Maven 3.x

by
Anand Kulkarni
anand.kulkarni1@zensar.com

What is Maven?

- Maven is not only a build tool but also a project management tool.
- A build tool such as Ant is focused solely on preprocessing, compilation, packaging, testing.
- A project management tool such as Maven provides a superset of features found in a build tool. In addition to providing build capabilities, Maven can also
- run reports, generate a web site, and facilitate communication among members of a working team.

Convention Over Configuration

- "Convention Over Configuration" concept believes that Systems, libraries, and frameworks should assume reasonable defaults. Without requiring unnecessary configuration, systems should "just work".
- Maven has adopted "Convention Over Configuration".

Maven installation

- Download Maven binary zip archive from

  http://maven.apache.org/download.cgi
- Extract the zip to root drive.. For example C:\apache-maven-3.6.3
- Set following environment variables:
- M2_HOME= C:\apache-maven-3.6.3
- PATH=%PATH%;%M2_HOME%/bin
- Open console & make sure “mvn” command runs successfully.

Maven installation directory

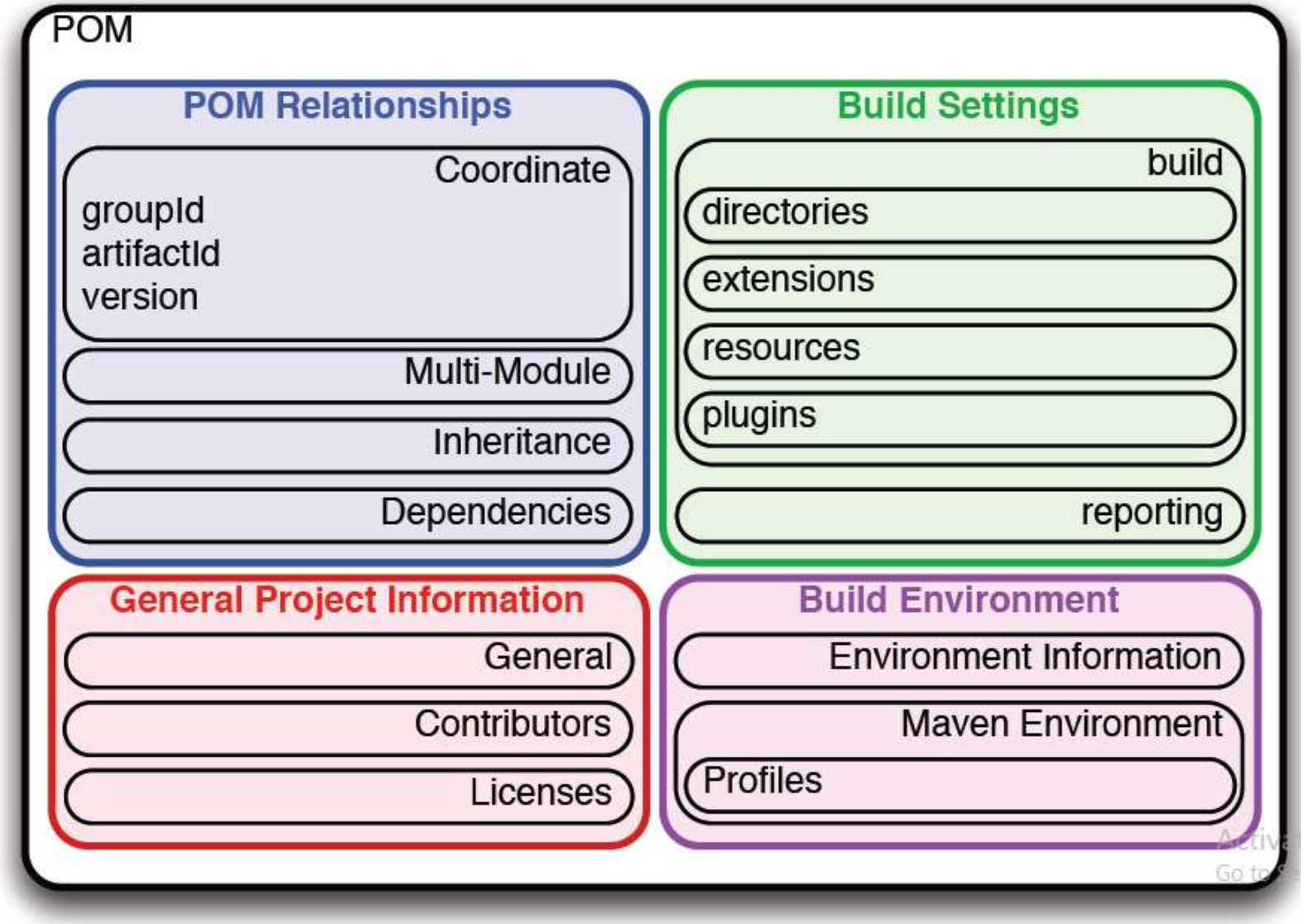
Maven installation directory contains following resources:

- ***LICENSE.txt*** – contains software license of Apache Maven.
- ***NOTICE.txt*** – contains some notices and attributions required by libraries that Maven depends on.
- ***README.txt*** – contains installation instructions.
- ***bin/*** - contains the mvn script that executes Maven.
- ***boot/*** - contains JAR file plexus-classworlds-2.6.0.jar that is responsible for creating a class loader in which Maven executes.
- ***conf/*** - contains global settings.xml that can be used to customize the behavior of maven installation.
- ***lib/*** - contains maven related jar files required to execute mvn operation.

POM (Project Object Model)

- POM describes project's identity, structure and build configuration details.
- Maven projects, dependencies, builds, artifacts: all of these are objects that are described by an XML file called pom.xml.
- The pom.xml is analogous to Ant build.xml.
- The POM tells Maven what sort of project it is dealing with and how to modify default behavior to generate output from source.

POM (Project Object Model)



POM categories

The POM contains 4 categories of description and configuration:

- **General project information**

  This includes a project's name, the URL for a project, the sponsoring organization, and a list of developers and contributors along with the license for a project.
- **Build settings**

  In this section, we customize the behavior of the default Maven build. We can change the location of source and tests, we can add new plugins, we can attach plugin goals to the lifecycle, and we can customize the site generation parameters.

POM categories

- **Build environment**

  The build environment consists of profiles that can be activated for use in different environments. For example, during development you may want to deploy to a development server, whereas in production you want to deploy to a production server. The build environment customizes the build settings for specific environments and is often supplemented by a custom settings.xml in ~/.m2.

- **POM relationships**

  A project rarely stands alone; it depends on other projects, inherits POM settings from parent projects, defines its own coordinates, and may include submodules.

Super POM

- Super POM defines some standard configuration variables that are inherited by all projects. It is found inside lib/maven-model-builder-xx.jar.
- The default Super POM defines a single remote Maven repository with an ID of central. This is the central Maven repository that all Maven clients are configured to read from by default.

```
<repositories>
  <repository>
    <id>central</id>
    <name>Central Repository</name>
    <url>https://repo.maven.apache.org/maven2</url>
    <layout>default</layout>
    <snapshots>
      <enabled>false</enabled>
    </snapshots>
  </repository>
 </repositories>
```

Effective POM

- Effective POM is a combination of your project's pom & super pom.
- If you want to see a project's effective POM, you'll need to run the
- effective-pom goal in the Maven Help plugin.
- You can also generate effective pom using following command on console:

  >mvn help:effective-pom

Thank you!!