

# Morpheus\_eProbe

## Software to probe your Java environment.

Version 20161228

Copyright © 2016 Dennis E. Slice



## Introduction

Morpheus\_eProbe is a small Java application I built to develop and test a new Java environment probe. This probe basically gathers data about the operating system, Java environment, memory usage and availability and OpenGL capabilities. The application displays this information to the user. My intention is to use the probe, itself, in future software including Morpheus et al.

The program is provided free of charge and with no warranty whatsoever. I also include the source code under the Apache 2.0 open source license (1) and JavaDoc documentation.

(1) Open source licenses are bewildering conglomerations of legalese. I have no idea what these really mean, but my intention is to allow you the right to use the source code for your own private purposes and/or freely distributed software so long as proper credit is given. You can read the Apache license here: <https://www.apache.org/licenses/LICENSE-2.0>

## Installation

To install the program, download the latest .zip file and unzip it. Note that on Windows there may be a distinction of simply looking inside of an archive versus actually extracting it. The program archives are named as:

morpheus\_eProbe\_YYYYMMDD.zip

The year of the revision is substituted for YYYY, then month for MM, and the day for DD, e.g., morpheus\_eProbe\_20161221.zip.

Unzipping this file will produce a directory with the same name as the archive sans “.zip”. The program and associated files are found within that directory.

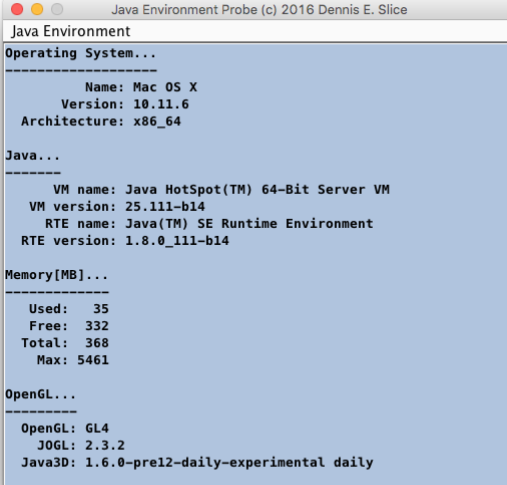
## Program Requirements

It is only assumed that you have an installed a properly configured Java environment to run this program. Mac users should see the “Note to Mac OS X Users” section below.

## Execution

To run the program, navigate into the directory created above and double-click on the file (2),  
`morpheus_eProbe.jar`

The program should execute, and you should see a screen like this containing a concise listing of the available information:



```
Java Environment Probe (c) 2016 Dennis E. Slice
Java Environment
-----
Operating System...
Name: Mac OS X
Version: 10.11.6
Architecture: x86_64
-----
Java...
VM name: Java HotSpot(TM) 64-Bit Server VM
VM version: 25.111-b14
RTE name: Java(TM) SE Runtime Environment
RTE version: 1.8.0_111-b14
-----
Memory[MB]...
Used: 35
Free: 332
Total: 368
Max: 5461
-----
OpenGL...
OpenGL: GL4
JOGL: 2.3.2
Java3D: 1.6.0-pre12-daily-experimental daily
```

As you can see, the program shows the name and version of the detected operating system and the system architecture upon which it is running. It also shows the names and versions of the Java Virtual Machines (VM) and Runtime Environment (RTE).

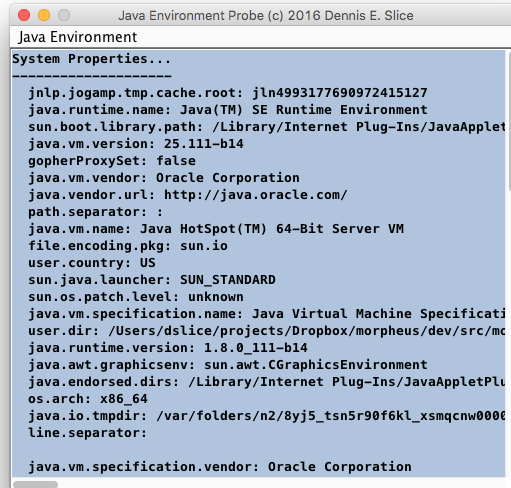
The program also reports on the memory environment of the Java VM.

