

CMS Software (CMSSW) Exercises:

Prepared by: Dr. Taimoor Khurshid & Dr. Muhammad Ahmad

Problem: Reconstruction of Z boson from its di-muon decay

Introduction

Welcome to the 5th School on LHC Physics exercises. The purpose of these exercises is to become familiar with the basic software tools required to perform physics analysis at the school.

setting cvmfs envirnment:

CernVMFS (cvmfs) is a network file system based on HTTP and optimized to deliver experiment software in a fast, scalable, and reliable way.

Please run the following commands at your Home directory:

```
ll /cvmfs/cms.cern.ch
setenv SCRAM_ARCH slc6_amd64_gcc493
setenv VO_CMS_DIR /cvmfs/cms.cern.ch
source $VO_CMS_DIR/cmsset_default.csh
scram list
```

The last command will tell you about the all available releases of CMSSW available under the architectures.

CMSSW release

Now we have to checkout the “CMSSW_7_4_15” release for our exercises purpose. So please run the following commands:

```
mkdir CMSSW
cmsrel CMSSW_7_4_1
cd CMSSW_7_4_1/src
cmsenv
```

Writing your own EDAnalyzer

First, create a new directory in src:

```
mkdir Test
cd Test
```

Now Create the "skeleton" of an EDAnalyzer module, and compile your first code in CMSSW:

```
mkedanlZr MiniAnalyzer
cd MiniAnalyzer
scram b
ls
```

Finally, your analysis directory looks like:

```
New package "MiniAnalyzer" of EDAnalyzer type is successfully generated
MiniAnalyzer/
| plugins/
| |-- BuildFile.xml
| |-- MiniAnalyzer.cc
| python/
| |-- CfiFile_cfi.py
| |-- ConfFile_cfg.py
| test/
| doc/
Total: 4 directories, 4 files
```

For your convenience, the root file is available locally at:

```
cp /scratch/m/mahmad/SLP2016/RootFiles/DYJetsToLL_M-
50_TuneCUETP8M1_13TeV-madgraphMLM-
pythia8_RunIISpring15DR74_MINIAODSIM.root
```

Standalone EDM (Event Data Model framework) Utilities:

The overall collection of CMS software, referred to as CMSSW, is built around a Framework, an Event Data and Services needed by the simulation, calibration and alignment, and reconstruction modules that process event data so that physicists can perform analysis.

The CMS Event Data Model (EDM) is centered around the concept of an Event. An Event is a C++ object container for all RAW and reconstructed data related to a particular collision. To understand what is in a data file and more, several EDM utilities are available. In this exercise, one will use two of these EDM utilities.

edmDumpEventContent

edmDumpEventContent is used to dump a summary of the products that are contained within the file we're interested in. Use edmDumpEventContent to see what class names etc.

```
edmDumpEventContent /scratch/m/mahmad/SLP2016/RootFiles/DYJetsToLL_M-50_TuneCUETP8M1_13TeV-
madgraphMLM-pythia8_RunIISpring15DR74_MINIAODSIM.root
```

The output of `edmDumpEventContent` has information divided into four variable width columns.

| C++ class type | module label | product instance label | process name |
|--|--|------------------------|---------------------|
| <code>vector<pat::Muon></code> | <code>"slimmedMuons"</code> | <code>""</code> | <code>"PAT"</code> |
| <code>vector<reco::Vertex></code> | <code>"offlineSlimmedPrimaryVertices"</code> | <code>""</code> | <code>" PAT"</code> |
| <code>vector<reco::GenParticle></code> | <code>"prunedGenParticles"</code> | <code>""</code> | <code>" PAT"</code> |

This information will help us when we are accessing the data for our physics analysis.

And if you want to look for specific module you can use the command:

```
edmDumpEventContent --all --regex pfJet /scratch/m/mahmad/SLP2016/RootFiles/DYJetsToLL_M-50_TuneCUETP8M1_13TeV-madgraphMLM-pythia8_RunIISpring15DR74_MINIA0DSIM.root
```

output of `edmDumpEventContent` can be stored in text file as:

```
edmDumpEventContent /scratch/m/mahmad/SLP2016/RootFiles/DYJetsToLL_M-50_TuneCUETP8M1_13TeV-madgraphMLM-pythia8_RunIISpring15DR74_MINIA0DSIM.root > edmDump.txt
```

edmEventSize

Now we will execute `edmEventSize` to determine the size of different branches in the data file.

```
edmEventSize -v /scratch/m/mahmad/SLP2016/RootFiles/DYJetsToLL_M-50_TuneCUETP8M1_13TeV-madgraphMLM-pythia8_RunIISpring15DR74_MINIA0DSIM.root > edmEvent.txt
```

As already shown above “mkedanlzl” has also created a configuration file `ConfFile_cfg.py` in the `DemoAnalyzer/python` directory.

Open `ConFile_cfg.py` using vi editor:

```
vi python/ConfFile_cfg.py
```

Change the root file path in the configuration file by:

```
'file: /scratch/m/mahmad/SLP2016/RootFiles/DYJetsToLL_M-50_TuneCUETP8M1_13TeV-madgraphMLM-  
pythia8_RunII_Spring15DR74_MINIAODSIM.root'
```

```
cmsRun Demo/DemoAnalyzer/python/ConfFile_cfg.py
```

So everything is working fine yet.

Now replace the files by following files: (`ConfFile_cfg.py`, `MiniAnalyzer.cc`, `BuildFile.xml`)

```
/scratch/m/mahmad/MiniAOD_Ex1/python/ConfFile_cfg.py  
/scratch/m/mahmad/MiniAOD_Ex1/plugin/MiniAnalyzer.cc  
/scratch/m/mahmad/MiniAOD_Ex1/plugin/BuildFile.xml
```

Exercise_2 (Adding Histograms):

We do the steps:

- Definition of Histogram
- Initialization of Histogram
- Fill the values in Histogram
- Also added `TFileService` to save the output in the root file.

I have added the Muon's p_t , η , ϕ and energy variables as histograms. You can just copy and paste files from:

```
/scratch/m/mahmad/MiniAOD_Ex2/python/ConfFile_cfg.py  
/scratch/m/mahmad/MiniAOD_Ex2/plugin/MiniAnalyzer.cc  
/scratch/m/mahmad/MiniAOD_Ex2/plugin/BuildFile.xml
```

Exercise_3 (Reconstructing Z mass decaying to di-muons):

Add another for loop of muon.



5th School on LHC Physics (15 – 26 August 2016)



Save the values of μ_1 and μ_2 in two different `TLorentzVector`.
Add the μ_1 and μ_2 `TLorentzVector`s which makes the `Z` `TLorentzVector` vector.

Exercise_4 (Accessing all Objects in CMSSW)

In this exercise we learn to access all the physics objects.

cmsShow

How to use cmsShow at your local Machine

<https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookFireworks>