ROHIT KUMAR
Posted on Jun 29 • Updated on Jul 18

Create an executable JAR file on VS Code n Command line

#java #tutorial #vscode #productivity

So what exactly is JAR?

The **Jar (*Java Archive*)** tool of *JDK* used to package one or more `Java class files` and associated metadata and resources (text, images, etc.) into one file to distribute application software or libraries on the Java platform ~~And~~ it provides the facility to create the executable jar file which calls the main method of the class if you double click it.

Simply speaking, this is a format for archiving data. This is similar to you using WinRAR or WinZIP. You can read Oracle documentation on jar here

In this article,

We will learn how to create an executable jar file, we will take a Java application and explore two ways to run it as a jar, just by double clicking on it. Using **VsCode** & **Command line**

What does it mean when we say that the file is an **executable JAR file**?
Well, when you double click on the JAR file, it automatically calls the main method of the program And if program have JAVA GUI such as `frames`, panel present inside the main method, that would also be executed.

VisualStudioCode

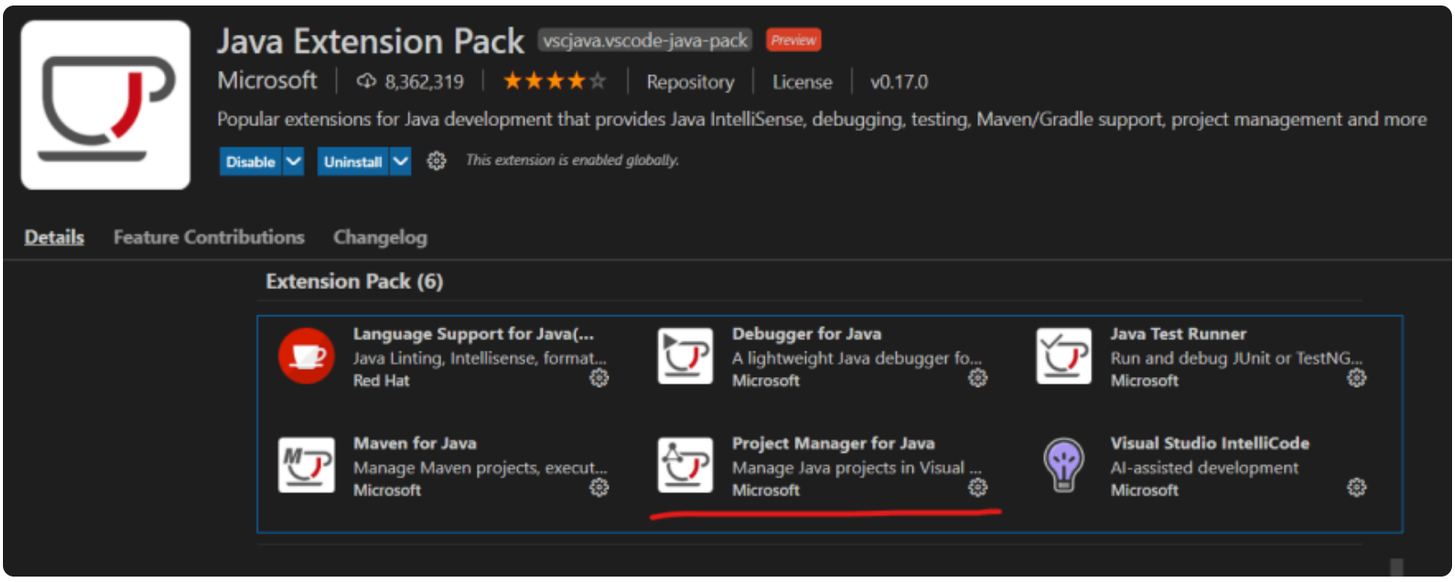
VS Code is a more flexible IDE in today's competition, so its important to know to create jar file here which will get our job done faster with ease.

