

## JENKINS FREE STYLE PROJECT-BATCH FILE

Step1: Download Jenkins war file from the URL: <https://jenkins.io/>

Step2: Download LTS Release

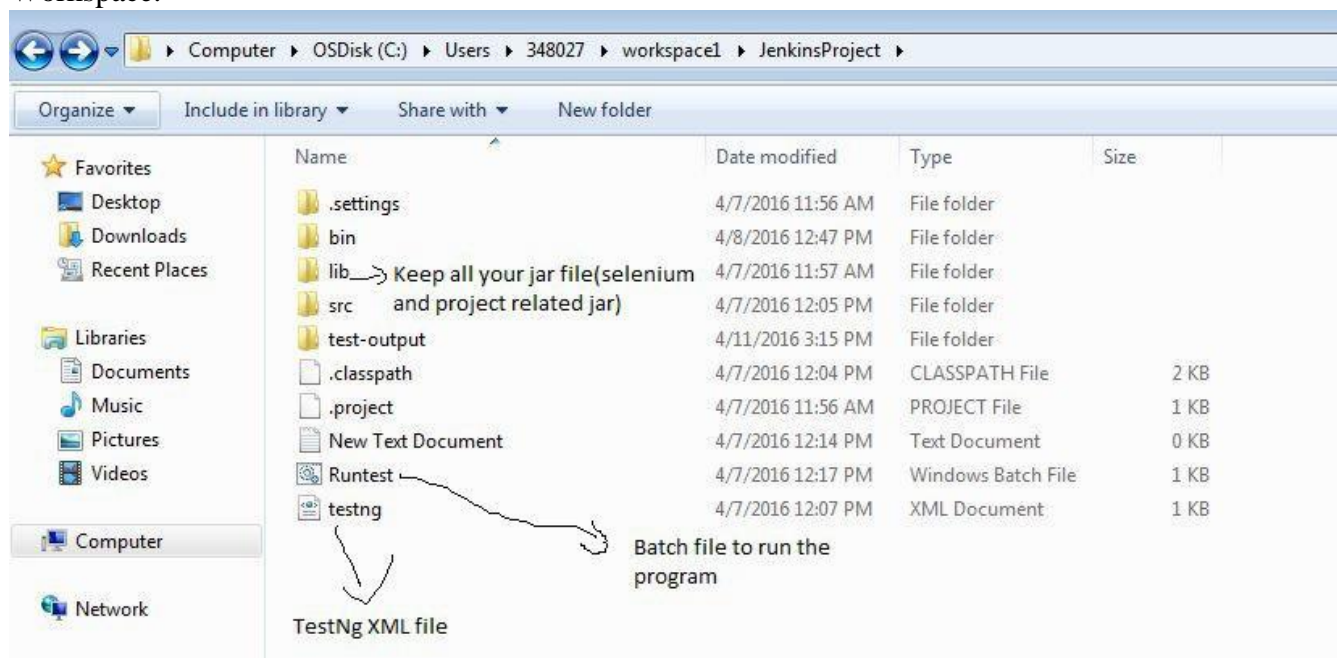


Step3: \*As like Java jar file. Run the Downloaded jenkins.war in the command prompt using the command. **Java -jar jenkins.war**

