**MAVEN**

* **Maven**

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 from a central piece of information

* **Maven Repository**

A Maven repository is a location where all the project jars, library jars, plugins or any other particular project related artifacts are stored and can be easily used by Maven.

Their types are local, central and remote

* **Local Repository**

Maven local repository is a folder location on your machine. It gets created when you run any maven command for the first time. Maven local repository keeps your project's all dependencies (library jars, plugin jars etc).

* **Central Repository**

It is repository provided by Maven community. It contains a large number of commonly used libraries. When Maven does not find any dependency in local repository

* **Remote Repository**

Sometimes, Maven does not find a mentioned dependency in central repository as well then it stops the build process and output error message to console. To prevent such situation, Maven provides concept of Remote Repository which is developer's own custom repository containing required libraries or other project jars.

* **Maven Build Lifecycle**

A Build Lifecycle is a well-defined sequence of phases which define the order in which the goals are to be executed. Here phase represents a stage in life cycle

The three build lifecycles are −

* **Clean**: cleans up artifacts created by prior builds.
* **Default (or build)**: This is used to build the application.
* **Site**: generates site documentation for the project.
* **Maven Artifact**

An artifact is a file, usually a JAR that gets deployed to a Maven repository. A Maven build produces one or more artifacts, such as a compiled JAR and a "sources" JAR. Each artifact has a group ID (usually a reversed domain name, like com.example.foo), an artifact ID (just a name), and a version string. The three together uniquely identify the artifact. A project's dependencies are specified as artifacts.

* **Difference between Ant and Maven**

|  |  |
| --- | --- |
| **Ant** | **Maven** |
| Ant **doesn't have formal conventions**, so we need to provide information of the project structure in build.xml file. | Maven **has a convention** to place source code, compiled code etc. So we don't need to provide information about the project structure in pom.xml file. |
| Ant is **procedural**, you need to provide information about what to do and when to do through code. You need to provide order. | Maven is **declarative**, everything you define in the pom.xml file. |
| There is **no life cycle** in Ant. | There is **life cycle** in Maven. |
| It is **a tool** box. | It is **a framework**. |
| It is **mainly a build tool**. | It is **mainly a project management tool**. |
| The ant scripts are **not reusable**. | The maven plugins are **reusable**. |

* **POM**

POM stands for Project Object Model. It is fundamental Unit of Work in Maven. It is an XML file. It always resides in the base directory of the project as pom.xml. It contains information about the project and various configuration details used by Maven to build the project(s).

* **Elements in POM**

|  |  |
| --- | --- |
| **Element** | **Description** |
| **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 to 4.0.0. |
| **groupId** | It is the sub element of project. It specifies the id for the project group. |
| **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. |
| **packaging** | Defines packaging type such as jar, war etc. |
| **name** | Defines name of the maven project. |
| **url** | Defines url of the project. |
| **dependencies** | Defines dependencies for this project. |
| **dependency** | Defines a dependency. It is used inside dependencies. |
| **scope** | Defines scope for this maven project. It can be compile, provided, runtime, test and system. |