
SmartCliff
CAREER MOBILITY SOLUTIONS
We are on a mission to address the digital skills gap for 10 Million+ young professionals, train and empower them to forge a career path into future tech

SmartCliff
Career Mobility Solutions

Maven Basics

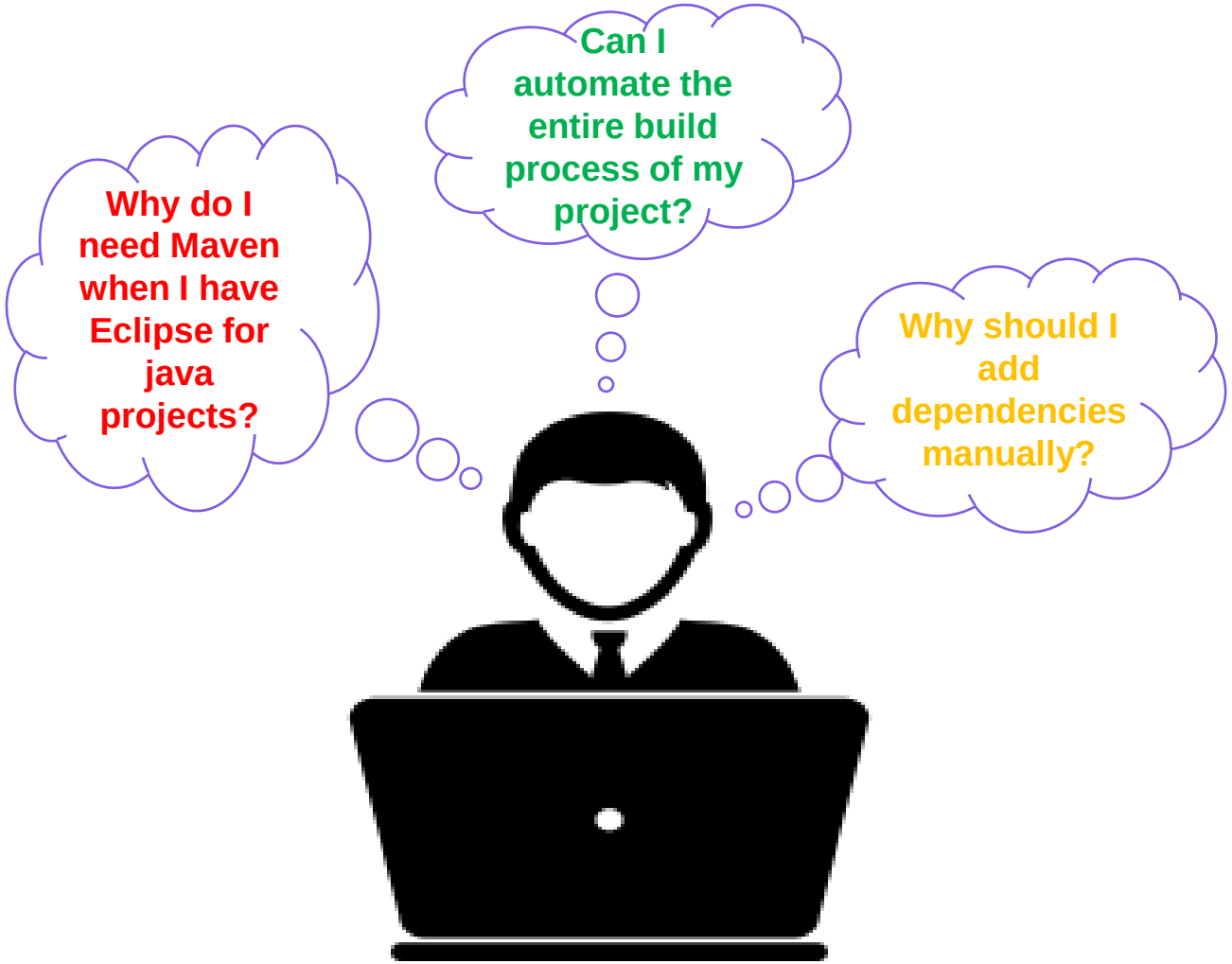
BUILD LIFE CYCLE, PLUGINS, RUN MAVEN BUILDS, COMPILING AND EXECUTING

Contents

What is Maven?

- Maven is an open-source **build automation tool and project management framework** primarily used for **Java-based projects**.
- It was developed by the **Apache Software Foundation .**
- It **simplifies** the entire **build process by automating tasks** such as:
  - Compiling source code
  - Running tests
  - Packaging code into distributable formats like JAR or WAR
  - Managing project dependencies

Why do we need Maven?



Key Features of Maven

1. **Project Object Model (POM)**:
   - Centralized **XML file (pom.xml)** that contains **project configuration, dependencies, plugins, and build instructions.**
2. **Dependency Management**:
   - Automatically **downloads and manages external libraries** and **dependencies from remote repositories** like **Maven Central.**
3. **Build Lifecycle**:
   - Provides a standard **set of build phases (e.g., compile, test, package, deploy).**
4. **Plugin System**:
   - Extensible through a **wide range of plugins** for tasks like code compilation, testing, packaging, and deployment.
5. **Multi-Module Support**:
   - Simplifies management of large projects with **multiple submodules**.

Build Life Cycle

- **Core concept** that outlines the **sequence of phases and goals** required to **build and manage** a Maven project
- The process for **building and distributing a particular artifact (project)** is clearly defined.
- Maven provides three built-in build life cycles
    - **Clean Lifecycle:** Focuses on **cleaning the project (removing previous build files).**
    - **Default (Build) Lifecycle:** Handles the **complete build process, from validation to deployment**.
    - **Site Lifecycle:** Manages the **creation and deployment of project documentation**.

Build Phases

- The **default-lifecycle** executes the following **build phases** in sequential order:

  1. **Validate :** Ensures that the project is correct and all necessary information is available
  2. **Compile :** Compiles the source code
  3. **Test :** Runs **unit tests** using a framework like JUnit
  4. **Package** – Packs the compiled code into a distributable format, such as a **JAR,WAR**
  5. **Verify** – Runs checks to ensure quality criteria are met
  6. **Install** – Installs the packaged application in the local Maven repository.
  7. **Deploy** – Deploys the application to a remote repository.

Plugin Goals

- A **build phase** in Maven is responsible for a **specific step in the build lifecycle.**
- The way a **build phase performs its tasks** can vary **depending on its implementation**.
- This variation is achieved **by associating plugin goals** with the build phases.
- One **plugin goal** can be bound to none, one or more than one phase.
- You can call a goal that is not linked to a build phase by mvn dependency:copy-dependencies.

