TMA Training Center (TTC)

Maven 2.0
Improve your build patterns

Course	Maven 2.0
Trainer	Son Nguyen
Designed by	Son Nguyen- AxS
Last updated	<Date>

Contents

- What is Maven?
- Maven Architecture
- Build patterns
- Maven 2 plugins

Course Objectives

Overall Presentation Goal

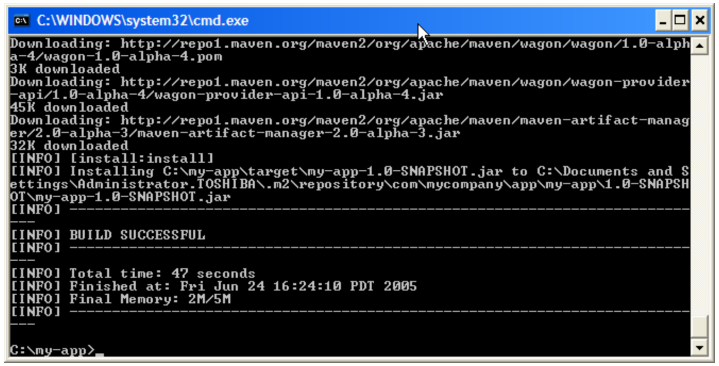
Discover Maven 2.0 through build patterns

Making your builds boring…

Building projects should be easy and standardized. You should not be spending a substantial amount of your project time on builds. Builds should just work!

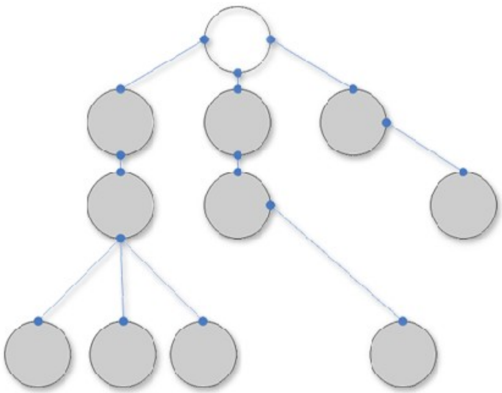
What is Maven? (1/2)

A build tool!

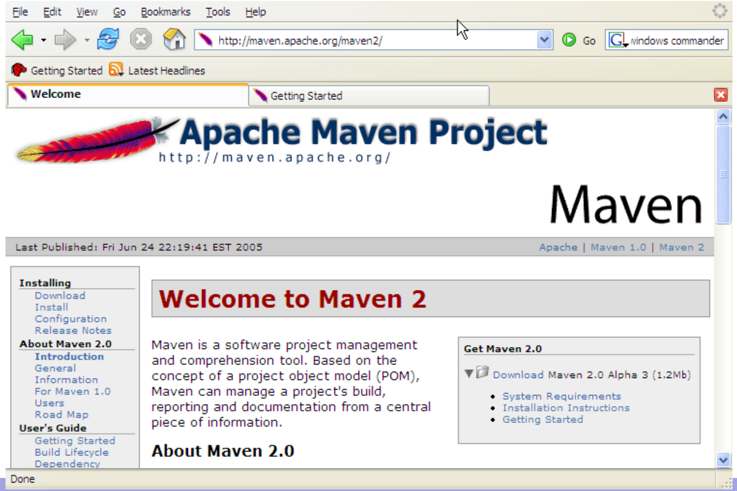


```
Downloading: http://repo1.maven.org/maven2/org/apache/maven/wagon/wagon/1.0-alph
a-4/wagon-1.0-alpha-4.pom
3K downloaded
Downloading: http://repo1.maven.org/maven2/org/apache/maven/wagon/wagon-provider
-api/1.0-alpha-4/wagon-provider-api-1.0-alpha-4.jar
45K downloaded
Downloading: http://repo1.maven.org/maven2/org/apache/maven/maven-artifact-manag
er/2.0-alpha-3/maven-artifact-manager-2.0-alpha-3.jar
32K downloaded
[INFO] [install:install]
[INFO] Installing C:\my-app\target\my-app-1.0-SNAPSHOT.jar to C:\Documents and S
ettings\Administrator.TOSHIBA\.m2\repository\com\mycompany\app\my-app\1.0-SNAPSH
OT\my-app-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
---
[INFO] BUILD SUCCESSFUL
[INFO] ------------------------------------------------------------------------
---
[INFO] Total time: 47 seconds
[INFO] Finished at: Fri Jun 24 16:24:10 PDT 2005
[INFO] Final Memory: 2M/5M
[INFO] ------------------------------------------------------------------------
---

C:\my-app>_
```

A dependency management tool!



A documentation tool!



Maven

Last Published: Fri Jun 24 22:19:41 EST 2005

Apache | Maven 1.0 | Maven 2

Installing
- Download
- Install
- Configuration
- Release Notes

About Maven 2.0
- Introduction
- General Information
- For Maven 1.0
- Users
- Road Map

User's Guide
- Getting Started
- Build Lifecycle
- Dependency

Welcome to Maven 2

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.

