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

Maven allows a project to build using its project object model (POM) and a set of plugins that are shared by all projects using Maven, providing a uniform build system.

The pom.xml file is the core of a project's configuration in Maven. It is a single configuration file that contains the majority of information required to build a project in just the way you want.

Build the Project

mvn package - Making a JAR file (the packaging element is set to jar)

mvn deploy

mvn clean install

mvn compile

mvn package - Making a JAR file (the packaging element is set to jar)

mvn install - want to install the artifact you've generated (the JAR file) in your local repository

mvn clean - will remove the target directory with all the build data before starting so that it is fresh.

Configuration File

settings.xml in apache-mavan\conf folder.

Change the local repository.

POM File

<Project>

<parent>

<groupId>net.apmoller.crb.soa.service.mailindexing</groupId>

<artifactId>mailindexing</artifactId>

<version>1.0.5-SNAPSHOT</version>

<packaging>pom</packaging>

</parent>

<repositories>

<repository>

</repository>

</repositories>

<dependencyManagement>

<dependency>

<groupId>log4j</groupId>

<artifactId>apache-log4j-extras</artifactId>

<version>1.1</version>

</dependency>

</dependencyManagement>

<Modules>

<module>different modules of the project</module>

</Modules>

</Project>

Architypes – directory structure for projects

Group id -

Actifact id – project name

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.

The POM contains information about the project and various configuration detail used by Maven to build the project(s).

POM also contains the goals and plugins.

While executing a task or goal, Maven looks for the POM in the current directory. It reads the POM, gets the needed configuration information, then executes the goal. Some of the configuration that can be specified in the POM are following:

project dependencies

plugins

goals

build profiles

project version

developers

mailing list

Super POM

All POMs inherit from a parent (despite explicitly defined or not). This base POM is known as the Super POM, and contains values inherited by default.

Maven use the effective pom (configuration from super pom plus project configuration) to execute relevant goal. It helps developer to specify minimum configuration detail in his/her pom.xml. Although configurations can be overridden easily.

An easy way to look at the default configurations of the super POM is by running the following command: mvn help:effective-pom