**How to Use TestNG Framework for Creating Selenium Scripts**

TestNG is an advance framework designed in a way to leverage the benefits by both the developers and testers. For people already using JUnit, TestNG would seem no different with some advance features. With the commencement of the frameworks, JUnit gained an enormous popularity across the Java applications, Java developers and Java testers, with remarkably increasing the code quality. Despite being an easy to use and straightforward framework, JUnit has its own limitations which give rise to the need of bringing [TestNG](http://testng.org/doc/index.html" \o "TestNG framework)into the picture. TestNG was created by an acclaimed programmer named as “Cedric Beust”. TestNG is an open source framework which is distributed under the Apache software License and is readily available for download.

Talking about our requirement to introduce TestNG with WebDriver is that it provides an efficient and effective test result format that can in turn be shared with the stake holders to have a glimpse on the product’s/application’s health thereby eliminating the drawback of WebDriver’s incapability to generate test reports. TestNG has an inbuilt exception handling mechanism which lets the program to run without terminating unexpectedly.

Both TestNG and JUnit belong to the same family of Unit Frameworks where TestNG is an extended version to JUnit and is more extensively used in the current testing era.

### ****Features of TestNG****

* Support for annotations
* Support for parameterization
* Advance execution methodology that do not require test suites to be created
* Support for Data Driven Testing using Dataproviders
* Enables user to set execution priorities for the test methods
* Supports threat safe environment when executing multiple threads
* Readily supports integration with various tools and plug-ins like build tools (Ant, Maven etc.), Integrated Development Environment (Eclipse).
* Facilitates user with effective means of Report Generation using ReportNG

**TestNG versus JUnit**

There are various advantages that make TestNG superior to JUnit. Some of them are:

* Advance and easy annotations
* Execution patterns can be set
* Concurrent execution of test scripts
* Test case dependencies can be set

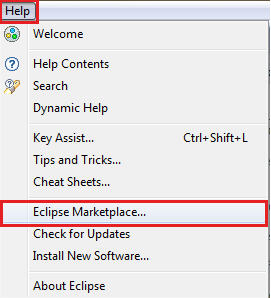
Annotations are preceded by a “@” symbol in both TestNG and JUnit.

So now let us get started with the installation and implementation part.

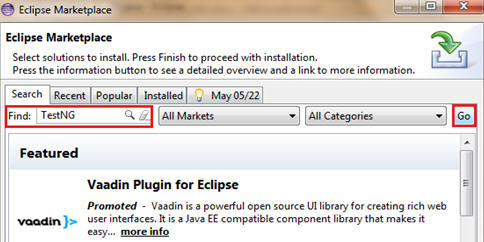
### ****TestNG Installation in Eclipse****

**Follow the below steps to TestNG Download and installation on eclipse:**

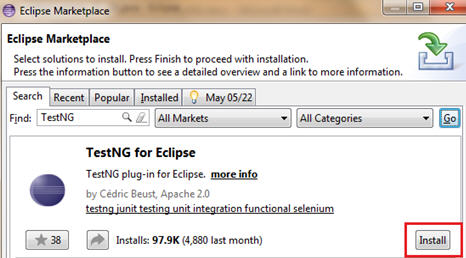
**Step 1:**Launch eclipse IDE -> Click on the Help option within the menu -> Select “Eclipse Marketplace..” option within the dropdown.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-TestNG-tutorial-1.jpg)

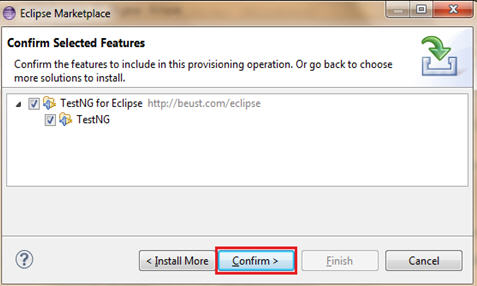
**Step 2:** Enter the keyword “TestNG” in the search textbox and click on “Go” button as shown below.

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

**Step 3:** As soon as the user clicks on the “Go” button, the results matching to the search string would be displayed. Now user can click on the Install button to install TestNG.

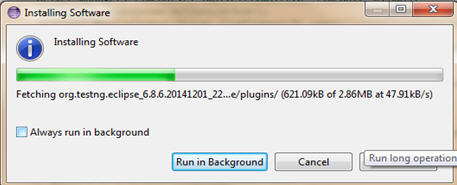
[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-TestNG-tutorial-3.jpg)

**Step 4:** As soon as the user clicks on the Install button, the user is prompted with a window to confirm the installation. Click on “Confirm” button.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-TestNG-tutorial-4.jpg)

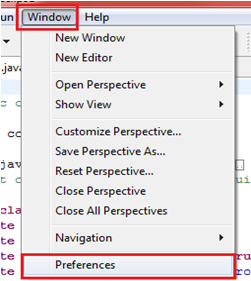
**Step 5:** In the next step, the application would prompt you to accept the license and then click on the “Finish” button.

**Step 6:** The installation is initiated now and the progress can be seen as following:

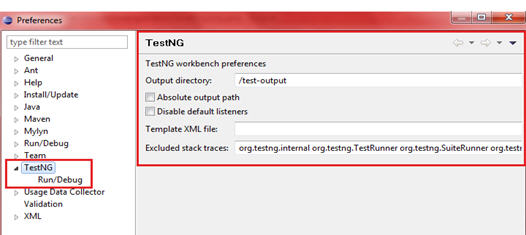
[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-TestNG-tutorial-5.jpg)

We are advised to restart our eclipse so as to reflect the changes made.

Upon restart, user can verify the TestNG installation by navigating to “Preferences” from “Window” option in the menu bar. Refer the following figure for the same.

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

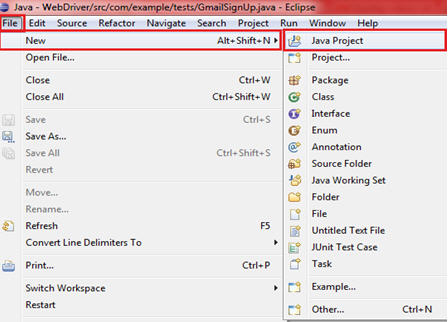
(Click on image to view enlarged)

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

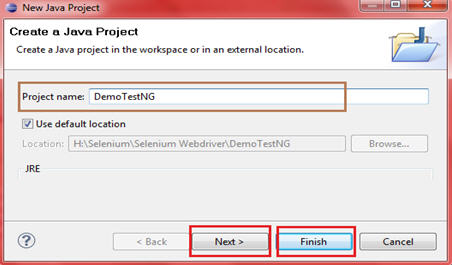
### ****Creation of Sample TestNG project****

Let us begin with the creation of TestNG project in eclipse IDE.

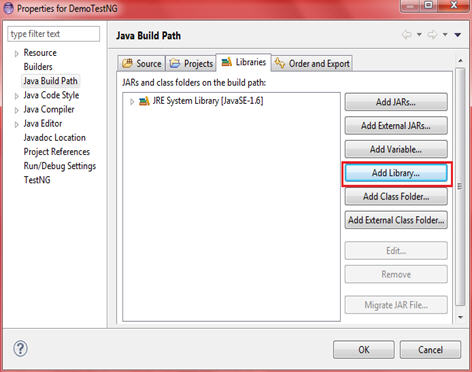
**Step 1:** Click on the File option within the menu -> Click on New -> Select Java Project.

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

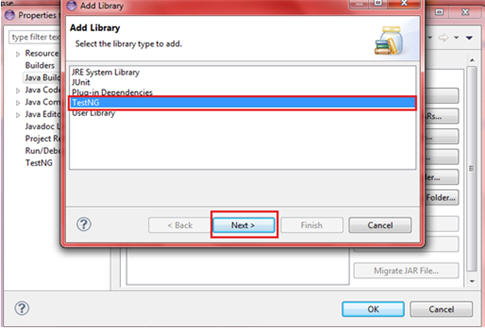
**Step 2:** Enter the project name as “DemoTestNG” and click on “Next” button. As a concluding step, click on the “Finish” button and your Java project is ready.

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

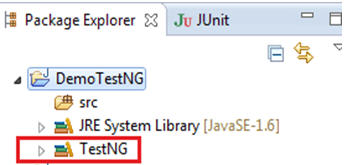
**Step 3:** The next step is to configure the TestNG library into the newly created Java project. For the same, Click on the “Libraries” tab under Configure Build Path. Click on “Add library” as shown below.

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

**Step 4:** The user would be subjected with a dialog box promoting him/her to select the library to be configured. Select TestNG and click on the “Next” button as shown below in the image. In the end, click on the “Finish” button.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-TestNG-tutorial-11.jpg)

The TestNG is now added to the Java project and the required libraries can be seen in the package explorer upon expanding the project.

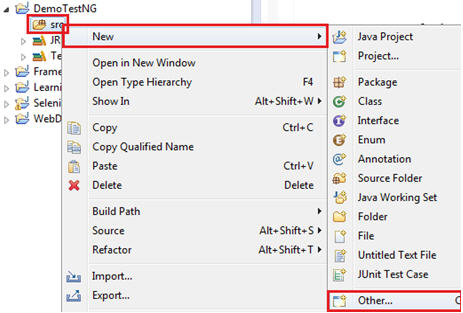
[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-TestNG-tutorial-12.jpg)

Add all the downloaded Selenium libraries and jars in the project’s build path as illustrated in the previous tutorial.

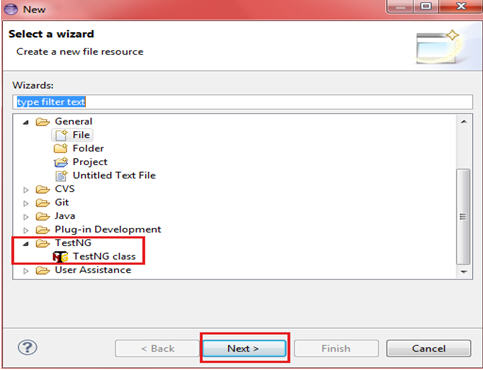
### ****Creating TestNG class****

Now that we have done all the basic setup to get started with the test script creation using TestNG. Let’s create a sample script using TestNG.

**Step 1:** Expand the “DemoTestNG” project and traverse to “src” folder. Right click on the “src”package and navigate to New -> Other..

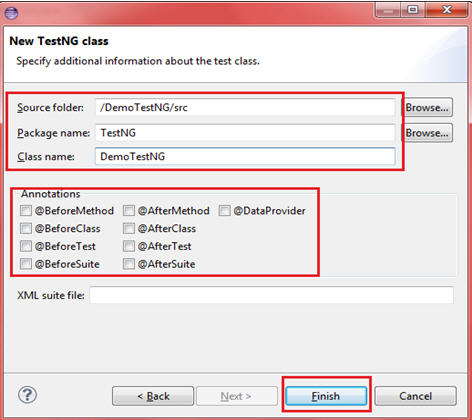
[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-TestNG-tutorial-13.jpg)

**Step 2:** Expand TestNG option and select “TestNG” class option and click on the “Next” button.

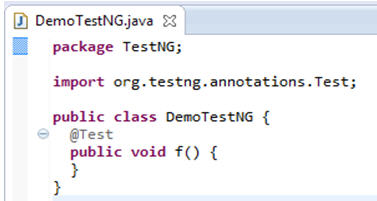
[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-TestNG-tutorial-14.jpg)

**Step 3:** Furnish the required details as following. Specify the Source folder, package name and the TestNG class name and click on the Finish button. As it is evident from the below picture, user can also check various TestNG notations that would be reflected in the test class schema. TestNG annotations would be discussed later in this session.

------------

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

The above mentioned TestNG class would be created with the default schema.

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

Now that we have created the basic foundation for the TestNG test script, let us now inject the actual test code. We are using the same code we used in the previous session.

**Scenario:**

* Launch the browser and open “gmail.com”.
* Verify the title of the page and print the verification result.
* Enter the username and Password.
* Click on the Sign in button.
* Close the web browser.

**Code:**

|  |  |
| --- | --- |
| 1 | package TestNG; |
| 2 | import org.openqa.selenium.By; | |

|  |  |
| --- | --- |
| 3 | import org.openqa.selenium.WebDriver; |
| 4 | import org.openqa.selenium.WebElement; | |

|  |  |  |
| --- | --- | --- |
| 5 | import org.openqa.selenium.firefox.FirefoxDriver; | |
| 6 | import org.testng.Assert; |

|  |  |  |
| --- | --- | --- |
| 7 | import org.testng.annotations.Test; | |
| 8 |  |

|  |  |  |
| --- | --- | --- |
| 9 | public class DemoTestNG { | |
| 10 | | public WebDriver driver = new FirefoxDriver(); | |

|  |  |  |
| --- | --- | --- |
| 11 | String appUrl = &quot;https://accounts.google.com&quot;; | |
| 12 |  |

|  |  |
| --- | --- |
| 13 | @Test |
| 14 | public void gmailLogin() { | |

|  |  |  |
| --- | --- | --- |
| 15 | // launch the firefox browser and open the application url | |
| 16 | driver.get(&quot;https://gmail.com&quot;); |

|  |  |
| --- | --- |
| 17 |  |
| 18 | // maximize the browser window | |

|  |  |  |
| --- | --- | --- |
| 19 | driver.manage().window().maximize(); | |
| 20 |  |

|  |  |
| --- | --- |
| 21 | // declare and initialize the variable to store the expected title of the webpage. |
| 22 | String expectedTitle = &quot; Sign in - Google Accounts &quot;; |

|  |  |
| --- | --- |
| 23 |  |
| 24 | // fetch the title of the web page and save it into a string variable | |

|  |  |
| --- | --- |
| 25 | String actualTitle = driver.getTitle(); |
| 26 | Assert.assertEquals(expectedTitle,actualTitle); | |

|  |  |
| --- | --- |
| 27 |  |
| 28 | // enter a valid username in the email textbox | |

|  |  |  |
| --- | --- | --- |
| 29 | WebElement username = driver.findElement(By.id(&quot;Email&quot;)); | |
| 30 | username.clear(); |

|  |  |  |
| --- | --- | --- |
| 31 | username.sendKeys(&quot;TestSelenium&quot;); | |
| 32 |  |

|  |  |
| --- | --- |
| 33 | // enter a valid password in the password textbox |
| 34 | WebElement password = driver.findElement(By.id(&quot;Passwd&quot;)); | |

|  |  |
| --- | --- |
| 35 | password.clear(); |
| 36 | password.sendKeys(&quot;password123&quot;); | |

|  |  |
| --- | --- |
| 37 |  |
| 38 | // click on the Sign in button | |

|  |  |  |
| --- | --- | --- |
| 39 | WebElement SignInButton = driver.findElement(By.id(&quot;signIn&quot;)); | |
| 40 | SignInButton.click(); |

|  |  |
| --- | --- |
| 41 |  |
| 42 | // close the web browser | |

|  |  |  |
| --- | --- | --- |
| 43 | driver.close(); | |
| 44 | } |

|  |  |
| --- | --- |
| 45 | } |

**Code Explanation with respect to TestNG**

**1)** @Test – @Test is one of the **TestNG annotations**. This annotation lets the program execution to know that method annotated as @Test is a test method. To be able to use different TestNG annotations, we need to import the package “**import**org.testng.annotations.\*”.

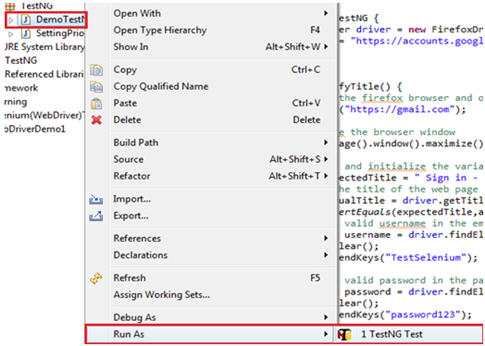
**2)** There is no need of main() method while creating test scripts using TestNG. The program execution is done on the basis of annotations.

**3)** In a statement, we used Assert class while comparing expected and the actual value. Assert class is used to perform various verifications. To be able to use different assertions, we are required to import “**import** org.testng.Assert”.

### ****Executing the TestNG script****

The TestNG test script can be executed in the following way:

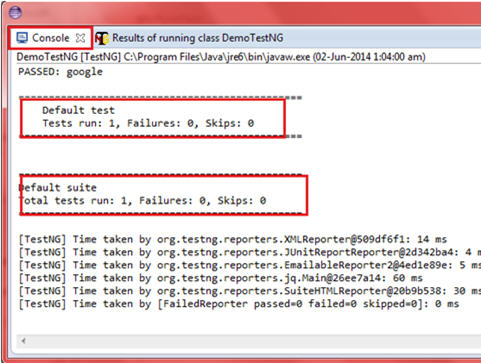
=> Right click anywhere inside the class within the editor or the java class within the package explorer, select “Run As” option and click on the “TestNG Test”.

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

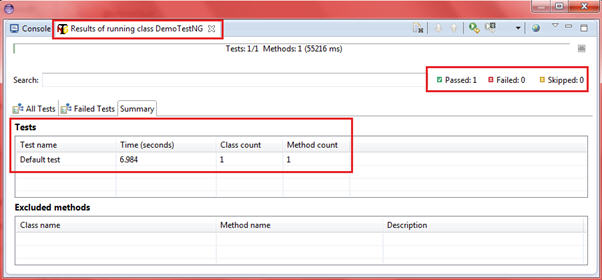
**TestNG result is displayed into two windows:**

* Console Window
* TestNG Result Window

**Refer the below screencasts for the result windows:**

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-TestNG-tutorial-18.jpg)

(Click on image to view enlarged)

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

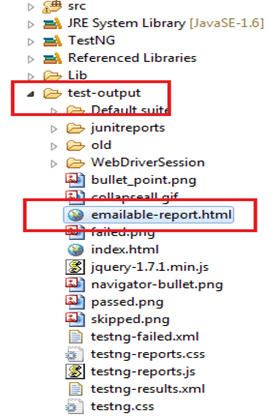
### ****HTML Reports****

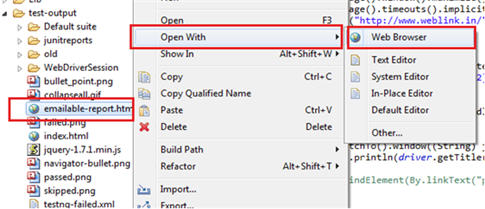
TestNG comes with a great capability of generating user readable and comprehensible HTML reports for the test executions. These reports can be viewed in any of the browser and it can also be viewed using Eclipse’s build –in browser support.

**To generate the HTML report, follow the below steps:**

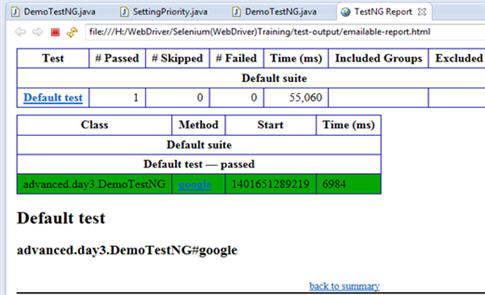
**Step 1:** Execute the newly created TestNG class. Refresh the project containing the TestNG class by right clicking on it and selecting “Refresh” option.

**Step 2:** A folder named as “test-output” shall be generated in the project at the “src” folder level. Expand the “test-output” folder and open on the “emailable-report.html” file with the Eclipse browser. The HTML file displays the result of the recent execution.

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

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-TestNG-tutorial-21.jpg)

**Step 3:** The HTML report shall be opened with in the eclipse environment. Refer the below image for the same.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-TestNG-tutorial-22.jpg)

Refresh the page to see the results for fresh executions if any.

### ****Setting Priority in TestNG****

**Code Snippet**

|  |  |
| --- | --- |
| 1 | package TestNG; |
| 2 | import org.testng.annotations.\*; | |

|  |  |  |
| --- | --- | --- |
| 3 | public class SettingPriority { | |
| 4 |  |

|  |  |
| --- | --- |
| 5 | @Test(priority=0) |
| 6 | public void method1() { | |

|  |  |  |
| --- | --- | --- |
| 7 | } | |
| 8 |  |

|  |  |  |
| --- | --- | --- |
| 9 | @Test(priority=1) | |
| 10 | | public void method2() { | |

|  |  |  |
| --- | --- | --- |
| 11 | } | |
| 12 |  |

|  |  |
| --- | --- |
| 13 | @Test(priority=2) |
| 14 | public void method3() { | |

|  |  |  |
| --- | --- | --- |
| 15 | } | |
| 16 | } |

### ****Code Walkthrough****

If a test script is composed of more than one test method, the execution priority and sequence can be set using TestNG annotation “@Test” and by setting a value for the “priority” parameter.

