Monte Carlo in ATLAS

Introduction

This tutorial has been adapted for the new workflow for event generation.

There are four excercises in this session following the new workflow, which went into operation in October 2019:

1. Generate minimum bias events with Pythia8
2. Generate $Z \rightarrow ee$ events with Sherpa
3. Run the same Sherpa generation and also run and plot a Rivet analysis
4. From the EVNT output of exercise 2. make a TRUTH3 DxAOD

There are four exercises in this session following the old workflow before October 2019:

1. Generate minimum bias events with Pythia8
2. Generate $Z \rightarrow ee$ events with MadGraph+Pythia8
3. Run the same MadGraph+Pythia8 generation and also run and plot a Rivet analysis
4. From the EVNT output of exercise 2. make a TRUTH3 DxAOD

At the end of this session you should be familiar with

- How to setup your environment for Monte Carlo generation
- How to run the generate transform
- The structure of the log.generate output log
- Some of the ways you can look at the output of MC generation

New Workflow

Setting up your environment

To setup an appropriate Athena environment please execute the following setup commands in a fresh directory on lxplus:

```
setupATLAS
mkdir <workdir>
cd <workdir>
asetup AthGeneration, 21.6.12,here
```

Running a Pythia8 minimum bias example

Start by creating a clean directory

```
mkdir Pythia8_MB_example/
cd Pythia8_MB_example/
```

We will be using the example jobOptions living here. For this first exercise we will run a minimum bias example with a jet filter. The corresponding DSID is 421114 and the jobOptions live in the 421xxx/421114 directory. If you want to inspect it locally, you can directly copy it into your directory from cvmfs:

```
cp /cvmfs/atlas.cern.ch/repo/sw/Generators/MCJobOptions/421xxx/421114/mc.Py8EG_A2MSTW2008LO_minbias_inelastic_high.py
```

or using git (assuming you have added your lxplus ssh-keys to gitlab):

```
git archive --remote=ssh://git@gitlab.cern.ch:7999/atlas-physics/pmg/mcjoboptions.git HEAD:421xxx/421114 mc.Py8EG_A2MSTW2008LO_minbias_inelastic_high.py | tar -x
```

In general, you will not need this step to run over a jobOption stored in gitlab and will only need it for doing local test.
Now you can run your first generation job! Execute the following command but instead of running it exactly as below, change the random seed to your date of birth first (DDMMYYYY). Then you can compare the difference in output with your neighbour.

```
Gen_tf.py --ecmEnergy=13000. --firstEvent=1 --maxEvents=1 --randomSeed=1234 --jobConfig=421114 --outputEVNTFile=test_MB_421114.EVNT.root
```

While the job is running, have a look at:

- the Gen_tf.py command you just executed. What are all the input arguments? Why?
- the jobOptions you just used: mc.Py8EG_A2MSTW2008LO_minbias_inelastic_high.py. Do you understand the structure?

When your job has finished running look at:

- your logfile: log.generate. An example log.generate output is here

See if you can answer the following questions:

- How would you change the centre-of-mass energy?
- How many events did you generate?
- Where was this specified?
- How many events did Pythia generate?
- Do they agree with the number of events you asked for generation?
- Which are the requirements of the filter?
- Which is its efficiency?
- How many events passed/failed the TestHepMC cuts?

Running a Sherpa Zee example

In this example we will run an on-the-fly Sherpa generation of $Z \to ee$.

```
cd <workdir>
mkdir Sherpa_Zee_example/
cd Sherpa_Zee_example/
```

As before, assuming you have added your lxplus ssh-keys to gitlab, you can export a particular MC card file using

```
git archive --remote=ssh://git@gitlab.cern.ch:7999/atlas-physics/pmg/mcjoboptions.git  HEAD:421xxx/421002 mc.Sherpa_Example2.py  | tar -x
```

or copy it directly from cvmfs:

```
cp /cvmfs/atlas.cern.ch/repo/sw/Generators/MCJobOptions/421xxx/421002/mc.Sherpa_Example2.py .
```

Now you can try running this:

```
Gen_tf.py --ecmEnergy=13000. --firstEvent=1  --maxEvents=-1 --randomSeed=1234 --jobConfig=421002 --outputEVNTFile=test_Zee_421002.EVNT.root
```

When your job has finished running look at:

- your logfile: log.generate. An example log file is attached here
- the jobOptions: mc.Sherpa_Example2.py

See if you can answer the following questions:

- How many events did you generate?
- What is the Sherpa cross section? (search for "cross-section" or "Total XS")
- Can you recognize the names of the weights variations included in the sample?
- Look in events.lhe, can you see the pp > Z > e+ e- hard scatter?

