



Cabinet d'Architectes en Systèmes d'Information

# Apache Maven

**Mise en œuvre en entreprise**

Java User Group - Nantes

Jeudi 13 Novembre 2008

## ■ OCTO Technology

- Expert sénior
- aheritier AT octo DOT com
- <http://www.octo.com>



## ■ Maven

- Committer depuis 2004
- Membre du PMC  
(Project Management Committee) depuis 2005
- aheritier AT apache DOT org
- <http://maven.apache.org>



## ■ OCTO Technology

- Expert
- ojacob AT octo DOT com
- <http://www.octo.com>

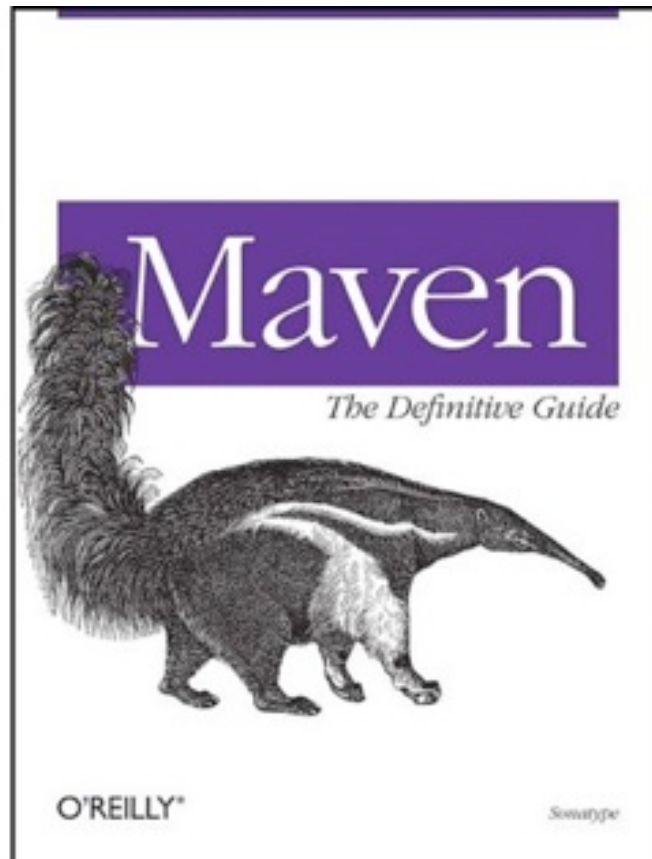


## ■ Maven

- Utilisateur depuis 2003
- Industrialisation des builds chez différents clients
- Quelques patches fournis pour bug fix ou évolutions sur les plugins
- <http://maven.apache.org>



<http://tinyurl.com/6r5v8p>



<http://tinyurl.com/5uv6lg>

**Java Productivity Primer**  
Twelve guidelines to boost your productivity  
with a software factory

October 2008

**Creative Commons**  
**Contrat Paternité**  
**Pas d'Utilisation Commerciale**  
**Partage des Conditions Initiales à l'Identique**  
**2.0 France**



<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

- Pourquoi Maven ?
- Maven, les concepts
- Maven, ses apports en entreprise
- Comment faire échouer son projet avec Maven ?
- Démonstration

Apache Maven

# POURQUOI MAVEN ?

# Construction d'un war en 2002 (1/2)

- Construire un war
  - Utilisation d'Eclipse limitée
    - En ces temps reculés où WTP n'existait pas (uniquement dans la version payante d'IBM), eclipse ne permettait pas d'exporter des Wars
  - Gestion manuelle
    - Modifier les fichiers de paramétrage
    - Exporter les différents jar
    - Copier les dépendances (et nos jars), dans un répertoire lib
    - Faire un zip que l'on renomme en war
    - Tagguer l'ensemble des sources dans le répertoire de sources (CVS)
    - Envoi du fichier par FTP sur le serveur d'intégration
    - Se connecter à la console d'administration du serveur et déployer l'application



# Construction d'un war en 2002 (2/2)

## ■ Combien de temps ça prend ?

- Quand tout va bien :
  - 15 minutes
- Quand il y a des problèmes :
  - ½ journée



## ■ Un seul problème, Y'a toujours des problèmes

- Erreur dans la configuration
- Oubli d'une dépendance
- Oubli d'un fichier
- Correction de dernière minute qui introduit une régression...
- Autres

# Une première réponse : ANT

- Ecriture d'un script
  - Permet d'automatiser le processus
  - Durée du processus réduite de moitié
  - Le processus ne monopolise personne
    - On le lance et on passe à autre chose



# Les limites de ANT



- Ecrire le script, c'est long
- Modifier un script, c'est très long
- Au final, le gain de temps n'est pas évident
  - Mais c'est quand même plus amusant
  - Il est possible de réutiliser le script !

# La réutilisation de scripts ANT

- Les scripts ne sont pas directement réutilisables
  - Structure de projets différents
  - Besoins différents
- Encore du temps perdu
  - Modification du script
  - Réécriture pour le rendre plus générique
- → Un nouveau métier s'est créé : scripteur ANT



# Quelques exemples

[http://anonsvn.jboss.org/repos/seam/branches/community/Seam\\_2\\_0](http://anonsvn.jboss.org/repos/seam/branches/community/Seam_2_0)

<http://junit.cvs.sourceforge.net/viewvc/junit/junit/build.xml?view=markup>

<http://findbugs.googlecode.com/svn/trunk/findbugs/build.xml>

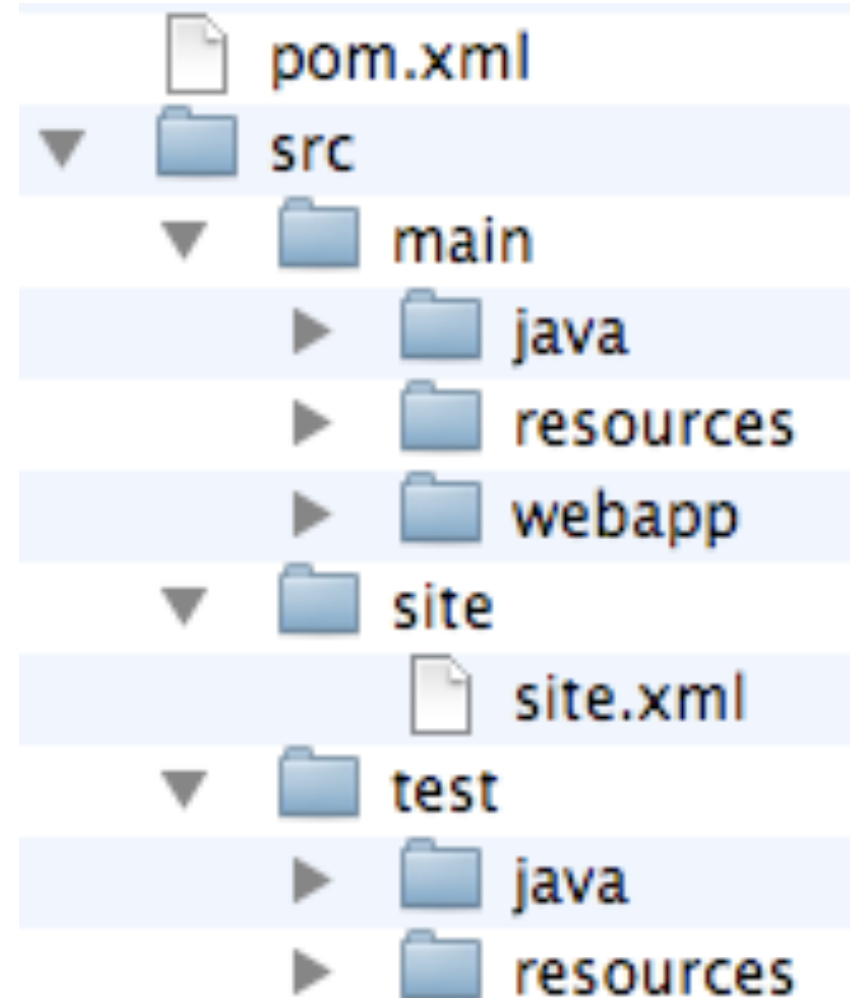


Apache Maven

# LES CONCEPTS

- **Convention Over Configuration**
  - On ne scripte plus, on décrit son projet (POM)
- **Architecture extensible**
  - Des plugins réutilisables et configurables évitent la multiplication des scripts de build
- **Gestion des dépendances du projet (versions, partage, ..)**
  - Référentiels d'artefacts
  - Dépendances déclaratives
  - Dépendances transitives
- **Amélioration de la qualité des livrables**
  - Exécution automatique des tests unitaires
  - Contrôles (conventions de nommage, codage, ...)
  - Rapports et documentations

- 1 projet Maven = 1 artéfact (pom, jar, war, ear, ...)
- 1 organisation des répertoires prédéfinie
- 1 descripteur standardisé (POM)
- Des cycles de construction standardisés





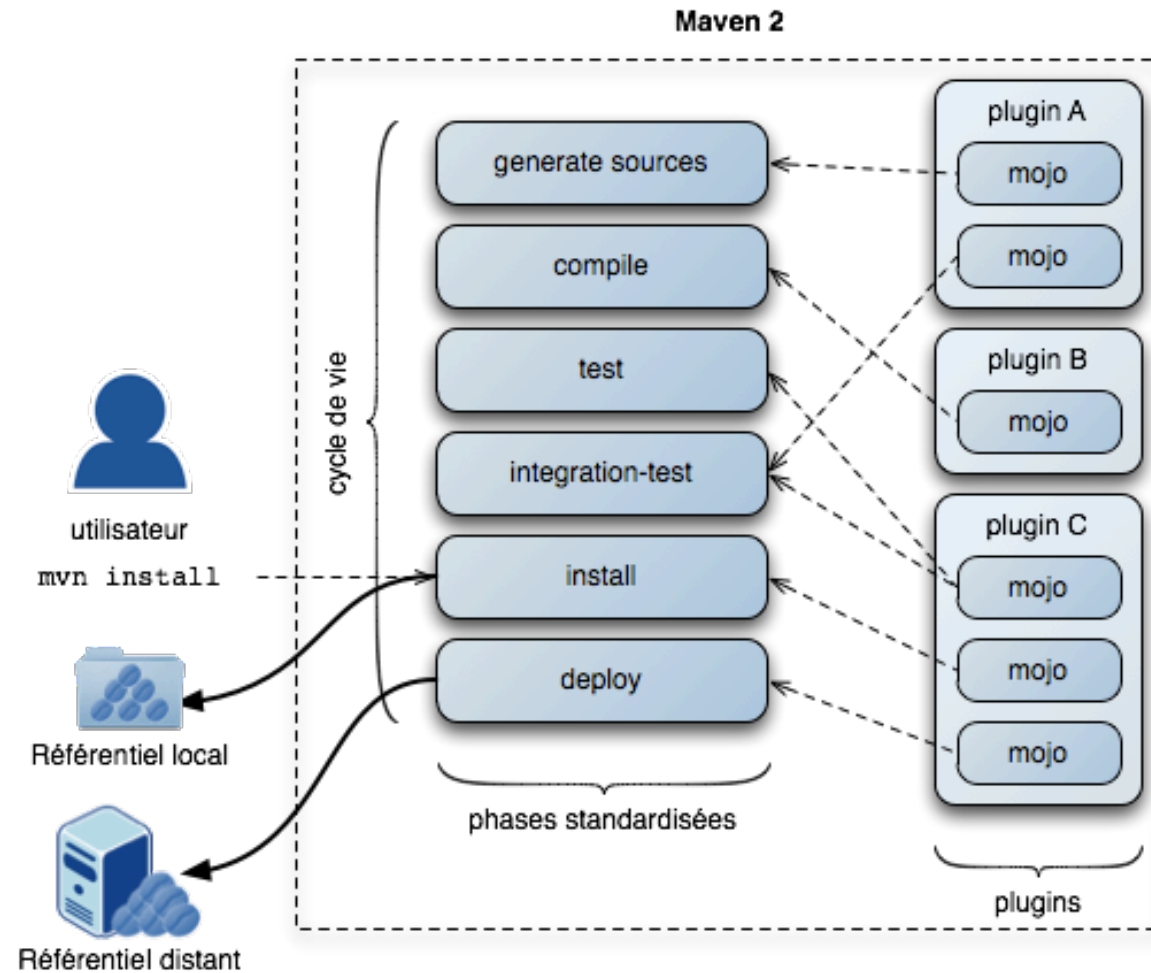
## ■ Un fichier XML (pom.xml)

