

A Comprehensive Guide on Installing the HEP Trifecta on macOS

Hancheng Li

Last Update: June 2024

1 Introduction

Welcome, brave soul, to the wild ride of installing the HEP Trifecta. If you’ve ever felt that taming a dragon might be easier than setting up your computing environment, you’re in the right place. This document is your trusty map through the “sophisticated” process of getting MadGraph, Pythia, and Delphes up and running on your machine.

Think of this as the ultimate quest, where the end reward is not a golden chest, but something far more precious: a functioning HEP software suite. Along the way, you’ll likely encounter a bunch of confusing bugs and errors that make you second-guess your decision to pursue HEP in the future. But fear not! With a lot of patience and the proper guidance of this document, you’ll emerge victorious.

Before we begin, I should warn you that, just like any other installation guide, this is only “comprehensive” to the extent that it includes all the errors I encountered during the two times I installed the software on my personal computers. If you come across unexpected errors not mentioned in the guide, let me introduce you to your best friend throughout this journey—ChatGPT. Like any good friend, ChatGPT might not be reliable 100% of the time, but you can be sure that, for the most part, it is genuinely trying to help. Just use some discernment about whether the advice it offers is correct and safe. Also, don’t hesitate to copy and paste error messages from your terminal directly into Google Search. If you’re lucky, a solution will appear that addresses your issue exactly.

So, without further ado, let’s get started!

2 Preliminaries

Believe it or not, having the correct macOS version is crucial for installing this software. For your information, my installation process was completed on M-series machines with macOS Sonoma 14.5. Using an incompatible macOS version can lead to many confusing bugs. As far as I know, any version of macOS Sonoma should work fine. To avoid unexpected errors that might arise from an OS update, you should turn off automatic OS updates in your system settings. Once you have confirmed that you have the correct macOS version and have disabled automatic updates, you can proceed with the installation process.

3 Install Python

I highly recommend using Anaconda to manage Python versions since this makes it much easier to maintain a stable environment for the software. If you've already installed Anaconda and created an environment, please skip this part and go to section 3.

First, make sure that you have Anaconda installed. If not, please click this link <https://www.anaconda.com/download/success>, scroll to the bottom, and click the **64-Bit (Apple silicon) Graphical Installer** one to download Anaconda. Alternatively, you can follow any instructions on YouTube for a better demonstration to install Anaconda or ask ChatGPT for guidance.

Then create a new virtual environment by going to your terminal and typing:

```
$ conda create -n name_of_your_env python=3.11
```

Replace `name_of_your_env` with the name of your new environment.

Then type:

```
$ conda activate name_of_your_env
```

to activate your conda environment. (You will need to do this every time you open up a new terminal)

Now, you have a virtual Python environment installed on your computer!

4 Install ROOT

We'll also need to install ROOT in our current conda environment. Don't forget to do:

```
$ conda activate name_of_your_env
```

whenever you open up a new terminal to ensure that you are in the correct environment.

Now, type the following lines on your terminal:

```
$ conda activate name_of_your_env
$ conda config --set channel_priority strict
$ conda install -c conda-forge root
```

If you see something like this:

```
PackagesNotFoundError: The following packages
are not available from current channels:
```

```
- root
- forge
```

Current channels:

```
- https://conda.anaconda.org/conda
- defaults
```

This means that your conda is not up-to-date. You will need to type the following lines to your terminal:

```
$ conda activate base
$ conda update conda
```

Then type:

```
$ conda activate name_of_your_env
```

The above updates the conda package which only exists in your base environment.

Then, retype these three lines to your terminal:

```
$ conda activate name_of_your_env
$ conda config --set channel_priority strict
$ conda install -c conda - forge root
```

To check if you have ROOT installed, type this on your terminal:

```
$ root --version
```

If you see something like this:

```
ROOT Version: 6.30/04
Built for macosxarm64 on Feb 22 2024, 13:40:57
From heads/master@tags/v6-30-04
```

You've successfully installed ROOT in your environment.

Lastly, you will need to add the environmental variables to your .zshrc file by doing the following:

```
$ nano ~/.zshrc
```

Scroll to the bottom and add the following lines at the end:

```
# >>> ROOT Configuration >>>
export ROOTSYS=$CONDA_PREFIX
export PATH=$ROOTSYS/bin:$PATH
export LD_LIBRARY_PATH=$ROOTSYS/lib:$LD_LIBRARY_PATH
export DYLD_LIBRARY_PATH=$ROOTSYS/lib:$DYLD_LIBRARY_PATH
# <<< ROOT Configuration <<<
```

Exit and save the changes, then type the following commands to your terminal:

```
$ source ~/.zshrc
$ conda activate name_of_your_env
```

Everything should be set up with ROOT on your computer now.

