

Join the Stack Overflow Community

Stack Overflow is a community of 6.6 million programmers, just like you, helping each other.

Join them; it only takes a minute:

Sign up

How to use classes from .jar files?

I read the Java tutorials on Sun for JAR files, but I still can't find a solution for my problem. I need to use a class from a jar file called jtwt.jar, I downloaded the file, and tried executing it (I found out yesterday that .jar files can be executed by double clicking on them) and Vista gave me an error saying "Failed to load Main-Class Manifest attribute from [path]/jtwt.jar".

The guy who coded the .jar file wants me to import it, but where do I store the .jar file to import it in my code? I tried putting both the .jar file and my .java file in the same directory, didn't work.

The file I'm trying to work for is here: <http://www.winterwell.com/software/jtwt.php>

I'm using JCreator LE.

java jar

edited Aug 6 '15 at 16:36

user155407

asked Jan 20 '09 at 7:33

 [Gaurav Dadhania](#)

1,692 4 27 51

11

Why is there a downvote? In my opinion, it's a good question and there are interesting information in the answers! (+1) – [guerda](#) Jan 20 '09 at 7:50

5 Answers

Not every jar file is executable.

Now, you need to import the classes, which are there under the jar, in your java file. For example,

```
import org.xml.sax.SAXException;
```

If you are working on an IDE, then you should refer its documentation. Or at least specify which one you are using here in this thread. It would definitely enable us to help you further.

And if you are not using any IDE, then please look at [javac -cp](#) option. However, it's much better idea to package your program in a `jar` file, and include all the required `jar` s within that. Then, in order to execute your `jar` , like,

```
java -jar my_program.jar
```

you should have a `META-INF/MANIFEST.MF` file in your `jar` . See [here](#), for how-to.

edited Mar 18 '16 at 10:03

answered Jan 20 '09 at 7:39

 [Adeel Ansari](#)

30.3k 9 67 110

Let's say we need to use the class `Classname` that is contained in the jar file `org.example.jar`

And your source is in the file `mysource.java` Like this:

```
import org.example.Classname;

public class mysource {
```

```

    public static void main(String[] argv) {
        .....
    }
}

```

First, as you see, in your code you have to import the classes. To do that you need `import org.example.Classname;`

Second, when you compile the source, you have to reference the jar file.

Please note the difference in using `:` and `;` while compiling

- If you are under a unix like operating system:

```
javac -cp '.:org.example.jar' mysource.java
```

- If you are under windows:

```
javac -cp .;org.example.jar mysource.java
```

After this, you obtain the bytecode file `mysource.class`

Now you can run this :

- If you are under a unix like operating system:

```
java -cp '.:org.example.jar' mysource
```

- If you are under windows:

```
java -cp .;org.example.jar mysource
```

edited Feb 10 '15 at 17:57



iamcreasy

1,128 3 17 53

answered May 12 '10 at 8:42



GabrieleV

1,403 2 11 8

2 Could you explain this notation a little more? This fixed my problem, but unfortunately I have NO IDEA what the semicolon/colon means in this particular case. – [cemulate](#) Apr 27 '12 at 23:50

1 This semicolon / colon story just costed me the full day. At least I found it now, thanks! – [thelamborghinihistory](#) May 24 '13 at 23:54

1 @ChaseMeadors The [semicolon / colon](#) essentially means "and" here. It's separating `.` from `org.example.jar` . – [duozmo](#) May 26 '14 at 19:04

If you're skimming, heed a distinction from most `*PATH` variables: `.jar` files [need to be individually named](#). – [duozmo](#) May 26 '14 at 19:13

You need to add the jar file in the classpath. To compile your java class:

```
javac -cp .;jwitter.jar MyClass.java
```

To run your code (provided that MyClass contains a main method):

```
java -cp .;jwitter.jar MyClass
```

You can have the jar file anywhere. The above work if the jar file is in the same directory as your java file.

answered Jan 20 '09 at 7:41



kgiannakakis

77.9k 16 126 169

You need to put the `.jar` file into your classpath when compiling/running your code. Then you just use standard imports of the classes in the `.jar`.

answered Jan 20 '09 at 7:38



workmad3

17.3k 3 29 48

As [workmad3](#) says, you need the jar file to be in your classpath. If you're compiling from the commandline, that will mean using the `-classpath` flag. (Avoid the `CLASSPATH` environment variable; it's a pain in the neck IMO.)

If you're using an IDE, please let us know which one and we can help you with the steps specific to that IDE.

answered Jan 20 '09 at 7:40



Jon Skeet

924k51467327613

- classpath env variable is useful if you are using scripts or something like firedaemon to run your app. It's only a pain when you use it globally :) – workmad3 Jan 20 '09 at 7:43
- I still like to see it explicitly. – Jon Skeet Jan 20 '09 at 7:44
- I agree, I like to see it explicitly. – J3r3myK Jan 20 '09 at 8:21

protected by **Community ♦** Jun 24 '15 at 23:50

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 [reputation](#) on this site (the [association bonus](#) [does not count](#)).

Would you like to answer one of these [unanswered questions](#) instead?