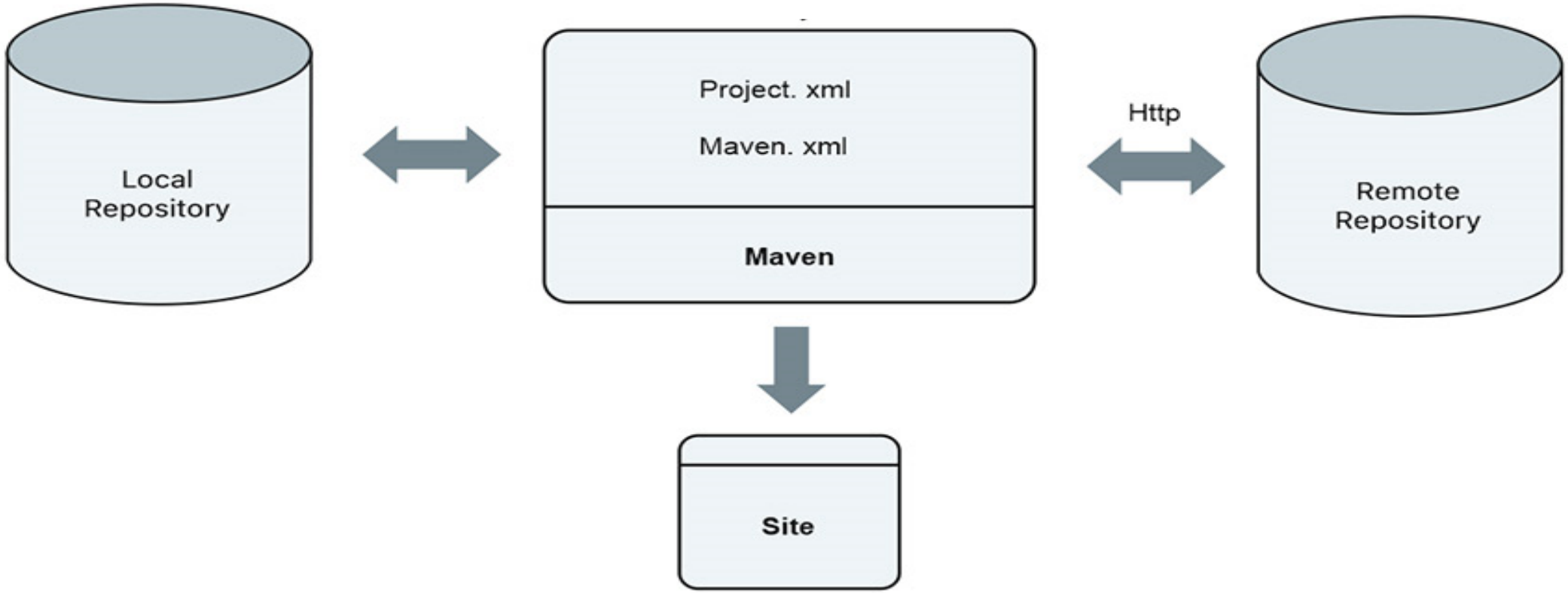
Plugin Goals

- Another way to **add goals** to phases is to **configure plugins** in your project.
- Plugins are **artifacts** that **provide goals to Maven.**
- A plugin may have **one or more goals** wherein each goal represents a capability of that plugin.
- For example, the Compiler plugin has two goals: **compile and testCompile**. The former compiles the source code of your main code, while the latter compiles the source code of your test code.
- There are basically
    - **Build plugins** are responsible for the build process (configurable in the **<build>** element of the POM)
    - **Reporting plugins** are responsible for the site generation (configurable in the **<reporting>** element of the POM)

Maven Repositories

- Maven searches the **repositories** for **dependencies in sequence**.
- First in the **local repository**, then in **remote repositories** if specified in the POM.
- The **local repository** is a **directory on the computer** where Maven runs. It caches remote downloads and contains temporary build artifacts that you have not yet released.
- The **remote repositories** refer to any other type of repository, accessed by a variety of protocols such as **file:// and https://.**
- Other **"remote" repositories** may be internal repositories set up on a **file or HTTP server within your company**, used to share **private artifacts between development teams and for releases.**

Maven Architecture



Environment Setup

- **Prerequisites**
  - Maven is written in Java. So, to run Maven, we need a system that has Java installed and configured properly.
  - To check, we will run the command below in CMD to get the currently installed version

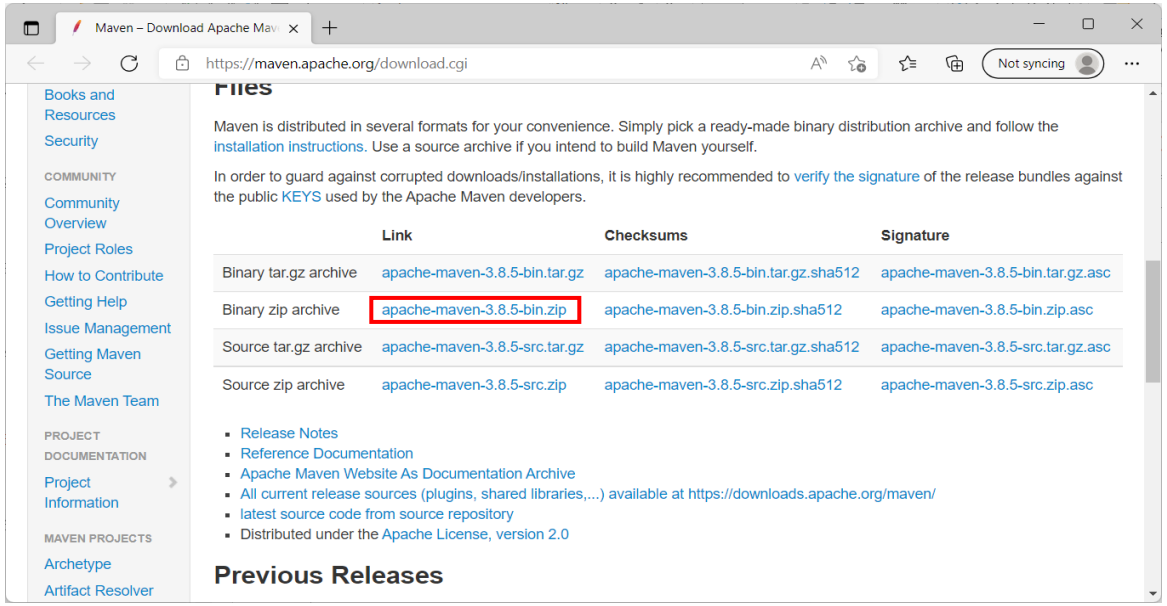
```
Select C:\WINDOWS\system32\cmd.exe
(c) Microsoft Corporation. All rights reserved.

C:\Users\SHEEBA>java -version
java version "17.0.2" 2022-01-18 LTS
Java(TM) SE Runtime Environment (build 17.0.2+8-LTS-86)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.2+8-LTS-86, mixed mode, sharing)

C:\Users\SHEEBA>
```

Environment Setup

- Download the Maven binary Archive from the link https://maven.apache.org/download.cgi



- Unzip the downloaded archive into the folder **C:\Program Files\Maven.** Copy the path to the clipboard.

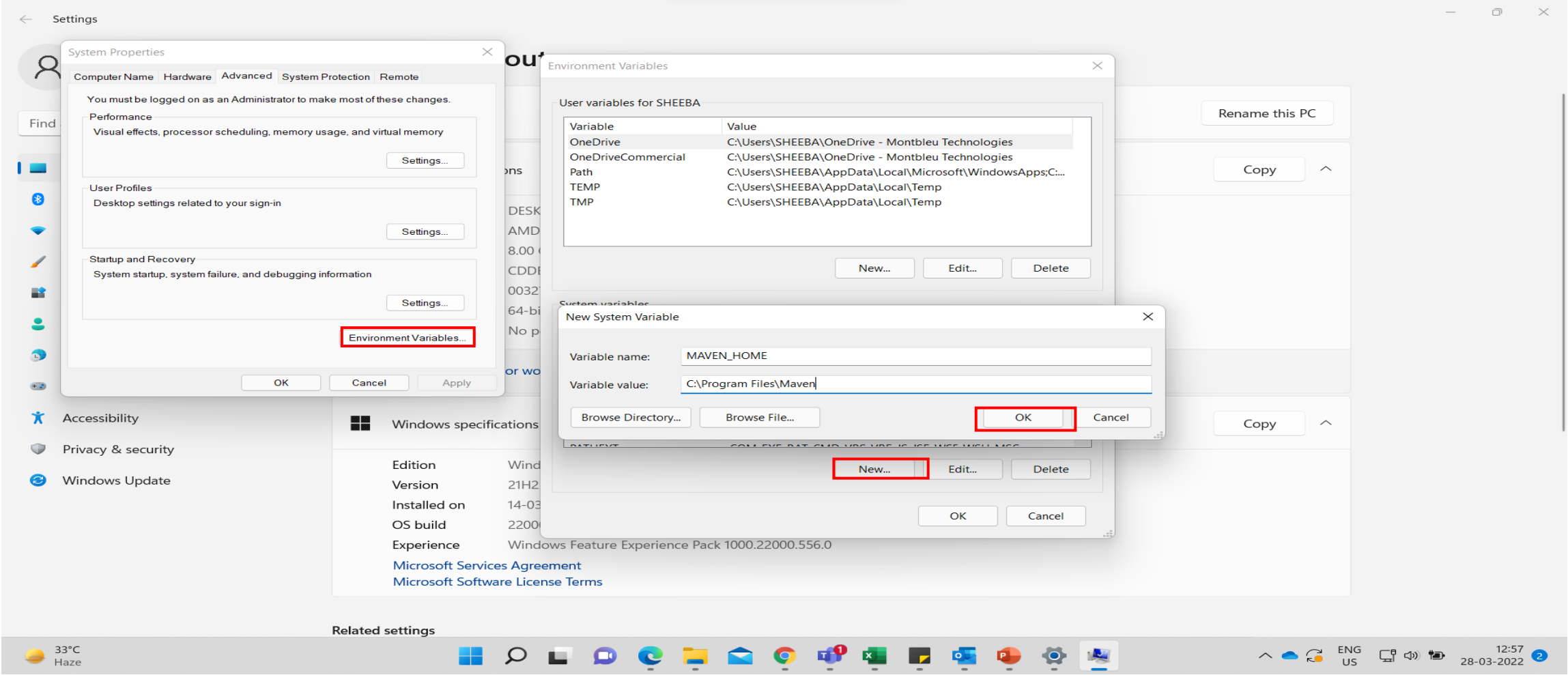
You can move the folder to any location of your choice, but it would be more convenient to keep it in the mentioned folder.

