



Apache Maven Project

<http://maven.apache.org/>

L'outil de construction et de gestion de projet JAVA

.....

En route vers Maven

Michel VONGVILAY

4 novembre 2004

+Introduction

- La problématique
- Ant ou Maven
- Objectifs de Maven

- Concepts

- POM
- Repository
- Goal

- Cas pratiques

- Installation
- Générer un projet
- Compilation
- Site web du projet
- Test unitaire

- Plugins existants

- Code source
- Documentations
- Activités CVS
- Dépendances

Comment construire et maintenir les projets de développement Java ?

- La technique pékinoise
 - Trouver un ou deux étudiants ;-)
- Le bon vieux *Make*, toujours fidèle ...
 - ... à Unix. Windows il s'en fout.
- *Ant* autre, le plus populaire
 - Peu réutilisable, maintenance fastidieuse

+Introduction

La problématique

 Ant ou Maven

Objectifs de Maven

- Concepts

POM

Repository

Goal

- Cas pratiques

Installation

Générer un projet

Compilation

Site web du projet

Test unitaire

- Plugins existants

Code source

Documentations

Activités CVS

Dépendances

Concurrents !?

- Tous 2 Développés par Jakarta
- « Il n'y a rien que Maven fasse que Ant ne puisse faire »

Alors pourquoi Maven ?

- Plus adapté aux besoins d'aujourd'hui
- Plus flexible, Plus simple
- Fonctionnalités réutilisables

Analogie : C et Java

- Réduit la complexité
- Maven suffit, mais seul Ant peut faire certaines tâches
- Maven outil de construction de plus haut niveau



+Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

■ Comparaison Maven et Ant

	Maven	Ant
Fonction	Goal (création + prédéfinis)	Target
Fichier build	Project.xml (POM) Maven.xml	Build.xml
Propriétés	La dernière définition l'emporte. (Permet l'héritage)	La première définition l'emporte.
Règle et langage	-Règle de construction plus dynamique -Jelly script, inclus Ant Task	-Règle de construction plus ou moins statique (à moins d'utiliser le tag <script>)
Pré requis des fonctions	-Attribut <i>prerequisite</i> du tag <goal>	Attribut <i>depends</i> du tag <target>
Extensibilité des règles de construction	-Les Build goals sont extensibles en définissant <preGoal> et <postGoal>	Elles ne sont pas vraiment extensible mais elle peuvent être simuler en utilisant les <script>



+Introduction

La problématique
Ant ou Maven

 Objectifs de Maven

- Concepts

POM
Repository
Goal

- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

Envie d'Ant'core mieux !? Voilà Maven

- Un peu d'histoire
 - Initialement pour Turbine
 - Devenu une application *Open Source*
- Principaux objectifs:
 - Faciliter le processus de construction
 - Faciliter la gestion du projet
 - Fournir une vision cohérente et globale
- Basé sur le concept POM

- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

+ Concepts

POM
Repository
Goal

- Cas pratiques

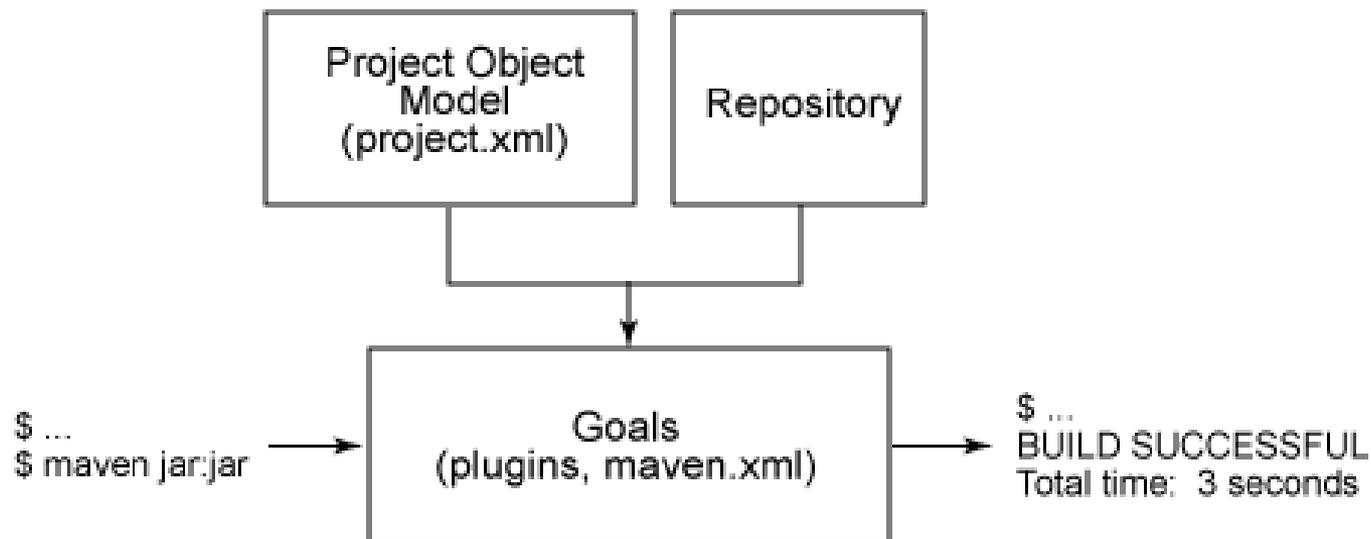
Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

