

Record of Installation M1 Mac mini

Answers to Mac Install screens:

English (UK)

Country- United Kingdom

Accessibility - Not Now

Data & Privacy - Continue

Migration Assistant - Not Now

Apple ID - Sign In

Terms and Conditions - Tick I have read, Agree

Create a Computer Account - Details, Continue

Find My - Continue

Express Set Up - Continue

Analytics - Continue

Screen Time - Continue

Siri (enabled) - Continue

Improve Siri & Dictation - Share Audio Recordings, Continue

FileVault Disk Encryption - both ticked, Continue

Choose Your Look - Dark, Continue

(Undid disk encryption later to allow remote VNC login after reboot)

After Logging in:

Calendar would like to use your current location - OK

About this mac - 11.4 Serial No MYSERIAL, Software Update

Update Now macOS Big Sur 11.5.2, Agree

Restart

Settings - Prevent from sleeping display off, start up after power failure

All your files in icloud ticked Continue

Downloaded Xcode 12.5.1 from App Store

Open Xcode Agree, To run Intel-based Install Rosetta, Open test project

In terminal xcode-select - -install

Settings - Sharing tick Screen Sharing, Remote login. Added user MYUSER

From code.visualstudio.com downloaded VSCode 1.60.0 Apple Silicon to Applications

My extensions added to VSCode

Installed Parallels Access Desktop Agent and sign in

From Desktop.github.com installed Github Desktop for macOS Apple Silicon

From julialang.org downloaded Julia 1.6.2 (x86) and 1.7.0-beta4

From nodejs.org other downloads nodejs .pkg for 64-bit-ARM64 16.8.0, Run package install from Downloads

Downloaded Wolfram Engine for Developers macOS ARM

In Downloads executed dmg WolframEngine_12.3.1_MAC-ARM.dmg

2 steps drag Wolfram Engine to Applications and Install script from package

In Applications run Wolframscript - enter wolfram login credentials

In Terminal

git clone <https://github.com/WolframResearch/WolframLanguageForJupyter.git>

In /Applications/Utilities copy Terminal and Paste Terminal, Rename one of them Terminal_x86, File get info and tick Open with Rosetta for Terminal_x86

In Terminal_x86:

```
/usr/bin/python3 -m venv as382
```

```
source as382/bin/activate
```

```
pip install --upgrade pip setuptools wheel
```

```
pip install jupyterlab
```

```
pip install numpy sympy matplotlib pandas plotly bokeh scipy seaborn
```

```
statsmodels octave-kernel ipynb pyvista ipyvtklink vispy jupyter_rfb ipywidgets
```

```
watermark scilab-kernel gnuplot-kernel mdanalysis stochastic mdanalysis tests
```

```
nglview (Do them in blocks of 4)
```

```
jupyter-nbextension enable nglview --py --sys-prefix
```

```
python -m gnuplot_kernel install - -user
```

```
In -s /Applications/Julia-1.7.app/Contents/Resources/julia/bin/julia /Users/
```

```
MYUSER/as382/bin/julia
```

```
In -s /Applications/node-v16.8.0/bin/node /Users/MYUSER/as382/bin/node
```

```
julia
```

```
using Pkg
```

```
Pkg.add("Julia")
```

```
Pkg.add("Plots")
```

```
Pkg.add("DifferentialEquations")
```

```
Pkg.add("StaticArrays")
```

```
Pkg.add("BoundaryValueDiffEq")
```

```
Pkg.add("OrdinaryDiffEq")
```

```
Pkg.add("Sundials")
```

```
Pkg.add("SciMLBase")
```

```
Pkg.add("Plotly")
```

```
Pkg.add("PlotlyBase")
```

```
Pkg.add("Pluto")
```

```
Pkg.add("Makie") *
```

```
Pkg.add("AbstractPlotting") *
```

```
Pkg.add("Molly") *
```

```
Pkg.add("DynamicalSystems")*
```

```
Pkg.add("Symbolics")
```

```
Pkg.add("GLMakie") *
```

```
exit()
```

A star * indicates package errors in 1.7 beta kernel (not ARM compatible yet?) Use 1.6.2 kernel which runs with Rosetta

Downloaded trial of Mathematica 12.3.1.0 downloaded, In Downloads open the Mathematica_12.3.1_MAC-ARM.dmg

Licence - Other ways to activate and connect to a network licence server

```
MYINSTITUTIONSERVER Activate
```

```
cd WolframLanguageForJupyter
```

```
export WOLFRAMSCRIPT_KERNELPATH=/Applications/Wolfram\ Engine.app/
```

```
Contents/MacOS/WolframKernel
```

