SLAC Common Ntuple Package

Introduction

The SLAC ATLAS group has developed an inclusive software package for producing flat ROOT ntuples from ATLAS Pool files, either ESD or AOD.

Current Versions

Compatible with release 15.4.0 (RECOMMENDED):

JetTrackVertexAnalysis-00-03-11

Changed with respect to JetTrackVertexAnalysis-00-03-10

- Added the possibilit to apply cell energy density weighting to standard calo towers
- Removed deletion of pointers in extrapolateTrack()
- Updated JVF section of RecoJetBlock to use the functionality of the JetVertexAssociationTool and the JetVertexFraction in release 15.4.0
- Uncommented the line in the TRackBlock which prevents the usage of the track impact parameters
- Added TrigVertex branches to the TriggerBlock for beamspot analysis
- Protection against missing measuredPerigee() in TrackJetBlock

Compatible with release 15.3.0:

JetTrackVertexAnalysis-00-03-10

Changes from JetTrackVertexAnalysis-00-03-08:

- Some includes changed in order to compile in 00-03-10
- b-tagging code in RecoJetBlock.cxx cleaned up and updated for 15.3.0 b-tagging changes

Known bugs:

- Setting DoExtrapolation = True in the TrackBlock appears to cause a segmentation fault. Under investigation

Install the following packages on top of 15.3.0 in order for the ntuple maker to work. Check out all packages before building any of them.

- To avoid a seg fault in the magnetic field tool
  - InDetTrackingGeometry-01-04-03 in InnerDetector/InDetDetDescr/InDetTrackingGeometry
  - TrkDetDescrSvc-00-14-07 in Tracking/TrkDetDescr/TrkDetDescrSvc

- To avoid zeros in all TrackParticle parameters
  - TrkEventTPCnv-00-21-00-01 in Tracking/TrkEventCnv/TrkEventTPCnv
  - TrkEventAthenaPool-01-30-05 in Tracking/TrkEventCnv/TrkEventAthenaPool
  - TrackParticleTPCnv-00-02-01 in Reconstruction/TrackParticleTPCnv
  - TrackParticleAthenaPool-00-04-04 in Reconstruction/TrackParticleAthenaPool

Compatible with release 15.1.0:

JetTrackVertexAnalysis-00-03-08

After 15.0.0 there were changes in the L1 Calo jetEtSum computation and related tools that are not compatible with with releases <= 15.0.0

- includes the L1 jet ET sum computation in the trigger block, there are two possibilites:

1. extract the weights provided in the L1 xml configuration file (L1ET_JetEtSum)
2. passing the corresponding weights directly through job options (L1ET_myJetEtSum)

Also includes the a block for track jets and job options to construct them.

00-03-08 includes commented-out JVF changes which will work in the next release (JVF branches do NOT work in 00-03-08), various bugfixes and naming convention updates, and updates to jet moment handling. 00-03-06 will still work if 00-03-08 gives problems for any reason. 00-03-07 does not compile in 15.1.0.

Compatible with release <= 15.0.0:

We'll try to keep notes on the current recommended version of the code here, close to the top, for reference.

JetTrackVertexAnalysis-00-02-12

- Updated CaloCellESDBlock to include H1 information
- Ignacio fixed EMTrackMatch for EoverPcalculation

Structure and usage

General structure

The package is divided into several independent algorithms, each of which is responsible for adding groups (or *blocks*) of data to the output ROOT TTree:

1. Calo-Tower Block
2. Electron Block
3. Photon Block
4. MET Block
5. Muon Block
6. Reco-Jet Block
7. Topo-Cluster Block
8. Track Block
9. TrackJetBlock (>=15.1.0)
10. Trigger Block
11. Truth-Jet Block
12. Truth-MET Block
13. Truth Particle Block
14. Truth Vertex Block
15. Vertex Block

Package location

The package resides in the SLAC ATLAS CVS repository here:

- SLAC CVS Location

Note that the CMTCVSOFFSET is thus different and should be explicitly defined when checking in or out of this CVS location

```
cmt co -r JetTrackVertexAnalysis-00-02-12 -o groups/slac JetTrackVertexAnalysis
```

Checkout and build from a lxplus CERN computer account

This would apply to anyone who wants to start using the package from scratch on a lxplus account at CERN.

