TO RUN MESQUITE ON MACOS

PREREQUISITE: JAVA

To start Mesquite, you need Java installed. The version of Mesquite you have here runs on Java 1.8, or higher. We have tested it most under Java 1.8. You can get Java 1.8 from java.com. You can get newer versions of Java from java.oracle.com.

You should NOT try to run Mesquite from the .dmg file that you downloaded. You should copy Mesquite_Folder to your storage drive and run it from there.

STARTING MESQUITE

To start Mesquite, double click on one of these apps or executable files:
Mesquite_Starter_E1.app (or **E2**, **E4**) — These are more likely to work with older versions of macOS or Java; choose 2 or 4 for more memory.
Mesquite_Starter_Flex.app - This is more likely to work with newer systems, but has some drawbacks.
Mesquite_Starter_S.command — This is for power users or if all else fails.

We recommend that you first try starter apps E1, E2, or E4. If those don't work, then try the Flex app. Finally, you can use the S starter to customize how Mesquite starts up.

In general, we expect the E1, E2, and E4 apps to work on older systems and versions of Java. The Flex app is more likely to work on newer systems, but it has some drawbacks.

NEED MORE MEMORY?

If you are using Mesquite for large data files you may need to request more memory. You can do this by choosing the E2 or E4 starters instead of the E1 starter. If you are using the Flex or X starters, you can set the memory requested by editing the file Mesquite-MemoryToUse.txt that is within the folder RunningOptionsForFlex. The file has only a single thing in it, a number, specifying the number of gigabytes of running memory requested. Set it to 4 or higher for larger datasets.

LIMITATIONS OF THE FLEX AND S STARTERS

These starters run a script that gets Mesquite started, which then runs as a separate program. This separate program appears on your dock as a generic Java icon (☕), with a black spot beside it. This icon represents the running Mesquite. Thus, if you want to bring all Mesquite windows to the front by clicking on the icon in the dock, you need to click on that generic Java icon, NOT the Mesquite icon itself (🌿).

As well, you won't be able to drop data files onto the running Mesquite to open additional files. You'll need to use the File>Open menu item.

SOME DETAILS, AND CUSTOMIZATION

These starter apps are in the Mesquite_Folder, and should stay in that same place in Mesquite_Folder. These double-clickable apps do not hold all of Mesquite's working code; they just get the code started. The working code is distributed among various files in Mesquite_Folder.

We give you multiple Starter apps because some may not work for you. Recent changes to the macOS and diverse versions of Java make it challenging for us to supply a single version that works everywhere.

If needed, the Flex starter allows you to specify what version of Java is used to run Mesquite, whether special options ("flags") are passed to Java when it starts, and how much memory is requested. You can specify these options in the files in the folder RunningOptionsForFlex.

Finally, if you need to take control of the startup process directly, use the Mesquite_Starter-S.command, which is a script file that starts Mesquite. To use this file effectively, you may need to edit it with a text editor, which would require knowledge of technical details of paths and flags.

UNIDENTIFIED DEVELOPER WARNING OR OTHER SECURITY WARNINGS

If, when you try to start Mesquite, you get a warning about an unidentified developer (most likely if you try to run the S starter), you can bypass this hurdle by going to System Preferences, Security & Privacy, General. After you have tried to run Mesquite, that Security panel should show a message that Mesquite tried to start, and offer a button to run it anyway. Hit this button, and then try again to start Mesquite.

WHY IS THIS SO MESSY?

Transitions in both macOS and in Java have generated diverse operating conditions for Mesquite: the macOS is either 32 bit (before Catalina) or 64 bit (Catalina and later), and the security hurdles have increased with each new version; Java has also increased its security and also split into two streams, the open-license OpenJDK (Java 8/1.8 and below) and the restricted-license newer versions of Java.

This forces us to maintain multiple versions (e.g., what works on Java 16 will not work on Java 8) to handle multiple OSs (Windows, macOS, Linux), multiple versions of OSs, and multiple versions of Java. The OS/Java system changes frequently enough that every year something breaks. The problems are usually nothing to do with the basic functioning of Mesquite as a program for evolutionary biology. It's almost only about getting Mesquite to simply start.

If you are technically inclined, and find a way to build a starter app that bypasses whatever is the latest challenge thrown our way, please share it with the Mesquite community!