Creating an Executable jar File

In Java, it is common to combine several classes in one .jar ("java archive") file. Library classes are stored that way. Larger projects (such as the Case Study in the AP program) use jar files. You can create your own jar file combining several classes, too.

jar files are created using the `jar.exe` utility program from JDK. You can make your jar file runnable by telling `jar.exe` which class has `main`. To do that, you first need to create a manifest file. A manifest is a one-line text file with a "Main-Class" directive. For example:

```
Main-Class: DanceStudio
```

This line must end with a newline.

A jar file created with a main class manifest can be used both as a library and a runnable jar. If you use it as a library, you can edit and compile any of the classes included in the jar, and add it to your project. Then it will override the one in the jar file.

You can create a manifest file in any text editor, or even by using the *MS-DOS* `echo` command. You can give your manifest file any name, but it's better to use something standard, such as `manifest.txt`.

Once you have a manifest and all your classes have been compiled, you need to run JDK's `jar.exe` utility. It is located in the JDK's `bin` folder, the same place where `javac.exe` and `java.exe`. are. `jar.exe` takes command-line arguments; if you run it without any arguments, it will display the usage information and examples. You need

```
C\mywork> jar cvfm MyJarName.jar manifest.txt *.class
```

`cvfm` means "create a jar; show verbose output; specify the output jar file name; specify the manifest file name." This is followed by the name you wish to give to your jar file, the name of your manifest file, and the list of `.class` files that you want included in the jar. `*.class` means all class files in the current directory.

Below are the detailed steps for doing this in *Command Prompt* and in *JCreator*.

Creating a jar File in *JCreator*

You can configure a "tool" that will automate the jar creation process. You only need to do it once.

1. Click on `Configure/Options`.
2. Click on `Tools` in the left column.
3. Click `New`, and choose `Create Jar file`.
4. Click on the newly created entry `Create Jar File` in the left column under `Tools`.
5. Edit the middle line labeled `Arguments`: it should have

   ```
   cvfm $[PrjName].jar manifest.txt *.class
   ```

6. Click OK.

Now set up a project for your program, create a manifest file `manifest.txt` or copy and edit an existing one. Place `manifest.txt` in the same folder where the `.class` files go. Under `View/Toolbars` check the `Tools` toolbar. Click on the corresponding tool button or press `Ctrl-1` (or `Ctrl-`*n* if this is the *n*-th tool) to run the `Create Jar File` tool.

With *Windows Explorer*, go to the jar file that you just created and double click on it to run.

Creating a jar File in *Command Prompt*

1. Start *Command Prompt*.
2. Navigate to the folder that holds your class files:

   ```
   C:\>cd \mywork
   ```

3. Set path to include JDK's bin. For example:

   ```
   C:\mywork> path c:\Program Files\Java\jdk1.5.0_09\bin;%path%
   ```

4. Compile your class(es):

```
C:\mywork> javac *.java
```

5. Create a manifest file:

```
C:\mywork> echo Main-Class: DanceStudio >manifest.txt
```

6. Create a jar file:

```
C:\mywork> jar cvfm DanceStudio.jar manifest.txt *.class
```

7. Test your jar:

```
C:\mywork> DanceStudio.jar
```

Skylight Publishing
support@skylit.com