■ **Trois Principaux éléments**

- POM
- Repository
- Goal



- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

+ Concepts

 POM
Repository
Goal

- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

Project Object Model (POM)

- Entité principale : Maven est orienté Projet
 - Fichier *project.xml*, situé à la racine du projet
 - Tout est contrôlé depuis le POM
- Description du projet en XML

Trois principales sections

- Partie gestion de projet
 - Organisation, liste des développeur, liste de diffusion, CVS
 - Localisation du code source et l'url du système détection de bug.
- Partie dépendances du projet
 - Information sur les JAR dépendant du projet
 - Permet de créer le classpath du projet
- Partie construction et documentation
 - Répertoire du code source, des tests unitaires, et les rapports à générer



- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

+ Concepts

 POM
Repository
Goal

- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- l'element racine d'un fichier projet -->
- <project>
  <!-- version du POM. Ce tag n'est pas encore exploité. -->
  <pomVersion>3</pomVersion>
  <!-- Unique id du projet. Il est aussi utilisé pr générer des noms de fichier -->
  <!-- i.e, le fichier JAR est nommé : <id>-<version>.jar -->
  <id>chronos</id>
  <!-- Le nom court du projet -->
  <name>Chronos vnR</name>
  <!-- groupe id du projet. -->
  <!-- Si present, l'id est utilisé comme nom de dossier ds le repository -->
  <!-- pour regrouper tous les jars du projet -->
  <groupId>myGroup</groupId>
  <!-- numero de version du projet. (Pas de norme à suivre) -->
  <currentVersion>0.0.1</currentVersion>
  <!-- ***** -->
  <!-- Partie gestion de projet -->
  <!-- ***** -->
  <!-- Info sur l'organisation -->
+ <organization>
  <!-- Liste des developpeurs -->
+ <developers>
  <!-- Liste de diffusion -->
+ <mailingLists>
  <!-- ***** -->
  <!-- Partie dépendances du projet -->
  <!-- ***** -->
  <dependencies />
  <!-- ***** -->
  <!-- Partie build et documentation du projet -->
  <!-- ***** -->
  <build />
</project>
```

- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

+ Concepts

POM
Repository
Goal

- Cas pratiques

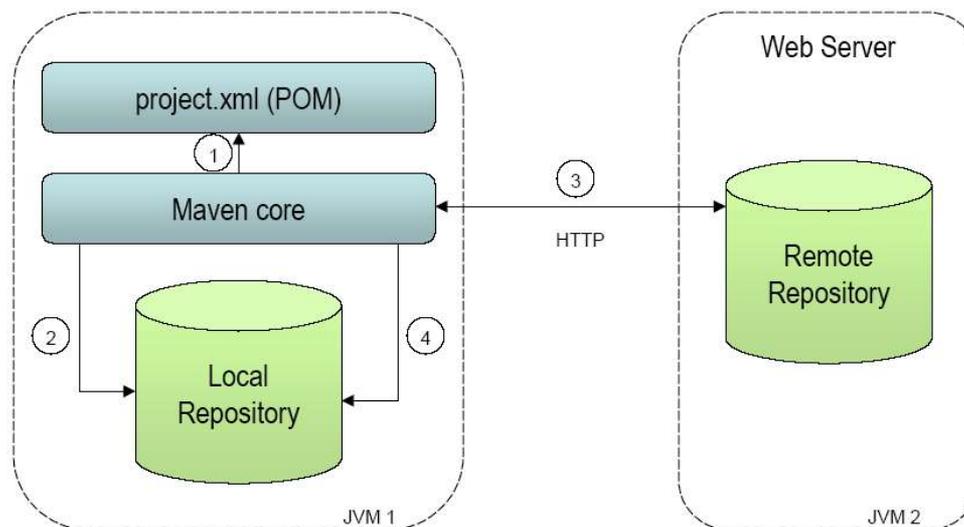
Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

■ Un autre élément important de Maven : Le repository

- Permet de mieux gérer les dépendances
- Permet le partage des JARs entre projet



- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

+ Concepts

POM
Repository
Goal

- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

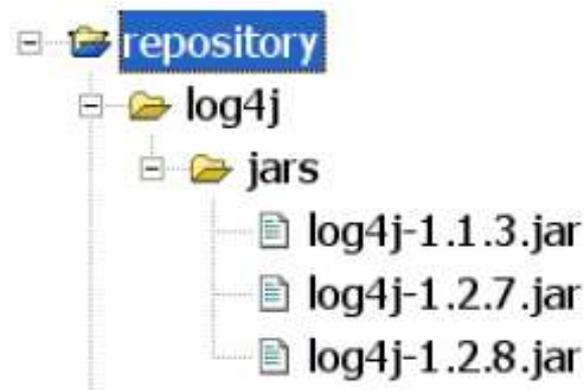
- Repository normalisé

