

# The RootTreeWriter and a processor template script

Ch. Rosemann

DESY

25.11.2009 MarlinTPC week

# RootTreeWriter

Looking at your (LCIO) Data

Author: Jörgen Samson

<http://www-flc.desy.de/flc/flcwiki/RootTreeWriter>

- A way to write self defined RooT-Trees within a processor
- Publicly available source from:  
</afs/desy.de/group/flc/soft/softcvs; module: RootTreeWriter>
- Please report bugs:  
<http://www-flc.desy.de/flc/flcwiki/RootTreeWriter/Bugs>  
Or: [joergen.samson@desy.de](mailto:joergen.samson@desy.de)
- The RootTreeWriter is licensed under a dual LGPL/BSD license

# Using the RootTreeWriter

- 1 Create an *engine* using `createEngineTemplate.sh` in the *bin* directory
- 2 Adapt the `.h` and `.cc` file to your needs:
  - Data collections
  - Tree variables
  - Computations
  - ...
- 3 Compile with the usual `cmake` and `make` commands
- 4 A shared object (`.so`) library is created
- 5 Set/extend your `$MARLIN_DLL` shell variable to find it (like with MarlinTPC)
- 6 Edit your steering file accordingly
- 7 Run it!

# How to create a processor?

## The processor template script

- Please find in **MarlinTPC/tools/scripts**:  
`createProcessor.py`
- It takes either a single argument (a name) ...
- ... or additionally a path
- And it will create two files: a `.h` and a `.cc` file (optionally in the given path)
- In this the skeleton of a Marlin processor is built
- Edit this to your needs

**Current version is buggy!**

Please blame [christoph.rosemann@desy.de](mailto:christoph.rosemann@desy.de)