Get Maven 2.0

▼ Download Maven 2.0 Alpha 3 (1.2Mb)

- System Requirements
- Installation Instructions
- Getting Started

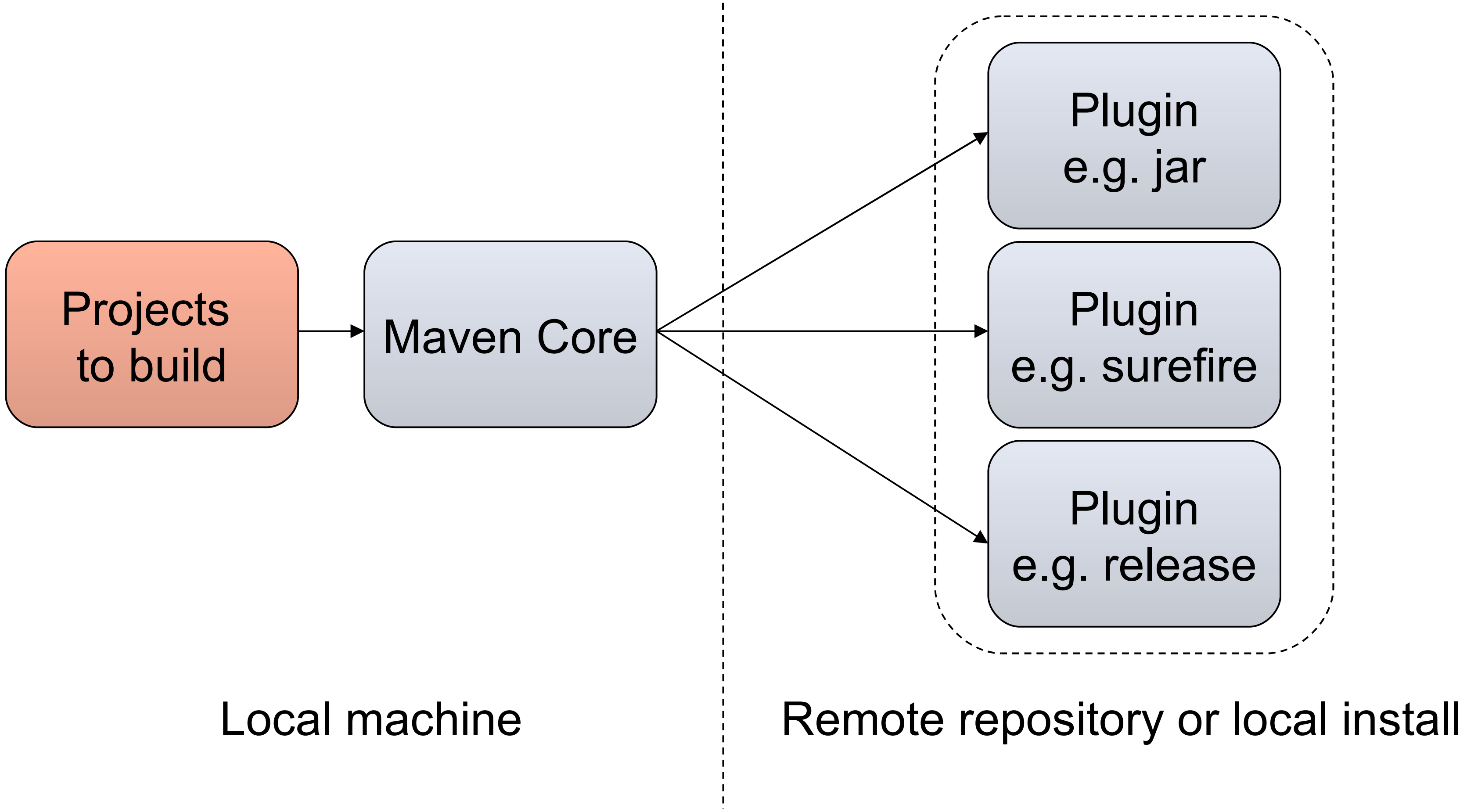
About Maven 2.0

What is Maven? (2/2)

Maven is really a process of applying patterns to a build infrastructure in order to provide a coherent view of software projects.

- Objectives
  - Make the development process visible or transparent
  - Provide an easy way to see the health and status of a project
  - Decreasing training time for new developers
  - Bringing together the tools required in a uniform way
  - Preventing inconsistent setups
  - Providing a standard development infrastructure across projects
  - Focus energy on writing applications