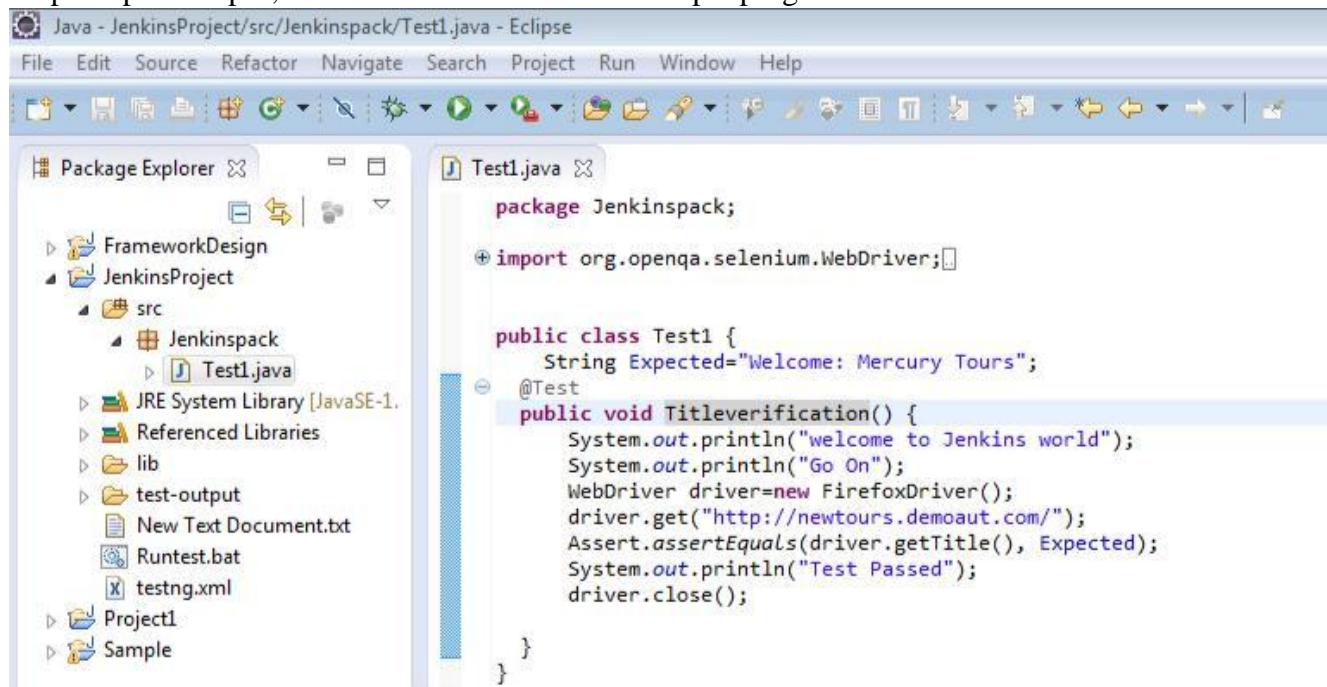
\*Wait till Jenkins is Up and Running. Don't ever close that Command prompt.

\*To check jenkins is working. Use the URL <http://localhost:8080> , it will open the jenkins environment. \*Now Jenkins platform ready to execute the program.

Step4: Open Eclipse Create a workspace. For best practices keep all the project files inside the Workspace.



