

# Table of Contents

<b>Josh McFayden</b> .....	<b>1</b>
<b>Setting up release from MR in r21</b> .....	<b>2</b>
<b>To-do table test</b> .....	<b>3</b>
<b>MC production notes</b> .....	<b>4</b>
Getting started with evgen.....	4
Job submission script.....	4
<b>MC production coordinator notes</b> .....	<b>6</b>
Meetings.....	6
Useful webpages.....	6
Misc.....	6
tarball dir.....	6
Permissions.....	6
<b>SUSY MC contact notes</b> .....	<b>7</b>
Workflow.....	7
BIG Spreadsheet columns/colours.....	8
Links.....	8
Misc.....	8
Tags.....	8
Bugfix:.....	9
Replacing existing jOs:.....	9
New.py:.....	9
Sherpa integration grids.....	9
UEEE3 in 13.....	9
<b>Sherpa</b> .....	<b>10</b>
<b>LondonFramework</b> .....	<b>11</b>
<b>Direct stop 0-lepton 2012 8TeV data</b> .....	<b>12</b>
Technicalities.....	12
Code.....	12
Main changes wrt 2011 object definitions.....	12
<b>trigEmulator</b> .....	<b>14</b>
Basic Instructions.....	14
Setting up.....	14
Feeding branches.....	14
Usage.....	14
Files:.....	15

# Josh McFayden

# Setting up release from MR in r21

```
mkdir testdir
cd testdir
mkdir source build run
cd build
setupATLAS
acmSetup AthGeneration,21.6,latest
acm sparse_clone_project athena zmarshal/athena.git 21.6_MGC_Update
acm add_pkg athena/Generators/MadGraphControl
acm compile
```

# To-do table test

ExtensionsTestPage

---

# MC production notes

---

## Getting started with evgen

See this TWiki page for a very general overview of running generators in ATLAS: <https://twiki.cern.ch/twiki/bin/view/AtlasProtected/RunningGenerators> And this one which is a bit more specific to MC15: <https://twiki.cern.ch/twiki/bin/view/AtlasProtected/AtlasMcProductionMC15>

The basic idea is that you have some python jobOptions that load and run your MC generator of choice. You run this in Athena by doing

```
setupATLAS
asetup <some release>,here

Generate_tf.py <arguments>
```

Where the generate transform (Generate\_tf.py) is essentially a bit of python that creates a simplified version of a more complicated Athena command/jobOptions that has nice command line arguments. should be whatever the most recent release that MC production is using, but this information can be hard to find and for now I would suggest you use 19.2.3.8.

Then the main thing you will use to generate events is MadGraphControl. This is a package for running MadGraph5\_aMC@NLO in Athena via some python steering. You can take a look at this TWiki page to get an idea of how this works and some simple examples: <https://twiki.cern.ch/twiki/bin/view/AtlasProtected/MadGraph5aMCatNLOForAtlas>

## Job submission script

Script is sendJobs.sh (please remove ".text" suffix).

Usage is essentially:

```
./sendJobs.sh <generation mode [-f:on-the-fly,-g: gridpack mode]> <run mode [-b:build tarball,-t:
```

- A couple of examples:
  - ◆ On-the-fly:

```
./sendJobs.sh -f -b -o -i "999999" -c 7000 -a 19.2.3.8,here -j MC15.999999.MadGraphPythia8EvtGen_
./sendJobs.sh -f -t -o -i "999999" -c 7000 -a 19.2.3.8,here -j MC15.999999.MadGraphPythia8EvtGen_
```

(the first line builds the panda tarball, the second submits the jobs - this is useful if you want to send e.g. several DSIDs because you don't have to remake the tarball each time)

- ◆ From gridpack:

```
./sendJobs.sh -g -b -o -i "999999" -c 7000 -a 19.2.3.8,here -j MC15.999999.MadGraphPythia8EvtGen_
./sendJobs.sh -g -t -o -i "999999" -c 7000 -a 19.2.3.8,here -j MC15.999999.MadGraphPythia8EvtGen_
```

(only send jobs to CERN and BNL) I suggest you start first with running some of the examples on the MG5\_aMC@NLO TWiki page above first as that's a bit simpler.



