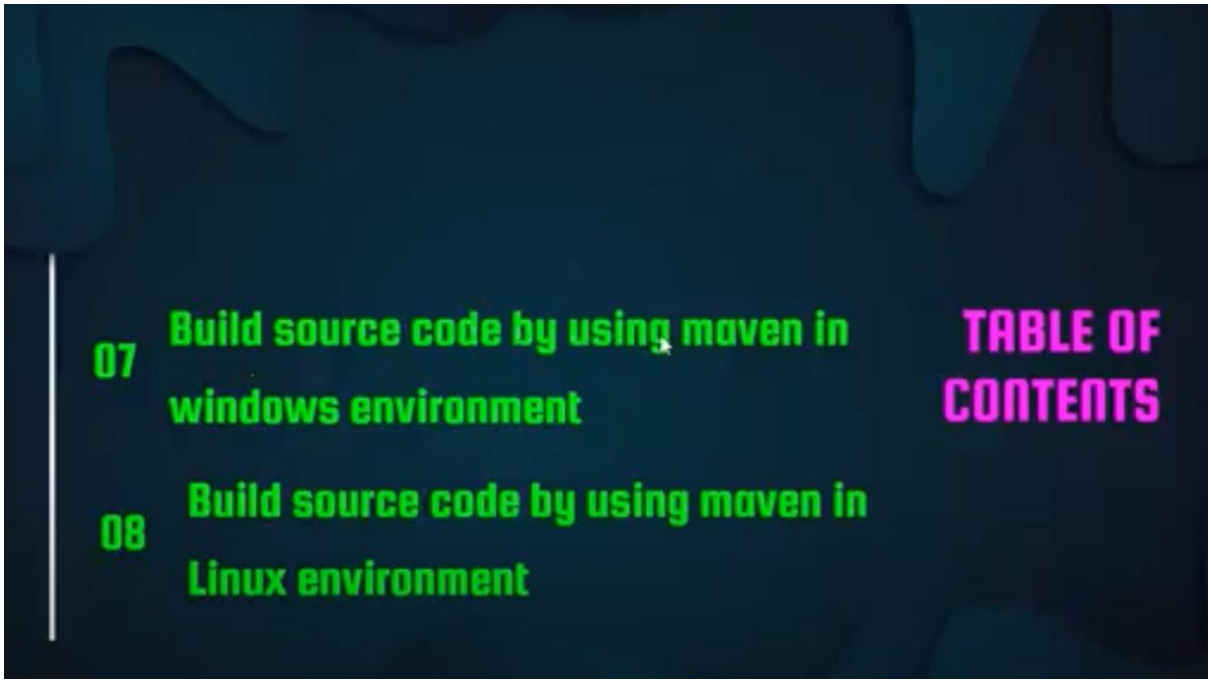
BUILD TOOL (MAVEN)

TABLE OF CONTENTS



Introduction to Build tool

1. A build tool takes care of everything for building a process.

➔ It does the following:

➢ Generate source code

➢ Generates documentation from source code

➢ Compiles source code

➢ Packages compiled code into JAR of ZIP file

Introduction to Build tool

- Installs the packaged code in local repository, server repository or central repository

Introduction to Maven

1. Maven is a build tool
2. It always produce an artifact or component or war file
3. It always helps to manage the dependencies
4. Maven can also use as project management tool
5. It handle the version and release
6. It is developed by apache software foundation