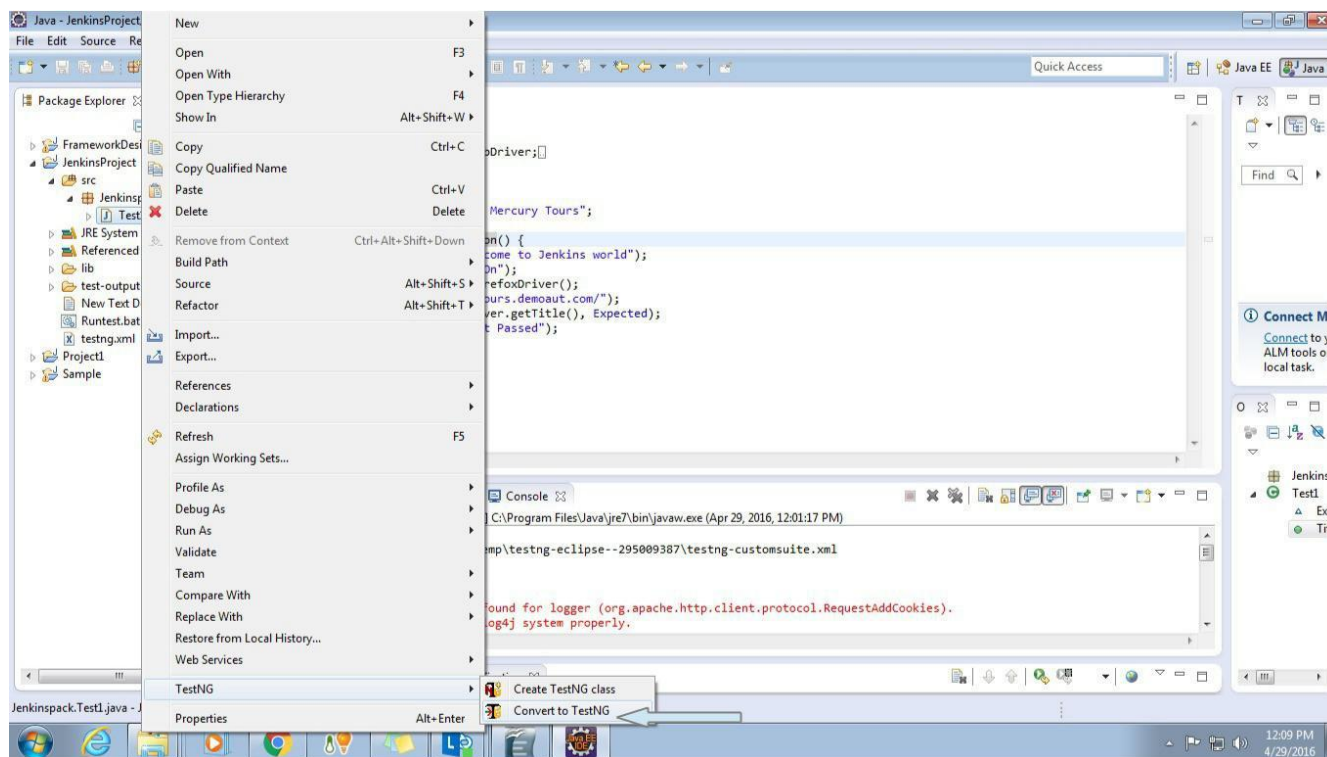
Step5: Open Eclipse, Create a class and write the simple program.



Two methods to run the above program. 1.Eclipse and 2. Command prompt.

To Run this program via Jenkins you need two files(Batch and TestNG.XML).

Step6: To Create xml file just right click the class file and go to TestNG ->Convert testng.



Step7: Save the XML inside your workspace.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-
1.0.dtd"> <suite name="Suite" parallel="none">
  <test name="Test">
    <classes>
      <class name="Jenkinspack.Test1"/>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

Step8: Now create the batch file inside your workspace.

`java -cp bin;lib/* org.testng.TestNG testng.xml` => Batch file content.

Note: while save, use double quotes("Runtest.bat") else it will store as text file.

Note:

cp: compiler.

bin: path of your compiled class.

lib: Contains the jar files,

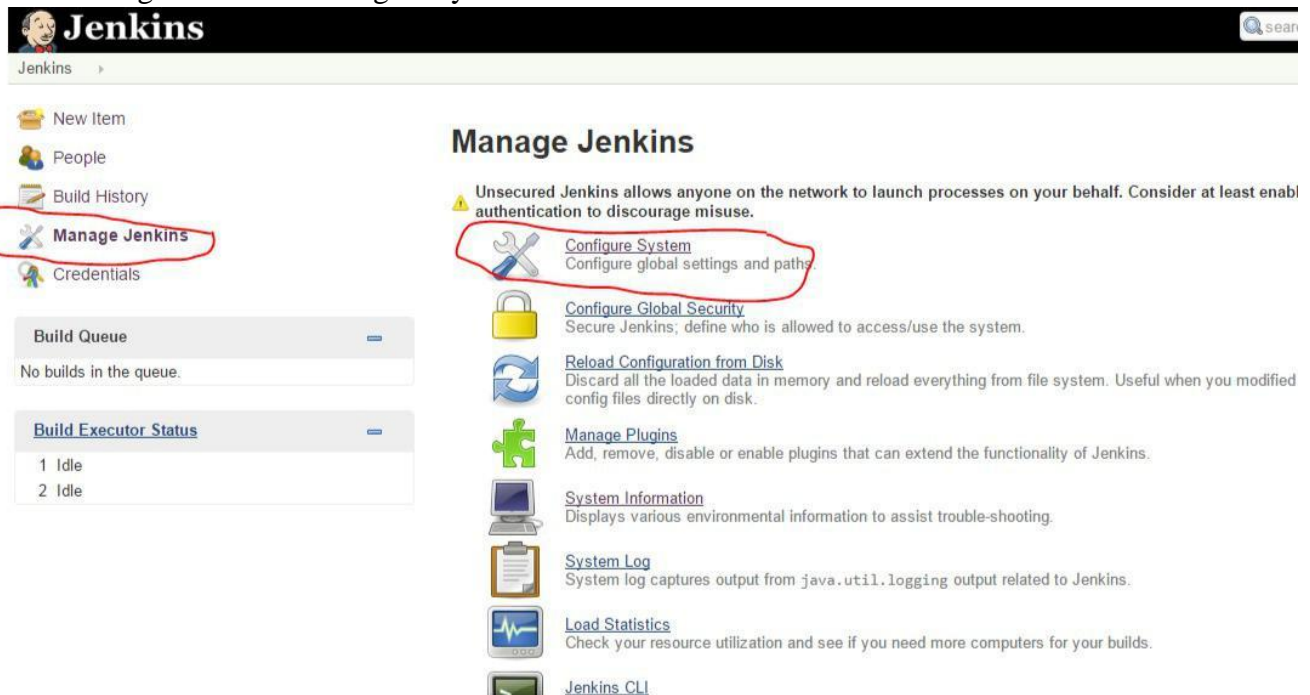
testng.xml: Your xml file.

Step9: Open Jenkins using the URL <http://localhost:8080>

The screenshot shows a Windows desktop with a command prompt window on the left and a web browser window on the right. The command prompt displays the output of the command `java -jar jenkins.war`, showing Jenkins starting up and listening on port 8080. The web browser shows the Jenkins dashboard at [localhost:8080](http://localhost:8080). The dashboard includes a search bar, a sidebar with links like 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'Credentials', and a main area with a 'Build Queue' (empty) and 'Build Executor Status' (showing 2 idle executors). A table lists the 'Jenkinsdemo' job with 'N/A' for last success, failure, and duration. RSS links for all, failures, and latest builds are also present.

S	W	Name ↓	Last Success	Last Failure	Last Duration
		Jenkinsdemo	N/A	N/A	N/A

\*Go to Manage Jenkins->Configure system.



**Jenkins**

Manage Jenkins

Unsecured Jenkins allows anyone on the network to launch processes on your behalf. Consider at least enabling authentication to discourage misuse.

[Configure System](#)  
Configure global settings and paths.

[Configure Global Security](#)  
Secure Jenkins; define who is allowed to access/use the system.

[Reload Configuration from Disk](#)  
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

