# TestNG

## What is TestNG?

TestNG is a testing framework designed to simplify a broad range of testing needs, from unit testing to integration testing.

## What are the advantages of TestNG?

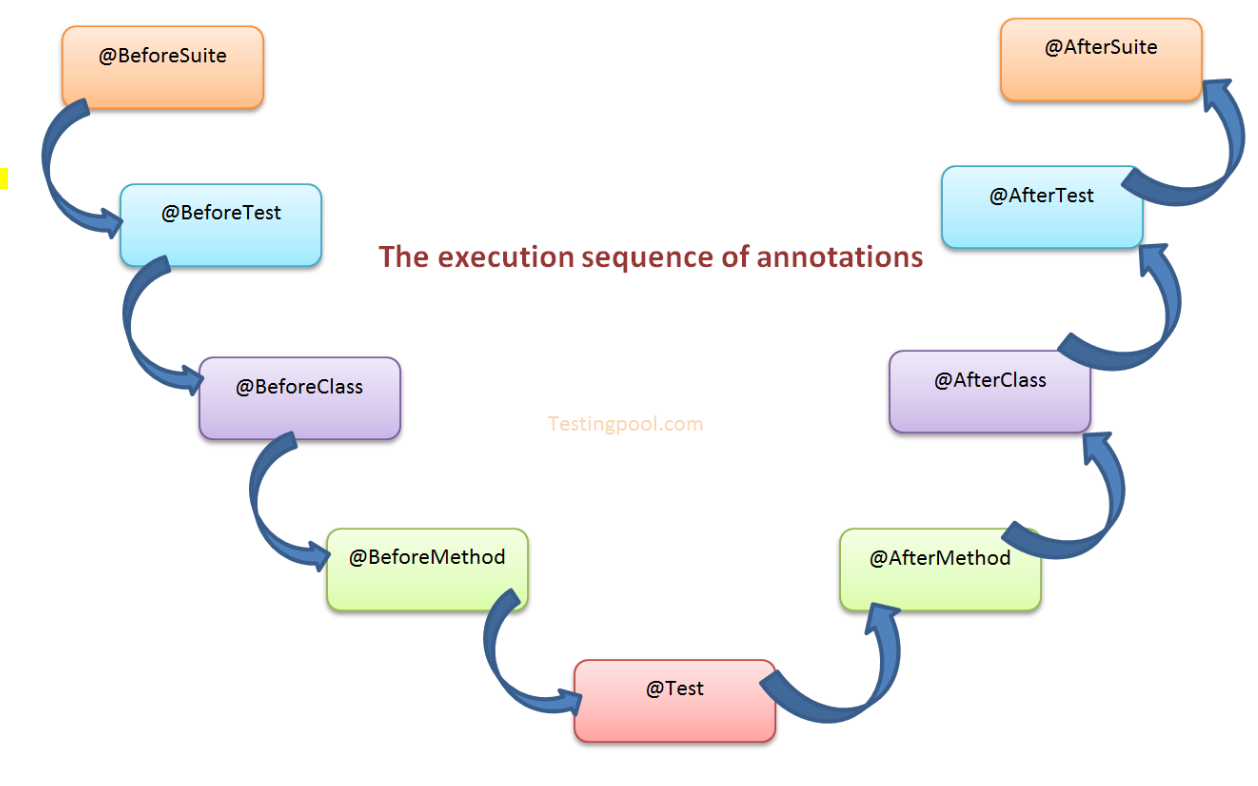
* TestNG provides parallel execution of test methods
* It allows to define dependency of one test method over other method
* It allows to assign priority to test methods @Test(priority=1, groups = “Login”)
* It allows grouping of test methods into test groups
* It has support for parameterizing test cases using @Parameters annotation
* It allows data driven testing using @DataProvider annotation
* It has different assertions that helps in checking the expected and actual results
* Detailed (HTML) reports
* Testng Listeners

# Annotations

## What are the annotations available in TestNG?

* @BeforeTest
* @AfterTest
* @BeforeClass
* @AfterClass
* @BeforeMethod
* @AfterMethod
* @BeforeSuite
* @AfterSuite
* @BeforeGroups
* @AfterGroups
* @Test
* @DataProvider
* @Parameters

## Execution Sequence of Annotations in TestNG:



## **@BeforeSuite:**

**The annotation method will be run before all tests in this suite have run.**

## **@BeforeTest:**

A method which is marked with this annotation will be executed before ***first @Test*** annotated method.

Or

The annotated method will run before any test method belongs to the classes inside the <test> tag is run.

## **@BeforeClass:**

A method which is marked with this annotation will be executed before ***first @Test*** method execution. It runs only once per class.

