INSIGHT TECH CONSULTING GROUP

PROVIDING IT SOLUTIONS…


Apache
maven

WHAT IS MAVEN?

- Maven is a project management and comprehension tool that provides developers a complete build lifecycle framework. Development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle.

- Maven is a project management and comprehension tool that provides developers a complete build lifecycle framework. Development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle.

FEATURES OF MAVEN

As most of the project setups are simple and reusable, Maven makes life of developer easy while creating reports, checks, build and testing automation setups.

Maven provides developers ways to manage the followings:

- Builds
- Documentation
- Reporting
- Dependencies
- Releases
- Distribution

FEATURES OF MAVEN

- Simple project setup that follows best practices.
- Consistent usage across all projects.
- Dependency management including automatic updating.
- A large and growing repository of libraries.
- Extensible, with the ability to easily write plugins in Java or scripting languages.
- Instant access to new features with little or no extra configuration.

APACHE MAVEN

- To summarize, Maven simplifies and standardizes the project build process.
- It handles compilation, distribution, documentation, team collaboration and other tasks seamlessly.
- Maven increases reusability and takes care of most of the build related tasks.

MAVEN EVOLUTION

- Maven was originally designed to simplify building processes in Jakarta Turbine project. There were several projects and each project contained slightly different ANT build files. JARs were checked into CVS.
- Apache group then developed **Maven** which can build multiple projects together, publish projects information, deploy projects, share JARs across several projects and help in collaboration of teams.

OBJECTIVE

- The primary goal of Maven is to provide developer with the following:
  - A comprehensive model for projects, which is reusable, maintainable, and easier to comprehend.
  - Plugins or tools that interact with this declarative model.
- Maven project structure and contents are declared in an xml file, pom.xml, referred as Project Object Model (POM), which is the fundamental unit of the entire Maven system.

POM

POM stands for Project Object Model. It is fundamental unit of work in Maven. It is an XML file that 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).

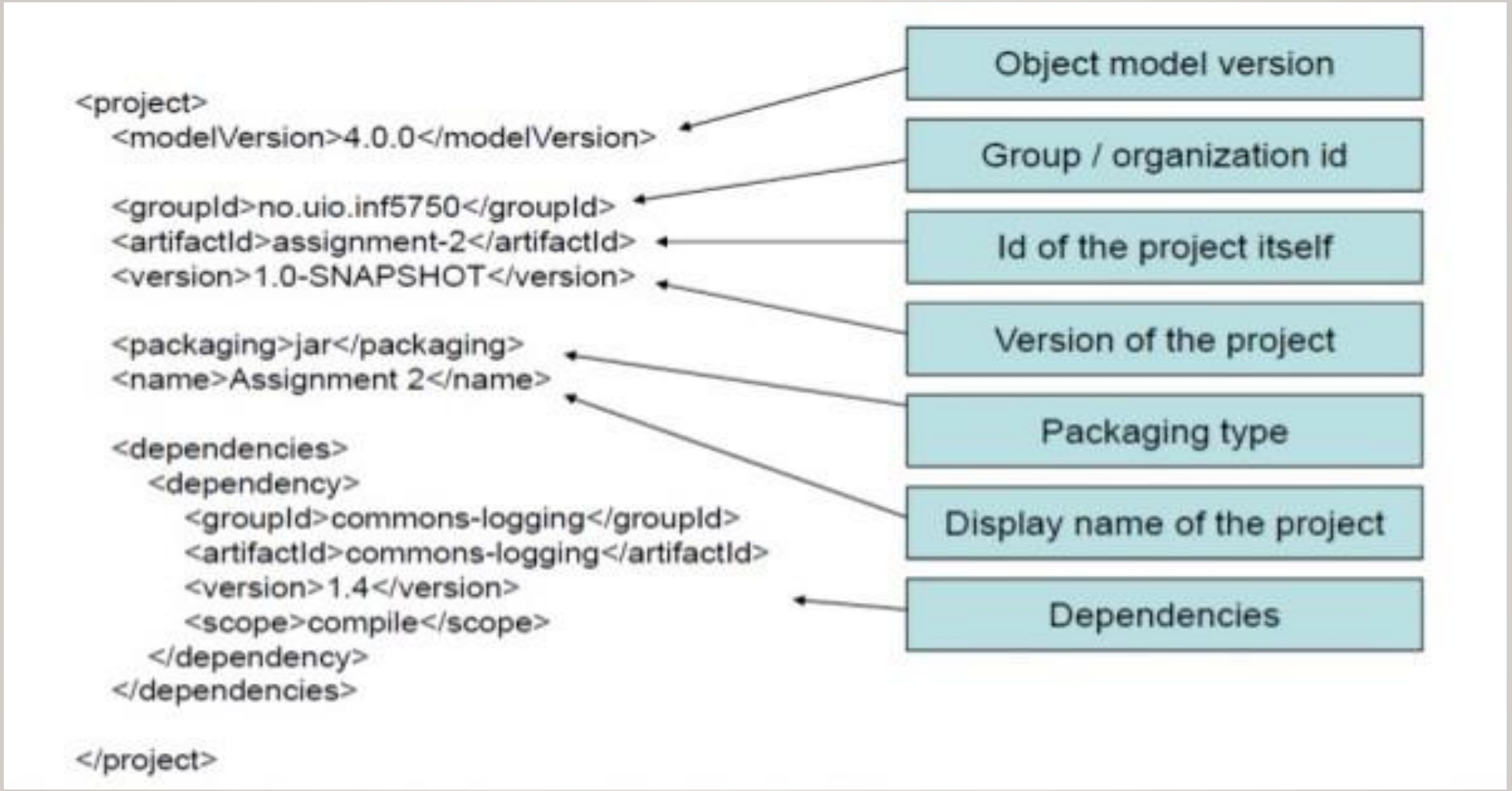
POM also contains the goals and plugins. While executing a task or goal, Maven looks for the POM in the current directory. It reads the POM, gets the needed configuration information, and then executes the goal. Some of the configuration that can be specified in the POM are following:

POM – CONTD..

