# **ANT**

Moteur de production

1

# **SOMMAIRE**

Présentation
Pour commencer
Propriétés et Types
Quelques Tâches
Principales Tâches pour un projet Java
Conclusion

ANT

### Présentation

3

### Qu'est ce que Ant?

- □ Un outil de développement open-source écrit en Java
- Appartient au projet Jakarta de la fondation Apache
- □ Equivalent au make du C++
- Utilise le formalisme XML pour son fichier de configuration (build.xml)
- □ Version 1.0 est apparue en mars 2000

13/10/2008 ANT

## Un outil de développement?

- □ Le développement d'une application peut nécessiter plusieurs tâches :
  - Compilation
  - Linkage
  - Copie de fichiers
  - Archivage (packaging)
  - Génération d'une documentation
  - ...
- □ Rébarbatif si réalisé manuellement
- □ Utilité d'automatiser toutes ces tâches

13/10/2008 ANT

5

#### Pour commencer

- 1. Installation
- 2. Exécution
- 3. Exemple de build.xml
- 4. Explication des concepts project, target, depends, task

#### Installation « en dur »

- □ Téléchargez l'archive : <a href="http://ant.apache.org/">http://ant.apache.org/</a>.
- □ Décompresser l'archive dans un répertoire
- □ Mettre à jour les variables d'environnements
  - ANT\_HOME doit pointer vers le répertoire de Ant.
  - Ajouter le répertoire bin de Ant au PATH.
  - JAVA\_HOME doit pointer vers le répertoire du JDK

Windows	SET ANT_HOME=C:\ant\apache-ant-1.7.1 SET PATH=%PATH%;%ANT_HOME%\bin SET JAVA_HOME=C:\JDK
Linux	export ANT_HOME=/usr/lib/ant export PATH=\${PATH}:\${ANT_HOME}/bin export JAVA_HOME=/usr/lib/jdk

13/10/2008 ANT

7

#### **Execution**

- □ Pas d'interface graphique.
- □ Ligne de commande.

ant [options] [target1 [target2 ...]]

- Ant utilise par défaut le fichier build.xml.
- L'option -f permet de forcer le fichier à utiliser.
- □ Exemples

```
ant
ant -help
ant clean
ant -f projet.xml
ant -f projet.xml clean
```

13/10/2008 ANT 8

#### Structure d'un fichier Ant

- □ Un fichier Ant contient un et un seul **projet.**
- □ Un **projet** contient au minimum une **cible**.
- □ Une **cible** contient zéro ou plusieurs **tâches**.

#### Note:

En anglais les termes sont project pour projet, target pour cible et task pour tâche.

13/10/2008 ANT

## Exemple de fichier Ant

```
<?xml version="1.0" ?>
project default="main">
  <target name="main" depends="compile,compress">
    <echo>Building the .jar file.</echo>
  </target>
  <target name="compile">
    <javac srcdir="." />
  </target>
  <target name="compress">
    <jar destfile="projet.jar" includes="*.class"/>
  </target>
</project>
13/10/2008
```

### Projet (project)

- □ Un **projet** est défini par le tag **project** qui est l'élément racine du fichier.
- □ Le tag **project** possède trois attributs **name**, **default** et **basedir**.
  - Seul default est requis. Il désigne la cible à exécuter par défaut.
- □ Exemple :

13/10/2008 ANT 11

11

### Cible (target)

- □ Une **cible** est un ensemble de tâches à réaliser dans un ordre précis.
- □ Elle peut dépendre d'autres cibles.
- Ces attributs sont name, depends, unless, if, et description.
  - Seul name est requis.
- □ Exemple :

13/10/2008 ANT 12

# Dépendance

```
<target name="A" depends="B,C"/>
> A est exécuté après B et C.

<target name="A"/>
<target name="B" depends="A"/>
<target name="C" depends="B"/>
<target name="D" depends="C,B,A"/>
    ant D
> Orde d'exécution : A - B - C - D.

<target name="A"/>
<target name="A"/>
<target name="B" depends="A"/>
<target name="B" depends="A"/>
<target name="C" depends="A,B"/>
    ant C
> Redondance, A est exécuter qu'une seule fois.

<target name="A" depends="B"/>
<target name="B" depends="B"/>
<target name="B" depends="B"/>
<target name="B" depends="A"/>
> Référence circulaire, génère une erreur.
```

13

## Tâche (task)

- □ Sert à réaliser un traitement bien défini.
- □ Peut avoir des attributs (ou paramètres) pour configurer le traitement à réaliser.
- □ Ant fournit en standard plus d'une centaine de tâches.
- □ Possibilité d'écrire ces propres **tâches** en Java.

