

# Table of Contents

<b>TAGSelectionToESD.....</b>	<b>1</b>
<b>Intro to TAG files.....</b>	<b>2</b>
<b>Get TAG_COMM (or TAG) data files.....</b>	<b>3</b>
<b>Local TAG_COMM (or TAG) analysis.....</b>	<b>5</b>
Select events from TAG_COMM (or TAG).....	5
Skimming ESDs with selected events from TAG (or TAG_COMM).....	5
Runs looked at using TAGanalysis.C.....	6
<b>Using TAG_COMM to select events directly from ESD.....</b>	<b>7</b>
Addendum.....	7
<b>Using TAG_COMM to select events from ByteStream.....</b>	<b>9</b>
<b>To Do.....</b>	<b>10</b>
<b>For Preema:.....</b>	<b>11</b>
Steps Overview:.....	11
1. Setting up on hal.....	11
2. Get TAG and TAG_COMM files.....	12
3. Create the TAG_MCP.....	12
4. Create events list for "tight" and "loose" selection.....	13
5. Create skimmed ESDs.....	13

# TAGSelectionToESD



This TWiki is still in the "guinea pig"  
stage...email me if you run into problems!  
([emily.thompson@cernNOSPAMPLEASE.ch](mailto:emily.thompson@cern.ch))

# Intro to TAG files

See talks:

<http://indico.cern.ch/conferenceDisplay.py?confId=73119> (Validation of TAG\_COMM and TAG output for MCP event selection)

and

<http://indico.cern.ch/conferenceDisplay.py?confId=75756> (First Data Scanning)

Two main types: TAG and TAG\_COMM

TAG:

- Built with `PhysicsAnalysis/MuonID/MuonTagTools/src/CombinedMuonTagTool.cxx`
- Looks at merged containers in the AOD
- CombMuonWord contains information on type of muon stored in bitword (SA, extrapolated, combined, tagged, calo). No algorithm information

TAG\_COMM:

- Built with `Commission/CommissionRec/src/ComTagWriter.cxx`
- Muons: Has information on SA only (MooreTracks, ConvertedMBoyTracks), Number of SA tracks, LeadingMuon1, 2 for the 5 track parameters

# Get TAG\_COMM (or TAG) data files

Steps:

- On lxplus (or afs), set up release:

```
source setup.sh -tag=15.5.3.11,AtlasTier0,slc4,gcc34,32,opt,runtime
```

- In another terminal, set up Grid, dq2 and create proxy:

```
source /afs/cern.ch/project/gd/LCG-share/current/etc/profile.d/grid_env.sh
source /afs/cern.ch/atlas/offline/external/GRID/ddm/DQ2Clients/setup.sh
voms-proxy-init -voms atlas -valid 96:00
```

- Look for the TAG or TAG\_COMM files on AMI, or directly search on castor. Location on castor for 900 GeV runs:

```
nsls /castor/cern.ch/grid/atlas/tzero/prod1/perm/data09_900GeV/<stream>/<run number>
```

- For example, take stream MuonswBeam and run 141270

```
> nsls /castor/cern.ch/grid/atlas/tzero/prod1/perm/data09_900GeV/physics_MuonswBeam
data09_900GeV.00141270.physics_MuonswBeam.merge.AOD.f180_m288
data09_900GeV.00141270.physics_MuonswBeam.merge.AOD.f181_m293
data09_900GeV.00141270.physics_MuonswBeam.merge.DESD_COLLICAND.f181_m294
data09_900GeV.00141270.physics_MuonswBeam.merge.DESD_MUONCOMM.f180_m289
data09_900GeV.00141270.physics_MuonswBeam.merge.DESD_MUONCOMM.f181_m294
data09_900GeV.00141270.physics_MuonswBeam.merge.HIST.f180_m285
data09_900GeV.00141270.physics_MuonswBeam.merge.HIST.f181_m290
data09_900GeV.00141270.physics_MuonswBeam.merge.NTUP_MUONCALIB.f180_m286
data09_900GeV.00141270.physics_MuonswBeam.merge.NTUP_MUONCALIB.f181_m291
data09_900GeV.00141270.physics_MuonswBeam.merge.NTUP_TRIG.f180_m286
data09_900GeV.00141270.physics_MuonswBeam.merge.NTUP_TRIG.f181_m291
data09_900GeV.00141270.physics_MuonswBeam.merge.RAW
data09_900GeV.00141270.physics_MuonswBeam.merge.RAW.f180.recon.task.log
data09_900GeV.00141270.physics_MuonswBeam.merge.RAW.f181.recon.task.log
data09_900GeV.00141270.physics_MuonswBeam.merge.TAG.f180_m288
data09_900GeV.00141270.physics_MuonswBeam.merge.TAG.f181_m293
data09_900GeV.00141270.physics_MuonswBeam.merge.TAG_COMM.f180_m287
data09_900GeV.00141270.physics_MuonswBeam.merge.TAG_COMM.f181_m292
data09_900GeV.00141270.physics_MuonswBeam.recon.ESD.f180
data09_900GeV.00141270.physics_MuonswBeam.recon.ESD.f181
```