## ■ Décrivant

- Identification du projet
- Version du projet
- Description du projet
- Liste des développeurs
- Dépendances
- ...

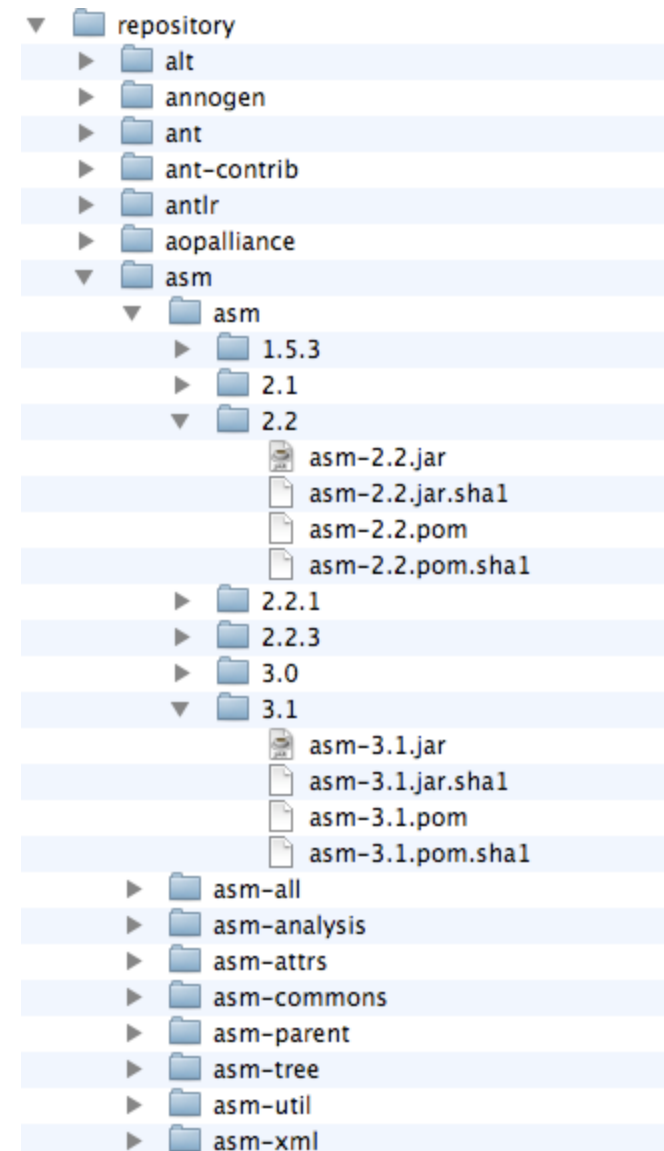
```
<?xml version="1.0" encoding="UTF-8"?>
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.octo</groupId>
  <artifactId>webapp-sample</artifactId>
  <version>1.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <name>Simple webapp</name>
  <inceptionYear>2007</inceptionYear>
  <dependencies>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-struts</artifactId>
      <version>2.0.2</version>
    </dependency>
    ...
  </dependencies>
</project>
```