> If you want to learn more about VsCode and other text editors you can go through this *Blogpost*
>
> 
>
> **Best IDE's and Text Editors!**
> KUMAR HARSH · Jun 24 · 6 min read
> #productivity #programming #codenewbie

NOW, Follow the following steps:-

1.Download the extension

In VsCode marketplace install extension Java Extension Pack which includes set of extensions needed for configuring java enviroment in vscode or you can just download Project Manager for Java extension which is just needed here



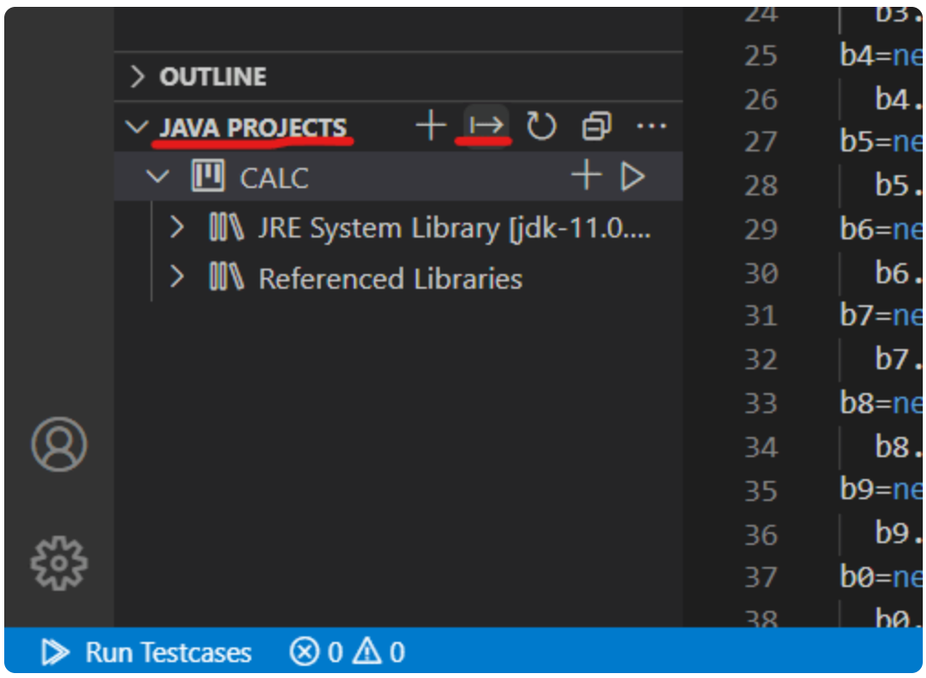
2.Write your Java program

I am using my java code of calculator ,a gui application using AWT
NOTE:-application needs to have a class with a main method. This class provides our entry point into the application

```
Run | Debug
public static void main(String args[]){
    new MyCalc();
    System.out.println("jar file executed");
  }
}
```

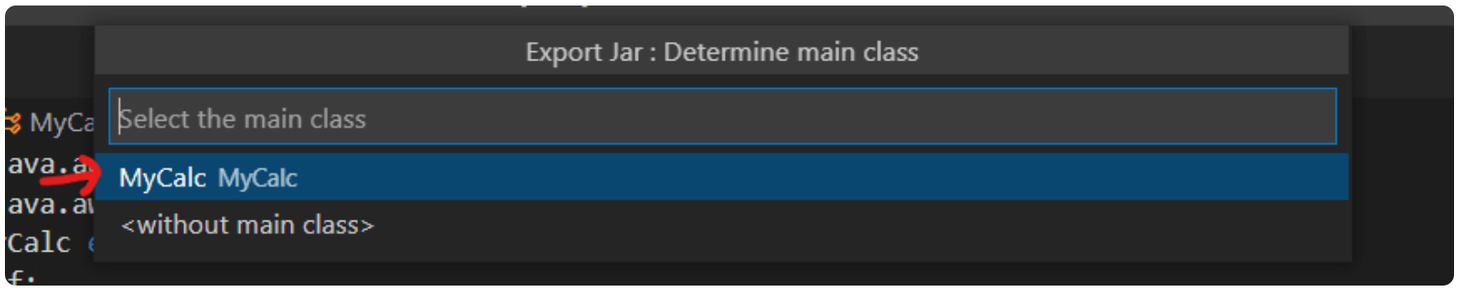
3.Export jar file

In order to compile the code by packing JAR inside Vscode, at bottom left corner you will find and option `JAVA PROJECTS` there you will find and symbol saying `EXPORT JAR`. *Click it*

