Table of Contents

Get your customized Analysis Code and get ready to analize data in 5 minutes!!

⚠ **Note!** *This is a quick start guide. If you don't know how to setup your ATLAS environment, checking out a package and compiling it, read this first.*

⚠ This version is meant to work with the newest release of ATLASWatchMan and with the **ATLAS release 15.X** . If you are looking for the old version, click here.

Setup the environment

- If you have not done it yet, you have to create your athena directory (only done once) like for athena release 15.6.0 :

```
mkdir -p testarea/15.6.0/cmthome
cd testarea/15.6.0/cmthome
```

In this directory, you have to create a requirements file (look at this link for release recipes). Below is an example of requirements file working at cern :

```
#-----------------------------------------------------------------
set CMTSITE CERN
set SITEROOT /afs/cern.ch
macro ATLAS_DIST_AREA ${SITEROOT}/atlas/software/dist
macro ATLAS_TEST_AREA ${HOME}/scratch0/testarea/15.6.0
apply_tag setup
apply_tag oneTest
apply_tag 32
use AtlasLogin AtlasLogin-* $(ATLAS_DIST_AREA)
#-----------------------------------------------------------------
```

Then, you have to setup CMT :

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

In case of troubles, please check the ATLAS computing workbook

- Go in your `Athena` working directory and set up the ATLAS environment: (more detailed info at ATLASWatchManSetting). For example:

```
cd scratch0/testarea/15.6.0/
source cmthome/setup.sh -tag=AtlasProduction,15.6.0.3
```

Checkout and compile the packages

- Checkout the two packages: ATLASWatchMan and SUSYTools (inserters and formulas) either via cmt :

```
cmt co -r ATLASWatchMan-00-02-04 $SVNGRP/Institutes/Freiburg/SUSY/ATLASWatchMan
cmt co -r SUSYTools-00-02-04 $SVNGRP/Institutes/Freiburg/SUSY/SUSYTools
```

(if you have problems see note (3))

💡 With SVN, in order to see the tags available you can use this command `svn ls $SVNGRP/Institutes/Freiburg/SUSY/ATLASWatchMan/tags/` . The same for `SUSYTools` . In case you have troubles accessing svn, please check the ATLAS SVN wiki.

- Compile the two packages *(see note (1))* :

```
cd Institutes/Freiburg/SUSY/SUSYTools/cmt
cmt make
cd -

cd Institutes/Freiburg/SUSY/ATLASWatchMan/cmt
cmt make
```

Run the example analysis

Generate the analysis code

- Go in the `run/` directory of ATLASWatchMan:

```
cd ../run
```

- Here you will find an example of **Steering File**. In this file you can define your analysis. Remember that you can define as many channels/analyses you want at the same time. To run the example:

```
python -m ATLASWatchMan/ATLASWatchMan_Parser ATLASWatchMan_AnalysisSteeringFile_benchmarkCl
```

You will get the Analysis Code automatically generated, dinamically based on the settings in the Steering File. You can find the generated analysis code in the `python/` folder. You can look at it: it's a plain Python code and it is organized as a standard Athena package; it's very easy to understand indeed! The generated code is at:

```
emacs ../python/ATLASWatchMan_GeneratedAnalysisLib.py
```

Together with the Analysis Code, other files are dinamically generated, among them an Athena jobOption that you will use in the next step.

Run the generated analysis code

- Now you can try to run the example code locally, for example with Athena:
    - ♦ This will run on an example AOD file in Castor (*very slow!!*)

    ```
    athena -c "EvtMax=500" ../share/ATLASWatchMan_Generated_jobOptionAthena.py
    ```
    - ♦ If you have a file ready to run upon (AOD, DPD, ...), you can use the standard Athena command:

    ```
    athena -c "InputCollections=['myNiceFile1', 'myNiceFile2']; EvtMax=500" ../share/ATI
    ```

Of course the file name has to contain the full path if it's not in the `run/` folder.