13/10/2008 ANT 14

# Propriétés et Types

Property et type

15

## Propriétés

- □ Définie par le tag **property.**
- □ Immuable et utilisable dans l'ensemble du fichier.
- □ Pour utiliser une propriété, la syntaxe est : \${nom\_propriété}
- Cinq façons de définir une propriété avec les attributs name, value, resource, file, url, ou environment

13/10/2008 ANT 16

### Propriétés

#### □ Attribut name.

- Associé à l'attribut value, il permet de définir une valeur associée à la propriété.
- Associé à l'attribut location, il permet de définir un chemin absolu vers un fichier ou un répertoire.

#### □ Attributs file, resource ou url.

 Permettent d'utiliser un fichier de propriété au format défini par la classe java.util.Properties.

#### Attribut environment.

 Défini le préfixe à utiliser pour référencer une variable d'environnement (ex: ANT\_HOME, PATH)

13/10/2008 ANT 1

17

#### Propriétés - Exemples

18.19/2000

#### **PatternSet**

- □ Le tag patternset permet de définir un ensemble de motifs pour sélectionner des fichiers.
  - Utilisation de « jokers »
    - > ? Remplace un caractère.
    - > \* Remplace un ou plusieurs caractères.
    - \*\* Désigne tous les sous répertoires.
  - Attributs includes et excludes pour définir la liste des motifs à inclure ou exclure.
- □ Exclusions par défaut (ex: \*\*/\*~, \*\*/.\_\*).

13/10/2008 ANT 19

19

#### **Exemple**

- Groups all files in directory \${server.src} that are Java source files and don't have the text Test in their name.

- Groups the same files as the above example, but also establishes a PatternSet that can be referenced in other <fileset> elements, rooted at a different directory.
- <fileset dir="\${client.src}" > <patternset refid="non.test.sources"/> </fileset>

13/10/2008 ANT 20

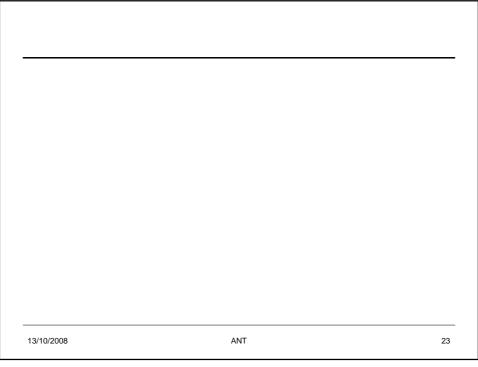
#### **FileSet**

- □ Le tag **fileset** permet de définir des ensembles de fichiers.
  - Attribut dir permet de définir le répertoire de base.
  - Attributs includes et excludes permettent de définir la liste des motifs à inclure ou exclure.
    - On peut aussi utiliser les tags include et exclude.
  - Attribut defaultexcludes permet d'ignorer les exclusions par défaut.
- □ En général, inclus dans une tâche qui gère des fichiers.

13/10/2008 ANT 21

21

### FileSet - Exemples



23

# Path

- □ Les tags **path** et **classpath** permettent de définir un ensemble de chemins d'accès.
  - classpath s'utilise uniquement dans une tâche.
- □ On utilise le tag **pathelement** pour définir un chemin.
  - Attribut location pour spécifiez un fichier ou un répertoire.
  - Attribut path pour spécifier un ensemble de chemins.

13/10/2008 ANT 2

# Path - Exemples

25

```
13/10/2008 ANT 26
```

# Quelques Tâches

27

# **Echo**

- Permet d'afficher un message durant l'exécution des traitements ou d'écrire dans un fichier.
- □ Exemple :

```
<echo message="Bonjour"/>
<echo>Bonjour</echo>
<echo file="log.txt" append="false" message="Bonjour"/>
<echo file="log.txt" append="true" message="Bonjour"/>
```

13/10/2008 ANT 28

#### MkDir

- □ Permet de créer un répertoire.
- Attribut dir précise le chemin et le nom du répertoire à créer.
- □ Exemple:

```
<target name="init">
  <mkdir dir="${basedir}/bin" />
  <mkdir dir="${basedir}/lib" />
  <mkdir dir="${basedir}/doc" />
  </target>
```

ANT

29

13/10/2008

### Copy

- □ Permet de copier un ou plusieurs fichiers.
- □ Règle de copie d'un fichier :
  - Le fichier de destination n'existe pas ou est moins récent.
  - L'attribut overwrite est présent.
- Exemple :

#### Delete

- Permet de supprimer des fichiers ou des répertoires.
- □ Première syntaxe :
  - Attribut dir qui spécifie le répertoire à supprimer.
     ATTENTION: Tous ses fichiers et sous-répertoires seront supprimés.
  - Attribut file spécifie le fichier à supprimer.
- □ Seconde syntaxe avec le tag **fileset**.
  - Attribut includeEmptyDirs qui permet de préciser la suppression des répertoires vides.

