

# How to Install JSindo for LINUX

Kiyoshi Yagi  
kiyoshi.yagi@riken.jp

Theoretical Molecular Science Laboratory  
RIKEN Pioneering Research Cluster

2018/06/04

# 1. Install Java

Let's check if you have Java installed or not, and the version of Java if you have. In the terminal, type "java -version" and you will see a message like this:

```
>java -version
openjdk version "1.8.0_121"
OpenJDK Runtime Environment (build 1.8.0_121-b13)
OpenJDK 64-Bit Server VM (build 25.121-b13, mixed mode)
```

If your Java is 1.8.xxx (=JDK8), then you can skip the installation and go to Chap. 2.

If your Java is a newer one, 1.9.xxx (=JDK9) and later, it is unfortunately **NOT** compatible with Java3D library, which JSindo use for visualization. You may either uninstall or switch to JDK8 keeping the current ones. Ask google for details.

To install JDK8, type in the terminal,

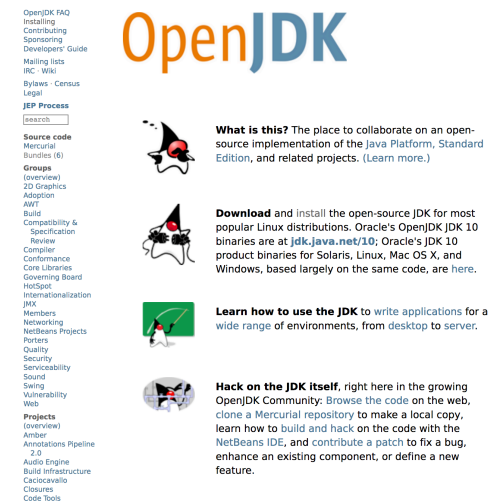
```
> yum install java-1.8.0-openjdk
```

(Fedora)

```
> apt-get install openjdk-8-jre
```

(Debian, Ubuntu)

See the OpenJDK website for further details.



<http://openjdk.java.net/>

Alternately, you may use Oracle Java 1.8.x. It is available from here.

Search “Java SE download” in Google and goto the following website.

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

ORACLE

Menu

Oracle Technology Network / Java / Java SE / Downloads

Overview Downloads Documentation Community Technologies Training

Java SE Downloads

Java Platform (JDK) 10

NetBeans with JDK 8

Java SE 10.0.1

Java SE 10.0.1 is the latest feature release for the Java SE Platform

• Installation Instructions

• Release Notes

• Oracle License

• Java SE Licensing Information User Manual

• Includes Third Party Licenses

• Certified System Configurations

• Readme

Which Java package

• Software Developers: JDK (Java SE Development Kit). For Java Developers. Includes a complete JRE plus tools for developing, debugging, and monitoring Java applications.

• Administrators running applications on a server: Server JRE (Server Java Runtime Environment). For deploying Java applications on servers. Includes tools for JVM monitoring and tools commonly required for server applications, but does not include browser integration (the Java plug-in), auto-update, nor an installer. Learn more

• End user running Java on a desktop: JRE (Java Runtime Environment). Covers most end-users needs. Contains everything required to run Java applications on your system.

Java SE 8u171/8u172

Java SE 8u171 includes important bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release. Java SE 8u172 is a patch-set update, including all of 8u171 plus additional bug fixes (described in the release notes).

