

# LumiCal Simulation Documentation

Jonathan Aguilar

November 27, 2009

## 1 Running Lcal-pro from the command line

### 1.1 Environment

The working version of LumiCal is typically **Lcal-pro**. Normally it does not appear as an executable until you set the environment. To do this, log in as **fc**, go to the home folder and run the **env.sh** script:

```
[fc@nz12-25 fc] . env.sh
```

### 1.2 Command line arguments

The syntax for running Lcal-pro is **Lcal-pro [options]**. **Lcal-pro -h** prints a list of command line arguments. The complete list is:

-h	print this help message.
-b	batch mode
-i	interactive mode
-m <filename>	specifies a macro file to be executed before running. default none
-o <filename>	specifies file name for ROOT output.No default
-M <mode>	specifies ROOT file opening mode ( default is CREATE to avoid accidental file overwriting ). Possible values are RECREATE/UPDATE
-A <Yes/No>	specifies whether events from entire Run are to be written in one Tree ( suitable only for beam background data ) default is 'No'
-c <double>	specifies the Geant 4 production range cut in mm. (default is 0.005 mm)
-x <double>	specifies the Beam Crossing Angle in mrad (default is 0 [mrad])
-s <filename>	specifies name of the file with geometry setup
-P <int>	specifies printout level ( default is 0= minimum print 3= debug printout

Note: `-M` currently has no default setting (see `fcal/LumiCal-pro/src/Setup.cc`)

Typically the most common options are `-b`, `-m`, and `-o`. `-o` should be followed by a filename ending in `.root`. You can write the Geant4 output to file by appending `>> logfile.log` to the end.

## 2 Batch mode, Macros

The option `-b` tells Lcal-pro to execute a macro, as opposed to running in interactive mode. Specify the macro by following `-b` with `-m macroname.mac`:

```
[fcal@nz12-25 LumiCal-pro] Lcal-pro -b -m
geant4macro.mac -o rootfile.root > logfile.log
```

Of course, the macros and output files can be stored in other directories, so if this is the case just prefix them with the path. Use `"&>"` if you want stderr sent to the log file in addition to stdout.

## 3 Geometry setup

**Lcal-pro** receives the geometric parameters for the LumiCal detector from an external file. This allows for the geometry to be changed without having to recompile the program. To change the geometry, change the file. It typically has the suffix `.ini`. It's probably a good idea to have the file name reflect the changes in geometry.

## 4 Pausing and restarting a run

### 4.1 Pausing

You can pause a run by placing a file in the current working directory called `aStopRun`. The file's contents don't matter. Lcal-pro periodically checks the current directory for this file, and quits cleanly if it finds it. My favorite way to do this is with **touch**:

```
touch aStopRun
```

Then, check the current processes until this instance of Lcal-pro has stopped running. Warning: all Lcal instances that were started in this folder will be able to see **aStopRun**, so the one that reads and deletes it first is the one that will stop. For this reason, I typically have a dedicated folder for each instance.

### 4.2 Restarting

To restart a run, use the same command and options you used to start it in the first place, with the following changes:

- Make sure the Setup file has **RootFileMode** set to **UPDATE!**
- Use `>>` to append the output to the same log file you used before.

Use the same filenames and macro as before.

### 4.3 Managing several instances

My current computer has two processors; therefore, I like to leave two instances of LumiCal running over the weekend to use the computer more efficiently. It is helpful to be able to stop and start them independently of each other when you want to check on their progress, but if you launch them from the same folder, they will both pause when you create **aStopRun**. Instead, launch each instance from a separate folder, e.g. **LumiCal-pro/job1** and **LumiCal-pro/job2**. Then, you can create aStopRun inside of the folder from where you launched the instance you want to pause.

## 5 ROOT file output

**Lcal-pro** is currently designed to write a ROOT tree once the Geant4 run is finished. It creates the file when the run starts, so you can check quickly to see if the file has been properly created. The tree has the following branches:

- numPrim - the number of primary particles
- pX[1], pY[1], and pZ[1] - momentum components of the primary(ies)
- primID[1] - primary particle ID number
- primPDG[1] - the type of primary (PDG code)
- numHits - the number of recorded hits (1 per unique cell)
- cellID[numHits] - the IDs of the cells which recorded hits
- rCell[numHits], phiCell[], zCell[] - cells' cylindrical coordinates
- xHit[numHits], yHit[], zHit[] - hits' cartesian coordinates
- eHit[numHits] - energy deposited into each cell
- Etot1 - total energy deposited in the run, in module 1
- Etot2 - total energy deposited in the run, in module 2
- numTracks - the number of primary tracks
- trackPID[1] - the ID of the track

## 6 Troubleshooting

1. I updated the version and now make quits with a “touch: cannot touch” error.
  - Make sure the executable in the makefile, the directory, and the main (foo.cc) all have the same name.
2. Lcal-pro isn't writing ROOT files.
  - Try running Lcal-pro in the foreground. Is it pausing after the run finishes? You might be running in interactive mode. Check that your setup file has **batchMode = 1**.

3. Lcal-pro isn't behaving according to the commands I gave from the terminal.
  - Setup files override command line options. If you are using the -s option with a setup file, check that the options in the file match with the ones you want to give from the terminal prompt.