Environment Setup

Environment variable setting:

- Right-click on the Windows Start button and select System in the menu.
- In the new window, click on the ‘Advanced System Settings’.
- Click on the ‘Environment Variables’ in the pop-up window.
- Click on ‘New’ in the new pop-up window in the System variables.
- Enter MAVEN_HOME (all characters in capital case) in the Variable name field.
- Paste the copied earlier path to the Maven folder in the Variable value. Click on the OK button.

Environment Setup

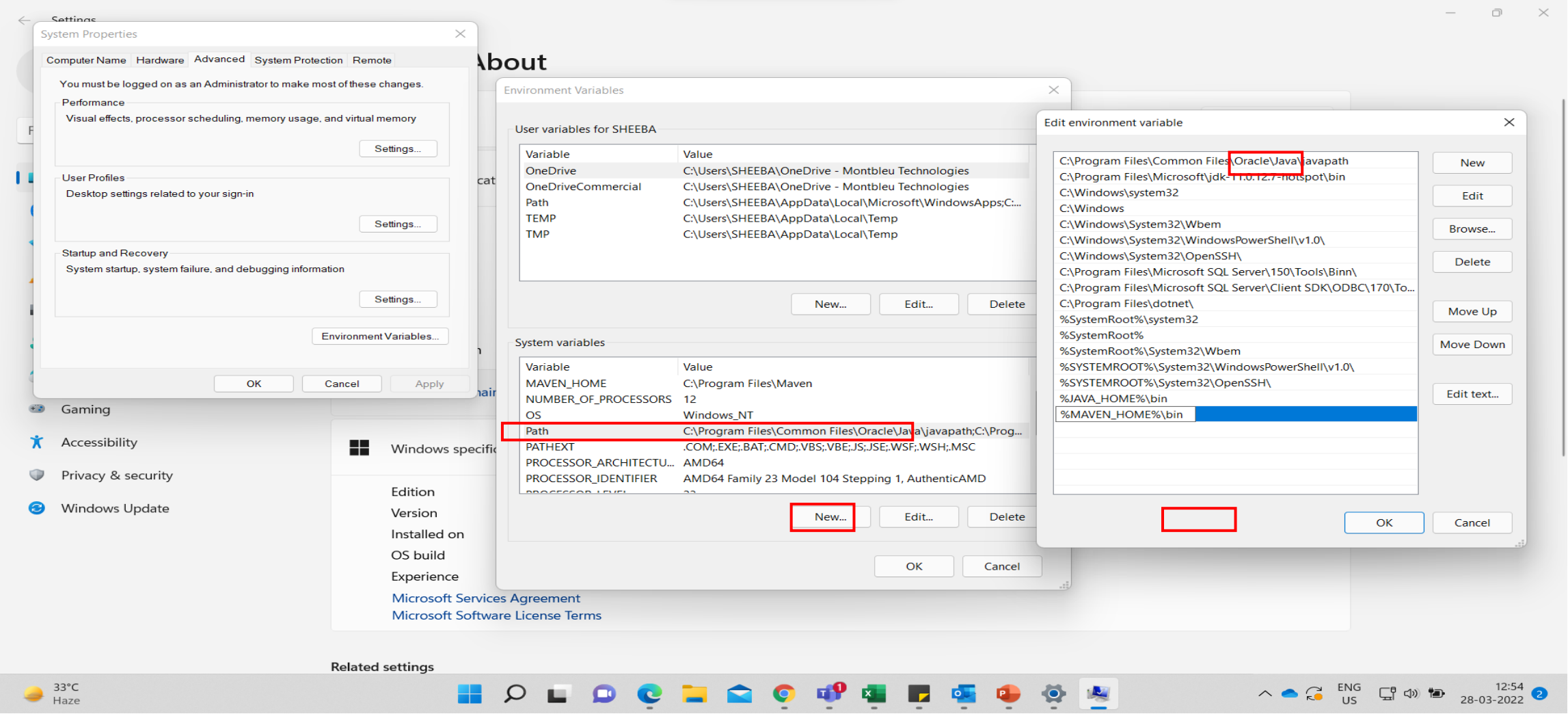


Environment Setup

Environment variable setting:

- Select the Path variable and click on Edit.
- In the new window, click on New.
- In the new field, enter %MAVEN_HOME%\bin.
- After that, click OK in all the previously opened windows.

Environment Setup



Environment Setup

- To verify the proper installation of Maven, we will run the command below in CMD to get the currently installed version

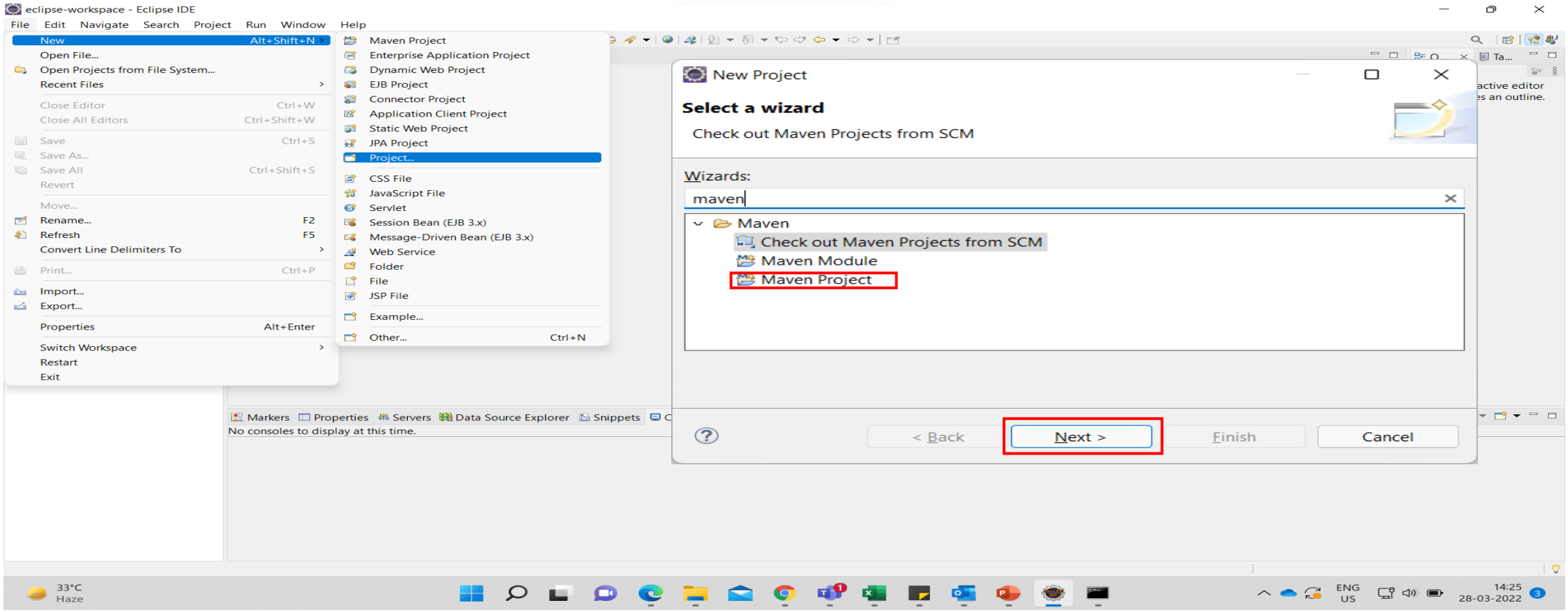
```
Select C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SHEEBA>mvn -version
Apache Maven 3.8.5 (3599d3414f046de2324203b78ddcf9b5e4388aa0)
Maven home: C:\Program Files\Maven
Java version: 17.0.2, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-17.0.2
Default locale: en_IN, platform encoding: Cp1252
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"

C:\Users\SHEEBA>
```