```

<project>
[... ]
<dependencies>
  <dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.8</version>
    <type>jar</type>
  </dependency >
</dependencies >
[... ]
</project>

```

$\left. \begin{array}{l} \text{\<groupId>log4j</groupId>} \\ \text{\<artifactId>log4j</artifactId>} \\ \text{\<version>1.2.8</version>} \\ \text{\<type>jar</type>} \end{array} \right\} \begin{array}{l} \$MAVEN_REPO/ \\ \text{\<groupId>/<type>s/<artifactId>-<version>.<type>} \end{array}$



- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

+ Concepts

POM
Repository
 Goal

- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

-  **Équivalent au *Target* de *Ant***
-  ***Mais extensible et réutilisable***
-  ***Repose sur des plugins (jar)***

-  ***Utilisation en ligne de commande***
 - *Maven <plugin>:<goal>*
 - *Ex: “Maven java:compile”*
 - *“Maven pdf” équivalent à “Maven pdf:pdf”*
 - *Exception levée si paramètres incorrects*

-  ***Commande : maven -g***



- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

+ Concepts

POM
Repository
Goal

- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

■ *maven.xml* : personnaliser des goals

■ *2 possibilités* :

- *Extension des goals existants*
 - *<preGoal>* : règles à faire **avant** le goal
 - *<postGoal>* : règles à faire **après** le goal

Exemple de *<preGoal>*

```
<preGoal name="java:compile">  
  <attainGoal name="xdoclet:ejbdoclet" />  
</preGoal>
```

- *Création d'un goal*

- Jelly
- Ant task

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<project xmlns:j="jelly:core">  
  <goal name="Echo Dependencies">  
    <j:forEach var="dep" items="{pom.dependencies}">  
      <echo>${pom.getDependencyPath(dep.getId())}</echo>  
    </j:forEach>  
  </goal>  
</project>
```

■ *Partager les Goals* : créer plugin

- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

+ Cas pratiques

 Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

Télécharger puis décompresser

Définir les variables d'environnement :

- MAVEN_HOME et JAVA_HOME
- PATH : ajouter `${MAVEN_HOME}/bin`

Créer le repository local

- `${MAVEN_HOME}/bin/install_repo.sh $HOME/.maven/repository`

Vérification : Maven -v

- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

+ Cas pratiques

Installation
 Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

Plugin Genapp :

