Table of Contents

3.5.3 Examples of FW Lite Macros

Complete:
Detailed Review status

Goals of this page:

In this page you will find working examples of FW Lite macros that can be used as starting points for your own analysis.

To use this tutorial you must follow the procedure described here.

For brevity the latest version is linked to the top of the page here.

We also need to create a PAT-tuple (described in detail in WorkBookPATTupleCreationExercise):

```
scram b -j 4
cmsRun PhysicsTools/PatAlgos/test/patLayer1_fromAOD_full_cfg.py
```

From there, we will have a file "PATLayer1_Output.fromAOD_full.root" to run over.

The tutorials that follow are contained in CMS.PhysicsTools/PatExamples.

Contents

Example 1: the Bare Bones -- Plot PAT objects

The first example of plotting using FWLite will be to plot PAT objects. The source for this can be found here. To run this example, type:

```
rehash
PatBasicFWLiteAnalyzer
```

(the "rehash" need only be done once after compilation of a new release).

Example 2: Using python configuration

The next example will be to use a python configuration to input parameters to our FWLite macro. The source for this can be found here. This is identical to the previous example, except that it now uses the python interpreter to input parameters to the FWLite macro:

```
// Get the python configuration
  PythonProcessDesc builder(argv[1]);
  edm::ParameterSet const& fwliteParameters = builder.processDesc()->getProcessPSet()->getParamet

  // Now get each parameter
  edm::InputTag muonLabel( fwliteParameters.getParameter<edm::InputTag>("muonSrc") );
```

The file now takes an additional input argument on the command line, which is the location of the python configuration

```
PatBasicFWLiteAnalyzer_PyConfig PhysicsTools/PatExamples/bin/patBasicFWLiteAnalyzer_PyConfig_fwli
```

This python configuration is, for the moment, extremely simple for demonstration and can be found here.

Example 3: Using Selectors

This example will show some more advanced features of the framework. Here we will be using an EventSelector to make the selection, rather than doing it in the "main loop". This will allow us to keep track of statistics of the event selection.

More details about Selectors can be found in SWGuidePATSelectors.

This particular example will create a "simple" W selector that will select events with muons with pt above 20 GeV/c, and MET above 20 GeV/c. The source is found here. To run this, type

```
PatBasicFWLiteAnalyzer_Selector PhysicsTools/PatExamples/bin/patBasicFWLiteAnalyzer_Selector_fwli
```

The new configuration is slightly more complicated than before and can be found here.

At the end of the job should be a snippet like this:

```
     0 :              Muon Pt         22
     1 :                  MET         86
```

These correspond to the number of events that pass the muon pt, and MET cuts, respectively. This can be made as complicated as one desires and automatically keeps track of cut flows.

Example 4: Realistic Example With First Data

This example will build upon the technology developed in the previous ones to create a full-functional analysis on first data and minbias MC. For further details on the events and event selections that follow, please see the following link:

- https://twiki.cern.ch/twiki/bin/viewauth/CMS/Collisions2010Analysis

To do this exercise, it will first be necessary to create a new PAT-tuple of the Minimum bias Monte Carlo:

```
cmsRun PhysicsTools/PatExamples/test/patLayer1_fromRECO_7TeV_firstdata_mc_cfg.py
```

Now, create a PAT-tuple of the Minimum Bias 7 TeV data:

Coming soon!!!

```
cmsRun PhysicsTools/PatExamples/test/patLayer1_fromRECO_7TeV_firstdata_cfg.py
```

We can then analyze the events using the analysis macro here. The configuration to run this example is found here.

This will perform a dijet selection on the events in question, and plot some jet ID variables. At the end of this FWLite job, you should see

```
Calo jet selection
     0 :            Calo Cuts        624
     1 :        Calo Kin Cuts         26
     2 :       Calo Delta Phi         14
     3 :          Calo Jet ID         12
     4 :              PF Cuts        off
     5 :          PF Kin Cuts        off
     6 :         PF Delta Phi        off
     7 :            PF Jet ID        off
PF jet selection
     0 :            Calo Cuts        off
     1 :        Calo Kin Cuts        off
     2 :       Calo Delta Phi        off
     3 :          Calo Jet ID        off
     4 :              PF Cuts        624
     5 :          PF Kin Cuts         57
     6 :         PF Delta Phi         25
     7 :            PF Jet ID         20
```

There should be histograms in the file "jetPlots.root".

Congratulations! You're commissioning!

Example 5: Accessing Conditions Database

Coming soon!

Example 6: Access to Runs,, and their products

This macro is a demonstration of the various ways to get at luminosity block and run information.

```
#if !defined(__CINT__) && !defined(__MAKECINT__)
#include "DataFormats/FWLite/interface/Handle.h"
#include "DataFormats/FWLite/interface/Event.h"

#include "DataFormats/FWLite/interface/LuminosityBlock.h"
#include "DataFormats/FWLite/interface/Run.h"
#include "DataFormats/Luminosity/interface/LumiSummary.h"
#endif

#include "TFile.h"


void print_data() {
   TFile file("SomeData.root");

   fwlite::Event ev(&file);
   fwlite::Handle<LumiSummary> summary;

   // We can get run and luminosity blocks from events as well as associated products
   // All this works for ChainEvent and MultiChainEvent as well, not just on one
   for(ev.toBegin(); !ev.atEnd(); ++ev) {
      // We can get the Luminosity block ID from the event
      std::cout << " Luminosity ID " << ev.getLuminosityBlock().id() << std::endl;
      // or the Run ID from the event
      std::cout <<" Run ID " << ev.getRun().id()<< std::endl;
      // We can get the Run ID from the luminosity block we got from the event
      std::cout << "Run via lumi " << ev.getLuminosityBlock().getRun().id() << std::endl;
      // We can get the integrated luminosity (or any luminosity product) from the event
      summary.getByLabel(ev.getLuminosityBlock(),"lumiProducer");
      std::cout << " Inst.  Luminosity = " << summary->avgInsDelLumi()<<std::endl;
   }

   // We can also loop over luminosity blocks themselves, same as events
   fwlite::LuminosityBlock ls(&file);
   for(ls.toBegin(); !ls.atEnd(); ++ls) {
      summary.getByLabel(ls,"lumiProducer");
      std::cout  << ls.id() << " Inst.  Luminosity = " << summary->avgInsDelLumi() << std::endl;
      // And get the associated run from this lumi
      std::cout << "Run from lumi " << ls.getRun().id() << std::endl;
   }

   // And of course do the same for runs.
   fwlite::Run r(&file);
   for(r.toBegin(); !r.atEnd(); ++r) {
      std::cout << "Run " << r.id() <<std::endl;
   }
}
```

Review status

Reviewer/Editor and Date (copy from screen)	Comments
YourName - 14 Mar 2006	What you did (e.g., created page, reviewed, edited, added/answered questions or comments, etc.) and/or what you think needs to be done next (e.g., "needs review by ...", "still needs content on subject x", etc.)

Responsible: SalvatoreRoccoRappoccio
Last reviewed by: BenediktHegner - 21 Jun 2009

-- SalvatoreRoccoRappoccio - 08-Mar-2010

This topic: CMSPublic > WorkBookFWLiteExamples35X
Topic revision: r6 - 2010-04-27 - RogerWolf

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback