MAVEN

www.hexaware.com | © Hexaware Technologies. All rights reserved.

Course Objective

To understand and execute Maven build tool

- Maven Overview
- POM

Reference Link

- Web Link

  https://maven.apache.org

 | © Hexaware Technologies. All rights reserved.

MAVEN Overview

Maven - Introduction

- Apache Maven is a software project management and comprehension tool.
- It is based on the concept of a project object model (POM)
- Maven can manage a project's build, reporting and documentation from a central piece of information.
- In Yiddish, the word *maven* means "accumulator of knowledge"

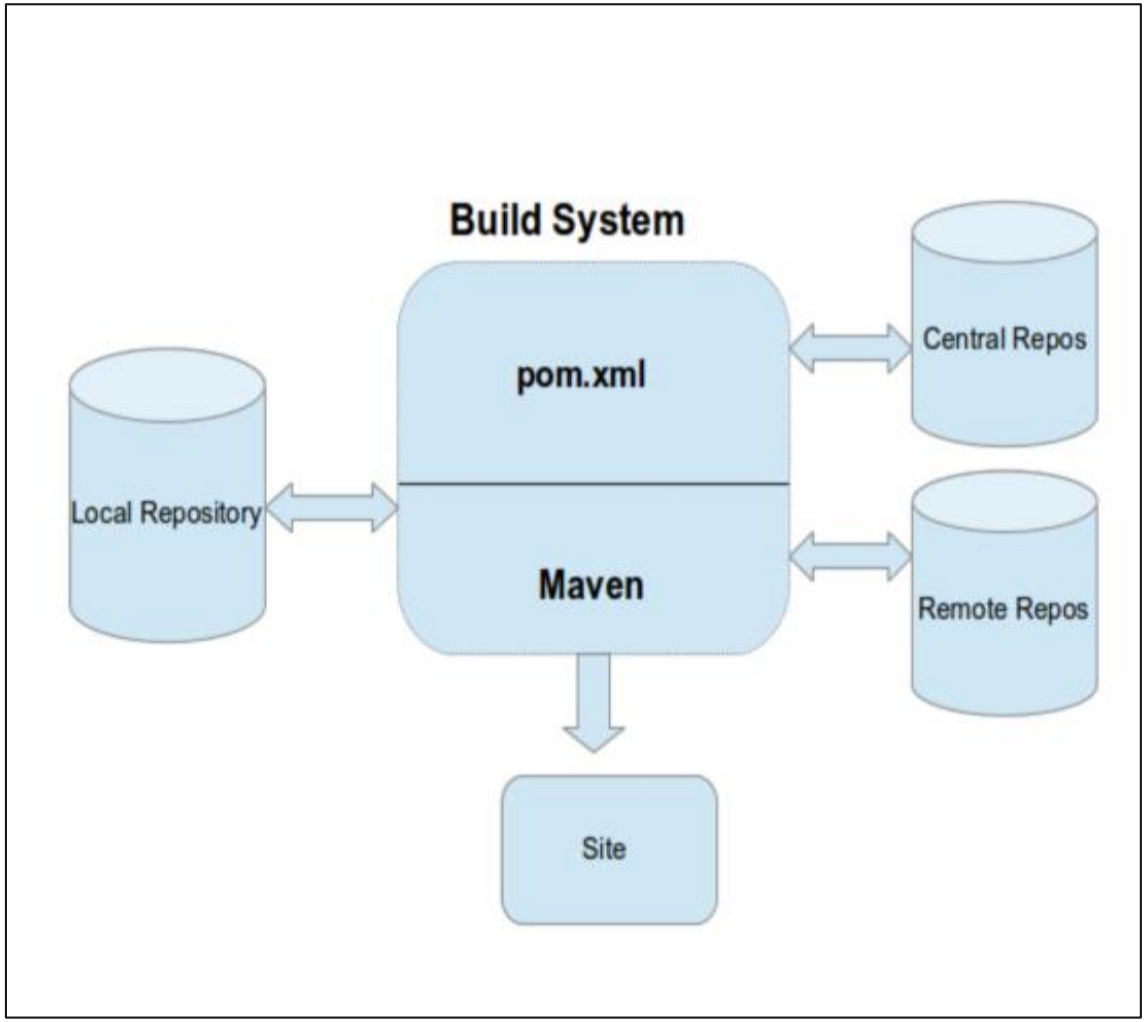
 | © Hexaware Technologies. All rights reserved.