Learn more

Installation Instructions

Release Notes

Oracle License

Java SE Licensing Information User Manual

Includes Third Party Licenses

Certified System Configurations

Readme Files

JDK ReadMe

JRE ReadMe

JDK DOWNLOAD

Server JRE DOWNLOAD

JRE DOWNLOAD

accept

Java SE Development Kit 8u172

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

☒ Accept License Agreement ☐ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.99 MB	<a href="#">jdk-8u172-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	74.9 MB	<a href="#">jdk-8u172-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	170.07 MB	<a href="#">jdk-8u172-linux-i586.rpm</a>
Linux x86	184.91 MB	<a href="#">jdk-8u172-linux-i586.tar.gz</a>
Linux x64	167.15 MB	<a href="#">jdk-8u172-linux-x64.rpm</a>
Linux x64	182.08 MB	<a href="#">jdk-8u172-linux-x64.tar.gz</a>
Mac OS X x64	247.87 MB	<a href="#">jdk-8u172-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	140.05 MB	<a href="#">jdk-8u172-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	99.35 MB	<a href="#">jdk-8u172-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	140.63 MB	<a href="#">jdk-8u172-solaris-x64.tar.Z</a>
Solaris x64	97.06 MB	<a href="#">jdk-8u172-solaris-x64.tar.gz</a>
Windows x86	199.11 MB	<a href="#">jdk-8u172-windows-i586.exe</a>
Windows x64	207.3 MB	<a href="#">jdk-8u172-windows-x64.exe</a>

i586 and x64 is 32- and 64-bit, respectively.  
I think you will normally choose 64-bit.

# 2. Download Java3D

JSindo uses Java3D for visualization. A stable version, 1.6.0, is available from JogAmp. Goto <http://jogamp.org>



## Main Page

Welcome to the [JogAmp](#) wiki. It documents JOGL, JOCL and JOAL, the cross-platform bindings to the OpenGL, OpenCL, and OpenAL APIs.

click here

### Getting Started

- [Downloading and installing](#)
- [Versioning and Releases](#)
- [Setting up a JogAmp project in your favorite IDE](#)
- [Source Code Repositories](#)
- [Tracking Reports](#)
- [Build and Test Server](#)

### Community

- [Contribute to JogAmp](#)
- [Build JogAmp](#)
- [Maintainer and Contacts](#)
- [Report a bug](#)
  - [Bugzilla](#)
- [Ask a question in the forum](#)
- [JogAmp IRC](#)

## Downloading and installing JOGL

Before you can build a project that uses JOGL [in your IDE](#) or [on the command line](#), you'll need to download and install the JOGL JAR files and native JARs or native library files (`.dll/.so/.jniLib` files).

You have a choice of JOGL versions to download. The [latest stable version](#) is the safest, but lags behind in features. The [latest automatic build](#) contains all checked-in code, but may be failing some tests.

### Contents [\[hide\]](#)

- 1 [Downloading the latest stable version](#)
  - 1.1 [Using the 7z jogamp-all-platforms archive](#)
- 2 [Downloading the latest aggregated autobuild](#)
- 3 [Downloading the latest automatic build](#)
  - 3.1 [Native JARs vs. native library files](#)
  - 3.2 [Unzipping the files](#)
- 4 [Maven](#)
- 5 [More information](#)

## Downloading the latest stable version

Go to [this page](#) and download the all-in-one 7z archive file:

[jogamp-all-platforms.7z](#)

click here and download  
jogamp-all-platforms.7z

Go back to the Main page and scroll down

## Main Page

Welcome to the [JogAmp](#) wiki. It documents JOGL, JOCL and JOAL, the cross-platform bindings to the OpenGL, OpenCL, and OpenAL APIs.

⋮ ↓ Scroll down

## Related Projects

### Java3D

- Overview
- Downloading and installing
- Tutorial
- API Documentation
- FAQ

### Ji Gong

- Overview
- Motivation
- FAQ

click here

Page Discussion

## Downloading and installing Java3D

### Downloading the latest stable version

Go to [this page](#) and download the 7z archive file:

[jogamp-java3d.7z](#)

Do the same for JogAmp as it is indicated [here](#).

click here and download  
jogamp-java3d.7z

Unarchive the two files you've just downloaded. 7z files can be unarchived by,

```
> 7za x jogamp-all-platforms.7z
> 7za x jogamp-java3d.7z
```

If you don't have the command, install p7zip package,

```
> yum install p7zip
```

(Fedora)

```
> apt-get install p7zip
```

(Debian and Ubuntu)

