

MICHIGAN STATE  
UNIVERSITY

# Generator Software

28th of September 2015  
MC in ATLAS Tutorial

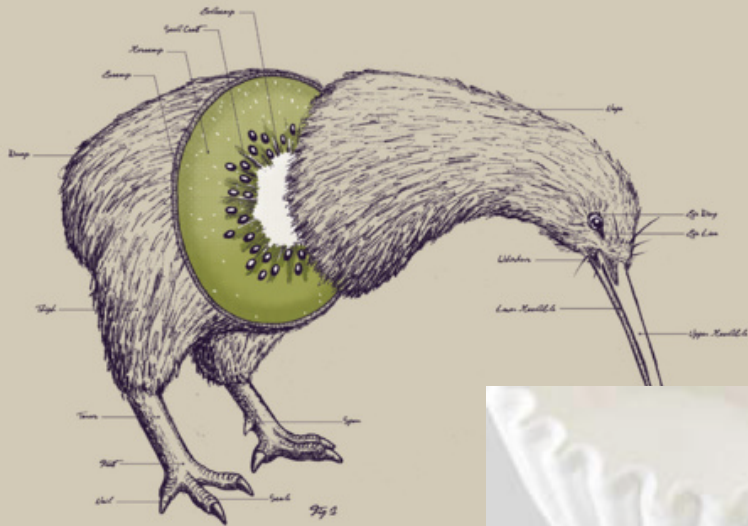
Daniel Hayden  
[daniel.hayden@cern.ch](mailto:daniel.hayden@cern.ch)



## Contents

- Cross-Sections
- Filters & Filtering Efficiencies.
- How to calculate and work with **k-Factors**.
- Let's take a look - **JobOptions**.
  - On-the-fly production.
  - Afterburners.
  - JobOption structure.
- Taking it for a spin - **How to test locally**.
- On your marks... **Get ready for the hands-on session**.

## Cross-Sections, Filters, and k-Factors





## Cross-Sections

- Your chosen generator will return the cross-section of the process being produced depending on the PDF, and generator-level settings.
- This cross-section is used to normalise the sample to a given integrated luminosity.
- Where to find the cross-section after local generation:

Generator	Filename	Line	Note
Pythia	log.generate	sum   [Process Code]   [X +/- X]	Default mb
Powheg	log.generate	sum   [Process Code]   [X +/- X]	Default mb
Sherpa	sherpa.log	Total XS is X pb +- ( X pb = X % )	Default nb
Madgraph	Webpage	-	Default nb
aMC@NLO	log.generate	sum   [Process Code]   [X +/- X]	Default mb
Herwig/++	log.generate	sum   [Process Code]   [X +/- X]	Default mb

N.B. when running e.g. Powheg+Pythia 8, you will get the  $\sigma$  from Pythia.

## Cross-Sections - Continued

- Once the sample has been run in production, these numbers are quoted in the metadata on AMI: <https://ami.in2p3.fr/>.

mc15\_13TeV.%PowhegPythia8%Zee%merge.AOD%

Search!

more fields +	logicalDatasetName Q	nFiles Q	totalEvents Q	totalSize Q	dataType Q
details	mc15_13TeV.361106.PowhegPythia8EvtGen_AZNLOCTEQ6L1_Zee.merge.AOD.e3601_a766_a777_r6282 RuciInfo - Provenance - Campaigns - GANGA export - Series	1953	19503000	7.699 TB	AOD

Details → Dataset\_Extra ( $\sigma$  unit always nb)

more fields +	field Q	value Q	createdBy Q	modifiedBy Q	lastModified Q	created Q
details	beam_energy	[6500000.0]	root	root	2015-09-11 16:00:33	2015-09-11 16:00:33
details	approx_crossSection	1.9012	root	root	2015-09-10 07:53:26	2015-09-10 07:53:26
details	approx_GenFiltEff	1.0000E+00	root	root	2015-09-10 07:53:26	2015-09-10 07:53:26
details	genFiltEff	1.0000E+00	root	root	2015-09-10 07:53:26	2015-09-10 07:53:26

## Aside: Log.generate

- You can find lots of useful info in the log file !
  - Version used: INFO using release [AtlasProduction-19.2.4.2]
  - E.g: If you run MG, can know about run\_card (and parameters), MG version, the MG path, the shower version and details on it
  - If Photos or Tauola used, details on that as well
  - Cross section: e.g. “MetaData: cross-section (nb)= 8.113e-06”
  - Filter efficiency:

Py:EvgenFilterSeq INFO Filter Efficiency = 1.000000 [20 / 20]

Py:EvgenFilterSeq INFO Weighted Filter Efficiency = 1.000000 [20.000000 / 20.000000]

MetaData: GenFiltEff = 1.000000e+00

## FixHepMC

In the log file, you will also find something like:

```
FixHepMC      INFO Removed X of 36501 particles because of loops.  
FixHepMC      INFO Removed Y of 36501 particles because of PDG ID 0.  
FixHepMC      INFO Removed Y of 36501 particles while cleaning decay chains.
```

FixHepMC lives in EvgenProdTools and removes possible „bad“ structures – it runs without you knowing it, but important to know is there for deep debugging!

Description for example in:

[http://atlas-computing.web.cern.ch/atlas-computing/links/nightlyDocDirectory/  
EvgenProdTools/html/classFixHepMC.html](http://atlas-computing.web.cern.ch/atlas-computing/links/nightlyDocDirectory/EvgenProdTools/html/classFixHepMC.html)

# TestHepMC

Informative only, gives info about several matters – listed in an explicative way in the log file – in particular energy balance and momentum balance are very important – the fraction of events with unbalanced is > 3% the job crashes ....

```

22:29:36 TestHepMC      INFO Events passed = 20, Events Failed = 0
22:29:36 TestHepMC      INFO Event rate with invalid Beam Particles = 0% (not included in test efficiency)
22:29:36 TestHepMC      INFO Event rate with beam particles and status not equal to 4 = 0% (not included in test efficiency)
22:29:36 TestHepMC      INFO Event rate with incorrect beam particle energies = 0% (not included in test efficiency)
22:29:36 TestHepMC      INFO Event rate with NaN (Not A Number) or inf found in the event record vertex positions = 0%
22:29:36 TestHepMC      INFO Event rate with vertices displaced more than 100~mm in transverse direction for particles with status codes 1
and 2 = 0% (not in
cluded in test efficiency)
22:29:36 TestHepMC      INFO Event rate with vertices displaced more than 100~mm in transverse direction for particles with status code
other than 1 and 2 =
0% (not included in test efficiency)
22:29:36 TestHepMC      INFO Event rate with vertices displaced more than 1000~mm = 0%
22:29:36 TestHepMC      INFO Event rate with NAN (Not A Number) or inf found in particle momentum values = 0%
22:29:36 TestHepMC      INFO Event rate with undecayed pi0's with status 1 or 2 = 0% (not included in test efficiency)
22:29:36 TestHepMC      INFO Event rate with unstable particles with no end vertex = 0% (not included in test efficiency)
22:29:36 TestHepMC      INFO Event rate with negative total energy like for tachyonic particles = 0% (not included in test efficiency)
22:29:36 TestHepMC      INFO Event rate with particles with improper decay properties = 0% (not included in test efficiency)
22:29:36 TestHepMC      INFO Event rate with undisplaced daughters of long lived hadrons = 0% (not included in test efficiency)
22:29:36 TestHepMC      INFO Event rate with non zero photon mass = 0% (not included in test efficiency)
22:29:36 TestHepMC      INFO Event rate with no energy balance = 0%
22:29:36 TestHepMC      INFO Event rate with no momentum balance = 0%
22:29:36 TestHepMC      INFO Event rate with negative energy particles = 0%
22:29:36 TestHepMC      INFO Event rate with tachyons = 0%
22:29:36 TestHepMC      INFO Event rate with stable or unstable particles with no parents = 0%
22:29:36 TestHepMC      INFO Event rate with unstable particle with no decay vertex = 0%
22:29:36 TestHepMC      INFO Event rate with undecayed Pi0's = 0%
22:29:36 TestHepMC      INFO Event rate with undisplaced decay daughters of displaced vertices = 0%
22:29:36 TestHepMC      INFO Event rate with particles with status 1 but lifetime < 0.00333333~ns = 0%
22:29:36 TestHepMC      INFO Event rate with energy sum of interacting particles non known by Geant4 above 100 MeV = 0%
22:29:36 TestHepMC      INFO Efficiency = 100%























































```



# Filters and Filtering Efficiencies

- Clarification:

- **Generator Phase-Space Cuts** → Does not imply a filtering efficiency.
- **Generator Filters** → After generation, events are thrown away according to certain criteria → Generator reports the total generated cross-section so filtering efficiency needs to be applied accordingly.

 <b>AsymJetFilter.cxx</b>	11.2 KB 	547099 
 <b>ATauFilter.cxx</b>	10.4 KB 	659883 
 <b>Boosted2DijetFilter.cxx</b>	6.1 KB 	627918 
 <b>BoostedHadTopAndTopPair.cxx</b>	7.3 KB 	652705 
 <b>BSignalFilter.cxx</b>	26.7 KB 	691638 
 <b>BSubstruct.cxx</b>	8.8 KB 	652705 
 <b>ChargedTracksFilter.cxx</b>	1.8 KB 	602715 
 <b>DecayLengthFilter.cxx</b>	2.9 KB 	601190 
 <b>DecayModeFilter.cxx</b>	12.1 KB 	660873 
 <b>DecayPositionFilter.cxx</b>	2.3 KB 	613665 
 <b>DecaysFinalStateFilter.cxx</b>	3.4 KB 	663227 
 <b>DiBjetFilter.cxx</b>	6.2 KB 	659883 
 <b>DiLeptonMassFilter.cxx</b>	4.2 KB 	494807 
 <b>DiPhotonFilter.cxx</b>	6.6 KB 	547099 
 <b>DirectPhotonFilter.cxx</b>	6.0 KB 	657398 
 <b>DstD0K3piFilter.cxx</b>	6.5 KB 	547099 
 <b>DuplicateEventFilter.cxx</b>	923 bytes 	654009 
 <b>ElectronFilter.cxx</b>	887 bytes 	494807 

# Filters and Filtering Efficiencies - Continued

source: **Generators / GeneratorFilters / trunk / src / TTbarWToLeptonFilter.cxx**

Last change on this file was **547099**, checked in by bruneli, **2 years ago**

see [ChangeLog](#), intermediate commit

File size: 6.4 KB

Line	
1	#include "GeneratorFilters/TTbarWToLeptonFilter.h"
2	
3	TTbarWToLeptonFilter::TTbarWToLeptonFilter(const std::string& name, ISvcLocator* pSvcLocator)
4	: GenFilter(name,pSvcLocator)
5	{
6	declareProperty("Ptcut", m_Ptmin=200000.);
7	declareProperty("NumLeptons", m_numLeptons=-1); // Negative for >0, positive integers for the specific number
8	}
9	
10	
11	StatusCode TTbarWToLeptonFilter::filterEvent() {
12	int N_quark_t = 0;
13	int N_quark_tbar = 0;
14	int N_quark_t_all = 0;
15	int N_quark_tbar_all = 0;
16	int N_pt_above_cut = 0;
17	
18	for (McEventCollection::const_iterator itr = events()->begin(); itr!=events()->end(); ++itr) {
19	const HepMC::GenEvent* genEvt = (*itr);
20	for (HepMC::GenEvent::particle_const_iterator pitr = genEvt->particles_begin(); pitr != genEvt->particles_end(); ++pitr) {
21	if (fabs((*pitr)->pdg_id()) == 6) {
22	if ( (*pitr)->pdg_id() == 6 ) N_quark_t_all++;
23	if ( (*pitr)->pdg_id() == -6 ) N_quark_tbar_all++;
24	
25	int n_daughters = 0;
26	
27	HepMC::GenParticle * mcpart = (*pitr);
28	const HepMC::GenVertex * decayVtx = mcpart->end_vertex();
--	

We'll see later how this is used in your jobOptions.

## Filters and Filtering Efficiencies - Continued

Generator Phase-Space Cuts are **NOT** Filters.

```

1  #-----
2  # Powheg Z setup starting from ATLAS defaults
3  #-----
4  include('PowhegControl/PowhegControl_Z_Common.py')
5  PowhegConfig.vdecaymode = 1    # ee
6
7  # Configure Powheg setup
8  PowhegConfig.ptsqmin = 4.0 # needed for A2NLO tune
9  PowhegConfig.nEvents *= 1.1 # increase number of generated events by 10%
10 PowhegConfig.running_width = 1
11 PowhegConfig.mass_low = 120.
12 PowhegConfig.mass_high = 180.

```

Powheg (DY)

```

1 include('MC15JobOptions/Pythia8_A14_NNPDF23LO_EvtGen_Common.py')
2
3 genSeq.Pythia8.Commands += ["WeakSingleBoson:ffbar2gmZ = on",
4                             "23:onMode = off",                # turn off all decays modes
5                             "23:onIfAny = 11",                # turn on the ee decay mode
6                             "PhaseSpace:mHatMin = 120.",        # lower invariant mass
7                             "PhaseSpace:mHatMax = 180." ]       # upper invariant mass

```

Pythia 8 (DY)

## More Generator Information

Very well documented [and maintained!] twiki pages.

<https://twiki.cern.ch/twiki/bin/view/AtlasProtected/McGeneratorsForAtlas>



## k-Factors

- What order is my process calculated to?
  - Leading Order Generators: Pythia 6/8, Herwig, ...
  - Next-to-Leading Order: Powheg, Sherpa, aMC@NLO, Madgraph, ...
  - Also depends on your PDF choice!
    - ★ LO: CTEQ6L1, NNPDF23LO, MSTW2008LO, ...
    - ★ NLO: CT10NLO, NNPDF3.0NLO, ...
- How do I increase the accuracy of my generated process?
  - Higher-Order Calculators exist, such as FEWZ, TOPNNLO, etc.
  - Able to calculate some processes up to NNLO accuracy.

## k-Factors - Continued

- How can this be calculated for my lower-order sample?

- **Global k-Factor:**

- ★ Numerator = Higher-Order Calculation of Process  $\sigma$ , given specific truth-level criteria (lepton  $\eta$ ,  $p_T$ , etc).
- ★ Denominator = Your sample  $\sigma$  in the same phase-space region.
- ★ Already done for you in some cases (PMG Core Samples):

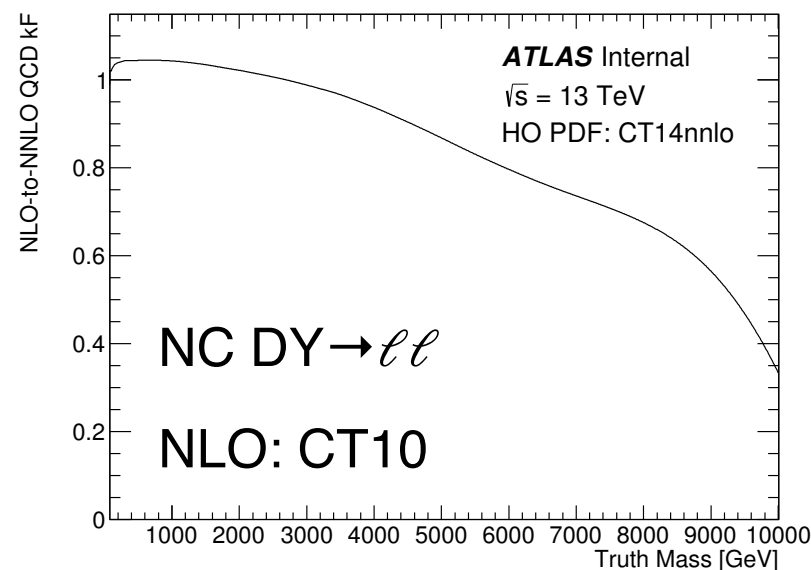
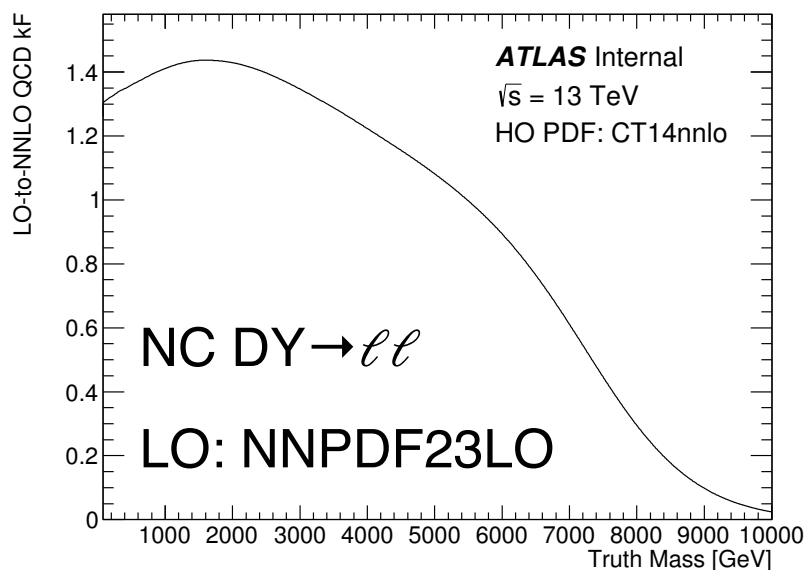
[TWiki](#) > 
 [AtlasProtected Web](#) > 
 [AtlasPhysics](#) > 
 [PhysicsModellingGroup](#) > 
 [CentralMC15ProductionList](#) > 
 [XsecSummaryTTbar](#)

