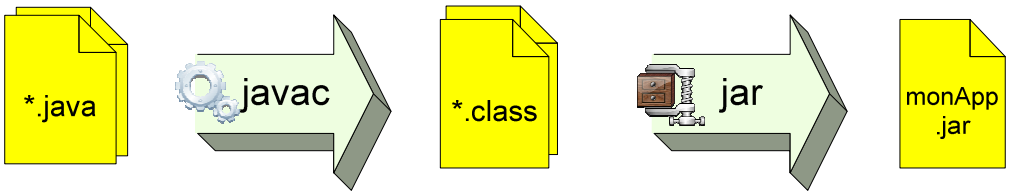
L'archive JAR pour une application Java Desktop

Table des matières

Pour toute information, reportez-vous sur le site de Sun à l'adresse suivante :
http://java.sun.com/javase/6/docs/technotes/guides/jar/jar.html#Intro

1 Présentation de l'archive java pour une application « desktop »



Application desktop = application standalone
Une archive Java est un format de fichier particulier permettant de regrouper dans un seul fichier (l'archive) plusieurs autres fichiers. En général, on regroupe ainsi tous les fichiers nécessaires au fonctionnement d'une application. Ceci comprend bien sûr les fichiers `.class` mais également toutes les autres ressources indispensables au bon fonctionnement de l'application (les ressources).

Cette possibilité de regroupement procure de nombreux avantages pour le déploiement d'applications.

- Le premier et certainement le plus appréciable réside dans le fait que le déploiement d'une application se résume à la recopie d'un seul et unique fichier sur le poste client. Même si l'application exige pour son fonctionnement plusieurs ressources organisées sous forme d'une arborescence bien précise. Cette arborescence est créée à l'intérieur du fichier archive et n'a pas besoin d'être reproduite sur le poste client.
- Les fichiers archive peuvent être compressés afin d'optimiser leur stockage et leur transfert sur un réseau.
- Cet avantage est encore plus appréciable pour les applets qui peuvent être ainsi récupérées par le navigateur avec une seule requête http.
- La sécurité est également améliorée par signature et le scellement de l'archive.
- Le format des archives étant standard, il n'y a aucune restriction vis-à-vis d'un système spécifique.

2 La commande jar : manipulation des archives

La manipulation d'une archive Java (fichier `jar`) reprend les mêmes principes que la manipulation d'une archive dans le monde unix avec la commande `tar`. Les options de la commande `jar` permettant de manipuler une archive Java sont d'ailleurs très ressemblantes à

celles de la commande `tar` d'Unix. Le format utilisé en interne par les fichiers archive est également bien connu puisqu'il s'agit du format `ZIP`. Les fichiers archive Java peuvent d'ailleurs être manipulés avec des outils dédiés à la manipulation de fichiers `ZIP`.
La commande standard de manipulation d'archive est la commande `jar`. Elle fait partie des outils fournis avec le jdk.

La syntaxe de base de manipulation d'une archive Java est la suivante :

```
jar [options [argumentDoptions] ] <listeFichiers>
```

2.1 Création d'une archive JAR

La syntaxe de base de création d'une archive Java est la suivante :

```
jar cf <nomDuFichier> <listeFichier>
```

Avec

- `c` est destiné à indiquer à la commande `jar` que l'on souhaite créer une archive
- `f` indique que la commande doit générer un fichier.
- <nomDuFichier> est le paramètre qui suit l'option `f` et donc, qui indique le nom du fichier d'archive (par convention, l'extension sera jar).
- `<listeFichier>` est le dernier argument et représente le ou les fichiers à inclure dans l'archive ; si plusieurs fichiers sont à inclure dans l'archive, leurs noms doivent être séparés par un espace ; le caractère joker * est également autorisé dans la liste ; si un nom de répertoire est présent dans la liste, son contenu entier est ajouté à l'archive.

L'archive est générée dans le répertoire courant. Un fichier manifest par défaut est également ajouté à l'archive. Les options suivantes sont également disponibles.

- **`V`** affiche le nom des fichiers au fur et à mesure de l'ajout dans l'archive.
- **`0`** désactive la compression du fichier archive.
- **`M`** désactive la génération du manifest.
- **`m`** ajoute le manifest indiqué à l'archive.
- **`-C`** supprime le nom de répertoire dans l'archive.

2.2 Visualisation du contenu

Le contenu d'une archive peut être visualisé avec la commande suivante :

```
jar tf ardoise.jar
```

La commande affiche sur la console le contenu de l'archive.

```
META-INF/MANIFEST.MF
.classpath
.project
Client$1.class
Client$2.class
Client$3.class
Client$4.class
Client$5.class
Client$6.class
Client.class
ClientArdoiseMagique.class
PanelDessin.class
ThreadEntree.class
```

Des informations supplémentaires peuvent être obtenues en ajoutant l'option `v` à la commande. La date de modification et la taille des fichiers sont ajoutées au résultat de la commande.

```
jar tvf ardoise.jar
```

Les chemins d'accès aux fichiers sont affichés avec le caractère / comme séparateur et sont relatifs à la racine de l'archive. Le contenu de l'archive n'est bien sûr pas modifié par l'exécution de cette commande.

