Java on Linux HOWTO

maintained by Eric S. Raymond, ¡esr@snark.thyrsus.com¿

v.2.0, 20 Jan 1996

This document describes how to get started with Java and HotJava under Linux, as either a user or a programmer.

Contents

1 Introduction

This document is a Linux-centric introduction to the world of Java and HotJava. These technologies are rapidly evolving, and we welcome contributions from anywhere.

Places in this document that are in serious need of checking or filling out are bracketed with *** ***. Also see the 9 section below. Please help us improve this HOWTO. Send updates and change requests to ¡esr@snark.thyrsus.com¿.

2 General Questions About Java and HotJava

This section is a general (non-Linux-specific) introduction to Java and HotJava.

2.1 What are are Java and HotJava and why are they interesting?

Java is a network-aware language superficially resembling C++, but much smaller and more compact and cleanly designed. It’s an unlimited-extent language with garbage collection like Lisp, but with static type checking (it's been aptly described as "Smalltalk with sane syntax"). It includes lightweight processes (threads) as a native facility and has powerful network-security features. So far, its major application is the HotJava browser, but it holds considerable promise as a general-purpose application language.

HotJava is a WWW browser written in Java. Its major advance over other browsers is that it knows about a new HTML construct called an APPLET, which is some Java class that executes on the client machine. Thus, WWW documents written with Java in mind can have "live" code objects embedded in them, as opposed to just data.

The ability to safely pass around code objects probably represents the most significant advance in WWW technology since the first release of Mosaic. At minimum, it delivers an extensible Web browser that won't need perpetual upgrading to handle new image formats and tag types.

2.2 Where do Java and HotJava come from? Who can use them?

Java and HotJava were developed at Sun Microsystems by a team headed by James Gosling (well known as the designer of Gosling Emacs and NeWS).

The last time Sun tried to set a major technical standard was NeWS, its Network Window System. Though NeWS was pretty universally conceded to be technically superior to X, X won because its sources were freely redistributable. Sun learned from this mistake, and has made Java/HotJava much more generally available; the sources can be downloaded under a fairly relaxed license (see 3). Sun is encouraging ports to non-Sun environments.

Netscape now interprets Java. Microsoft licensed the technology in early December 1995. So it appears that Java support will probably become universal in 1996.

Java used to be called Oak. HotJava was once known as WebRunner.

2.3 How mature is Java?

On December 12 1995 Sun released the 'Beta 2' version of Java. The Java environment API defining its access to the host OS and windowing system has allegedly been semi-frozen – it may be extended, but won't be incompatibly changed.

The 1.0 version of this FAQ is being issued along with the beta 1.0 Java Developer's Kit (JDK) for Linux, ported by Randy Chapman.

Significant holes are known to exist in the Java security implementation. It is not yet a good idea to use Java for sensitive applications. These problems are expected to be fixed in the production (post-beta) releases.

2.4 Where can I find documentation on Java and HotJava?

Sun maintains an extensive HTML web of Java and HotJava-related documents at `(http://java.sun.com)`. These documents are mirrored at `(http://java.blackdown.com)` and elsewhere; see Sun's list of mirror sites.

2.5 Yes, but where can I find paper documentation?

SunSoft Press's official series of Java books is in the production pipeline at Addison-Wesley. Some details about these can be found in the *comp.lang.java FAQ* (`http://www.city-net.com/ krom/java-faq.html`).

Several Java early-adopters have recommended "Hooked on Java" by the members of the Java team. "Well written, though I'd appreciate something less basic" was one comment.

O'Reilly Associates is working on its own series of Java books in cooperation with Sun. These will include (at least) a "Nutshell Guide To Java", a language reference, a class library reference, and a book on the underlying byte-code virtual machine. (Full-disclosure statement: Your HOWTO editor has been invited by O'Reilly Associates to serve as primary technical reviewer for this series, and is being paid for that work.)

SAMS has a book called "Learning Java in 21 Days". No revieww yet.

One of our contributors, browsing his local Bookstop/Barnes & Noble/Borders, came up with 3 books already out:

- One called 'Java in 60 minutes' that looked like a pretty strict syntactic description (about $20).
- One from SamsNet called just 'Java' (about $20).
- One by Tim Ritchey called 'Java!' that includes a CD-ROM (about $35).