Setup the analysis environment

Log in to lxplus.

```
phansson@phansson-laptop~/% ssh <nop>phansson@lxplus.cern.ch
```

Create the working directories.

```
phansson@lxplus253~/% mkdir work
phansson@lxplus253~/% mkdir work/jetmetbtag
phansson@lxplus253~/% mkdir work/jetmetbtag/mytest
phansson@lxplus253~/% cd work/jetmetbtag/mytest
phansson@lxplus253~/work/jetmetbtag/mytest% mkdir 14.2.25
```

Setup the CMT environment

Source the CMT setup script.

```
 [phansson@lxplus253]~/work/jetmetbtag/mytest% source /afs/cern.ch/sw/contrib/CMT/v1r20p20080222/mgr/setup.sh
```

Create an empty home *requirements* file.

```
[phansson@lxplus253]~/work/jetmetbtag/mytest% touch requirementsMore information on what the requirements file
is doing can be found in AtlasLogin
```

Below is a an example *requirements* file:

```
#-------------------------------------------------------------------
#CMT home requirements file
set   CMTSITE  CERN
set   SITEROOT /afs/cern.ch
macro ATLAS_DIST_AREA /afs/cern.ch/atlas/software/dist
macro ATLAS_TEST_AREA /afs/cern.ch/user/p/phansson/work/jetmetbtag/mytest
apply_tag setup
apply_tag simpleTest
use AtlasLogin AtlasLogin-* $(ATLAS_DIST_AREA)
set CMTCONFIG i686-slc4-gcc34-opt
#-------------------------------------------------------------------
```

Create analysis environment including the CMT setup scripts.

```
[phansson@lxplus253]~/work/jetmetbtag/mytest% cmt config
----------------------------------------
Configuring environment for standalone package.
CMT version v1r20p20080222.
System is amd64_linux26
----------------------------------------
Creating setup scripts.
Creating cleanup scripts.
```

Setup your release (ask your closest expert which one to use).

```
[phansson@lxplus253]~/work/jetmetbtag/mytest% source setup.sh -tag=14.2.25
#CMT> Warning: template <src_dir> not expected in pattern install_scripts (from TDAQCPolicy)
#CMT> Warning: template <files> not expected in pattern install_scripts (from TDAQCPolicy)
```

Detailed info on the the account setup steps can be found in the WorkBookSetAccount.

Check out and compile nTupleMaker from CVS using CMT

In this example the tag used is *JetTrackVertexAnalysis-00-02-04*.

Check out the package.

```
[phansson@lxplus253]~/work/jetmetbtag/mytest% cd 14.2.25

[phansson@lxplus253]~/work/jetmetbtag/mytest/14.2.25% cmt co -r JetTrackVertexAnalysis-00-02-12 -o groups/slac
JetTrackVertexAnalysis
# ================= working on package JetTrackVertexAnalysis version JetTrackVertexAnalysis-00-02-12 in /afs
/cern.ch/user/p/
phansson/work/jetmetbtag/mytest/14.2.25/JetTrackVertexAnalysis
  # get top files
cvs update: Updating .
Creating setup scripts.
Creating cleanup scripts.
Installing the run directory
```

Setup the package in the analysis environment.

```
[phansson@lxplus253]~/work/jetmetbtag/mytest/14.2.25% source JetTrackVertexAnalysis/cmt/setup.sh
```

Compile the package.

```
 [phansson@lxplus253]~/work/jetmetbtag/mytest/14.2.25% cd JetTrackVertexAnalysis/cmt

[phansson@lxplus253]~/work/jetmetbtag/mytest/14.2.25/JetTrackVertexAnalysis/cmt% make
...
...
...
#CMT---> all ok.
```

Setup the package from a lxplus CERN computer account

After doing the setup from scratch described above it is easy to setup the environment for consecutive logins as the CMT setup scripts are already generated.

Log in to lxplus.

```
 [phansson@phansson-laptop]~/% ssh <nop>phansson@lxplus.cern.ch
```

Go directly to the working directory and setup the release.

```
[phansson@lxplus253]~/work/jetmetbtag/mytest% source setup.sh -tag=14.2.25
#CMT> Warning: template <src_dir> not expected in pattern install_scripts (from TDAQCPolicy)
#CMT> Warning: template <files> not expected in pattern install_scripts (from TDAQCPolicy)
```

Ignore the warnings.