# Cycle de vie et plugins

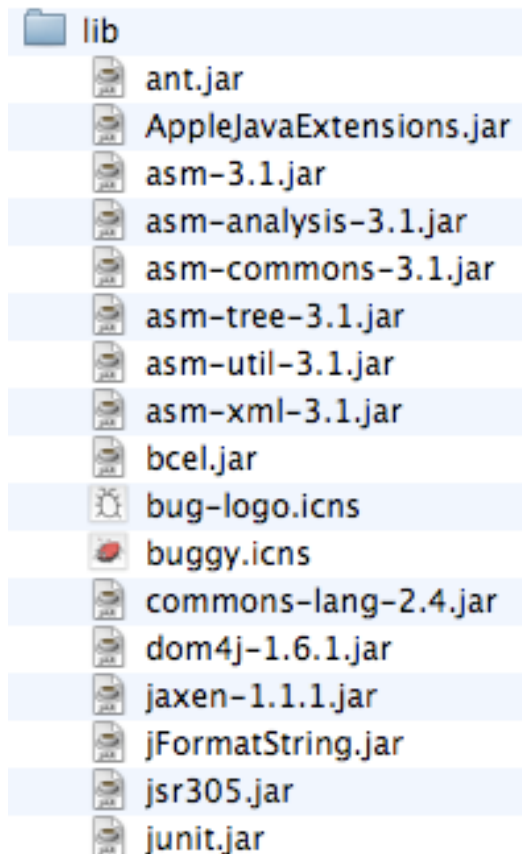


# Les référentiels d'artéfacts

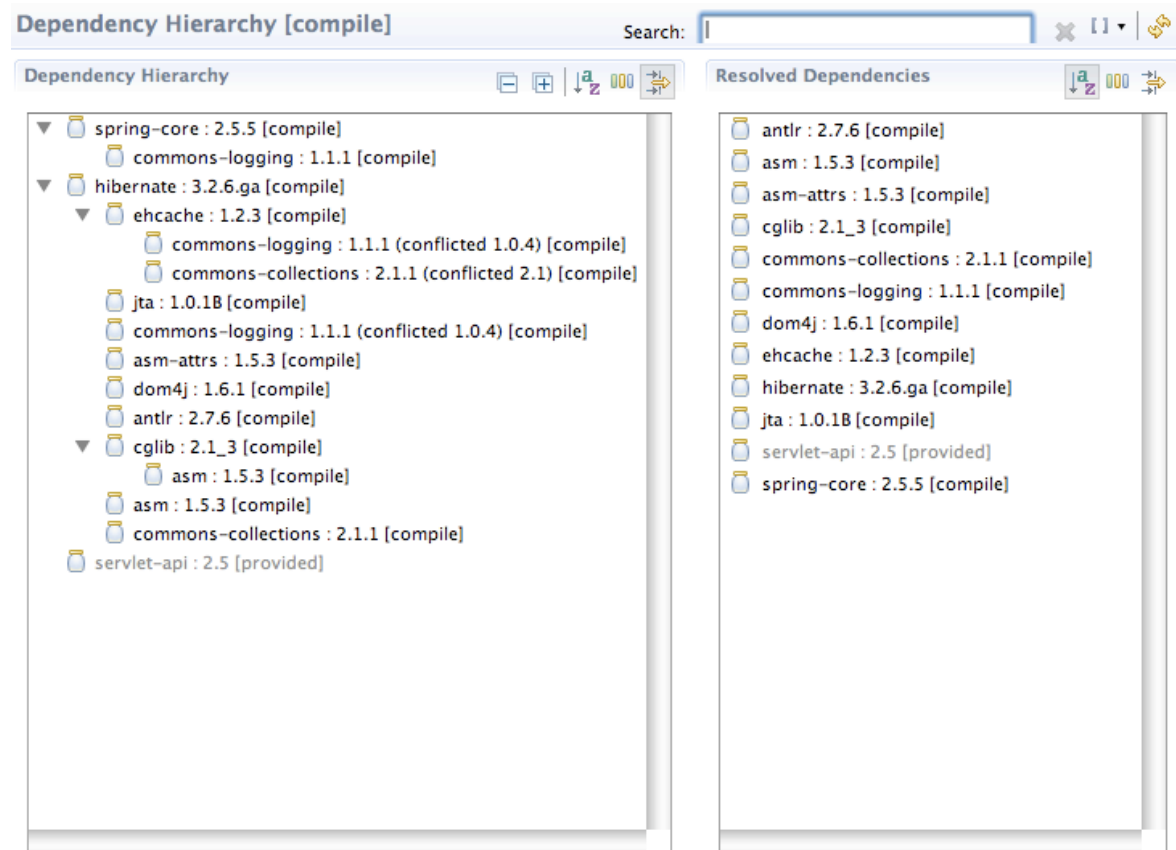
- Par défaut :
  - Un central
    - <http://repo1.maven.org/maven2>
    - Propose plusieurs dizaines de Go de librairies OSS
  - Un local
    - `${user.home}/.m2/repository`
    - Centralise tous les artéfacts utilisés et créés par vos projets



## Sans Maven



## Avec Maven



## ■ Déclaratives

- groupId + artifactId + version (+ classifier)
- Type (packaging) : jar, war, pom, ear, ...

## ■ Transitives

- Lib A → Lib B
- Lib B → Lib C
- Alors Lib A → Lib C

## ■ Scopes d'utilisation

- Runtime : taglibs
- Provided : Servlet API, Driver SGBD, ...
- Compile : par défaut
- Test : Junit, TestNG, DbUnit, ...
- *System : non recommandé (emplacement physique sur disque)*

- Le snapshot est une version particulière d'un artéfact / projet
  - Indique qu'un artéfact est en cours de développement
  - Le numéro de version est suffixé par -SNAPSHOT
  - L'artéfact est écrasé à chaque nouvelle construction en local
  - L'artéfact est déployé avec un timestamp sur le repository central du projet
  
- Les versions fixes (releases)
  - Sans -SNAPSHOT dans leur numéro

## ■ Dépendances snapshot

- Politique de vérification des mises à jour
  - always
  - daily (par défaut)
  - interval:X (valeur personnalisée en minutes)
  - never
- Ne peuvent pas être utilisées lors d'une release
  - Maven recommande une version figée

## ■ Intervalle de versions

- De ... à ...
- Maven recherche automatiquement les nouvelles versions
- A utiliser avec précaution
  - Risque de non reproductibilité du build
  - Risque de dysfonctionnement des consommateurs

- Construit un projet et tous ses modules en résolvant l'ordre de leurs dépendances

- Projet 1

- Module A (dépend de B)
    - Module B
    - Module C (dépend de A)

- ➔ Construire B

- puis A
    - puis C

- Les modules sont déclarés explicitement

pom.xml de projet 1

```
<modules>
  <module>moduleA</module>
  <module>moduleC</module>
  <module>moduleB</module>
</modules>
```