- • ◆ If you have a text file containing a list of files you can use it like this (custom command):

  ```
  athena -c "CollectionFiles='myNiceList.txt'; EvtMax=500" ../share/ATLASWatchMan_Gene
  ```

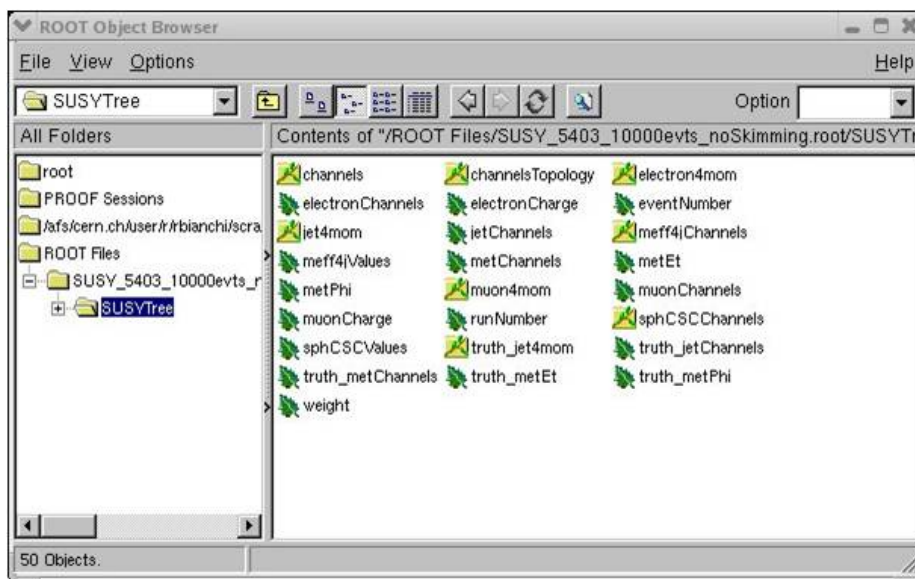# Look into the output D3PD produced

- Open the generated D3PD file with ROOT and look at its content:

  ```
  root -l OutputD3PD.root
  ```

  At the ROOT prompt open a `TBrowser`

  ```
  TBrowser b1;
  ```

  then click on `ROOT files`, then on the file "OutputD3PD" and then on the TTree "D3PDTree". You will get a window like this:

- In the output TTree you have particles and other user-defined quantities (*more details on them in the full documentation*). The important thing to notice is that the TTree contains the output for all the analyses we define in the Steering File. Try to double-click on "channels" branch, and then on "data" method... You will get a distribution of events having passed the analyses/channels defined in the steering file. (*see note (2)*)
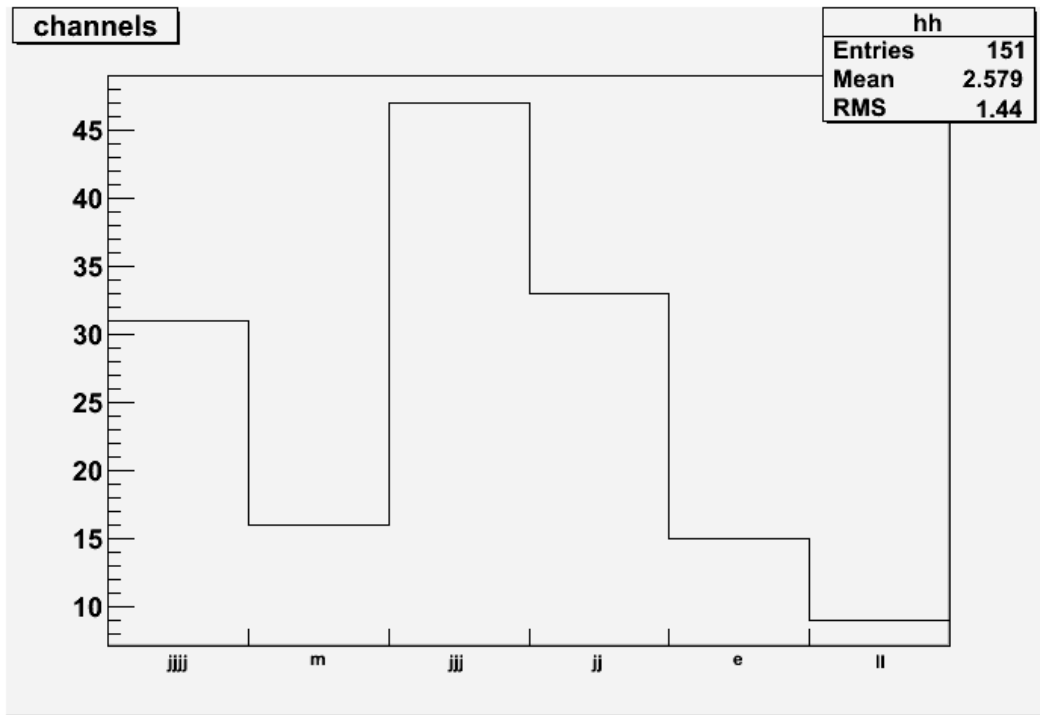
Start changing the example file and get your own automatically generated analysis code!