5 Install MadGraph

Now, we are ready to install MadGraph. First download the latest release from Madgraph's official website. For reference, I'm using the 3.5.4 version. After unzipping the file and putting it in a desired directory(I highly recommend sticking with this directory in the future to prevent file retrieval issues due to path changes), cd to that directory and type:

```
$ ./bin/mg5_aMC
```

You should see the following drawing:

```
*****
*
*               W E L C O M E to
*          M A D G R A P H 5 _ a M C @ N L O
*
*
*               *
*            *      * *      *
*          * * * * 5 * * * *
*            *      * *      *
*               *
*
*          VERSION 3.5.4                2024-04-05
*
*  The MadGraph5_aMC@NLO Development Team - Find us at
*      http://madgraph.phys.ucl.ac.be/
*              and
*      http://amcatnlo.web.cern.ch/amcatnlo/
*
*          Type 'help' for in-line help.
*
```

```
*           Type 'tutorial' to learn how MG5 works           *
*   Type 'tutorial aMCatNLO' to learn how aMC@NLO works      *
*   Type 'tutorial MadLoop' to learn how MadLoop works        *
*                                                             *
*****
```

You now have MadGraph installed on your computer!

6 Install Pythia

Without exiting the MadGraph command-line interface, type the following to your terminal:

```
install pythia8
```

You should see a bunch of messages pop up on your screen. Be patient and press **y** and **Enter** whenever you are prompted to install some necessary packages along the way. At the end, you should see:

```
Finished PYTHIA8 installation
```

You now have PYTHIA installed on your computer!

7 Install Delphes

This section is a little complicated as it is against the common practice of installing Delphes using MadGraph. However, I've encountered multiple difficulties using the default method on MacOS. I scrambled my way through and finally got Delphes installed. This section is my way of getting Delphes to work with MadGraph.

First, you can try to install Delphes using the default method by typing the following to the command-line interface of MadGraph(aka. after running `./bin/mg5_aMC`):

```
install Delphes
```

If you encountered an error like this during your installation:

```
>> Building libDelphesNoFastJet.so
clang-16: error: no such file or directory: '
  libDelphesNoFastJet.so'
make: *** [libDelphesNoFastJet.so] Error 1
Error detected during the compilation. Please check the
  compilation error and run make manually.
```

This means you will need to follow the steps below.

There are two ways to install Delphes manual.

First, if you have a GitHub account, and have already had your SSH keys linked to your GitHub account, you can type the following commands to your terminal:

```
$ git clone git@github.com:delphes/delphes.git
```

Or, you can go to this link <https://github.com/delphes/delphes/releases/tag/3.5.0>, scroll to the bottom, download the zip file, and finally unzip the file.

Then type:

```
$ cd /path/to/delphes/  
$ make
```

Replace `/path/to/delphes/` with the actual path to the Delphes folder that you just downloaded. You should see a bunch of outputs and warning messages pop up on the terminal. Don't get scared, it's the process. However, if you encounter a compilation error like this:

```
>> Compiling external/fastjet/ClosestPair2D.cc  
In file included from external/fastjet/ClosestPair2D.cc:31:  
In file included from external/fastjet/internal/ClosestPair2D  
.hh:34:  
/opt/anaconda3/envs/muon_collider/bin/./include/c++/v1/  
vector:549:52: error: arithmetic on a pointer to an  
incomplete type 'fastjet::ClosestPair2D::Point'
```

You likely have the wrong version for the clang compiler. You can check by typing the following command to your terminal(make sure that you are in the correct environment):

```
$ clang --version
```

If you see something like this:

```
clang version 16.0.6  
Target: arm64-apple-darwin23.5.0  
Thread model: posix  
InstalledDir: /Users/name/anaconda3/envs/name_of_your_env/bin
```

This means that you have clang 16.0.6 installed on your current environment, which is **NOT** the correct version of clang compiler. We need to go back to clang 14 to make everything work together. First type the following command to your terminal:

```
$ conda install -c conda-forge clang=14
```

You will see a bunch of messages warning you that conda will need to automatically downgrade some packages in your current environment. Don't worry, press **y** and **Enter** and continue with the process.

After it's done, you can check your clang version of your current conda environment by typing the following to your terminal:

```
$ clang --version
```

Make sure that you see something like this:

```
clang version 14.0.6
Target: arm64-apple-darwin23.5.0
Thread model: posix
InstalledDir: /Users/name/anaconda3/envs/name_of_your_env/bin
```

The crucial thing is that the first number after `clang version` should be 14 and not other numbers.