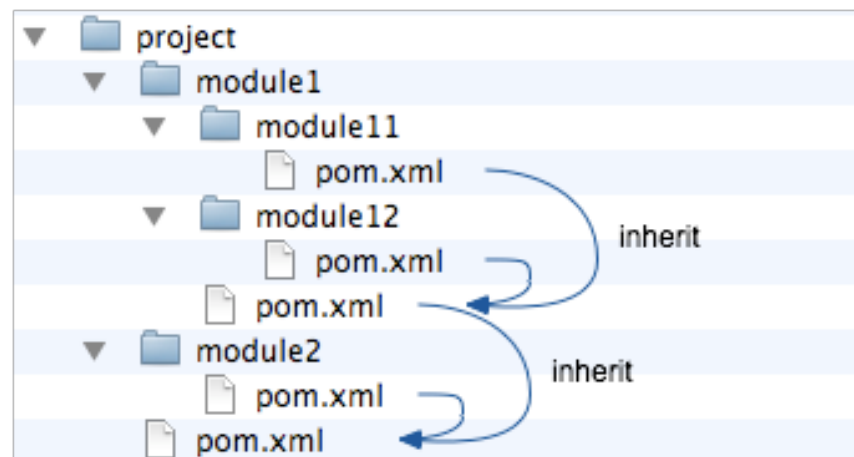


- Factorisation de paramètres entre projets/modules
- L'organisation des sous modules et de l'héritage doit être technique (ejbs, wars, jars) et non pas fonctionnelle (serviceA, serviceB, ...)
- Projet1
  - Jars
    - Jar1
    - Jar2
  - Wars
    - War1
    - ...

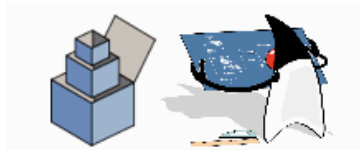
pom.xml de module Jar1

```
<parent>
  <groupId>X.Y.Z</groupId>
  <artifactId>jars</artifactId>
  <version>1.0-SNAPSHOT</version>
</parent>
```

- Lorsque le parent sert aussi de reactor
  - A condition que nom du module = son artifactId
  - Evite la re-définition de certains éléments du pom
    - Url du site
    - Déploiement du site
    - Informations SCMs
  - Est recommandé pour éviter quelques bugs dans les plugins



- Génération d'un site documentaire à partir de documents dans différents formats :
  - Xdocs (xml)
  - Apt (wiki)
  - Docbook
  - ....
- Génération de rapports sur le projet
  - Descriptif
  - Informations sur les dépendances
  - Documentation automatique des plugins maven
  - Javadoc
  - Xref
  - Tests unitaires, couverture, ...



Last Published: 08 Oct 2008 16:35:36 | Version: 1.1-SNAPSHOT

## Project Documentation

- ▼ Project Information
  - About
  - Continuous Integration
  - Dependencies**
  - Dependency
  - Convergence
  - Issue Tracking
  - Mailing Lists
  - Project License
  - Project Summary
  - Project Team
  - Source Repository

## Project Reports



## Project Dependencies

### compile

The following is a list of compile dependencies for this project. These dependencies are required to compile and run the application:

GroupId	ArtifactId	Version	Classifier	Type	Optional
commons-dbcp	commons-dbcp	1.2.1	-	jar	
displaytag	displaytag	1.1	-	jar	
hsqldb	hsqldb	1.8.0.7	-	jar	
log4j	log4j	1.2.14	-	jar	
org.ajaxtags	ajaxtags	1.2-beta3	-	jar	
org.apache.geronimo.specs	geronimo-jta_1.0.1B_spec	1.1	-	jar	
org.hibernate	hibernate-annotations	3.2.0.ga	-	jar	
org.springframework	spring-hibernate3	2.0	-	jar	
org.springframework	spring-struts	2.0.2	-	jar	
struts	struts	1.2.9	-	jar	

### test

The following is a list of test dependencies for this project. These dependencies are only required to compile and run unit tests for the application:

- Types de contrôles / reporting
  - Résultat de l'exécution des tests : surefire
  - Couverture de tests : clover, cobertura
  - Qualité du code : findbugs
  - Conformité aux standards : checkstyle
  - Patterns / Antipatterns : pmd

## ■ Contrôle - Pro-actif

- Le build échoue si mon code ne respecte pas certaines règles
- Les problèmes ne sont pas dans le gestionnaire de sources puisque je ne commit jamais un code qui ne build pas 😊

## ■ Reporting - Réactif

- Je corrige les problèmes une fois étudié le rapport
- Les problèmes sont dans le gestionnaire de sources


## Test Cases

[\[Summary\]](#) [\[Package List\]](#) [\[Test Cases\]](#)



### → ContactsServicesTest

	checkAddressSaveGenerateId	0.469
	checkAddressSave	0.469
	checkTwoAddressSavedHaveDifferentIds	0.469


### → TestContactService

	testContactsService	0.454
---	---------------------	-------

### → CountriesServletHttpUnitTest

	setupServletRunner	4.859
	callServletDoesntFireMalformedException	0.718
	callServletDoesntFireException	0.359
	callServletWithParameters	0.86