<https://twiki.cern.ch/twiki/bin/view/AtlasProtected/XsecSummaryTTbar>

DSID	Sample	AMIXsec	BR or <b>FileEff</b>	Total higher order xsec	K-factor	Higher order xsec sample
301528	mc15_13TeV.301528.PowhegPythiaEvtGen_P2012_ttbar_hdamp172p5_nonallhad_mtt_1	696.25	0.00471933	831.76	1.1949	3.92624246409
301529	mc15_13TeV.301529.PowhegPythiaEvtGen_P2012_ttbar_hdamp172p5_nonallhad_mtt_2	696.25	0.00194400	831.76	1.1949	1.617309099
301530	mc15_13TeV.301530.PowhegPythiaEvtGen_P2012_ttbar_hdamp172p5_nonallhad_mtt_3	696.23	0.00086308	831.76	1.1949	0.718018024919
301531	mc15_13TeV.301531.PowhegPythiaEvtGen_P2012_ttbar_hdamp172p5_nonallhad_mtt_4	696.24	0.00051910	831.76	1.1949	0.431858588062
301532	mc15_13TeV.301532.PowhegPythiaEvtGen_P2012_ttbar_hdamp172p5_nonallhad_mtt_5	696.25	0.00030919	831.76	1.1949	0.257230349959
429007	mc15_13TeV.429007.PowhegPythia_P2012_ttbar_hdamp172p5_nonallhad	696.31	0.54316	831.76	1.19417355255	451.64568
407009	mc15_13TeV.407009.PowhegPythiaEvtGen_P2012CT10_ttbarHT6c_1k_hdamp172p5_nonAH	696.22	0.022921	831.76	1.1949	19.068284245
407010	mc15_13TeV.407010.PowhegPythiaEvtGen_P2012CT10_ttbarHT1k_1k5_hdamp172p5_nonAH	696.22	0.003204335	831.76	1.1949	2.66572883366
407011	mc15_13TeV.407011.PowhegPythiaEvtGen_P2012CT10_ttbarHT1k5_hdamp172p5_nonAH	696.23	0.000565234	831.76	1.1949	0.470232423758
407012	mc15_13TeV.407012.PowhegPythiaEvtGen_P2012CT10_ttbarMET200_hdamp172p5_nonAH	696.22	0.009219475	831.76	1.1949	7.66980366869

## k-Factors - Continued

- How can this be calculated for my lower-order sample?
  - **Variable Dependent k-Factor** (i.e. Dilepton Mass):
    - ★ Numerator = Higher-Order Calculation of Process  $\sigma$ , given specific truth-level criteria (lepton  $\eta$ ,  $p_T$ , etc), **per mass-bin**.
    - ★ Denominator = Your sample  $\sigma$  in the same phase-space region, **per mass-bin**.
    - ★ Usually done by people in specific analyses, for the case above Uta Klein performs these calculations.



## Bringing it all together - $\sigma$ , filtering efficiency, kF

- To normalise my sample to a unit of integrated luminosity, I need:

$$\text{Event Weight} = (\sigma * \varepsilon * kF) / N$$

$\sigma$  = Generator Cross-Section.

$\varepsilon$  = Generator Filtering Efficiency.

kF = Calculated k-Factor.

N = Total N Generated Events.

If  $\sigma$  is in picobarns, then the result of the weighted spectrum will correspond to one inverse picobarn.



# Let's take a look - JobOptions



## On-The-Fly Production

- The ATLAS MC Group has gone to great lengths to simplify the process of writing jobOptions and producing samples efficiently.
- One large area of this is called “On-The-Fly” or OTF production.
- Currently available for Powheg, Madgraph, Alpgen, aMC@NLO.

Old Way	OTF Way
Download powhegbox, setup your process and generate truth events (LHE file).	Write a jobOption file to call on an ATLAS maintained version of powhegbox (on afs).
Write a jobOption file to read in the LHE file, and shower the events with e.g. Pythia 8.	Events are generated and then directly showered, all within the production system.