Maven Architecture



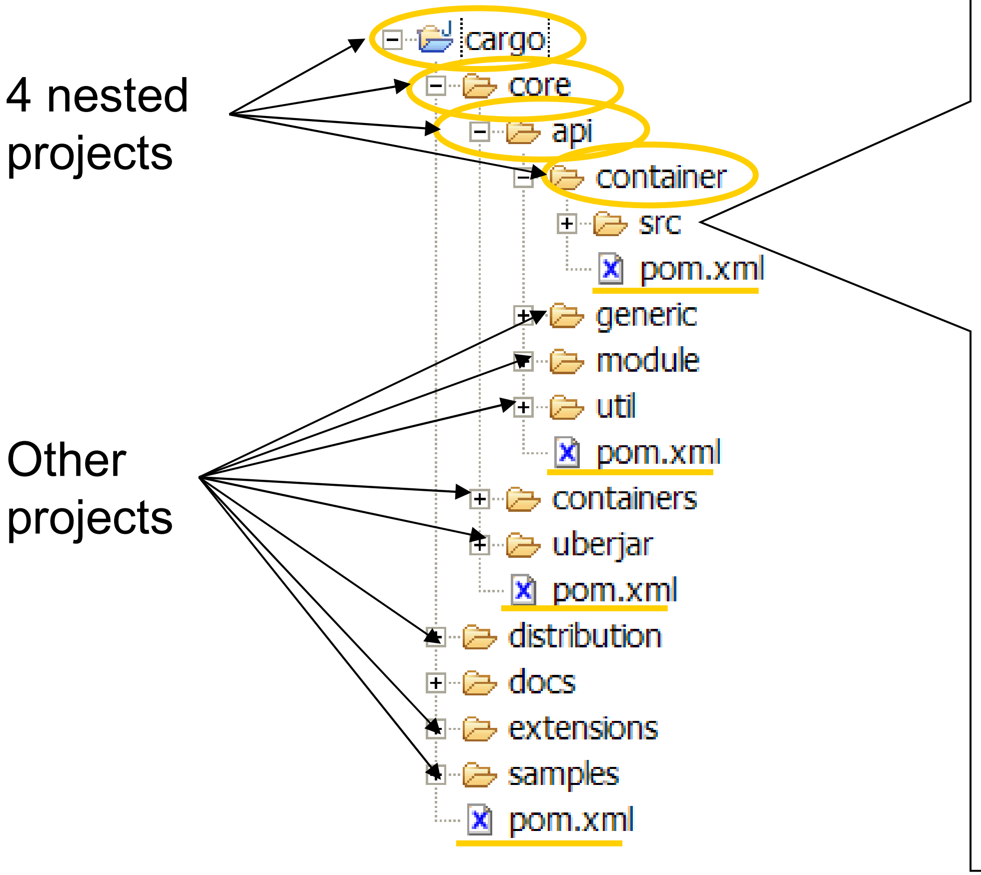
Common project metadata format

- POM = Project Object Model = pom.xml
- Contains metadata about the project
  - Location of directories, Developers/Contributors, Issue tracking system, Dependencies, Repositories to use, etc
- Example:

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.codehaus.cargo</groupId>
  <artifactId>cargo-core-api-container</artifactId>
  <name>Cargo Core Container API</name>
  <version>0.7-SNAPSHOT</version>
  <packaging>jar</packaging>
  <dependencies/>
  <build/>
[…]
```

Minimal POM

Common directory organization



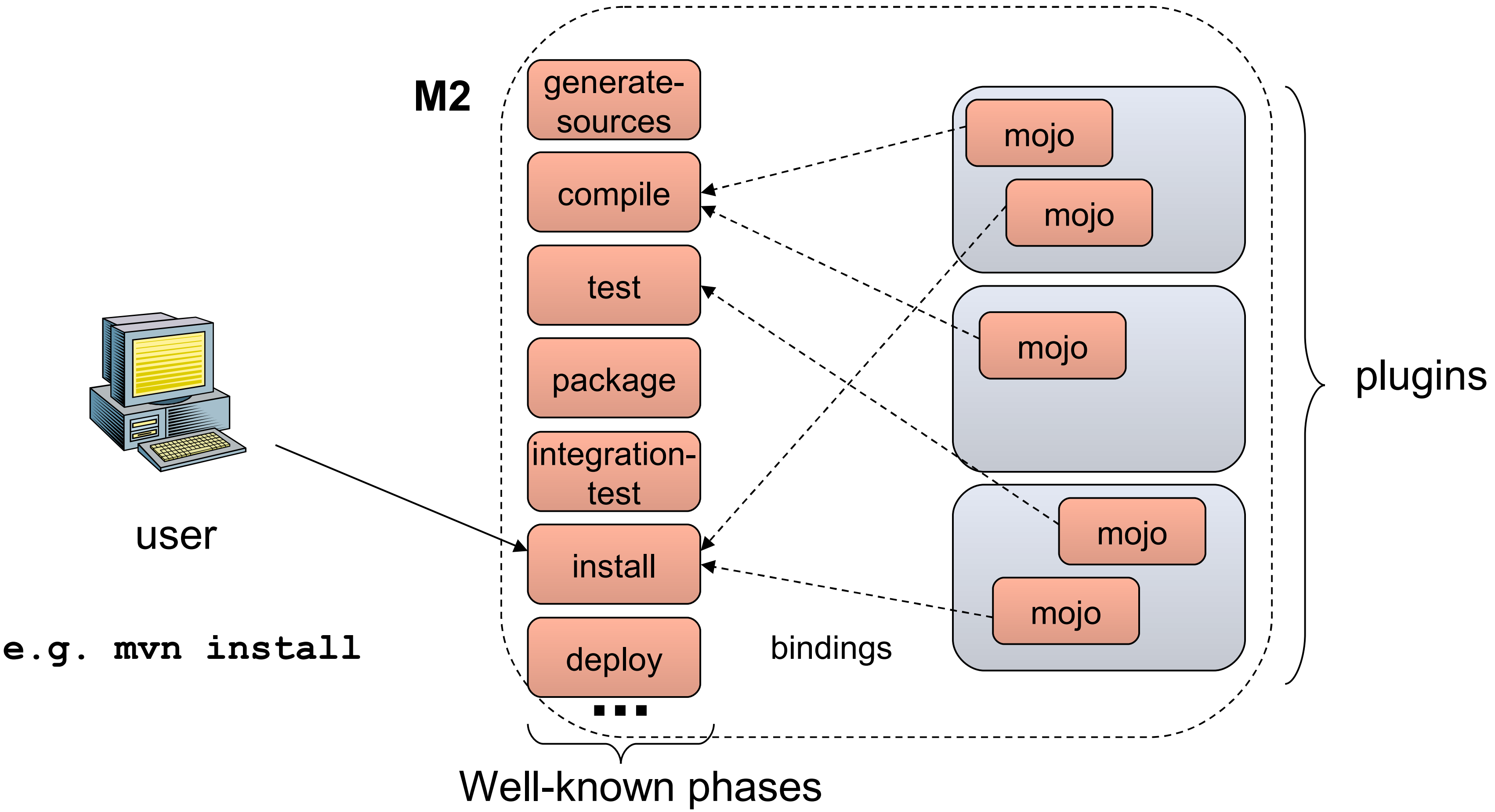
4 nested projects

Other projects