[Manage Plugins](#)  
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

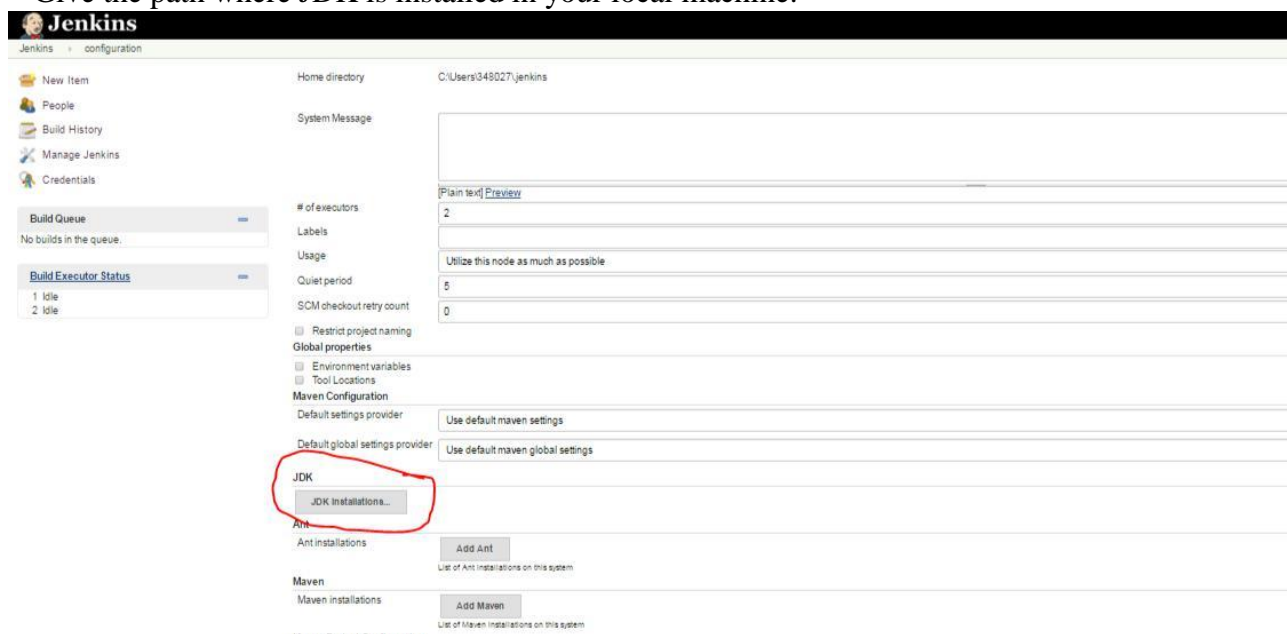
[System Information](#)  
Displays various environmental information to assist trouble-shooting.

[System Log](#)  
System log captures output from java.util.logging output related to Jenkins.

[Load Statistics](#)  
Check your resource utilization and see if you need more computers for your builds.

[Jenkins CLI](#)

Give the path where JDK is installed in your local machine.



**Jenkins**

configuration

Home directory: C:\Users\348027\jenkins

System Message

# of executors: 2

Labels

Usage: Utilize this node as much as possible

Quiet period: 5

SCM checkout retry count: 0

☐ Restrict project naming

Global properties

☐ Environment variables

☐ Tool Locations

Maven Configuration

Default settings provider: Use default maven settings

Default global settings provider: Use default maven global settings

**JDK**

JDK installations...

Ant

Ant installations: Add Ant

List of Ant installations on this system

Maven

Maven installations: Add Maven

List of Maven installations on this system

Click JDK installation and assign the path.



Default global settings provider: Use default maven global settings

---

**JDK**

JDK installations

JDK Name	JAVA_HOME
JAVA_HOME	C:\Program Files\Java\jdk1.7.0_80
<input type="checkbox"/> Install automatically	

Add JDK Delete JDK

List of JDK installations on this system

Don't check install automatically flag(because it will create the compatibility issue between java and Selenium). If you upgrade your JDK, manually replace the jdk path here. Then save it and back to home page.

Step10: Click New item to create a Jenkins project. Please do the below process.

Jenkins

Jenkins > All

**New Item**

People

Build History

Manage Jenkins

Credentials

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Item name: JenkinsDemoProject

☒ **Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used build.

☐ **Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

☐ **External Job**  
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can integrate with your existing automation system. See [the documentation for more details](#).

☐ **Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds.

OK

Step11: Link the Project Workspace to Jenkins. Give your workspace path and select Execute windows batch.

Jenkins: JenkinsDemoProject: configuration

[Delete Project](#)

[Configure](#)

**Build History** [trend](#)

[RSS for all](#) [RSS for failures](#)

☐ This build is parameterized

☐ Disable Build (No new builds will be executed until the project is re-enabled.)

☐ Execute concurrent builds if necessary

**Advanced Project Options**

☐ Quiet period

☐ Retry Count

☐ Block build when upstream project is building

☐ Block build when downstream project is building

☒ Use custom workspace

Directory:

Display Name:

☐ Keep the build logs of dependencies

**Source Code Management**

☒ None

☐ CVS

☐ CVS Projectset

☐ Subversion

**Build Triggers**

☐ Build after other projects are built

☐ Build periodically

☐ Poll SCM

**Build**

☒ Execute Windows batch command

☐ Execute shell

☐ Invoke Ant

☐ Invoke top-level Maven targets

\*Give the batch file name and save it.

