Thanks, Mike! I got the RSinc example code to run and I wanted to
document the steps I took in case it helps anyone else running IDL
JavaBridge on Mac OS X. (Of course, don't run these steps without
making sure they make sense on your machine....)

All of this is done from Terminal or an xterm in X11.
Move into the directory with the IDL Java configuration files

1) cd /Applications/idl_6.0/external/objbridge/java/
2) ls -a

See that IDL has an original version of the .idljavabrc.orig file.
We're going to duplicate it, and call the duplicate ".idljavabrc". We
need temporary super-user access to make these modifications, so we
use sudo. Then we also change the file permissions to make them
editable to us. I use "vi" in the Terminal, when BBEdit isn't around.
Use whatever works...

3) sudo cp .idljavabrc.orig .idljavabrc (enter password when asked)
4) sudo chmod ogu+rw .idljavabrc   (so I could edit it)
5) vi .idljavabrc   (edit the file)

In the file, I changed this following line to include a .jar my
colleague wrote. Also I made full paths for everything. This line was
changed

6a) JVM Classpath = $CLASSPATH:$JAVABRIDGE_DIR/jbexamples.jar

to (all on one line, no break)

6b) JVM Classpath =
 /Applications/idl_6.0/external/objbridge/java/jbexamples.jar
:/Volumes/Torque/idllib/somegreat.classes.jar

Note that my personal idl library directory lives on a second
partition on my computer called
/Volumes/Toruqe/idllib/   (if that wasn't obvious).

The configuration file (.idljavabrc) asks to set
$IDLJAVAB_LIB_LOCATION to the location of the "libjvm.so" file. On Mac
OS X the file is called "libjvm.dylib" and its location is (for me at
least):
 /System/Library/Frameworks/JavaVM.framework/Versions/1.4.2/L ibraries/libjvm.dylib

There's an explicit note in the file
 /Applications/idl_6.0/external/objbridge/java/README.txt
that states: 'Note: Do NOT set "JVM LibLocation" in the config file.'
That's a bit tricky since it appears that they set the variable right
there in the .idljavabrc file. I decided to setenv the variable in my
personal idl_setup file which gets called every time IDL is run from
the command line. The lines I added are:

7a) echo "Setting up IDLJAVAB_LIB_LOCATION"
7b) setenv IDLJAVAB_LIB_LOCATION
 /System/Library/Frameworks/JavaVM.framework/Versions/1.4.2/L ibraries/libjvm.dylib

I put in the "echo" just so I would see it go by on my screen. I'll
remove that later on.
You can set back the proper permission on the .idljavabrc file. Repeat
(1), then
8) sudo chmod og-w .idljavabrc (enter password again if asked)

The last step was to add IDL's Java examples to my path variable. I do
that within IDL. !path is a single string with directory paths
separated by ":". I append this directory
 "/Applications/idl_6.0/external/objbridge/java/examples".
Within IDL, I combine an array of paths into a single string that I
add to my !path, but I believe you can just do this:

9) !path += ':/Applications/idl_6.0/external/objbridge/java/examples'

Last step. Run IDL and type "hellojava" at the prompt. You should get
 hello IDL (from Java)
as a response. The exmaple files are described in
 /Applications/idl_6.0/external/objbridge/java/examples/READM E.txt

So at this point, I've got RSinc's Java files to work. I'm still
trying to get my colleague's .jar to work. He created it on a PC and
we haven't worked out why it breaks in IDL with a "class not found"
message.

M. Katz

Michael Wallace <mwallace.removethismunge@swri.edu.invalid> wrote in message
news:<104p3h8dm144v81@corp.supernews.com>...
>> Is anyone else using the JavaBridge from IDL on Mac OS X?
>
> I helped a co-worker set up the IDL-Java Bridge on OS X a couple weeks
> ago and it is working for him. It is possible!
>
>> Following the documentation I added this to my .cshrc file

>>   setenv IDLJAVAB_CONFIG /Applications/idl_6.0/external/objbridge/java
>> although I'm not sure it was necessary.
>
> That shouldn't be necessary. Once you place a .idljavabrc file in the
> root of your directory, you should be good to go.
>
>> The documentation calls for a file called .idljavabrc, so I copied
>> their .idljavabrc.orig into a new file with the proper name. Within
>> that file the first configuration is this line.
>>   JVM Classpath = $CLASSPATH:$JAVABRIDGE_DIR/jbexamples.jar
>> It's not clear to me that $JAVABRIDGE_DIR is defined anywhere that I
>> know about. So I added the full path to my .jar files.
>
> Full paths are always best when trying to debug a problem. You can find
> out if $JAVABRIDGE_DIR is defined by issuing the following command:
>
> echo $JAVABRIDGE_DIR
>
> If it is defined, you'll see the value. If not, you'll see a blank line.
>
>> Next, it calls for a path to the libjvm.so file. I couldn't locate
>> that file anywhere on my Mac.
>
> That's because Macs don't have a libjvm.so file. Search for a file in
> your Java directories that starts with "libjvm". I forget what the
> filename extension is for OS X, but it's not ".so" -- the extension for
> libjvm is whatever OS X uses to denote a system library.
>
>> If anyone out there has tackled these issues, I'd love to hear what
>> you did to these files and how you got it working.
>
> If you still have problems, just ask. I should be able to get the
> config files my co-worker is using.
>
> -Mike