Download Maven 3.3.3 from <http://maven.apache.org/download.cgi>

Set the environment variables using system properties.

*M2\_HOME=C:\Program Files\Apache Software Foundation\apache-maven-3.3.3*

*M2=%M2\_HOME%\bin*

*MAVEN\_OPTS=-Xms256m -Xmx512m*

Append the string ;%M2% to the end of the system variable, Path. PATH=M2;

Open cmd:

|  |  |
| --- | --- |
|  | c:\> mvn --version |

POM stands for *Project Object Model*. It is fundamental Unit of Work in Maven. It is an XML file. It always resides in the base directory of the project as pom.xml. Some of the configuration that can be specified in the POM are following:

* project dependencies
* plugins
* goals
* build profiles
* project version
* developers
* mailing list

Before creating a POM, we should first decide the project **group** (groupId), its **name**(artifactId) and its version as these attributes help in uniquely identifying the project in repository.

Example POM

<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.companyname.project-group</groupId>

<artifactId>project</artifactId>

<version>1.0</version>

</project>

Maven lifecycle:

|  |  |  |
| --- | --- | --- |
| **Phase** | **Handles** | **Description** |
| prepare-resources | resource copying | Resource copying can be customized in this phase. |
| compile | compilation | Source code compilation is done in this phase. |
| package | packaging | This phase creates the JAR / WAR package as mentioned in packaging in POM.xml. |
| install | installation | This phase installs the package in local / remote maven repository. |

There are always **pre** and **post** phases which can be used to register **goals** which must run prior to or after a particular phase.

|  |  |
| --- | --- |
| **Lifecycle Phase** | **Description** |
| validate | Validates whether project is correct and all necessary information is available to complete the build process. |
| initialize | Initializes build state, for example set properties |
| generate-sources | Generate any source code to be included in compilation phase. |
| process-sources | Process the source code, for example, filter any value. |
| generate-resources | Generate resources to be included in the package. |
| process-resources | Copy and process the resources into the destination directory, ready for packaging phase. |
| compile | Compile the source code of the project. |
| process-classes | Post-process the generated files from compilation, for example to do bytecode enhancement/optimization on Java classes. |
| generate-test-sources | Generate any test source code to be included in compilation phase. |
| process-test-sources | Process the test source code, for example, filter any values. |
| test-compile | Compile the test source code into the test destination directory. |
| process-test-classes | Process the generated files from test code file compilation. |
| test | Run tests using a suitable unit testing framework(Junit is one). |
| prepare-package | Perform any operations necessary to prepare a package before the actual packaging. |
| package | Take the compiled code and package it in its distributable format, such as a JAR, WAR, or EAR file. |
| pre-integration-test | Perform actions required before integration tests are executed. For example, setting up the required environment. |
| integration-test | Process and deploy the package if necessary into an environment where integration tests can be run. |
| post-integration-test | Perform actions required after integration tests have been executed. For example, cleaning up the environment. |
| verify | Run any check-ups to verify the package is valid and meets quality criterias. |
| install | Install the package into the local repository, which can be used as a dependency in other projects locally. |
| deploy | Copies the final package to the remote repository for sharing with other developers and projects. |

Problem creating project: the error was caused by a cached file, I fixed it by deleting all the files under the .m2 repository folder.