☐ Poll SCM

**Build**

☒ Execute Windows batch command

Command:

[See the list of available environment variables](#)

**Post-build Actions**

Step12: Now your project is created.

New Item
 People
 Build History
 Manage Jenkins
 Credentials

All	+	S	W	Name ↓	Last Success
				JenkinsDemoProject	N/A

Icon: [S](#) [M](#) [L](#)

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Click the JenkinsDemoProject and press Build Now to start integration test.

Back to Dashboard
 Status
 Changes
 Workspace
 Build Now
 Delete Project
 Configure

Build History

find

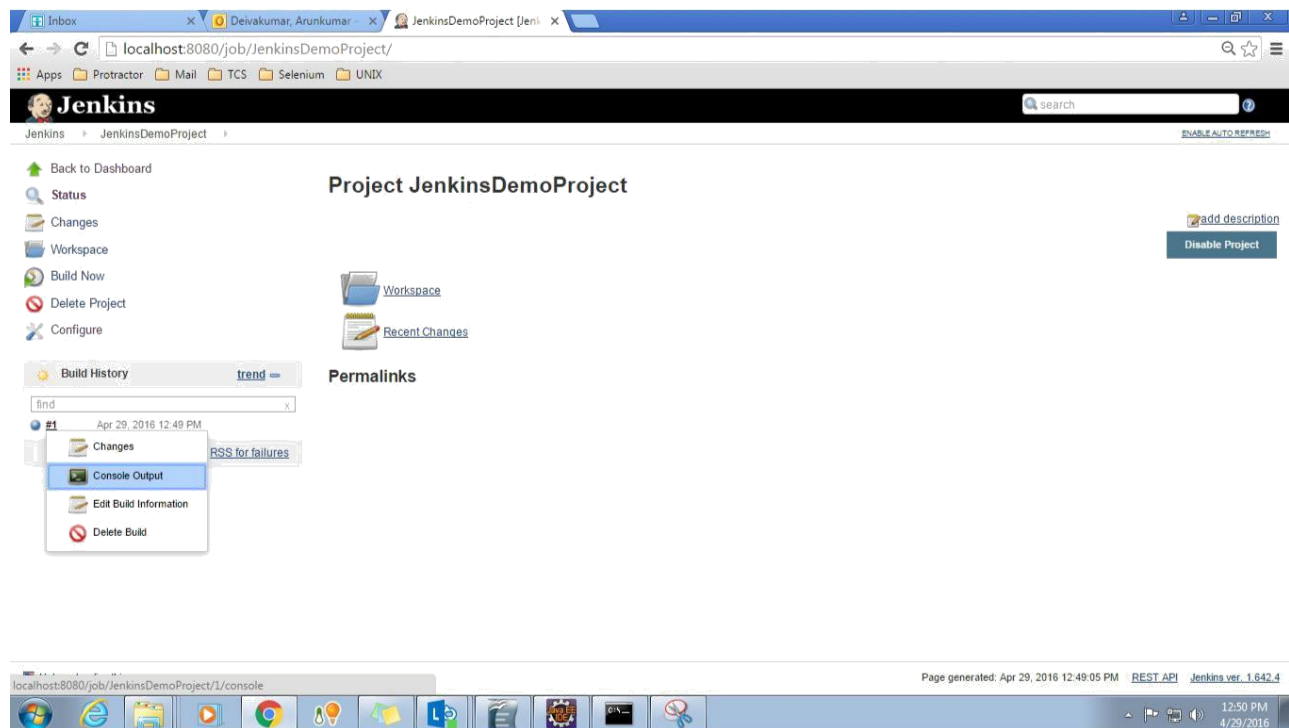
RSS for all
 RSS for failures

## Project JenkinsDemoProject

Workspace
 Recent Changes

### Permalinks

\* Congratulation your first integration test is ran successfully.