Or

The annotated method will be run before the first test method in the current class is invoked.

## **@BeforeMethod:**

A method which is marked with this annotation will be executed before every **@test** annotated method.

## **@Test:**

Marks a class or a method as a part of the test.

## **@AfterMethod:**

A method which is marked with this annotation will be executed after every **@test** annotated method.

BM=>Test=>AM

## **@AfterClass:**

A method which is marked with this annotation will be executed after all the test methods in the current class have been run.

## **@AfterTest:**

A method which is marked with this annotation will be executed when **all @Test** annotated methods complete the execution of those classes which are inside <test> tag in testng.xml file.

## **@AfterSuite:**

A method which is marked with this annotation will run **once after** executions of all tests in the suite have run

## **@BeforeGroups:**

This annotated method will run **before the first test run** of that specific group.

## ****@AfterGroups:****

This annotated method will run **after all test methods** of that group completes its execution.Some other TestNG Annotations, we need to discuss here are mentioned below:

## **@Parameters:**

This annotation is used to pass parameters to test methods.

## **@DataProvider:**

If we use @DataProvider annotation for any method that means you are using that method as a data supplier. The configuration of @DataProvider annotated method must be like it always return Object[][] which we can use in @Test annotated method. The @Test method that wants to receive data from this DataProvider needs to use a dataProvider name equals to the name of this annotation.

## **@Factory:**

Marks a method as a factory that returns objects that will be used by TestNG as Test classes. The method must return Object[ ]

## **@Listeners:**

This annotation is used with test class. It helps in writing logs and results.

# Command or Method

## What do you mean by grouping?

We can assign method with proper context and refine groping of test methods. Using Groping we can easily call that particular test for run.

## How to associate a single test to multiple Groups in TestNg?

TestNg framework allows multiple tests to run by using test group feature

We can associate a single test to multiple groups as shown below example.

@Test (groups= {“regression-testing”,”Smoke-Testing”})

## Is TestNG Capable of running multiple suites?

Yes we can run multiple testing suites in the following manner:

<suite name=*"Suite"*>

<suite-files>

< suite-files path=”testscript.xml”/>

< suite-files path=”testscript2.xml”/>

…..

</suite-files>

</suite> <!-- Suite -->

## Can you arrange the below testng.xml tags from parent to child?

<test>

<suite>

<class>

<methods>

<classes>

The correct order of the TestNG tags is as follows

<suite>

<test>

<classes>

<class>

<methods>

## How to create and run testng.xml?

In TestNG framework, we need to create testng.xml file to create and

Handle multiple test classes. We do configure our test run, set test dependency, include or exclude any test, method, class or package and set priority etc in the xml file.

## What is the importance of testng.xml file?

In a Selenium TestNG project, we use testng.xml file to configure the complete test suite in a single file. Some of the features are as follows.

* Testng.xml file allows including or excluding the execution of test methods and testing groups.
* It allows passing parameters to the test cases.
* Allows adding group dependencies.
* Allows adding priorities to the test cases.
* Allows configuring parallel execution of test cases.
* Allows parameterizing the test cases.

## How to pass parameter through testng.xml file to a test case?

## What is parameterization and how do you use it in TestNG

We could define the parameters in the testng.xml file and then reference those parameters in the source files.

Create a java test class, say, ParameterizedTest.*java*and add a test method say parameterizedTest() to the test class. This method takes a string as input parameter. Add the annotation @Parameters (“browser”) to this method.

// TestNG Interview Questions

Public class ParameterizedTest {

            @Test

            @Parameters ("browser")

            Public void parameterizedTest(String browser){

                        if(browser.equals("firefox")){

                                    System.out.println("Open Firefox Driver");

                        } else if (browser.equals("chrome")){

                                    System.out.println("Open Chrome Driver");

                        }

            }

}

The parameter would be passed a value from testng.xml, which we will see in the next step.

We could set the parameter using the below syntax in the testng.xml file.

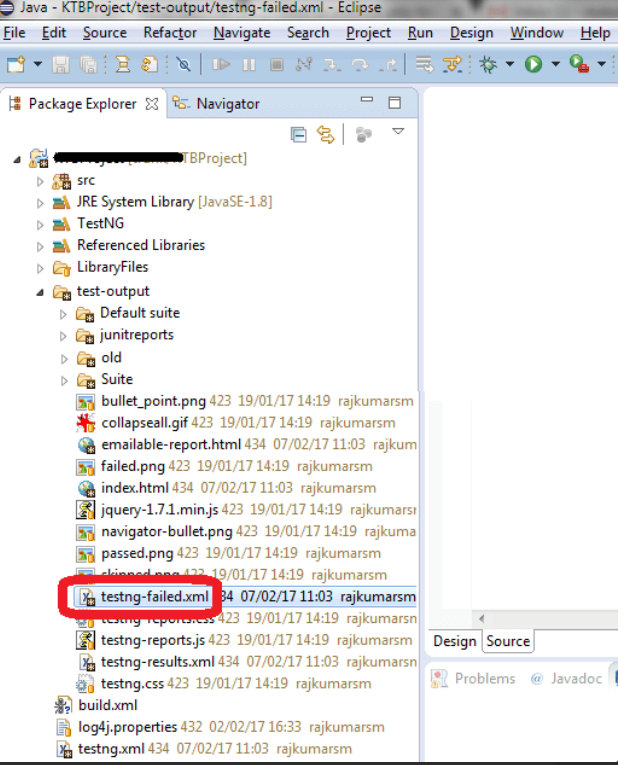
<parameter name="browser" value="firefox"/>

Here, name attribute represents the parameter name and value represents the value of that parameter.

## How Execute failed test cases using TestNG in Selenium – By using “testng-failed.xml”

**Steps To follow:**

1. After the first run of an automated test run. Right click on Project – Click on Refresh
2. A folder will be generated named “test-output” folder. Inside “test-output” folder, you could find “testng-failed.xml”
3. Run “testng-failed.xml” to execute the failed test cases again.



## What is TestNG Assert and list out common TestNG Assertions?

TestNG Asserts help us to verify the condition of the test in the middle of the test run. Based on the TestNG Assertions, we will consider a successful test only if it is completed the test run without throwing any exception.

Some of the common assertions supported by TestNG are

* assertEqual(String actual,String expected).
* assertEqual(String actual,String expected, String message).
* assertEquals(boolean actual,boolean expected).
* assertTrue(condition).
* assertTrue(condition, message).
* assertFalse(condition).
* assertFalse(condition, message).

## What is Soft Assert in TestNG?

A soft Assert collects error during @Test. Soft Assert does not throw an exception

When an assert fails and would continue with the next step after the assert statement.

If there is any exception and you want to throw it then you need to use assertAll() method as a last statement in the @Test and test suite again continue with next @Test as it is.

T1(){

1 sendkeys()

2 HA –passed à 3 else à terminate test case / failed

3 SA -- passed

4 SA  --- failed à 5

5 SA – passed à 6

6 SA – Failed

softAssert.assertAll();

}

## What is Hard Assert in TestNG?

Hard Assert throws an Assert Exception immediately when an assert statement fails and test suite continues with next *@Test*

What is exception test in TestNG?

TestNG gives an option for tracing the Exception handling of code. You can verify whether a code throws the expected exception or not. The expected exception to validate while running the test case is mentioned using the expectedExceptions attribute value along with @Test annotation.

@Test

t1(expectedExceptions = ElementNotFoundException){

}

## How to set test case priority in TestNG?

## How do you execute the tests based on priority?

We use priority attribute to the @Test annotations. In case priority is not set then the test scripts execute in alphabetical order.

// TestNG Interview Questions

package TestNG;

import org.testng.annotations.\*;

public class PriorityTestCase{

@Test(priority=0)

            public void testCase1() {

             system.out.println("Test Case 1");

            }

    @Test(priority=1)

            public void testCase2() {

                        system.out.println("Test Case 2");

            }

}

Output:

Test Case 1

Test Case 2

## In TestNG Priority and without priority which one will be first Excute?

## Can Priority have negative Number?

Yes because it takes integer number.

## What is parameterized testing in TestNG?

Parameterized tests allow developers to run the same test over and over again using different values.

There are two ways to set these parameters:

* Using testng.xml.
* Using Data Providers .

## How to run a group of test cases using TestNG?

TestNG allows you to perform sophisticated groupings of test methods. Not only can you declare that methods belong to groups, but you can also specify groups that contain other groups. Then TestNG can be invoked and asked to include a certain set of groups (or regular expressions) while excluding another set.  This gives you maximum flexibility in how you partition your tests and doesn’t require you to recompile anything if you want to run two different sets of tests back to back.

Groups are specified in your testng.xml file and can be found either under the <test> or <suite> tag. Groups specified in the <suite> tag apply to all the <test> tags underneath.

