Plan

- What is Maven ?
- Links :
- mvn command line tool
- POM : 1 pom.xml = 1 artifact
- POM
- POM Inheritance
- Standard Directory Layout
- Demo on JMMC projects
- Plugins
- Conclusion

What is Maven ?

Apache Maven is a software project management and comprehension tool (build, test, packaging, reporting, site, deploy).

!= Apache ant (make like)

Key features :

- POM : Project Object Model
  - Dependency management
  - Convention over configuration
- Plugins
- IDE integration (eclipse, netbeans ...)

Links :

Project page :
http://maven.apache.org/

Plugins :
http://maven.apache.org/plugins/index.html

Maven book :
http://books.sonatype.com/mvnref-book/reference/index.html

En français :
http://maven-guide-fr.erwan-alliaume.com/

Mvn repository :
http://mvnrepository.com/

mvn command line tool

Install Maven (Java tool) => mvn command

Settings : <home>/.m2/settings.xml
Local repository : <home>/.m2/repository/

mvn install:install-file (missing library)
mvn clean
mvn package
mvn clean install (into local repository)
mvn test
...

GOALS

Easy to use Maven in IDE

.m2/settings.xml

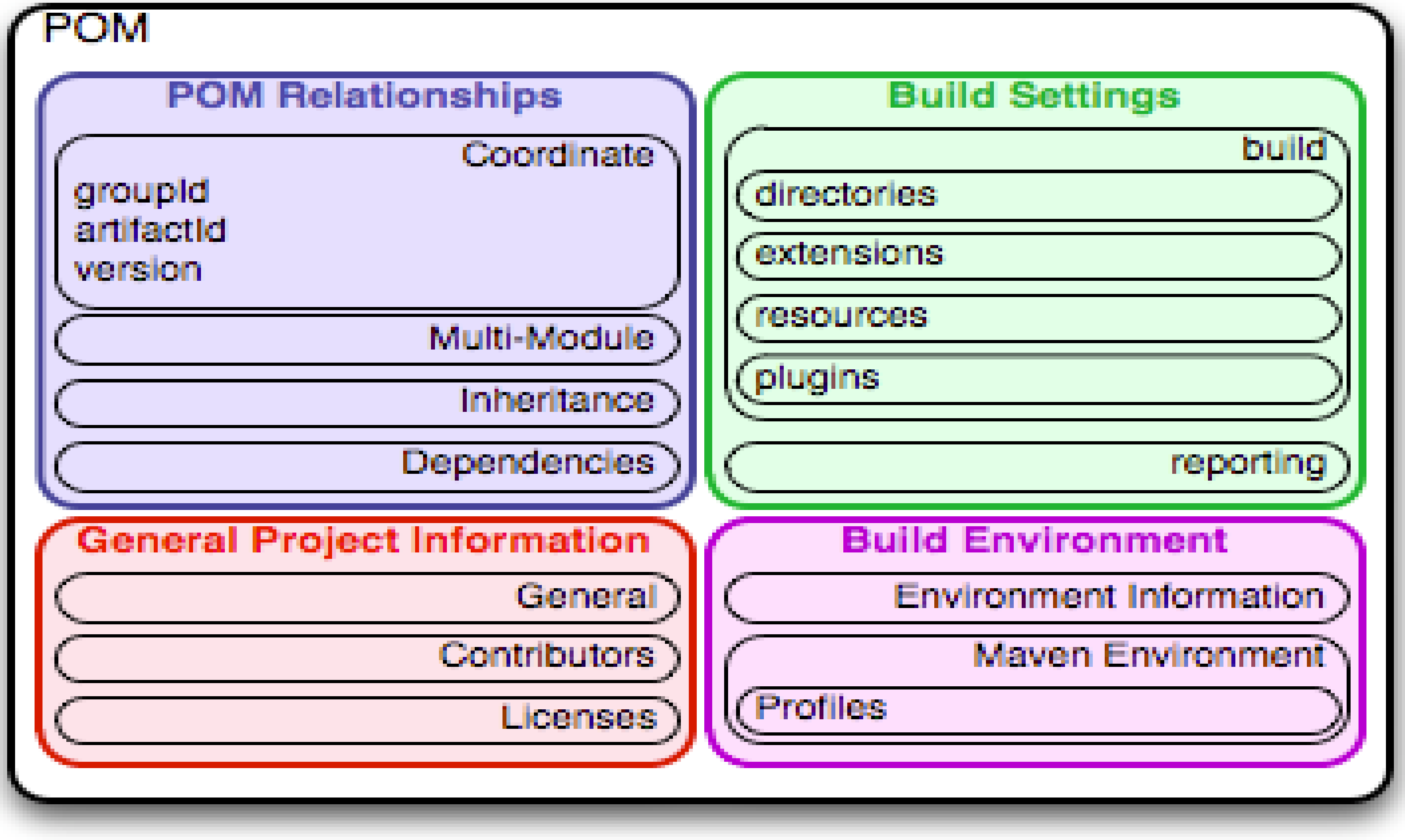
```
<settings>
    <offline>false</offline>

    <proxies>
        <proxy>
            <active>true</active>
            <protocol>http</protocol>
            <host>www-cache.ujf-grenoble.fr</host>
            <port>3128</port>
            <nonProxyHosts>*.jmmc.fr</nonProxyHosts>
        </proxy>
    </proxies>

    <profiles>
        <profile>
            <id>dev</id>
            <properties>
                <!-- disable jar signer -->
                <jarsigner.skip>true</jarsigner.skip>
                <!-- disable javadoc -->
                <maven.javadoc.skip>true</maven.javadoc.skip>
                <!-- disable tests -->
                <maven.test.skip>true</maven.test.skip>
            </properties>
        </profile>
    </profiles>

    <activeProfiles>
        <activeProfile>dev</activeProfile>
    </activeProfiles>
</settings>
```