As distributed, the program will report on the versions of the available OpenGL, JOGL, and Java3D environments. Access to this functionality, which the program does not require, requires the availability of JOGL libraries. Java3D is another library set that, in turn, requires JOGL. And JOGL is required to query the OpenGL capabilities. The necessary libraries are found in the `./lib` subdirectory. However, the program, `morpheus_eProbe.jar`, can run without these libraries (just move the `.jar` file to another directory). If the JOGL and Java3D libraries are available on the system's Java search path, then their versions will be reported. If not, a simple 'not detected' message will be seen.

The "Java Environment" submenu allows the choice of two output formats...

"Concise" – as seen above, this is the basic information about the Java environment.

"Verbose" – is a dump to the screen of everything the probe captures about the environment. Here is a partial look at a verbose listing:



The “Exit” submenu item shuts down the program, as does the window-close button, whatever that may look like on your system.

The `./icons` directory contains some images/icons you can use to pretty-up the program. The image was listed in Google search as “Labeled for reuse with modification”. I cropped and rescaled the original image and converted it to `.png` and `.icns` (Mac icon) formats. The original was found here:

[http://www.accesspaymentsystems.com/wp-content/uploads/2012/05/MagnifyingGlass\\_20747271.jpg](http://www.accesspaymentsystems.com/wp-content/uploads/2012/05/MagnifyingGlass_20747271.jpg)

I am not sure how to change the program icon on Windows or Linux environments, but a `.png` file is provided for this purpose. On an OS X system, you can right-click on the `.jar` file and select the “Get Info” menu choice. When that window opens, simply drag and drop the `.icns` file into the upper, left area of the info window (over the current icon).

(2) Note that the program is setup to only work properly if executed from this directory. I have not provided the `.bat` and `.sh` files, as with Morpheus et al., that would preserve the originating directory needed to find the associated library files. These files are not necessary, however, so if you move just this `.jar` file, the program will operate and report on the system-wide availability of Java-based OpenGL capabilities.

## Source

The program was developed as a NetBeans project. I did not spend too much time investigating how to properly package project source for distribution. Instead, I just zipped the source directory. NetBeans user’s should be able to figure it out. Others can figure out how to make use of the `.java` files. The `.form` file is what NetBeans uses to manage the visual editing of the main program window.

Compiling the program, as is, will require the inclusion in the build of JOGL and Java3D files. The necessary files can be found at <http://jogamp.org/jogl/www/>

The main `Morpheus_eProbe` does not require these, but the two probes, for JOGL and Java3D, do. To divorce `Morpheus_eProbe` from the JOGL and Java3D libraries, the JOGL and Java3D-specific probes accept as a constructor parameter a `Morpheus_eProbe` object. They then do their work, and use the probe’s public functions to set the OpenGL-specific strings. To compile the program without these

libraries, simply comment out the indicated parts in the Morpheus\_eProbe\_startup.java file and don't include the JOGL and Java3D probe classes.

Internally, the results of the probe are returned as a single newline-delimited, formatted string that can, in turn, be sent to any variety of text display devices – edit window, console display, etc. For instance, the return string might look like, where “\t” and “\n” indicate tab and newline characters, respectively:

```
“Operating System...\n-----\n\t          Name: Mac OS X”
```

and when displayed would look like:

```
Operating System...
-----
                Name: Mac OS X
```

Future versions may include query methods to return individual string or numeric information.

