# Files and macros in the ROOT framework

Computational Physics M.Sc.
20/03/2017

# Contents

- ROOT files
  - Trees and branches
  - Flat NTuples
  - Opening, exploring, creating

- Macros
  - Named
  - Unnamed
  - Compiled C++ code

# Files in ROOT

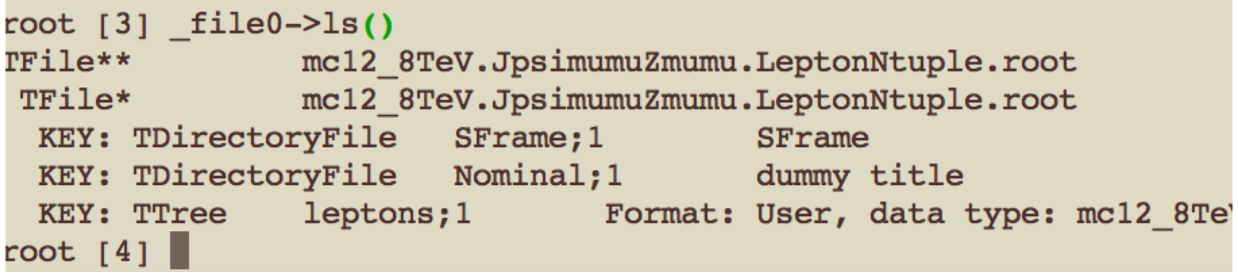
- A ROOT file (.root) is a machine-independent format containing all the data (and the description of the data)
- ROOT uses file compression when saving, thus reducing size and allowing for large amount of data to be stored in a file
- A file in ROOT is following a directory structure, as in the folders hierarchy of an OS. But it can also get more complicated (**TTrees**) or even be much simpler (i.e flat **NTuples**, simple functions and/or histograms, e.t.c)

ROOT files are described by the TFile class:
https://root.cern.ch/doc/master/classTFile.html

More on ROOT files:
https://root.cern.ch/input-and-output

# ROOT Trees

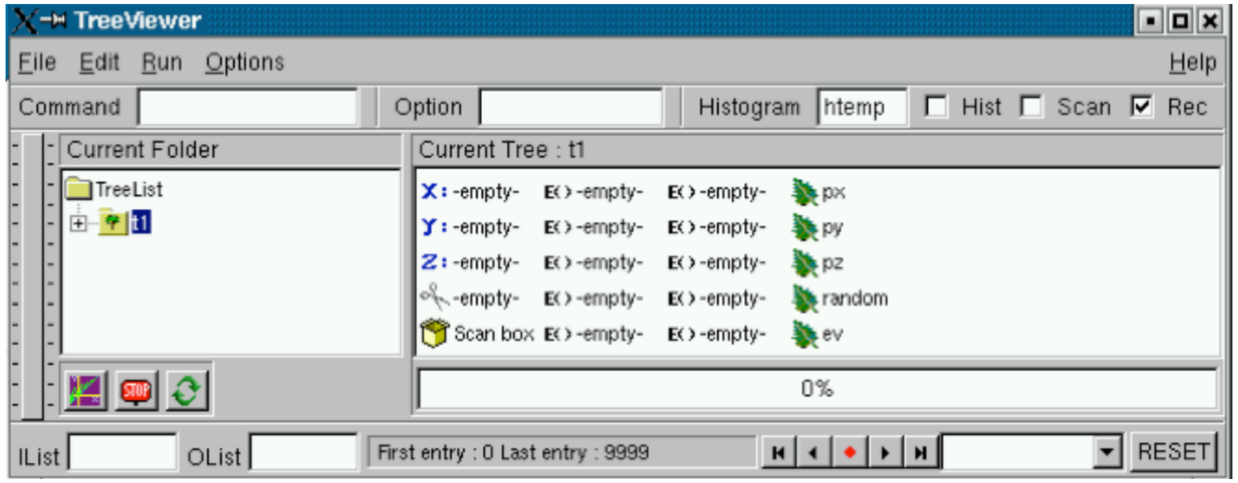
**Q**: What is a tree in ROOT?
**A**: A TTree is a data structure containing several variable types



```
root [3] _file0->ls()
TFile**         mc12_8TeV.JpsimumuZmumu.LeptonNtuple.root
 TFile*         mc12_8TeV.JpsimumuZmumu.LeptonNtuple.root
  KEY: TDirectoryFile   SFrame;1        SFrame
  KEY: TDirectoryFile   Nominal;1       dummy title
  KEY: TTree    leptons;1       Format: User, data type: mc12_8Te
root [4]
```