You will find jar files in jogamp-all-platforms/jar and in jogamp-java3d. The following jar files are needed for JSindo:

```
jogamp-all-platforms/jar/  
  gluegen-rt.jar  
  gluegen.jar  
  gluegen-rt-natives-linux-XXX.jar  
  jogl-all.jar  
  jogl-all-natives-linux-XXX.jar
```

```
jogamp-java3d/  
  j3dcore.jar  
  j3dutils.jar  
  vecmath.jar
```

where **XXX** = amd64 or i586 for 64 or 32-bit, respectively. You can check if your Java is 32- or 64-bit. In the terminal, type “java -version” and you will see a message like this:

```
>java -version  
openjdk version "1.8.0_121"  
OpenJDK Runtime Environment (build 1.8.0_121-b13)  
OpenJDK 64-Bit Server VM (build 25.121-b13, mixed mode)
```

This is an example of 64-bit. If “64-Bit” is absent, then it’s 32-bit.

(\* **XXX** = armv6 and armv6hf are for ARM cpus, which are used for phones.)

# 3. Download JAMA

JAMA is a linear algebra library for JAVA. We use it for matrix multiplications, diagonalization, and so on. It can be downloaded from,

<https://math.nist.gov/javanumerics/jama/>

## JAMA : A Java Matrix Package

[\[ Background \]](#) ..... [\[ The Package \]](#) ..... [\[ Request for Comments \]](#) ..... [\[ Authors \]](#) ..... [\[ Related Links & Libraries \]](#)

### Background

JAMA is a basic linear algebra package for Java. It provides user-level classes for constructing and manipulating real, dense matrices. It is meant to provide sufficient functionality for routine problems, packaged in a way that is natural and understandable to non-experts. It is intended to serve as *the* standard matrix class for Java, and will be proposed as such to the [Java Grande Forum](#) and then to [Sun](#). A straightforward public-domain reference implementation has been developed by the [MathWorks](#) and [NIST](#) as a strawman for such a class. We are releasing this version in order to obtain public comment. There is no guarantee that future versions of JAMA will be compatible with this one.

⋮ ↓ Scroll down

### The Package

Version 1.0.3 (November 9, 2012)

- [Documentation](#)
- [Example](#)
- Source [ [Jama-1.0.3.zip](#) ] [ [Jama-1.0.3.tar.gz](#) ]
- Jar file [ [Jama-1.0.3.jar](#) ]
- [ChangeLog](#)

→ click here and download a jarfile.

# 4. Download and test JSindo

Download JSindo-4.x.jar and sample.tar.gz (or sample.zip) from our website:  
<http://www.riken.jp/TMS2012/tms/en/research/software/sindo/index.html>

Now, create a new folder (`${HOME}/JSindo/jar` in the example below), and copy all jar files in there,

```
> jsindo_jar=${HOME}/JSindo/jar
> mkdir -p $jarfiles

> cp /path/to/jogamp-all-platforms/jar/gluegen.jar      $jsindo_jar
> cp /path/to/jogamp-all-platforms/jar/gluegen-rt.jar  $jsindo_jar
> cp /path/to/jogamp-all-platforms/jar/jogl-all.jar   $jsindo_jar
> cp /path/to/jogamp-all-platforms/jar/gluegen-rt-natives-linux-xxx.jar $jsindo_jar
> cp /path/to/jogamp-all-platforms/jar/jogl-all-natives-linux-xxx.jar  $jsindo_jar

> cp /path/to/jogamp-java3d/j3dutils.jar $jsindo_jar
> cp /path/to/jogamp-java3d/j3dcore.jar  $jsindo_jar
> cp /path/to/jogamp-java3d/vecmath.jar  $jsindo_jar

> cp /path/to/Jama-1.0.3.jar $jsindo_jar
> cp /path/to/JSindo-4.0.jar $jsindo_jar
```

where **xxx** = amd64 or i586 for 64-bit or 32-bit, respectively.