Our informant continues: "I bought the 3rd one (by Tim Ritchey). Paper leadtimes being what they are, the CD-ROM only has alpha stuff, though it mentions that the beta version might be out by the time you read it and suggests java.sun.com as a place to get more recent information. Other than that, it's pretty good, even to pointing out that 'well, it's pointless for me to spam you with pages and pages of API descriptions because 1) it would quadruple the size of the book and 20 they're still ch anging and 2) you can get that info online at ...etc'"

Dave Dittrich writes: I'd say that "Java in 60 minutes" may take you 60 minutes to read, but won't teach you how to program in Java in 60 minutes (surprise, surprise!) It looks to me like "Java in 60 minutes" is one of the many attempts be first to market with a book on a hot topic. It didn't seem to include much more than you can find right now on the Web, including lots of text that looks like it was lifted straight out of the Java Language Specification (http://java.sun.com/JDK-beta2/psfiles/javaspec.ps).

Dave continues: Same thing with "Java!", only the latter goes into much more detail on object oriented programming concepts, etc. Someone pointed me to another text book that is equally as good at covering data/method abstraction and other object oriented programming concepts, which is "Structure and Interpretation of Computer Programs" by Abelson, Sussman and Sussman (MIT Press).

Final caveat: at this point (January 1996), it is probly a good idea to stay away from the Ritchey book or anything else based on the alpha API. The beta API is substantially different, and the soon-to-be-released Java 1.0 will have its own differences.

2.6 Where can I find on-line collections of Java code?

WWW archives of applets are available at the following locations:

- (`http://java.sun.com/`)
- (`http://www.applets.com/`)
- (`http://www.gamelan.com/`)
- (`http://www.javasoft.com/applets/applets.html`)

Most of the applets on these pages come with source code, and programmers are invited to use them.

Pointers to others may be available in the *comp.lang.java FAQ* (`http://www.city-net.com/ krom/java-faq.html`).

3 Java as Freely Redistributable Software

Many Linux programmers are attached to producing freely redistributable software (FRS), and try to avoid committing a lot of time to tools for which sources are not generally available. In this section, we discuss Sun's and Java's relationship with the FRS world.

(Warning: I am not a lawyer. I am neither employed by, nor an agent of, nor a stockholder in, Sun Microsystems. This section is based on my interpretation of the current copyright law and the Sun licensing language. Treat this as an introduction; the Sun licensing page, (`http://www.blackdown.org/Java/licensing.html`) is definitive. If you are in serious doubt about what it means, consult an attorney.)

3.1 Are Java programs and applets freely redistributable?

They are if you write them and choose to make them FRS through some mechanism (such as the GPL, or a BSD-style license, or declaring them public domain).

3.2 What can I do with Java implementation binaries?

The Sun licensing page has this to say:

> The unmodified Java and HotJava binary releases may be redistributed free of charge in both commercial and non-commercial applications.

Also:

> Companies or individuals who wish to create a new port of the Java language have the right under this agreement to post the binaries of that port to the Internet for use by others, as long as the port is free of charge and passes the publicly available test suites. (Test suites will be available sometime in the first half of 1996.) The "diffs" may also be posted on the web as long as the underlying source code is not posted. The Java source code is

So if you've got a working Java or HotJava binary, you can give it to your buddies. Or put it on a CD-ROM. Or do anything except patch it and represent the patched version as Java. (This is reasonable. Sun obviously can't leave itself without recourse against Trojan horses sailing under the Java banner.) Binaries are FRS.

3.3 Which parts of the Java sources are freely redistributable?

According to Sun's licensing page, none of it is. But that sounds harsher than it is. In practice, anyone can get the Java and HotJava sources for educational, porting, and non-commercial purposes by filling out a Web form which obligates them not to redistribute the sources. And the agreement *does* permit redistribution of diffs against the sources.

The only circumstance that requires you to sign a commercial license with Sun and pay them money is if you want to use the sources in a commercial product. In particular, the way the language is written, it's within the letter and spirit of the agreement for you to sign Sun's noncommercial source license, snarf the source, port it, and give away the resulting binaries as completely unencumbered freeware!

Sun says that its primary purpose in keeping as much control as it has is to keep the language from mutating into incompatible dialects. The license language supports this; it seems to have been designed to allow hackers to play for free.

3.4 What is Sun's attitude towards FRS Java implementations and tools?

The Sun licensing page says:

> The specifications for the Java Language and the Java Virtual Machine are OPEN and are copyrighted by Sun Microsystems, Inc.
>
> Reimplementations of the Java Compiler or the Java Runtime Interpreter are permitted without requiring a license from Sun provided such implementations are created directly from the published specifications and without the direct or indirect use of Sun's own implementations or other intellectual property rights, including trademarks.