- Permet de créer rapidement un projet
- Plusieurs templates de projet prédéfinies :
 - Default – Projet simple Jar
 - Ejb - Projet simple Ejb
 - Web - Projet simple application web Structs
 - Complex – Projet complexe avec ear, wars, ejb, structs, xdoclet
- À exécuter : Maven genapp

- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

+ Cas pratiques

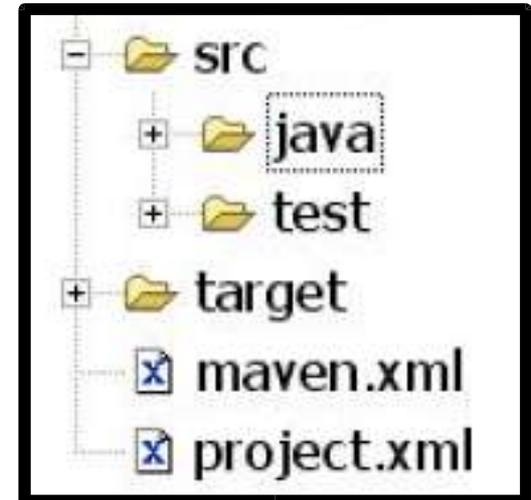
Installation
 Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

Simple projet Jar :

- project.xml
 - **Maven POM**
- project.properties (optionnel)
 - **Configurations des plugins**
- build.properties (optionnel)
 - **Configuration locale**
- maven.xml (optionnel)
 - **Goal personnalisé**
- target/
 - **Dossier pour fichiers générés**
- xdocs/
 - **Documentation en xml**
- src/java
 - **Dossier Code source**
- src/test
 - **Dossier Test JUNIT**
- src/conf
 - **Fichier de configuration**



- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

+ Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

Information pour la compilation

- Fichier POM (project.xml) entre tag <build>
- Répertoire(s) dédié(s) pour les sources

```
<build>
  ...
  <sourceDirectory>
    ${basedir}/src
  </sourceDirectory>
  ...
</build>
```

Compiler : maven java:compile

- Répertoire target : \${project_home}/target/classes
- Maven clean:clean : supprime tous les fichiers générés

Packaging : maven jar:jar

- Fichier jar créé : <id>-<version>.jar

- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

+ Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

Création du site basée sur la POM

Génère un site de documentation en pré exécutant les Goals suivants:

- Jdepend, checkstyle, changelog, activity, javadoc, jxr, junit-report, tasklist
- xdoc : tous documentations générées en xdoc
- JSTL transforme XDOC en HTML

Le plan du site est défini à travers xdoc/navigation.xml

```
<project>
  [...]
  <build>
    [...]
    <reports>
      <report>maven-junit-report-plugin</report>
      <report>maven-checkstyle-plugin</report>
    </reports>
  </build>
</project>
```

A exécuter : Maven site:generate

- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

+ Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

- Plugins existants

Code source
Documentations
Activités CVS
Dépendances

 **Exécute les tests unitaires JUNIT**

 **Créer un rapport xdoc**

 **Puis xdoc vers HTML**

 **Le rapport de test JUNIT est ensuite intégré au site généré**

- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

+ Plugins existants

Code source
Documentations
Activités CVS
Dépendances

- ant
- antlr
- appserver
- ashkelon
- aspectj
- cactus
- castor
- changelog
- changes
- checkstyle
- clean
- clover
- codeswitcher
- console
- deploy
- developer-activity
- dist
- docbook
- ear
- eclipse
- ejb
- examples
- faq
- file-activity
- genapp
- gump
- hibernate
- html2xdoc
- idea
- j2ee
- jalopy
- jar
- java
- javadoc
- jboss
- jbuilder
- jdee
- jdepend
- jdeveloper
- jdifff
- jellydoc
- jnlp
- junit-report
- junitdoclet
- jxr
- latex
- latka
- license
- linkcheck
- native
- pdf
- perforce
- plexus
- plugin
- pmd
- pom
- release
- repository
- runner
- sea
- shell
- site
- statcvs
- struts
- summit
- tasklist
- test
- torque
- touchstone
- touchstone-partner
- uberjar
- vdoclet
- war
- was40
- webserver
- wizard
- word2html
- xdoc

