Defining basic objects in the MadAnalysis 5 expert mode

Sam Bein

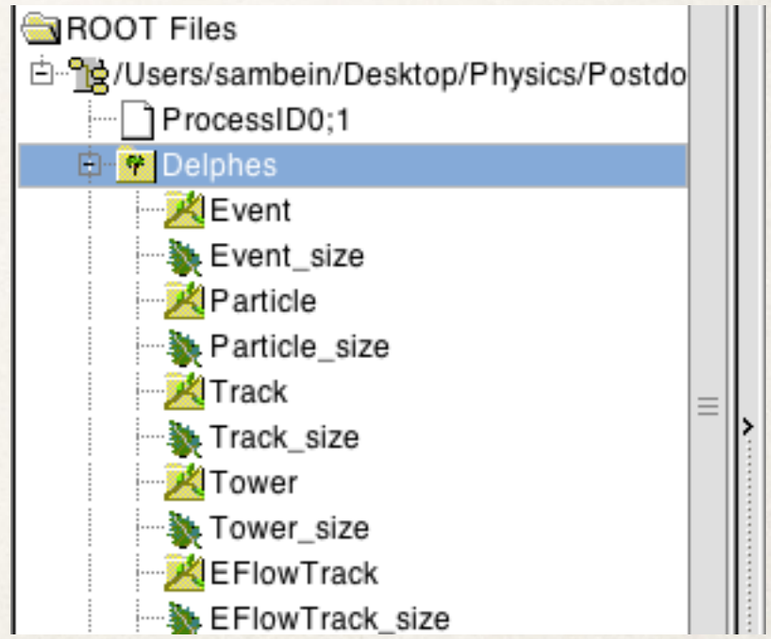
Tuesday, August 22

Object selection vs. event selection

- In HEP jargon, an "object" refers to a component of single event,
  - i.e., a jet, lepton, MET, etc.

- Object selection is typically carried out before the event selection.

- Objects are accessed in MadAnalysis 5 directly from the branches of root trees.



let's go through an example

Tutorial (1)

- We'll use the MA5 version installed yesterday with Delphes and we'll begin in the madanalysis5 directory.

```
cd madanalysis5
```

- In a familiar way, we'll set up a new analyzer by running ma5 in expert mode:

```
./bin/ma5 -E
```

- You will be asked to input a directory and analysis name. We'll use the name **obj_tutorial** for both.

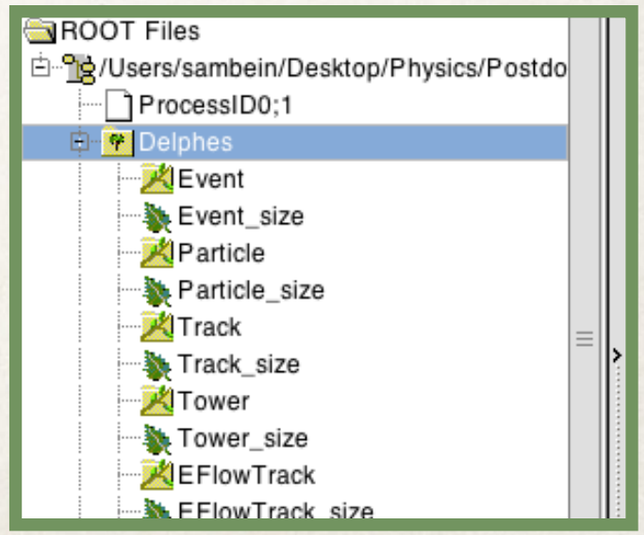
```
#note that the files
#obj_tutorial/Build/SampleAnalyzer/User/Analyzer/obj_tutorial.[cpp,h]
#have just been created.
```

Tutorial (2)

- After exiting ma5, obtain a set of working Delphes ROOT files (1 is sufficient) and create a text list for these files; if you're following the tutorial closely,

```
wget http://phys.onmybike.nl/ma5/testfile.root
ls -1 -d $PWD/testfile.root > obj_tutorial/Input/filelist.txt
```

- The root file testfile.root contains a tree with a small sample of events. Feel free to open up the file and have a look:




```
root -l testfile.root
[0]TBrowser b
#You can open the tree by double clicking
#Delphes in the browser to reveal branches
```

Tutorial (3)

- When you're done looking at the root file, we can close ROOT by entering .q into the prompt.

- We'll go into the newly-created folder called Build and call a couple of scripts

```
cd obj_tutorial/Build
source setup.sh #[or source setup.csh if using a c shell]
make
#setup.sh sets a number of environment variables and make compiles
#the c++ program called SampleAnalyzer/User/Analyzer/
#obj_tutorial.cpp
```

Tutorial (4)

- We can open up SampleAnalyzer/User/Analyzer/obj_tutorial.cpp with our favorite text editor and start exploring the code (and we'll remain in the Build dir).

- After glancing at the code, let's just run it to make sure it works; note the file list is specified as arg 1:

```
./MadAnalysis5job ../Input/filelist.txt
```

- Assuming there is no crash, we're in business to start playing with objects and their selection.

Tutorial (5)

- Let's remove the comment bookends on lines 239 and 124; try not to displace lines to preserve the accuracy of the line number references in this tutorial.

- We have just uncommented the meat and potatoes of the event loop, aka, execute function.

