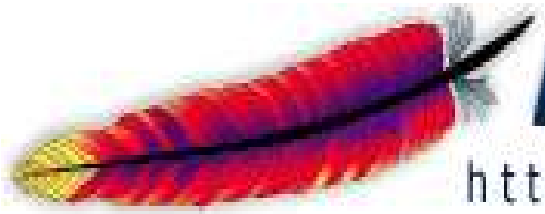
Maven 2
&
Continuum

by
Trygve Laugstøl <trygvis@apache.org>



Apache Maven Project

http://maven.apache.org/

Agenda

- About Maven
- Maven 2 Highlights
- Changes
  - ✓ The POM
  - ✓ Project layout
  - ✓ Plugin architecture
- Continuum

About Maven

- It's a different kind of build tool than Make and Ant
- Plugins are used to build the project
- Each project results in a single artifact
  - ✓ Typically a JAR or a WAR
- Artifacts is installed in a repository
  - ✓ User repository: `~/.maven/repository`
  - ✓ Company repository: `http://intra/repository`
  - ✓ Central repository: `http://ibiblio.org/maven`

About Maven

- Declarative configuration
  - ✓ The POM
  - ✓ `project.xml`
  - ✓ Dependencies
- Documentation
  - ✓ HTML
  - ✓ PDF
- Pushes for standardization
  - ✓ Source layouts
  - ✓ Reports

About Maven – Example POM

```
<project>
  <groupId>plexus</groupId>
  <artifactId>plexus-artifact-container</aritfactId>
  <version>1.0-SNAPSHOT</version>
  <dependencies>
    <dependency>
      <groupId>plexus</groupId>
      <artifactId>plexus</artifactId>
      <version>1.0-SNAPSHOT</version>
    </dependency>
  </dependencies>
  .
  .
```

About Maven – Example POM

```
  .
  .
<build>
  <sourceDirectory>src/main/java</sourceDirectory>
  <unitTestSourceDirectory>src/test/java</unitSourceTestDirectory>
  <resources>
   <resource>
    <directory>src/main/resources</directory>
    <includes>
     <include>**/*.png</include>
    <includes>
   </resource>
  </resources>
 </build>
</project>
```

Maven 2 - Highlights

- Status
  - ✓ Some experimental features
- Changes
  - ✓ Enhanced project definition tags
  - ✓ More strict rules
  - ✓ Transitive dependencies
- Generic reusable libraries
  - ✓ Maven Wagon
  - ✓ Maven Scm
  - ✓ Maven Artifact

Changes – The POM

- The extend tag is replaced with a more declarative parent tag
  - ✓ Old:

```
<extend>../project.xml</extend>
```

  - ✓ New:

```
<parent>
  <groupId>continuum</groupId>
  <artfiactId>continuum-core</artifactId>
  <version>1.0-SNAPSHOT</version>
</parent>
```

- Makes it easier to find parent POMs

Changes – The POM

- All POMs extend the super POM unless explicitly set
  - ✓ Follows the inheritance model in Java where all objects inheriting from Object
  - ✓ Contains default values for most projects
    - Main sources is in `src/main/java`
    - Unit test sources is in `src/test/java`
    - Main resources is in `src/main/resources`
    - Unit test resources is in `src/test/resources`
  - ✓ Uses the new guidelines for project layout
  - ✓ Reduce the size of the POM

Changes – The POM

- Issue Management
  - ✓ Old

```
<issueTrackingUrl>
  http://host/?BrowseProject.jspa&id=10030&bugId=
</issueTrackingUrl>
```

  - ✓ New

```
<issueManagement>
  <system>jira</system>
  <url>http://jira.codehaus.org/</url>
</issueManagement>
```

- Source Control Management
  - ✓ `<repository>` has been renamed to `<scm>`

Changes – The POM

- Removed
  - ✓ `<versions>`
  - ✓ `<branches>`
    - Will be collected from a set of POMs
- Added
  - ✓ Continous integration info

```
<ciManagement>
  <nagEmailAddress>dev@foo.org</nagEmailAddress>
  <system>continuum</system>
  <url>http://continuum.codehaus.org/builds</url>
</ciManagement>
```

Changes – The POM

- Pre and post goals are moved from `maven.xml` to the POM:
  - ✓ Old

```
<project>
 <goal
  name=”company:install”
  prereqs=”jar:jar,company:post-process-jar,jar:install”/>
</project>
```

  - ✓ New

```
<preGoals>
 <preGoal>
  <name>company:install</name>
  <attain>jar:jar,company:process-jar,jar:install</attain>
 </preGoal>
</preGoals>
```

Changes – Project Layout

- The directory of the project must match the artifact id
- The parent POM must be in a parent directory
- The POM is renamed from `project.xml` to `pom.xml`
  - ✓ Both descriptors can coexist
  - ✓ Makes migration easier

Changes – Project Layout

- All project metadata is in the POM
  - ✓ Only one place to look for metadata
- No more:
  - ✓ `maven.xml`
  - ✓ `project.properties`
    - Moved inside the POM
    - Unfinished
  - ✓ `build.properties`

```
<plugins>
  <plugin>
    <configuraion/>
  </plugin>
</plugins>
```

- `$MAVEN_HOME/configuration.xml`
  - ✓ Installation settings
- `$HOME/.m2/configuration.xml`
  - ✓ Specifies user configuration