POM : 1 pom.xml = 1 artifact



POM

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<project xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>

  <parent>
…
  </parent>

  <groupId>fr.jmmc.jmcs</groupId>
  <artifactId>jmcs</artifactId>
  <version>X.Y</version>
  <packaging>jar</packaging>
...
  <properties>
...
  </properties>

  <dependencies>
...
  </dependencies>

  <build>
    <plugins>
...
    </plugins>
  </build>

</project>
```

INHERITANCE

```
<parent>
  <groupId>fr.jmmc</groupId>
  <artifactId>jmmc</artifactId>
  <version>TRUNK</version>
</parent>
```

ARTIFACT

DEPENDENCY

```
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>jul-to-slf4j</artifactId>
  <version>1.7.6</version>
</dependency>
```

PLUGIN

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.1</version>
  <configuration>
    <source>1.6</source>
   <target>1.6</target>
 </configuration>
</plugin>
```

POM Inheritance

Override default configuration in POM :

1) Maven settings
2) Parent POM(s) (hierarchy)
3) POM (current project)
4) Active Profiles (dev) to finally set properties, login/password and env settings ...

=> Use Effective POM or an IDE

```
mvn help:effective-pom
```

Standard Directory Layout

http://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html

Use configuration to customize folders to your project setup !

src/main/java	Source code
src/main/resources	Resource files
src/main/config	Configuration files
src/main/scripts	Scripts
src/main/webapp	Web application source
src/test/java	Test source code
src/test/resources	Test resource files
src/site	Site
pom.xml	POM
LICENSE.txt	License
target/	**Build output**

Demo on JMMC projects

Many artifacts :

- jmmc (super pom) gathers common config
- Jmcs library + deps
- Jmal library + deps
- …
- Aspro2 application

Multi-module approach : exist oidb project

Interest :

- dependencies are shared and transitive
- final package gathers all jar files + code signing

Plugins

http://maven.apache.org/plugins/index.html

- resource filtering (variable substitution)
- site : generate a web site for the project
- surefire : unit tests (junit)
- packaging : jar, war, source ...
- reporting : javadoc, pmd, checkstyle, changelog …

Tools :

- antrun : executes custom ant tasks
- archetype : create a new project (template)
- assembly (packaging) : jar with deps, zip, project or custom
- dependency : copy dependencies …
- scm : versioning integration (svn, git)
- tomcat : run + deployment

Conclusion

Very powerful but …
- A lot of conventions / configurations and Plugins
= a bit cumbersome to find what you need
- Easier if you adopt the standard directory layout

Other language support :
- PHP : http://www.php-maven.org/
- C/C++ : http://maven-nar.github.io/