In the above code snippet, all the methods are annotated with the help @Test and the priorities are set to 0, 1 and 2. Thus the order of execution in which the test methods would be executed is:

* Method1
* Method2
* Method3

**Support for Annotations**

There are number of annotations provided in TestNG and JUnit. The subtle difference is that TestNG provides some more advance annotations to JUnit.

### ****TestNG Annotations:****

**Following is the list of the most useful and favorable annotations in TestNG:**

| **Annotation** | **Description** |
| --- | --- |
| @Test | The annotation notifies the system that the method annotated as @Test is a test method |
| @BeforeSuite | The annotation notifies the system that the method annotated as @BeforeSuite must be executed before executing the tests in the entire suite |
| @AfterSuite | The annotation notifies the system that the method annotated as @AfterSuite must be executed after executing the tests in the entire suite |
| @BeforeTest | The annotation notifies the system that the method annotated as @BeforeTest must be executed before executing any test method within the same test class |
| @AfterTest | The annotation notifies the system that the method annotated as @AfterTest must be executed after executing any test method within the same test class |
| @BeforeClass | The annotation notifies the system that the method annotated as @BeforeClass must be executed before executing the first test method within the same test class |
| @AfterClass | The annotation notifies the system that the method annotated as @AfterClass must be executed after executing the last test method within the same test class |
| @BeforeMethod | The annotation notifies the system that the method annotated as @BeforeMethod must be executed before executing any and every test method within the same test class |
| @AfterMethod | The annotation notifies the system that the method annotated as @AfterMethod must be executed after executing any and every test method within the same test class |
| @BeforeGroups | The annotation notifies the system that the method annotated as @BeforeGroups is a configuration method that enlists a group and that must be executed before executing the first test method of the group |
| @AfterGroups | The annotation notifies the system that the method annotated as @AfterGroups is a configuration method that enlists a group and that must be executed after executing the last test method of the group |

Note: Many of the aforementioned annotations can be exercised in JUnit 3 and JUnit 4 framework also.

### ****Conclusion****

Through this tutorial, we tried to make you acquainted with a java based testing framework named as TestNG. We started off the session with the installation of the framework and moved with the script creation and advance topics. We discussed all the annotations provided by TestNG. We implemented and executed our first TestNG test script using annotations and assert statements.

**Article summary:**

* TestNG is an advance framework designed in a way to leverage the benefits by both the developers and testers.
* TestNG is an open source framework which is distributed under the Apache software License and is readily available for download.
* TestNG is considered to be superior to JUnit because of its advance features.
* **Features of TestNG**
  + Support for Annotations
  + Advance execution methodology that do not require test suites to be created
  + Support for parameterization
  + Support for Data Driven Testing using Dataproviders
  + Setting execution priorities for the test methods
  + Supports threat safe environment when executing multiple threads
  + Readily supports integration with various tools and plug-ins like build tools (Ant, Maven etc.), Integrated Development Environment (Eclipse).
  + Facilitates user with effective means of Report Generation using ReportNG
* **Advantages of TestNG over JUnit**
  + Added advance and easy annotations
  + Execution patterns can be set
  + Concurrent execution of test scripts
  + Test case dependencies can be set
* TestNG is freely available and can be easily installed in the Eclipse IDE using Eclipse Market.
* Upon installation, TestNG would be available as a library within the Eclipse environment.
* Create a new Java Project and configure the build path using TestNG library.
* Create a new TestNG class by expanding the created TestNG project and traverse to its “src” folder. Right click on the “src” package and navigate to New -> Other. Select TestNG class option.
* @Test is one of the annotations provided by TestNG. This annotation lets the program execution to know that method annotated as @Test is a test method. To be able to use different TestNG annotations, we need to import the package “**import**org.testng.annotations.\*”.
* There is no need of main() method while creating test scripts using TestNG.
* We use Assert class while comparing expected and the actual value. Assert class is used to perform various verifications. To be able to use different assertions, we are required to import “**import**org.testng.Assert”.
* If a test script is composed of more than one test methods, the execution priority and sequence can be set using TestNG annotation “@Test” and by setting a value for the “priority” parameter.
* TestNG has a capability of generating human readable test execution reports automatically. These reports can be viewed in any of the browser and it can also be viewed using Eclipse’s built – in browser support.