How can I package my application and execute it in a JAR file?

Question:

How do I package my application with resources in a JAR file?
How do I run a program that is in a JAR file?
How can I create an executable JAR file so that the user can just double-click on it?

Answer:

The Java Archive Utility, JAR for short, is what you use to combine multiple files into a single file in order to simplify distribution and deployment. It replaces the functionality you used to get with Acucorp's cblutil.

You can also use jar as another solution for helping your application to find resource files such as bitmap images.

For example, suppose you had the following file structure:

```
MyprojectobjectPROG1.classMyprojectresourceimage1.bmp
```

and you wanted to package it all into a single file. If your W$BITMAP loads "image1.bmp" then you could put PROG1.class and image1.bmp into a JAR file with:

```
jar cfv myprog.jar -C Myprojectobject PROG1.class -C Myprojectresource image1.bmp
```

or

```
jar cfv myprog.jar -C Myprojectobject . -C Myprojectresource .
```

The dots tell JAR to include all files in the directories.

Then you would execute the program on Windows with:

```
java -cp "myprog.jar;C:VeryantisCOBOL_SDK20xxR*libiscobol.jar" PROG1
```

or

```
set CLASSPATH="myprog.jar;C:VeryantisCOBOL_SDK20xxR*libiscobol.jar" isrun PROG1
```

When you are ready to package your application for distribution you can combine all necessary class files and resources into one JAR file and create what is known as an executable Jar. Then double clicking on the .jar file will execute your application. Or you can execute it with simply "java -jar executable.jar"

To do this you create a manifest file. For example, in the above example, create a file named MANIFEST.MF with the following contents:

```
Main-Class: PROG1Class-Path: iscobol.jar
```

The last line of the file should be a blank line.

Create the executable JAR file with the following command line:

```
jar cmfv MANIFEST.MF myprog.jar -C Myprojectobject . -C Myprojectesource .
```

At run time make sure that iscobol.jar is in the same directory as myprog.jar. Then you can execute your program by double-clicking on it (if your desktop file association is set correctly) or with the following command:

```
java -jar myprog.jar
```

If desired, you can put iscobol.jar and other isCOBOL JAR files that your application uses in a subdirectory. For example if your application needs both iscobol.jar and ctree-rtg.jar then you could put those in a subdirectory named iscobollib. Then you add this subdirectory name to the JAR files in the MANIFEST.MF Class-Path. For example, with the following installation file structure:

```
apps/myapp/myprog.jarapps/myapp/iscobollib/iscobol.jarapps/myapp/iscobollib/coblib.jarY
ou would create the following MANIFEST.MF file:Main-Class: PROG1Class-
Path: iscobollib/iscobol.jar iscobollib/coblib.jar
```

(separate URLs in Class-Path with spaces and end the file with a blank line)

Online URL: https://support.veryant.com/phpkb/article.php?id=54