2.3 Extraction

Les fichiers peuvent être extraits de l'archive avec la commande suivante :

```
jar xvf ardoise.jar
```

Les fichiers présents dans l'archive sont recréés sur le disque dans le répertoire courant. Si l'archive contient une portion d'arborescence, celle-ci est recréée dans le répertoire courant. Les éventuels fichiers et répertoires existants sont écrasés par ceux présents dans l'archive. L'extraction d'une archive peut être sélective en indiquant comme paramètre supplémentaire la liste des fichiers à extraire de l'archive en séparant les noms de ces fichiers par un espace. La commande suivante permet d'extraire de l'archive uniquement le fichier ClientArdoiseMagique.class :

```
jar xvf ardoise.jar ClientArdoiseMagique.class
```

Le contenu de l'archive n'est pas modifié par cette commande.

2.4 Mise à jour

Le contenu d'une archive peut être mis à jour par l'ajout de fichiers après sa création. Il faut dans ce cas utiliser la commande suivante :

```
jar uf ardoise.jar connect.gif
```

Le dernier paramètre de cette commande représente la liste des fichiers à mettre à jour dans l'archive. Si ces fichiers n'existent pas dans l'archive, ils sont ajoutés sinon ils sont remplacés par la nouvelle version. Si l'archive contient des répertoires, le chemin d'accès complet doit être spécifié dans la liste des fichiers à ajouter.

2.5 Exécution

Une application présente dans une archive Java peut être exécutée directement depuis l'archive sans avoir besoin d'extraire les fichiers. Vous devez indiquer à la machine virtuelle Java qu'elle doit elle-même extraire les fichiers de l'archive en utilisant l'option `-jar` lors du lancement de l'application. Le nom du fichier archive doit être spécifié à la suite de cette option.

```
java -jar ardoise.jar
```

La machine virtuelle Java a cependant besoin d'une information supplémentaire pour déterminer quel fichier de l'archive contient la méthode `main` par laquelle elle doit débuter l'exécution de l'application. Pour cela, elle recherche le fichier manifest de l'archive qui doit bien sûr contenir cette information pour que l'application puisse être exécutée à partir de l'archive.

3 Le fichier manifest

Les fichiers archives Java sont bien plus que des simples fichiers compressés puisqu'ils proposent une multitude de fonctionnalités complémentaires :

- Exécution directe depuis l'archive.
- Signature du contenu de l'archive.
- Scellement de parties de l'archive.
- Gestion des versions.

Toutes ces fonctionnalités sont disponibles grâce au fichier manifest inclus dans l'archive.

3.1 Présentation

Le fichier manifest est un simple fichier texte contenant des couples nom de paramètre et valeur de paramètre. Ces deux informations sont séparés par le caractère : .
Ce fichier est toujours nommé `MANIFEST.MF` et se trouve dans le répertoire `META-INF` de l'archive.

3.2 Création

À la création d'un fichier archive, un fichier manifest est créé par défaut. Il contient les informations suivantes :

```
Manifest-Version: 1.0
Created-By: 1.6.0 (Sun Microsystems Inc.)
```

La première ligne indique la version du fichier manifest, la seconde indique la version du jdk avec laquelle le fichier archive a été généré.
Pour ajouter d'autres informations au fichier manifest, vous devez procéder en deux étapes. Vous devez dans un premier temps préparer un fichier texte contenant les informations que vous souhaitez inclure dans le manifest de l'archive. La dernière ligne de ce fichier doit obligatoirement se terminer par un caractère retour chariot ou saut de ligne (ou les deux). Vous devez ensuite fusionner ce fichier texte avec le manifest par défaut de l'archive en utilisant l'option m de la commande `jar`. La syntaxe de la commande est donc la suivante :

```
jar cfm ardoise.jar infos.txt *
```

Cette commande génère un fichier archive nommé **ardoise.jar** contenant tous les fichiers du répertoire courant et ajoute au manifest par défaut les informations contenues dans le fichier **infos.txt**. Ce fichier peut par exemple contenir une information permettant d'indiquer le nom de la classe contenant la méthode `main` par laquelle doit débuter l'exécution de l'application.
Le fichier **infos.txt** contient dans notre cas la ligne suivante :

```
Main-Class: ClientArdoiseMagique
```

Ne pas oublier le retour chariot à la fin de la ligne et l'espace après le caractère : .
L'archive est générée avec le fichier manifest suivant :

```
Manifest-Version: 1.0
Created-By: 1.6.0 (Sun Microsystems Inc.)
Main-Class: ClientArdoiseMagique
```

3.3 Création depuis JAVA 6

La version de la commande jar fournie avec le jdk 6 propose également l'option e permettant d'indiquer le point d'entrée dans l'application sans avoir à créer de fichier intermédiaire. La syntaxe peut donc être la suivante :

```
jar cvfe ardoise.jar ClientArdoiseMagique.class *
```

4 Packaging d'une application Java-JEE

(Empaquetage)
Nous avons vu que pour faciliter le déploiement d'une application, il est possible d'inclure l'ensemble des fichiers nécessaires au fonctionnement de l'application dans une archive Java. En Java-JEE (module web), le fichier d'archive correspondant porte l'extension war (web archive). Ce fichier est simplement recopié dans l'arborescence du serveur qui prend automatiquement en charge le déploiement de l'application web qu'il contient. Il est généré grâce à l'utilitaire jar.
La manipulation d'une archive Java (fichier jar ou war) reprend les mêmes principes que la manipulation d'une archive dans le monde Unix avec la commande tar. Les options de la commande jar permettant de manipuler une archive Java sont d'ailleurs étrangement ressemblantes à celles de la commande tar d'Unix. Le format utilisé en interne par les fichiers archive est également bien connu puisqu'il s'agit du format ZIP. Les fichiers archive Java peuvent d'ailleurs être manipulés avec des outils dédiés à la manipulation de fichiers ZIP.