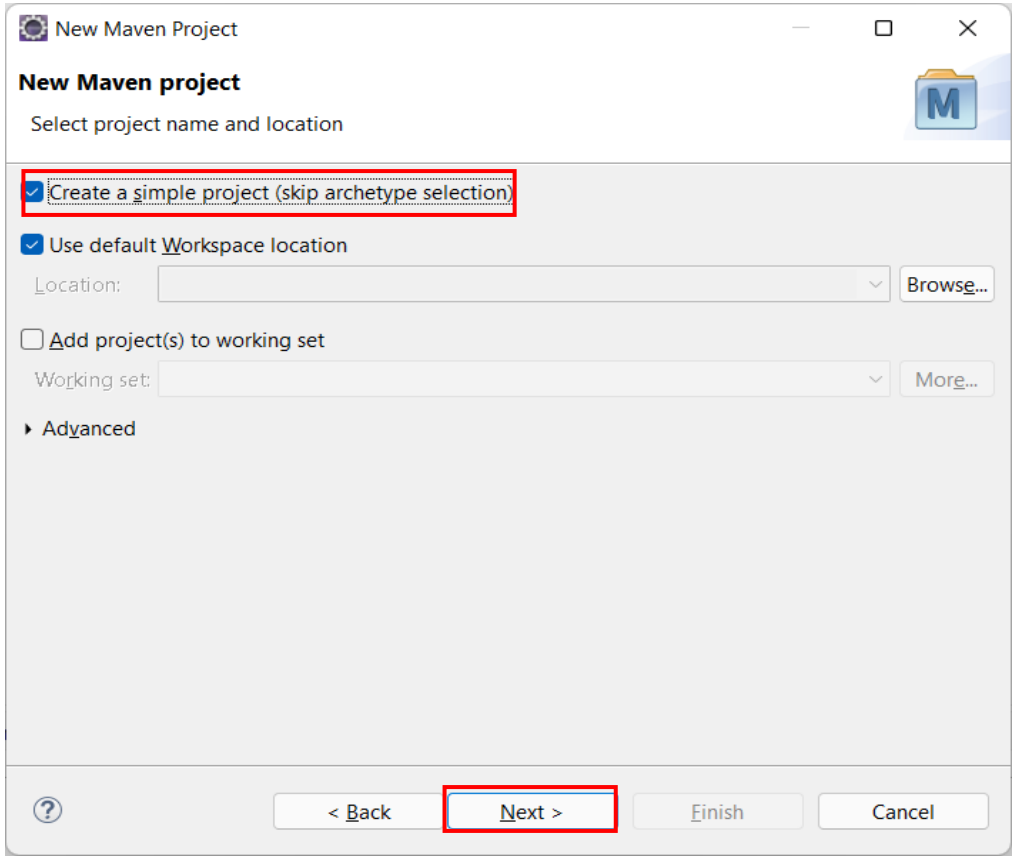
Demo - Using Maven with Eclipse IDE

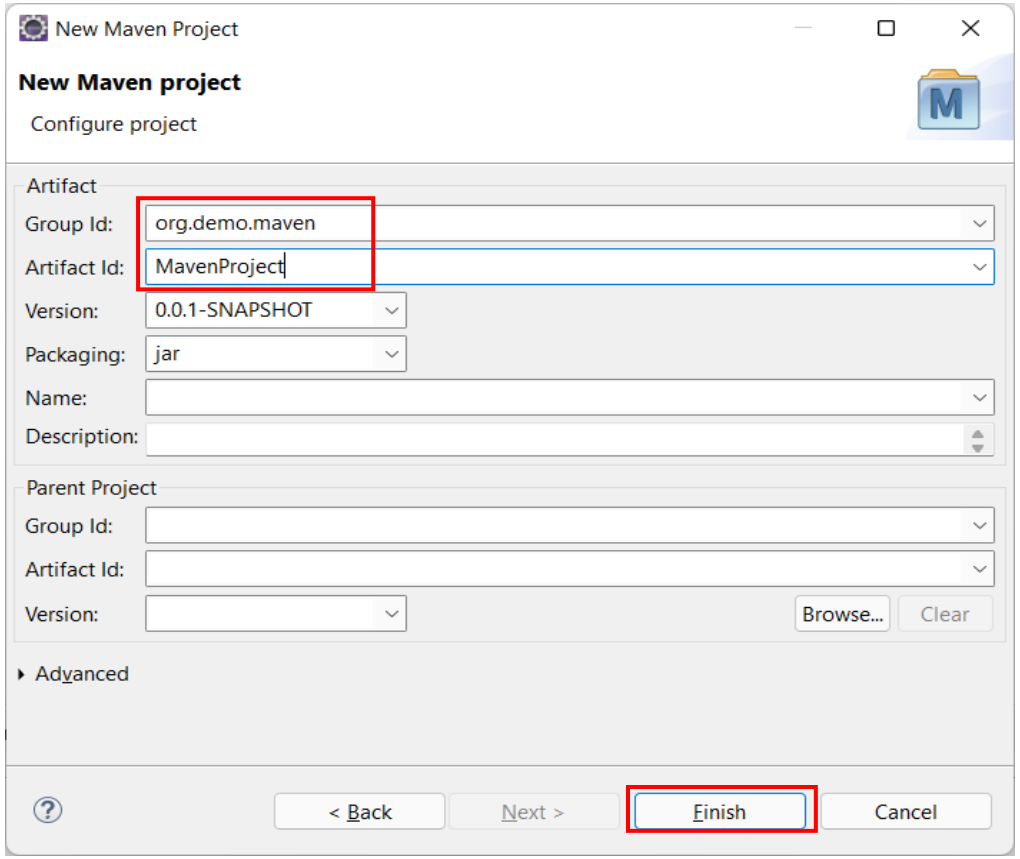
Create the new project -> Maven Project



Demo - Using Maven with Eclipse IDE

Change/Provide the required configuration details

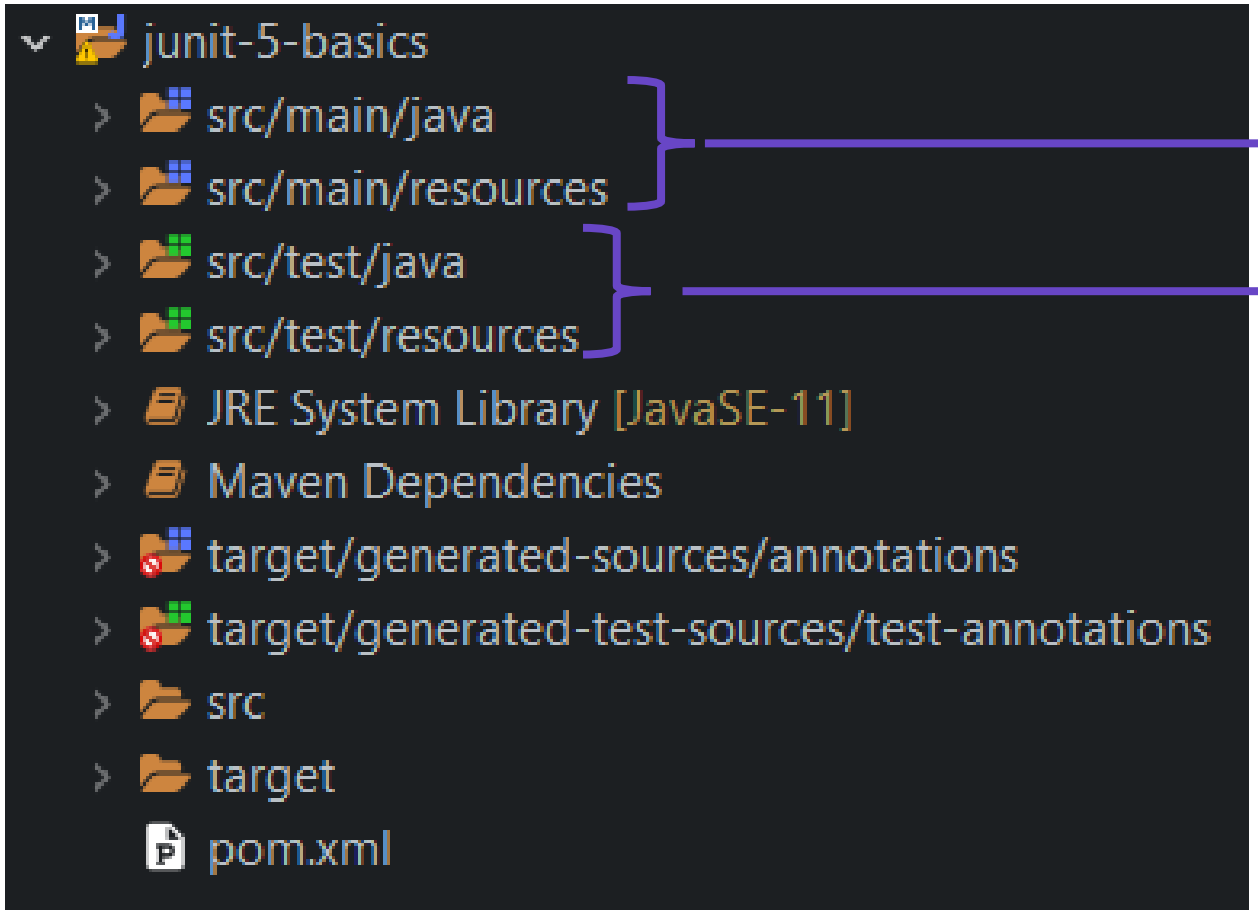




Guide to naming conventions on groupId, artifactId, and version

- **groupId** uniquely **identifies** your **project across all projects**. A group ID should follow Java's package name rules. This means it starts with a reversed domain name you control.
  - For **example**, org.apache.maven, org.apache.commons
- **artifactId** is the **name of the jar without version**. If you created it, then you can choose whatever name you want with lowercase letters and no strange symbols. If it's a third party jar, you must take the name of the jar as it's distributed.
  - **Example**: maven, commons-math
- version if you distribute it, then you can choose any typical version with numbers and dots (1.0, 1.1, 1.0.1, ...). Don't use dates .
- For more details look: https://maven.apache.org/guides/mini/guide-naming-conventions.html

Demo – Understand the Project Structure



Java Source and Resources(like images…) Package

Testing Source and Resources Package

Project Object Model(POM)

Configuration of a Maven project is done via **pom.xml** configuration file.

The POM file defines:

- **identifiers** for the project to be build
- **properties** relevant for **build configuration**
- **plugins** which provide functionality for **the build via a build section.**
- **library and project dependencies** via the dependencies section

pom.xml


Project Explorer
MavenDemo
src/main/java
src/test/java
JRE System Library [J2SE-1.5]
Maven Dependencies
junit-3.8.1.jar - C:\Users\
target
pom.xml
MavenDemo/pom.xml
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http:/
2 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven
3 <modelVersion>4.0.0</modelVersion>
4
5 <groupId>training</groupId>
6 <artifactId>MavenDemo</artifactId>
7 <version>0.0.1-SNAPSHOT</version>
8 <packaging>jar</packaging>
9
10 <name>MavenDemo</name>
11 <url>http://maven.apache.org</url>
12
13 <properties>
14 <project.build.sourceEncoding>UTF-8</project.build.sourceEncodin
15 </properties>
16
17 <dependencies>
18 <dependency>
19 <groupId>junit</groupId>
20 <artifactId>junit</artifactId>
21 <version>3.8.1</version>
22 <scope>test</scope>
23 </dependency>
24 </dependencies>

Dependencies

- Get it from: https://mvnrepository.com/
- In search box -------→search selenium---→ under selenium java--→3.13.0(Any stable version)
- In search box -------→search testing--→ under testing-→6.14.3(Any stable version)

Getting Selenium Java dependency

Maven Repository: org.seleniumh | Google

mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java/3.13.0

Selenium Java » 3.13.0

Selenium automates browsers. That's it! What you do with that power is entirely up to you.

License	Apache 2.0
Categories	Web Testing
HomePage	http://www.seleniumhq.org/
Date	(Jun 25, 2018)
Files	pom (3 KB) jar (293 bytes) View All
Repositories	Central Spring Plugins
Used By	1,356 artifacts

Note: There is a new version for this artifact

New Version	4.0.0-beta-4

Maven | Gradle | SBT | Ivy | Grape | Leiningen | Buildr

```
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>3.13.0</version>
</dependency>
```

☑ Include comment with link to declaration

Copied to clipbaord!

Popular Categories

Aspect Oriented
Actor Frameworks
Application Metrics
Build Tools
Bytecode Libraries
Command Line Parsers
Cache Implementations
Cloud Computing
Code Analyzers
Collections
Configuration Libraries
Core Utilities
Date and Time Utilities
Dependency Injection
Embedded SQL Databases
HTML Parsers
HTTP Clients
I/O Utilities
JDBC Extensions
JDBC Pools

Central
Sonatype
Spring Plugins
Spring Lib M
Hortonworks
JCenter
Atlassian
JBossEA
BeDataDriven
JBoss Releases

Popular Tags

android apache api application assets aws build build-system camel client clojure cloud config data database eclipse example extension framework github gradle groovy http integration io jboss library logging maven module osgi persistence platform plugin repository rest rlang scala sdk security server service spring starter streaming testing tools ui web webapp

Web site developed by @frodriguez

Powered by: Scala, Play, Spark, Akka and Cassandra

Type here to search

64 ENG IN 11:08 AM 6/19/2021

Paste dependencies in pom.xml

eclipse-workspace - MavenProject/pom.xml - Eclipse

File Edit Source Navigate Search Project Run Window Help

Package Explorer

- MavenProject
  - src/main/java
  - src/main/resources
  - src/test/java
  - src/test/resources
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
  - target
  - pom.xml

Testcase.java | *MavenProject/pom.xml

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/PO
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.ddfProj</groupId>
  <artifactId>MavenProject</artifactId>
  <version>0.0.1-SNAPSHOT</version>


  <dependencies>


  <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>3.13.0</version>
</dependency>


  </dependencies>


</project>
```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

Console | Results of running class Testcase

No consoles to display at this time.

Writable | Smart Insert | 9 : 3

Plugins

- Maven is a **plugin execution framework** where every task is actually done by plugins.
- Maven Plugins are generally used to
  - create jar file
  - create war file
  - compile code files
  - unit testing of code
  - create project documentation
  - create project reports

Plugin

Surefire plugin

https://maven.apache.org/surefire/maven-surefire-plugin/usage.html

Clean plugin

https://maven.apache.org/plugins/maven-clean-plugin/plugin-info.html

Compiler plugin

https://maven.apache.org/plugins/maven-compiler-plugin/plugin-info.html

Jar plugin

https://maven.apache.org/plugins/maven-jar-plugin/plugin-info.html

Adding Plugins

eclipse-workspace - MavenProject/pom.xml - Eclipse

File Edit Source Refactor Navigate Search Project Run Design Window Help

Package Explorer

- DataDrivenFramework
- MavenProject
  - src/main/java
  - src/main/resources
  - src/test/java
  - src/test/resources
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
  - target
  - test-output
  - pom.xml

*MavenProject/pom.xml | *DataDrivenFramework/pom.xml

```
    </dependencies>
    <build>
        <pluginManagement>

            <plugins>
                <plugin>
                    <artifactId>maven-clean-plugin</artifactId>
                    <version>3.1.0</version>
                </plugin>
                <plugin>
                    <artifactId>maven-compiler-plugin</artifactId>
                    <version>3.8.0</version>
                </plugin>
                <plugin>
                    <artifactId>maven-surefire-plugin</artifactId>
                    <version>2.22.1</version>
                </plugin>
                <plugin>
                    <artifactId>maven-jar-plugin</artifactId>
                    <version>3.0.2</version>
                </plugin>
            </plugins>
        </pluginManagement>
    </build>