```
cargo
  core
    api
      container
        src
        pom.xml
      generic
      module
      util
      pom.xml
    containers
    uberjar
    pom.xml
  distribution
  docs
  extensions
  samples
  pom.xml
```

- src/
  - main/
    - java/
    - resources/
    - webapp/
    - application/
    - groovy/
  - test/
    - java/
    - resources/
    - cactus/
  - site/

Common way to build applications (1/2)



M2

generate-sources

compile

test

package

integration-test

install

deploy

...

Well-known phases

user

e.g. mvn install

bindings

mojo

mojo

mojo

mojo

mojo

plugins

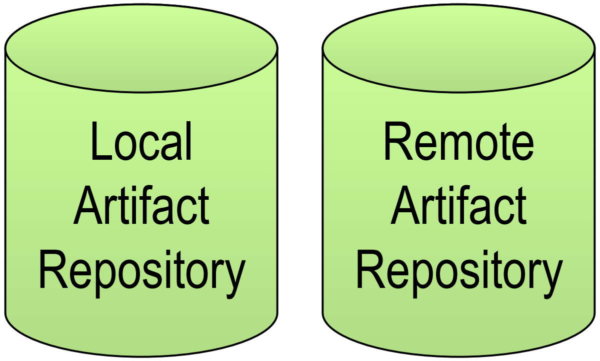
Common way to build applications (2/2)

- The lifecycle depends on the project type (packaging)
  - Defined in pom.xml (pom, jar, ear, war, etc)
  - Ex: `<packaging>jar</packaging>`
- User can modify lifecycles by adding a goal to a phase:

```
<plugin>
  <groupId>com.mycompany.example</groupId>
  <artifactId>touch-maven-plugin</artifactId>
  <executions>
    <execution>
      <phase>process-test-resources</phase>
      <configuration>[…]</configuration>
      <goals>
        <goal>timestamp</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

Artifact repositories (1/3)

- Used to store all kind of artifacts
  - JARs, EARs, WARs, NBMs, EJBs, ZIPs, plugins, …
- All project interactions go through the repository
  - No more relative paths!
  - Easy to share between teams