Then, type the following command to invoke JSindo:

```
>java -cp "$jsindo_jar/*" JSindo
```

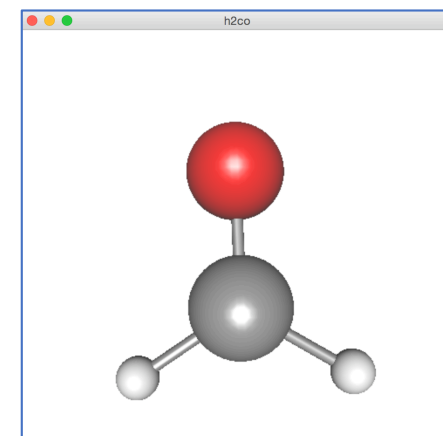
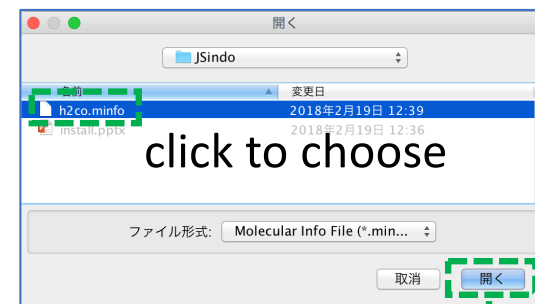
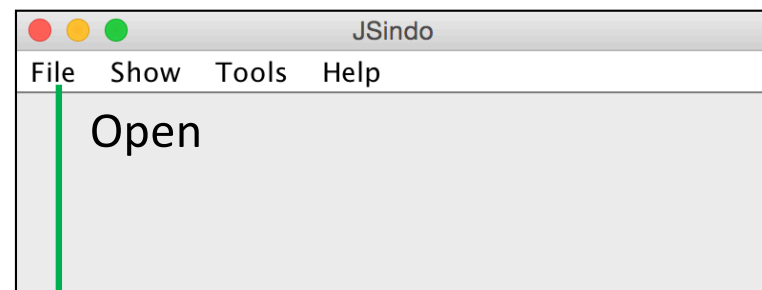
With this command, you should see a control panel of JSindo. If you don't see the panel, you might try to reinstall or update Java. See FAQ for some tips.

Let's open "sample/h2co.minfo", which is included in sample.tar.gz. You can extract the file by,

```
>tar -zxvf sample.tar.gz
```

In JSindo control panel, click File -> Open, choose "h2co.minfo", and click Open. If you see formaldehyde, you're done with the first step!

If this step fails, it is highly likely that JlogAmp/Java3D has a problem. Double check if the right jarfiles (xxx = amd64 or i586) are located in the folder.



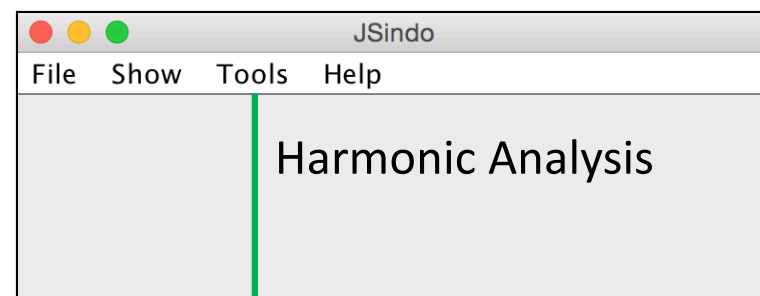
Finally, goto Tools -> Harmonic Analysis. This should create a panel of “Normal modes”.

If you don't see this panel, JAMA isn't working. Check if the jarfile of JAMA is located in the folder.