</project>
```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

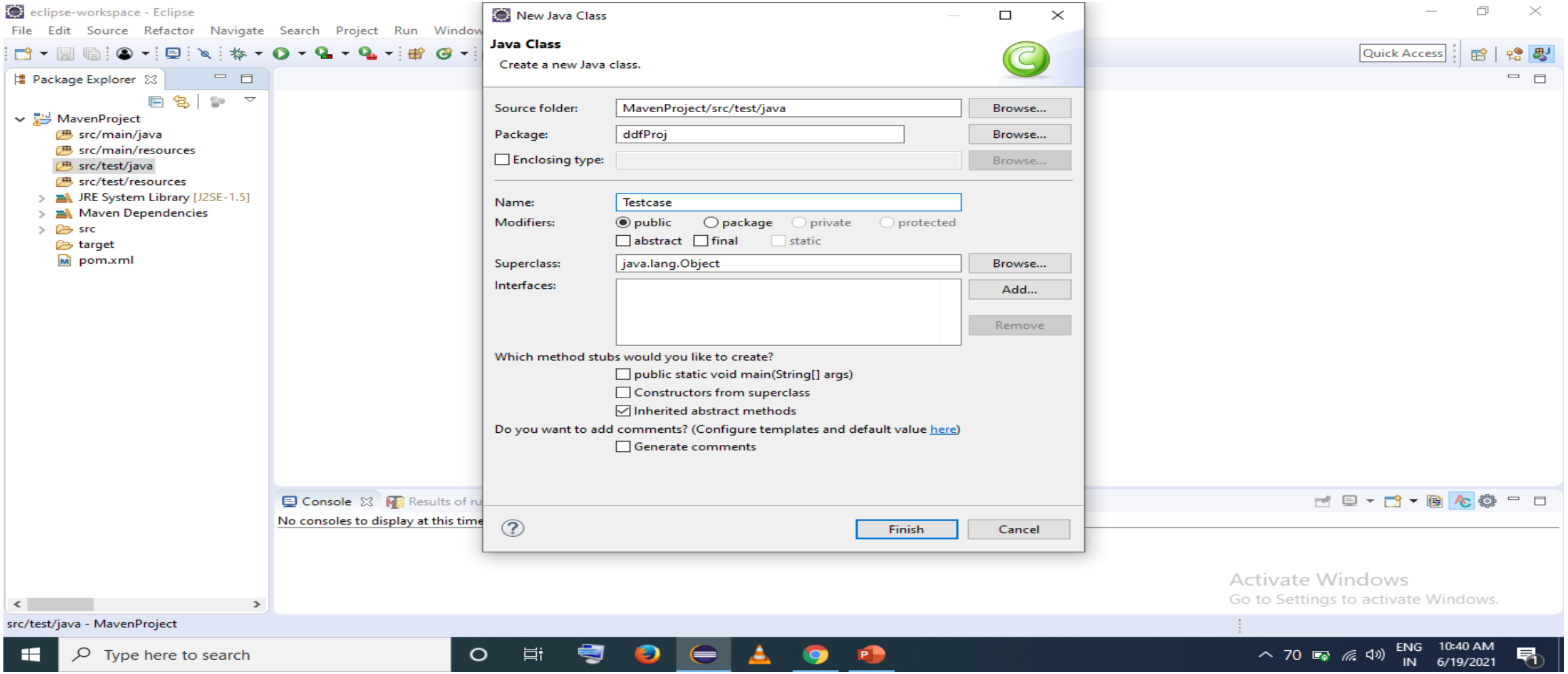
Console

No consoles to display at this time.

Type here to search

ENG US 7:13 PM 6/19/2021

Creating Testcase



Creating Testcase

eclipse-workspace - MavenProject/src/test/java/Testcase.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- MavenProject
  - src/main/java
  - src/main/resources
  - src/test/java
  - src/test/resources
  - JRE System Library [JavaSE-1.6]
  - Maven Dependencies
  - src
  - target
  - test-output
  - pom.xml

MavenProject/pom.xml | *Testcase.java

```java
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;

public class Testcase {

    @Test
    public static void website() {

        System.setProperty("webdriver.chrome.driver", "D:\\selenium complete\\Drivers\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("http://www.google.com");
        driver.quit();

