MAVEN OVERVIEW

What is Maven?

Maven is a project management and comprehension tool. Maven 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.

In case of multiple development teams environment, Maven can set-up the way to work as per standards in a very short time. 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 following:

- Builds

- Documentation

- Reporting

- Dependencies

- SCMs

- Releases

- Distribution

- mailing list

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 build related tasks.

Maven History

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.

Maven Objective

Maven primary goal is to provide developer

- 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. Refer to Maven POM section for more detail.

Convention over Configuration

Maven uses *Convention* over *Configuration* which means developers are not required to create build process themselves.

Developers do not have to mention each and every configuration detail. Maven provides sensible default behavior for projects. When a Maven project is created, Maven creates default project structure. Developer is only required to place files accordingly and he/she need not to define any configuration in pom.xml.

As an example, following table shows the default values for project source code files, resource files and other configurations. Assuming, **${basedir}** denotes the project location:

Item	Default
source code	${basedir}/src/main/java
resources	${basedir}/src/main/resources
Tests	${basedir}/src/test
Complied byte code	${basedir}/target/classes
distributable JAR	${basedir}/target

In order to build the project, Maven provides developers options to mention life-cycle goals and project dependencies (that rely on Maven pluging capabilities and on its default conventions). Much of the project management and build related tasks are maintained by Maven plugins.

Developers can build any given Maven project without need to understand how the individual plugins work. Refer to Maven Plug-ins section for more detail.