Changes – Project Layout

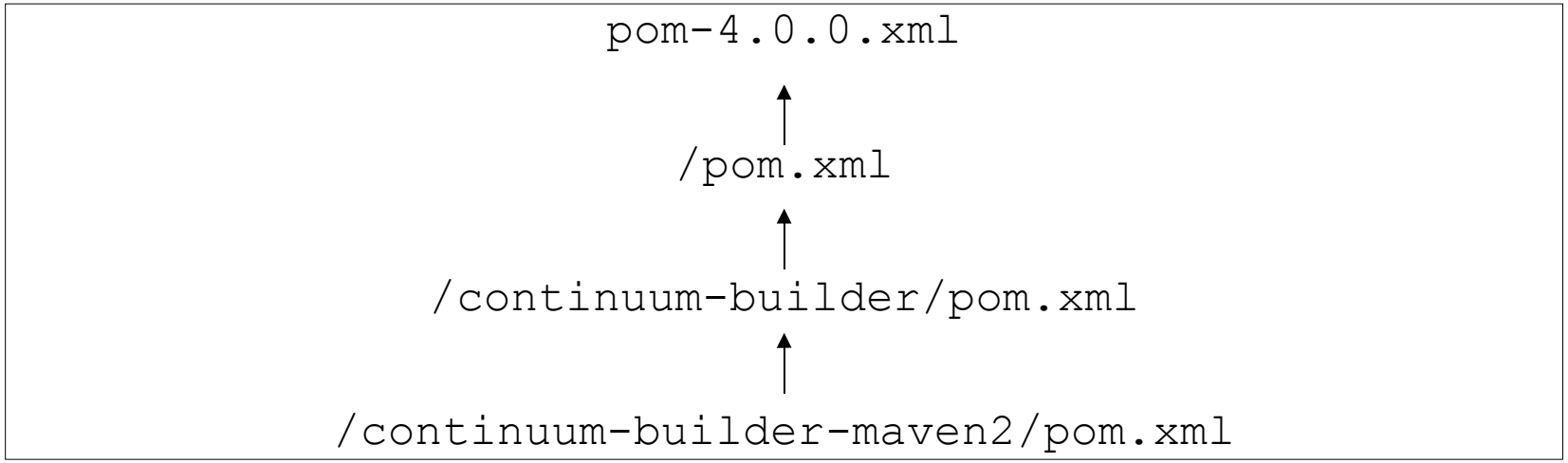
- Example layout – Continuum:

```
pom.xml
/continuum-api
  pom.xml
/continuum-core
  pom.xml
/continuum-builders
  pom.xml
  /continuum-builder-maven2
    pom.xml
```

- ✓ Given a fresh check out and empty repository "`$ m2 jar:install`" in /continuum-builder/continuum-builder-maven2 will build and install the artifact

Changes – Project Layout

- Example POM inheritance - Continuum



Changes – The Reactor

- The reactor is used to build multiple projects with a single command
  - ✓ Takes interproject dependencies in account so the project will be built in the correct order
- Built into the core

Changes – Repository Layout

- Proper support for
  - ✓ Primary artifacts
    - JARs, EJB implementations, WARs
  - ✓ Secondary artifacts
    - EJB client stubs, RMI stubs
    - Javadoc, source zips/tarballs
- Extensible
  - ✓ Plugins can define their own artifact types

Changes – Repository Layout

- Example artifact:
  - ✓ Group id: `org.apache.maven`
  - ✓ Artifact id: `maven-wagon`
  - ✓ Version: `1.0`
  - ✓ Type: `jar`

```
org/
    apache/
            maven/
                  maven-wagon/
                                1.0/
                                    maven-wagon-1.0.jar
                                    maven-wagon-1.0.jar.md5
                                    maven-wagon-1.0.jar.sources
                                    maven-wagon-1.0.jar.sources.md5
                                    maven-wagon-1.0.jar.javadocs
                                    maven-wagon-1.0.jar.javadocs.md5
```

Changes – Plugin Architecture

- No more Jelly!
- Transparently downloaded
  - ✓ Installed in the current session on the fly
- Maven 2 plugins are called Mojos
  - ✓ All core plugins are written in pure Java
  - ✓ All core plugins will be rewritten
- Mojos are completely reusable
  - ✓ Ant, Jelly and bean generators

Generic Reusable Libraries

- Maven Wagon
  - ✓ Library for transporting artifacts
- Maven SCM
  - ✓ Library for working with SCMs
    - CVS, Subversion, StarTeam, Perforce, Clearcase
- Maven Artifact
  - ✓ Library for working with artifacts.
    - Artifact layout
    - Deployment, installation, fetching
    - Conflict resolving
  - ✓ Usable for any tool using the Maven repository

Legacy Support

- Conversion tools for `project.xml` and `project.properties` files
- We'll make a survey to figure exactly what people want after the first alpha

Continuum

- Continuous integration server
- Currently supports
  - ✓ Maven 2
  - ✓ Maven 1
    - Some restrictions
- Will support
  - ✓ Ant
  - ✓ Shell scripts

Continuum - Features

- Will be able to build on
  - ✓ Schedule
  - ✓ Changes in the SCM
  - ✓ Changes in dependencies
- Manages inter-project dependencies
- Will read unit test reports from the build and store them

Resources

- Continuum
  - ✓ http://continuum.codehaus.org
- Mojo
  - ✓ http://mojo.codehaus.org
- Maven blog
  - ✓ http://www.mavenblogs.com
- Maven 2 CVS
  - ✓ http://cvs.apache.org/viewcvs.cgi/maven-components
  - ✓ CVS Check out:

```
:pserver:anoncvs@cvs.apache.org:/home/cvspublic/
cvs -d co maven-components
```