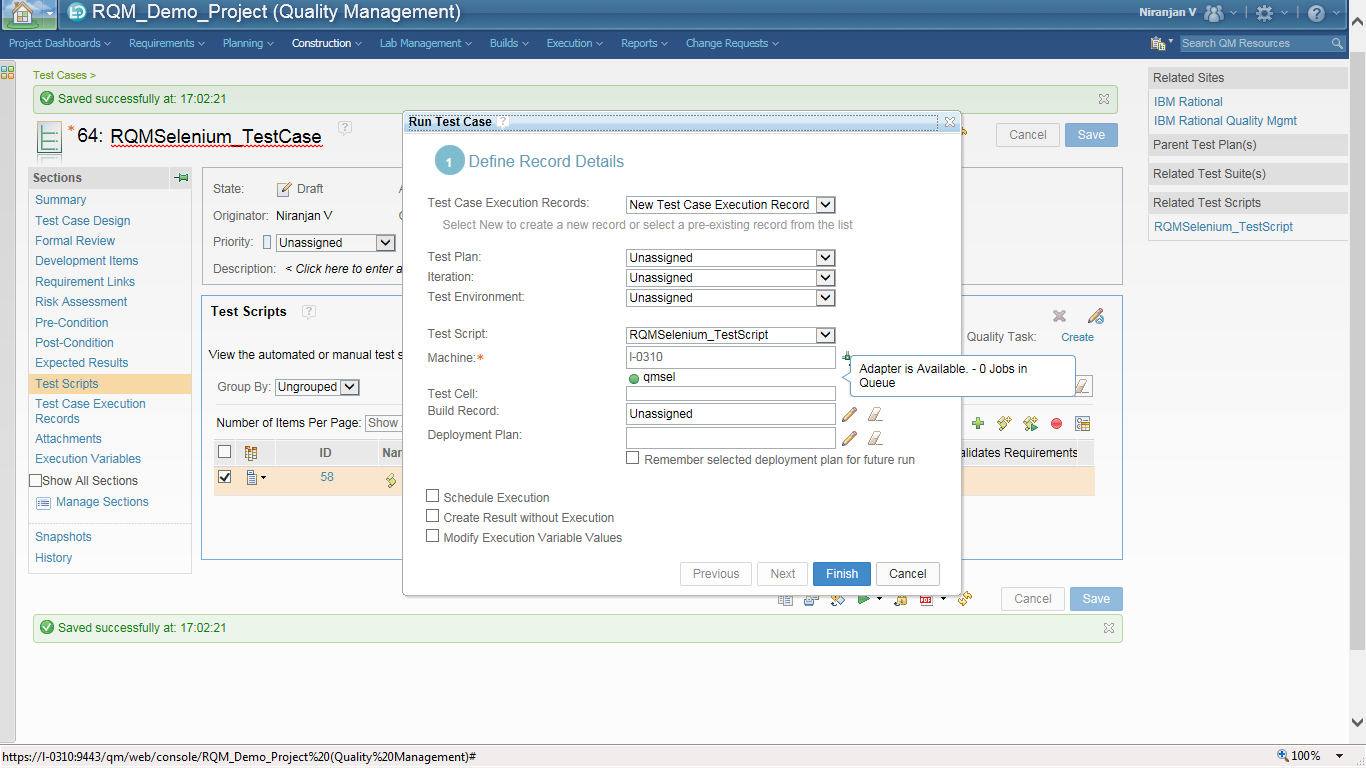




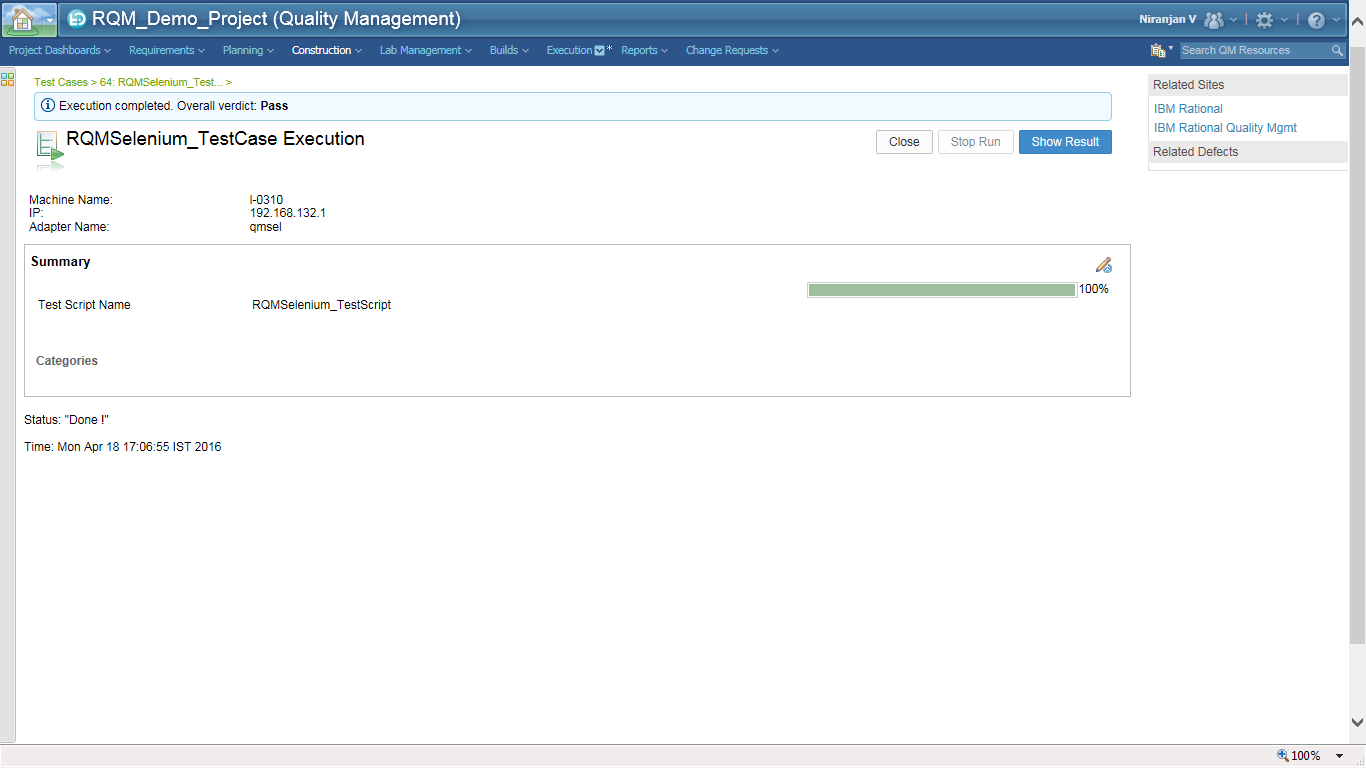
Select the Test Script and click on **Run.**



Click on **Finish** to begin the execution



The execution is completed and the result is shown. ***You would have noticed that in the background the Selenium test would have been executed.***



Click on Show Result to view the log file created in IBM Rational Quality Manager

# Summary

In this article we have seen the following

* Installing Selenium as plugin to Firefox
* Record a sample Selenium test in the IDE
* Configuring Eclipse to hold the Selenium Junit test
* Configure the Selenium test in IBM Rational Quality Manager
* Create the Selenium Test script in IBM Rational Quality Manager
* Run the Automated Selenium Test from IBM Rational Quality Manager and view the results.