- OTF Advantages:
  - Everyone uses same version (documented, reproducible).
  - User options to change generator settings from the jobOption.
  - Much smaller preparation overhead, less mistakes / confusion.

## After-Burners

### Photos++

- Program to simulate effective radiative decays.
- Only used for “Precision” EW processes such as W and Z.
- Otherwise Pythia 8 and Herwig++ are actually quite accurate.

### Tauola [Warning: Not advised after C++ Generators]

- MC Generator dedicated to tau-lepton decays.
- Includes matrix element reweighting for Tau decays.
- Only Fortran Generators and by EvtGen. Can lead to HEPMC failures.

### EvtGen

- Particle decay simulator, designed for the needs of B-physics.
- Includes spin effects and maintains proper decay angles.

[Link to JobOption on SVN.](#)

## Powheg

```

1  #-----
2  # EVGEN configuration
3  #-----
4  evgenConfig.description = 'POWHEG+Pythia6 ttbar production with Powheg hdamp equal top mass,
5  evgenConfig.keywords   = [ 'SM', 'top', 'ttbar', 'lepton' ]
6  evgenConfig.contact    = [ 'james.robinson@cern.ch', 'onofrio@liverpool.ac.uk' ]
7
8  if runArgs.trfSubstepName == 'generate' :
9
10     include('PowhegControl/PowhegControl_tt_Common.py')    <----- Process Fragment
11     PowhegConfig.topdecaymode = 22222
12     PowhegConfig.hdamp        = 172.5                      <----- User Options
13     # compensate filter efficiency
14     PowhegConfig.nEvents      *= 3.                        <----- ~Counteract  $\epsilon$ 
15     PowhegConfig.generateRunCard()
16     PowhegConfig.generateEvents()                          <----- Run Powheg OTF
17
18  #-----
19  # Pythia6 (Perugia2012) showering
20  #-----
21     include('MC15JobOptions/PowhegPythia_Perugia2012_Common.py') <-- Showering / Tune
22     include('MC15JobOptions/Pythia_Tauola.py')
23     include('MC15JobOptions/Pythia_Photos.py')                <----- Afterburners
24
25  #-----
26  # Event filter
27  #-----
28     include('MC15JobOptions/TTbarWToLeptonFilter.py')          <----- Remember Our Filter?
29     filtSeq.TTbarWToLeptonFilter.NumLeptons = -1
30     filtSeq.TTbarWToLeptonFilter.Ptcut = 0.                    <----- User Filtering Options
31
32  # Run EvtGen as afterburner
33  include('MC15JobOptions/Pythia_Powheg_EvtGen.py')             <----- EvtGen

```



[Link to JobOption on SVN](#)

Pythia 8

Leading-Order Generator, so no need for pre-generation step.

```

1 include('MC15JobOptions/Pythia8_A14_NNPDF23LO_EvtGen_Common.py') (1)
2
3 genSeq.Pythia8.Commands += ["WeakSingleBoson:ffbar2gmZ = on",
4                             "23:onMode = off",                # turn off all decays modes
5                             "23:onIfAny = 11",                # turn on the ee decay mode
6                             "PhaseSpace:mHatMin = 120.",        # lower invariant mass (2)
7                             "PhaseSpace:mHatMax = 180."],      # upper invariant mass
8
9 evgenConfig.description = "Pythia 8 DY->ee production with NNPDF23LO tune"
0 evgenConfig.contact = ["Daniel Hayden <daniel.hayden@cern.ch>"]
1 evgenConfig.keywords = ["SM", "drellYan", "electroweak", "2electron"] (3)
2 evgenConfig.generators += ["Pythia8"]
3

```

1. PDF and Tune.
2. Generator Settings / Phase-Space Cuts.
3. MetaData.

[Link to JobOption on SVN.](#)

[Madgraph](#)

**Last change** on this file was **667071**, checked in by mcfayden,  
Fix MGPpy8 Vjets control file for renamed Py8 CKKWL fragment  
**File size:** 6.4 KB

```

Line
1 from MadGraphControl.MadGraphUtils import *
2
3 # General settings
4 nevents=150000
5 mode=0
6 nJobs=1
7 gridpack_dir=None
8 gridpack_mode=False
9 cluster_type=None
10 cluster_queue=None
11
12 # MG Particle cuts
13 mllcut=40
14
15 # Merging settings
16 maxjetflavor=5
17 ickkw=0
18 nJetMax=4
19 ktDurham=30
20 dparameter=0.4
21
22
23 ### DSID lists (extensions can include filters, sys
24 Zee_5fl_Np0=[361500]
25 Zee_5fl_Np1=[361501]
26 Zee_5fl_Np2=[361502]
27 Zee_5fl_Np3=[361503]
28 Zee_5fl_Np4=[361504]
29
30 Zmumu_5fl_Np0=[361505]
31 Zmumu_5fl_Np1=[361506]
32 Zmumu_5fl_Np2=[361507]
33 Zmumu_5fl_Np3=[361508]
34 Zmumu_5fl_Np4=[361509]