Introduction to Maven

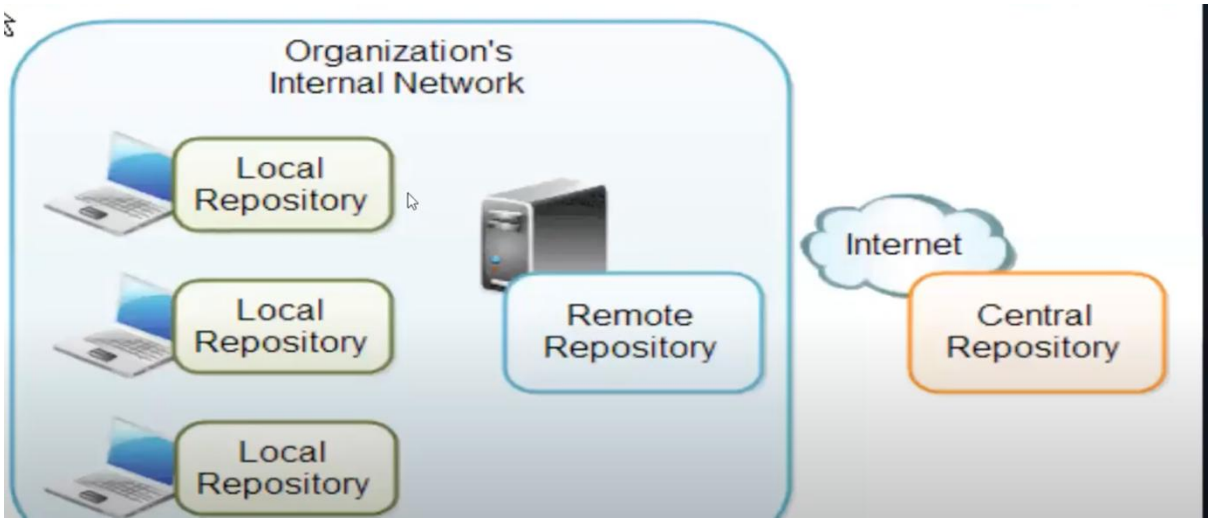
7. It is written in java language and used to build and manage projects written in any other languages

Uses of Maven

1. Use as build tool
2. Use as to manage project structure
3. Building, publishing and deploying
4. Documentation
5. Reporting
6. Releases
7. Distribution

What does Maven

1. We can create our builds for any environment
2. Downloading dependency with also pull other components
3. Work with local repo ie., .m2
4. Work with IDE(integrated development environment)
5. It can work with CI tools like jenkins



Maven build life cycle

1. Maven build follows a specific life cycle to deploy and distribute the target project
2. There are three built in life cycles like below:

- Default: the main life cycle as it is responsible for project deployment
- Clean: to clean the project and remove all files generated by the previous build

Maven build life cycle

- Site: To create the project's site documentation

3. Each life cycle consists of a sequence of phases
4. The default build life cycle consists of 23 phases as it is the main build lifecycle
5. The clean life cycle consists of 3 phases while the site life cycle is made up of 4 phases

Maven Default build lifecycle phases or goals

1. Some of the most important phases in the default build life cycle:

- compile: compile the source code as well as download the dependencies jar files available in pom.xml
- test: run unit tests

Maven Default build lifecycle phases or goals

- package: package compiled source code into the distributable format(JAR, WAR and EAR etc...)
- install: install the package to a local repository ie., .m2

Maven pom.xml file

1. POM is an acronym for Project Object Model.
2. The pom.xml file contains information of project and configuration information for the maven to build the project such as dependencies, build directory, source directory, test source directory, plugin, goals etc.
3. Maven reads the pom.xml file then executes the goals

Elements of maven pom.xml file

project	It is the root element of pom.xml file
modelVersion	It is the sub element of project. It specifies the modelVersion. It should be set 4.0.0
groupId	It is the sub element of project. It specifies the id for the project group

Elements of maven pom.xml file

artifactId	It is the sub element of project. It specifies the id for the artifact(project). An artifact is something that is either produced or used by a project. Examples of artifacts produced by maven for a project include: JARs, source and binary distributions and WARs
version	It is the sub element of project. It specifies the version of the artifact under given group

Elements of maven pom.xml file

packaging	Defines packaging type such as JAR, WAR and EAR etc....

Maven pom.xml file

```
<project>
<modelVersion>4.0.0</modelVersion>
<groupId>wipro.raviLogin</groupId>
<artifactId>raviLogin</artifactId>
<version>1.0</version>
<packaging>war</packaging>
</project>
```

Build source code by using maven in windows environment

1. prerequisites:

- Install jdk and maven
- Verify jdk ie., java -version
- Verify maven ie., mvn -version
- Verify git ie., git --version