\*To see the output :go to Console output

- Back to Project
- Status
- Changes
- Console Output**
- View as plain text
- Edit Build Information
- Delete Build

## Console Output

```

Started by user anonymous
Building in workspace C:\Users\348027\workspace1\JenkinsProject
[JenkinsProject] $ cmd /c call C:\Users\348027\AppData\Local\Temp\hudson3355415362619941963.bat

C:\Users\348027\workspace1\JenkinsProject>RunTest.bat

C:\Users\348027\workspace1\JenkinsProject>java -cp bin;lib/* org.testng.TestNG testng.xml
[TestNG] Running:
  C:\Users\348027\workspace1\JenkinsProject\testng.xml

welcome to Jenkins world
Go On
Test Passed

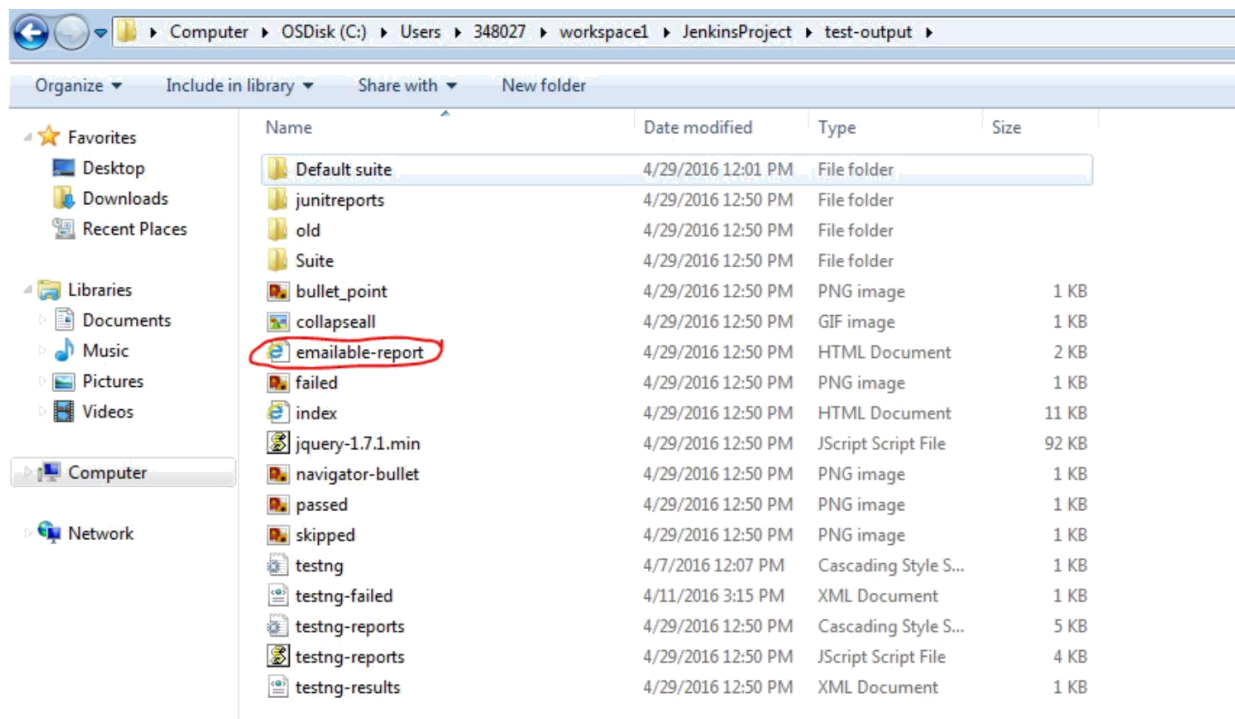
=====
Suite
Total tests run: 1, Failures: 0, Skips: 0
=====

Finished: SUCCESS

```

For Testng report go to the path: Workspace\test-output





TestNG Report

Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Suite						
<a href="#">Test</a>	1	0	0	15,123		

Class	Method	Start	Time (ms)
Suite			
Test — passed			
Jenkinspack Test1	<a href="#">Titleverification</a>	1461914404542	15030

## Test

### Jenkinspack.Test1#Titleverification

[back to summary](#)

**Thank You**

Arun Kumar  
D