@Test (groups = { "smokeTest", "functionalTest" })

public void loginTest(){

System.out.println("Logged in successfully");

}

## How to create Group of Groups in TestNG?

Groups can also include other groups. These groups are called MetaGroups. For example, you might want to define a group all that includes smokeTest and functionalTest. Let’s modify our testng.xml file as follows:

<groups>

   <define name="all">

             <include name="smokeTest"/>

             <include name="functionalTest"/>

   </define>

   <run>

         <include name="all" />

   </run>

</groups>

## How to run test cases in parallel using TestNG?

We can use “parallel” attribute in testng.xml to accomplish parallel test execution in TestNG

The parallel attribute of suite tag can accept four values:

Tests – All the test cases inside <test> tag of testng.xml file will run parallel

Classes – All the test cases inside a java class will run parallel

Methods – All the methods with @Test annotation will execute parallel

Instances – Test cases in same instance will execute parallel but two methods of two different instances will run in different thread.

<suite name="softwaretestingmaterial" parallel="methods">

## How to exclude a particular test method from a test case execution?

By adding the exclude tag in the testng.xml

<classes>

  <class name="TestCaseName">

     <methods>

       <exclude name="TestMethodNameToExclude"/>

     </methods>

  </class>

</classes>

## How to exclude a particular test group from a test case execution?

By adding the exclude tag in the testng.xml

<groups>

    <run>

             <exclude name="TestGroupNameToExclude"/>

    </run>

</groups>

## How to disable a test case in TestNG ?

To disable the test case we use the parameter enabled = false to the @Test annotation.

@Test(enabled = false)

## How to skip a @Test method from execution in TestNG?

By using throw new SkipException()

Once SkipException() thrown, remaining part of that test method will not be executed and control will goes directly to next test method execution.

Throw new SkipException("Skipping - This is not ready for testing ");

## How to ignore a test case in TestNG?

To ignore the test case we use the parameter enabled = false to the @Test annotation.

@Test (enabled = false)

## How TestNG allows to state dependencies?

TestNG allows two ways to declare the dependencies.

Using attributes dependsOnMethods in @Test annotations using attributes dependsOnGroups in @Test annotations

## What are the different ways to produce reports for TestNG results?

TestNG offers two ways to produce a report.

Listeners implement the interface *org.testng.****ITestListener*** and are notified in real time of when a test starts passes, fails, etc…

Reporters implement the interface *org.testng.****IReporter*** and are notified when all the suites have been run by TestNG. The IReporter instance receives a list of objects that describe the entire test run.

## How to write regular expression in testng.xml file to search @Test methods containing “smoke” keyword.

Regular expression to find @Test methods containing keyword “smoke” is as mentioned below.

<methods>

     <include name=".\*smoke.\*"/>

</methods>

## What is the time unit we specify in test suites and test cases?

We specify the time unit in test suites and test cases is in milliseconds.

## List out various ways in which TestNG can be invoked?

TestNG can be invoked in the following ways

* Using Eclipse IDE.
* Using maven/ant build tool.
* From the command line.
* Using IntelliJ’s IDEA.

## How To Run TestNG Using Command Prompt?

C: test

Java c://testing.jar test.java

## What is the use of @Test (invocationCount=x)?

The invocationcount attribute tells how many times TestNG should run a test method

@Test (invocationCount = 10)

Public void testCase1 (){

In this example, the method *testCase1* will be invoked ten times

## What is the use of @Test (threadPoolSize=x)?

The threadPoolSize attribute tells to form a thread pool to run the test method through multiple threads.

Note: This attribute is ignored if invocationCount is not specified

@Test(threadPoolSize = 3, <code class="plain">invocationCount = </code><code class="value">10</code>) public void testCase1(){

In this example, the method *testCase1* will be invoked from three different threads

## What does the test timeout mean in TestNG?

The maximum number of milliseconds a test case should take.

@Test (threadPoolSize = 3, invocationCount = 10,  timeOut = 10000)

Public void testCase1 (){

In this example, the function testCase1 will be invoked ten times from three different threads. Additionally, a time-out of ten seconds guarantees that none of the threads will block on this thread forever.

## How do you run tests using Testng?

Using >Xml file

## How do you execute your framework from command prompt?

* Open Eclipse and create a Java class.
* Write a Java program.
* Convert the Java Program into TestNG.
* Open command prompt.
* Run the TestNG using command prompt.

## Command Syntex:

java org.testng.TestNG Path of XMl file

# Data Provider

## How do you use load elements through properties file and run using TestNG ?

We can create data provider and get data from properties file.

## How can we create data driven framework using TestNG?

By using @DataProvider annotation, we can create a Data Driven Framework.

// TestNG Interview Questions

@DataProvider(name="getData")

            Public Object [][] getData(){

                        //Object [][] data = new Object [rowCount][colCount];

                        Object [][] data = new Object [2][2];

                        data [0][0] = "FirstUid";

                        data [0][1] = "FirstPWD";

                        data[1][0] = "SecondUid";

                        data[1][1] = "SecondPWD";

                       return data;

          }

## What are @Factory and @DataProvider annotation?

@Factory: A factory will execute all the test methods present inside a test class using a separate instance of the respective class with different set of data.

