

Creating a JAR package in Eclipse

The instructions for submitting homework require you to package the class files as an executable JAR file. An executable JAR is a good format also for bringing your solution to class. Here's how you do an executable JAR in Eclipse:

1. First, create a project for the Hello World and import the source as described in Section 1.
Compiling and running the "Hello, World".
2. From the "File" menu -> "Export ..."
3. "Select": JAR file
4. <Next>
5. "JAR Package Specification": Select the files that you will include into the JAR. For our "Hello, World!" example, this is just the Java source file "Hello.java". You don't need to include the "hidden" files and folders (e.g., .settings, .classpath, .project).
6. **Tick the box to "Export generated class files and resources" and "Export java source files and resources" (otherwise we will not be able to see your source files).**
7. Select a suitable destination for the output (**.JAR**) file.
8. You can choose to compress the contents (smaller executable).
9. <Next>
10. "JAR Packaging Options": You should make sure to export files with compile **warnings**. If there are compiler errors, the program won't run in any case (you can select to export those files that have errors, though).
11. <Next>
12. "JAR Manifest Specification": In order to make a self-contained executable JAR, you should include a Manifest file. Eclipse can generate it for you. Select "Generate the manifest file". You don't have to save it to workspace, though.
13. You can **seal** the default package in this JAR, if you want to. Sealing isn't covered on this course, but you can study what it means from the Java Tutorial: <http://java.sun.com/docs/books/tutorial/jar/manifest/sealman.html> (If you really want to.)
14. If the project is not an applet, select the main class of the application. This is the class whose main method will be used as the entry point for the application. Again, Eclipse automatically finds those classes that have the main method, when you click "Browse ...".
15. <Finish>
16. The JAR has been exported! Eclipse also tells you if there were any warnings/errors. Warnings don't usually affect the JAR.

Jar File Import (To test your jar creation)

If you want to import a jar file, you'll want to select your project, right click and choose "Import", but instead choose "Archive file". The options here are very similar to when importing from the file system, since an archive file is really a miniature file system stored in a single file. After you choose the Jar file you want to import from, you'll see a directory structure with checkboxes that behaves just like when importing from the file system. Check the folders or individual files you want to import and select "Finish." Copies of the contents of the Jar file are made, so you can safely manipulate the files and delete them without affecting the source Jar file.

I wanted to import a jar file which was a complete project. To do that, I created an empty project. Before importing (however), I selected the "src" directory and imported to that directory. Otherwise, it yelled about the "selection not containing an applet" or "<project root> [in projectName]" does not exist".

OR: I selected "Use project folder as root for sources and class files" and followed the procedure for importing a jar (above). However, in response to the prompt, I chose NOT to overwrite .classpath and .project.