# Petit tutoriel sur
# java, javac et javadoc

Stéphane NICOLAS

24 mai 2001

**Résumé**

Le présent document vous aidera à mieux comprendre le fonctionnement des outils de la jdk, à savoir :

- java : l'interpréteur java
- javac : le compilateur de java
- javadoc : le générateur de documentation de code source

Ce texte vous aidera à mieux comprendre le fonctionnement de ces outils et leurs options. De nombreux/SES étudiantEs ont des difficultés à comprendre le fonctionnement des outils en ligne de commande, en suivant attentivement les conseils indiqués ici, vous devriez pouvoir utiliser sans problème ces outils.

## Table des matières

# 1 javac : Le compilateur java de la SDK

javac transforme les *fichiers* «.java» en *fichiers* «.class». Les fichiers «.class» sont dans un format appelé ByteCode, qui permet leur interpretation par la JVM.

## 1.1 Les options de javac

Format general des lignes de commandes de javac :
javac [options] <fichier java> <fichier java> ....

- -d <DEST>
  indique le répertoire de destination de tout ce qui est géneré par javac. Que javac crée des repertoires (pour les packages) ou des fichiers .class (pour les classes et interfaces), javac génerera tout à partir du repertoire DEST.
  **NOTE** : Le repertoire DEST doit exister avant de lancer la commande.
  (La commande mkdir <DEST> peut être utilisée pour créer des repertoires sur les deux OS.)
  Si on ne spécifie pas cette option, javac dépose les fichiers «.class» dans le même répertoire que le fichier java correspondant.
- -classpath <liste de repertoires et de fichiers jars>
  Le caractère " :" (resp " ;") sépare les différents éléments de la liste sur linux (resp windoz).
  L'option classpath est sans doute la plus importante option de javac, elle permet d'indiquer à la JVM l'endroit où elle pourra trouver les classes dont elle a besoin pour compiler les fichiers .java . Ce qu'il est important de comprendre ici est l'association (mapping) entre répertoire et nom de package de java.
  Supposons que l'on veuille compiler le fichier Toto.java dans lequel on déclare la classe Toto

comme appartenant au package titi. Supposons également que l'on aie besoin pour compiler cette classe des classes du package tata et que les fichiers «.class» du package tata se trouvent dans le repertoire C :/temp/tata/*.class.

Alors, javac compilera le fichier Toto.java avec la commande :

javac -classpath c :/temp <chemin relatif ou absolu du fichier Toto.java> En effet, lorsque javac a besoin des classes du package tata, il va chercher un répertoire tata contenant les fichiers .class qu'il lui faut... et le répertoire C :/temp contient bien un repertoire tata contenant les fichiers «.class».

Il est très important de comprendre que le répertoire C :/temp/tata n'est pas une entrée valide du classpath.

On indique bien à java l'endroit où se trouve l'endroit qui lui permet d'accéder aux classes, et pas l'endroit où se trouve le fichier «.class». Les fichiers jars qui peuvent etre specifiés avec l option -classpath sont des fichiers compressés. Les fichiers jars contiennent en fait une arborescence de répertoires et sont donc une entrée valide du classpath.

**NOTE** : Dans le cas où les classes ne sont pas incluses dans un package, vous devez indiquez après l'option -classpath, l'endroit exact où se trouve vos classes, c'est à dire le nom du répertoire qui contient le fichier «.class».

# 2 java : L'interpreteur java de la SDK

java execute des *classes* (**et non des fichiers «.class»** ), ces classes doivent contenir une méthode ayant la signature

```
public static void main(String[] args)
```

pour être éxécutée par la JVM.

## 2.1 Les options de java

Format general des lignes de commandes de javac :

java [options] <nom complet de la classe>

– -classpath <liste de répertoires et de fichiers jars>

Le caractere " :" (resp " ;") sépare les differents éléments de la liste sur linux (resp windoz).

L'option classpath est sans doute la plus importante option de java, elle permet d'indiquer à la JVM l'endroit où elle pourra trouver la classe a éxecuter et les classes dont elle dépend.