@DataProvider: A test method that uses DataProvider will be executed the specific methods multiple number of times based on the data provided by the DataProvider. The test method will be executed using the same instance of the test class to which the test method belongs.

# Difference

## What is difference Between Assert and verify?

## Assert:

* When the “assert” command fails, the test execution will be canceled. So when the Assertion fails, then all test steps are skipped / omitted after that line of code.
* The solution to overcome this issue is to use a try-catch block.
* We use try-catch in the block.
* Mostly, the assurance order is used when the final result of the check value should be passed to continue in the next step.
* In simple terms, if the assert condition is true then the program control will execute the next testing phase, but if the situation is wrong, the execution will stop and further testing phase will not be executed.
* Failure of verification the execution of that particular test method is stopped and the test method has been marked as failed.
* When an “assert” fails, the test will be aborted / stopped.

## Verify:

* When the “Verify” command fails, then execution will continue and logging the failure.
* Mostly, the verification command is used to view non-critical things.
* In cases where we move forward, though the end result of the check value has failed.
* In simple words, there will be no interruption in test execution, even if the verification status is true or false.
* Verify has been verified using the SoftAssert class.

# Frameworks

## What is the approach for HybridFW you used to develop?

## 

# Listeners

## What is the use of Testng Listeners?

Listeners use for logs or customize TestNG reports in Selenium Webdriver

## What are different types of TestNG Listeners in Selenium?

* IAnnotationTransformer
* IAnnotationTransformer2
* IConfigurable
* IConfigurationListener
* IExecutionListener
* IHookable
* IInvokedMethodListener
* IInvokedMethodListener2
* IMethodInterceptor
* IReporter
* ISuiteListener
* ITestListener

## What is the use of @Listener annotation in TestNG?

TestNG listeners are used to configure reports and logging. One of the most widely used listeners in testNG is ITestListener interface. It has methods like onTestStart, onTestSuccess, onTestFailure, onTestSkipped etc. We should implement this interface creating a listener class of our own. Next we should add the listeners’ annotation (@Listeners) in the Class which was created.

## What is the diff b/w WebDriver Listeners and TestNG Listener?

## WebDriver listener are triggered when you call webdriver methods like findElement, click etc example-afterClickOn; beforeFindBy

WebDrier Event Listener is to listen the events triggered by webdriver like beforeClickOn, afterClickOn, beforeFindBy, afterFindBy, etc and take actions. It is mainly used to write log file for selenium test execution.

TestNG Listener is triggered when testng methods are executed such as  
 TestNG listener mainly used to generate the report for the test. Also, you can capture screenshot when there is test failure. TestNG events are like onTestFailure, onTestSkipped, onTestSuccess, etc.

## What is ITest result in selenium?

ITest result in selenium is an Listener API used to monitor the execution status of test

## What are Listeners in Selenium?

It is defined as an interface that modifies the behavior of the

system. [*Listeners*](https://www.edureka.co/blog/listeners-in-selenium/) allow customization of reports and logs.

Listeners mainly comprise of two types, namely

* WebDriver listeners
* TestNG listeners

# Maven

## What is maven life cycle?

* Maven Compile
* Maven test
* Maven Resource jar

Maven Compile:

Maven Compile Plugin

Maven Test:

Maven Test Happing through sure fire plugin

Maven Resource:

Generate Jar,war,ear file

## How to organize POM.xml or java file?

Ctr+Shift+F for organizie

## What is POM XML Maven?

A Project Object Model or **POM** is the fundamental unit of work in **Maven**. It is an **XML** file that contains information about the project and configuration details used by **Maven** to build the project. It contains default values for most projects.

## What are main 4 important selectors in POM.xml File?

* Project Selector
* Build Selector.
* Plugins Selector
* Configuration Selectors.

## After Update Sometime error coming in POM files how to fix it?

Just go to project right click go to mave and update it

## What difference between .xml & .html?

XML is a markup language which is designed to store data. It's popularly used or transfer of data. It is case sensitive. XML offers you to define markup elements and generate customized markup language. The basic unit in the XML is known as an element. Extension of XML file is .xml

HTML is the markup language which helps you to create and design web content. It has a variety of tag and attributes for defining the layout and structure of the web document. It is designed to display data in a formatted manner. A HTML document has the extension .htm or .html.

## **What kind of dependencies you have added in the Maven and for what?**

There is various kind of dependency we use in mave here is some example

* Java.
* Java Standalone.
* For Excel Poi API.
* Testeport.
* Log4j.

## How do we clean the Maven environment?

Mvn clean

## How do you run with compile maven Project?

Mvn clean Install.

