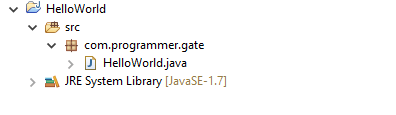
How to create an executable jar in java

A jar file is executable if it contains at least one entry point class with a main method. In this tutorial, we define the different ways to create an executable jar.

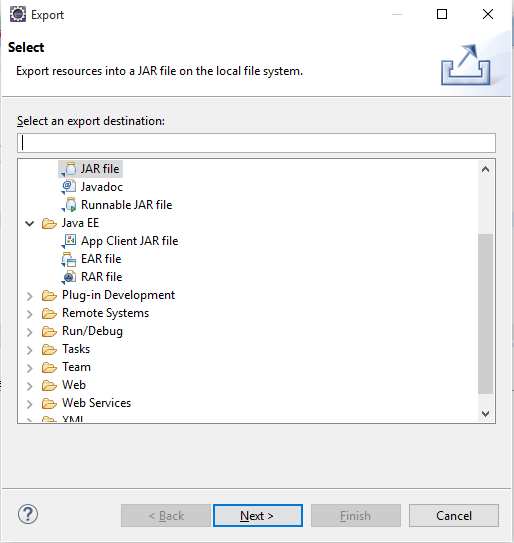
1. Eclipse export tool

Let’s consider a typical hello world example as the following:

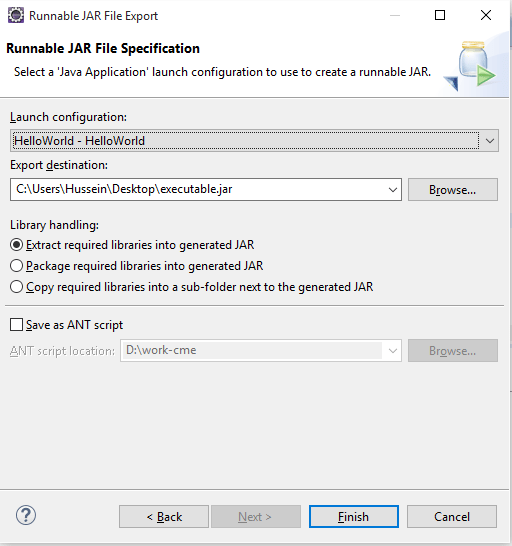


Eclipse provides a tool to generate an executable jar out of any standalone application, to do this:

**right click HelloWorld -> Export,**then you get the following screen:

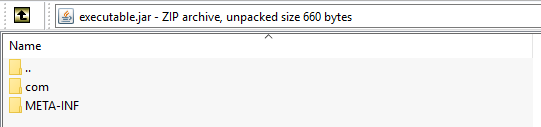


Search for ***Runnable JAR file ,***then click next. In the next page select your main class (*Launch configuration*) and name the jar file as the following:

****

We select our main class ***“HelloWorld”***and click finish, eclipse then generates an executable jar file called ***executable.jar.***

If you open ***executable.jar*** (using *winrar* or *7-zip* tools), you will get something like this:



The jar holds all the compiled classes along with their packages, in addition to a folder named as ***META-INF***which holds a file called ***MANIFEST.MF,***this is the file which makes the jar executable, since it holds the path of the main class of the application, without this file the jar cannot be executed.

2. Command line

In this section, we consider that you have already installed java on your machine, and you already added the java path to your ***PATH*** environment variable.

**2.1 Using manifest file**

Another way to create an executable jar is through using the ***‘jar’***command in the command line*.*The command accepts as input the manifest file as well as the list of compiled classes to be packaged in the jar.

Going back to our example, we move to the ***bin*** directory of our project and create a file called ***manifest.txt***as the following:

|  |  |
| --- | --- |
| 1 | Main-Class: com.programmer.gate.HelloWorld |

**P.S: it’s necessary to insert a new line at the end of the manifest file in order to successfully generate the jar file, otherwise *‘no main manifest attribute’*error is thrown.**

Here, we set the path of the entry point class of the application to *com.programmer.gate.HelloWorld,*note that it’s recommended to create the manifest file inside ***bin***directory so that java can find your main class while execution without the need to explicitly set the classpath.

After creating the manifest file, open cmd as administrator and run the following command:

|  |  |
| --- | --- |
| 1 | jar cfm executable.jar manifest.txt com/programmer/gate/\*.class |

**2.2 Explicitly setting the main class**

You can still generate an executable jar without creating a manifest file, you just set the main class of your application and the jar tool automatically creates the manifest file, the following command does the job:

|  |  |
| --- | --- |
| 1 | jar cfe executable.jar com.programmer.gate.HelloWorld com/programmer/gate/\*.class |

where *com.programmer.gate.HelloWorld*is the main class to be added in the generated ***MANIFEST.MF***

3. Make an existing jar file executable

Normally if you try to execute a non-executable jar file, you get ***‘Failed to load Main-Class’ error***. This message means that your manifest file doesn’t explicitly set the path of the application’s entry point. So in order to make your file executable, you can directly update the manifest file of the jar through the following steps:

* change the extension of your JAR file to ZIP, e.g. just rename ***non-executable.jar*** to ***non-executable.zip***.
* unzip your file.
* edit your manifest file ***MANIFEST.MF*** using a text editor, e.g. notepad++ and set your main class as per section **2.1**
* zip the contents of the unzipped folder.
* rename the zipped file to ***executable.jar***

Here we go, you get an executable jar !!