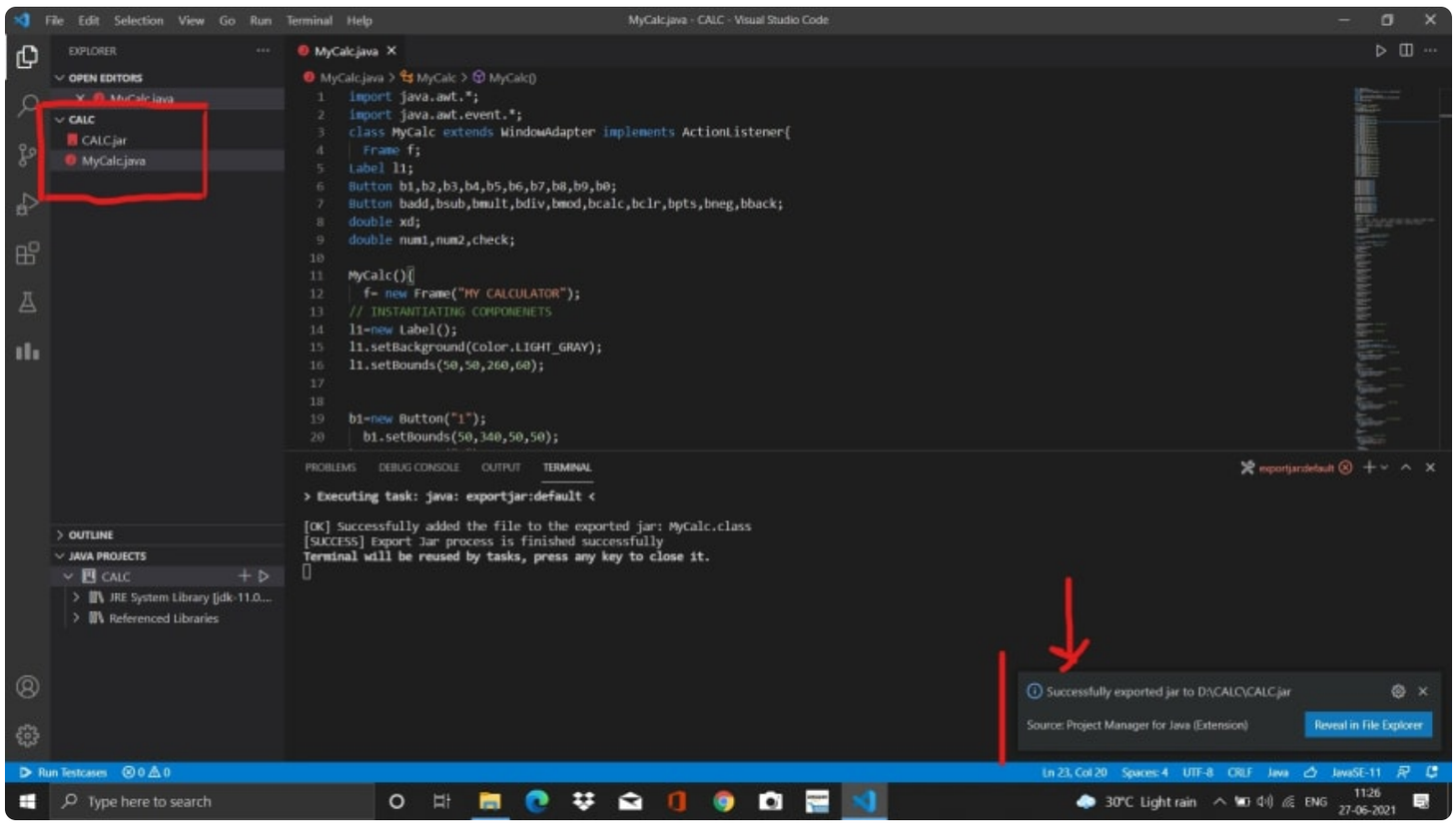


IMP: now its a very important task it will ask you to specify `main class` just provide the main class ,here `MyCalc`
this class provide the entry point so its important to mention.

> NOTE:-Otherwise the jar file will become, normal jarfile not an executable ones



NOW ENJOY your *EXECUTABLE JAR FILE* has been created ,here `CALC.jar` you can go to your directories and just `double click` it or choose and option ,open with `JAVA(TM) Platform SE Binary`



Command line

NOW using command line you should know the jar tool ,provides certain `switches` with which we can create an executable jarfile
some of them are as follows:

> - **c** creates new archive file
> - **v** generates verbose output. It displays the included or extracted resource on the standard output.
> - **m** includes manifest information from the given mf file.
> - **f** specifies the archive file name
> - **x** extracts files from the archive file