## How do you run maven Project?

Mvn test.

# GIT

## What Is Git ?

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

## What is a repository in GIT?

A repository contains a directory named .git, where git keeps all of its metadata for the repository. The content of the .git directory are private to git.

## What Is GitHub?

Git hub is basically web service .GitHub brings together the world's largest community of developers to discover, share, and build better software. From open source projects to private team.90% time github give you repository 10 times its throw error in this scenario we need to do manually.

## How to Create a New Repository?

* In the upper-right corner of any page, click, and then click **new repository**.
* In the Owner drop-down, select the account you wish to **create** the **repository** on.
* Type a name for your **repository**, and an optional description.
* Choose to make the **repository** either public or private.

## ****How to**** [Tell Git who you are](https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-config)****?****

git config --global user.name [mdclaymont](mailto:mdclaymont@gmail.com)

git config --global user.email [mdclaymont@gmail.com](mailto:mdclaymont@gmail.com)

## How to initialize system that Git understand Folder?

Using following command we can initialized

* git init

It will create a hidden .git file in order to see that file you need to unhide file and folder.

## What Is Commit?

Commit is. A commit, or "revision", is an individual change to a file (or set

Of files). It's like when you save a file, except with Git, every time you save it creates a unique ID (a.k.a. the "SHA" or "hash") that allows you to keep record of what changes were made when and by whom

## How to Staging?

Git add \* command

## How to Commit?

git commit -m "This is my first commit"

## What is Origin?

Its remote repository.

## How to Push a Project to Git hub?

Go to Command in your system then use following line separately for initialization

First Got to project Directory Cd Path of project

Example:

Cd C:\Users\nyc\eclipse-workspace\Practise

* git init
* git add README.md
* git commit -m "first commit"
* git remote add origin https://github.com/mdclaymont/TestPrivate.git
* git push -u origin master

## How to fix this error "error: failed to push some refs to” URL"?

git reset --mixed origin/master

git add .

git commit -m "This is a new commit for what I originally planned to be amended"

git push origin master

## What is the difference between GIT and SVN?

The difference between GIT and SVN is

Git is less preferred for handling extremely large files or frequently changing binary files while SVN can handle multiple projects stored in the same repository.

GIT does not support ‘commits’ across multiple branches or tags.  Subversion allows the creation of folders at any location in the repository layout.

Gits are unchangeable, while Subversion allows committers to treat a tag as a branch and to create multiple revisions under a tag root.

## What are the advantages of using GIT?

* Data redundancy and replication
* High availability
* Only one.git directory per repository
* Superior disk utilization and network performance
* Collaboration friendly
* Any sort of projects can use GIT

## What language is used in GIT?

GIT is fast, and ‘C’ language makes this possible by reducing the overhead of runtimes associated with higher languages.

## What is the function of ‘GIT PUSH’ in GIT?

‘GIT PUSH’ updates remote refs along with associated objects.

## Why GIT better than Subversion?

GIT is an open source version control system; it will allow you to run ‘versions’ of a project, which show the changes that were made to the code overtime also it allows you keep the backtrack if necessary and undo those changes.  Multiple developers can checkout, and upload changes and each change can then be attributed to a specific developer.

## What is “Staging Area” or “Index” in GIT?

Before completing the commits, it can be formatted and reviewed in an intermediate area known as ‘Staging Area’ or ‘Index’.

## What is GIT stash?

GIT stash takes the current state of the working directory and index and puts in on the stack for later and gives you back a clean working directory.  So in case if you are in the middle of something and need to jump over to the other job, and at the same time you don’t want to lose your current edits then you can use GIT stash.

## What is GIT stash drop?

When you are done with the stashed item or want to remove it from the list, run the git ‘stash drop’ command.  It will remove the last added stash item by default, and it can also remove a specific item if you include as an argument.

## How will you know in GIT if a branch has been already merged into master?

Git branch—merged lists the branches that have been merged into the current branch

Git branch—-no merged lists the branches that have not been merged

## What is the function of git clone?

The git clone command creates a copy of an existing Git repository.  To get the copy of a central repository, ‘cloning’  is the most common way used by programmers.

## What is the function of ‘git config’?

The ‘git config’ command is a convenient way to set configuration options for your Git installation.  Behaviour of a repository, user info, preferences etc. can be defined through this command.

## What does commit object contain?

A set of files, representing the state of a project at a given point of time

Reference to parent commit objects

An SHAI name, a 40 character string that uniquely identifies the commit object.

## How can you create a repository in Git?

In Git, to create a repository, create a directory for the project if it does not exist, and then run command “git init”. By running this command .git directory will be created in the project directory, the directory does not need to be empty.