./configure-jupyter.wls add

In Terminal (not _x86):

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

```
echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/MYUSER/.zprofile  
eval "$(/opt/homebrew/bin/brew shellenv)"
```

```
brew install qt
```

```
brew install octave
```

```
Run with octave - -gui
```

```
brew install wxmaxima
```

From scilab.org downloaded Scilab 6.1.1-MacOS 64bits dmg

Installed from dmg with install JRE option

```
In -s /Applications/scilab-branch-6.1/Contents/bin/scilab-cli /Users/MYUSER/  
as382/bin/scilab-cli
```

Ensure iCloud Drive , options has Desktop and Documents folder ticked

```
In -s /Users/MYUSER/Documents/Developer /Users/MYUSER/Developer
```

In Terminal_x86 In as382:

```
export JUPYTER_CONFIG_DIR=~/.as382
```

```
export WOLFRAMSCRIPT_KERNELPATH=/Applications/Wolfram\ Engine.app/  
Contents/MacOS/WolframKernel
```

```
export SCILAB_EXECUTABLE=/Applications/scilab-6.1.1.app/Contents/bin/scilab-  
adv-cli
```

```
jupyter notebook - -generate-config
```

```
jupyter notebook password
```

```
jupyter-lab --ip=MYIP - -port=8888 --no-browser --notebook-dir=~/.Developer/  
solutions
```

From <https://github.com/conda-forge/miniforge> downloaded Miniforge3-MacOSX-arm64

In Terminal (not x86) cd Downloads

```
bash Miniforge3-MacOSX-arm64.sh
```

Quit Terminal and restart

If forget to get script to init then do:

```
/Users/YOURUSER/miniforge3/bin/conda init zsh
```

```
conda configure - -set auto_activate_base false
```

Close terminal and re enter

```
/Users/YOURUSER/miniforge3/bin/conda create -n p396 python=3.9.6
```

```
/Users/YOURUSER/miniforge3/bin/conda activate p396
```

which python shows /Users/YOURUSER/miniforge3/envs/p396/bin/python

```
python -V shows Python 3.9.6
```

```
which pip shows /Users/YOURUSER/miniforge3/envs/p396/bin/pip
```

```
conda install -c conda-forge jupyterlab
```

```
conda install -c conda-forge numpy
```

```
conda install -c conda-forge sympy
```

```
conda install -c conda-forge matplotlib
conda install -c conda-forge pandas
conda install -c conda-forge plotly
conda install -c conda-forge bokeh
conda install -c conda-forge scipy
conda install -c conda-forge seaborn
conda install -c conda-forge ipympl
conda install -c conda-forge pyvista
conda install -c conda-forge ipyvtklink
conda install -c conda-forge vispy
conda install -c conda-forge watermark
conda install -c conda-forge lfortran
conda install -c conda-forge stochastic
conda install -c conda-forge nglview
jupyter-nbextension enable nglview --py --sys-prefix
pip install octave-kernel scilab-kernel gnuplot-kernel mdanalysis mdanalysis-tests
ln -s /Applications/Julia-1.7.app/Contents/Resources/julia/bin/julia /Users/
MYUSER/miniforge3/envs/p396/bin/julia
ln -s /Applications/node-v16.8.0/bin/node /Users/MYUSER/miniforge3/envs/p396/
bin/node
ln -s /Applications/scilab-6.1.1/Contents/bin/scilab-cli /Users/MYUSER/
miniforge3/envs/p396/bin/scilab-cli
In Setting > Security > Data and Privacy gave Terminal Full Disk Access
jupyter labextension list
jupyter kernelspec list
jupyter nbextension list
mv .jupyter miniforge3/envs/p396
export JUPYTER_CONFIG_DIR=~/.miniforge3/envs/p396
export WOLFRAMSCRIPT_KERNELPATH=/Applications/Wolfram\ Engine.app/
Contents/MacOS/WolframKernel
export SCILAB_EXECUTABLE=/Applications/scilab-6.1.1.app/Contents/bin/scilab-
adv-cli
jupyter notebook -generate-config
jupyter notebook password
jupyter-lab build
jupyter-lab -no-browser -ip=MYIP -port=8889 -notebook-dir=~/.Developer/
solutions
Verify working with test notebook Then shutdown
```

In Julia 1.6 , using Pluto; Pluto.run(host="0.0.0.0")

FileVault turned off to allow auto login and remote VNC access on power up.

```
sudo chown -R root:wheel /Library/Preferences/Parallels
```

(Might not have needed - solution is to set Mac to never sleep otherwise Parallels Access cannot run Apps a second time)