Maven Objective

- Making the build process easy
- Providing a uniform build system
- Providing quality project information
- Providing guidelines for best practices development
- Allowing transparent migration to new features

Maven - Architecture

- Every Maven project contains a POM file that defines the project essentials.
- Maven uses the POM details to decide upon different actions and artifact generation.
- The dependencies specified are first searched for in the local repository and then in the central repository and later in Remote repository.



 | © Hexaware Technologies. All rights reserved.

POM

- A Project Object Model (POM) provides all the configuration for a single project.
- General configuration covers the project's name, its owner and its dependencies on other projects.
- Individual phases of the build process can also be configured, they are implemented as plugins.

  Example: Configure the compiler-plugin to use Java version 1.8 for compilation

 | © Hexaware Technologies. All rights reserved.

Maven - Configuration

Configuration that can be specified in the POM

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

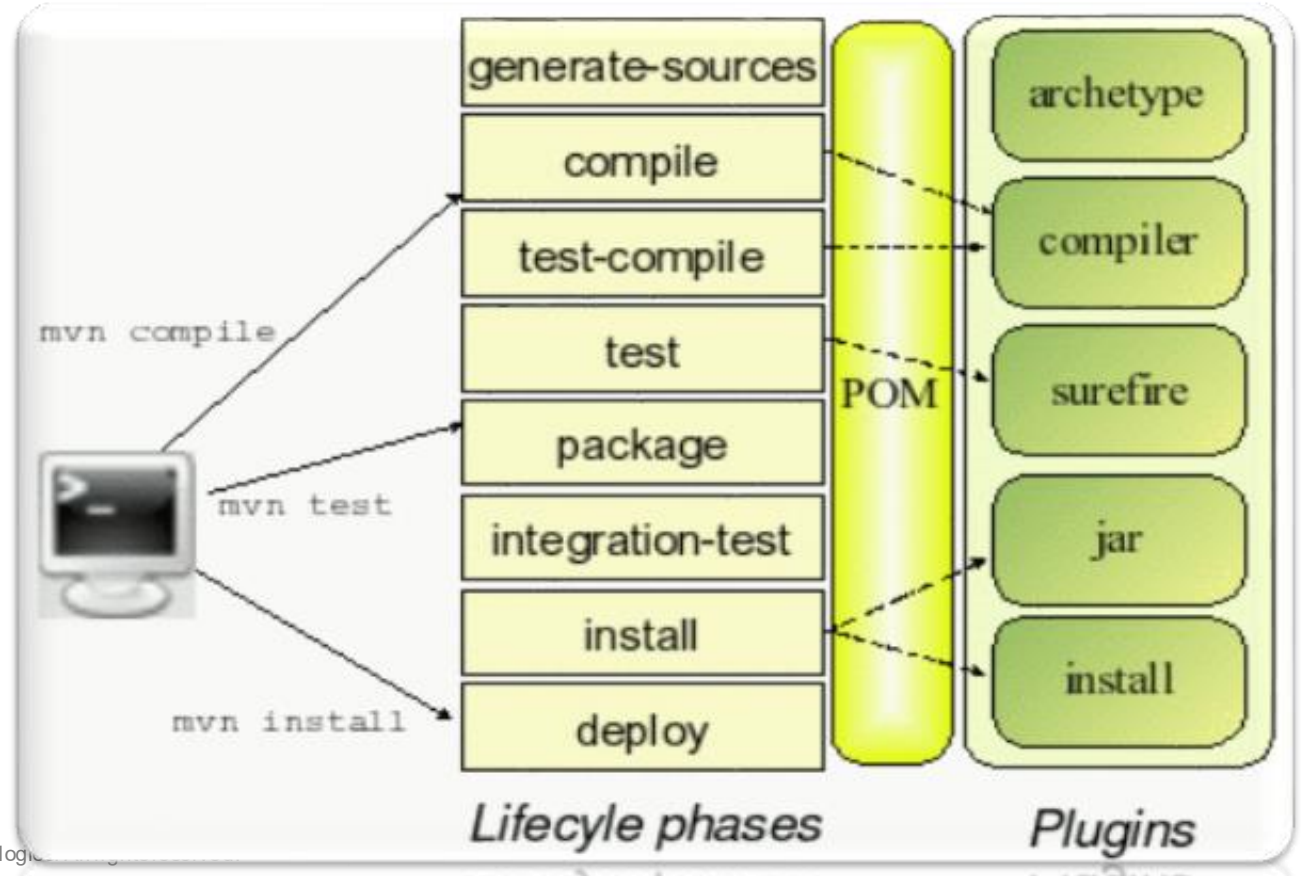
 | © Hexaware Technologies. All rights reserved.

Maven – Configuration Cont…

```
<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>
```

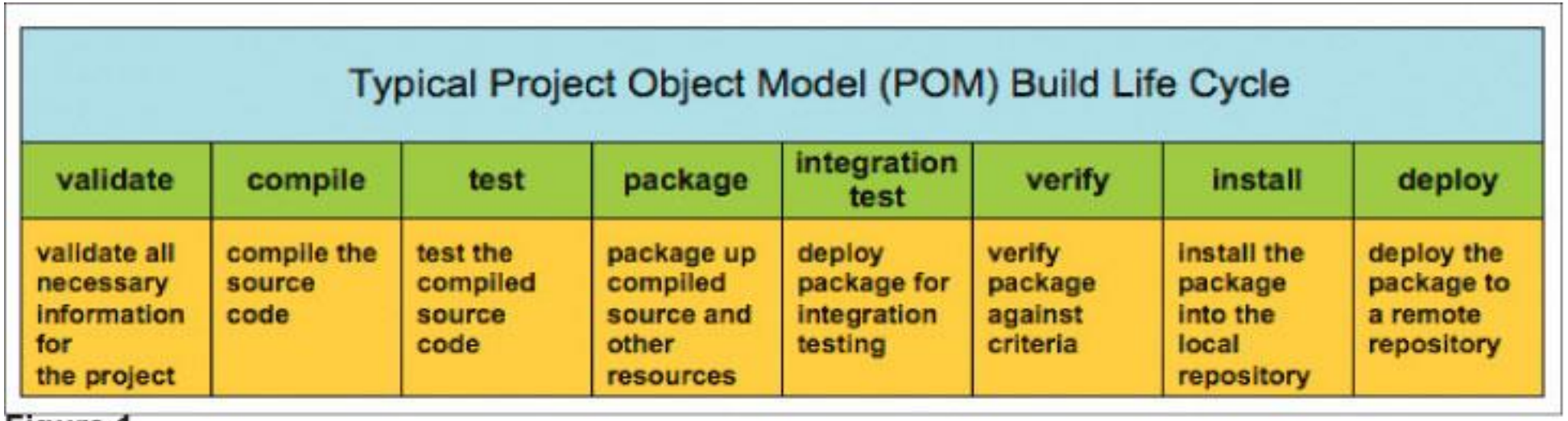
Maven - Build lifecycle Cont…

Build lifecycle is a list of named *phases* that can be used to give order to goal execution.



Maven - Build lifecycle

- Maven has the following three standard lifecycles −
- clean
- default(or build)
- site