Local Artifact Repository

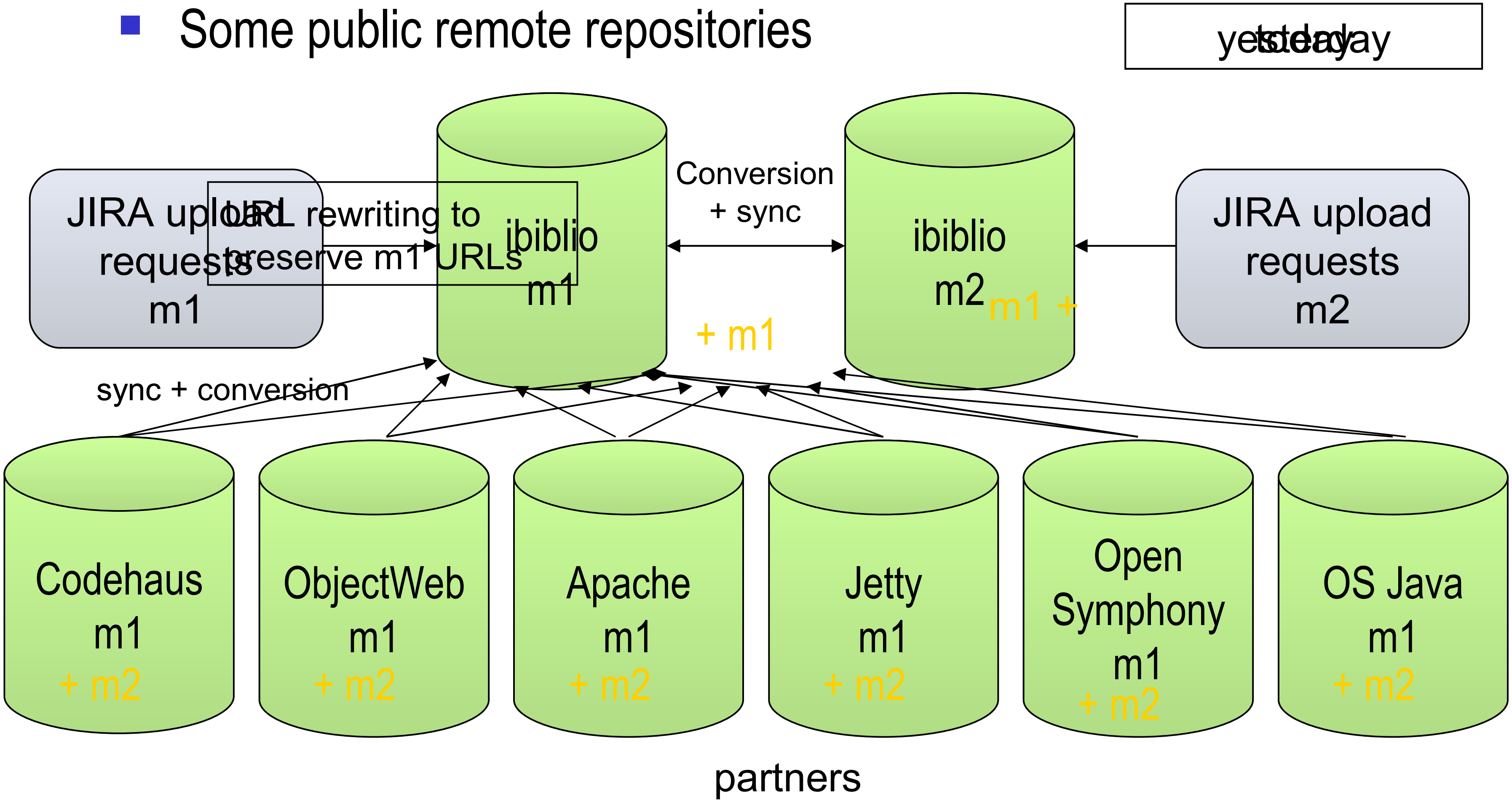
Remote Artifact Repository

e.g. http://ibiblio.org/maven2

```
<repositories>
  <repository>
    <id>maven2-snapshot</id>
    <releases>
      <enabled>true</enabled>
    </releases>
    <name>Maven Central Development Repository</name>
    <url>http://snapshots.maven.codehaus.org/maven2</url>
    <layout>legacy|default</layout>
  </repository>
</repositories>
```

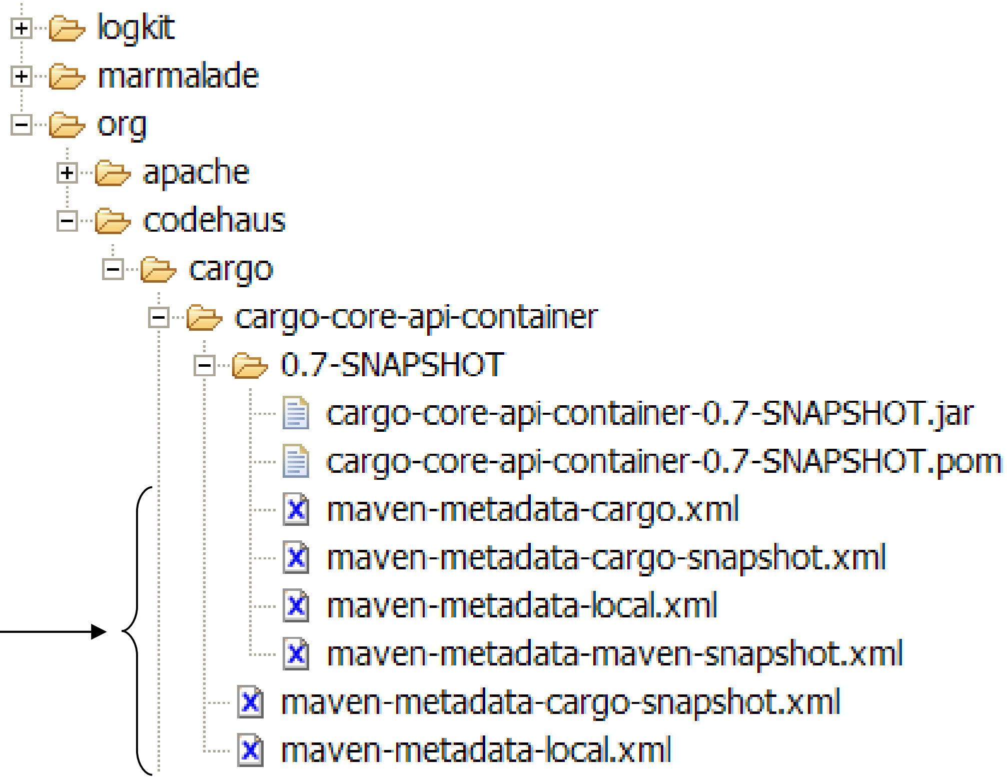
Artifact repositories (2/3)

- Some public remote repositories



Artifact repositories (3/3)