So if you want to go to the effort of creating a Java clone from the published specifications that is FRS, Sun won't stop you.

Sun people use the term "rogue port" for Java implementations that either (a) are performed outside Sun, or (b) don't rely on Sun-licensed code (sometimes the term seems to mean one thing, sometimes the other). They seem to think some of these already exist, but we don't know where they are, nor if they are FRS. They're pretty relaxed about the situation.

Sun has a Java validation suite. They have said they'll certify any Java port that passes it. (Whether this validation will cost money is unknown, but Sun says in writing that the suite wil be "publicly available" in 1996, which at least hints that it will not.) They're prepared to certify rogue ports, though this apparently hasn't happened yet.

3.5 Is anyone cloning Java in freely redistributable source?

There is a clone of the Java compiler in early development. It's called guavac. You can find more information at `(http://http.cs.berkeley.edu/ engberg/guavac)`

There is a rumor afloat that Cygnus Software's GROW project has plans for a Java byte code interpreter to be issued under GPL. However, they haven't yet responded to a query about this. You can get details on the GROW project at `(http://www.cygnus.com/tiemann/grow/)`.

Erik Troan of Red Hat is attempting to put together a development group to do the rest of the job. Right now it looks like Erik will do the class library and your humble editor will do the byte-code interpreter (if Cygnus hasn't gotten there first). This project has been tentatively named "Mr. Coffee". Your editor would now prefer to call it "joe" (for Java Open to Everyone) but that name is taken.

3.6 Are there any FRS Java Tools?

There is a GNU Emacs mode for editing Java. You can fetch it from `(ftp://java.sun.com/pub/java/contrib/emacs)`. Note: this mode assumes you're using c++-mode.el, and won't work with the cc-mode.el distributed with Emacs 19.

4 Java On Linux Questions

Here you can learn the nuts and bolts of getting Java running on your Linux.

4.1 Are Java and HotJava included in any of the Linux distributions?

No, not yet, but expect it any week now.

4.2 How can I get the latest Java distribution for Linux?

Look in `(ftp://java.blackdown.org/pub/Java/linux)`. The latest Java-for-Linux can be downloaded from there. You should browse the `(ftp://java.blackdown.org/pub/Java/linux/README)` first.

The files you'll need are linux.jdk.common.tar.gz and one of either linux.jdk.x86-static-motif-bin.tar.gz ot linux.jdk.x86-shared-motif-bin.tar.gz, depending on whether you have Motif shared libraries on your system.

This port (from the Sun sources by Randy Chapman) is the one that's referred to (as the JDK) elsewhere in this HOWTO.

4.3 What environment will I need to run Java?

For starters, you need an ELF-based Linux. There is no a.out support, and at the speed the Linux world is switching over to ELF there is not likely to be any in the future.

You need a 1.2.13 or later kernel. Kernels 1.2.12 and older seem to have a bug in getcwd(3) that tanks bin/javac because it doesn't check the getcwd(3) return code.

You'll need these pieces:

- libc.so.5 =¿ /lib/libc.so.5.2.16
  (The standard C library)
- libX11.so.6 =¿ /usr/X11/lib/libX11.so.6.0
  (The base X11R6 library)
- libXt.so.6 =¿ /usr/X11/lib/libXt.so.6.0
  (The Athena toolkit library)
- libXext.so.6 =¿ /usr/X11/lib/libXext.so.6.0
  (The X extensions library)
- libXpm.so.4 =¿ /usr/X11/lib/libXpm.so.4.3
  (The X library for pixmap handling.)
- libdl.so.1 =¿ /lib/libdl.so.1.7.9
  (Linux dynamic-loader support)

The C and X support libraries may be in your Linux already.

If you don't already have it, get libc.5.2.16.bin.tar.gz from `(ftp://tsx-11.mit.edu/pub/linux/packages/GCC/)`. Uncompress and untar and copy the files in the lib subdirectory of the top level of the hierarchy to /lib.

If you don't have a current version of ld.so (or libdl.so.1.7.*) you will need to update. ld.so.1.7.9 and .11 will work; .10 and other early versions will not. Get ld-so.1.7.11.tar.gz from `(ftp://ftp.ods.com/linux/)`; Uncompress and untar it and run *instldso.sh* which is in the top level of the resulting hierarchy.

You can get an Xpm library that will work from `(ftp://sunsite.unc.edu/pub/Linux/libs/X/libXpm.3.4f-ELF.tar.gz)`.

4.4 Do I need Netscape? Can I use Netscape?

Yes, you need Netscape. HotJava isn't yet available for the beta release. Randy Chapman says:

> Sun massively changed the awt interface when they shipped the beta JDK and have not yet converted HotJava to it. They have promised they will, and it should work great with linux-jdk when they release it.

It's unknown when this will happen.

Netscape versions starting from 2.0b3 is Java-aware. It is available at ftp1 − 7.netscape.com (i.e. at ftp1.netscape.com, ftp2.netscape.com,... etc.) Version 2.0b4 will allow you to read applets from file: URLs, 2.0b3 didn't. This should be a big help for folks with small-memory machines (they don't need an httpd running).

The file to get is /2.0b4/unix/netscape-v20b4-export.i486-unknown-linux.tar.Z.

Note that Netscape 2.0b3 or 4 will run on an a.out system, so it is, in theory, possible to write your java app, give it to someone else to compile (like maybe the *BlackStar Public Compiler* (`http://mars.blackstar.com`)), and see the output on your system. If you want more than that, get an ELF system.

Make sure CLASSPATH is *not* set before running netscape; having it set seems to crash Netscape, and if you follow the directions below it won't need tweaking.

The Linux Netscape port has a few known problems. It doesn't do sound. Only 8-bit displays work; 16bpp displays yield crashes. The 2.0b3 version occasionally yields bus errors when caling the dispose method for frames. While it's pretty robust otherwise, Gamelan's 'Impressionism' applet is known to crash it.

4.5 What mailing lists or newsgroups exist for supporting Java on Linux?

- *java-linux* (Maintained by karl@blackdown.org) Discussions and developments concerning the port of Java to the Linux operating system. Email to java-linux-request@java.blackdown.org with the word 'subscribe' in the subject to be added to the list.
- *java-linux-announce* (Maintained by karl@blackdown.org) Moderated list for announcements concerning the Java-Linux porting projects. Please send e-mail to java-linux-announce-request@java.blackdown.org with the word subscribe in the subject, to be added to the list.
- comp.lang.java Newsgroup for general Java discussion.
- alt.www.hotjava Newsgroup for discussion of the HotJava browser.

5 Building the Linux Java port on stock Linuxes

In this section, we collect recipes sent to us for building Randy Chapman's Java port on various current Linux distributions:

5.1 Slackware distribution ELF kernel 1.2.13.

John Franks <john@math.nwu.edu> writes that he succeeded with the following steps:

- Get linux-x86.jdk.pre2.static-motif.tar.gz from (`ftp://www.blackdown.org/pub/Java/linux/`) and uncompress it and untar it. (The filenames you must fetch have changes for the 1.0 beta JDK.)
- Get libc.5.2.16.bin.tar.gz (binary distribution, not source) from (`ftp://tsx-11.mit.edu/pub/linux/packages/GCC/`) Uncompress and untar and copy the files in the lib subdirectory of the top level of the hierarchy to /lib. Make sure that /lib/libc.so.5 is a symlink to this file.

- Get ld-so.1.7.11.tar.gz from ftp://ftp.ods.com/linux/ Uncompress and untar it and run "instldso.sh" which is in the top level of the resulting hierarchy.

You should now be able to compile and try the "hello world" program and applet from Sun following the instructions at `(http://java.sun.com/progGuide/index.html)`.

5.2 RedHat 2.1/Caldera Preview 2

Steve Greene <sgreene@access.digex.net> reports success doing the following steps. I have edited the recipe slightly, so blame any mistakes on me. The 'JDK' he refers to is the Chapman port of the Java Developer's Kit.

- From a Red Hat mirror site, get the rpm available for ld.so.1.7.11. To find it, check any RedHat mirror site for:

  /pub/mirrors/redhat-2.1/updates/RPMS/ld.so-1.7.11-i386.rpm

  I know it is available at ftp.pht.com, ftp.caldera.com. and is probably at the other RedHat mirrors as well. Grab it, and do 'rpm -Uvh ld.so-1.7.11-1.i386.rpm' on your system.
- Red Hat 2.0 and Caldera Preview 2 users may need the 'which' program (it's called by some of the scripts in the JDK). An rpm file for 'which' is available at the usual Red Hat mirror sites as part of the Red Hat 2.1 distribution files. Get and install which-1.0.i386.rpm. RedHat 2.1 users should already have which on their system or available within their original distribution.
- Get the tar file(s) for the statically-linked JDK. Untar it from a convenient point. I put mine in /usr/local, so the untar creates /usr/local/java/....
- So you can run java as someone other than root (this is a good idea!), do 'chmod 666 /dev/zero'. Red Hat and Caldera have the wrong permissions set on this device out of the box.
- If you haven't already, run ldconfig as root to load the new libraries. If you do "ldconfig -v" you can verify you have all the libraries loaded. (At least we don't have to build an ELF system first!)
- Make sure you're logged in as a user (e.g. non-root). Start X- windows, open an X-Term, and try something!