Which Feynman diagrams did the process considered? (only for curious people)

Let's modify the jobOptions to get an output with the Feynman diagrams:

1. Create a directory with the DSID:

   ```
   mkdir 421002
   ```

2. Place the jobOption inside the directory
3. Modify the jobOption to have an output folder with the diagrams. Add

   ```
   Print_Graphs mydiagrams
   ```

   after the line

   ```
   Process 93 93 ->11 -11 93{0}
   ```

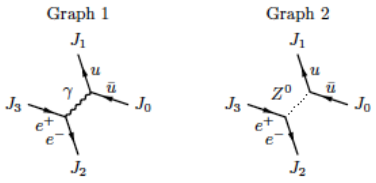
4. Run as before. This time the jobOption stored locally in 421002 will be read instead the one in gitlab.
5. Setup a recent texlive version

   ```
   export PATH=/afs/cern.ch/sw/XML/TL2016/bin/x86_64-linux/:$PATH
   ```

6. Run the script

   ```
   /cvmfs/atlas.cern.ch/repo/sw/software/21.6/sw/lcg/releases/LCG_88/MCGenerators/sherpa/2.2.8p3/x86_64-slc6-gcc62-opt/bin/plot_graphs.sh mydiagrams
   ```

7. Inspect the diagrams. An example is



Running Rivet from Gen_tf.py

Setup a new directory and run the generate transform with an additional flag to add a Rivet analysis at the end of the job. Since we are looking at Z production we will load the MC_ZINC Rivet analysis for inclusive Z production - the necessary flag is --rivetAnas=MC_ZINC

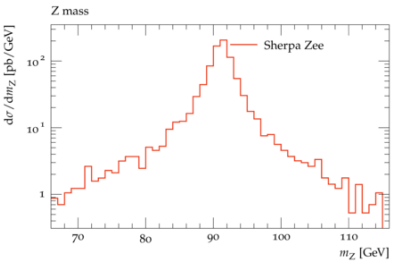
```
cd <workdir>
mkdir Sherpa_Zee_example_Rivet/
cd Sherpa_Zee_example_Rivet/
asetup 21.6.12,AthGeneration,here
source setupRivet.sh

Gen_tf.py --ecmEnergy=13000. --firstEvent=1  --maxEvents=-1 --randomSeed=1234 --jobConfig=421002 --outputEVNTFile=test_Zee_421002.EVNT.root --rivetAnas=MC_ZINC
```

Now you should have the same result as before but with an additional Rivet.yoda file. Yoda is the output format of Rivet. You can plot the MC_ZINC analysis plots by doing:

```
rivet-mkhtml Rivet.yoda:"Title=Sherpa Zee"
firefox plots/index.html &
```

If you select the MC_ZINC analysis link you should see a webpage where amongst other plots there should be a plot similar to the one below:



Generating a truth DxAOD using the derivation framework

In a new terminal window:

```
cd <workdir>
mkdir TRUTH3/
cd TRUTH3/
setupATLAS
asetup 21.2.65.0,AthDerivation,here
Reco_tf.py --inputEVNTFile ../Sherpa_Zee_example/test_Zee_421002.EVNT.root --outputDAODFile tutorial.pool.root --reductionConf TRUTH3
```

This will give you the output file DAOD_TRUTH3.tutorial.pool.root.

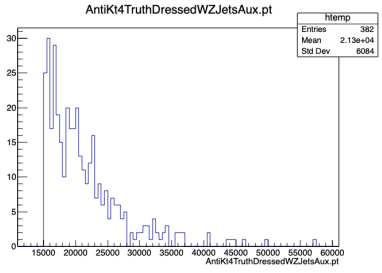
To have a very quick browse of your new truth derivation you can look at the contents with root:

```
root DAOD_TRUTH3.tutorial.pool.root
```

Then, at the root prompt:

```
CollectionTree->Draw("AntiKt4TruthDressedWZJetsAux.pt")
```

This will give you a plot of the jet pT that should look something like this:



Submitting a Generate transform to the grid

Usually you will not run MC event generation on your laptop! To produce large samples you need to use the grid. The submission to the grid is done using "Panda". There are many more instructions and details about running panda during the workshop and here: PandaAthena. But for this tutorial we just show a simple example below.
(You will need you grid certificate and password.)

```
mkdir pathena_test
cd pathena_test
setupATLAS -c SLC6
asetup  21.6.12,AthGeneration,here
lsetup panda


pathena --trf "Gen_tf.py --ecmEnergy=13000. --firstEvent=1 --maxEvents=-1 --randomSeed=%RNDM:0 --jobConfig=421002 --outputEVNTFile=%OUT.test_Zee_421002.EVNT.root" --outDS=user.$U
```