- Now you can start to edit the Steering File.

```
cp ATLASWatchMan_AnalysisSteeringFile_benchmarkChannels.py ATLASWatchMan_AnalysisSteer
emacs ATLASWatchMan_AnalysisSteeringFile_myAnalysis.py
```

You can start changing simple things, like the required number of jets or leptons in a channel, or the value of `pT` in the `JetPtCuts` field. The name of the variables of the Python dictionary are quite self-explicative, and you can find the full documentation about the Steering File here, together with many practical examples.

- Then run the analysis code generation with

```
python -m ATLASWatchMan/ATLASWatchMan_Parser ATLASWatchMan_AnalysisSteeringFile_myAnalysis
```

- And then run again the analysis as above, for example with:

```
athena -c "InputCollections=["myNiceFile"]; EvtMax=500" ../share/ATLASWatchMan_Generated_j
```

Notes:

- (1) ATLASWatchMan needs to be compiled only **once**, to create symlinks and get the ROOT dict for vector. Even when you change your Steering File to change your analysis you don't need to compile, because almost all the code is written in Python.

- (2) In order to use the interactive ROOT `TBrowser` to navigate and plots the particles `4mom` , you have to use at least ROOT version 5.22. The ROOT version shipped with Athena (5.18 so far, for 14.2.20) does not handle `vector< TLorentzVector >` correctly. Future versions of Athena will have the right ROOT version. But **you can access all the data** from a ROOT macro, a C++ or a Python code, **as usual**. More examples in the full version of the ATLASWatchMan documentation.

- (3) A small hint: when you get problems with SVN first always try to see what you get with

```
    echo $SVNGRP
```

If it's empty or it's not svn+ssh://svn.cern.ch/reps/atlasgrp, so probably the Athena environment in your shell got corrupted in some way. So open a new shell, set up Athena and try again. Sometimes we also observed temporary malfunctioning in the SVN and/or AFS server. After a while they usually work again.
If CMT does not work for you, you may try with SVN (but CMT is preferable):

```
    svn co $SVNGRP/Institutes/Freiburg/SUSY/ATLASWatchMan/tags/ATLASWatchMan-00-02-01 Insti
    svn co $SVNGRP/Institutes/Freiburg/SUSY/SUSYTools/tags/SUSYTools-00-02-01 Institutes/Fr
```

-- RiccardoMariaBianchi - 28 Jan 2009 -- JanetDietrich - 20 Mar 2009

This topic: Main > ATLASWatchManQuickStart
Topic revision: r21 - 2009-12-09 - RenaudBruneliere