### → CreateContactActionStrutsTestCaseTest

	testAddressCreator	0.454
---	--------------------	-------

## ■ Profiles

- (Re-)définition de certaines parties du POM en fonction de profils de build

## ■ Properties

- Définition de propriétés réutilisables dans le POM

## ■ Assemblies

- Définition de packaging personnalisés

## ■ Archetypes

- Définition de squelettes de projets personnalisés

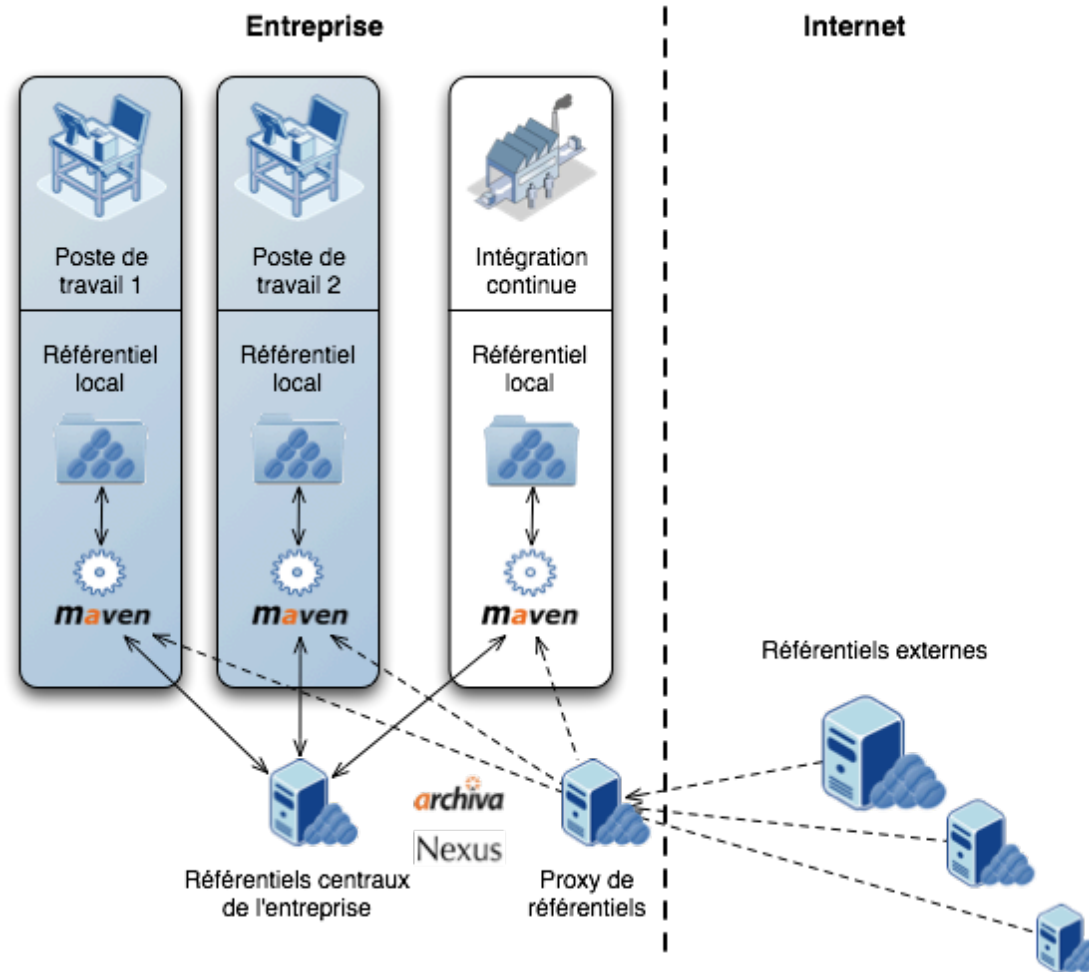


Apache Maven

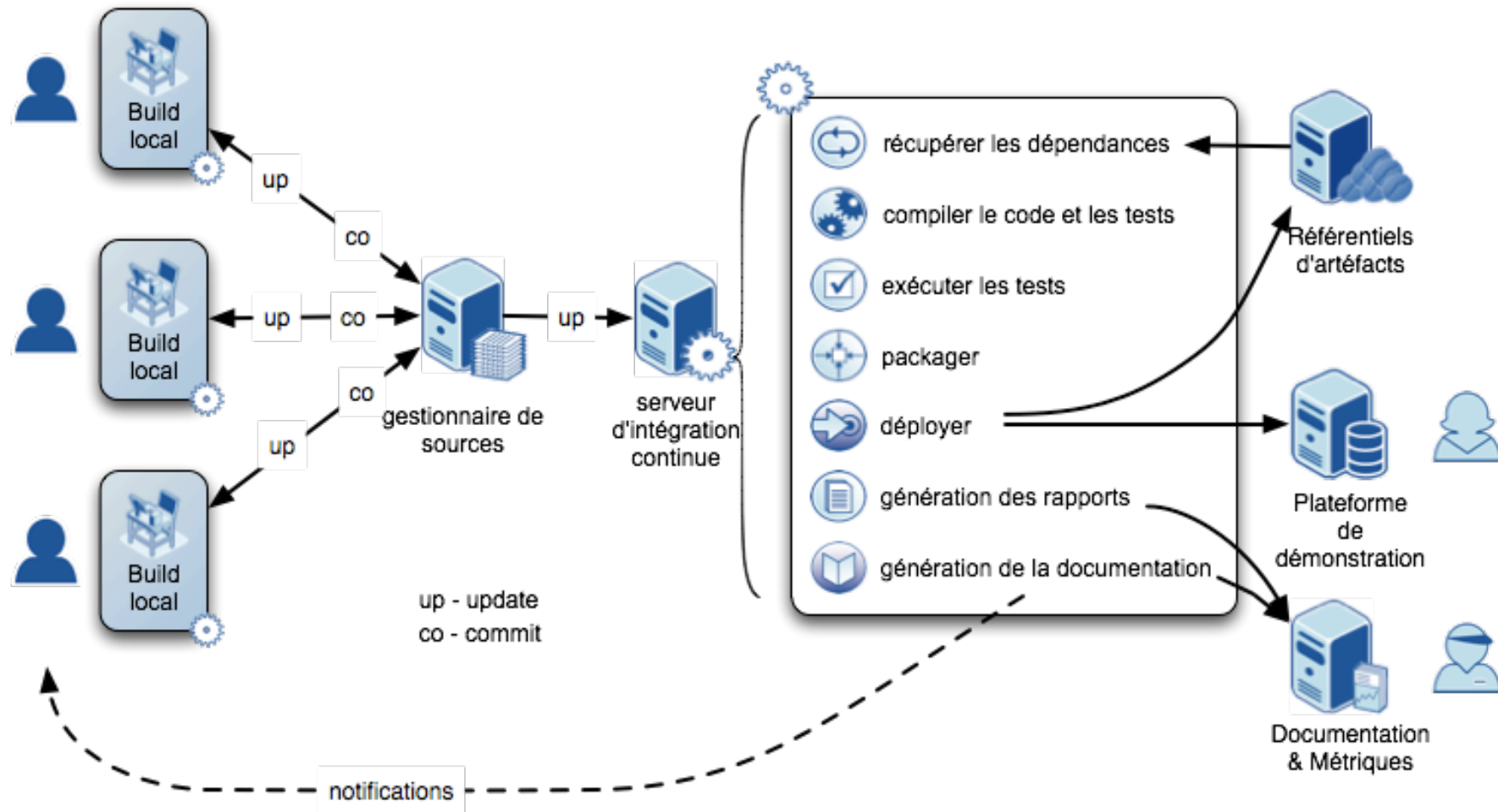
# SES APPORTS EN ENTREPRISE

- Pour les équipes
  - Capitalisation des connaissances
  - Permet de facilement passer d'un projet ou module à un autre
- Pour les cellules transverses
  - Facilite le support