5.3 Unifix

Stefan Middendorf writes: Unifix is a popular German-language Linux distribution. There are a few distributions based on it: Linux Universe, sold in USA, Linux 4U, currently only distributed in Germany and Linux FT in Britain. This recipe is known to work for Unifix 1.7, Linux 4U 1st Edition, Linux Universe 2nd Edition, and Linux FT 1.1.

- Put linux-x86.jdk.pre2.static-motif.tar.gz from `(ftp://www.blackdown.org/pub/Java/linux/)` in /usr/local and uncompress it and untar it.

- Change the first line of the script .java_wrapper from PRG=‘which $0‘ ¿/dev/null 2¿&1 to

  PRG=‘type $0 | cut -d ’ ’ -f 3‘ ¿/dev/null 2¿&1
- Get libc-5.2.18.bin.tar.gz (I succeeded with libc-5.2.16.bin.tar.gz, too), extract and uncompress it in the / drectory.
- Get ld-so.1.7.12.tar.gz from tsx-11.mit.edu (or any mirror site) and uncompress and untar it, e.g. in /usr/local. Change to the resulting directory and run instldso.sh. This will also run ldconfig. Probably newer versions of ld-so will work too.
- Add /usr/local/java/bin to the PATH environment variable.
- Get netscape-v20b4-export.i486-unknown-linux.tar.Z, extract and uncompress it in /usr/local/<yourNetscapeDir>.
- Create a Link in /usr/local/bin to ../<yourNetscapeDir>/netscape.
- Put /usr/local/<yourNetscapeDir>/moz2_0.zip in /usr/local/lib/netscape/.

As an alternative to changing your PATH, Ralf Strobel suggests:

- Copy the scripts .java_wrapper and appletviewer and the links @java @javac, @javadoc, @javap, and @jdb from /usr/local/java/bin to /usr/local/bin and you can use them wherever you are.
- In /java/bin/i586 you can add a script called ‘javadoc‘

  #!/bin/sh ‘dirname $0‘/java sun.tools.javadoc.Main -d <yourhtmldir> $*

  where if you are user jrandom, <yourhtmldir> would be jrandom/.java-html.

6 Viewing Applets

To run a demo without going through the applet viewer, you can enter this command from the top of my Java directory tree (/usr/local/java on most systems):

bin/java sun.applet.AppletViewer demo/<directory_name>/exampleN.html

where <directory name> is the subdirectory off demo, and N is the number of the example file (some directories have more than one).

Steve Greene says: I’ve started the tutorials available from Sun’s java site and the similar one put out by the NTMUG. I’ve discovered some problems with the syntax in Sun’s tutorial, so I’ve been following the NYMUG tutorial instead for now.

The appletviewer expects as an argument, an html file with an APPLET tag inside it.

For example, if your html file Hello.html looks like:

```
<HTML>
<HEAD>
```

```
<TITLE> Hello test program </TITLE>
< /HEAD>
<BODY>
This is an appletviewer test
<APPLET CODE="Hello1.class" WIDTH=150 HEIGHT=25>
< /APPLET>
< /BODY>
< /HTML>
```

(If you see "¡ /" in the above, ignore the space. It's a workaround for a formatter bug.)

Running "appletviewer Hello.html" will show you the applet. One advantage of using the appletviewer is that events sent to the applet (start(), init(), etc) are in compliance with Sun specs as opposed to Netscape 2.0b4 (probably a bug in Netscape).

A disadvantage of using the appletviewer is that it is much more slower than Netscape.

The first time you run Java, a license screen resembling Netscape's will be displayed.

7 Notes on Known Problems

7.1 GNU Make breaks after I install Java

Recent versions of libc fix a bug which masked a bug in GNU Make (the symptom is that make doesn't look in Makefile for rules). The libc 5.2.8 release notes give a patch to make that fixes the problem.

