Rivet fact sheet

Terascale Monte Carlo School 2014, DESY Hamburg

2014-03-13

Introduction

Rivet is designed as a generic plotting and analysis tool for event generators, following the idea

Event generators → HepMC output $\overset{pipe}{\rightarrow}$ Rivet → analyses, validation, tuning...

The `rivet` command

Rivet is already pre-installed in your virtual machine and should be usable right away. To test that it's working properly, try listing the available analyse by typing:

```
rivet --list-analyses
rivet --show-analyses RESULT_#1
```

Details on the standard analyses can be found under `http://rivet.hepforge.org/analyses` The `rivet` command is primarily used to analyse HepMc events. HepMc events in the file `events.hepmc` can be analysed by using

```
rivet -a ANALYSIS_1_NAME -a ANALYSIS_2_NAME -H PLOTS.aida events.hepmc
```

where `ANALYSIS_1_NAME` and `ANALYSIS_2_NAME` are taken from the list of rivet analyses (see above). All options of `rivet` can be examined by typing

```
rivet --help
```

HepMc event files quickly become prohibitively large. It is therefore possible to have `rivet` read events from a FIFO (first-in-first-out) pipe in "real time". This reduces disk space as you will see in the tutorials. To read from a FIFO pipe, use the following commands

```
mkfifo hepmc.fifo
my-generator --num-events=10000 --hepmc-output=hepmc.fifo &
rivet -a ANALYSIS_NAME hepmc.fifo
```

Here, the second line is meant symbolically. You'll encounter concrete examples in the tutorials.

Plotting the analysis output

`rivet` produces output histograms in a file in the `AIDA` format. Such `.aida`-files can be plotted by using the command

```
rivet-mkhtml --cm -o plots PLOTS.aida:'Title=My Plots'
```

This will build a directory `plots`, with an `index.html` file. This index to your results can be opened with any standard browser (e.g. `firefox`). You can examine all options of the `rivet-mkhtml` command by typing

```
rivet-mkhtml --help
```

Layout tweaks (like the statement `:'Title=My Plots'` above) are documented under `https://rivet.hepforge.org/make-plots.html` .

Writing an analysis

In the course of these tutorials, you will get to modify / write your very own RIVET analysis. Each analysis is encapsulated in a standalone `c++` class. This class has to contain the members

- `init`: book histograms and define projections
- `analyze`: select particles, filter according to cuts, construct observables, fill histograms. This is where the per-event aspect of the analysis algorithm goes.
- `finalize`: normalize/scale histograms, etc.

Have a look at one of the standard analyses, e.g. `/opt/share/Rivet/MC_ZINC.cc`, to get a feel for the code. Try to understand what `projections` do, and how the observables are calculated and booked into the histograms. Then, try to answer

- What are the numbers in : `bookHisto1D("Z_mass", 50, 66.0, 116.0);`?
- Whats the action of `vetoEvent` in line 44?
- What is the difference between `normalize(_h_Z_mass, xsec);` and `normalize(_h_Z_mass, 1.);`?