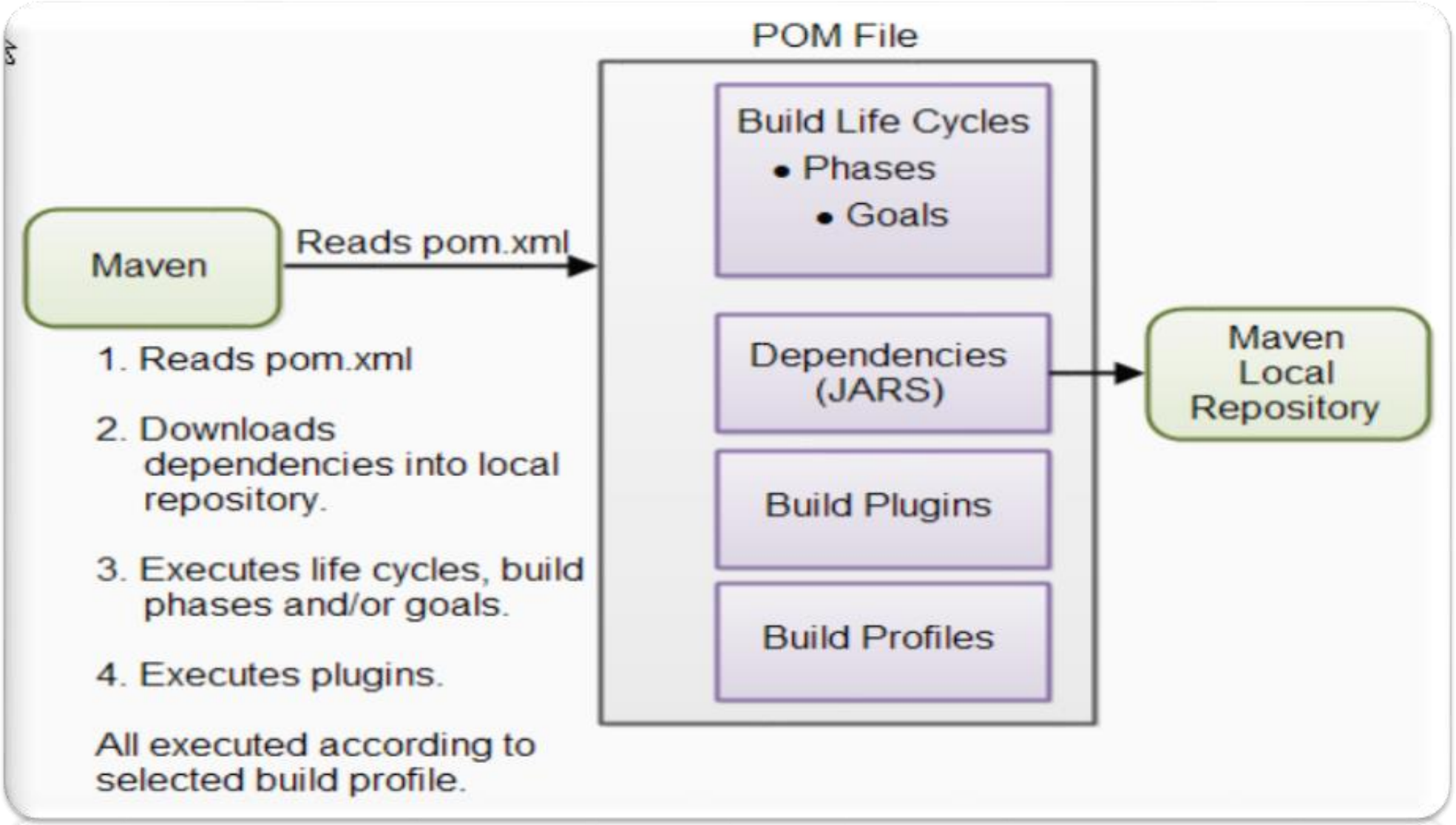


Typical Project Object Model (POM) Build Life Cycle							
validate	compile	test	package	integration test	verify	install	deploy
validate all necessary information for the project	compile the source code	test the compiled source code	package up compiled source and other resources	deploy package for integration testing	verify package against criteria	install the package into the local repository	deploy the package to a remote repository

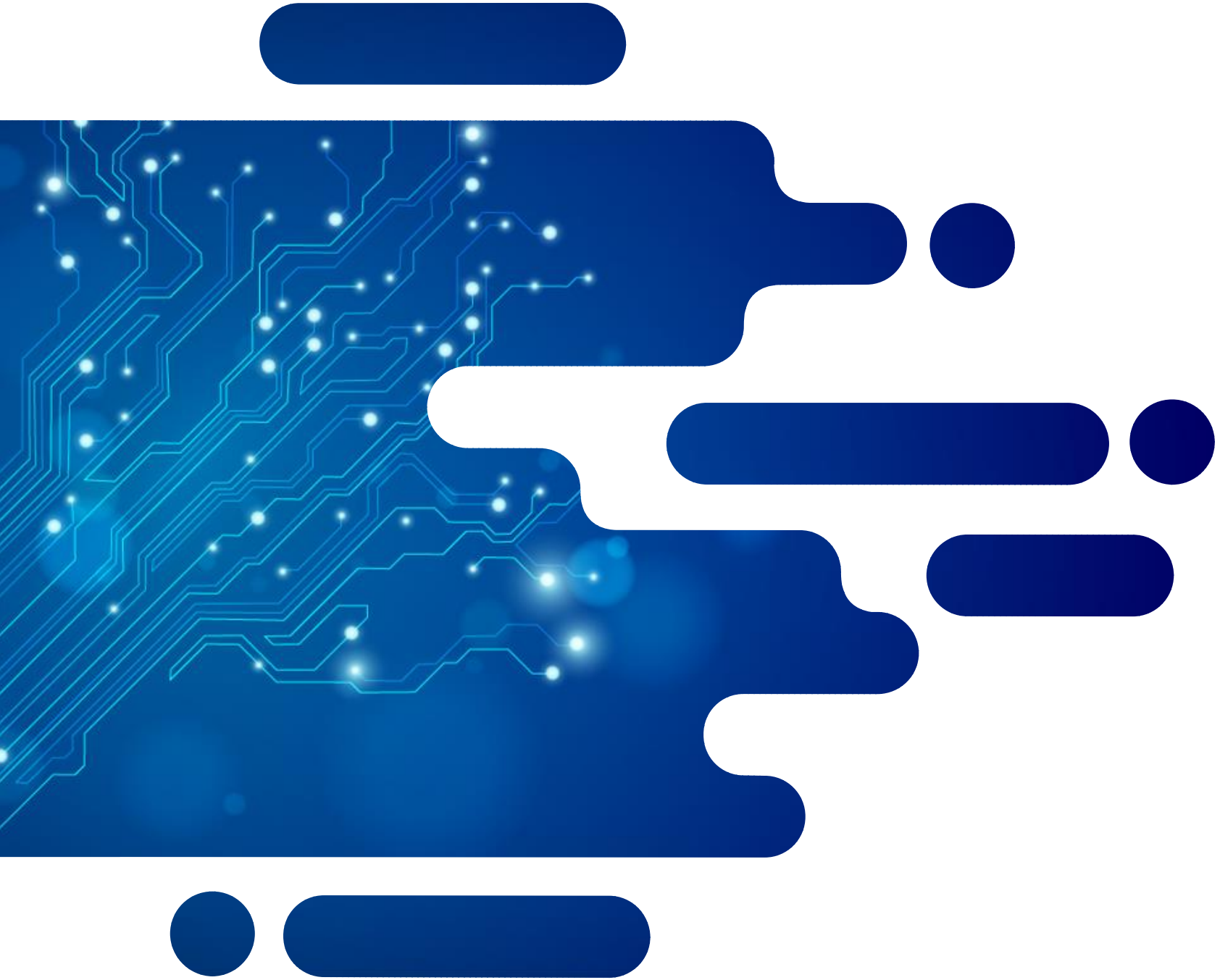
Figure 1

 | © Hexaware Technologies. All rights reserved.

Maven - Overview



 | © Hexaware Technologies. All rights reserved.



POM

POM - Basic

Cont…

```
<!-- The Basics -->
<groupId>...</groupId>
<artifactId>...</artifactId>
<version>...</version>
<packaging>...</packaging>
<dependencies>...</dependencies>
<parent>...</parent>
<dependencyManagement>...
</dependencyManagement>
<modules>...</modules>
<properties>...</properties>
```

This is generally unique amongst an organization or a project.

The artifactId is generally the name of the project

Along with the groupId, It is used within an artifact's repository to separate versions from each other.

www.hexaware.com | © Hexaware Technologies. All rights reserved.

POM – Dependeny Management Cont…

```
<dependencyManagement>
<dependencies>
<dependency>
<groupId>org.glassfish.jersey</groupId>
<artifactId>jersey-bom</artifactId>
<version>${jersey.version}</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>
```

- ✓ The dependency management section is a mechanism for centralizing dependency information.
- ✓ When a set of projects inherits a common parent it's possible to put all information about the dependency in the common POM and have simpler references to the artifacts in the child POMs.

 | © Hexaware Technologies. All rights reserved.

POM – Dependency Management

```
<dependency>
<groupId>org.glassfish.jersey</groupId>
<artifactId>jersey-bom</artifactId>
<version>${jersey.version}</version>
<type>pom</type>
<scope>import</scope>
</dependency>
```

`${jersey.version}` is a placeholder, configure its actual value in <properties> block

```
<properties>
<project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>
<jersey.version>2.25</jersey.version>
<build.number>SNAPSHOT</build.number>
</properties>
```

 | © Hexaware Technologies. All rights reserved.

POM - Dependencies

```
<dependencies>
    <dependency>
        <groupId>org.glassfish.jersey.containers</groupId>
        <artifactId>jersey-container-servlet-core</artifactId>
    </dependency>
    <dependency>
        <groupId>org.glassfish.jersey.media</groupId>
        <artifactId>jersey-media-json-jackson</artifactId>
    </dependency>
</dependencies>
```

 | © Hexaware Technologies. All rights reserved.

Maven Plugins Cont…

- Maven is a plugin execution framework
- **Build plugins** will be executed during the build and it should be configured in the <build/> element from the POM.
- **Reporting plugins** will be executed during the site generation and it should be configured in the <reporting/> element from the POM.

Note: All plugins should have minimal required informations: groupId, artifactId,version.

 | © Hexaware Technologies. All rights reserved.

Maven Plugins

Cont…

Maven Plugins are generally used to −

- create jar file
- create war file
- compile code files
- unit testing of code
- create project documentation
- create project reports

 | © Hexaware Technologies. All rights reserved.

Maven - Plugins

```
<build>
    <pluginManagement>
      <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.5.1</version>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-checkstyle-plugin</artifactId>
            <version>2.17</version>
        </plugin>
        </plugins>
    </pluginManagement>
</build>
```

 | © Hexaware Technologies. All rights reserved.

Sample Application

Sample Application 1 Cont…

Command to set path for Maven

1. Set java path in JAVA_HOME
2. Set java path in MAVEN_HOME
3. Set java path in M2_HOME
4. Set the class path

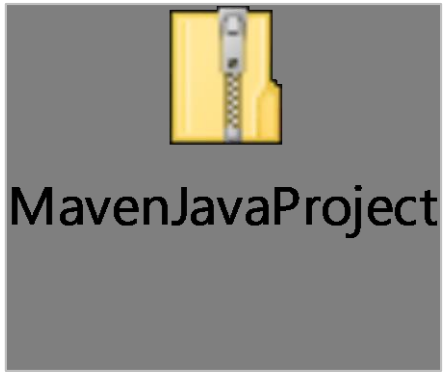
 | © Hexaware Technologies. All rights reserved.

Sample Application 1 Cont…

Steps to create the application:

1. Create a Java project
2. Convert the Java project into Maven project

   Right click the project -> Configure -> Convert to Maven Project
3. Finish
4. Create a Java class with a simple print statement



 | © Hexaware Technologies. All rights reserved.

Sample Application 1

Steps to Execute the application:

1. $ mvn install
2. $ mvn clean
3. $ mvn compile
4. $ mvn package
5. $ mvn exec:java -Dexec.mainClass="com.hexa.Sample.Welcome"

 | © Hexaware Technologies. All rights reserved.