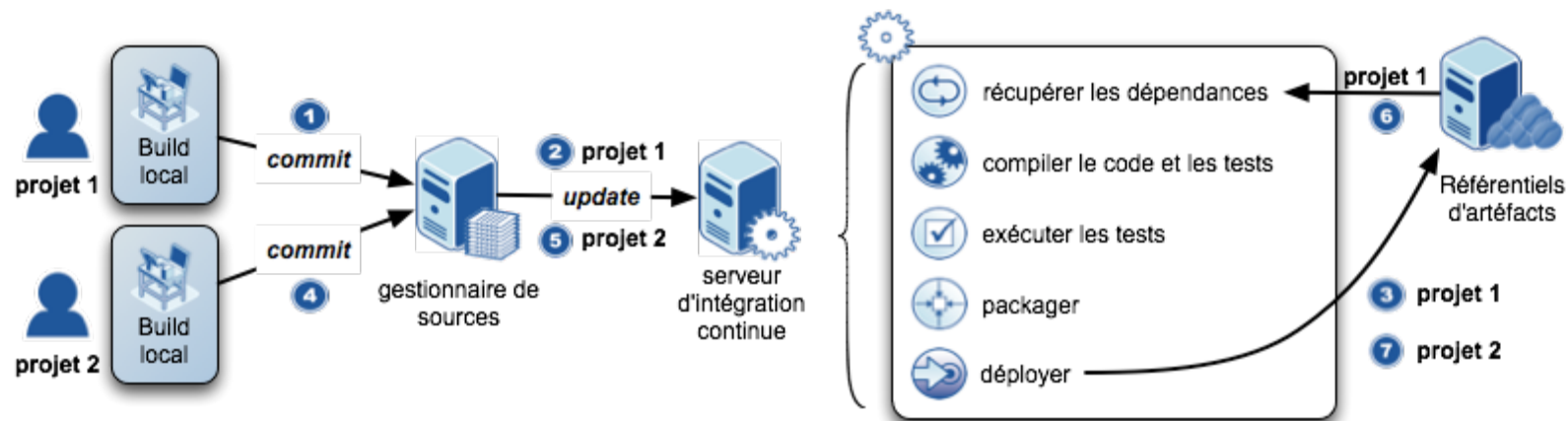
# Une véritable gestion des binaires



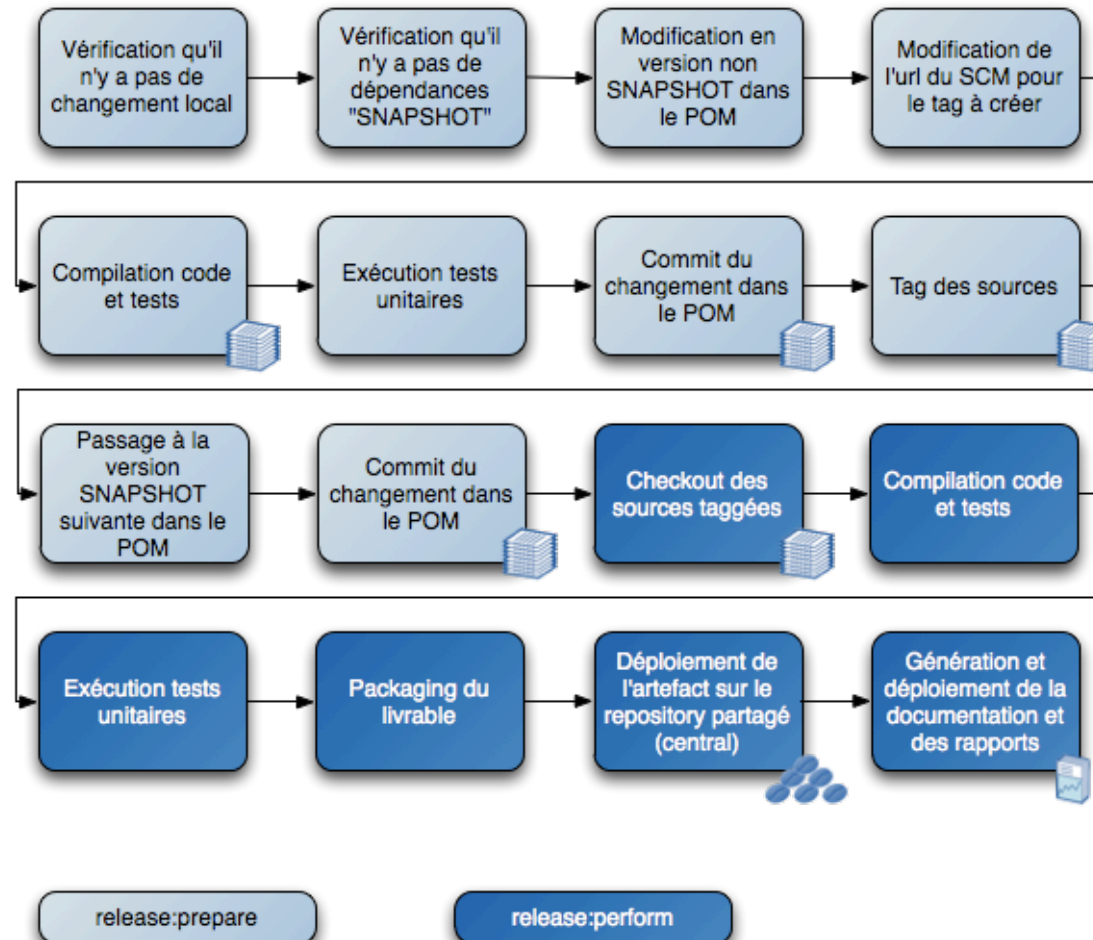
# L'intégration continue facilitée



- De configuration
  - Héritage de POMs
- De recommandations sur les dépendances
  - Import de dependencyManagement
- De traitements
  - Plugins
- De librairies