# MC production coordinator notes

---

## Meetings

- Physics Coord (Monday 15:00)
- ADC Weekly (Tuesday 14:45)
- Simulation Steering (Tuesday 16:00)
- MC Production (Wednesday 10:00)
- PMG subgroup+software+production (Monday, 13:00)
- ProdSys2 ("coffee" meetings) (Tues & Thurs 9:00) [indico](#)

## Useful webpages

- [AtlasProductionGroup](#)
- [AtlasProductionGroupInternal](#)
- For PC meeting summaries
- [ProdSys2RequestCreation](#)
- [MC production requests](#)
- [Things to fix in ProdSys2](#)

## Misc

### tarball dir

`/afs/cern.ch/atlas/software/kits/EvgenJobOpts/`

### Permissions

- "pts add mcfayden zp:mcgen" should work...
- "pts membership `whoami`" it will at least tell you what you have permissions for...
- "fs sa . mcfayden rlidwka" also?

# SUSY MC contact notes

---

## Workflow

- Start
  - ◆ Fill in BIG spreadsheet as far as possible
  - ◆ Assign DSIDs
  - ◆ Request spreadsheet and jobOptions tarball
  - ◆ Check jobOptions (use scripts)
- Follow JIRA workflow:
  - ◆ Start at "To do" --> ask for SUSY subgroup approval - "@A and @B", has this request been discussed, and are you happy with it?"
  - ◆ When approved click "Initial sign-off" --> "In progress"
  - ◆ "Create" issue with MC prod people (see below)
  - ◆ Add spreadsheet to SUSY MC spreadsheets page here
  - ◆ "In progress" --> Get jobOptions and spreadsheet etc. click "Inputs/jOs" ...
  - ◆ ... this takes us to --> "Prepare"
  - ◆ "Prepare" --> Ask for validation --> "Validation" (validation optional to an extent)
  - ◆ "Validation" --> if validation ok --> "Request ready"
  - ◆ "Request ready" --> ask SUSY group convenors for approval
    - ◇ Here add in comments "We now have MC12JobOptions-XX-YY-ZZ for this, so it's up to [~jboyd] and [~onofrio] to approve this request for N {type} events in M datasets at priority Y" \* Now click "Group sign-off" --> "In review" \* "In review" --> ask Physics Coordination for approval:
    - ◇ Use standard template (below) - information can be taken from spreadsheet.

Subject: SUSY <MCCampaign> <CoMEnergy> Request: <description>

Hi Andreas,

The SUSY group would like to request ...<brief description (JIRA title)>... samples with <Generat

The JIRA thread is here: ...

Technical details:

- The DSIDs are ...
- The spreadsheet for the request is at ...
- <type> priority <Y>
- <jobOptions tag/info>
- DESDs to be kept

- - ◆ Include in cc: SUSY convenors, subgroup convenors, atlas\_csc\_mcprodman
  - ◆ Once approved, click "PC approved" --> "Approved"
  - ◆ "Approved" - now marked as done in JIRA
- Request to MC prod people
  - ◆ In JIRA "create issue"
    - ◇ Project: default
    - ◇ Issue type: Bug
    - ◇ Summary: description and DSIDs
    - ◇ Priority: Ignore this unless it is very urgent
    - ◇ Components: MCJobOptions
    - ◇ Labels: SUSY, MC12JobOptions



- ◇ Attach tarball with jOs - needs format so that it could slot straight into MC12JobOptions
- ◇ Description: "The SUSY group would like to request the attached files for inclusion in the next tag of MC12JobOptions. They implement There are a few dummy job options files, one modified control file, a couple of slha files, .. Please let me know if you spot any problems."