- look out for TAG (or TAG\_COMM) files and ESDs
  - ♦ reco tags: f180 = AtlasTier0 15.5.3.8, f181 = AtlasTier0 15.5.3.9, etc....
- It turns out to be good practice to choose your tag matching to your release, so in this case you might want to look at

```
data09_900GeV.00141270.physics_MuonswBeam.merge.TAG_COMM.f180_m287
data09_900GeV.00141270.physics_MuonswBeam.recon.ESD.f180
```

- Copy them from castor or get them with

```
dq2-get data09_900GeV.00140541.physics_MinBias.merge.TAG.f175_m273
```

- Once you have the TAG files, you must rename them all with .root at the end so that the macro can read them. Here is a simple script you can use to do that:

```
#!/bin/zsh
```

```
if [ $# -ne 1 ]
then
  echo "Usage: ./give_root_name.sh [directory]."
```

echo "Renames files in [directory] to have .root at the end."

```
  exit
fi

directory=$1

for file in `ls $directory`;do
  newname=`echo $file | sed s/$file/$file.root/`
  mv $directory/$file $directory/$newname
done
```

---

# Local TAG\_COMM (or TAG) analysis

For the instructions below, I've been looking at the MinBias stream in run 140541 (the run with our now-famous first collision!)

- Get and untar attachments (at the bottom of this TWiki): TAGanalysis.tgz or TAG\_COMManalysis.tgz
- Contains everything you need to do TAG or TAG\_COMM analysis
- Edit runTAG.C (or runTAG\_COMM.C) for input TAG (or TAG\_COMM) files
- TAGanalysis.C (or TAG\_COMManalysis.C) is just a root macro built from MakeSelector
  - ◆ Contains the main selection code
  - ◆ As of now, there isn't much selection at all, but you can add it to the code as you like. TAG variables are in TAGanalysis.h (or TAG\_COMManalysis.h)
- CombinedMuonWord is a class needed for TAGanalysis.C to decode the CombMuonWord in the TAG files.

## Select events from TAG\_COMM (or TAG)

To run:

```
> root -l
[0] .x runTAG.C
```

- Output: a file named EventsTAG.txt (or EventsTAG\_COMM.txt) and events.txt (this "events.txt" may be named something else from the code in TAGanalysis.C or TAG\_COMManalysis.C)
- EventsTAG.txt (or EventsTAG\_COMM.txt) has LumiBlock information, which is useful when copying ESDs (so that we don't have to copy all of them, just the ones with lumiblocks containing the events we find interesting).
  - ◆ For example, if one found interesting events in lumi blocks 154 and 296, one could dq2-get -f [file name with lb0154],[file name with lb0296] data09\_900GeV.00140541.physics\_MinBias.recon.ESD.f175
  - ◆ Again, if you want to know the names of the data sets, I would just look on castor using rfdir or nsls (using the full directory structure starting with /castor).
- events.txt is just a text file with event numbers only, used for the filter in the next section.

## Skimming ESDs with selected events from TAG (or TAG\_COMM)

- dq2-get or rfcp the ESDs of interest
- Get and untar attachment: ESDanalysis.tgz
  - ◆ ESDanalysis\_combined.py is the top options, here you can turn VP1 on or off, etc...
  - ◆ readesd\_combined.py is the event filter, and is read by ESDanalysis\_combined.py
  - ◆ Edit run140541\_MinBias.py to add the ESDs you've copied locally. This file is read in by readesd\_combined.py.
  - ◆ Copy events.txt to this directory, which is a list of events from your selection from the previous section. This is also read in by readesd\_combined.py

To run:

```
> source setup.sh -tag=15.5.3.8,AtlasTier0,slc4,gcc34,32,opt,runtime
> athena ESDanalysis_combined.py
```

At this point, a "skimmed" ESD file will be created, and if VP1 is switched on, then only events in `events.txt` will be displayed.

