

Experiment no 7

Title:

Demonstrate Maven Build Life Cycle

Objective:

To understand and implement the Maven Build Life Cycle and observe how different build phases work in a Java project.

Theory (Simple Words):

Maven is a **build tool** used in Java projects. It helps us **compile**, **test**, **package**, and **install** Java programs easily.

Maven uses a file called **pom.xml** to manage:

- Project info
- Dependencies (external JARs)
- Plugins (for extra features)

Maven has **lifecycles** (like steps) such as:

- clean – remove old files
- compile – compile Java code
- test – run test cases
- package – create JAR file
- install – put JAR in local repo
- deploy – send JAR to remote repo

Materials Needed:

- Windows PC
- Java JDK (8 or above)
- Maven
- Any code editor (like VS Code or IntelliJ)

Experiment Steps

Step 1: Install Java & Maven

```
Command Prompt

C:\Users\sanga>java --version
java 23 2024-09-17
Java(TM) SE Runtime Environment (build 23+37-2369)
Java HotSpot(TM) 64-Bit Server VM (build 23+37-2369, mixed mode, sharing)

C:\Users\sanga>mvn -v
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfcdc97d260186937)
Maven home: C:\Program Files\apache-maven-3.9.9
Java version: 23, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-23
Default locale: en_IN, platform encoding: UTF-8
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"

C:\Users\sanga>mvn -v|
```

Step 2: Create Maven Project (Manually)

Name of folder is : MavenDemo

1.Create a folder:

```
mkdir MavenDemo
```

```
cd MavenDemo
```

2. Create folders manually:

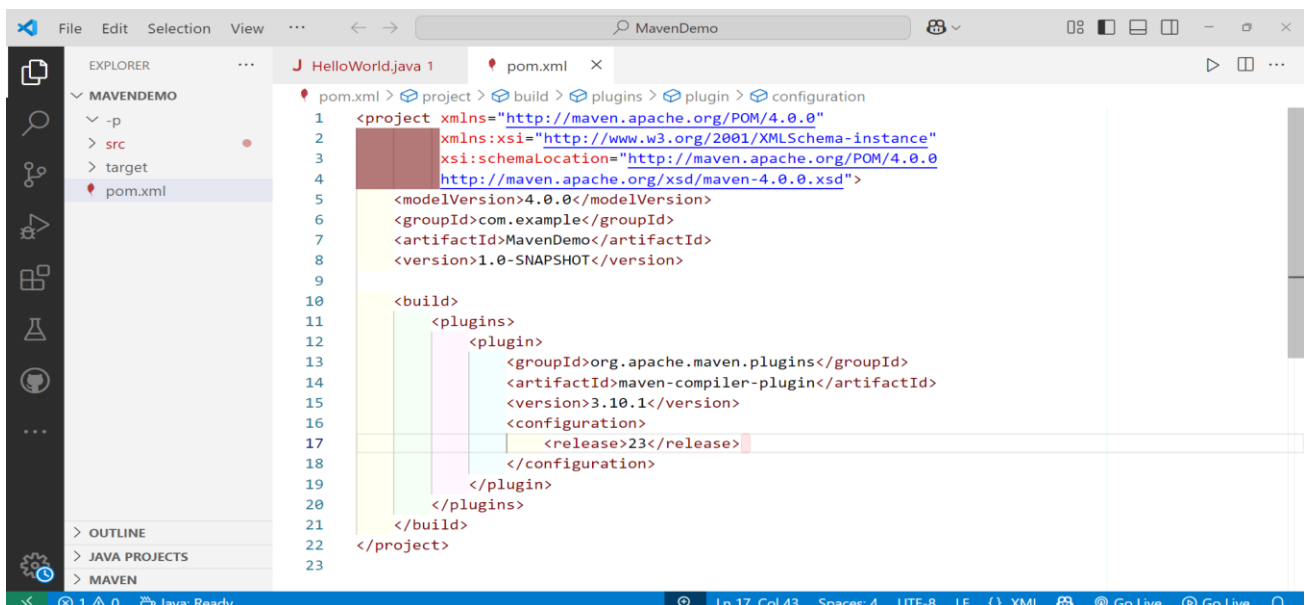
```
mkdir -p src\main\java
```

3. Create Java file inside src\main\java\HelloWorld.java

CopyEdit

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, Maven!");
    }
}
```

4. Create pom.xml in MavenDemo folder



```
File Edit Selection View MavenDemo pom.xml
pom.xml > project > build > plugins > plugin > configuration
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4     http://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6   <groupId>com.example</groupId>
7   <artifactId>MavenDemo</artifactId>
8   <version>1.0-SNAPSHOT</version>
9
10  <build>
11    <plugins>
12      <plugin>
13        <groupId>org.apache.maven.plugins</groupId>
14        <artifactId>maven-compiler-plugin</artifactId>
15        <version>3.10.1</version>
16        <configuration>
17          <release>23</release>
18        </configuration>
19      </plugin>
20    </plugins>
21  </build>
22 </project>
23
```

```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.5189]
(c) Microsoft Corporation. All rights reserved.