Standard Project Structure Cont…

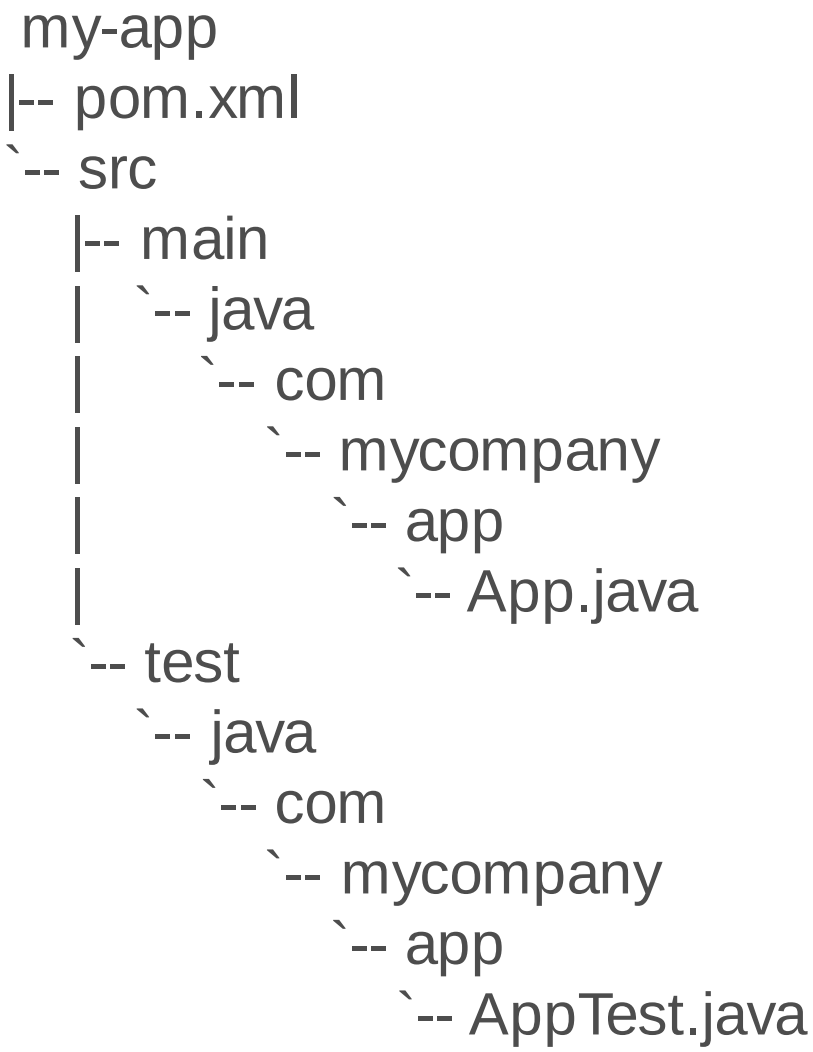
- To create the standard project structure:

```
mvn archetype:generate
```

- This creates src/main/java directory which 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.

 | © Hexaware Technologies. All rights reserved.

Standard Project Structure



 | © Hexaware Technologies. All rights reserved.

Sample Application 2 Cont…

Steps to create application:

STEP 1:

-mvn archetype:generate

STEP 2:

-Search for “Sample Maven Project”

- Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains): 1055: 1055
- Choose org.apache.maven.archetypes:maven-archetype-quickstart version:6
- Define value for property 'groupId': com.Hexa

 | © Hexaware Technologies. All rights reserved.

Sample Application 2 Cont…

- Define value for property 'artifactId': Maven-Sample
- Define value for property 'version' 1.0-SNAPSHOT: : 1.0-SNAPSHOT
- Define value for property 'package' com.Hexa: : com.Hexa.Testing
- Confirm properties configuration:
- groupId: com.Hexa
- artifactId: Maven-Sample
- version: 1.0-SNAPSHOT
- package: com.Hexa.Testing
- Y: : Y

 | © Hexaware Technologies. All rights reserved.

Sample Application 2 Cont…

STEP 3:

Open the project from the workspace

STEP 4:

Create a test method in the AppTest.java

STEP 5:

Add dependencies in the POM.XML

STEP 6:

Execute the application using Maven commands.

 | © Hexaware Technologies. All rights reserved.

Sample Application 2 Cont…

Sample Maven application with Junit testing

 | © Hexaware Technologies. All rights reserved.

Innovative Services

Passionate Employees

Delighted Customers

© Copyright 2012 by Hexaware Technologies Limited

All rights reserved. All text included in this document is the exclusive property of Hexaware Technologies Ltd, and may not be copied, reproduced, or used in any way without the express permission in writing of Hexaware Technologies Ltd.