Setup the JetTrackVertexAnalysis? package.

```
[phansson@lxplus253]~/work/jetmetbtag/mytest% cd 14.2.25
[phansson@lxplus253]~/work/jetmetbtag/mytest/14.2.25% source JetTrackVertexAnalysis/cmt/setup.sh
#CMT> Warning: template <src_dir> not expected in pattern install_scripts (from TDAQCPolicy)
#CMT> Warning: template <files> not expected in pattern install_scripts (from TDAQCPolicy)
```

Ignore the warnings.

Done.

Run the common nTupleMaker on an AOD residing on a ***local*** disk

FInd an AOD file that can be used. If you have access to pcphuat disks a file that should work with this example can be found here

> */u1/phansson/data/WbbNp1.250evt.0skip.no_trig.AOD.pool.root*

Copy this file, or run it from that directory.

If you copy the file to any other directory, remember to change the file location as necessary in the steps below.

Go to run directory:

```
[phansson@lxplus253]~/work/jetmetbtag/mytest/14.2.25/JetTrackVertexAnalysis/cmt% cd ../run
```

The configuration of the AthenaFramework job is done using a python configuration file, the so-called *jobOption* file located in the */share* directory of the package. In this example the default job option file is used: **CommonNtuple_defaultOptions.py**.

In order to run the program, edit this file with your favorite text editor.

```
[phansson@lxplus253]~/work/jetmetbtag/mytest/14.2.25/JetTrackVertexAnalysis/run% cp ../share
/CommonNtuple_defaultOptions.py .
[phansson@lxplus253]~/work/jetmetbtag/mytest/14.2.25/JetTrackVertexAnalysis/run% emacs
CommonNtuple_defaultOptions.py
```

Required editing:

1. Change the input file:

```
svcMgr.EventSelector.InputCollections = [ "/home/fizisist/work/data/WbbNp0_AOD_2K.pool.root" ]
```

   to (or the location of the AOD file that you are using)

```
svcMgr.EventSelector.InputCollections = [ "/u1/phansson/data/WbbNp1.250evt.0skip.no_trig.AOD.pool.root" ]
```

2. Change output directory of the resulting ROOT file that contains the common TTree

```
OutputNtupleDir  = "/home/fizisist/work/data/""
```

   to (or wherever you want the ROOT file to end up)

```
OutputNtupleDir  = "/afs/cern.ch/user/p/phansson/scratch0/"
```

3. ***(optional)***Change name of the output ROOT file

```
OutputNtupleName = "test_2K_fromAOD.root"
```

   to

```
OutputNtupleName = "your_file_name.root"
```

Run over the local file.

```
[phansson@lxplus253]\~/work/jetmetbtag/mytest/14.2.25/JetTrackVertexAnalysis/run% athena
CommonNtuple_defaultOptions.py
```

This should produce a root file named whatever you put in *OutputNtupleName* in the directory specified in *OutputNtupleDir*.

Note that the events may be large and thus choose your output directory accordingly. The number of events can be changed in *CommonNtuple_defaultOptions.py* by modifying the line

```
theApp.EvtMax = 10
```

Run the common nTupleMaker on the GRID using PANDA using a CERN lxplus account

This explains how to run the JetTrackVertexAnalysis? nTupleMaker on the GRID using PANDA.

Initial fact-finding mission

It's probably good practice to know something about your dataset before you start an analysis. Go to AMI

- AMI

and type in the name of the dataset, or the configuration tag used to produce it (e.g. *e352_s462_r541*).

Some of the information you'll want to know, which I am taking from *mc08.105404.SU6_jimmy_susy.recon.AOD.e352_s462_r541* as an example is:

- Geometry version used: e.g *ATLAS-GEO-02-01-00*
- Events per file: *Children Elements (right side) > event_range > details > nMaxEventPerFile = 250*
- Release used to produce the dataset: *Transformation Package > 14.2.20.3*
- Job configuration at the simulation stage (e.g. if an offset beamspot is used): *Production step > simul > JobConfig =* *VertexPos.py*

Setup PANDA

Log in to lxplus.

