Robert Harder
PhD Student

home phd about

Java 6 and librxtxSerial.jnilib on Intel Mac OS X August 18, 2009

Posted by Robert Harder in : Utility , trackback

[UPDATE: This library works with Mac OS X 10.6 Snow Leopard.]

In order to access a serial port in Java, you need the RXTX libraries compiled for your specific hardware. Java uses the Java Native Interface (JNI) to bridge between your platform-independent application code and the hardware-specific serial port drivers.

If you've tried this on an Intel Mac (perhaps to play with a Sun SPOT), you may be disappointed, since software keeps shipping from people that is either PowerPC only or isn't compiled for 64-bit Intel. You've probably seen the `UnsatisfiedLinkError` message. Here you'll find a `librxtxSerial.jnilib` file with support for both 32- and 64-bit PPC and Intel architectures, fitting the bill perfectly for both Java 5 and Java 6 on the Mac.

```
$ file librxtxSerial.jnilib
librxtxSerial.jnilib: Mach-O universal binary with 4 architectures
librxtxSerial.jnilib (for architecture x86_64): Mach-O 64-bit bundle x86_64
librxtxSerial.jnilib (for architecture i386):   Mach-O bundle i386
librxtxSerial.jnilib (for architecture ppc7400): Mach-O bundle ppc
librxtxSerial.jnilib (for architecture ppc64):  Mach-O 64-bit bundle ppc64
```

The Need for the RXTX Library

Java's "write once, run many" theory of operation works because the Java Virtual Machine (JVM), which must be made for each platform, abstracts away the underlying hardware. As a programmer you can draw circles, label buttons, and even play multimedia without specific knowledge of the host operating system.

A computer's serial ports can be abstracted away in the same way, but the standard JVM does not provide a mechanism for this. Sun decided (reasonably, though regrettably, I think) that the serial port would not be a required component for a JVM, and so there are no built-in classes for working with serial ports.

Sun experimented, for a period of time, with a Java Communications API that would be a sort of plugin for working with things like serial ports (and parallel ports!), but the project and its `javax.comm` package died.

Thanks to the team at RXTX.org, we now have a gnu.io package modeled after Sun's javax.comm package that is maintained and works. Thanks!

Why Doesn't It Work for You?

If you're reading this, it might be because you **can't** get it to work on your Mac. Probably you have an Intel Mac and are using Java 6 or later which requires a 64-bit Intel processor. Perhaps you've seen error messages that say thinks like UnsatisfiedLinkError and so forth.

Software talking to the serial port must communicate with the host operating system, and so the underlying native library must be compiled per-platform. Presumably your librxtxSerial.jnilib file is not compiled for your platform. Here's how to find out. Open the Terminal, navigate to the folder with a librxtxSerial.jnilib file, and use the file command. You'll probably see this:

```
$ cd /Users/rob/SunSPOT/sdk/lib
$ file librxtxSerial.jnilib
librxtxSerial.jnilib: Mach-O universal binary with 2 architectures
librxtxSerial.jnilib (for architecture i386):   Mach-O bundle i386
librxtxSerial.jnilib (for architecture ppc7400): Mach-O bundle ppc
```

If you're running Java 5 (which comes in 32- and 64-bit flavors on the Mac) you're OK, but if you're running Java 6, which is 64-bit Intel only, it won't work.

A Library with the Right Architectures

No problem; all you have to do is re-compile RXTX from sources for your platform, right? I wish. I can't even remember all the contortions I went through before I finally got it compiled.

You're welcome to follow the various instructions online for compiling it yourself, but it gave me a lot of grief, so I'm placing a copy on this site (if you trust me not to insert nefarious code). I finally had to patch SerialImpl.c and SerialImpl.h (manually) based on the patch instructions here and blog posting here.

At the end of the day, we have a librxtxSerial.jnilib file that has 32- and 64-bit PPC and Intel architectures.

Find all instances of librxtxSerial.jnilib on your Mac and replace them with the one you downloaded from here (or compiled yourself). Try the command locate librxtxSerial.jnilib in the Terminal to find extra copies hidden in various Java applications.

Comments»

1. Pete – October 12, 2009

THanks man, save me trying out the hours of compiling you went through! I'm wondering, will the .jar from RXTX be the one I need for the jnilib? It looks like it's working so far.

Ta

2. Robert Harder – October 12, 2009

Yes, the standard .jar file from rxtx should be fine. Happy coding to all. -Rob

3. Patrick – November 1, 2009

Thanks for the post. It's a really useful info!!

A quick question, do I need to compile 64-bit .dll if I want to use rxtx with JDK6 in Windows environment?

4. Robert Harder – November 1, 2009

Good question. I suppose if you have 64-bit Windows with a 64-bit Java then you'd need a 64-bit RXTX library. I'm sorry I don't have one compiled for you. -Rob

5. Peter – November 4, 2009

Robert, I had been banging my head against the wall for a couple of days until now. Thanks you so much for working better than Tylenol.

6. andrew – November 25, 2009

thanks for sharing. btw, java comm. is not dead, but they discontinued support for windows and never supported mac.

7. SP – February 3, 2010

Hi
Can you please also post RXTXcomm.jar file?