... **Et Plus**

- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

+Plugins existants

 Code source
Documentations
Activités CVS
Dépendances



Checkstyle

- Analyse le code selon les conventions de codage
- Plusieurs style de codage peut être défini:
 - Convention Sun, Turbine ou Avalon
 - Sa propre convention
- Les règles peuvent être personnalisé à travers un fichier XML
 - Longueur maximum d'une ligne
 - Patterns des variables et constantes
 - Longueur de la tabulation



Exemple de rapport

Checkstyle Results			
The following document contains the results of Checkstyle  .			
Summary			
Files	Infos	Warnings	Errors
57	0	0	567
Files			
Files	I	W	E
org/apache/maven/changelog/AbstractChangeLogGenerator.java	0	0	28
org/apache/maven/changelog/ChangeLog.java	0	0	34
org/apache/maven/changelog/AbstractChangeLogGenerator.java			
Error			Line
 Line has trailing spaces.			49
 Line has trailing spaces.			54
 Variable 'base' must be private and have accessor methods.			57
 Line has trailing spaces.			59
 Line has trailing spaces.			61

- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

+ Plugins existants

 Code source
Documentations
Activités CVS
Dépendances

Jalopy

- Permet de formater le code source
 - Indentation
 - Générer modèle de commentaire javadoc
 - Organiser l'ordre des clauses import et les membres d'une classe

Bon compromis

- Utiliser Jalopy et checkstyle
- Supprimer de checkstyle tous ce qui peut être corrigé par Jalopy

- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

+ Plugins existants

Code source
 Documentations
Activités CVS
Dépendances

Plugin Javadoc

- Génère la javadoc

Plugin Tasklist

- Génère un rapport avec toutes les tâches à faire listées, dans la javadoc du code sources, par le tag @todo

Todos

Class	Class Todos Count	Method with Todos
<code>org.apache.maven.cvslib.CvsChangeLogGenerator</code>	0	1

`org.apache.maven.cvslib.CvsChangeLogGenerator`

Method Todos

`getEntries()`: Currently the output from the logListener is a String, which is then converted to an InputStream. The output of logListener really should be an input stream.

-

- Introduction

La problématique
 Ant ou Maven
 Objectifs de Maven

- Concepts

POM
 Repository
 Goal

- Cas pratiques

Installation
 Générer un projet
 Compilation
 Site web du projet
 Test unitaire

+ Plugins existants

Code source
 Documentations
 Activités CVS
 Dépendances

Plugin Java Cross Reference

- Transforme les fichiers sources en HTML
- Fournit une syntaxe en couleur et des liens

Plugin PMD

- Analyse le code source

PMD Results

The following document contains the results of PMD .

Summary

Files	Errors
3	4

Files

Files	Violations
org/apache/maven/clearcaselib/ClearcaseChangeLogGenerator.java	1
org/apache/maven/clearcaselib/ClearcaseChangeLogParser.java	2

[org/apache/maven/clearcaselib/ClearcaseChangeLogParser.java](#)

Violation	Line
Avoid unused private fields such as 'LOG'	49
Avoid unused private fields such as 'GROUPED_ENTRY_KEY_TIMESTAMP_FORMAT'	73

- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

+ Plugins existants

Code source
Documentations
 Activités CVS
Dépendances

Plugin Activity

- accède directement au CVS pour extraire les informations d'activités de développement
 - Changes et Commits
 - Nombre de Commit par développeur
 - Nombre de « changement » d'un fichier

Activity by File

Timeframe: 30 days, Total Commits: 12 Total Number of Files Changed: 32

File Name	Number of times changed
project.xml 	7
xdocs/changes.xml 	6
src/main/org/apache/maven/clearcaselib/ClearcaseChangeLogParser.java 	2

Activity by Developer

Timeframe: 30 days, Total Commits: 12 Total Number of Files Changed: 32

Name	Number of Commits	Number of files changed
Emmanuel Venisse	3	17



- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

+ Plugins existants

Code source
Documentations
Activités CVS
Dépendances

Plugin changelog

- Résume tous les *changeslog* d'un CVS
- Produit un rapport clair
- Supporté : CVS, Perforce, StarTeam, IBM Rational ClearCase and Subversion