Ce qu'il est important de comprendre ici est l'association (mapping) entre répertoire et nom de package de java. Supposons que l'on veuille éxecuter la classe titi.Toto (défine dans le fichier Toto.java. Ce fichier ayant pour première ligne java : `package titi`. La classe Toto a une methode `public static void main(String[] args)`.)

Votre fichier Toto.class se trouve dans le répertoire titi, lequel se trouve dans le répertoire c :/temp. Alors, java éxecutera la classe Toto avec la commande :

java -classpath C :/temp titi.Toto

java doit éxecuter la classe titi.Toto, il chercher donc un répertoire titi contenant le fichier Toto pour ensuite l'éxecuter. Et le répertoire C :/temp contient bien un repertoire titi qui contient Toto.class ...

Il est très important de comprendre que le répertoire C :/temp/titi n'est pas une entrée valide du classpath. On indique bien à java l'endroit où se trouve l'endroit qui lui permet d'accéder aux classes, et pas l'endroit où se trouve le fichier «.class».

Les fichiers jars qui peuvent être spécifiés avec l'option -classpath sont des fichiers compressés. Les fichiers jars contiennent en fait une arborescence de répertoires et sont donc une entrée valide du classpath.

# 3 javadoc : Le generateur de document (du code source) de la SDK

javadoc crée un ensemble de fichiers «.html» qui vous permettent de documenter vos fichiers java. La documentation générée par javadoc est à l'attention des programmeurs qui desirent comprendre ou utiliser les fonctionnalités de votre logiciel. Il sert à en donner les spécifications mais *ne sert pas* à en expliquer chaque ligne de code.

Pour plus de détails sur l'utilisation de javadoc dans un fichier java, reportez vous à l'annexe G du livre de Deitel et Deitel. Le présent document explique uniquement l'utilisation du programme javadoc.

## 3.1 Les options de javadoc

Format général des lignes de commandes de javadoc :

javadoc [options] <nom de package à documenter> <nom de package à documenter> . . .

ou

javadoc [options] <nom de fichier java à documenter> <nom de fichier java à documenter> . . .

– -sourcepath <liste de répertoires >

  Le caractère " :" (resp " ;") sépare les différents éléments de la liste sur linux (resp windoz).

  L'option sourcepath permet d'indiquer à la javadoc l'endroit où il pourra trouver les fichiers et packages à commenter .

  Le classpath fonctionne d'une facon très similaire au classpath.

  Supposons que l'on veuille documenter le package titi dont les fichiers java sont dans le repertoire C :/temp/titi/, on génère la documentation du package par la commande :

  javadoc -sourcepath C :/temp titi

  La documentation javadoc d'un package entier permet de naviguer par des liens html entre les documentation d'un package, si on compile plusieurs packages ensemble (avec une seule commande) leurs documentations sont navigables également. Malgre tout, si vous voulez *vraiment* faire une documentation individuelle pour chacun des fichiers sources, tapez la commande :

  javadoc -sourcepath C :/temp titi.java

– -d <DEST>

  indique le répertoire de destination de tout ce qui est généré par javadoc. Que javadoc crée des répertoires (pour les packages) ou des fichiers «.html» (pour les classes et interfaces), javadoc générera tout à partir du repertoire DEST. **NOTE** : Le répertoire DEST doit exister avant de lancer la commande. (La commande mkdir <DEST> crée des répertoires sur les deux OS.)

  Si on ne spécifie pas cette option, javadoc dépose les fichiers «.html» à partir du répertoire courant.

# 4 Documentations officielles

– Setting the classpath :
  http://java.sun.com/j2se/1.3/docs/tooldocs/win32/classpath.html
– JDK, Tools And Utilities :
  http://java.sun.com/j2se/1.3/docs/tooldocs/tools.html

# 5 Licence du présent document

Ce document est soumis à la licence GPL (Gnu Public Licence), cette licence vous permet de modifier, distribuer, copier et utiliser tout document. Dans le cas où vous trouver des améliorations à ce document, veuillez les faire parvenir à l'adresse : ift-21133@ift.ulaval.ca.

La licence GPL peut etre trouvée à l'adresse : http://www.gnu.org/philosophy/license-list.html.