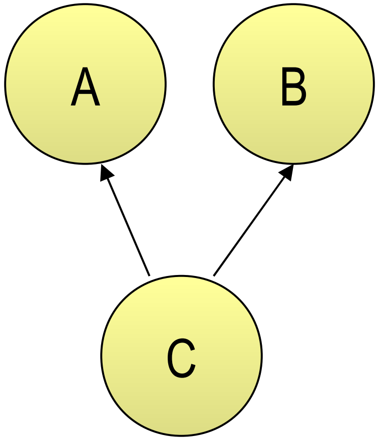
- Hierarchical structure
- Automatic plugin download
- Plugins are read directly from the repository
- Configurable strategies for checking the remote repositories for updates
  - Daily check by default for plugin and ranges updates
- Remote repositories contain Metadata information
  - Releases, latest, and more to come



```
logkit
marmalade
org
  apache
  codehaus
    cargo
      cargo-core-api-container
        0.7-SNAPSHOT
          cargo-core-api-container-0.7-SNAPSHOT.jar
          cargo-core-api-container-0.7-SNAPSHOT.pom
          maven-metadata-cargo.xml
          maven-metadata-cargo-snapshot.xml
          maven-metadata-local.xml
          maven-metadata-maven-snapshot.xml
        maven-metadata-cargo-snapshot.xml
        maven-metadata-local.xml
```

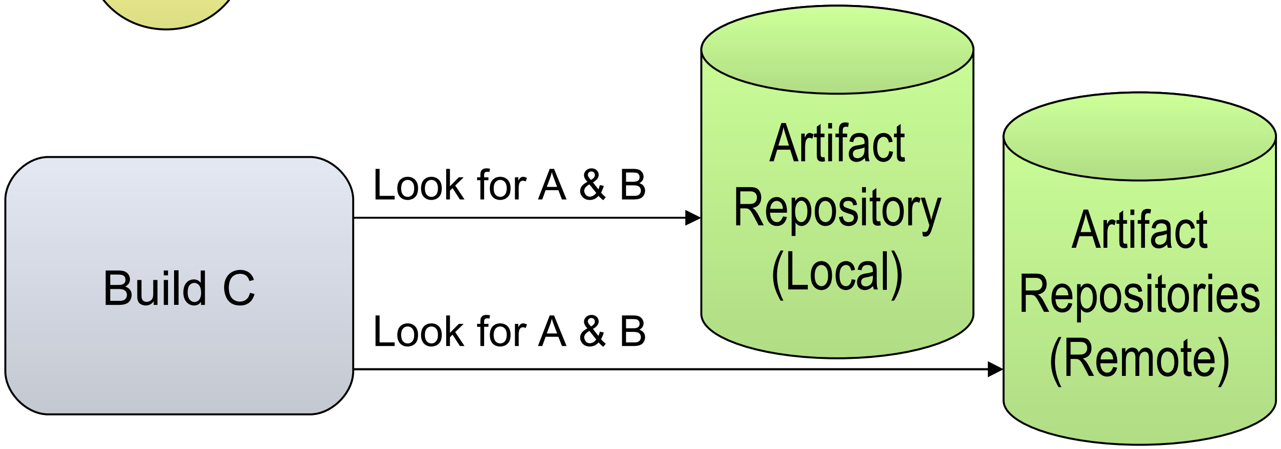
Dependency management (1/2)

- Maven uses binary dependencies



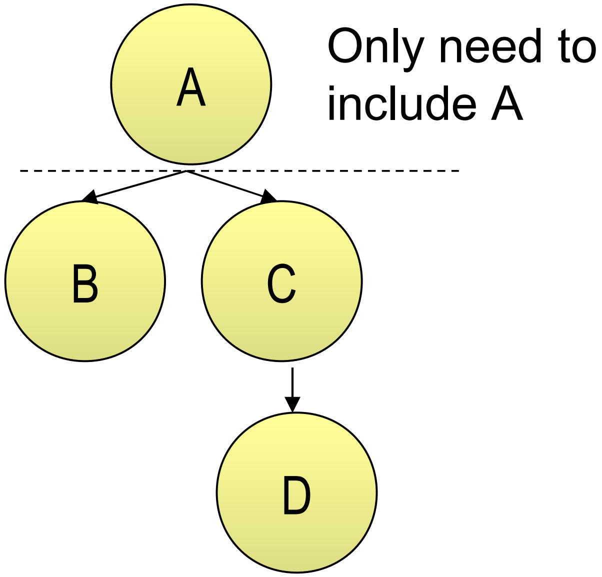
```
<dependencies>
  <dependency>
    <groupId>com.acme</groupId>
    <artifactId>B</artifactId>
    <version>[1.0,)</version>
    <scope>compile</scope>
  </dependency>
</dependencies>
```

« any version after 1.0 »



Dependency management (2/2)

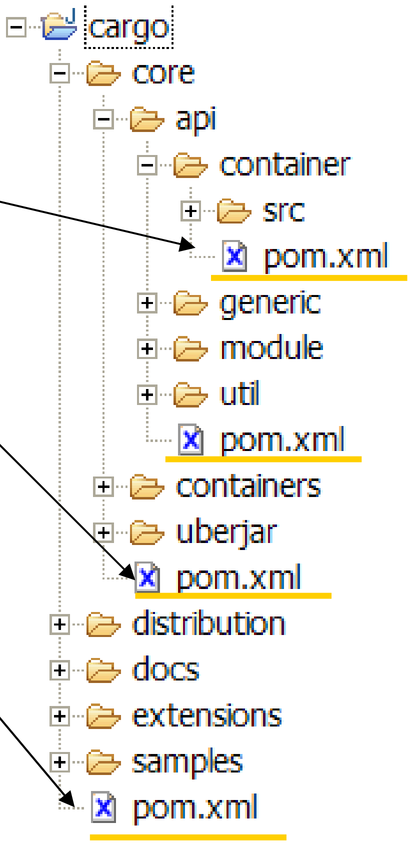
- Transitive dependencies
  - Possibility to exclude some deps
  - Need good metadata
  - Ideally projects should be split
- SNAPSHOT handling
  - Always get latest
- Automatic dep updates
  - By default every day



Multi-module builds

- Integrated into Maven 2
- Run « `mvn` » at parent level
  - E.g. **`mvn install`** in `cargo/core/api`
  - E.g. **`mvn install`** in `cargo/core`
  - E.g. **`mvn install`** in `cargo/`
- Declare children projects in parents:

```
<modules>
  <module>core</module>
  <module>extensions</module>
  <module>samples</module>
</modules>
```



Environment-dependent builds (1/2)

- Based on profiles
  - Located in pom.xml, in profiles.xml or in settings.xml

```
<profiles>
  <profile>
    <id>tomcat5x</id>
    <activation>
      <activeByDefault>true</activeByDefault>
    </activation>
    <properties>
      <containerId>tomcat5x</containerId>
      <downloadUrl>…jakarta-tomcat-5.0.30.zip</downloadUrl>
    </properties>
  </profile>
  <profile>
    <id>orion2x</id>
    <properties>
      <containerId>orion2x</containerId>
      <downloadUrl>…orion2.0.5.zip</downloadUrl>
[…]
```

Profile that is always active

Environment-dependent builds (2/2)

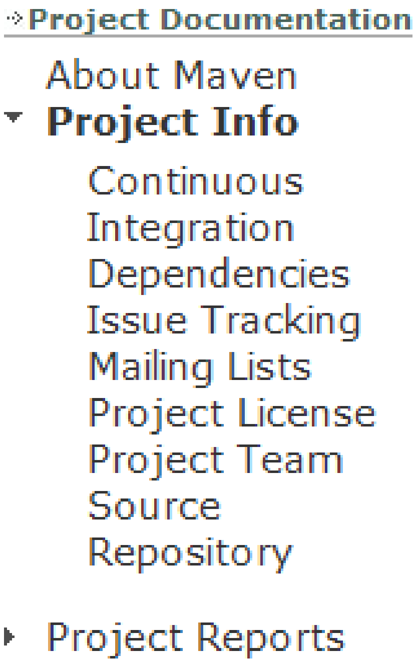
- Different activation conditions
  - JDK version, OS, property defined, existence of file or directory
- Profiles can also modify plugin configurations and other POM elements
  - Merged with the main pom.xml content
- Profiles can be selected on the command line:

```
mvn –P orion2x,resin3x install
```

Site and reports (1/4)

- Lots of reports
  - Project information (mailing lists, SCM, dependencies, etc)
  - PMD, Checkstyle, Javadoc, etc

```
<reporting>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-javadoc-plugin</artifactId>
    </plugin>
    […]
  </plugins>
</reporting>
```



Site and reports (2/4)

- Accepts several input formats
  - Almost Plain Text (Wiki like)
  - Xdoc (Maven 1.0 compatible)
  - FAQ (Maven 1.0 compatible)
  - Docbook

