Apache Maven: Overview

What is Maven?

Conceptionally

- standardized build infrastructure for software projects

Functionally

- build tool
- dependency management tool
- quality tool

Organisationally

- open source project (ASF)

Philosophy

Principles/Goals

- convention over configuration
- ease the build process
- best practises patterns
- consistent build

Project Descriptor

POM - „Project Object Model“

- pom.xml
- contains all metadata of a project
  - name
  - packaging type
  - dependencies
  - tool links
  - license
  - ...
- one POM == one artifact
- POM relationships
  - inheritance
  - aggregation

POM example

```
<?xml version="1.0" encoding="UTF-8"?>
<project ...>
  <groupId>de.tuberlin.dima.aim</groupId>
  <artifactId>core</artifactId>
  <version>1.0-SNAPSHOT</version>
  <name>AIM Core Modules</name>
  ...
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.8.1</version>
      <scope>test</scope>
    </dependency>
    ...
  </dependencies>
</project>
```

Project Layout

Standard directories

- src/main/java
- src/main/resources
- src/test/java
- src/test/resources
- src/site
- target

Build Lifecycle

Cycles

- default
- clean
- site

Phases

- validate
- compile
- test
- package
- integration-test
- verify
- install
- deploy

Plugins

Integration

- can be „plugged“ into the build lifecycle
- allow customization and additional features

Examples

- compiler
- release
- checkstyle
- findbugs

Packaging Type

Used to

- specify artifact's type
- configure *default* lifecycle

Examples

- jar (default)
- war
- ear
- pom
- ...

Dependencies and repositories

Dependencies

- identified by: groupId, artifactId, version, scope
- scope affects transitivity and classpath
- available scopes: compile, provided, runtime, test

Repositories

- contains artifacts and plugins (versioned) with associated POMs
- official remote repositories
- local repository for caching and customization

Profiles

Dependencies

- adapt a build for a particular environment / user
- defined as part of pom.xml or local settings.xml
- can be: active by default, activated based on conditions or enabled on command line

Examples

- development
- production
- fastinstall

Attribution

Derived from

FastConnect „Maven Overview“

available at http://www.slideshare.net/fastconnect/maven-overview-presentation