DKI FSL Calcs: Setup on New Computer

Carolyn McClaskey

04/04/2025

Use these directions to set up the DKI FA FSL calculation scripts on a new computer. This will step you through installing everything needed, so if you get to a part that isn’t necessary just skip it.

# Requirements:

* A Unix or Linux operating system. This is setup specifically for a WSL2 on a windows
* The admin password to the computer
* Familiarity with terminal commands, basic understanding of bashrc in linux, and very basic competency with a shell text editor
  + On unix/linux the “~” symbol means “home directory”. So if you see “~/.bashrc”, then the full path to that file is “/Users/USERNAME/.bashrc” on a mac or /home/USERNAME/ on a linux

# Recommendations:

* Windows with WSL (Ubuntu)
* Install Visual Studio Code for debugging

# Setup Instructions

## 1. Install WSL2 if you are on a PC

This can be done on a mac, but I originally set this up for us to use on our PCs via WSL. See Microsoft’s directions for how to do this.

## 2. Install FSL

Open a shell/command window/terminal and type the following command into it:

fsl

If you see the following GUI pop up, you have FSL on the machine and can skip subsequent FSL install steps:

A screenshot of a cell phone

Description automatically generated

Otherwise, go to <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FslInstallation> and follow the instructions for your operating system.

To test if you have FSL correctly installed after following those instructions, type fsl in the terminal again and the FSL GUI should open. For troubleshooting, see the FSL website.

## 3. Install python

Python often comes natively on new machines but you may need to download it. Open a shell/command window/terminal, hereafter called terminal, and type:

python

If that doesn’t work, your computer may not have the python alias. Instead try:

python3

If nothing is found, download the latest stable python release from <https://www.python.org/downloads/> and try again.

If a version of python is found, it will start in the terminal. Scan the lines to check the version. Ensure you have a version that is at least 3.11. If it’s a few years old, it’s probably a good idea to download a newer version just in case. To quit python, type quit().

Close the terminal.

## 4. Clone the repo to your computer (or get scripts)

This step clones the repo to your computer so you have access to the scripts. Either go to where you store repos or create a new folder on your WSL machine. Mine is stored at ~/repos/. Change directories until you are in the folder where the repo will be stored, then clone the repo by typing:

git clone

You will need to know the path to this repository directory for subsequents steps, so make a note of it here.

As an example, because mine is stored in ~/repos/, the path to the repository directory for me was ~/repos/dkifa\_fslcalcs\_env.

## 5. Install the virtualenvwrapper python package for path management

These steps are also found in <https://virtualenvwrapper.readthedocs.io/en/latest/install.html> but I will detail them here. If these steps fail, see that webpage for troubleshooting.

Open a new terminal window and type:

pip3 install virtualenvwrapper

If that doesn’t work, your computer may have pip and not pip3. Try:

pip install virtualenvwrapper

Next, print the path to the virtualenvwrapper.sh file by typing:

which virtualenvwrapper.sh

The location to the shell file will print into the terminal window. Make a note of it because you will need it for the next step.

Next you need modify the .bashrc file to do 3 things: (1) define where to store the virtual environments (2) define the default project location, and (3) run the virtualenvwrapper.sh file. Detailed instructions can be found in <https://virtualenvwrapper.readthedocs.io/en/latest/install.html#shell-startup-file> but I will list the minimum necessary steps here. First, open your .bashrc file in a text editor. If you are on a WSL you can open the file in your Windows File Explorer.

Otherwise, open it directly in Linux by typing this in the command line if you use a bash shell:

sudo nano ~/.bashrc

or, for .zshrc if you use a zsh shell instead of bash:

sudo nano ~/.zshrc

Because this uses sudo, the terminal will prompt you for the admin password. Enter it and the text file will open in the terminal window.

Once you’ve opened the text file by any means, add the following lines to the bottom:

export WORKON\_HOME=$HOME/.virtualenvs

export PROJECT\_HOME=$HOME/Devel

source /usr/local/bin/virtualenvwrapper.sh

But replace /usr/local/bin/virtualenvwrapper.sh with the location to the virtualenvwrapper.sh location you looked up above.

Then type ctrl+X to close the nano text editor (if not using Windows). When it ask if you want to save your changes, press Y on the keyboard to say yes. When it asks you to confirm the filename, press enter to save the file under its current name.

Close the terminal window.

CMcC’s notes for the lab:

* *If you add it to the .bashrc file then those commands will only run if you open a normal bash shell. If you add it to the .zshrc file then they will run every time you open a zsh shell.*
* *Your goal is to set it up so that those lines run every time you open the shell you’re going to use, otherwise the computer won’t be able to find the environment*
* *The lines you added must always be at the bottom of the file, so that any modifications to the path will be known to virtualenvwrapper. If it ever breaks in the future, it may be that relevant path changes were inserted below those virtualenvwrapper lines (for example FSL)*
* *If you find that you can’t edit the ~/.zshrc or ~/.basrc file to add those three lines, then you will need to manually run them each time a new shell is opened. The 1st and 3rd are the most important ones to run and the 2nd line can be skipped*

## 6. Create the virtual environment (dkifa\_fslcalcs\_env)

Open a new terminal window and type:

mkvirtualenv dkifa\_fslcalcs\_env

This will create the environment and also activate it. It will print something like this:

A screen shot of a computer

AI-generated content may be incorrect.

When the environment is activated you will see its name in the terminal like in the last line above.

Now cd to the main repository directory. If following my example, your command will look like this:

cd ~/repos/dkifa\_fslcalcs

If you have cloned the repository to a different location on your computer, replace ~/repos/dkifa\_fslcalcs\_env with the path to your custom repository.

Now we need to link the virtual environment that you just created to the current DKI FA FSL project folder. This step requires you to know where the environment is located. If you followed all directions using my example directories and used the line export WORKON\_HOME=$HOME/.virtualenvs then it’s located at ~/.virtualenvs/vista\_env/. If you don’t know where it is you can search in the terminal for it.

Once you know the location, run the following in the terminal, but replace <path/to/dkifa\_fslcalcs\_env/> with the path to the vista\_env that you found above:

setvirtualenvproject <path/to/dkifa\_fslcalcs\_env> $(pwd)

*As an example, my command looked like this when I did it:*

setvirtualenvproject ~/.virtualenvs/dkifa\_fslcalcs\_env $(pwd)

If it works you’ll see this:



Close the shell and reopen a new one. Test that this link worked by running the following:

workon dkifa\_fslcalcs\_env

You should see the vista\_env at the start of the line as before, like this:



Now print the current working directory using this command:

pwd

It should print the DKI\_FA\_FSLcalcs script directory. If these steps don’t work, then the virtualenv didn’t work correctly. See their webpage for troubleshooting.

Close the terminal window.

## 7. Install python packages into the dkifa\_fslcalcs\_env:

Open a fresh terminal window and run the following to activate the dkifa\_fslcalcs\_env:

workon dkifa\_fslcalcs\_env

With the environment activated as shown above, run the follow command in the command line to the required packages into the environment:

pip install -r requirements.txt

You are now ready to run the main scripts.

*Note: you can install any additional packages by typing the “pip install X” into the environment once it’s activated. It’s a good idea to only install what is absolutely necessary though, and to only use this environment for vista processing. Use separate environments for separate projecs.*