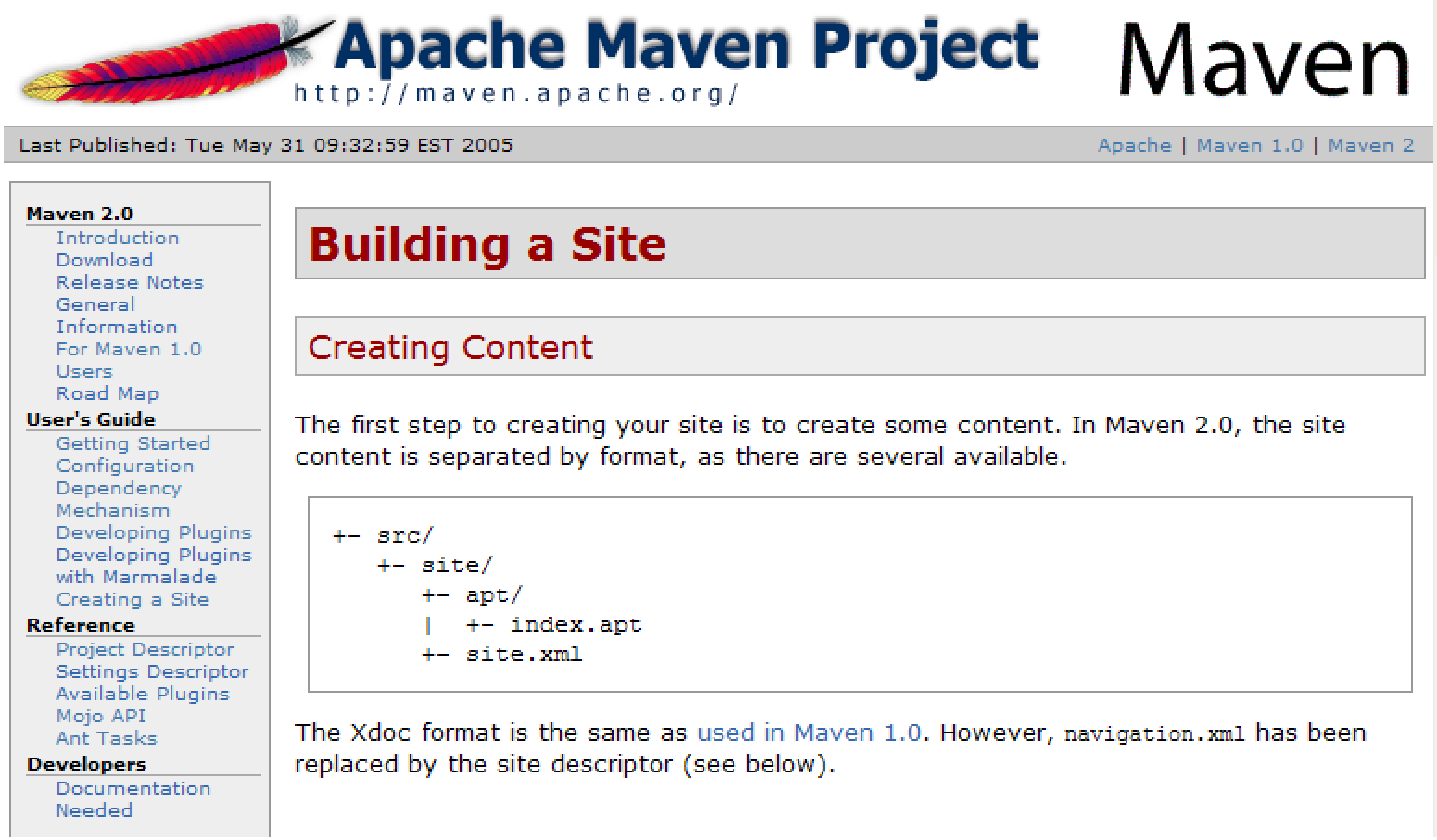
```
mvn site
```

```
+- src/
   +- site/
      +- apt/
      |  +- index.apt
      |
      +- xdoc/
      |  +- other.xml
      |
      +- fml/
      |  +- general.fml
      |  +- faq.fml
      |
      +- site.xml
```

Site and reports (3/4)

```
 ------
 Generating a Site
 ------
 Apache Maven Team
 ------
 13 May 2005
 ------

Building a Site

* Creating Content

  The first step to creating your site is to create some content. In
  Maven 2.0, the site content is separated by format, as there are several
  available.

-------------------
+- src/
   +- site/
      +- apt/
      |  +- index.apt
      +- site.xml
--------------------

  The Xdoc format is the same as
  {{{http://maven.apache.org/using/site.html} used in Maven 1.0}}.
  However, <<<navigation.xml>>> has been replaced by the site descriptor
  (see below).
```

Site and reports (3/4)



Last Published: Tue May 31 09:32:59 EST 2005

Apache | Maven 1.0 | Maven 2

Maven 2.0
- Introduction
- Download
- Release Notes
- General Information
- For Maven 1.0 Users
- Road Map

User's Guide
- Getting Started
- Configuration
- Dependency Mechanism
- Developing Plugins
- Developing Plugins with Marmalade
- Creating a Site

Reference
- Project Descriptor
- Settings Descriptor
- Available Plugins
- Mojo API
- Ant Tasks

Developers
- Documentation Needed

Building a Site

Creating Content

The first step to creating your site is to create some content. In Maven 2.0, the site content is separated by format, as there are several available.

```
+- src/
   +- site/
      +- apt/
      |  +- index.apt
      +- site.xml
```

The Xdoc format is the same as used in Maven 1.0. However, `navigation.xml` has been replaced by the site descriptor (see below).

Maven 2 Plugins (1/2)

- Antlr
- Ant
- AntRun
- AspectJ
- Assembly
- Assembly-report
- Cargo
- Castor
- Changelog
- Changes
- Commons-attributes
- Checkstyle
- Clean
- Clover
- Csharp
- Cobertura
- Compiler
- Deploy
- Ear
- Eclipse
- Ejb
- Ejb3
- Exec
- Groovy
- Help
- Hibernate2
- Idea
- Install
- Issue
- It
- Jalopy
- Jar
- Javacc
- Javadoc
- Javancss
- Jboss
- Jcoverage Jdepend
- Jdiff
- Jelly
- Jetty
- Jpox
- Jspc
- Jxr
- MAnt
- Native
- One
- Par
- Plugin
- Pmd
- Project-info-reports
- Rar
- Release
- Repository
- Resources
- Repository
- Sablecc
- Site
- Slimdog
- Source
- Surefire
- Surefire-report
- Taglist
- Tomcat
- Verifier
- Xslt
- War
- Wsdl2java
- Xdoclet
- Xmlbeans

Status: docs.codehaus.org/display/MAVEN/Maven+Plugin+Matrix

Maven 2 Plugins (2/2)

- Plugins are downloaded on demand
  - First time they are used
- Updates downloaded automatically
  - Opt-in notification if newer plugin found

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <configuration>
        <source>1.5</source>
        <target>1.5</target>
      </configuration>
    </plugin>
  </plugins>
</build>
```

Q&A