## What is ‘head’ in git and how many heads can be created in a repository?

A ‘head’ is simply a reference to a commit object. In every repository, there is a default head referred as “Master”.  A repository can contain any number of heads.

## What is the purpose of branching in GIT?

The purpose of branching in GIT is that you can create your own branch and jump between those branches. It will allow you to go to your previous work keeping your recent work intact.

## What is the common branching pattern in GIT?

The common way of creating branch in GIT is to maintain one as “Main“

Branch and create another branch to implement new features. This pattern is particularly useful when there are multiple developers working on a single project.

## How can you bring a new feature in the main branch?

To bring a new feature in the main branch, you can use a command “git merge” or “git pull command”.

## How to Create a Branch?

Using following command we can create a new branch

git checkout -b develop

## How To Check which branch You are?

## git branch

## How to Swithch to Main Branch?

git checkout master

## Command For change branch name locally?

Git checkout branchname.

## What is a ‘conflict’ in git?

A ‘conflict’ arises when the commit that has to be merged has some change in one place, and the current commit also has a change at the same place. Git will not be able to predict which change should take precedence.

## What is cherry pick?

Cherry picking is the act of picking a commit from a branch and applying it to another. git cherry-pick can be useful for undoing changes.

## How can conflict in git resolved?

To resolve the conflict in git, edit the files to fix the conflicting changes and then add the resolved files by running “git add” after that to commit the repaired merge,  run “git commit”.  Git remembers that you are in the middle of a merger, so it sets the parents of the commit correctly.

## To delete a branch what is the command that is used?

Once your development branch is merged into the main branch, you don’t

Need

Development branch.  To delete a branch use, the command “git branch –d [head]”.

## How to merge develop branch to master branch?

First go to master branch check up-to-date is master branch then use following command:

git pull origin master

If its up to date then use following command

git merge develop

## What is another option for merging in git?

“Rebasing” is an alternative to merging in git.

## What is the syntax for “Rebasing” in Git?

The syntax used for rebase is “git rebase [new-commit] “

## What is the difference between ‘git remote’ and ‘git clone’?

‘git remote add’  just creates an entry in your git config that specifies a name for a particular URL.  While, ‘git clone’ creates a new git repository by copying and existing one located at the URI.

## What is GIT version control?

With the help of GIT version control, you can track the history of a collection of files and includes the functionality to revert the collection of files to another version.  Each version captures a snapshot of the file system at a certain point of time. A collection of files and their complete history are stored in a repository.

## Mention some of the best graphical GIT client for LINUX?

Some of the best GIT client for LINUX is

Git Cola

Git-g

Smart git

Giggle

Git GUI

qGit

## What is Subgit? Why to use Subgit?

‘Subgit’ is a tool for a smooth, stress-free SVN to Git migration.  Subgit is a solution for a company -wide migration from SVN to Git that is:

It is much better than git-svn

No requirement to change the infrastructure that is already placed

Allows to use all git and all sub-version features

Provides genuine stress –free migration experience.

## What is the function of ‘git diff ’ in git?

‘git diff ’ shows the changes between commits, commit and working tree etc.

## What is ‘git status’ is used for?

As ‘Git Status’ shows you the difference between the working directory and the index, it is helpful in understanding a git more comprehensively.

## What is the difference between the ‘git diff ’and ‘git status’?

‘git diff’ is similar to ‘git status’, but it shows the differences between various commits and also between the working directory and index.

## What is the function of ‘git checkout’ in git?

A ‘git checkout’ command is used to update directories or specific files in your working tree with those from another branch without merging it in the whole branch.

## What is the function of ‘git rm’?

To remove the file from the staging area and also off your disk ‘git rm’ is used.

## What is the function of ‘git stash apply’?

When you want to continue working where you have left your work, ‘git stash apply’ command is used to bring back the saved changes onto the working directory.

## What is the use of ‘git log’?

To find specific commits in your project history- by author, date, content or history ‘git log’ is used.

## What is ‘git add’ is used for?

‘git add’ adds file changes in your existing directory to your index.

## What is the function of ‘git reset’?

The function of ‘Git Reset’ is to reset your index as well as the working directory to the state of your last commit.

## What is git Is-tree?

‘git Is-tree’ represents a tree object including the mode and the name of each item and the SHA-1 value of the blob or the tree.

## How git instaweb is used?

‘Git Instaweb’ automatically directs a web browser and runs webserver with an interface into your local repository.

## What does ‘hooks’ consist of in git?

This directory consists of Shell scripts which are activated after running the corresponding Git commands.  For example, git will try to execute the post-commit script after you run a commit.

## Explain what is commit message?

Commit message is a feature of git which appears when you commit a change. Git provides you a text editor where you can enter the modifications made in commits.