```

## Setup and User Settings

```

51 ### Electrons
52 if runArgs.runNumber in Zee_5fl_Np0:
53     mgproc="generate p p > e+ e- @0"
54     name='Zee_Np0'
55     process="pp>e+e-"
56 elif runArgs.runNumber in Zee_5fl_Np1:
57     mgproc="generate p p > e+ e- j @1"
58     name='Zee_Np1'
59     process="pp>e+e-"
60 elif runArgs.runNumber in Zee_5fl_Np2:
61     mgproc="generate p p > e+ e- j j @2"
62     name='Zee_Np2'
63     process="pp>e+e-"
64 elif runArgs.runNumber in Zee_5fl_Np3:
65     mgproc="generate p p > e+ e- j j j @3"
66     name='Zee_Np3'
67     process="pp>e+e-"
68 elif runArgs.runNumber in Zee_5fl_Np4:
69     mgproc="generate p p > e+ e- j j j j @4"
70     name='Zee_Np4'
71     process="pp>e+e-"
72
73     nevents=5000
74     gridpack_mode=True
75     gridpack_dir='madevent/'
76
77     mode=1
78     cluster_type='pbs'
79     cluster_queue='medium'
80     nJobs=20

```

## Set Processes

[Link to JobOption on SVN.](#)

Madgraph

## Write MG5 Input Card

```

186 fcard = open('proc_card_mg5.dat','w')
187 fcard.write("""
188 import model sm-no_b_mass
189 define p = g u c d s b u- c- d- s- b-
190 define j = g u c d s b u- c- d- s- b-
191 """+mgproc+""")
192 output -f
193 """)
194 fcard.close()
195
196
197 beamEnergy=-999
198 if hasattr(runArgs,'ecmEnergy'):
199     beamEnergy = runArgs.ecmEnergy / 2.
200 else:
201     raise RuntimeError("No center of mass energy found.")

```

```

206 #Fetch default LO run_card.dat and set parameters
207 extras = { 'lhe_version' : '2.0',
208            'cut_decays'   : 'F',
209            'pdlabel'      : "'nn23lo1'",
210            #              'pdlabel'      : "'lhapdf'",
211            #              'lhaid'       : 247000,
212            'maxjetflavor' : maxjetflavor,
213            'asrwtflavor'  : maxjetflavor,
214            'ickkw'        : 0,
215            'ptj'          : 20,
216            'ptb'          : 20,
217            'mml1'         : mllcut,
218            'mmjj'         : 0,
219            'drjj'         : 0,
220            'drll'         : 0,
221            'drjl'         : 0.4,
222            'ptl'          : 0,
223            'etal'         : 10,
224            'etab'         : 6,
225            'etaj'         : 6,
226            'ktdurham'     : ktdurham,
227            'dparameter'   : dparameter }

```

[Link to JobOption on SVN.](#)

[Madgraph](#)

```

229 build_run_card(run_card_old=get_default_runcard(),run_card_new='run_card.dat',
230               nevents=nevents,rand_seed=runArgs.randomSeed,beamEnergy=beamEnergy,xqcut=0.,
231               extras=extras)
232
233 print_cards()
234 process_dir = new_process(grid_pack=gridpack_dir)
235 generate(run_card_loc='run_card.dat',param_card_loc=None,mode=mode,njobs=nJobs,proc_dir=process_dir,
236          grid_pack=gridpack_mode,gridpack_dir=gridpack_dir,cluster_type=cluster_type,cluster_queue=cluster_queue,
237          nevents=nevents,random_seed=runArgs.randomSeed)
238 arrange_output(proc_dir=process_dir,outputDS=stringy+'._00001.events.tar.gz')
239
240
241
242 ##### Shower
243 evgenConfig.description = 'MadGraph_'+str(name)
244 evgenConfig.keywords+=['Z','electron','jets','drellYan']
245 evgenConfig.inputfilecheck = stringy
246 runArgs.inputGeneratorFile=stringy+'._00001.events.tar.gz'
247
248 include("MC15JobOptions/Pythia8_A14_NNPDF23LO_EvtGen_Common.py")
249 include("MC15JobOptions/Pythia8_MadGraph.py")
250
251
252 PYTHIA8_nJetMax=nJetMax
253 PYTHIA8_TMS=ktdurham
254 PYTHIA8_Dparameter=dparameter
255 PYTHIA8_Process=process
256 PYTHIA8_nQuarksMerge=maxjetflavor
257 include("MC15JobOptions/Pythia8_CKKWL_kTMerge.py")

```

Run MG5,  
Showering,  
and Meta Data



# General - JobOptions on SVN

<https://svnweb.cern.ch/trac/atlasoff/browser/Generators/MC15JobOptions/trunk/share>

Wiki

Timeline

Roadmap

Browse Source

View Tickets

New Ticket

Search

[Last Change](#)
[Revision Log](#)

source: **Generators / MC15JobOptions / trunk / share**

View revision:

View diff against:

Name ▲	Size	Rev	Age	Author	Last Change
<div> <div>⤴</div> <div>../</div> </div>					
<div> <div>▸</div> <div> <div>📁</div> <div>DSID300xxx</div> </div> </div>		685189	2 months	mcjo	Add 300001-4,100,201-2,400-3, replace 301400-81
<div> <div>▸</div> <div> <div>📁</div> <div>DSID301xxx</div> </div> </div>		696291	5 days	mcjo	Add 304039-53,301-6, 301820-33, update 303349-436, 341462
<div> <div>▸</div> <div> <div>📁</div> <div>DSID302xxx</div> </div> </div>		696504	4 days	mcjo	Add 423003, 410009, 302066, 423099,107-12
<div> <div>▸</div> <div> <div>📁</div> <div>DSID303xxx</div> </div> </div>		696286	5 days	mcjo	update 303349-436
<div> <div>▸</div> <div> <div>📁</div> <div>DSID304xxx</div> </div> </div>		696291	5 days	mcjo	Add 304039-53,301-6, 301820-33, update 303349-436, 341462
<div> <div>▸</div> <div> <div>📁</div> <div>DSID305xxx</div> </div> </div>		636762	9 months	jmonk	More DSID blocks
<div> <div>▸</div> <div> <div>📁</div> <div>DSID306xxx</div> </div> </div>		636762	9 months	jmonk	More DSID blocks
<div> <div>▸</div> <div> <div>📁</div> <div>DSID307xxx</div> </div> </div>		636762	9 months	jmonk	More DSID blocks
<div> <div>▸</div> <div> <div>📁</div> <div>DSID308xxx</div> </div> </div>		636762	9 months	jmonk	More DSID blocks
<div> <div>▸</div> <div> <div>📁</div> <div>DSID309xxx</div> </div> </div>		636762	9 months	jmonk	More DSID blocks
<div> <div>▸</div> <div> <div>📁</div> <div>DSID310xxx</div> </div> </div>		636762	9 months	jmonk	More DSID blocks
<div> <div>▸</div> <div> <div>📁</div> <div>DSID311xxx</div> </div> </div>		636762	9 months	jmonk	More DSID blocks
<div> <div>▸</div> <div> <div>📁</div> <div>DSID312xxx</div> </div> </div>		636762	9 months	jmonk	More DSID blocks

- If you know the DSID of the jobOption, look in the appropriate directory using the first 3-digits.
- DSID blocks are organised according to physics and CP group. Makes grouping and lookup easier.

## MC15 production

- Relevant info about MC15 production in <https://twiki.cern.ch/twiki/bin/view/AtlasProtected/AtlasMcProductionMC15>
- MC15 uses the Generate\_tf.py script
- Production job option names are strictly enforced: any generator used except Photos, Tauola, or Fortran HERWIG, must be listed in the first "free" part of the JO name, before the first underscore.
- The evgenConfig.keywords values are now restricted to a fixed list to avoid minor variations and near-duplications: this will make the AMI search more useful. The available keywords are listed in [MC15JobOptions/common/evgenkeywords.txt](#)
- For fortran showers / generators used together with [EvtGen](#), the steering afterburn must be added to the Generate\_tf command (as --steering="afterburn")
  - This will produce a second log file, log.afterburn
- Synthax to use filters (filtSeq) given in twiki

## Taking it for a Spin - Testing Locally



## Release set up and versions

- MC generators live in Release 19

- For set up:

`asetup 19.2.4.11, AtlasProduction, here`

→ for a 4-digit official cache (latest shown)

`asetup 19.2.X.Y-VAL, rel_6, here`

→ for a nightly

`asetup 19.2.4.10.2, MCProd, here`

→ for a 5-digit cache (special generators, generator versions in validation)

- To see which version you are using, e.g.:

`cmt show versions External/Pythia8`

`cmt show versions External/Powheg`

`cmt show versions External/Sherpa`

`cmt show versions External/Herwigpp`

## How do I know what setup to use?

- A) Look on the generator specific [twiki page](#).
- B) Useful link from Jose: <http://atlas-project-mc-production.web.cern.ch/atlas-project-mc-production/Generators/AtlasProduction>
- C) Setup in a release and check / Ask the production team.

E.g. “I need to use Powheg-00-03-04, what release should I use?”

### Recommended releases

Recent releases that one can setup and run Powheg OTF in. See [this section](#) for the corresponding installation for each tag.

Rel19 Release	External/Powheg Tag	Generators/PowhegControl Tag	Comment
19.2.4.11	Powheg-00-03-04	PowhegControl-00-02-15	<b>Latest release</b>
19.2.4.6	Powheg-00-03-03	PowhegControl-00-02-13	-

Powheg-00-03-01

19.2.3.8

Powheg-00-03-02

19.2.3.9, 19.2.3.10, 19.2.3.11

Powheg-00-03-04

19.2.4.11



## Testing Locally - JobOption → Truth xAOD

```
# Run EVGEN
export ATLAS_LOCAL_ROOT_BASE=/cvmfs/atlas.cern.ch/repo/ATLASLocalRootBase
source ${ATLAS_LOCAL_ROOT_BASE}/user/atlasLocalSetup.sh
asetup 19.2.3.5,64,here
export JOBOPTSEARCHPATH=/cvmfs/atlas.cern.ch/repo/sw/Generators/MC15JobOptions/latest/common:$JOBOPTSEARCHPATH
```

```
COM_ENERGY = 13000           # GeV
RUN_NUMBER = 999999         # DSID of your jobOption.
START_EVENT = 1             # Starting Event
NUMBER_EVENTS = 1000        # Maximum Number of Events
RANDOM_SEED = 1234           # Starting Seed.
INPUT_JOBOPTIONS = MC15.301000.PowhegPythia8EvtGen_AZNLOCTEQ6L1_DYee_120M180.py
OUTPUT_NTUPLE = EVNT.root   # Output file name
```

```
Generate_tf.py --ecmEnergy $COM_ENERGY --runNumber $RUN_NUMBER --firstEvent
$START_EVENT --maxEvents $NUMBER_EVENTS --randomSeed $RANDOM_SEED --jobConfig
$INPUT_JOBOPTIONS --outputEVNTFile $OUTPUT_NTUPLE
```

```
# If you need a specific release of JobOptions you can also append the line below, but by default the latest will be taken:
# --evgenJobOpts=/afs/cern.ch/atlas/software/kits/EvgenJobOpts/MC15JobOpts-00-00-15_v0.tar.gz
```