- In order to not drink from a firehose, let's put a return statement on l 150 after the electron loop:

```
return true;
```

Tutorial (6) Electrons

- The loop will now only look at electrons. Note how the electrons are accessed using the built-in MA5 collections

```
for (MAuint32 i=0;i<event.rec()->electrons().size();i++)
 {
   const RecLeptonFormat& elec = event.rec()->electrons()[i];
  …
 }
```

- Ok, let's run the code and look at the output

```
make
./MadAnalysis5job ../Input/filelist.txt
```

Tutorial (7) Muons and taus

- To add the other leptons, we can move the return statement from l 150 to l 195 (preserving line numbers)

```
return true;
```

- Once again, run and check your expectations

```
make
./MadAnalysis5job ../Input/filelist.txt
#feel free to ask questions here or any time :)
```

-

Tutorial (8) Jets

- Moving the return statement from 199 to 220 (again preserving line numbers) will print the jet information

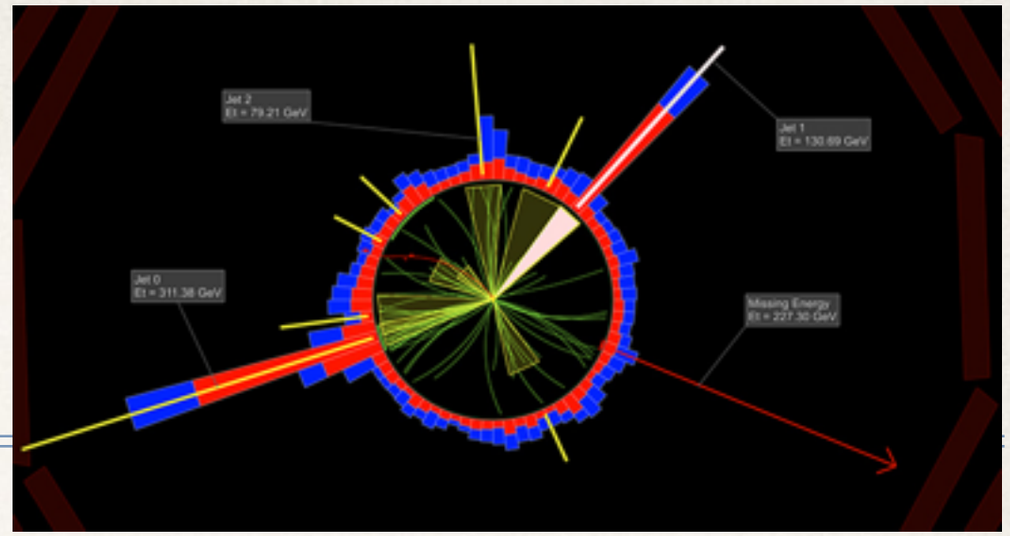
```
return true;
```

- Once again, run and check your expectations

```
make
./MadAnalysis5job ../Input/filelist.txt
#feel free to ask questions here or any time :)
```

-

Tutorial (9) Missing transverse energy



- Moving the return statement from 220 to 230 (again preserving line numbers) will print the MET and MHT information

```
return true;
```

- In the first event, how does the MET compare with the MHT? How about in the 2nd? Any thoughts as to why?

```
make
./MadAnalysis5job ../Input/filelist.txt
#feel free to ask questions here or any time :)
```

Tutorial (10) Total transverse energy

- Removing the return statement on line 230 will reveal the rest of the objects in the code; we're now drinking from a firehose.

```
make
./MadAnalysis5job ../Input/filelist.txt
#feel free to ask questions here or any time :)
```

- After examining a few events, is the THT larger than MET, or vice versa? Any thoughts as to why?

Tutorial (11) MT2, stransverse mass

This is the first example of the use of the special Services class in MA5; more can be found in v1.6beta/tools/SampleAnalyzer/Commons/Service/Physics.h

- The MT2 is used in some searches as a substitute for the MET for various reasons. It is constructed as a function of two 4-vectors, as well as the MET, and a test mass typically set to 0. The test mass is a guess for the dark matter mass in SUSY models.

- Considering only events with at least two jets, can you determine the MT2 value based on the two leading jets?

```
#hint 1:at this point, feel free to abandon the line number preservation mission.

#hint 2:
    if (event.rec()->jets().size()>=2)
      {
    MA5::ParticleBaseFormat * p1 = new MA5::ParticleBaseFormat(event.rec()->jets()[0]);
    // line of code missing here ;)
    double mt2 = PHYSICS->Transverse->MT2( p1,  p2, event.rec()->MET(), 0);
    cout << "found mt2 to be " << mt2 << endl;
      }
```

Tutorial (12) Building collections

This is the second example of the use of the special Services class in MA5; more can be found in v1.6beta/tools/SampleAnalyzer/Commons/Service/Physics.h

- The isolation is often used to select objects of interest. It can can be retrieved in ma5 in various forms. The following snippet builds a collection of isolated tau leptons with certain criteria applied:

```
std::vector<const RecTauFormat*> SelectedTaus;
for(unsigned int ii=0; ii<event.rec()->taus().size(); ii++)
 {
   const RecTauFormat *myTau = &(event.rec()->taus()[ii]);
   double eta = fabs(myTau->eta());
   double pt = myTau->pt();
   double iso_var = PHYSICS->Isol->eflow->sumIsolation(myTau,
   event.rec(),0.4,0.,IsolationEFlow::TRACK_COMPONENT);
   if(eta > 2.4) continue;
   if(iso_var>0.20*pt) continue;
   if (pt>10)  SelectedTaus.push_back(myTau);
 }
cout  << "n(taus)=" << SelectedTaus.size() << endl;
```

$$\text{isolation(j)} = [\sum_{i\neq j}^{\Delta R<0.4}(p_{\mathrm{T}})_i]/(p_T)_j$$

- **the**

Tutorial (13) Optional extra questions

- what fraction of electrons within |eta|<2.1 have an isolation of less than 0.2?

- what fraction of jets with at least 3 tracks and pT>30 GeV and are within |eta|<2.1?

- what is the azimuthal angle between the leading jet and the MET in the first event with at least two jets?

Great job! Thanks for you attention.