Testing Framework - TestNG

Shahzeb Imtiaz Satti

Basic of TestNG Framework

10/5/17

Contents

1. Introduction of TestNG

1.1. What is TestNG?

TestNG is a testing framework inspired from JUnit and NUnit but introducing some new functionalities that make it more powerful and easier to use. TestNG is designed to cover all categories of tests: unit, functional, end-to-end, integration, etc.

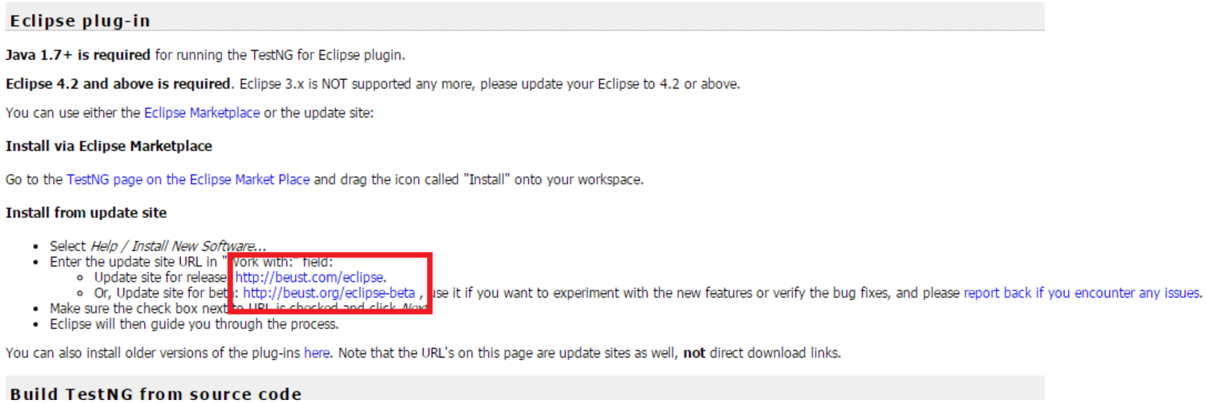
1.2. Requirements

- TestNG requires JDK7 or higher.

For more visit: http://testng.org/doc/

2. Installation of TestNG

Open following URL: http://testng.org/doc/download.html



Eclipse plug-in

Java 1.7+ is required for running the TestNG for Eclipse plugin.

Eclipse 4.2 and above is required. Eclipse 3.x is NOT supported any more, please update your Eclipse to 4.2 or above.

You can use either the Eclipse Marketplace or the update site:

Install via Eclipse Marketplace

Go to the TestNG page on the Eclipse Market Place and drag the icon called "Install" onto your workspace.

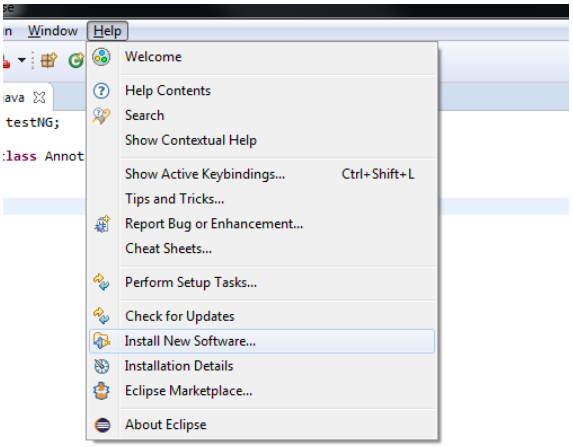
Install from update site

- Select *Help / Install New Software...*
- Enter the update site URL in "Work with:" field:
  - Update site for release: http://beust.com/eclipse.
  - Or, Update site for beta: http://beust.org/eclipse-beta , use it if you want to experiment with the new features or verify the bug fixes, and please report back if you encounter any issues.
- Make sure the check box next to URL is checked and click *Next*.
- Eclipse will then guide you through the process.

You can also install older versions of the plug-ins here. Note that the URL's on this page are update sites as well, **not** direct download links.

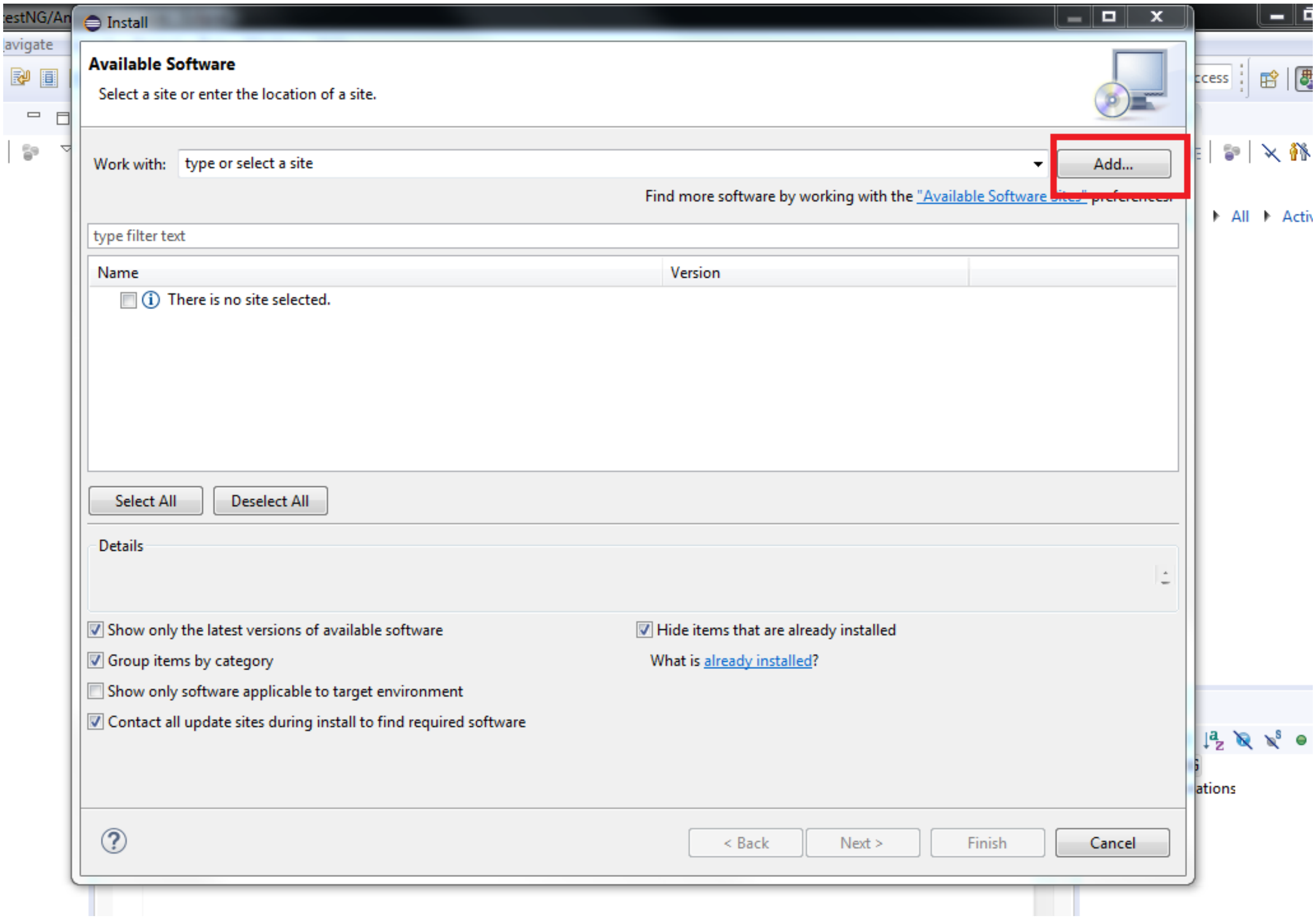
Build TestNG from source code

Copy link according to your installed eclipse and open eclipse.



ava
testNG;
lass Annot

Click on Add button.