13/10/2008 ANT 31

31

### Delete - Exemples

```
<delete file="log.txt" />
<delete dir="lib" />
```

> Supprime le fichier log.txt, le répertoire lib et ces sous-répertoires.

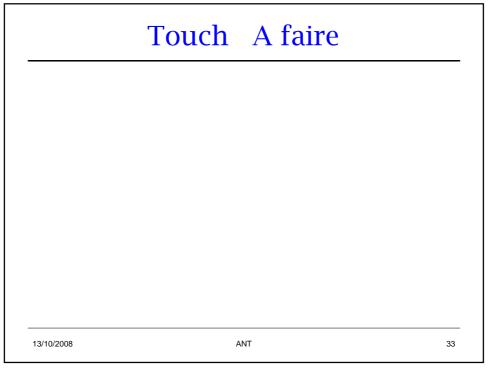
```
<delete>
    <fileset dir="build" includes="**/*.log"/>
</delete>
```

> Supprime tous les fichiers .log du répertoire et des sous-répertoires de build.

```
<delete includeemptydirs="true">
    <fileset dir="classes" includes="**/*"/>
</delete>
```

> Supprime tous les fichiers et sous-répertoires de **classes**, ainsi que le répertoire **classes**.

13/10/2008 ANT 32



13/10/2008 ANT 34

# Principales Tâches pour un projet Java

35

### **Javac**

- □ Permet la compilation de fichiers source.
  - Attribut srcdir précise le répertoire contenant les fichiers sources, sauf si le tag src est utilisé.
  - Attribut **destdir** précise le répertoire de destination.
  - Attribut ou tag classpath pour désigner le classpath à utiliser.
- □ Le tag **compilerarg** permet de spécifier des arguments de ligne de commande.

13/10/2008

ANT

# Javac -Exemples

```
<javac srcdir="${src}" destdir="${classes}"/>
<javac destdir="${classes}">
    <src path="${src}"/>
    <classpath refid="projet.classpath"/>
<javac srcdir="${src}" destdir="${classes}">
    <classpath refid="projet.classpath"/>
    <compilerarg value="-xlint"/>
</javac>
<javac srcdir="${src}" destdir="${classes}">
    <include name="monpackage/p1/**"/>
    <include name="monpackage/p2/**"/>
    <exclude name="monpackage/p1/testpackage/**"/>
</javac>
13/10/2008
                                ANT
                                                                  37
```

37

### **Javadoc**

- □ Permet la génération d'une documentation.
  - Attribut sourcepath détermine le répertoire de base qui contient les sources, sauf si tag fileset est utiliser.
  - Attribut destdir détermine le répertoire qui va contenir les fichiers de documentation générés.
- □ Exemple :

#### Jar

- □ Permet le création d'une archive jar.
  - Attribut jarfile spécifie le nom du fichier jar à créer
  - Attribut basedir précise le répertoire qui contient les éléments à ajouter dans l'archive.
  - Attribut manifest précise le fichier manifest qui sera utilisé dans l'archive.
    - On peut utiliser le tag manifest pour créer le fichier manifest dans la tâche Jar.

#### Exemple

```
<jar jarfile="${build}/demo.jar" basedir="${classes}"
manifest="${src}/MANIFEST.MF" />
```

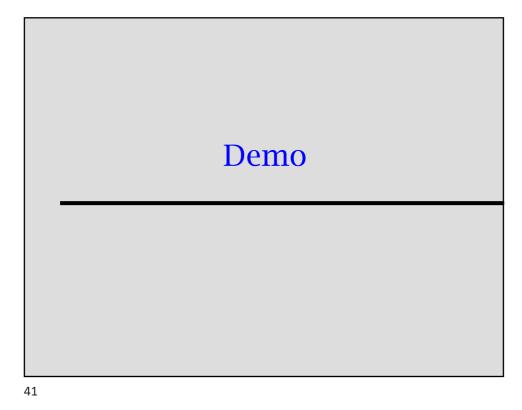
13/10/2008 ANT 39

39

#### Java

- □ Permet de lancer une application.
  - Attribut classname spécifie le nom de la classe à exécuter.
  - Attribut jar spécifie le nom du fichier jar à exécuter.
  - Attribut classpath détermine le classpath pour l'exécution.
    - Il est aussi possible d'utiliser un tag classpath.

#### □ Exemple :



# Conclusion

- □ Ant est un outil de développement.
- □ Utilise le formalisme XML.
- □ Permet d'automatiser un grand nombre de tâches.
- □ Supporté par de nombreux IDE, comme Eclipse, IntelliJ IDEA, NetBeans,...
- □ Une version 2.0 ??

ANT 4