Write the java file and then and then follow the following step:-

1.Compile java code

We can do this with `javac` from the command line:

```
javac MyCalc.java
```

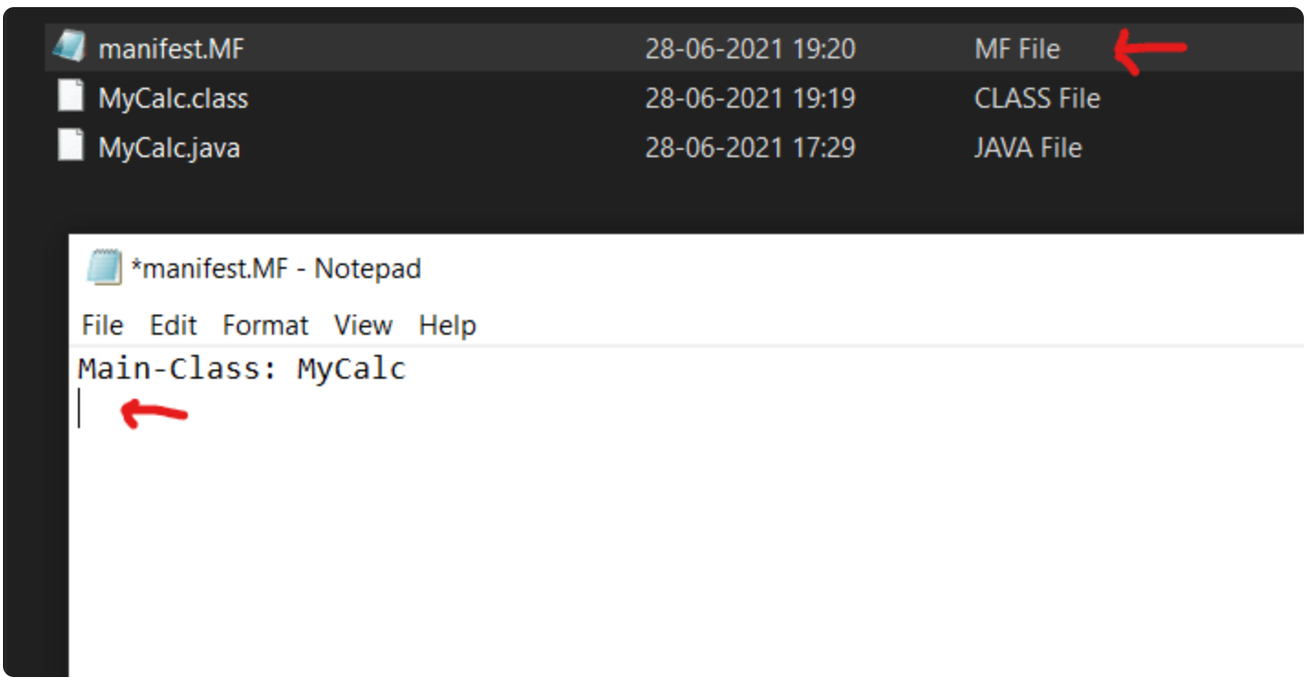
The `javac` command creates `MyCalc.class` in the current directory. If you have multiple java file compile them too, We can now package that into a jar file.

2.Creating manifest file

- A manifest file is essential as it sets an entry points,to our application which main class that we are going to execute to the jarfile.
- So, Create a manifest file with **.MF** extensions in same directories ,so that not needed to set class path explicitly
- You need to write Main-Class, then colon, then space, then classname which you want to make an entry point (here `MyCalc`) then press enter.

> NOTE:It's important that we end our manifest file with a newline.Otherwise ~~*no main manifest attribute*~~ error is thrown.

- now save the .MF file here `manifest.MF`



3.Creating the executable jar

- Open the Command Prompt, write the command using jar tool switches provided
- using `jar` command

```
jar -cvfm <jarfilename.jar> <manifestfile> <classname.class>
```

Command order shouldn't be change

1.`jar` command to create a jar file.
2.switch `c` used to indicates we are creating new file
3.switch `v` generates verbose output information
4.switch `f` tell about the jarfile name we are creating
5.switch `m` it includes the manifest information
Hence, the corresponding filename are also written in the same order ,and if there is multiple class files, then include them too.