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

Note:

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.

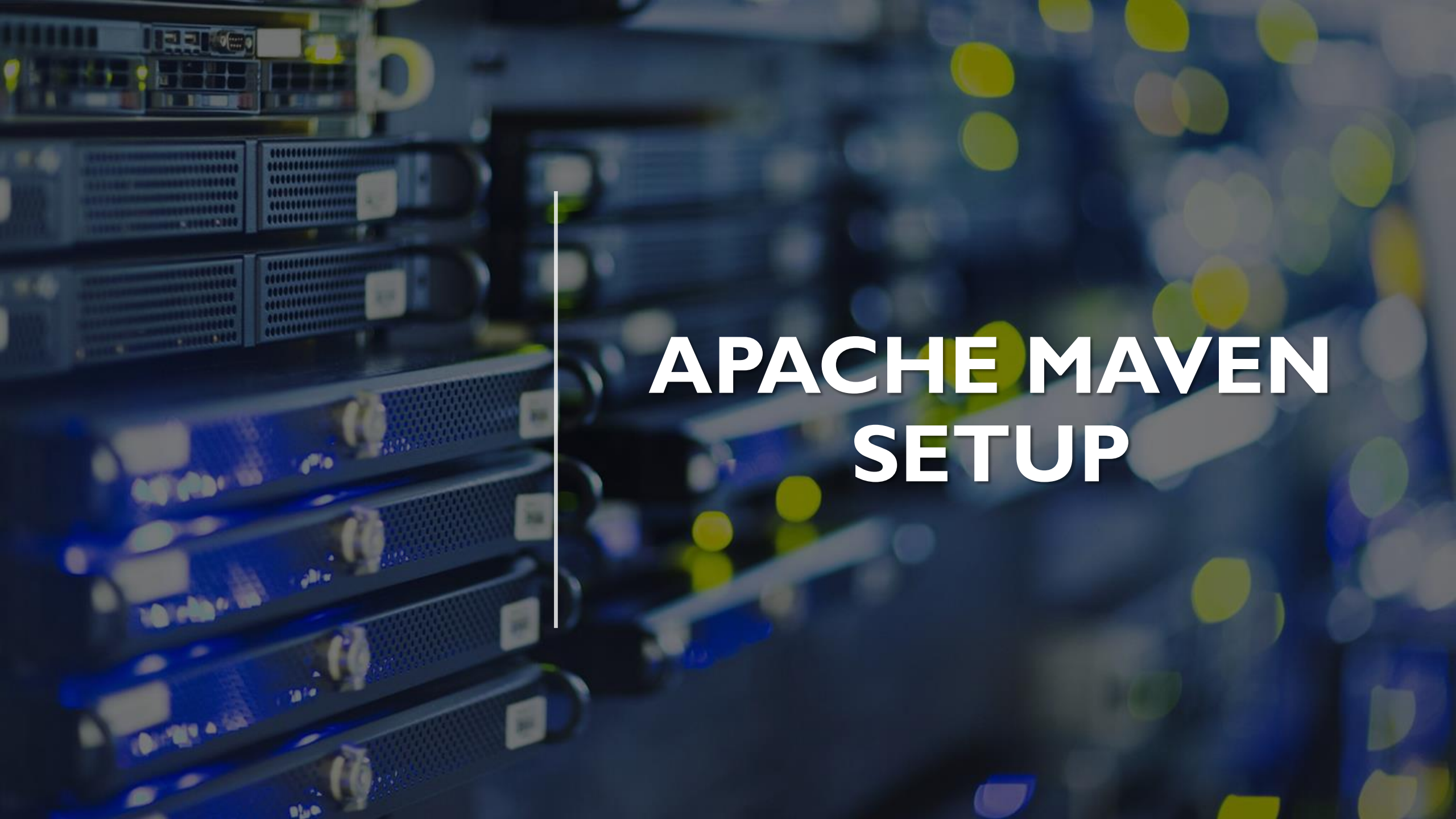

<project>
<modelVersion>4.0.0</modelVersion>
<groupId>no.uio.inf5750</groupId>
<artifactId>assignment-2</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>jar</packaging>
<name>Assignment 2</name>
<dependencies>
<dependency>
<groupId>commons-logging</groupId>
<artifactId>commons-logging</artifactId>
<version>1.4</version>
<scope>compile</scope>
</dependency>
</dependencies>
</project>
Object model version
Group / organization id
Id of the project itself
Version of the project
Packaging type
Display name of the project
Dependencies

WAR FILE

In software engineering, a **WAR file** (Web Application Resource or Web application ARchive) is a file used to distribute a collection of JAR-**files**, JavaServer Pages, Java Servlets, Java classes, XML **files**, tag libraries, static web pages (HTML and related **files**) and other resources that together constitute a web

JAR FILE

A JAR (**J**ava **AR**chive) is a package file format typically used to aggregate many Java class files and associated metadata and resources (text, images, etc.) into one file for distribution. JAR files are archive files that include a Java-specific manifest file.


APACHE MAVEN
SETUP