Old Workflow

Setting up your environment

To setup an appropriate Athena environment please execute the following setup commands in a fresh directory on lxplus:

```
setupATLAS
mkdir <workdir>
cd <workdir>
lsetup "asetup 19.2.5.31,here"
```

Running a Pythia8 minimum bias example

```
mkdir Pythia8_MB_example/
cd Pythia8_MB_example/
cp /cvmfs/atlas.cern.ch/repo/sw/Generators/MC15JobOptions/latest/share/DSID361xxx/MC15.361036.Pythia8_A2MSTW2008LO_minbias_inelastic.py .
```

Now you can run your first generation job! Execute the following command but instead of running it exactly as below, change the random seed to your date of birth first (DDMMYYYY). Then you can compare the difference in output with your neighbour.

```
Generate_tf.py --ecmEnergy=13000. --firstEvent=1 --runNumber=361036 --maxEvents=1000 --randomSeed=1234 --jobConfig=MC15.361036.Pythia8_A2MSTW2008LO_minbias_inelastic.py --outp
```

While the job is running, have a look at:

- the Generate_tf.py command you just executed. What are all the input arguments? Why?
- the jobOptions you just used: MC15.361036.Pythia8_A2MSTW2008LO_minbias_inelastic.py. Do you understand the structure?

When your job has finished running look at:

- your logfile: log.generate

See if you can answer the following questions:

- How would you change the centre-of-mass energy?
- How many events did you generate?
- Where was this specified?
- How many events passed/failed the TestHepMC cuts?

An example log file is attached here.

The output will be stored in the EVNT file test_MB_361036.EVNT.root. EVNT files can only be read/analysed using athena, with a joboption like (this will work in any athena release, including AthAnalysis), you just need to use the "POOLAccess" read mode (since EVNT is a POOL format file):

```
jps.AthenaCommonFlags.AccessMode = "POOLAccess"
jps.AthenaCommonFlags.FilesInput = ["test_MB_361036.EVNT.root"] #or use "--filesInput=test_MB_361036.EVNT.root"   on the command line
athAlgSeq += CfgMgr.YourAnalysisAlg() #whatever your algorithm is called which will analyse the McEventCollection
```

You can also convert the EVNT content into an xAOD format on-the-fly, by adding the following lines (**add them before your analysis algorithm in the main alg sequence**):

```
#put these lines BEFORE your AnalysisAlg is added to the athAlgSeq
athAlgSeq += CfgMgr.xAODMaker__xAODTruthCnvAlg(AODContainerName="GEN_EVENT",WriteTruthMetaData=False) #creates xAOD::TruthParticles, TruthVertices, TruthEvents
athAlgSeq += CfgMgr.xAODMaker__xAODEventInfoCnvAlg() #creates xAOD::EventInfo
```

see here for how.

But if you don't want to use athena to analyse this EVNT, you will need to convert it into a truth DxAOD to be read (see further down this tutorial)

Running a MadGraph+Pythia8 Zee example

In this example we will run an on-the-fly (OTF) MadGraph+Pythia8 generation of $Z \to ee$.

```
cd <workdir>
asetup 21.6.4,AthGeneration,here
mkdir MadGraphPythia8_Zee_example/
cd MadGraphPythia8_Zee_example/
```

Assuming you have added your lxplus ssh-keys to gitlab, then you can export a particular MC card file using

```
git archive --remote=ssh://git@gitlab.cern.ch:7999/atlas-physics/pmg/infrastructure/mc15joboptions.git HEAD:share/DSID429xxx MC15.429820.MadGraphPythia8EvtGen_A14NNPDF23LO_Zee_Np0.
```

Otherwise you can download it via the browser from the URL: https://gitlab.cern.ch/atlas-physics/pmg/infrastructure/mc15joboptions/blob/master/share/DSID429xxx/MC15.429820.MadGraphPythia8EvtGen_A14NNPDF23LO_Zee_Np0.py and copy it (via scp) to the working directory.

