# Versionen

modelVersion = Version der POM.xsd

|  |  |
| --- | --- |
| **Maven Version** | **modelVersion** |
| 1.1 | 3.0.0 |
| Ab 2.0 (akt. 3.2.5) | 4.0.0 |

Maven 3 unterscheidet sich grundlegend von Maven 2 – z.B. der Standard für Unterprojekte.

# Allgemeines

* Convention over Configuration
* POM: Project Object Model

# Lifecycles (jeder durch Plugin ersetzbar)

Eingebaute Lifecycles:

* default
* clean
* site

## Default lifecycle phases

1. validate
2. compile
3. test z.B. JUnit, Surefire
4. package z.B. jar, war
5. integration-test
6. verify
7. install lokales maven repo .m2
8. deploy fernes maven repo, z.B. Apache Archiva

s. Tag <distributionManagement>

## Clean lifecycle

clean up artifacts from prior builds

## Site lifecycle

generate site documentation

## Goals

Eine Build phase besteht aus goals.

* archetype Projekt template (Bsp. s.u.)
* dependency z.B. dependency:copy-dependencies

*Bemerkungen:*

* Lifecycles sind durch Plugins erweiterbar.
* Spätere Phase inkludieren vorherige Phasen. D.h. install führt auch package durch.
* Phasen werden in der angegebenen Reihenfolge durchgeführt.
* Lifecycle kann vom packaging abhängen
* complete list of the build phases: http://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html#Lifecycle\_Reference

# Standard-Verzeichnisstruktur (convention)

|  |  |  |
| --- | --- | --- |
| (Kurzform) |  | Assembly descriptors  Integration Tests (primarily for plugins)  Configuration files  Resource filter files  Application/Library sources  Application/Library resources  Application/Library scripts  Web application sources  Site  Test resource filter files  Test sources  Test resources  Project's license  Notices and attributions required by libraries  Project‘s configuration  Project’s readme |

## beans.xml

For EJB and JAR packaging you should place the beans.xml in src/main/resources/META-INF/.

For WAR packaging you should place the beans.xml in src/main/webapp/WEB-INF/.

# Repos

* Öffentliche Maven Repos: Apache, ibiblio.org, Codehaus, Java.Net
  + http://repo.maven.apache.org/maven2/
* Firmenweite Repos: Apache Archiva, Nexus Sonatype, Artifactory, …
* Lokales Maven Repo: C:\Users\Markus\.m2

# Maven-Config

* Globale maven-Config: C:\Program Files (x86)\apache-maven-3.2.5\conf\settings.xml
* User: C:\Users\Markus\.m2\settings.xml
* Projekt: pom.xml

# Maven Projekt erstellen

* mvn archetype:generate -DgroupId=org.beispielurl -DartifactId=meine-anwendung
* mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

## Doku erstellen

mvn archetype:generate -DarchetypeGroupId=org.apache.maven.archetypes -DarchetypeArtifactId=maven-archetype-site -DgroupId=com.mycompany.app -DartifactId=my-app-site

## Web App erstellen

mvn archetype:generate -DarchetypeGroupId=org.apache.maven.archetypes -DarchetypeArtifactId=maven-archetype-webapp -DgroupId=com.mycompany.app -DartifactId=my-webapp

## Maven Projekt in eclipse Projekt konvertieren

* mvn eclipse:eclipse

Erzeugt .classpath und .project

## Maven Projekt in Eclipse erstellen

Project Explorer > Rechtsklick > New Project > Filter „Maven“> Maven Project

Im Allgemeinen „Create simple project (skip archetype selection)“.

## Fix Maven Project

1. Do a non-destructive delete of the project from Eclipse i.e. do NOT check "Delete project contents on disk"
2. In the file system, remove the .settings/, .classpath, and .project files.
3. Also in the file system, add the resources folders under src/main and src/test.
4. In Eclipse, do an "Existing Maven Projects" import with the project's folder as the root.

# Plugin

„a *plugin* is a collection of *goals* with a general common purpose“

## Plugin ausführen

mvn <groupId>:<artifactId>:<goal>@<execution id from pom>

# Java Projekt starten

java -cp my-app-1.0-SNAPSHOT.jar com.mycompany.app.App

# Eclipse-Konfiguration: embedded vs. selbst installiertes maven

Wählbar in Run Configurations, beides möglich.

# Versionierung

1. Kommandozeile interaktiv:

mvn versions:set

1. Kommandozeile nicht interaktiv:

mvn versions:set -DnewVersion=

1. Eclipse

Run > Run Configurations

Goals: versions:set

Parameter Name: newVersion

Value: Gewünschter Wert

# Variablen

Variablen beginnen immer mit project, werden getrennt durch . (Punkt). Referenziert wird immer ein Blatt (Feldwert). Beispiel: ${project.version}.

Properties können direkt referenziert werden. Beispiel: <property><myProperty>… 🡪 ${myProperty}

Vererbung geht vor Variablen-Prozessierung. D.h. wenn die Kind-POM eine eigene Version hat, referenziert ${project.version} in der Kind-POM auf diese Version, nicht auf die vom Parent.

## Spezielle Variablen

|  |  |
| --- | --- |
| project.basedir | The directory that the current project resides in. |
| project.baseUri | The directory that the current project resides in, represented as an URI. |
| maven.build.timestamp | The timestamp that denotes the start of the build.  Kann durch die Property maven.build.timestamp.format konfiguriert werden (Angabe im SimpleDateFormat). |

# Vererbung und Aggregation

|  |  |  |
| --- | --- | --- |
| **Technik** | **Merkmale** | **Vorteil** |
| Vererbung | Kind-POM hat Element <parent> | Viele Elemente(s.u.) vererben sich und müssen für mehrere Projekte nur einmal definiert werden |
| Aggregation | ParentPOM hat Element <modules>  ParentPOM hat Packing pom | Phase/goal auf Parent wird auf alle Module angewendet |

## Eigenschaften

* Vererbung und Aggregation können kombiniert werden!
* Folgende Elemente vererben sich:
  + dependencies
  + developers and contributors
  + plugin lists (including reports)
  + plugin executions with matching ids
  + plugin configuration
  + resources

## Non-Default-Directory-Structure

Best practice ist es, die Module in Unterverzeichnissen vom ParentPOM zu haben. Stärkstes Argument: das versions-Plugin baut darauf. Das release-Plugin wiederum nutzt das versions-Plugin.

Wenn aus irgendeinem Grund diese Struktur nicht verwendet werden kann/soll (sondern z.B. die ParentPOM parallel zu den Modulen liegt):

* muss in den <modules> der relative Pfad angegeben werden:

<module>../childModule</module>

* muss in <parent> der relative Pfad angegeben werden:

<relativePath>../parent/pom.xml</relativePath>

## Reactor Builds

Build über aggregierte Module

# Minimale POM

Besteht aus modelVersion, groupId, artifactId, version:

<project>

<modelVersion>4.0.0</modelVersion>

<groupId>com.mycompany.app</groupId>

<artifactId>my-app</artifactId>

<version>1</version>

</project>

Fully qualified artifact name: <groupId>:<artifactId>:<version>