- example

```
D:\CALC>jar -cvfm CALC.jar manifest.MF MyCalc.class
added manifest
adding: MyCalc.class(in = 6300) (out= 3152)(deflated 49%)

D:\CALC>
```

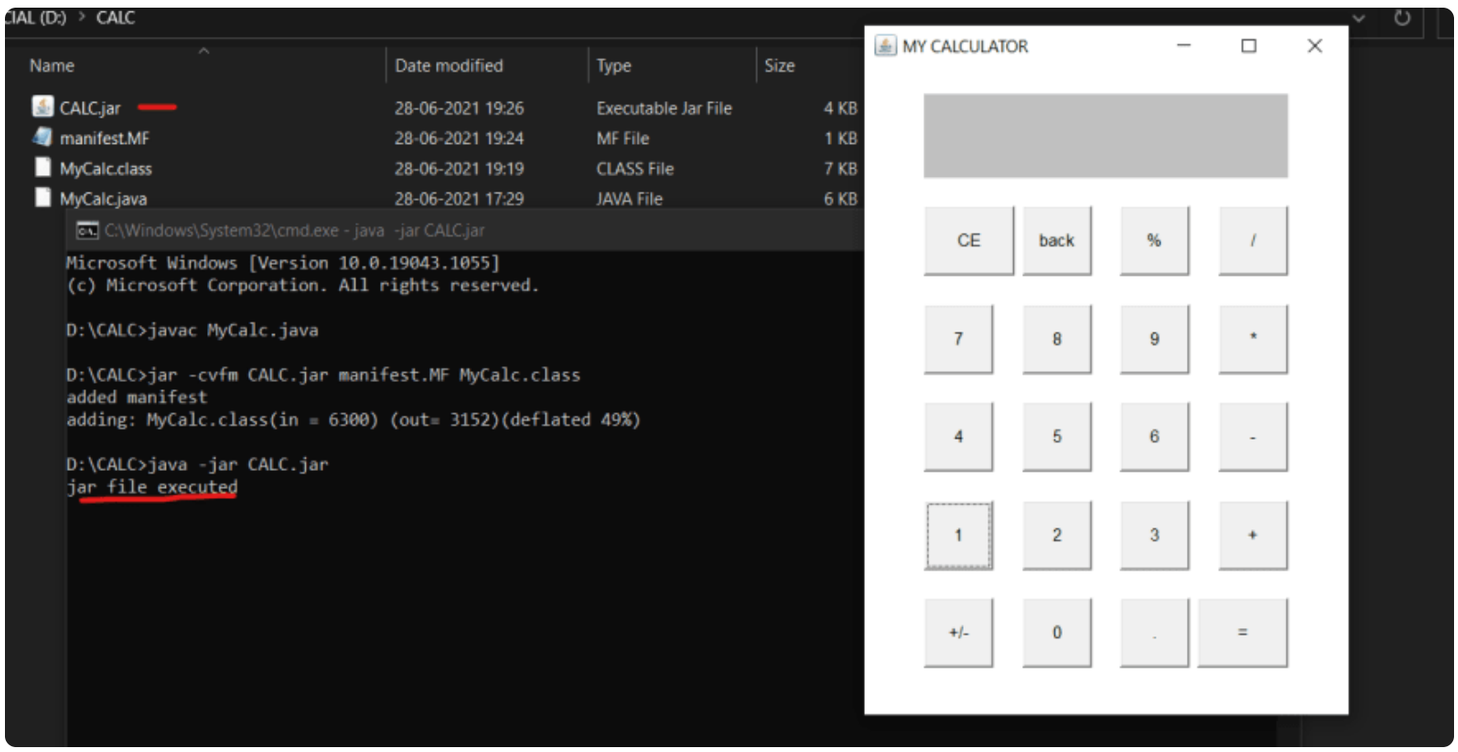
4.Running the Jar

NOW ,we can use the `-jar` option of the java command to run our application since executable

jarfile has been created.

```
java -jar CALC.jar
```

here I named my jarfile as `CALC.jar`
now our application will be executed after this command



OR
You can run this, just by *DOUBLE CLICKING* as our main motive of making our jar file executable.

And that's it. Now you can also run your jarfile with an ease just by clicking it.
If you would like to **download & run** my `CALC.jar` you can find here
as to show how one can easily share ,justifying its properties that import anywhere it is required.
AND

You can check my Blogpost where I had explained how I created this basic calculator using java AWT

Creating a Calculator using Java AWT

ROHIT KUMAR · Jun 18 · 6 min read
#java #programming #tutorial #guiapplication

With this I will end my articles here,
Hope you all find it valuable, and if you have any doubt ,then you ask me just by *commenting below.*

If the program runs, you're done! Happy coding!

Thank You!