Then, type the following command to your terminal:

```
$ make clean
$ make
```

Now everything should compile without errors. If you encounter any problem along the way—I can't stress this enough—don't forget that ChatGPT is always your best friend!

8 Get Everything to Work Together

Now you are at the last step of the process. We've come a long way. Keep up, it's almost there.

First, you need to rename your `Delphes` directory to be, well, `Delphes`. The default name of your `Delphes` folder may not be `Delphes`, but in order for `MadGraph` to work together successfully with `Delphes`, you need to make sure that your `Delphes` directory is named "Delphes" (the characters within the quotes EXACTLY). You can do this by typing the following command to your terminal (make sure that you are in the parent/upper directory of your `Delphes` folder):

```
$ mv name_of_your_delphes_folder Delphes
```

Then, we will need to move the **Delphes** folder inside the **MadGraph** folder. You can achieve this by typing the following command to your terminal:

```
$ mv Delphes /path/to/MadGraph/
```

Replace `/path/to/MadGraph/` with the actual path to the **MadGraph** folder.

Then, we can generate an event to see if everything is indeed working together as we expected.

First, go to your **MadGraph** folder by typing the following command to your terminal:

```
$ cd /path/to/MadGraph/  
$ ./bin/mg5_aMC
```

You should see the interactive interface of **MadGraph**.

Second, type the following commands line by line:

```
generate e+ e- > mu+ mu-  
output eemm  
launch
```

You should see a bunch of messages and something like this at the end:

```
/===== Description =====|===== values =====| other options =\  
|1.Choose the shower/hadronization program | shower = OFF      | Pythia8      | | |
|2.Choose the detector simulation program  | detector = OFF    | Delphes      |  
|3.Choose an analysis package (plot/convert)| analysis = Not Avail. | Please install mo |  
|4.Decay onshell particles                 | madspin = OFF     | ON|onshell|full |  
|5.Add weights to events for new hypp.     | reweight = OFF    | ON           |  
|=====
```

Now, type the following commands line by line:

```
shower=Pythia8  
detector=Delphes
```

And then press **Enter** two times in a row. You should see a Safari/Chrome tab pop up in a few moments. If you see an **ERROR** symbol on your screen next to **PYTHIA**'s row, click on the **ERROR**. If you see some output similar to this:

```
File "/Users/bobli/Desktop/MG5_aMC_v3_5_4/madgraph/various/  
histograms.py", line 1681, in __init__  
    self.parse_histos_from_PY8_XML_stream(stream, run_id,  
File "/Users/bobli/Desktop/MG5_aMC_v3_5_4/madgraph/various/  
histograms.py", line 1797, in  
    parse_histos_from_PY8_XML_stream  
    raise MadGraph5Error("Histogram with run_id '%d' was not  
found in the "%run_id+\
```



```
madgraph.MadGraph5Error: Histogram with run_id '0' was not
found in the specified XML source.
```

You will need to do the following. First, open up a new terminal session and type the following commands to your terminal:

```
$ mkdir -p /opt/anaconda3/envs/name_of_your_env/etc/conda
/activate.d
$ nano /opt/anaconda3/envs/name_of_your_env/etc/conda/
activate.d/set_pythia8data.sh
```

Add the following line to the file:

```
export PYTHIA8DATA="/Users/name/path/to/MadGraph/HEPTools
/pythia8/share/Pythia8/xmldoc"
```

Replace `/name` with your username, and replace `/path/to/MadGraph/` with the actual path to your MadGraph folder. Now, to double-check that you indeed have the correct environment variable, you can type the following commands into your terminal:

```
$ conda activate name_of_your_env
$ echo $PYTHIA8DATA
```

If you see the path that you just put in, i.e:

```
/Users/name/path/to/MadGraph/HEPTools/pythia8/share/
Pythia8/xmldoc
```

You should be good to go.

Now, for the one last time, we will generate an event to check if everything works together smoothly. Please repeat the part of generating an e^+e^- to $\mu^+\mu^-$ event. If, at the end, you see something like this on your screen:

Available Results

Run	Collider	Banner	Cross section (pb)	Events	Data	Output	Action	
run_01	e^+e^- 500.0 x 500.0 GeV	tag_1	0.1043 ± 8e-05	10000	parton madevent	LHE	remove run	launch detector simulation
					pythia8	LOG HEPMC	remove run	launch detector simulation
					delphes	LOG rootfile	remove run	

[Main Page](#)

Congratulations, your journey with installing the HEP trifecta is finally over!