    }

}
```

Console

No consoles to display at this time.

Activate Windows
Go to Settings to activate Windows.

Type here to search

ENG IN 12:22 PM 6/19/2021

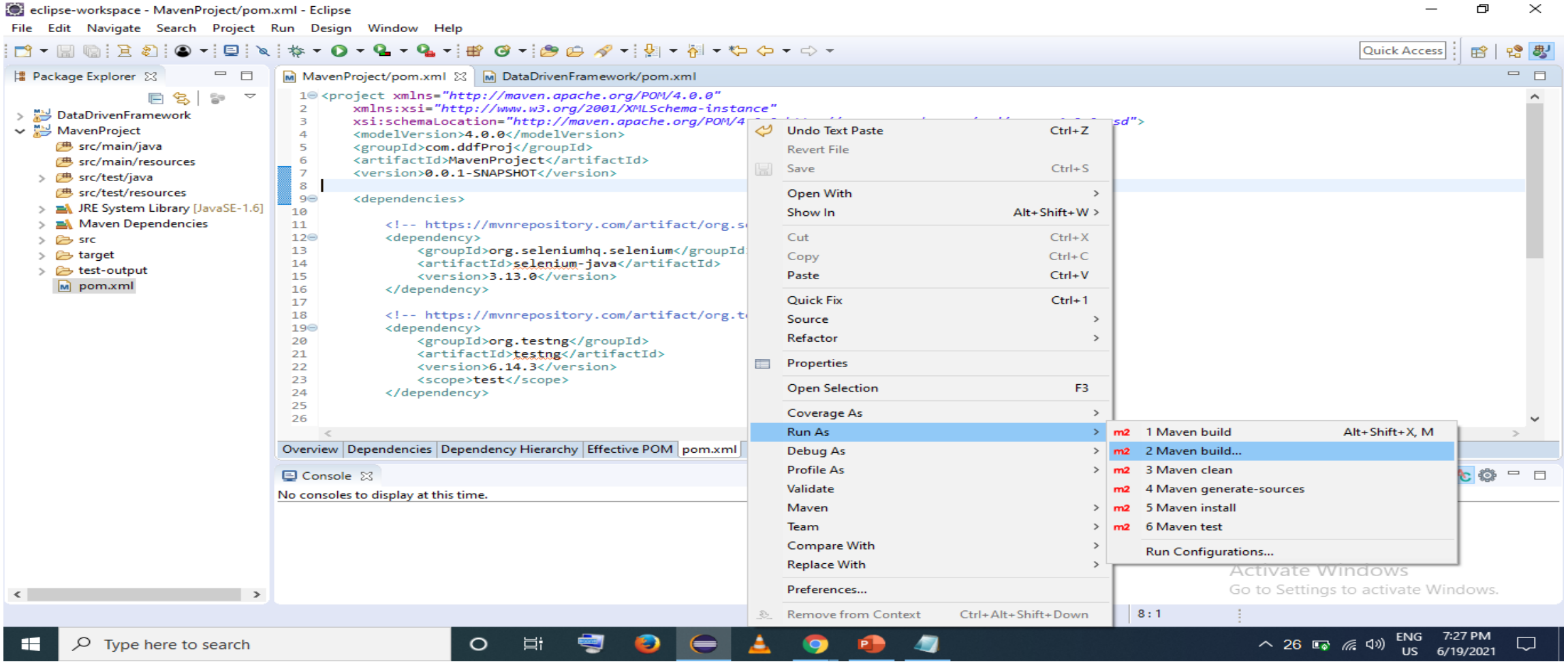
Maven Build

Using Maven Build we can use the following command:

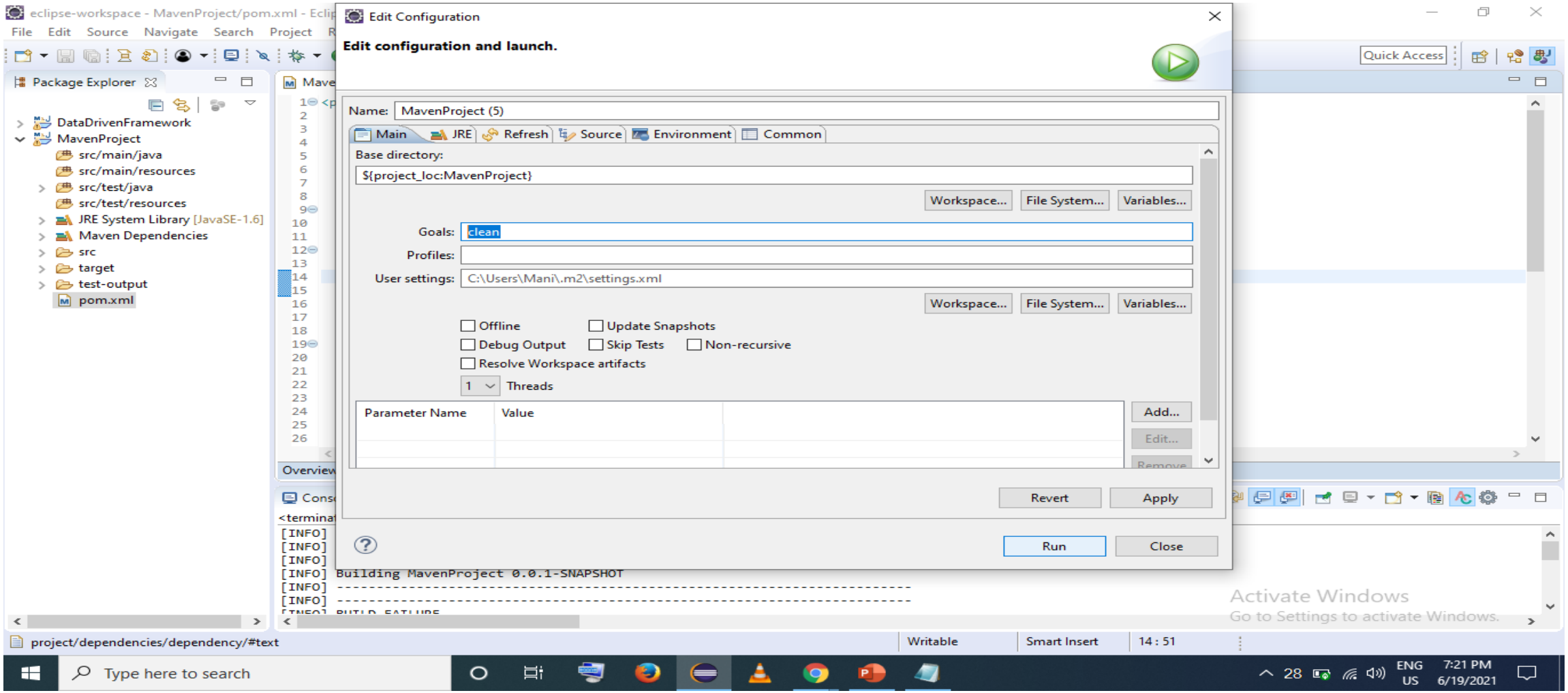
- Clean
- Compile
- Test

And click -----run

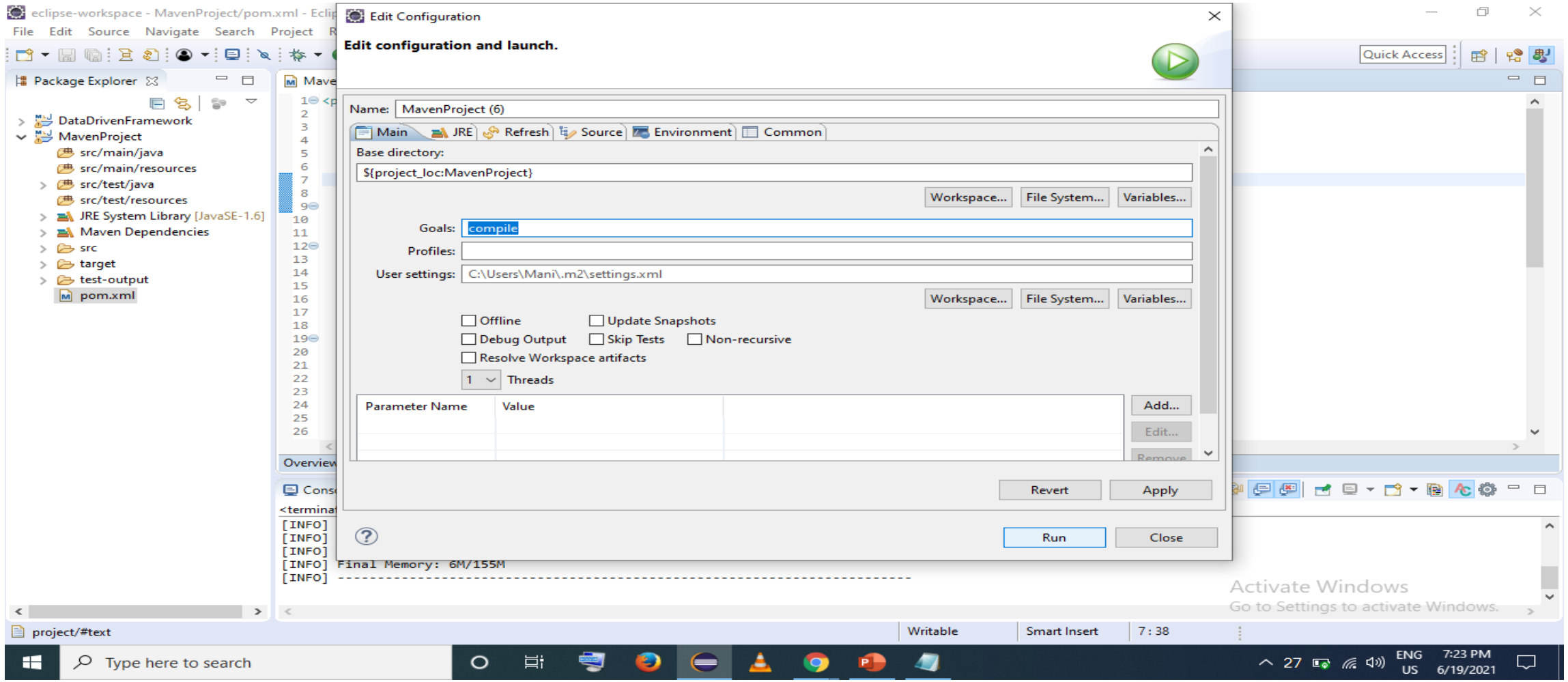
Executing Maven Build



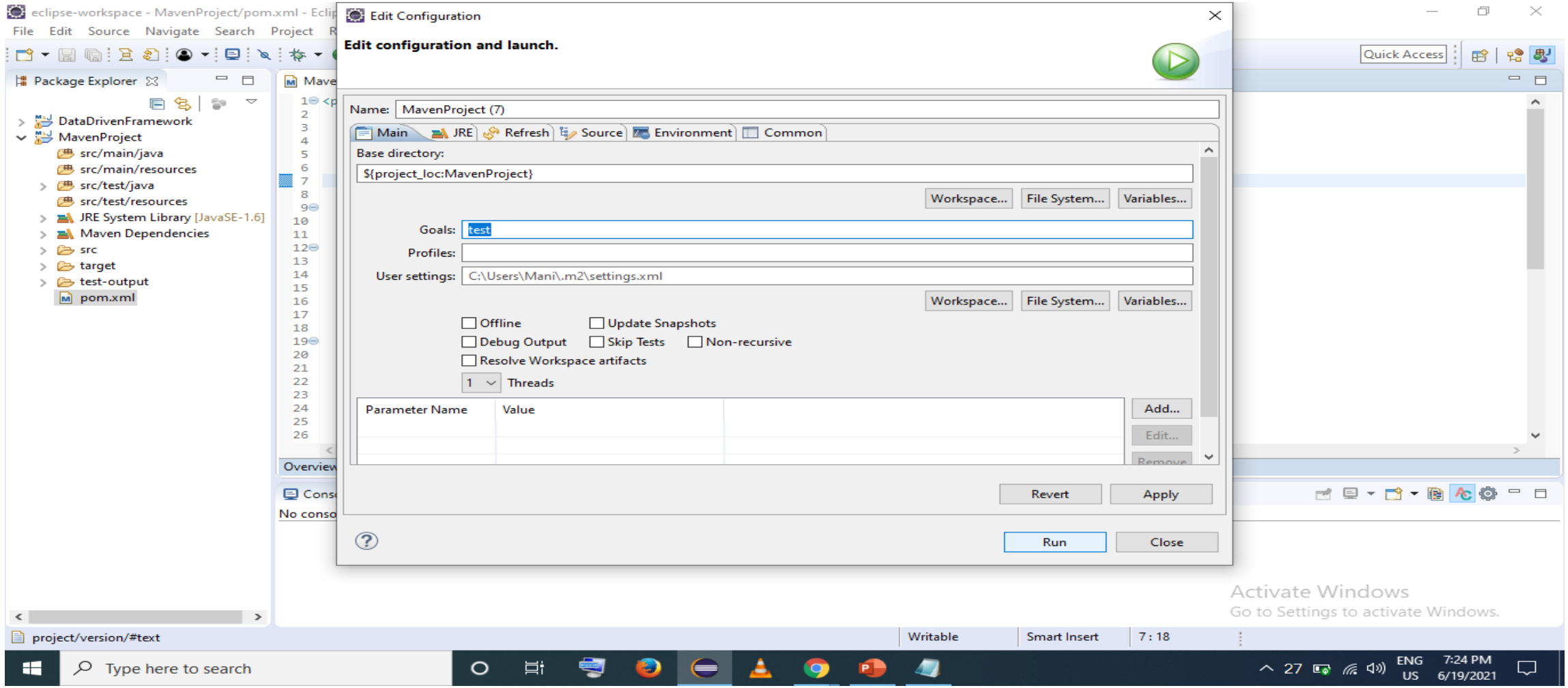
Launch Project



Launch Project

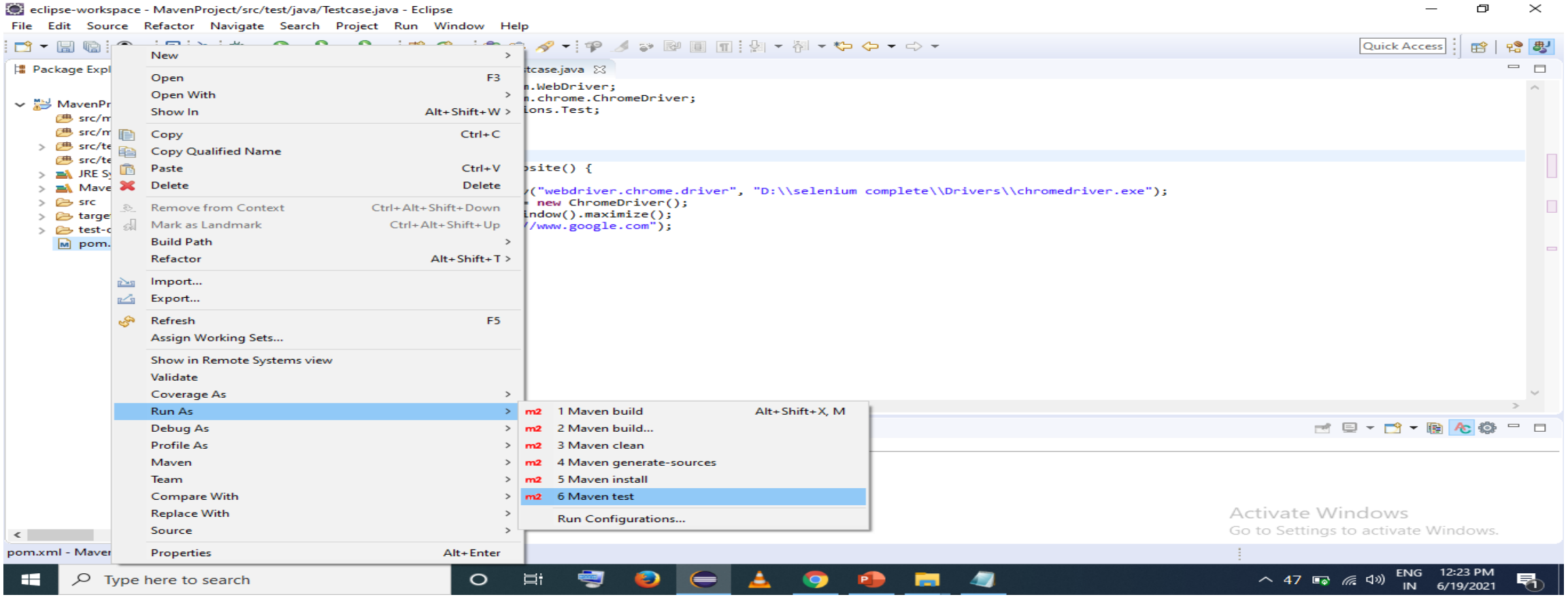


Test


eclipse-workspace - MavenProject/pom.xml - Eclip
File Edit Source Navigate Search Project
Package Explorer
DataDrivenFramework
MavenProject
src/main/java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.6]
Maven Dependencies
src
target
test-output
pom.xml
Edit Configuration
Edit configuration and launch.
Name: MavenProject (7)
Main
JRE
Refresh
Source
Environment
Common
Base directory:
${project_loc:MavenProject}
Workspace...
File System...
Variables...
Goals: test
Profiles:
User settings: C:\Users\Mani\.m2\settings.xml
Workspace...
File System...
Variables...
Offline
Update Snapshots
Debug Output
Skip Tests
Non-recursive
Resolve Workspace artifacts
1
Threads
Parameter Name
Value
Add...
Edit...
Remove
Revert
Apply
Run
Close
Quick Access
No conso
Activate Windows
Go to Settings to activate Windows.
project/version/#text
Writable
Smart Insert
7 : 18
Type here to search
ENG
US
7:24 PM
6/19/2021

Without build command

Directly click clean and test by right clicking on pom.xml without using build



Without build command

eclipse-workspace - MavenProject/src/test/java/ddfProj/Testcase.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- MavenProject
  - src/main/java
  - src/main/resources
  - src/test/java
    - ddfProj
      - Testcase.java
  - src/test/resources
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
  - target
  - pom.xml

Testcase.java