Now you can try running this:

```
Generate_tf.py --ecmEnergy=13000. --firstEvent=1 --runNumber=429820 --maxEvents=-1 --randomSeed=1234 --jobConfig=MC15.429820.MadGraphPythia8EvtGen_A14NNPDF23LO_Zee_Np0.py -
```

When your job has finished running look at:

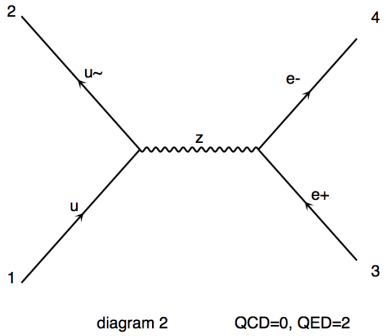
- your logfile: log.generate
- the jobOptions: MC15.429820.MadGraphPythia8EvtGen_A14NNPDF23LO_Zee_Np0.py

See if you can answer the following questions:

- How many events did you generate?
- How many events did Pythia8 try? Why is it significantly more than the final number of events?
- What is the MadGraph cross section? (search for "Cross-section")
- What is different in MadGraphPythia8_Zee_example/ compared to Pythia8_MB_example/?
- Look in events.lhe, can you see the pp > Z > e+ e- hard scatter?
- Look at the figures: PROC_sm-no_b_mass_0/SubProcesses/P0_qq_ll/matrix11.jpg

An example log file is attached here.

You should be able to see the Feynman diagrams produced by MadGraph, e.g.



Running Rivet from Generate_tf.py

Setup a new directory and run the generate transform with an additional flag to add a Rivet analysis at the end of the job. Since we are looking at Z production we will load the MC_ZINC Rivet analysis for inclusive Z production - the necessary flag is --rivetAnas=MC_ZINC

```
cd <workdir>
mkdir MadGraphPythia8_Zee_example_Rivet/
cd MadGraphPythia8_Zee_example_Rivet/
asetup 21.6.4,AthGeneration,here
source setupRivet.sh
```

Assuming you have added your lxplus ssh-keys to gitlab, then you can export a particular MC card file using

```
git archive --remote=ssh://git@gitlab.cern.ch:7999/atlas-physics/pmg/infrastructure/mc15joboptions.git HEAD:share/DSID429xxx MC15.429820.MadGraphPythia8EvtGen_A14NNPDF23LO_Zee_Np0.
```

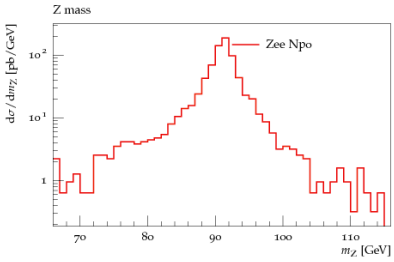
Otherwise you can download it via the browser from the URL: https://gitlab.cern.ch/atlas-physics/pmg/infrastructure/mc15joboptions/blob/master/share/DSID429xxx/MC15.429820.MadGraphPythia8EvtGen_A14NNPDF23LO_Zee_Np0.py and copy it (via scp) to the working directory.

```
Generate_tf.py --ecmEnergy=13000. --firstEvent=1 --runNumber=429820 --maxEvents=-1 --randomSeed=1234 --jobConfig=MC15.429820.MadGraphPythia8EvtGen_A14NNPDF23LO_Zee_Np0.py -
```

Now you should have the same result as before but with an additional Rivet.yoda file. Yoda is the output format of Rivet. You can plot the MC_ZINC analysis plots by doing:

```
rivet-mkhtml Rivet.yoda:"Title=Zee Np0"
firefox plots/index.html &
```

If you select the MC_ZINC analysis link you should see a webpage where amongst other plots there should be a plot similar to the one below:



Generating a truth DxAOD using the derivation framework

In a new terminal window:

```
cd <workdir>
mkdir TRUTH3/
cd TRUTH3/
setupATLAS
asetup 21.2.65.0,AthDerivation,here
Reco_tf.py --inputEVNTFile ../MadGraphPythia8_Zee_example/test_Zee_429820.EVNT.root --outputDAODFile tutorial.pool.root --reductionConf TRUTH3
```

This will give you the output file `DAOD_TRUTH3.tutorial.pool.root`.

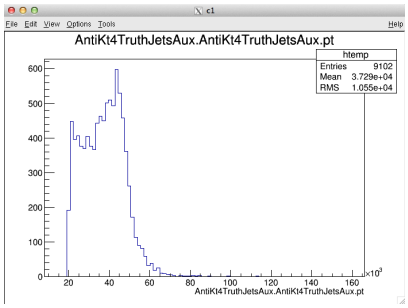
To have a very quick browse of your new truth derivation you can look at the contents with root:

```
root DAOD_TRUTH3.tutorial.pool.root
```

Then, at the root prompt:

```
CollectionTree->Draw("AntiKt4TruthJetsAux.AntiKt4TruthJetsAux.pt")
```