- If there are input files
  - ◆ Edit evgeninputfiles.csv
  - ◆ Tag updated version
- Check if jobs are running - if not contact requester

## BIG Spreadsheet columns/colours

- DS start-DS end, N samples
- Color code: white - just came in, yellow - I need to do something, light green - jobs not submitted or still running, purple - problem with initial submission and not yet resolved, dark green - all done
- SUSY: SUSY convenors approved?
- ATLAS: ATLAS approved?
- Prio: Priority in JIRA - colour RED if prio=high (3=minor,5=major,...)
- PP: Production priority (from request spreadsheet)
- nEe: N events EvGen only
- nEG: N events FullSim only
- nEF: N events FastSim only
- SG: Which SUSY subgroup group

## Links

JIRA SUSY summary [↗](#) BIG spreadsheet [↗](#)  
MC spreadsheets - Google Drive [↗](#) Example request spreadsheet [↗](#)  
evgeninputfiles.csv [↗](#)  
SUSYMcRequestProcedure TWiki  
Release status [↗](#)  
AtlasMcProductionMC12 TWiki  
AtlasProductionGroupMC12a TWiki  
PreparingLesHouchesEven TWiki  
SUSYEvGenValidation TWiki  
MCProdJira TWiki  
To register inputs/integration grids/config files [↗](#)

## Misc

### Tags

- For an Atlfast-II job the black magic of Atlfast-II happens in the reco step in this case (sim just does the ID, the calorimeter simulation actually doesn't happen until that second step) so the "recon" job gets given a second "\_aXYZ" tag rather than a "\_rABC" tag. E.g. e2880\_a188\_a171
- For the spreadsheets the following convention should be followed - for the dummy tags  
e1111\_a2222\_a3333\_r4444:

- e1111 goes in the "Evgen" column  
- a2222 goes in the "Simul" column

- a3333 goes in the "Atlfast" column (even though it's a reco tag)
- r4444 goes in the "Atlf Merge" column (even though it's a reco merge tag)

## Bugfix:

The usual procedure is if it is a **bugfix** and the old samples are **buggy** (read: will not be used by analyses) you can go with the same DSID and rename if necessary (Josh, make sure during the request that you ask them to mark the old ones as obsolete and schedule them for deletion). If those old samples will be used by any analyses, then go with new DSIDs.

If it's a bug fix and it's small (20k-100k events) you don't strictly have to go through PC; you can directly iterate with the production team. If it's more than about 100k then it's good practice to make sure PC is aware.

## Replacing existing jOs:

Always do a diff of new and old to make sure changes make sense

## New.py:

Need to request a new tag. Try to check how much has changed between new version and previous version via diff and/or ChangeLog if it's not too much then you probably Zach to) and request a new pcache with new tag. Add this new pcache to the release column of request spreadsheet.

## Sherpa integration grids

These are always required even for processes that are "quick" to generate.

## UEEE3 in 13

We have a proper energy extrapolation tune (as opposed to a tune, with adjusted parameters for specific CM energies) for 13 TeV with UEEE4(5) so UEEE3 is not an option for 13 TeV jobs (it is for 7,8,14 TeV).

---

# Sherpa

---

asetup 17.2.12.7.10,MCProd,64,slc5,here... tar czf input.tar.gz Process/ Results.db

---

# LondonFramework

---

LondonFramework

---

# Direct stop 0-lepton 2012 8TeV data

---

## Technicalities

- Working from SUSYTools-00-01-03
- Using p1032(1033) tags of NTUP\_SUSY

## Code

Code can be found here:

[https://svnweb.cern.ch/trac/atlasusr/browser/mcfayden/SUSY/rel17\\_8TeV/stop/trunk/](https://svnweb.cern.ch/trac/atlasusr/browser/mcfayden/SUSY/rel17_8TeV/stop/trunk/)

- Object definitions applied in:
  - ◆ [analysis/shared/NTAnalysisBase/NTGet\\*.cxx.h](#) (Physics objects)
  - ◆ [analysis/shared/NTAnalysisBase/NTAnalysisBase.cxx.h](#) (Tool initialisations)
  - ◆ [analysis/shared/NTAnalysisBase/NTMasterFunctions\\_V1\\_rel17\\_SUSY.cxx.h](#) (Overlap removal and scale factors)
- Kinematic variables defined in:
  - ◆ [analysis/progs/rel17\\_8TeV\\_stop/NTKinematics.cxx](#)
  - ◆ [analysis/progs/rel17\\_8TeV\\_stop/FillHists.cxx](#)
  - ◆ [analysis/progs/rel17\\_8TeV\\_stop/rel17\\_8TeV\\_stop.cxx](#)

