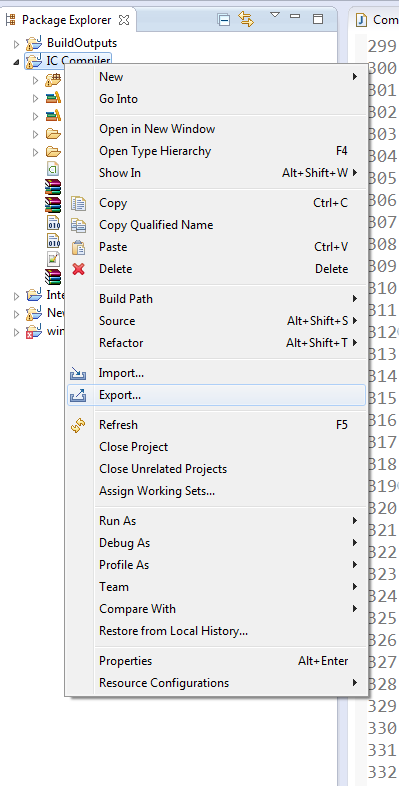
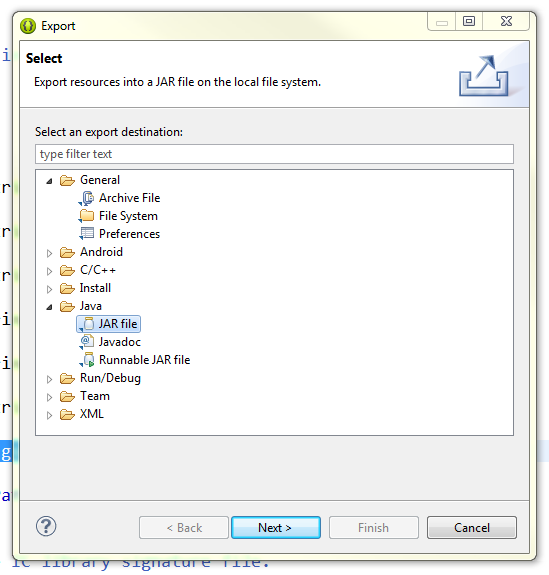
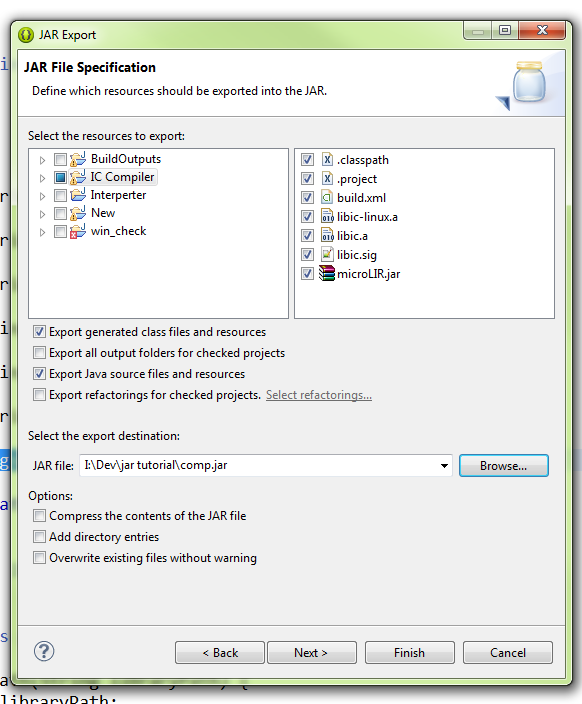
**Step 1: Create jar file**

1. Right click on project->export

2. Choose Java\JAR file

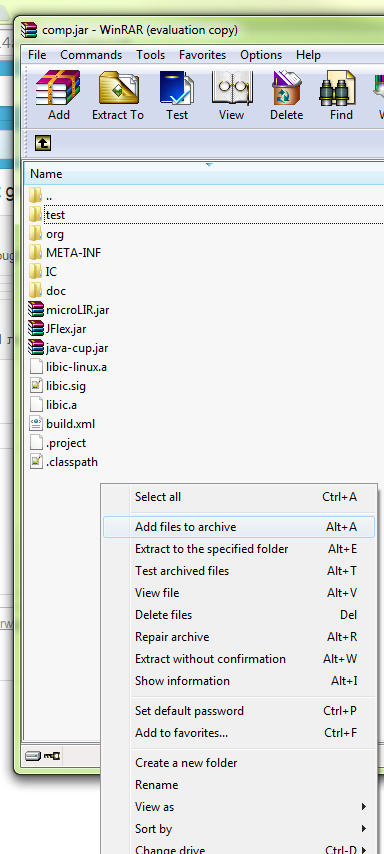
1. Select options: "Export generated class files and resources" and "Export Java source files and resources"



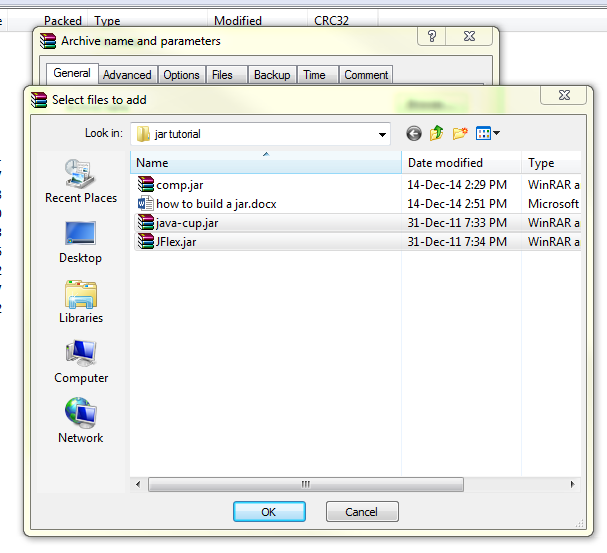
1. Finish

**Step 2: Add relevant library jars to the root of created jar**

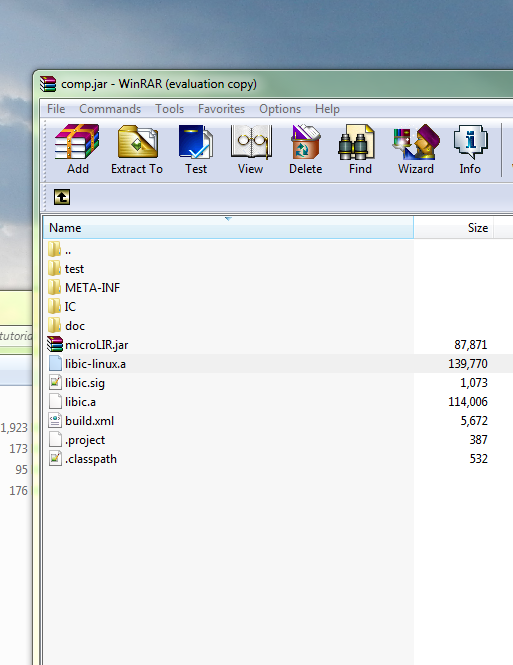
1. **Open the archive->Right click->Add files to archive**

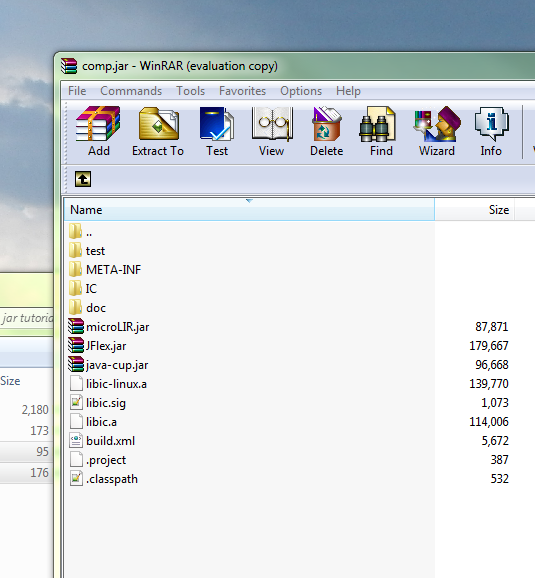
****

1. **Select your jars**



**3. Press OK**

**Before:**

**After:**

**Step 3: Edit the manifest file which is location in META-INF in the jar you just created**

Add the following lines (the <> parts are yours)

Rsrc-Class-Path: ./ **<lib1.jar> <lib2.jar>**

Class-Path: .