Thanks

8. Robert Harder – February 4, 2010

Rather than post stale files, and in keeping with good practices, here's the link to the download page, if you're having trouble finding it:

http://rxtx.qbang.org/wiki/index.php/Download

-Rob

9. Ignacio de Córdoba – February 20, 2010

Hello,
just to be sure… which of the 3/4 versions available in the qbang download page is the one you

used for this compile?
Thanks a lot

10. Robert Harder – February 20, 2010

I don't remember, but surely I chose the line that reads, "rxtx 2.1-7r2 (stable) rxtx-2.1-7-bins-r2.zip rxtx-2.1-7r2.zip." -Rob

11. Ignacio de Córdoba – February 21, 2010

Hi again Robert and thanks for your help…
I can only put to work rxtx-2.2pre2 with the built in file (3 architectures including Mach-O 64-bit bundle x86_64). I anyway get a full JVM crash when I am finished and do SerialPort.close().
With rxtx 2.1-7r2 and your jndi file I get a gnu.io.PortInUseException: Unknown Application at gnu.io.CommPortIdentifier.open(CommPortIdentifier.java:354)

Thanks for any directions 😃

12. Robert Harder – February 23, 2010

Ignacio, I'm sorry that I can't think of anything really useful to try, but if you have other versions of java on your Mac, you might try running your code with them just to see if the crash is unique to one version or another. -Rob

13. Stephane D. – March 5, 2010

Thank you a lot for your help!! I can finally do my application work on Max OS 10.6 . Your librxtxSerial.jnilib fix my problem.
Have a good day!

14. Christoph – March 25, 2010

@Ignacio: The problem is with the lock file. See http://atelier.tkrworks.net/doc/about-picnomeserial?lang=en how to fix it

15. Art – April 11, 2010

Worked splendidly for me on Snow Leopard, after I created the /var/lock file directory (didn't work without that change). The RXTX installation page talks about how to create the lock file directory using the fixperm.sh script. That page correctly tells you that you need to modify the path in the script to /var/lock, but the script uses niutil, which apparently isn't part of Snow Leopard anymore. The instructions Christoph pointed to are better.

16. Joe – May 29, 2010

Thank you so much, its always nice to find someone else with the same problem who has managed a solution.

17. Dave – June 3, 2010

Thanks for making this jnilib available. Works great on my Mac. Now if I could just find a solution for the kernel panics caused by the Silabs CP210x Mac virtual com port driver. Anyone else running into that?

18. Dietmar – September 11, 2010

Thanks Robert for your work.

Now also Mac OS 64bit users can use you vna/J.

Have a nice day ahead

Dietmar, DL2SBA

19. Michael Shimniok – September 15, 2010

THANK YOU for your hard work. I really appreciate the time you saved me in recompiling the rxtx jnilib and making it available to the world.

Am on OS X 10.6, was able to get my serial app launched using your jnilib and then fixed the "gnu.io.PortInUseException: Unknown Application" issue with instructions posted in comments above: http://atelier.tkrworks.net/doc/about-picnomeserial?lang=en

-Michael

20. WHY – November 22, 2010

Thank you SO much!! My application finally works with your librxtxSerial.jnilib.

however, still there is a problem. My device is connected to a MACBook thorugh a usb-serial adaptor. The application can only run once correctly, if I colse the appliction, start again, it will not able to access the serial port. I have to unplug and plug again the usb-serial port adaptor. Do you think anything wrong?

21. Robert Harder – November 22, 2010

@WHY That happened to me before, and I'm trying to remember how I resolved it. I think what it came down to was that the JVM wasn't releasing the virtual machine, but I don't remember how that was happening even after I had closed the Java app. I don't remember what I did to fix it, or if it just stopped on its own.

22. Seb – January 26, 2011

Thank you very much for this. I was trying to find out why it was not working on my macbook until I realised I was running 64-bit.

23. Bill Dunbar – May 2, 2011

Mr. Harder:

Thanks a bunch for the modified librxtx.Serial.jnilib file. It solved a problem we were having with MorseKOB on the Mac, a program that some of us old (and new) telegraph operators use to communicate in authentic landline Morse code (url below).

Problem was that using the Options to change either the Mode or Port resulted in the browser crashing.

Thanks again!

24. togarha – May 12, 2011

Hi,

I'm an user of rxtx library, I'm trying to compile the rxtx library for mac osx as you say in your web, but I can't get the same result as you. I can't compile for ppc64. If I put only -arch i386, -arch ppc, -arch x86_64 it doesn't not compile for ppc64, If I try to compile with -arch ppc64 the compilation makes error…

Furthermore, my compilation doesn't work with x86_64 with java6. Your file works fine, but I want this file without lock files, this is why I want to compile a new file…

I hope you can help me…

Regards,

25. Robert Harder – May 12, 2011

I've been meaning to try to recreate my compilation instructions. Maybe I can get around to that. I don't remember for sure, but I think I had to compile PPC separately and merge it with the other architectures. I don't remember if I had to do it on a PPC machine or not, but I'm pretty sure it was pre-Snow Leopard, and I hope I still have the ability to cross compile to PPC.

26. togarha – May 12, 2011

Really I don't need the ppc compilation, I only need x86_64, but my compilation doesn't work in x86_64 (making changes you say)
Thanks

27. Stefan – May 20, 2011

I got an gnu.io.PortInUseException on OS X 10.6.7 (x86_64) Macbook Air and had to run:

```
sudo mkdir /var/lock
sudo chmod a+wrx /var/lock
```

in the terminal to get it working…

Thanks for the effort!