## JavaDoc

The source code has been extensively commented and documented using the JavaDoc markup system. This is then used to produce HTML code documentation. You can view these files by double-clicking on the index.html file in the javadoc directory created when the archive was unpacked. You can also access these files by choosing to open a file, navigating to the javadoc subdirectory of the Morpheus\_eProbe program directory, and opening index.html.

## Note to Mac OS X Users

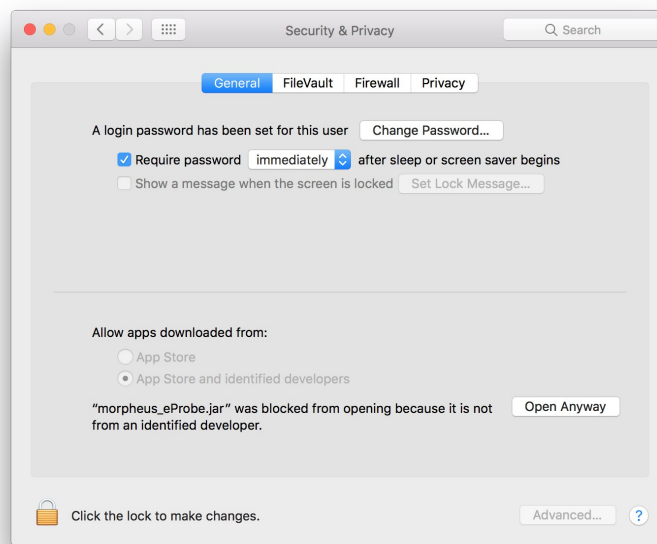
I develop on a Mac, so there may be issues of which I am not aware running the program on Linux and Windows platforms. If you find any issues on these platforms, please let me know at the address given in the “Contact Info” section below.

For the Mac, there may be a couple of issues you need to consider.

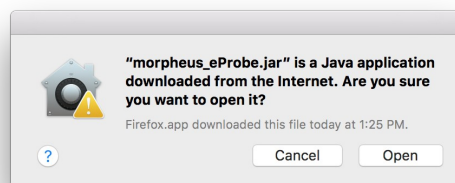
1. You should have your OS X system set up to not run just any old software you download from the web, such as morpheus\_eProbe. In this case, when you try to run morpheus\_eProbe, you will see a message like the following:



If you get this, open the “System Preferences” under the Apple icon in the upper left corner of the screen. Then, click on the “Security & Privacy” icon on the top row. You should see on the panel that opens a message about Morpheus\_eProbe being blocked and a button that says “Open Anyway”:



Clicking on this button will bring up another window confirming your request:



Click “Open”, and the program should run. You will then be able to run the program from then on until you download a new version or re-unzip the old one.

2. The second possible issue you might have is that Apple used to distribute outdated versions of Java3D libraries with its operating system. I cannot find these files any longer on any of my systems

running OS X 10.11.6 or 10.12. I cannot ascertain if they are no longer distributed or if I have somehow permanently deleted them. You should be able to tell if they (or some other inappropriate versions) are there by dragging the morpheus\_eProbe.jar out of its own directory (where the latest library files are stored). If you then run the program and get any message other than “not detected” for the OpenGL information, you might have to move or rename some files – and you should probably do so anyway.

These files are found in the /System/Library/Java/Extensions directory. They include any file with j3d in its name and the file, vecmath.jar. Just create another directory, say hold\_j3d, and move the files into it. You may need administrator privileges to make these changes. There is an app, mac\_osx\_prepare, distributed with Morpheus that does this, and another, mac\_osx\_unprepare, that undoes it. I have not included this app here because I think it is becoming less of an issue for general users with newer operating systems.

## Contact Info

Please address comments, bug reports, ecnomia, etc. to:

[morphlab@sc.fsu.edu](mailto:morphlab@sc.fsu.edu)

And please use the subject line:

Morpheus\_eProbe: \*

Where ‘\*’ is replaced by text indicating the nature of your communication.

## Disclaimer

And again, this program is provided as is and is not guaranteed to do anything whatsoever. Use at your own risk.