# Le processus de release automatisé



Apache Maven

# COMMENT FAIRE ÉCHOUER SON PROJET AVEC MAVEN ?

# 10 bonnes pratiques pour faire échouer son projet

1. Ne pas utiliser les conventions
2. Mettre tout ce qui est possible de mettre dans le pom
3. Se rendre dépendant de l'environnement
4. Multiplier les niveaux d'héritage
5. Utiliser systématiquement `"-Dmaven.test.skip=true »`
6. Faire les releases à la main
7. S'échanger les jars par mail
8. Utilisation massive du plugin antrun
9. Confondre dependencies et dependencyManagement



# 10. Passer son temps dans la console

```

, parameter value [1], value class [java.lang.Long], SQL type 4>
2008-10-08 03:34:41.140 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
, parameter value [Wed Oct 08 03:34:41 CEST 2008], value class [java.util.Date], SQL type 93>
2008-10-08 03:34:41.140 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
, parameter value [null], value class [null], SQL type 93>
2008-10-08 03:34:41.140 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
, parameter value [STARTED], value class [java.lang.String], SQL type 12>
2008-10-08 03:34:41.140 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
, parameter value [0], value class [java.lang.Integer], SQL type 4>
2008-10-08 03:34:41.156 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
, parameter value [0], value class [java.lang.Integer], SQL type 4>
2008-10-08 03:34:41.156 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
0, parameter value [V], value class [java.lang.String], SQL type 1>
2008-10-08 03:34:41.156 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
1, parameter value [COMMITTABLE], value class [java.lang.String], SQL type 12>
2008-10-08 03:34:41.171 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
2, parameter value [1], value class [java.lang.String], SQL type 12>
2008-10-08 03:34:41.171 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
3, parameter value [0], value class [java.lang.Integer], SQL type 4>
2008-10-08 03:34:41.171 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
4, parameter value [0], value class [java.lang.Integer], SQL type 4>
2008-10-08 03:34:41.187 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
5, parameter value [0], value class [java.lang.Integer], SQL type 4>
2008-10-08 03:34:41.187 DEBUG [org.springframework.jdbc.core.JdbcTemplate] - <SQL update affected 1 rows>
2008-10-08 03:34:41.187 DEBUG [org.springframework.jdbc.datasource.DataSourceUtils] - <Returning JDBC Connection to DataSource>
2008-10-08 03:34:41.187 DEBUG [org.springframework.jdbc.core.JdbcTemplate] - <Executing prepared SQL update>
2008-10-08 03:34:41.187 DEBUG [org.springframework.jdbc.core.JdbcTemplate] - <Executing prepared SQL statement [UPDATE BATCH_STEP_EX
ec START_TIME = ?, END_TIME = ?, STATUS = ?, COMMIT_COUNT = ?, ITEM_COUNT = ?, CONTINUABLE = ?, EXIT_CODE = ?, EXIT_MESSAGE = ?, U
?, READ_SKIP_COUNT = ?, WRITE_SKIP_COUNT = ?, ROLLBACK_COUNT = ? where STEP_EXECUTION_ID = ? and VERSION = ?]>
2008-10-08 03:34:41.187 DEBUG [org.springframework.jdbc.datasource.DataSourceUtils] - <Fetching JDBC Connection from DataSource>
2008-10-08 03:34:41.203 DEBUG [org.springframework.jdbc.datasource.DriverManagerDataSource] - <Creating new JDBC DriverManager Conne
ction [jdbc:hsqldb:mem:testdb;sql.enforce_strict_size=true]>
2008-10-08 03:34:41.203 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
, parameter value [Wed Oct 08 03:34:41 CEST 2008], value class [java.util.Date], SQL type 93>
2008-10-08 03:34:41.203 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
, parameter value [null], value class [null], SQL type 93>
2008-10-08 03:34:41.203 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
, parameter value [STOPPED], value class [java.lang.String], SQL type 12>
2008-10-08 03:34:41.203 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
, parameter value [0], value class [java.lang.Integer], SQL type 4>
2008-10-08 03:34:41.203 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
, parameter value [0], value class [java.lang.Integer], SQL type 4>
2008-10-08 03:34:41.203 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
, parameter value [V], value class [java.lang.String], SQL type 1>
2008-10-08 03:34:41.218 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
, parameter value [COMMITTABLE], value class [java.lang.String], SQL type 12>
2008-10-08 03:34:41.218 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
, parameter value [1], value class [java.lang.String], SQL type 12>
2008-10-08 03:34:41.218 DEBUG [org.springframework.jdbc.core.StatementCreatorUtils] - <Setting SQL statement parameter value: column
, parameter value [0], value class [java.lang.Integer], SQL type 4>

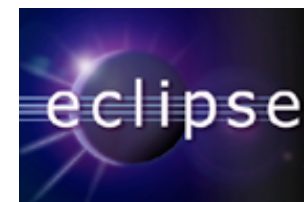
```



# Apache Maven

## DÉMONSTRATION

**maven**





Cabinet d'Architectes en Systèmes d'Information

## Questions ?

Merci pour votre attention

Merci aux sponsors



**NETAPSYS**  
ingénierie informatique