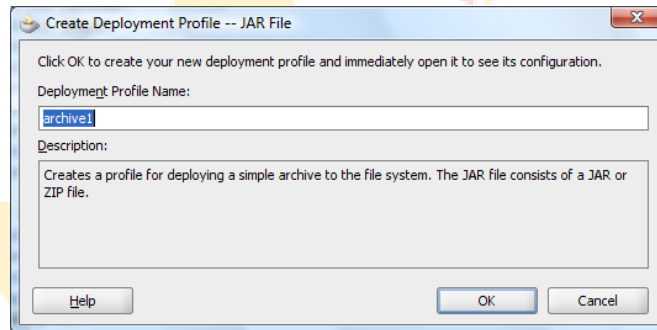


Annexe 10 - Procédure pour créer un Jar

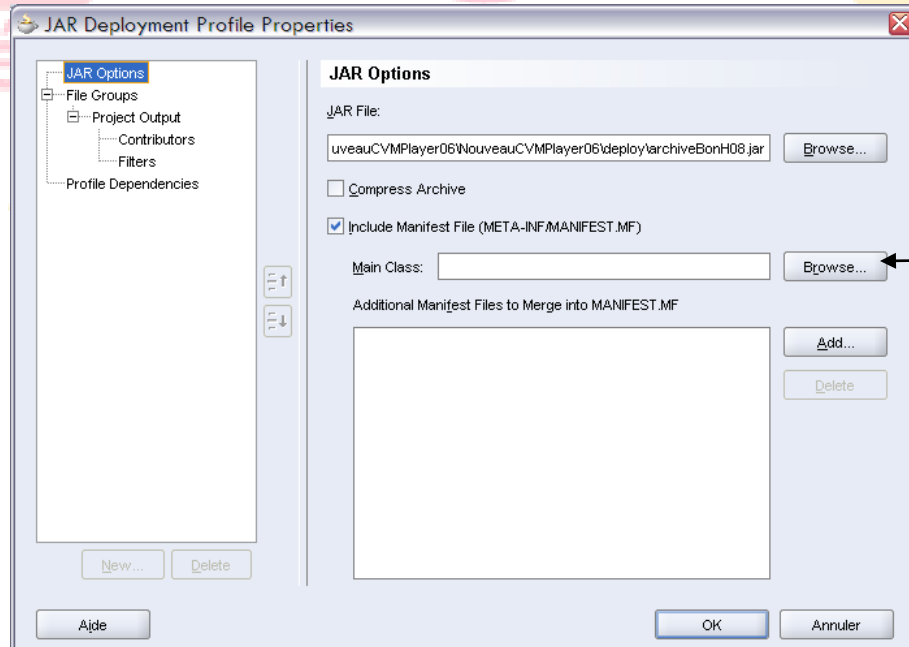
→ permet d'encapsuler une application dans un seul fichier JAR (Java Archive File). Le Jar étant exécutable, on peut donc démarrer l'application sans l'aide de JDeveloper. On doit simplement avoir une Java Virtual Machine (JVM) installée sur l'ordinateur.

A) JAR simple (ne comprenant pas d'autres librairies ou de ressources externes comme des images ou du son)

1. (procédure normale pour créer un Jar) Dans JDeveloper, créer un Deployment Profile (New → General → Deployment Profile → JAR file
2. La fenêtre suivante apparaît :