```java
1  package ddfProj;
2
3  import org.openqa.selenium.WebDriver;
4  import org.openqa.selenium.chrome.ChromeDriver;
5  import org.testng.annotations.Test;
6
7  public class Testcase {
8
9      @Test
10     public void website() {
11
12         System.setProperty("webdriver.chrome.driver", "D:\\selenium complete\\Drivers\\chromedriver.exe");
13
14         WebDriver driver = new ChromeDriver();
15
16         driver.manage().window().maximize();
17
18         driver.get("https://www.google.com");
19
20         driver.quit();
21     }
```

Console | Results of running class Testcase

```
<terminated> Testcase [TestNG] C:\Program Files\Java\jre1.8.0_281\bin\javaw.exe (Jun 19, 2021, 10:59:27 AM)
[RemoteTestNG] detected TestNG version 6.14.3
Starting ChromeDriver 91.0.4472.19 (1bf021f248676a0b2ab3ee0561d83a59e424c23e-refs/branch-heads/4472@{#288}) on port 31386
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jun 19, 2021 10:59:34 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
PASSED: website

===============================================
    Default test
    Tests run: 1, Failures: 0, Skips: 0
===============================================
```

Writable | Smart Insert | 16 : 45

Without build command

eclipse-workspace - MavenProject/src/test/java/Testcase.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Console

```
<terminated> C:\Program Files\Java\jre1.8.0_281\bin\javaw.exe (Jun 19, 2021, 12:23:34 PM)
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building MavenProject 0.0.1-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ MavenProject ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ MavenProject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ MavenProject ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ MavenProject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ MavenProject ---
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running Testcase
Starting ChromeDriver 91.0.4472.19 (1bf021f248676a0b2ab3ee0561d83a59e424c23e-refs/branch-heads/4472@{#288}) on port 2318
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jun 19, 2021 12:23:44 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 8.848 s - in Testcase
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
```


Type here to search

ENG IN 12:24 PM 6/19/2021

THANK YOU

SmartCliff
Career Mobility Solutions