## How can you fix a broken commit?

To fix any broken commit, you will use the command “git commit—amend”. By running this command, you can fix the broken commit message in the editor.

## Why is it advisable to create an additional commit rather than amending an existing commit?

There are couple of reason

The amend operation will destroy the state that was previously saved in a commit.  If it’s just the commit message being changed then that’s not an issue.  But if the contents are being amended then chances of eliminating something important remains more.

Abusing “git commit- amend” can cause a small commit to grow and acquire unrelated changes.

## What is ‘bare repository’ in GIT?

To co-ordinate with the distributed development and developers team, especially when you are working on a project from multiple computers ‘Bare Repository’ is used. A bare repository comprises of a version history of your code.

Name a few Git repository hosting services

Pikacode

Visual Studio Online

GitHub

GitEnterprise

SourceForge.net

# GIT Command

|  |  |  |
| --- | --- | --- |
| Git task | Notes | Git commands |
| [**Tell Git who you are**](https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-config) | Configure the author name and email address to be used with your commits.  Note that Git [strips some characters](http://stackoverflow.com/questions/26159274/is-it-possible-to-have-a-trailing-period-in-user-name-in-git/26219423#26219423) (for example trailing periods) from user.name. | git config --global user.name "Sam Smith"  git config --global user.email sam@example.com |
| [**Create a new local repository**](https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-init) |  | git init |
| [**Check out a repository**](https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-clone) | Create a working copy of a local repository: | git clone /path/to/repository |
| For a remote server, use: | git clone username@host:/path/to/repository |
| [**Add files**](https://www.atlassian.com/git/tutorials/saving-changes#git-add) | Add one or more files to staging (index): | git add <filename>  git add \* |
| [**Commit**](https://www.atlassian.com/git/tutorials/saving-changes#git-commit) | Commit changes to head (but not yet to the remote repository): | git commit -m "Commit message" |
| Commit any files you've added with git add, and also commit any files you've changed since then: | git commit -a |
| [**Push**](https://www.atlassian.com/git/tutorials/syncing#git-push) | Send changes to the master branch of your remote repository: | git push origin master |
| [**Status**](https://www.atlassian.com/git/tutorials/inspecting-a-repository#git-status) | List the files you've changed and those you still need to add or commit: | git status |
| [**Connect to a remote repository**](https://www.atlassian.com/git/tutorials/syncing#git-remote) | If you haven't connected your local repository to a remote server, add the server to be able to push to it: | git remote add origin <server> |
| List all currently configured remote repositories: | git remote -v |
| [**Branches**](https://www.atlassian.com/git/tutorials/using-branches) | Create a new branch and switch to it: | git checkout -b <branchname> |
| Switch from one branch to another: | git checkout <branchname> |
| List all the branches in your repo, and also tell you what branch you're currently in: | git branch |
| Delete the feature branch: | git branch -d <branchname> |
| Push the branch to your remote repository, so others can use it: | git push origin <branchname> |
| Push all branches to your remote repository: | git push --all origin |
| Delete a branch on your remote repository: | git push origin :<branchname> |
| [**Update from the remote repository**](https://www.atlassian.com/git/tutorials/syncing) | Fetch and merge changes on the remote server to your working directory: | git pull |
| To merge a different branch into your active branch: | git merge <branchname> |
| View all the merge conflicts:  View the conflicts against the base file:  Preview changes, before merging: | git diff  git diff --base <filename>  git diff <sourcebranch> <targetbranch> |
| After you have manually resolved any conflicts, you mark the changed file: | git add <filename> |
| **Tags** | You can use tagging to mark a significant changeset, such as a release: | git tag 1.0.0 <commitID> |
| CommitId is the leading characters of the changeset ID, up to 10, but must be unique. Get the ID using: | git log |
| Push all tags to remote repository: | git push --tags origin |
| [**Undo local changes**](https://www.atlassian.com/git/tutorials/undoing-changes) | If you mess up, you can replace the changes in your working tree with the last content in head:  Changes already added to the index, as well as new files, will be kept. | git checkout -- <filename> |
| Instead, to drop all your local changes and commits, fetch the latest history from the server and point your local master branch at it, do this: | git fetch origin  git reset --hard origin/master |
| **Search** | Search the working directory for foo(): | git grep "foo()" |

## How to Delete Repository?

* Go to your account.
* Click on Repositories.
* Select your **Repository** (that you wants to **delete**)
* Click on settings tab.
* Go to "Danger Zone" blocks.
* Click on "**Delete** this **repository**" button.
* Type the **repository** name (that you wants to **delete**)
* Now click "I understand the consequences, **delete** this **repository**" button.