Follow the instructions here for setting up PANDA:

- PANDA Installation Instructions

Go directly to the working directory and setup the release:

```
[phansson@lxplus253]~/work/jetmetbtag/mytest% source setup.sh -tag=14.2.25
[phansson@lxplus253]~/work/jetmetbtag/mytest% cd 14.2.25
```

Check-out the HEAD of PandaTools?.

```
[phansson@lxplus253]~/work/jetmetbtag/mytest% cmt co PhysicsAnalysis/DistributedAnalysis/PandaTools
```

Setup the package in the environment.

```
[phansson@lxplus253]~/work/jetmetbtag/mytest% source PhysicsAnalysis/DistributedAnalysis/PandaTools/cmt/setup.sh
```

Go to the cmt directory of the package and compile.

```
[phansson@lxplus253]~/work/jetmetbtag/mytest% cd PhysicsAnalysis/DistributedAnalysis/PandaTools/cmt/
[phansson@lxplus216]~/work/jetmetbtag/mytest/14.2.25/PhysicsAnalysis/DistributedAnalysis/PandaTools/cmt% make
...
...
...
#CMT---> all ok.
```

Done. Your environment should now be able to submit jobs using panda.

Consecutive setup to enable PANDA

Log in to lxplus.

Go directly to the working directory and setup the release:

```
[phansson@lxplus253]~/work/jetmetbtag/mytest% source setup.sh -tag=14.2.25
```

Set up the package in the environment

```
[phansson@lxplus253]~/work/jetmetbtag/mytest% source PhysicsAnalysis/DistributedAnalysis/PandaTools/cmt/setup.sh
```

Done. Your environment should now be able to submit jobs using panda.

Submit jobs with PANDA

Log in to lxplus.

Set up your grid environment *(note that there is no obligation to be in the working directory when setting up the grid environment).*

```
[phansson@lxplus209]~/% source /afs/cern.ch/project/gd/LCG-share/sl4/etc/profile.d/grid_env.sh
[phansson@lxplus209]~/% voms-proxy-init -voms atlas
Cannot find file or dir: /afs/cern.ch/user/p/phansson/.glite/vomses
Enter GRID pass phrase:
Your identity: /O=Grid/O=NorduGrid/OU=kth.se/CN=Per Hansson
Creating temporary proxy ................................... Done
Contacting  voms.cern.ch:15001 [/DC=ch/DC=cern/OU=computers/CN=voms.cern.ch] "atlas" Done
Creating proxy ............................. Done
Your proxy is valid until Tue Jan 13 23:28:24 2009
```

DQ2 is a good tool to find and browse datasets. To set this tool up use the following command.

```
[phansson@lxplus209]~/% source /afs/cern.ch/atlas/offline/external/GRID/ddm/DQ2Clients/setup.sh
```

Information about DQ2 clients can be found in the DQ2ClientsHowTo herehttps://twiki.cern.ch/twiki/bin/view/Atlas/DQ2ClientsHowTo

Tips/recommendation: Put the last three commands in a shell script e.g. "setup_grid_tools.sh" that can be run at setup.

Find a dataset that you want to run over. Note that this dataset *have to be registered to the grid*. In this example I use a Wbb sample.

```
[phansson@lxplus209]~/% dq2-ls 'user*David*Miller*WbbNp1*AOD*'
user.DavidWilkinsMiller.misal1_mc12.006281.AlpgenJimmyWbbNp1.v12000605.no_trigger.AOD
```

Go to the working directory and setup the release

```
[phansson@lxplus253]~/% cd work/jetmetbtag/mytest
[phansson@lxplus253]~/work/jetmetbtag/mytest% source setup.sh -tag=14.2.25
```

Set up the JetTrackVertexAnalysis package in your environment.

```
[phansson@pcphuat27]~/work/jetmetbtag/mytest% cd 14.2.25
[phansson@pcphuat27]~/work/jetmetbtag/mytest/14.2.25% source JetTrackVertexAnalysis/cmt/setup.sh
#CMT> Warning: template <src_dir> not expected in pattern install_scripts (from TDAQCPolicy)
#CMT> Warning: template <files> not expected in pattern install_scripts (from TDAQCPolicy)
```

Setup Panda in your environmen.

```
[phansson@pcphuat27]~/work/jetmetbtag/mytest/14.2.25% source PhysicsAnalysis/DistributedAnalysis/PandaTools/cmt
/setup.sh
#CMT> Warning: template <src_dir> not expected in pattern install_scripts (from TDAQCPolicy)
#CMT> Warning: template <files> not expected in pattern install_scripts (from TDAQCPolicy)
```

Ignore these warnings.