This will give you a plot of the jet pT that should look something like this:



Submitting a Generate transform to the grid

Usually you will not run MC event generation on your laptop! To produce large samples you need to use the grid. The submission to the grid is done using "Panda". There are many more instructions and details about running panda during the workshop and here: PandaAthena. But for this tutorial we just show a simple example below.
(You will need you grid certificate and password.)

```
mkdir pathena_test
cd pathena_test
setupATLAS -c SLC6
lsetup panda "asetup 19.2.5.31,here"
```

Note, because we are submitting to the grid, we will setup the ATLAS software in a container using SLC6 (not the now native Centos7 on LXPLUS). This is needed for the tutorial in June 2019, but should not be necessary once the grid nodes switch to Centos 7 as well.
Assuming you have added your lxplus ssh-keys to gitlab, then you can export a particular MC card file using

```
git archive --remote=ssh://git@gitlab.cern.ch:7999/atlas-physics/pmg/infrastructure/mc15joboptions.git  HEAD:share/DSID429xxx MC15.429820.MadGraphPythia8EvtGen_A14NNPDF23LO_Zee_Np0.
```

Otherwise you can download it via the browser from the URL: https://gitlab.cern.ch/atlas-physics/pmg/infrastructure/mc15joboptions/blob/master/share/DSID429xxx/MC15.429820.MadGraphPythia8EvtGen_A14NNPDF23LO_Zee_Np0.py and copy it (via scp) to the working directory.

```
pathena --trf "Generate_tf.py --ecmEnergy=13000. --firstEvent=1 --runNumber=429820 --maxEvents=-1 --randomSeed=%RNDM:0 --jobConfig=MC15.429820.MadGraphPythia8EvtGen_A14NNPDF2
```

Major updates:
-- LouiseHeelan - 2016-01-20

Responsible: LouiseHeelan
Last reviewed by: **Never reviewed**

- MC15.429820.MadGraphPythia8EvtGen_A14NNPDF23LO_Zee_Np0.py.txt: MC15.429820.MadGraphPythia8EvtGen_A14NNPDF23LO_Zee_Np0

I	Attachment	History	Action	Size	Date	Who
	MC15.361036.Pythia8_A2MSTW2008LO_minbias_inelastic.py.txt	r1	manage	0.5 K	2016-02-02 - 08:10	JoshMcFayden
	MC15.361500.MadGraphPythia8EvtGen_A14NNPDF23LO_Zee_Np0.py.txt	r1	manage	0.1 K	2016-02-02 - 08:10	JoshMcFayden
	MC15.429820.MadGraphPythia8EvtGen_A14NNPDF23LO_Zee_Np0.py.txt	r1	manage	3.0 K	2019-06-06 - 13:48	JamesWalder2
	MadGraphControl_Zjets_LO_Pythia8.py.txt	r1	manage	6.5 K	2016-02-02 - 08:10	JoshMcFayden
	Sherpa_Zee_example_graph.png	r1	manage	10.5 K	2019-10-23 - 15:29	AnaRosarioCuetoGomez
	Z_jetpT.png	r1	manage	54.1 K	2016-02-02 - 12:07	JoshMcFayden
	Z_jetpT_Sherpa.png	r1	manage	26.2 K	2019-10-23 - 15:55	AnaRosarioCuetoGomez
	Z_mass.png	r1	manage	10.5 K	2016-02-02 - 11:56	JoshMcFayden
	Z_mass_Sherpa.png	r1	manage	33.8 K	2019-10-23 - 15:58	AnaRosarioCuetoGomez
	Zee_feyn.png	r1	manage	35.7 K	2016-02-02 - 02:14	JoshMcFayden
	log.generate_MGPy8Zee	r1	manage	867.1 K	2016-02-02 - 02:57	JoshMcFayden
	log.generate_Py8MB	r1	manage	536.1 K	2016-02-02 - 02:19	JoshMcFayden
	log.generate_Py8MB_filter	r1	manage	372.7 K	2019-10-23 - 13:48	AnaRosarioCuetoGomez
	log.generate_SHZee	r1	manage	460.9 K	2019-10-25 - 11:07	AnaRosarioCuetoGomez
	mc.Py8EG_A2MSTW2008LO_minbias_inelastic_high.py.txt	r1	manage	0.7 K	2019-10-23 - 13:48	AnaRosarioCuetoGomez
	mc.Sherpa_Example2.py.txt	r1	manage	0.6 K	2019-10-23 - 15:25	AnaRosarioCuetoGomez

Topic revision: r23 - 2019-10-25 - AnaRosarioCuetoGomez