



D Gangireddy



## - Agenda

- ✓ Introduction
- ✓ Installation
- ✓ Content – pom.xml
- ✓ Examples



## Apache - Introduction

<b>Type</b>	Java Based Build Tool
<b>Vendor</b>	Apache
<b>Is Open Source?</b>	Yes
<b>Version</b>	3.5.2
<b>Operating system</b>	Cross Platform
<b>Software Download URL</b>	<a href="https://maven.apache.org/download.cgi">https://maven.apache.org/download.cgi</a>
<b>Is executable software?</b>	No, download as zip, extract and use it.
<b>Reference Websites</b>	1) <a href="https://www.petrikainulainen.net/programming/gradle/getting-started-with-gradle-dependency-management/">https://www.petrikainulainen.net/programming/gradle/getting-started-with-gradle-dependency-management/</a>



# Apache - Introduction

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation.

The most powerful feature is able to download the project dependency libraries automatically from maven central repo, maven remote repo or local repo.



# - Contents in pom.xml

```
<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.bt</groupId>
  <artifactId>maven-java-project</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>maven-java-project</name>
  <url>http://maven.apache.org</url>

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

  <dependencies>

    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>

  </dependencies>

</project>
```



## - Contents in pom.xml

All POM files require the **project** element and three mandatory fields: **groupId**, **artifactId**, **version**.  
Root element of POM.xml is **project** and it has three major sub-nodes .

**groupId** : This is an Id of project's group. This is generally unique amongst an organization or a project. For example, a banking group com.company.bank has all bank related projects.

**artifactId** : This is an Id of the project. This is generally name of the project. For example, consumer-banking. Along with the groupId, the artifactId defines the artifact's location within the repository.

**version**: This is the version of the project. Along with the groupId, It is used within an artifact's repository to separate versions from each other. For example: com.company.bank:consumer-banking:1.0 com.company.bank:consumer-banking:1.1.



## - Maven Repositories

A Maven repository can be one of the following types:

- Local Repository
- Central Repository
- Remote Repository



## Apache - Maven Repositories

### Local Repository

The maven local repository is a local folder that is used to store all your project's dependencies (plugin jars and other files which are downloaded by Maven). In simple, when you build a Maven project, all dependency files will be stored in your Maven local repository.

By default, Maven local repository is default to .m2 folder:

- 1) Unix/Mac OS X – ~/.m2 (/Users/BhaskarReddy/.m2/repository)
- 2) Windows – C:\Users\{your-username}\.m2





## Apache - Maven Repositories

### Update Maven Local Repository

Normally, We will change the default local repository folder from default .m2 to another more meaningful name as follows.

Find {M2\_HOME}\conf\settings.xml, update localRepository to something else.

```
<!-- localRepository
| The path to the local repository maven will use to store artifacts.
|
| Default: ${user.home}/.m2/repository
<localRepository>/path/to/local/repo</localRepository>
-->
```

```
<localRepository>D:/maven_repo</localRepository>
```



## Apache - Maven Repositories

### Maven Central Repository

The central repository is the repository provided by the Maven community. It contains a large repository of commonly used libraries. This repository comes into play when Maven does not find libraries in the local repository.

The central repository can be found at: <http://search.maven.org/#browse>.



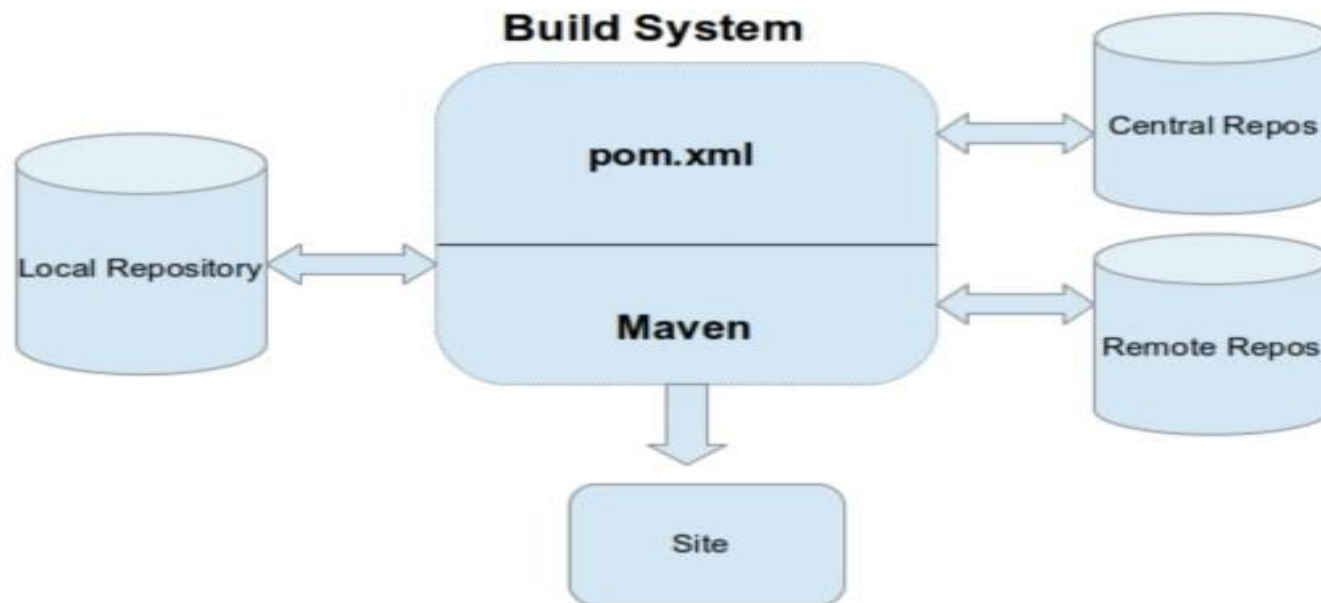
## - Maven Repositories

### Remote Repository

Enterprises usually maintain their own repositories for the libraries that are being used for the project. These differ from the local repository; a repository is maintained on a separate server, different from the developer's machine and is accessible within the organization.



## - Maven Architecture





## - Maven Goals

To invoke a maven build you set a life cycle goal.

`mvn install` : Invokes `generate*`, `compile`, `test`, `package`, `integration-test`, `install`.

`mvn clean`: Invokes just `clean`.

`mvn clean compile` : Clean old builds and execute `generate*`, `compile`.

`mvn compile install` : Invokes `generate*`, `compile`, `test`, `integration-test`, `package`, `install`.

`mvn test clean` : Invokes `generate*`, `compile`, `test` then `clean`.



## - Maven Environment Setup

M2\_HOME

JAVA\_HOME

```
mvn archetype:generate
```

Questions ?

Thank you