Install

Available Software

Select a site or enter the location of a site.

Work with: type or select a site

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
□ ⓘ There is no site selected.	

Details

☑ Show only the latest versions of available software

☑ Group items by category

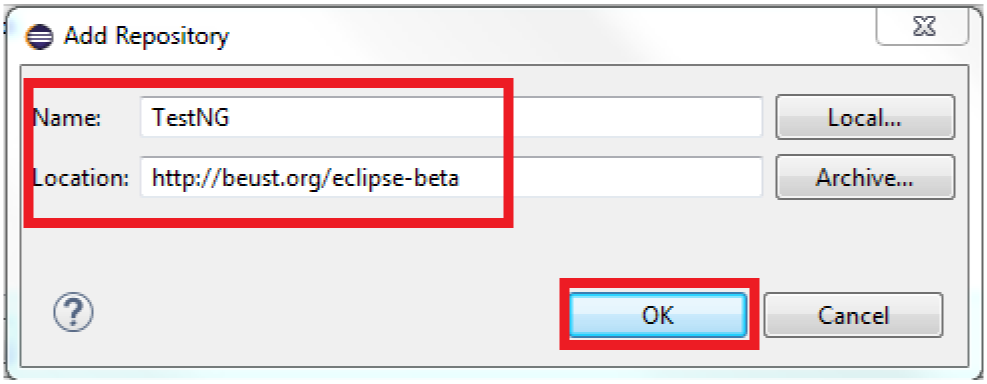
□ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

☑ Hide items that are already installed

What is already installed?

Paste URL in location field and click on “OK”.



Add Repository

Name: TestNG

Location: http://beust.org/eclipse-beta

After installation restart your eclipse.

3. Basic of TestNG

3.1. Annotations

- @BeforeSuite
- @AfterSuite
- @BeforeClass
- @AfterClass
- @BeforeTest
- @AfterTest
- @BeforeGroups
- @AfterGroups
- @BeforeMethod
- @AfterMethod
- @DataProvider
- @Factory
- @Listeners
- @Parameters
- @Test

3.2. Benefits of using Annotations

- TestNG identifies the methods it is interested in, by looking up annotations. Hence, method names are not restricted to any pattern or format.
- We can pass additional parameters to annotations.
- Annotations are strongly typed, so the compiler will flag any mistakes right away.
- Test classes no longer need to extend anything (such as Test Case, for JUnit 3).

For More visit: https://www.tutorialspoint.com/testng/index.htm

3.3. Sample Code using Annotations

```
package testNG;

import org.testng.annotations.AfterMethod;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;

public class Annotations {

    @BeforeMethod
    //will execute 2 times
    public void Beforemethod()
    {
            System.out.println("2");
    }

    @BeforeTest
    public void BeforeTest()
    {
            System.out.println("1");
    }

    @Test
    public void Test2()
    {
            System.out.println("5");

    }
    @Test
    public void Test()
    {
            System.out.println("3");
    }

    @AfterTest
    public void AfterTest()
    {
            System.out.println("6");
    }

    @AfterMethod
    //will execute 2 times
    public void AfterMethod()
    {
            System.out.println("4");
    }
}
```

Note: run it as TestNG Test

3.4. TestNG - Dependency Test

TestNG allows you to specify dependencies either with:

- Using attribute *dependsOnMethods* in @Test annotations, OR.
- Using attribute *dependsOnGroups* in @Test annotations.

3.4.1. Example Using DependsOnMethods, enabled and alwaysRun

```
package testNG;

import org.testng.annotations.Test;
```

```
public class DependencyAnnotations {

	@Test
	public void Test2()
	{
		System.out.println("5");

	}
	@Test(dependsOnMethods={"OpenBroweser"}, alwaysRun=true)
	public void Test()
	{
		System.out.println("3");
	}

	@Test(enabled=false)
	public void Test3()
	{
		System.out.println("5");

	}
}
```

For more visit: https://www.tutorialspoint.com/testng/testng_dependency_test.htm