## Main changes wrt 2011 object definitions

Jets:

- Now using **AntiKt4LCTopo** jets rather than **AntiKt4TopoNewEM**
- No calibration required anymore

b-jets:

- No new recommendations afaik

Electrons:

- electron quality flags wrong in D3PDs (not sure which tags) -> define using functions, using:

```
#include "egammaAnalysisUtils/IsEMPlusPlusDefs.h"  
#include "egammaEvent/egammaPIDdefs.h"
```

Muons:

- Don't appear to have changed...

Triggers:

- jet+MET:
  - ◆ EF\_j80\_a4tchad\_xe100\_tclw\_loose
  - ◆ EF\_j170\_a4tchad\_EFxe80\_tclw

- single electron:
  - ◆ EF\_e24vh\_medium1\_EFxe35\_tclw
- single muon:
  - ◆ EF\_mu24\_j65\_a4tchad\_EFxe40\_tclw

Initialisation of most tools now slightly different. See SUSYObjDef.cxx [↗](#).

---

# trigEmulator

---

Emulation code for EF jet triggers in 2011 using D3PDs. This tool should be quite versatile. Initially it was developed for emulation of EF\_\*a4tc\* jet triggers from mc10a has only EF\_\*a4\* type branches in the menu. From the inputted EF trigger name string it works out the type of trigger and thresholds to be applied. The L2 decision is required as an input.

- Emulates different jet algorithms, a4, a4tc, a10tc... etc.
- Only requires L2 decision and trigger name as an input

## Basic Instructions

### Setting up

You need to compile the .cxx file with the rest of your code (exactly how you do this depends on how you run on the D3PDs...). For example if you used a MakeClass in root you would want to do something like:

```
gROOT->ProcessLine(".L trigEmulator.cxx+");
```

before you start calling the function "Loop()"

Then in your main analysis file include the header:

```
#include "trigEmulator.h"
```

and in your main analysis function but outside the event loop you should do something like this:

```
trigEmulator::trigEmulator myEFTrig;
```

### Feeding branches

Then inside the event loop you need to give the emulator the branches it needs something like this (again it depends on your setup how you call the branches etc.):

```
myEFTrig.jetpt=(*trig_EF_jet_emscale_pt);  
myEFTrig.jeteta=(*trig_EF_jet_emscale_eta);  
myEFTrig.jetphi=(*trig_EF_jet_emscale_phi);  
myEFTrig.type_a4tc=(*trig_EF_jet_a4tc);  
myEFTrig.type_a4=(*trig_EF_jet_a4);  
myEFTrig.type_a10tc=(*trig_EF_jet_a10tc);  
myEFTrig.ptThresh=30.;  
myEFTrig.etaRange=3.2;  
myEFTrig.EF_metx=trig_EF_met_MEx;  
myEFTrig.EF_mety=trig_EF_met_MEy;
```

### Usage

Then again inside the event loop, you get the emulated EF decision by passing the L2 decision and the string of the EF trigger you want like this:

```
bool testBit_EF_j75_a4tc_EFFS = myEFTrig.isPassed(L2_j70,"EF_j75_a4tc_EFFS");
```

## Files:

- trigEmulator.cxx: trigEmulator class C++ file
- trigEmulator.h: trigEmulator class header file
- steer.C: Steering file for MakeClass example
- testClass.C: MakeClass .C file
- testClass.h: MakeClass header file

### Major updates:

-- JoshMcFayden - 16-May-2011

%RESPONSIBLE% Main.JMcfayden

%REVIEW% **Never reviewed**

---

This topic: Sandbox > JoshMcFayden

Topic revision: r38 - 2018-11-06 - JoshMcFayden



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

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