## Runs looked at using TAGanalysis.C

Here, using CombMuonWord, we count the total events in the run have:

MU: at least one muon seen by any algorithm

MU2: more than one muon seen by any algorithm

SA: at least one isStandAloneMuon()

SA2: more than one isStandAloneMuon()

EXTR: at least one hasMuonExtrapolatedTrackParticle() || hasInnerExtrapolatedTrackParticle()

EXTR2: more than one hasMuonExtrapolatedTrackParticle() || hasInnerExtrapolatedTrackParticle()

COMB: at least one isCombinedMuon()

COMB2: more than one isCombinedMuon()

TAG: at least one isLowPtReconstructedMuon()

TAG2: more than one isLowPtReconstructedMuon()

CALO: at least one isCaloMuonId()

CALO2: at least one isCaloMuonId()

Run	Stream	Events in Stream	MU	MU2	SA	SA2	EXTR	EXTR2	COMB	COMB2	TAG	TAG2	CALO	CALO2
141749	MinBias	19317	368	91	351	91	352	91	1	0	3	0	14	0
141707	MinBias	5433	104	36	104	36	104	36	0	0	0	0	0	0
141534	MinBias	11421	161	58	155	57	155	57	0	0	7	0	0	0
141226	MinBias	70212	739	268	473	188	473	188	0	0	314	68	6	0
140571	MinBias	24852	376	115	306	96	306	96	0	0	79	6	8	0
140541	MinBias	14290	303	58	193	51	193	51	0	0	48	0	69	3

# Using TAG\_COMM to select events directly from ESD

The following is a working recipe, validated with 15.5.3.8 AtlasTier0 release.

- look for available files on Castor:

```
nsls /castor/cern.ch/grid/atlas/tzero/prod1/perm/data09_900GeV/<stream>/<run_number>/
```

- create PoolFileCatalog for files on castor; replacing string "srm://srm-atlas.cern.ch" with "rfio:"

```
dq2-ls -L CERN-PROD_TZERO -P -R "srm://srm-atlas.cern.ch^rfio:" data09_900GeV.00141270
```

- get python

```
get_files -jo readesdusingtag.py
```

- edit/add some lines to your likings, for example:

- ◆ athenaCommonFlags.FilesInput=["rfio:/castor/cern.ch/grid/atlas/tzero/prod1/perm/data09\_900GeV/ph
- ◆ rec.doVP1.set\_Value\_and\_Lock(True)
- ◆ rec.doWriteESD.set\_Value\_and\_Lock(True)
- ◆ athenaCommonFlags.PoolInputQuery.set\_Value\_and\_Lock("(MooreSegments>0||ConvertedMBoySeg

- Before running, you must make sure STAGE\_SVCCLASS is set correctly. For the mentioned files do (in zsh):

```
export STAGE_SVCCLASS=atlcal
```

- Run

```
athena readesdusingtag.py >! logfile.log &
```

- I have some issues with the following error popping up at first try

```
Py:inputFilePeeker INFO Executing inputFilePeeker.py
Py:AthFile INFO opening [rfio:/castor/cern.ch/grid/atlas/tzero/prod1/perm/da
Py:inputFilePeeker WARNING Unable to open file rfio:/castor/cern.ch/grid/atlas/tzero/p
Py:inputFilePeeker ERROR Unable to build inputFileSummary from any of the specified
Py:AutoConfiguration ERROR No RunNumber stored in InputFile!
Py:AutoConfiguration ERROR No LumiBlock number stored in InputFile! Use 0
Shortened traceback (most recent user call last):
...
```

I don't know why, but this is 'solved' by setting up your release again... Happy running!

## Addendum

In the 'normal' readesdusingtag.py file one runs reconstruction by default. It could be that for your purposes this is too slow. Therefore we have created a version which does not do reconstruction and therefore 'skims' the original ESDs, and in addition creates a CalibrationNtuple on the fly. For this python see:

```
cp /afs/cern.ch/user/e/egge/public/readesdusingtagCalib.py .
cp /afs/cern.ch/user/e/egge/public/MuonCalibConfig.py .
```

Enjoy!





# Using TAG\_COMM to select events from ByteStream

Based on the experience for ESD described above, a similar approach is used for BS selection. Following the example stream and run from above we can use the RAW files from castor as well.

- set up release, grid, dq2, STAGE\_SVCCLASS as above
- create new dir

```
mkdir BS
cd BS
```

- choose TAG\_COMM and RAW files to use, for this example with same stream and run as above:

```
data09_900GeV.00141270.physics_MuonswBeam.merge.TAG_COMM.f180_m287
and
data09_900GeV.00141270.physics_MuonswBeam.merge.RAW
```

- create PoolFileCatalog for files on castor; replacing string "srm://srm-atlas.cern.ch" with "rfio:"

```
dq2-ls -L CERN-PROD_TZERO -P -R "srm://srm-atlas.cern.ch^rfio:" data09_900GeV.00141270
```

- create a file readbsusingtag.py with content (modify to your likings...):

```
from AthenaCommon.AthenaCommonFlags import athenaCommonFlags
athenaCommonFlags.FilesInput=["rfio:/castor/cern.ch/grid/atlas/tzero/prod1/perm/data09_900GeV.00141270.physics_MuonswBeam.merge.TAG_COMM.f180_m287"]
athenaCommonFlags.PoolInputQuery.set_Value_and_Lock(" (PixelTracks>0 || SCTTracks>0 || TRTTTracks>0 )")
athenaCommonFlags.EvtMax.set_Value_and_Lock(10)

from RecExConfig.RecFlags import rec
rec.readRDO.set_Value_and_Lock(True)
rec.doAOD.set_Value_and_Lock(False)
rec.doJiveXML.set_Value_and_Lock(True)
rec.doVP1.set_Value_and_Lock(False)
rec.doWriteAOD.set_Value_and_Lock(False)
rec.doESD.set_Value_and_Lock(True)
rec.doWriteESD.set_Value_and_Lock(True)
rec.doWriteTAG.set_Value_and_Lock(False)
include ("RecExCommon/RecExCommon_topOptions.py")

from AthenaCommon.ConfigurationShelve import saveToAscii
saveToAscii("config.txt")
```

- Run:

```
athena readbsusingtag.py >! logfileBS.log &
```

---

# To Do

- Need to set this up so that we can run the job options on the grid without having to copy ESDs locally
- Can we write a "skimmed" ESD directly from TAG selection without doing the two-step process?  
Then the whole thing could be on the grid... Should be possible with information on <https://twiki.cern.ch/twiki/bin/view/Atlas/CommissioningTag>.

# For Preema:

We'll do the skims in a series of steps (of course, a lot of this will be automated in the future, but in the mean time, what better automation is there than a hoard of grad students and post docs working around the clock in different time zones?)

## Steps Overview:

1. Set up your area on hal
2. Check out TAG and TAG\_COMM files for the run of interest.
3. Create the TAG\_MCP using the TAGMerger package
4. Create events list from TAG\_MCP for both "tight" and "loose" selection using TAG\_MCPanalysis.C
5. Create skimmed ESDs (dESD\_MCP) by running ESDanalysis job options with athena

## 1. Setting up on hal

Unfortunately I don't have code checked in to svn yet, so we'll have to make do with getting the newest code from me in a tar ball. If you've already done some of the steps above under "Local TAG\_COMM (or TAG) analysis" then I would suggest to get the latest versions of the code (ie: IM or email me and make sure I've put the newest ones up on this TWiki)

There is a new bit of code you have to set up now too, which makes the merged "TAG\_MCP" ntuple. You'll need to create the package yourself, but here are the steps to do it:

Before you begin, download the following file (attached to this wiki):

- requirements

In your working directory:

```
> mkdir generic
> cd generic
```

Copy the requirements file to this directory, and then:

```
> source /afs/cern.ch/sw/contrib/CMT/v1r20p20090520/mgr/setup.sh
> cmt config
```

The above only needs to be done once. However, every time a new terminal window is opened:

```
> source setup.sh -tag=15.5.4.2,AtlasTier0,slc4,gcc34,32,opt,runtime
```

(warning: the above steps will be a little slow because of copying all the way from cern...)