7.2 You get 'dirname: too many arguments' errors

Your CLASSPATH variable is not correctly initialized. In .java_wrapper, there is code akin to the follwing:

```
PRG=`which $0`
J_HOME=`dirname $PRG`/..
```

Unfortunately, linux's standalone which command is hideously broken, and certain shells will set $0 to the full pathname. Randy Chapman says the fix is either to use:

```
 J_HOME=`dirname $0`/..
```

Or, safer:

```
 J_HOME=/usr/local/java
```

An alternate fix from Dave Dittrich is:

```
 PRG=`csh -c "which $0"`
```

And another one from Tim Farnum is to change the PRG=‘which $0‘ line to

```
PRG=\$0
```

Lutz Behnke suggests:

```
PRG=‘type -path $0‘ >/dev/null 2>\&1
```

A similar change also needs to be made to the appletviewer script.

7.3 You get ‘cannot find class java/lang/Thread’ errors

Your CLASSPATH variable is not correctly initialized. See above.

7.4 An error message refers to /dev/zero

Go root and do ‘chmod 666 /dev/zero’.

7.5 SEGFAULT

Occasionally, you may get a screen full of error messages, and the system cheerfully fills up your swap space and locks-up.

You’re probably missing a library someplace. Rerun ldconfig -v and see what’s missing. Perhaps LD_LIBRARY_PATH or CLASS_PATH is not set. Finally, some applets are buggy or lock up the Linux JDK.

(BTW, you can stop the lock-up by having another Xterm open with top running; use top to kill the java process BEFORE it fills up swap and hangs your system!)

Java seems to want lots of resources, so I’d keep the number of running/open apps on my desktop to a minimum. It will load on a 486DX-2-75 with 8 Mbytes RAM and 16 Mbytes swap (it’ll take a minute, though). I was able to get two animation applets running simultaneously (sort of) before my system ran out of swap space and hung.

7.6 bin/java, bin/javac, or bin/appletviewer gives you a help screen

You left out some command-line parameter.

7.7 Applets show in the viewer but not when put on a web server

A common error which causes these results is failing to get the MIME type of the applet correct. Your server should send a header with the applet indicating that its MIME type is ”text/plain”, ”application/octet-stream”, or some other type that doesn’t have a defined special handler on the client side. How you arrange this depends on which server you are using. (John Franks)

It's also been alleged that tinyhttpd, an HTTP server written in Perl, returns a bad content type. Apache, OTOH, is pretty reliable.

7.8 Problem Logging

Joey Oravec tells us that HotJava keeps a log of what it does and any problems in encounters. If you're up to diagnosing something yourself, look at $HOME/.hotjava/weblog in the user's home directory. That file will make it more obvious if you're perhaps missing a library or something.

8 Related Resources

For general information on Java, there is a FAQ maintained on the comp.lang.java newsgroup; it is available at `(http://www.city-net.com/ krom/java-faq.html)`.

There is an older FAQ-style document by Joey Oravec *joey@sun.science.wayne.edu* `(mailto:joey@sun.science.wayne.edu)` which primarily refers to the alpha release. You can find it at `(http://www.science.wayne.edu/ joey/java/linux.html)`. At some point these documents may merge.

Here are some Java-related pages:

- *Karl Asha's General Linux Java Page* `(http://substance.blackdown.org/java-linux.html)` *LinuxJava Mailing List* `(http://homer.ncm.com/java-linux/)` *Netrek for Java* `(http://www.cs.utexas.edu/users/hiep/netjav.html)` *Blue-Skies for Java* `(http://cirrus.sprl.umich.edu/javaweather/)` *Java Hints Page* `(http://www.parnasse.com/java.shtml)`

9 To Be Added...

- More recipes for specific Linux variants.

10 Acknowledgements

Grateful acknowledgement is made to all contributors, including:

- John Franks <john@math.nwu.edu>
- Zachary DeAquila <zachary@zachs.place.org>
- Steve Greene <sgreene@access.digex.net>
- Dave Dittrich <dittrich@cac.washington.edu>
- Dave Flanagan <dave@ora.com>

- Joey Oravec <mailto:joey@sun.science.wayne.edu>
- Adam Smith <aws@cs.brown.edu>
- Joe Buck <jbuck@Synopsys.COM>
- Omar Loggiodice <ologgio@netdepot.com>
- Stefan Middendorf <mdorf@stud.fh-heilbronn.de<

This version incorporates Steve Greene's rumored but never-published Red Hat/Caldera mini-HOWTO. I've also swiped some stuff I consider useful off Joey Oravec's page – special thanks to him.

For other HOWTOs and FAQs I maintain, see my home page at (`http://www.locke.ccil.org/ esr/home.html`).