The panda job is sent by executing the *pathena* script which should now be found in the */InstallArea* (at least the a link to it).

```
[phansson@pcphuat27]~/work/jetmetbtag/mytest/14.2.25% more InstallArea/share/bin/pathena
```

To see what parameters that can be used by the script.

```
[phansson@lxplus209]~/work/jetmetbtag/mytest/14.2.25% pathena --help
```

In order to help sending (multiple) jobs over (possibly) multiple datasets a simple python script is used. This file is located in the */share* directory of the *JetTrackVertexAnalysis* package. Copy it to the */run* directory and open with your text editor.

```
[phansson@lxplus209]~/work/jetmetbtag/mytest/14.2.25% cp JetTrackVertexAnalysis/share/submitDefaultPathenaJob.
py JetTrackVertexAnalysis/run/
[phansson@lxplus209]~/work/jetmetbtag/mytest/14.2.25% cd JetTrackVertexAnalysis/run
[phansson@lxplus209]~/work/jetmetbtag/mytest/14.2.25% emacs submitDefaultPathenaJob.py
```

The main characteristics of this file is described below.

```
'user': the name of the user output dataset
'identifier': name that can be used to specify additional identifier in the output dataset
'inDS': list of datasets to run over
'options': parameters given to the pathena script (see pathena --help)
'job_options': the job_optionsfile to be used.
```

The script essentially loops over the datasets and submits the jobs with the given options. **Note**that the first submission will create a "job library" containing the compiled code and your environment. This same library is registered automatically on the grid and can then be used in subsequent jobs by specifying

```
--libDs LAST
```

It is also possible to specify the exact name of a library as long as it is registered on the grid.

A good cross-check before executing the pathena script is to run the common nTupleMaker on one of the files in (at least) one of the datasets locally and make sure everything behaves as expected. It is also possible to run over only a few files with panda as a test before sending large jobs, this is possible by specifying in the option

```
--split 1
```

to submit only 1 job. After editing the submission script with the dataset of your choice and (possibly) changing options, execute the script.

```
[phansson@lxplus209]~/work/jetmetbtag/mytest/14.2.25/JetTrackVertexAnalysis/run% python ../share
/submitDefaultPathenaJob.py
pathena --nFilesPerJob 2 --individualOutDS --split 20 --outDS user09.PerHansson.default.user.DavidWilkinsMiller.
misal1_mc12.0
06281.AlpgenJimmyWbbNp1.v12000605.no_trigger.AOD --inDS user.DavidWilkinsMiller.misal1_mc12.006281.
AlpgenJimmyWbbNp1.v1200060
5.no_trigger.AOD CommonNtuple_defaultOptions.py
extracting run configuration
ConfigExtractor > Input=POOL
ConfigExtractor > Output=AANT AANTupleStream AANT
archive sources
archive InstallArea
check symbolic links
post sources/jobO
query files in dataset:user.DavidWilkinsMiller.misal1_mc12.006280.AlpgenJimmyWbbNp1.v12000605.no_trigger.AOD
submit
====================
 JobID  : 707
 Status : 0
  > build
    PandaID=23151069
  > run
    PandaID=23151070-23151086
```

Tip: If the dataset files are not found, one solution could be to locate the files with *DQ2* and specify this site specifically in the *pathena* script.

```
[phansson@lxplus209]~/work/jetmetbtag/mytest/14.2.25/JetTrackVertexAnalysis/run% dq2-ls -r user.
DavidWilkinsMiller.misal1_mc1
2.006281.AlpgenJimmyWbbNp1.v12000605.no_trigger.AOD
user.DavidWilkinsMiller.misal1_mc12.006281.AlpgenJimmyWbbNp1.v12000605.no_trigger.AOD
 INCOMPLETE:
 COMPLETE:
   SLACXRD
```

In this example the files are at *SLACXRD* and thus a site option can be added to the pathena script options.

```
--site=SLACXRD
```

Details on analysis using Panda can be found at DAonPanda.

Status and re-submission of jobs sent with PANDA

Status of jobs

There are different ways of checking the status of submitted jobs.

1. *pathena_util*: a command line interface to the PANDA DB which is available after setting up PANDA in the release environment.
2. PandaMonitor:web interface available atPandaMonitor.

A notification email is automatically sent to the user indicating the result of the job after it has finished.

Re-submit failed jobs

Jobs can be re-submitted using *pathena_util*.

Retrieve jobs sent with panda

Jobs successfully finished will be registrered on the grid using the dataset given in the *pathena* script options. Datasets can be fetched using DQ2.