Check out the ARA skeleton package:

```
> cmt co -r AthenaROOTAccess-00-05-58 PhysicsAnalysis/AthenaROOTAccess
```

Now create the new package, called TAGMerger (this is what will eventually be on SVN, but for now you have to put the package together yourself with all the pieces attached to this TWiki)

```
> cmt create TAGMerger TAGMerger-00-00-01 PhysicsAnalysis
> cd PhysicsAnalysis/TAGMerger/
> rm -r src
```

Copy and untar TAGMerger\_h.tgz and src.tgz here:

```
> tar xvzf TAGMerger_h.tgz
TAGMerger/CollectionMetadata.h
TAGMerger/TAG_MCPMaker.h
TAGMerger/CombMuonWord.h
TAGMerger/TAGsMerger_Dict.h
TAGMerger/selection.xml
TAGMerger/TAG_COMM.h
TAGMerger/TAG.h
> tar xvzf TAGMerger_src.tgz
src/TAG_MCPMaker.cxx
src/CombMuonWord.cxx
```

```
> cd ../cmt/
```

- copy requirements2 to this directory (there is already a requirements file there, but you will overwrite it with this one in the next step)

```
> mv requirements2 requirements (had to rename it so that I could attach two versions on this wiki)
```

```
> cmt config
> make
```

Now you should have the package working.

After every edit to the source file, you need to recompile in the cmt directory (this should not be necessary for now). In any case, to do this faster, type

```
> make QUICK=1
```

## 2. Get TAG and TAG\_COMM files

see Get\_TAG\_COMM\_or\_TAG\_data\_files

## 3. Create the TAG\_MCP

Here you will use the newly created TAGMerger package.

Download the attachment:

- forest.C

Edit this file to add the TAG and TAG\_COMM root files you have gotten in step 2. The "CollectionMetadata" tree is not too important at the moment (it isn't working the way it's needed, but for now, just put one TAG root file in the path for "meta\_tree")

Now to create the new tree, do:

```
> root
[0] .x forest.C
```

## 4. Create events list for "tight" and "loose" selection

This is like `Select_events_from_TAG_COMM_or_TAG`, but with new code which merges both for `TAG_MCP`.

Download and untar the attachment:

- `TAG_MCPanalysis.tgz`

Edit `runTAG_MCP.C` to add the path of your newly produced `TAG_MCP` ntuple.

Edit `TAG_MCPanalysis.C` for "tight" and "loose" selection. At this point, be sure to find out exactly what BCIDs are needed, as well as lumi blocks. Contact Domizia if you need help, but this was the info for the following runs:

142174 all LB are good  
 142191 LB 1-43, 131-247  
 142193 LB 25-162  
 all with BCID good crossings: 1, 101, 2774

The "loose" selection consists of only:

```
if( !(PixelTracks > 2 || SCTTracks > 2 || TRTTracks > 2) ) failedSelection = true;
if( !(NsctSPs > 0 || NpixSPs > 0) ) failedSelection = true;
```

The "tight" selection has the other cuts as well:

```
if( RunNumber == [run number] && (LumiBlockN < [low LB bound] || LumiBlockN > [upper LB bound]) ) failedSelection = true;
if( !(BCID==1 || BCID==889 || BCID==2674) ) failedSelection = true;

if( !(ConvertedMBoysSegments > 0 || MooreSegments > 0 || CombinedMuonWord > 0) ) failedSelection = true;

if( !(fabs(LArEtimeDiff)<10 && fabs(MBTStimeDiff)<10) ) failedSelection = true;

if( !(1/fabs(TrackLead1_qoverp) > 2) ) failedSelection = true;

if( !(PixelTracks > 2 || SCTTracks > 2 || TRTTracks > 2) ) failedSelection = true;
if( !(NsctSPs > 0 || NpixSPs > 0) ) failedSelection = true;
```

## 5. Create skimmed ESDs

see `Skimming_ESDs_with_selected_events`

---

-- EmilyThompson - 30-Nov-2009

---

This topic: Main > TAGSelectionToESD

Topic revision: r16 - 2009-12-14 - EmilyThompson



Copyright &© 2008-2024 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