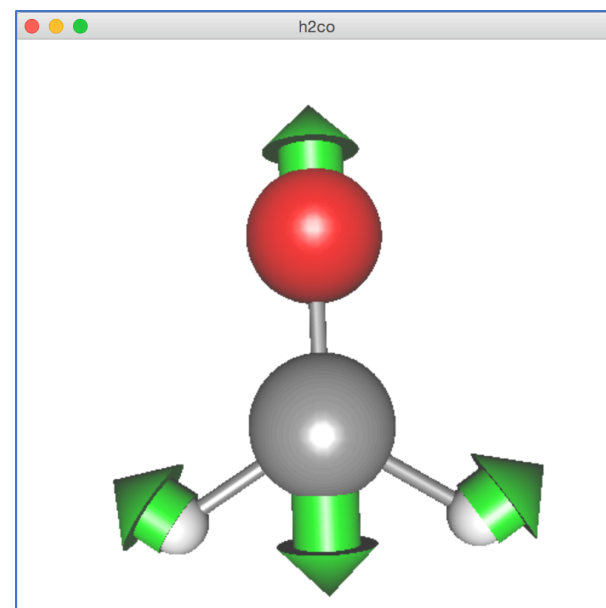
If the panel appears, you're all set! Congratulations!

Check on “show vibrational coordinates”, and choose a mode you want to see. Vibrational motion will be indicated by arrows. You can “Invert the arrows” by a check box, and change the magnitude using a slider.

Thanks for using JSindo!  
Enjoy!

A screenshot of the "Normal modes ( h2co )" window. It contains a table with 4 columns: Mode, Frequency (cm...), Reduced Mass (...), and Intensity (km m...). There are 6 rows of data. Below the table are two checkboxes: "Show vibrational coordinates." and "Invert the arrows.", and a slider bar.

Mode	Frequency (cm...)	Reduced Mass (...)	Intensity (km m...)
1	1196.9147	1.3615	7.0342
2	1266.7685	1.3335	9.3885
3	1540.1545	1.1550	10.7003
4	1752.9374	5.7700	67.7530
5	2973.6886	1.0439	66.6832
6	3047.6560	1.1221	88.4298



# FAQ

## 1. I can not get JSindo start up with the following error:

```
Could not find or load main class JSindo
```

You probably don't have JSindo-4.0.jar in the right folder. Review the jar folder.

## 2. I can not get JSindo start up with the following error:

```
Exception in thread "main" java.awt.HeadlessException
    at java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:204)
    ...
```

Some distributions have a headless version of Java by default. Try the following command to install a standard Java:

```
> yum install java-1.8.0-openjdk
```

(Fedora)

```
> apt-get install openjdk-8-jre
```

(Debian, Ubuntu)

### 3. How do I make the start up command shorter?

I recommend to create an alias. Edit your ~/.bashrc and add the following two lines,

```
> vi ~/.bashrc
jsindo_jar=${HOME}/JSindo/jar
alias jsindo='java -cp "$jsindo_jar/*" JSindo'
```

After source, you can invoke the program by typing “jsindo” in the command.

```
> . .bashrc
> jsindo
```

Alternately, you may want to set CLASSPATH,

```
> export CLASSPATH=${HOME}/JSindo/jar/*
> java JSindo
```

Note, however, that the use of CLASSPATH is discouraged, because it can easily mess things up (the library version, for example).

### 4. I get JSindo start up, but cannot open minfo file with the following error:

```
Exception in thread "AWT-EventQueue-0" java.lang.NoClassDefFoundError:
javax/media/j3d/Bounds
...
```

You probably don't have JogAmp/Java3D jarfiles in the jar folder. Review the jar folder.

5. I get JSindo start up, but cannot open minfo file with the following error:

```
Exception in thread "AWT-EventQueue-0" java.lang.InternalError: XXX0  
profile[1]: GL3bc -> profileImpl GL4bc !!! not mapped  
...
```

This error happened to me when I didn't have the right driver installed for my GPU card (Quadro K600). It was solved after installing a nVidia driver. The following website was useful for me to install a driver in my Fedora Core,

<https://www.if-not-true-then-false.com/2015/fedora-nvidia-guide>