## Testing Locally - Truth xAOD → Truth DxAOD

# Run EVGEN -> Truth DxAOD

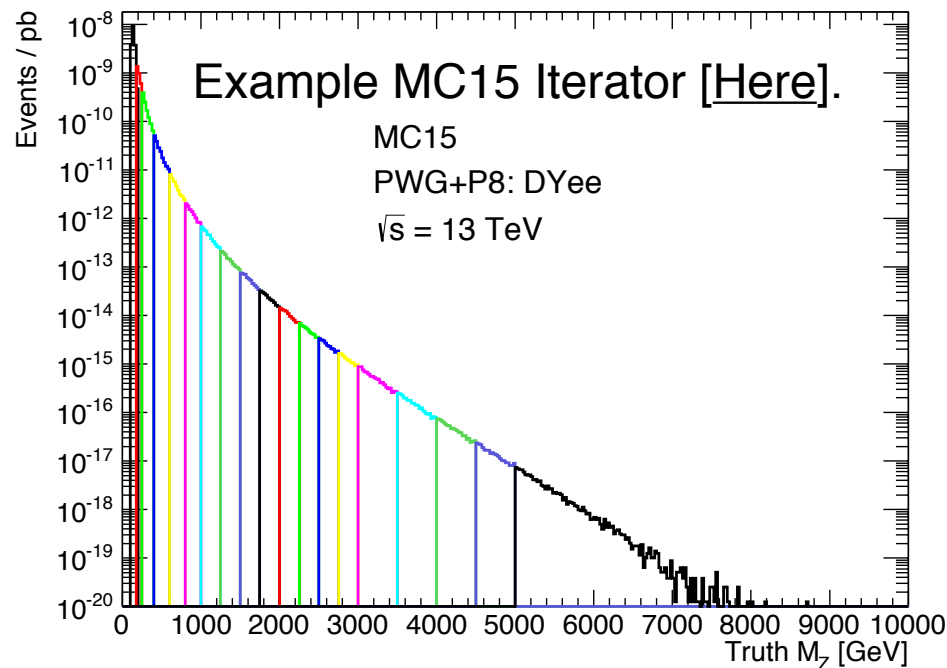
asetup 20.1.X.Y-VAL,rel\_1,AtlasDerivation,gcc48,here --nightliesarea=/afs/cern.ch/atlas/software/builds/nightlies

INPUT\_NTUPLE = EVNT.root      **# (Now the) Input file name**

OUTPUT\_xAOD = TRUTH.root      **# Derivation Truth DxAOD file name**

Reco\_tf.py --inputEVNTFile \$INPUT\_NTUPLE --outputDAODFile \$OUTPUT\_xAOD --reductionConf TRUTH0

## Testing Locally - Truth DxAOD → Simple Validation Plots



## Testing [Not so] Locally

# Run EVGEN on the GRID

```
pathena --trf "Generate_tf.py --ecmEnergy=13000. --runNumber=999999  
--firstEvent=1 --maxEvents=-1 --randomSeed=%RNDM:10  
--jobConfig=MC15JobOptions/MC15.999999.MyJobOptions.py  
--outputEVNTFile=EVNT.root" --outDS="user.$USER.TestEVNT_Date"  
--split=10
```

# Run EVGEN -> Truth DxAOD

```
pathena --trf "Reco_tf.py --inputEVNTFile=EVNT.root  
--outputDAODFile=truth.pool.root --reductionConf TRUTH1"  
--outDS user.$USER.TestTRUTH_Date  
--inDS user.$USER.TestEVNT_Date --noBuild --long --nGBPerJob=10  
--extOutFile=DAOD_TRUTH1.truth.pool.root
```

## Including Input Files

- All cases where inputConfig files are needed (LHE inputs or pre-integrations as for sherpa), this needs to be specified in the command.
- To work in production, this has to be added to the list in this file:
- <https://svnweb.cern.ch/trac/atlasoff/browser/Generators/MC15JobOptions/trunk/share/evgeninputfiles.csv>
- Run with input files:

```
Generate_tf.py --randomSeed=12345 --runNumber=361468 --ecmEnergy=13000 --
jobConfig=MC15.361468.Sherpa_CT10_Zee_Mll10to40_Pt0_70_BVeto.py --inputGenConfFile=../inputs/group.phys-
gener.sherpa020100.361468.Sherpa_CT10_Zee_Mll10to40_Pt0_70_BVeto_13TeV.TXT.mc15_v1._00001.tar.gz --
firstEvent=1 --maxEvents=10 --outputEVNTFile=test.sherpa.root
```

On your marks...

Getting Ready for the Hands-on Session





## Getting Ready for the Hands-on Session

- You will be having a go at running MadGraph5\_aMC@NLO in LO mode.
- Start by setting up athena according to the link below:  
[https://twiki.cern.ch/twiki/bin/viewauth/AtlasProtected/MadGraph5aMCA@NLOForAtlas#Athena\\_setup](https://twiki.cern.ch/twiki/bin/viewauth/AtlasProtected/MadGraph5aMCA@NLOForAtlas#Athena_setup)
- You can also take a look at the LHE-only jobOptions here:  
[https://twiki.cern.ch/twiki/bin/view/AtlasProtected/MadGraph5aMCA@NLOForAtlas#LO\\_on\\_the\\_fly\\_generation](https://twiki.cern.ch/twiki/bin/view/AtlasProtected/MadGraph5aMCA@NLOForAtlas#LO_on_the_fly_generation)

Try to get athena setup,  
and you could event set  
the jobOptions running.  
Then re-convene after  
the Coffee Break!

