



How to use a class from JAR in eclipse



Work at Booking.com

Amsterdam, Netherlands



We offer a competitive salary plus bonus and future career opportunities.



Travel away! We will pay for your commuting costs for all transport.



Enjoy lunch everyday or order take out dinners at our onsite restaurant.

I have two jar files - jar1 and jar2. Both of them are located in C:\Eclipse projects\ and I have added the paths to both of them to the Environment Variable CLASSPATH as follows

```
.;C:\Program Files\Java\jre6\lib\ext\QTJava.zip;C:\Eclipse projects\stdlib.jar;C
```

the ".," at the beginning were there so I left them. Then I added the jars to the project from their location C:\Eclipse projects\ and they showed up as Referenced Libraries. However, when I try to instantiate a class from the jars it does not recognize it. I am also not able to import the jar (import jar1).

After I tried adding a lib folder in the project and I added the jars there. After, I added them as references once again (so not they appear twice in the Referenced Libraries), however, I am still not able to use the inner classes. Any help will be much appreciated.

UPDATE: Something must be wrong on my end. None of the suggestions worked for me. Here is a video with all the steps: screencast.com/t/gC81YzCsLY0e

RESOLUTION In my project I had a package called TestProject and it seems that those jars needed a default package. After deleting the TestProject package and using a defaultPackage everything worked correctly after adding external JARs as explained below.

[java](#) [eclipse](#) [reference](#) [jar](#) [classpath](#)

edited Aug 28 '12 at 16:26

asked Aug 27 '12 at 19:18

[checho](#)
836 ● 6 ● 12

Could you provide some more detail on "it doesn't recognize it"? Is there a java compile error/warning or a runtime error? – [Chris Gerken](#) Aug 27 '12 at 19:20

I have added an image – [checho](#) Aug 27 '12 at 19:22

1 CTRL+Shift+1 on the red wavy line, I think you just miss an import – [RC.](#) Aug 27 '12 at 19:24

Yeah, I can't even see that image. Maybe a screenshot of your entire screen, which is mostly useless, wasn't the best approach. – [Dave Newton](#) Aug 27 '12 at 19:28

I have no suggestion for import and the error is "ClassX cannot be resolved to a type". Here is a better image: screencast.com/t/HuSY2gFYcv – [checho](#) Aug 27 '12 at 19:34

[add a comment](#)

4 Answers

In eclipse, right click on a project->Properties->Java Build Path->Add External JARs (Add JARs if the jar is inside the project's folder) and then choose your jar file. From now you can use the inner classes of the jars you added. Eclipse will import them when you'll start using them.

answered Aug 27 '12 at 19:26

[gfgtmakia](#)
134 ● 3 ● 12

please see the video in the initial question – [checho](#) Aug 28 '12 at 16:15

[add a comment](#)

**Work at Booking.com**

Amsterdam, Netherlands



We offer a competitive salary plus bonus and future career opportunities.



Travel away! We will pay for your commuting costs for all transport.



Enjoy lunch everyday or order take out dinners at our onsite restaurant.

Why don't you use these two JARs—— stdlib-package.jar and algs4-package.jar.

And below the code page(<http://algs4.cs.princeton.edu/code/>)

Q. If I use a named package to structure my code, the compiler can no longer access the libraries in stdlib.jar or algs4.jar. Why not?

A. The libraries in stdlib.jar and algs4.jar are in the "default" package. In Java, you can't access classes in the default package from a named package. If you need to use our libraries with a named package, you can use these package versions: stdlib-package.jar and algs4-package.jar.

Warning: if you are taking Princeton COS 226 or Coursera, Algorithms, Part I or II, you must use the default package version of our libraries to facilitate grading.

Showing my test success:

The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays the project structure, including 'Chapter10', 'ProAlgorithms', 'src', 'JRE System Library [JavaSE-1.7]', and 'Referenced Libraries'. The 'Referenced Libraries' section shows 'algs4-package.jar' and 'stdlib-package.jar' both located at 'H:\ALC'. The main editor shows the code for 'TestArray.java'. The code is as follows:

```
package fcp.ch1.basic;
import edu.princeton.cs.algs4.*;
import edu.princeton.cs.introcs.*;
import java.util.Arrays;
public class TestArray {
    public static void main(String[] args) {
        int a[][] = {
            {1,2,3,0},
            {4,5,6,0},
            {7,8,9,0}
        };
        System.out.println(a.length);

        int a1[] = {1,2,3,4};
        System.out.println("a1[1] = " + a1[1]);
    }
}
```

Below the code editor, there are two user avatars and their statistics: 'Andrew Barber' (edited Jan 12 at 20:08, 25.8k, 10, 45, 89) and 'MorningFung' (answered Jan 12 at 16:02, 76, 2).

I've got the same problem as you today, And no answer from the web can solve it. However, I fixed it at last.

In fact, there is nothing wrong with the setup, it is right to import those jars through "Add External JARs". The real problem is the location/package of you java code. I found that you have to put your .java file in the default package. For example, you will get errors if you put your java code in a package like com.xxx.yyy.ccc, below is an image which shows the right location/package you should use(see WTF.java). After doing that, you program will be able to run.

The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays the project structure, including 'Algorithms', 'JavaSource', '(default package)', 'WTF.java', 'JRE System Library [JavaSE-1.6]', and 'Referenced Libraries'. The 'Referenced Libraries' section shows 'algs4.jar' and 'stdlib.jar' both located at 'D:\Workspace_MyWork\jars'. The main editor shows the code for 'WTF.java'. The code is as follows:

```
public class WTF {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Integer q = 25;
        Integer p = 9;
        Integer m = q%p;
        StdOut.printf("%d mod %d = %d",q,p,m);
    }
}
```

However, that is how i fixed my problem, i'm not sure that could work for everyone..

answered May 23 '13 at 9:18



H Lai
501 ● 2 ● 5

If you have a folder with your JAR files into the project:

1. Right click on the project>Build Path>Configure Build Path;
2. At the tab "Libraries" click on Add JARs, search and select the JARs files you want to use.

If you have yours JAR files any other place outside the project:

1. Right click on the project>Build Path>Configure Build Path;
2. At the tab "Libraries" click on Add External JARs, search and select the JARs files you want to use.

answered Aug 27 '12 at 19:31



Igor de Lima

48 ● 5

please see the video in the initial question – [checho](#) Aug 28 '12 at 16:15

[add a comment](#)

Not the answer you're looking for? Browse other questions tagged [java](#) [eclipse](#)

[reference](#) [jar](#) [classpath](#) or [ask your own question](#).