**Maven**

**What is Maven?**

Maven is a project management and comprehension tool that provides developers with a complete build lifecycle framework. The development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle.

In the case of multiple development team's environment, Maven can set-up the way to work as per standards in a brief time. As most of the project setups are simple and reusable, Maven makes the life of developers easy while creating reports, checks, build and testing automation setups.

Maven provides developers ways to manage the following −

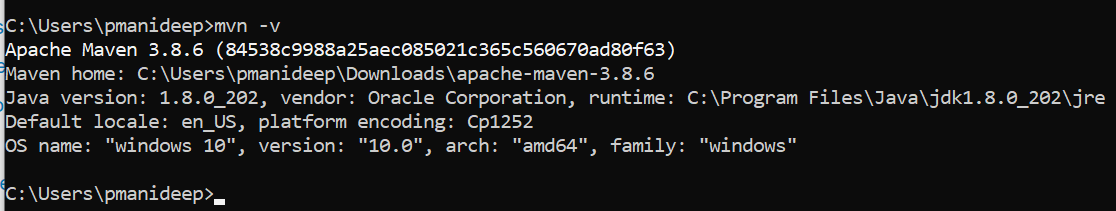
* Builds
* Documentation
* Reporting
* Dependencies
* SCMs
* Releases
* Distribution
* Mailing list

*Maven is a Java tool, so you must have Java installed to proceed.*

First, download Maven. After that, type the following in a terminal or in a command prompt:

Command:

mvn –version

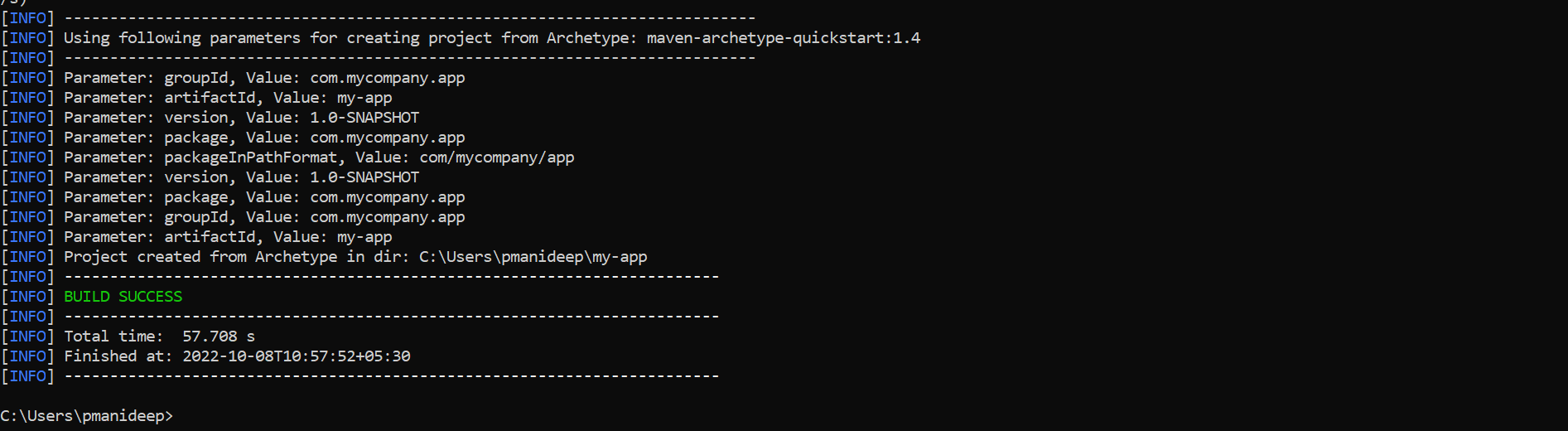
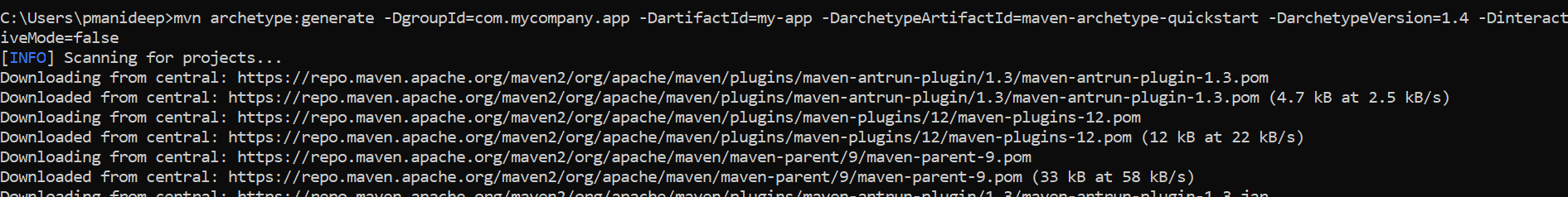
****

**Create a Project**

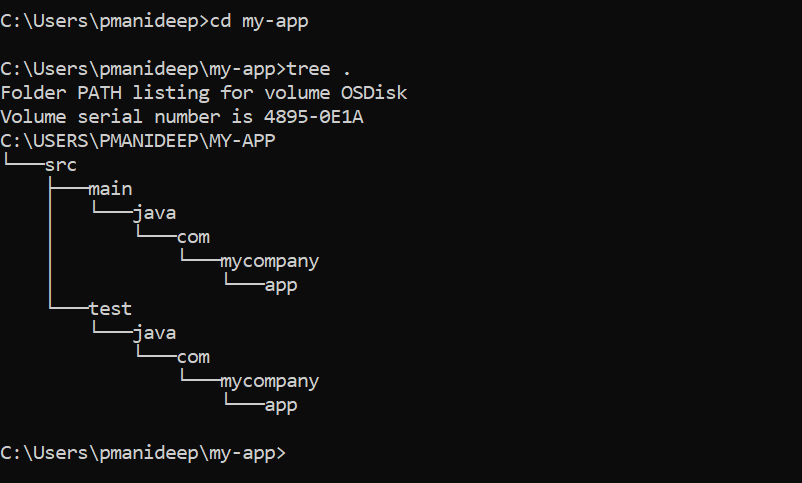
You need somewhere for your project to reside. Create a directory somewhere and start a shell in that directory. On your command line, execute the following Maven goal.

