
Subject: Re: Java Bridge Import
Posted by [humanumbrella](#) on Mon, 23 Jun 2008 14:07:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

Got it!

OK -- on the idljavabrc file, when you include another path - you
can't put a space after the ;

So:

WILL WORK: \$CLASSPATH;D:\IDLResearch\Bridge

WON'T WORK: \$CLASSPATH; D:\IDLResearch\Bridge

Hope this helps someone :P

Cheers

--Justin

Subject: Re: Java Bridge Import
Posted by [Nigel Wade](#) on Mon, 23 Jun 2008 14:13:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

humanumbrella@gmail.com wrote:

> Hey ~~

>

> So, I'm trying to re-use much existing Java code in this java->IDL
> rewrite process. However, it appears that I can only access Java
> classes if they're jar'd up. I need to be able to access class files
> without having to Jar them. (This is a huge application)

No, you don't need to jar them. Have you read the IDL manual about how to access
Java objects? Some of the examples there access individual class files.

>

> Here's what I mean:

> For example, I have a class "HelloWorld.java" and "HellowWorld.class"
> in D:\IDLResearch\Bridge\

>

> This code: javaClass = obj_new('IDLJavaObject\$HELLOWORLD',
> 'HelloWorld.java') returns

>

> "% Class HelloWorld.java not found

> % Execution halted at: IMPORTER 3 D:\IDLResearch\Bridge

> \importer.pro

> % \$MAIN\$ "

>

> My idljavabrc file contains: D:\IDL\Bridge\ and D:\IDLResearch\Bridge

> -- I've tried it with each.
>
> Any thoughts ??
> Thanks in advance!
> --Justin

The problem you have there is that you have not named your class file correctly. The classloader does not want your java source file. Further, the classloader locates classes by package name, and '.' is the separator for packages within the package hierarchy. What the classloader will be looking for in your example is a file called 'HelloWorld.java.class'.

If your java class is not part of a package then you should include the directory in which HelloWorld.class resides in the CLASSPATH and specify the class 'HelloWorld' in the obj_new.

If your class is part of package then you need to include the base directory of package hierarchy in the CLASSPATH and specify the fully qualified class name in obj_new.

--
Nigel Wade

Subject: Re: Java Bridge Import
Posted by [humanumbrella](#) on Mon, 23 Jun 2008 14:41:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jun 23, 10:13 am, Nigel Wade <n...@ion.le.ac.uk> wrote:
> humanumbre...@gmail.com wrote:
>> Hey ~~
>
>> So, I'm trying to re-use much existing Java code in this java->IDL
>> rewrite process. However, it appears that I can only access Java
>> classes if they're jar'd up. I need to be able to access class files
>> without having to Jar them. (This is a huge application)
>
> No, you don't need to jar them. Have you read the IDL manual about how to access
> Java objects? Some of the examples there access individual class files.
>
>
>
>
>
>> Here's what I mean:
>> For example, I have a class "HelloWorld.java" and "HellowWorld.class"
>> in D:\IDLResearch\Bridge\
>

```

>> This code: javaClass = obj_new('IDLJavaObject$HELLOWORLD',
>> 'HelloWorld.java') returns
>
>> "% Class HelloWorld.java not found
>> % Execution halted at: IMPORTER      3 D:\IDLResearch\Bridge
>> \importer.pro
>> %          $MAIN$ "
>
>> My idljavabrc file contains: D:\IDL\Bridge\ and D:\IDLResearch\Bridge
>> -- I've tried it with each.
>
>> Any thoughts ??
>> Thanks in advance!
>> --Justin
>
> The problem you have there is that you have not named your class file correctly.
> The classloader does not want your java source file. Further, the classloader
> locates classes by package name, and '.' is the separator for packages within
> the package hierarchy. What the classloader will be looking for in your example
> is a file called 'HelloWorld\java.class'.
>
> If your java class is not part of a package then you should include the
> directory in which HelloWorld.class resides in the CLASSPATH and specify the
> class 'HelloWorld' in the obj_new.
>
> If your class is part of package then you need to include the base directory of
> package hierarchy in the CLASSPATH and specify the fully qualified class name
> in obj_new.
>
> --
> Nigel Wade

```

Yes, I did have a typo -- but the ultimate problem was the specification of the folder w/ a space.

```

asdf = obj_new('IDLJavaObject$HELLOWORLD', $
  'HelloWorld')

```

with the proper set up in idljavabrc file works.

Thanks!
 --Justin