Rsrc-Main-Class: **<Main\_Class>**

Main-Class: org.eclipse.jdt.internal.jarinjarloader.JarRsrcLoader

**How to fill in the <>:**

**<lib1.jar> <lib2.jar>:** In Rsrc-Class-Path parameter are all you libraries jar file names (jflex and cup).

In my example it will look like

Rsrc-Class-Path: ./ java-cup.jar JFlex.jar

**These should be the exact libraries added to the jar in the previous step.**

**<Main\_Class>: In** Rsrc-Main-Class is your class which contains the main method.

In my case I put IC.Compiler (the '.' means the file Compiler.class in the directory IC).

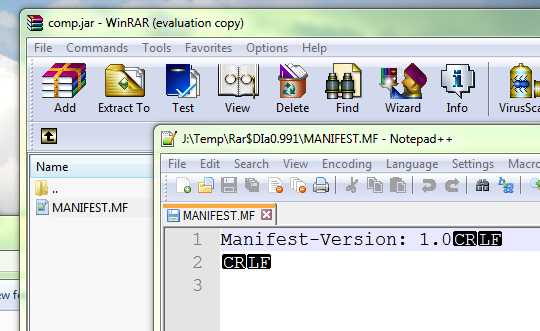
It will look like

Rsrc-Main-Class: IC.Compiler

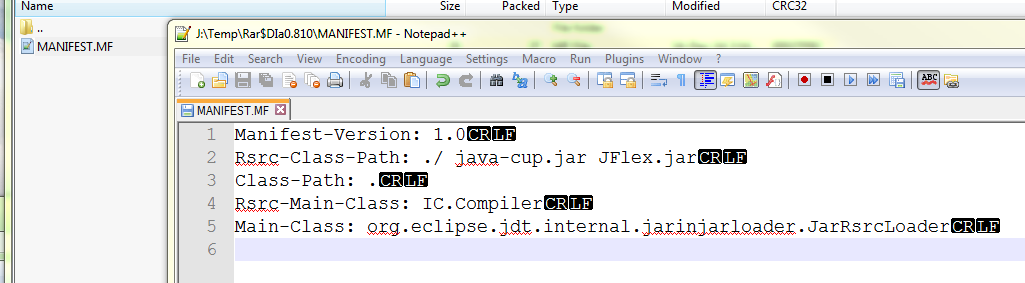
Leave the rest lines as they are.

Make sure the updated manifest is in the jar (archive update is required)

**Before**:



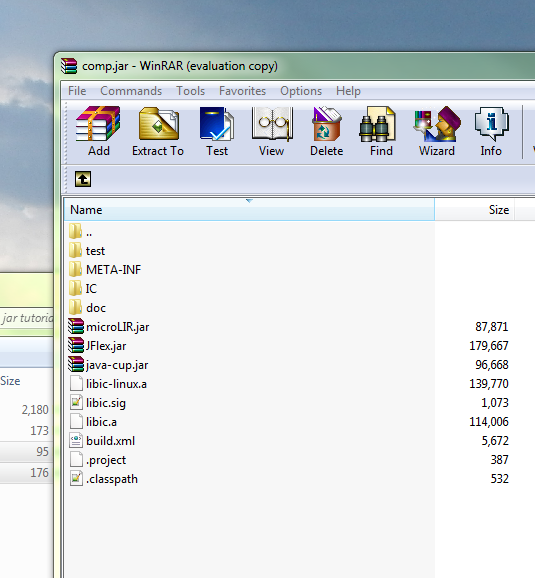
**After:**

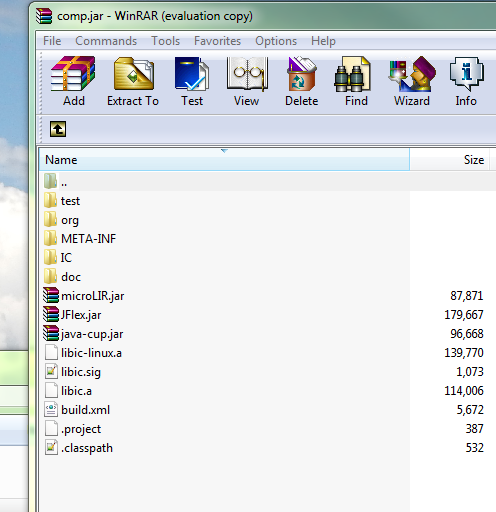


**Step 4: Add org.eclipse.jdt.internal.jarinjarloader.JarRsrcLoader to the root of the jar** (add the directory hierarchy)

You can take it from a runnable jar built by eclipse (will be found in the root of the jar)

or use my attachment: 

**Before:**



**After:**

Test run using "java -jar <jar\_path> <input>”