

Running Geant4 on the farm

G. Blume

August 2022

The purpose of this tech note is to deliver instructions on how to upload, compile, and run code on the farm at JLab. Additionally, I will detail common (my) confusions and discuss the instructions that were likely input incorrectly.

Contents

| | | |
|---|----------------------|---|
| 1 | Login | 2 |
| 2 | Upload of Code | 2 |
| 3 | Sourcing your Geant4 | 3 |
| 4 | Compiling your code | 3 |

1 Login

I will be discussing how this works with regards to working in Ubuntu, but I presume this should work in any Linux based OS. In order to sign into the farm you need to be using the ssh command in your terminal. You will begin by entering in the command:¹

```
$ ssh username@login.jlab.org
```

This will prompt you to enter your password, which upon completion you will now need to access the actual farm. This is done via the command:²

```
$ ssh username@ifarm
```

You will now be in the farm and can begin to navigate to your folder where you will run your code. If at any point you fail and are given an error of permission denied, talk to your advisor. They can give you access to the farm.

¹ \$ ssh username@login.jlab.org

² \$ ssh username@ifarm

2 Upload of Code

Once you have received permission to access the farm you will need to navigate to the directory. If you are reading this you are likely working within the positron group. Therefore, you will navigate to the positron directory via the familiar command:³

```
$ cd /group/positron
```

Now that you are here you will begin by creating a directory for your code to be in. Creating your directory is done via the command:⁴

```
$ mkdir yourdirectory
```

Personally, I choose to work a layer deeper by creating a folder for Geant4 in my directory. This is done by first navigating to your directory via the command:⁵

```
$ cd /yourdirectory
```

Then in your directory, creating the directory GEANT4 via the command:⁶

```
$ mkdir GEANT4
```

At this point, do not forget to navigate into your GEANT4 directory using command:⁷

```
$ cd /GEANT4
```

³ \$ cd /group/positron

⁴ \$ mkdir yourdirectory

⁵ \$ cd yourdirectory

⁶ \$ mkdir GEANT4

⁷ \$ cd /GEANT4

Now we wish to put the code you have written in this folder. This is done by first opening a new terminal, then using this copy command:

```
$ scp -r localdirectory/ username@login.jlab.org:/group/positron/yourdirectory/GEANT4
```

Notice this command is not included in the side bar. This is because it needs to be done in a separate terminal to that of which you are working in the farm. Upon completion of this command you will be prompted to enter your password. If entered correctly your code will copy to the farm

3 Sourcing your Geant4

In order to run your Geant4 scripts you need to source your Geant4 in the farm environment. To do this we follow the instructions found on the scientific computing website for JLab. To view their documentation please follow the link at <https://data.jlab.org/drupal/?q=node/44>. I personally make use of the command:⁸

```
$ source /site/12gev_phys/softenv.csh 2.5
```

⁸ \$ source /site/12gev_phys/softenv.csh 2.5

This will import the Geant4 folders for edition 11.0.1. There are other libraries that are also imported with this and the documentation can be found on that link aforementioned. We are now able to compile and run your code.

4 Compiling your code

As you have now sourced your Geant4, you are ready to build your executable in the way you likely have been. To begin we navigate into the directory you just copied. This is done via the command:⁹

```
$ cd /localdirectory
```

⁹ \$ cd yourdirectory

It is likely a good point to check where we are. To do this enter the command `pwd`. This should return the current directory you are in. At this point it should look like `/group/positron/yourdirectory/GEANT4/localdirectory`. Now here I create a build folder and compile but you can write scripts to automate the process (.sh files). I create my build folder via the command:¹⁰

```
$ mkdir build
```

¹⁰ \$ mkdir build

Now navigate into the build folder using command:¹¹

¹¹ \$ cd /build

```
$ cd /build
```

Now, we make use of the cmake capabilities to copy our cmake files into our build directory via the command:¹²

```
12 $ cmake ..
```

```
$ cmake ..
```

All that is left is to make your executable using command:¹³

```
13 $ make -j40
```

```
$ make -j40
```

You will notice I use 40 cores here. This is just an example, at the time of writing this I have no idea what number of cores I should use.

As of now you can run your code using the executable. Make sure you run in batch with an example being:¹⁴

```
14 $ ./executable example.mac
```

```
$ ./executable example.mac
```

This should run your code on the farm!