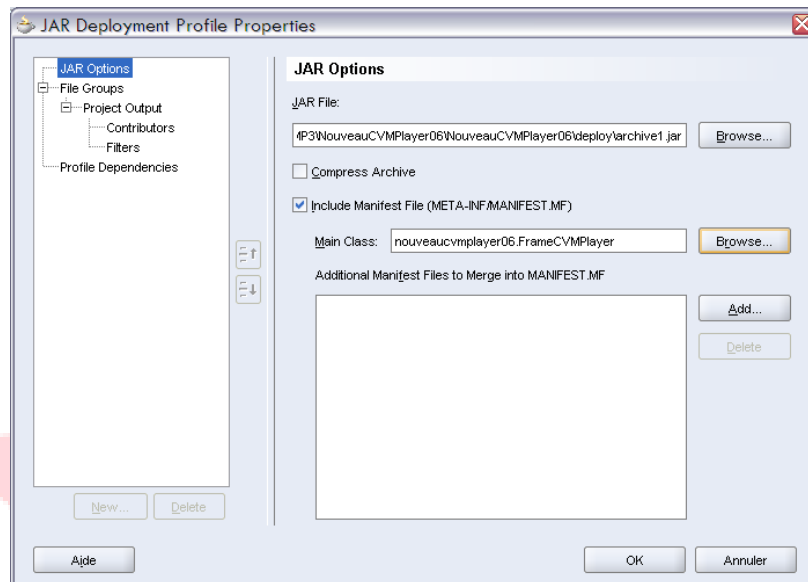


→ on choisit un nom pour l'archive Deployment Profile



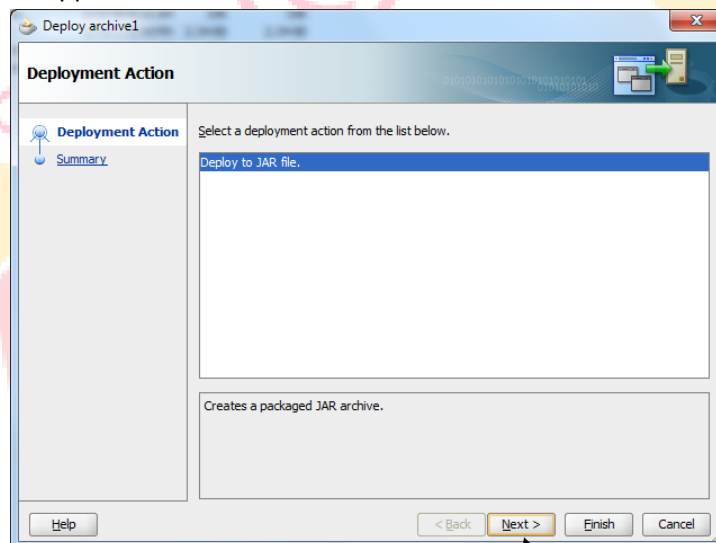
→ chercher le fichier de notre projet qui contient la méthode main

résultat :

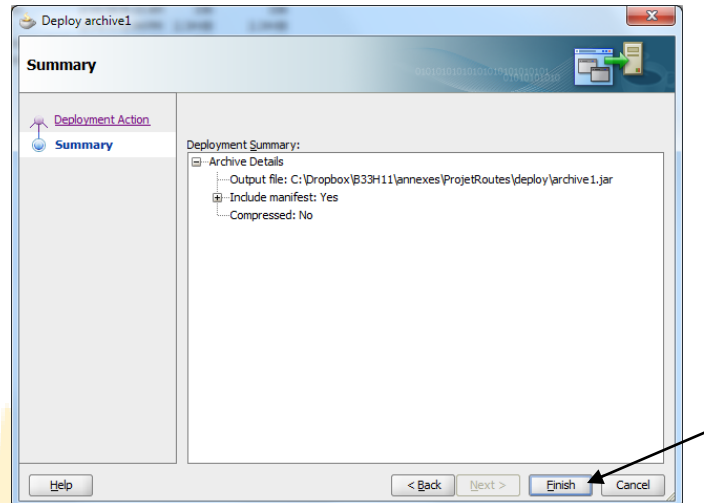


→ par exemple, le fichier FrameCVMP\Player comprend la méthode main pour ce projet

3. Faire bouton de droite sur le projet dans le navigateur → Deploy → choisir l'archive. La fenêtre suivante apparaît :



Faire Next

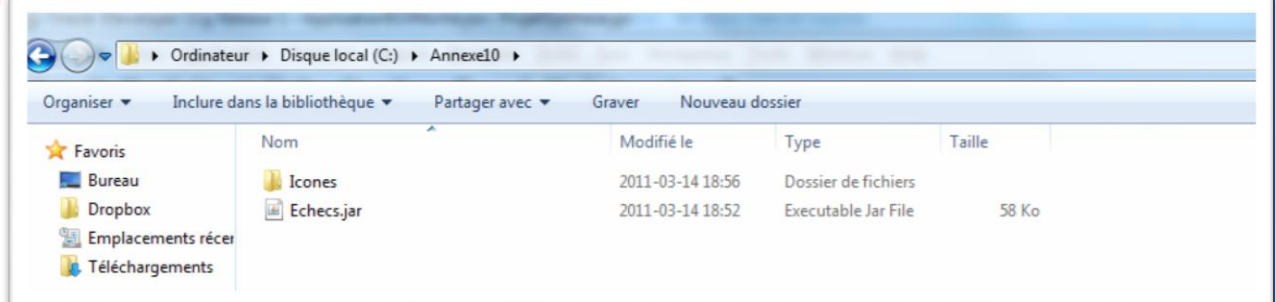


4. le JAR est créé dans un dossier `deploy` à l'emplacement de votre projet sur le disque. Vous pouvez l'utiliser dans n'importe quel contexte, sans les dossiers contenant les `.java` et les `.class`; l'archive JAR les contient.

B) JAR contenant des ressources externes (images par exemple)

La démarche suivante est basée sur le jeu d'échecs, où on a un dossier contenant des icônes représentant les pièces du jeu d'échecs.