Command:

mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false



Moving into the Directory

****

The src/main/java directory contains the project source code, the src/test/java directory contains the test source, and the pom.xml file is the project's Project Object Model, or POM.

The pom.xml file is the core of a project's configuration in Maven. It is a single configuration file that contains most of the information required to build a project in just the way you want. The POM is huge and can be daunting in its complexity, but it is not necessary to understand all the intricacies just yet to use it effectively. This project's POM is:

<?xml version="1.0" encoding="UTF-8"?>

<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/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.mycompany.app</groupId>

  <artifactId>my-app</artifactId>

  <version>1.0-SNAPSHOT</version>

  <name>my-app</name>

<!-- FIXME change it to the project's website -->

  <url>http://www.example.com</url>

  <properties>

    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

    <maven.compiler.source>1.7</maven.compiler.source>

    <maven.compiler.target>1.7</maven.compiler.target>

  </properties>

  <dependencies>

    <dependency>

      <groupId>junit</groupId>

      <artifactId>junit</artifactId>

      <version>4.11</version>

      <scope>test</scope>

    </dependency>

  </dependencies>

  <build>

    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->

      <plugins>

<!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean\_Lifecycle -->

        <plugin>

          <artifactId>maven-clean-plugin</artifactId>

          <version>3.1.0</version>

        </plugin>

<!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_jar\_packaging -->

        <plugin>

          <artifactId>maven-resources-plugin</artifactId>

          <version>3.0.2</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>

        <plugin>

          <artifactId>maven-install-plugin</artifactId>

          <version>2.5.2</version>

        </plugin>

        <plugin>

          <artifactId>maven-deploy-plugin</artifactId>

          <version>2.8.2</version>

        </plugin>

<!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site\_Lifecycle -->

        <plugin>

          <artifactId>maven-site-plugin</artifactId>

          <version>3.7.1</version>

        </plugin>

        <plugin>

          <artifactId>maven-project-info-reports-plugin</artifactId>

          <version>3.0.0</version>

        </plugin>

      </plugins>

    </pluginManagement>

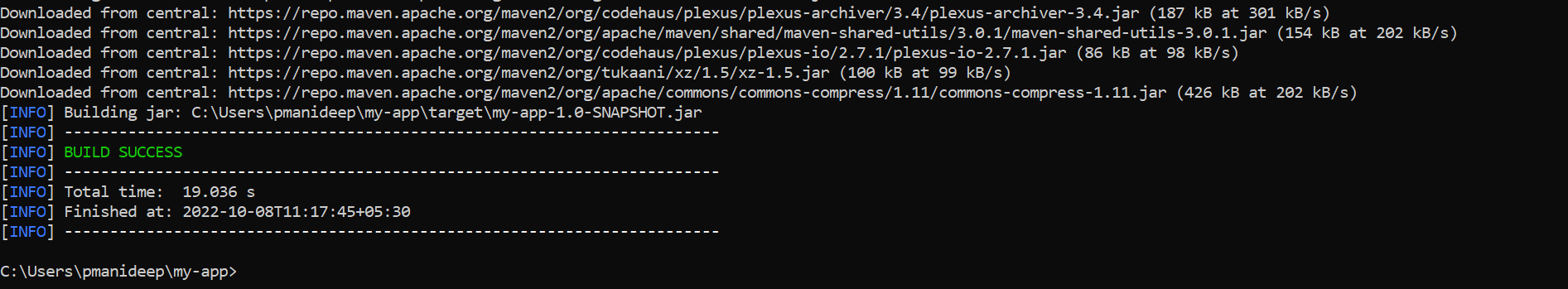
  </build>

</project>

**Build the Package**

Command

mvn package



 Maven executes every phase in the sequence up to and including the one defined. For example, if you execute the *compile* phase, the phases that get executed are:

* **validate**: validate the project is correct and all necessary information is available
* **compile**: compile the source code of the project
* **test**: test the compiled source code using a suitable unit testing framework. These tests should not require the code to be packaged or deployed
* **package**: take the compiled code and package it in its distributable format, such as a JAR.
* **integration-test**: process and deploy the package, if necessary, into an environment where integration tests can be run
* **verify**: run any checks to verify the package is valid and meets quality criteria
* **install**: install the package into the local repository, for use as a dependency in other projects locally
* **deploy**: done in an integration or release environment, copies the final package to the remote repository for sharing with other developers and projects.

There are two other Maven lifecycles of note beyond the *default* list above. They are

* **clean**: cleans up artifacts created by prior buildings
* **site**: generates site documentation for this project

Test the newly compiled and packaged JAR with the following command:

java -cp target/my-app-1.0-SNAPSHOT.jar com.mycompany.app.App