Changelog Report		
Timeframe: 30 days, Total Commits: 12 Total Number of Files Changed: 32		
Date	Author	File/Message
2004-07-26 02:47:28	dion	xdocs/navigation.xml v 1.6 Add tags doc to nav
2004-07-26 02:42:53	dion	xdocs/tags.xml v 1.1 Add tags
2004-07-26 02:39:28	dion	xdocs/changes.xml v 1.41 Add missing release tag
2004-07-15 00:27:45	dion	project.xml v 1.51 Upgrade to regexp 1.3
2004-07-14 22:50:40	dion	project.xml v 1.50 Upgrade maven dependency to 1.0
2004-07-14 22:41:12	dion	project.xml v 1.49 Use later version of jelly xml tags

- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

+ Plugins existants

Code source
Documentations
Activités CVS
Dépendances

Plugin JDepend :

- Analyse les fichiers .class
- Fournit un rapport sur les packages dépendants
- Trouve les dépendances cycliques
- Génère un rapport qui est ensuite intégré au site généré

- Introduction

La problématique
Ant ou Maven
Objectifs de Maven

- Concepts

POM
Repository
Goal

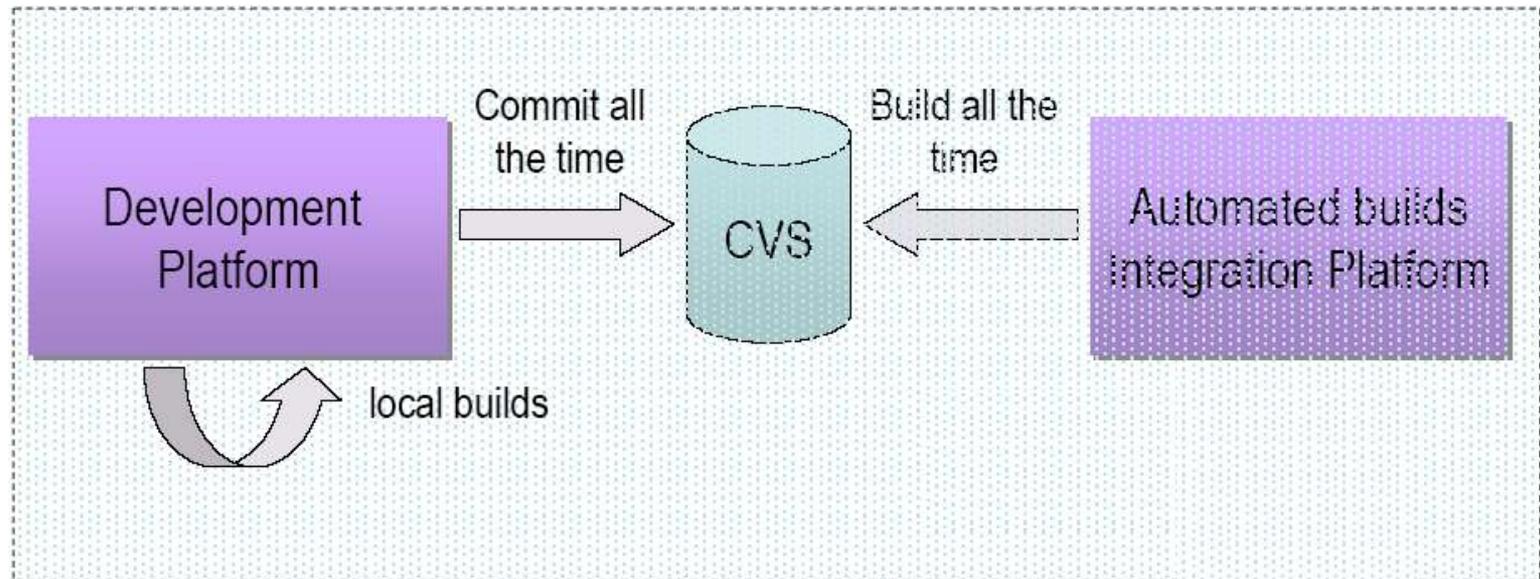
- Cas pratiques

Installation
Générer un projet
Compilation
Site web du projet
Test unitaire

+ Plugins existants

Code source
Documentations
Activités CVS
Dépendances

+ Intégration





Apache Maven Project

<http://maven.apache.org/>

QUESTION?

.....