Version simple : Faire la même démarche qu'en A) et utiliser le JAR avec le dossier `Icones`; le chemin relatif vers les icônes devient relatif au fichier `.JAR` :



On peut double-cliquer sur le Jar et utiliser le jeu d'échecs, le chemin relatif vers les icônes était `Icones/...`

Version complète (les images sont incluses dans le JAR)

1. On doit d'abord placer le dossier contenant les images ou autres ressources au même endroit que les fichiers `.class` (le jar est une archive des fichiers `.class` de l'application)

2. On doit dans le code changer la manière d'accéder à l'image (car le chemin relatif au .jpr n'est plus valide)

```
URL imageURL = FenetreJeuCorr.class.getResource("Icones/TN.gif");  
tab[0][0].setIcon(new ImageIcon ( imageURL));
```

où FenetreJeuCorr est le nom du fichier où les images seront affichées. Le processus serait le même pour un fichier .txt ou pour un fichier son.

3. On compile et on reprend la procédure de la partie A pour créer le JAR. De cette manière, vous n'aurez que le JAR à transporter.

C) JAR contenant d'autres JARs / librairies → basé sur le lecteur MP3

1. Créer, dans un fichier texte, un fichier manifest.mf à la main. Il contiendra les infos suivantes :

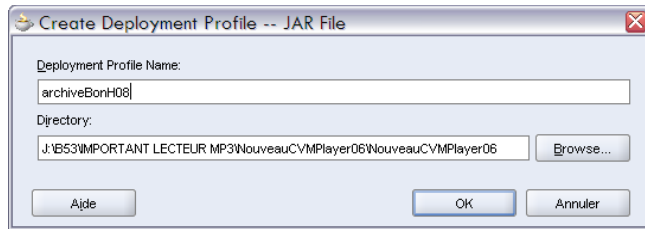
```
Manifest-Version: 1.0  
Created-By: Oracle JDeveloper 10.1.3.1.0  
Main-Class: nouveaucvmpayer06.FrameCVMPayer  
Class-Path: MpegAudioSPI1.9.4/mp3spi1.9.4.jar  
MpegAudioSPI1.9.4/lib/jl1.0.jar  
MpegAudioSPI1.9.4/lib/tritonus_share.jar
```

***important : les chemins doivent être relatifs par rapport à l'emplacement futur de votre .jar (dans cet exemple, le dossier MpegAudioSPI1.9.4 sera au même endroit que le .jar final)

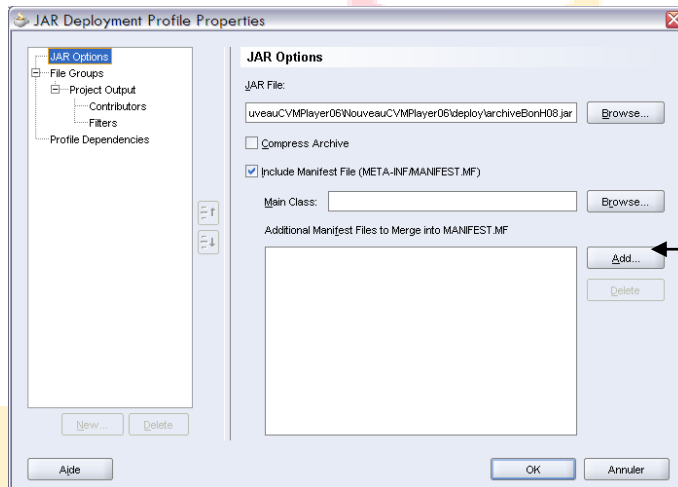
*** important : il faut laisser une ligne vide après le class-path puis sauvegarder.

*** main class : toujours *nom du package.nom du fichier contenant le main*

2. (procédure normale pour créer un Jar) Dans JDeveloper, créer un Deployment Profile (New → General → Deployment Profile → JAR file
3. La fenêtre suivante apparaît :



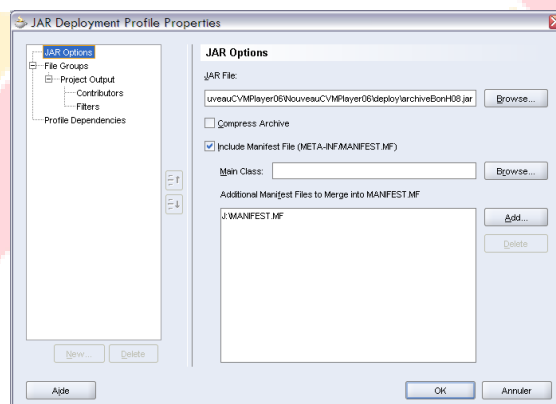
→ on choisit un nom pour l'archive Deployment Profile



→ pas nécessaire de trouver la classe contenant le main (déjà fait)

→ on doit ajouter notre *manifest file* précédent à celui-ci

résultat :



→ mon *manifest file* de l'étape 1 était sauvegardé directement sur le J

4. Faire bouton de droite sur le projet → Deploy → choisir l'archive → to JAR file

5. le JAR est créé dans un dossier deploy à l'emplacement de votre projet sur le disque

*** dans le cas du MP3 Player, si vous transportez le jar, vous devrez transporter le dossier MpegAudioSPI1.9.4 avec, il devra être au même endroit que votre JAR (Chemin relatif)