E:\Dekstop\Practical_SEM_8TH\Devops_Practical>mkdir MavenDemo

E:\Dekstop\Practical_SEM_8TH\Devops_Practical>cd MavenDemo

E:\Dekstop\Practical_SEM_8TH\Devops_Practical\MavenDemo>mkdir -p src\main\java

E:\Dekstop\Practical_SEM_8TH\Devops_Practical\MavenDemo>mvn clean
[INFO] Scanning for projects...
[INFO] -----< com.example:MavenDemo >-----
[INFO] Building MavenDemo 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO] --- clean:3.2.0:clean (default-clean) @ MavenDemo ---
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.310 s
[INFO] Finished at: 2025-04-15T12:00:46+05:30
```

Step 3: Run Maven Build Phases

1. Clean (Deletes target folder if exists) : mvn clean

```
C:\Windows\System32\cmd.e
E:\Dekstop\Practical_SEM_8TH\Devops_Practical\MavenDemo>mvn clean
[INFO] Scanning for projects...
[INFO] -----< com.example:MavenDemo >-----
[INFO] Building MavenDemo 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO] --- clean:3.2.0:clean (default-clean) @ MavenDemo ---
[INFO] Deleting E:\Dekstop\Practical_SEM_8TH\Devops_Practical\MavenDemo\target
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.293 s
[INFO] Finished at: 2025-04-15T12:11:35+05:30
[INFO] -----
```

2. Compile (Compiles your Java file into .class file) : mvn compile

```
C:\Windows\System32\cmd.e x + v
E:\Dekstop\Practical_SEM_8TH\Devops_Practical\MavenDemo>mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example:MavenDemo >-----
[INFO] Building MavenDemo 1.0-SNAPSHOT
[INFO]    from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ MavenDemo ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i
.e. build is platform dependent!
[INFO] skip non existing resourceDirectory E:\Dekstop\Practical_SEM_8TH\Devops_P
ractical\MavenDemo\src\main\resources
[INFO]
[INFO] --- compiler:3.10.1:compile (default-compile) @ MavenDemo ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. bu
ild is platform dependent!
[INFO] Compiling 1 source file to E:\Dekstop\Practical_SEM_8TH\Devops_Practical\
```

3. Test (Even if there are no test cases, it will run test phase) : mvn test

```
C:\Windows\System32\cmd.e x + v
E:\Dekstop\Practical_SEM_8TH\Devops_Practical\MavenDemo>mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example:MavenDemo >-----
[INFO] Building MavenDemo 1.0-SNAPSHOT
[INFO]    from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ MavenDemo ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
build is platform dependent!
[INFO] skip non existing resourceDirectory E:\Dekstop\Practical_SEM_8TH\Devops_Pract
ical\MavenDemo\src\main\resources
[INFO]
[INFO] --- compiler:3.10.1:compile (default-compile) @ MavenDemo ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ MavenDemo ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
build is platform dependent!
```

4. Package (Packages into .jar file) : mvn package

```
C:\Windows\System32\cmd.e x + v
E:\Dekstop\Practical_SEM_8TH\Devops_Practical\MavenDemo>mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example:MavenDemo >-----
[INFO] Building MavenDemo 1.0-SNAPSHOT
[INFO]    from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ MavenDemo ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
build is platform dependent!
[INFO] skip non existing resourceDirectory E:\Dekstop\Practical_SEM_8TH\Devops_Pract
ical\MavenDemo\src\main\resources
[INFO]
[INFO] --- compiler:3.10.1:compile (default-compile) @ MavenDemo ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ MavenDemo ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
build is platform dependent!
```

5. Install (Installs the .jar in local Maven repository) : mvn install

```
C:\Windows\System32\cmd.e x + v
E:\Dekstop\Practical_SEM_8TH\Devops_Practical\MavenDemo>mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example:MavenDemo >-----
[INFO] Building MavenDemo 1.0-SNAPSHOT
[INFO]    from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ MavenDemo ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
build is platform dependent!
[INFO] skip non existing resourceDirectory E:\Dekstop\Practical_SEM_8TH\Devops_Pract
ical\MavenDemo\src\main\resources
[INFO]
[INFO] --- compiler:3.10.1:compile (default-compile) @ MavenDemo ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ MavenDemo ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
build is platform dependent!
```

```
C:\Windows\System32\cmd.e  X + v
[INFO] --- surefire:3.2.5:test (default-test) @ MavenDemo ---
[INFO] No tests to run.
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ MavenDemo ---
[INFO]
[INFO] --- install:3.1.2:install (default-install) @ MavenDemo ---
[INFO] Installing E:\Dekstop\Practical_SEM_8TH\Devops_Practical\MavenDemo\pom.xml to
C:\Users\sanga\.m2\repository\com\example\MavenDemo\1.0-SNAPSHOT\MavenDemo-1.0-SNAP
SHOT.pom
[INFO] Installing E:\Dekstop\Practical_SEM_8TH\Devops_Practical\MavenDemo\target\Mav
enDemo-1.0-SNAPSHOT.jar to C:\Users\sanga\.m2\repository\com\example\MavenDemo\1.0-S
NAPSHOT\MavenDemo-1.0-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.534 s
[INFO] Finished at: 2025-04-15T12:12:48+05:30
[INFO] -----
```

Step 4: Run the .jar File

cd target

java -cp MavenDemo-1.0-SNAPSHOT.jar HelloWorld

```
E:\Dekstop\Practical_SEM_8TH\Devops_Practical\MavenDemo>cd target

E:\Dekstop\Practical_SEM_8TH\Devops_Practical\MavenDemo\target>java -cp MavenDemo-1.
0-SNAPSHOT.jar HelloWorld
Hello, Maven!

E:\Dekstop\Practical_SEM_8TH\Devops_Practical\MavenDemo\target>
```

Conclusion:

This experiment helped us learn how Maven works, including:

- Creating a Maven project manually
- Understanding pom.xml
- Using build lifecycle commands like clean, compile, package, etc.
- Generating and running a .jar file

Maven simplifies Java project management by automating builds, handling dependencies, and standardizing structure.