- A TTree can contain branches (subdirectories) and leaves (variables that contain the actual data)
- We can Print(), Scan() a TTree, or Draw() one of the leaves

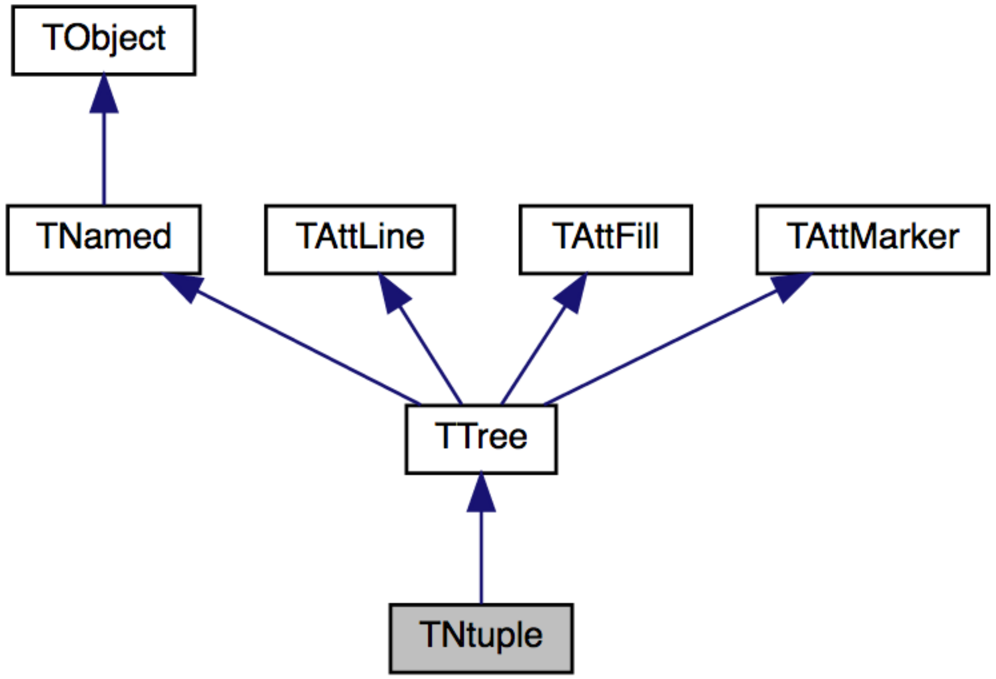
```
mytree->Print();                        //list all variables in the tree
mytree->Draw(“track momentum”);  //name of one column
mytree->Draw(“px:py”);                  //scatter plot
mytree->Draw(“px:py”,”pz>5”);           //scatter plot with cut
mytree->Scan(“px:py”,”pz>5”);    // Print out values with cut
```



[http://root.cern.ch/root/htmldoc/TTree.html](http://root.cern.ch/root/htmldoc/TTree.html)

# NTuples

**Q**: What is the NTuple?
**A**: The simple version of a TTree.



NTuples contain only floating-point variables where each and every one of them goes to a separate branch

Basically to keep the backwards compatibility with PAW…

https://root.cern.ch/doc/v608/classTNtuple.html

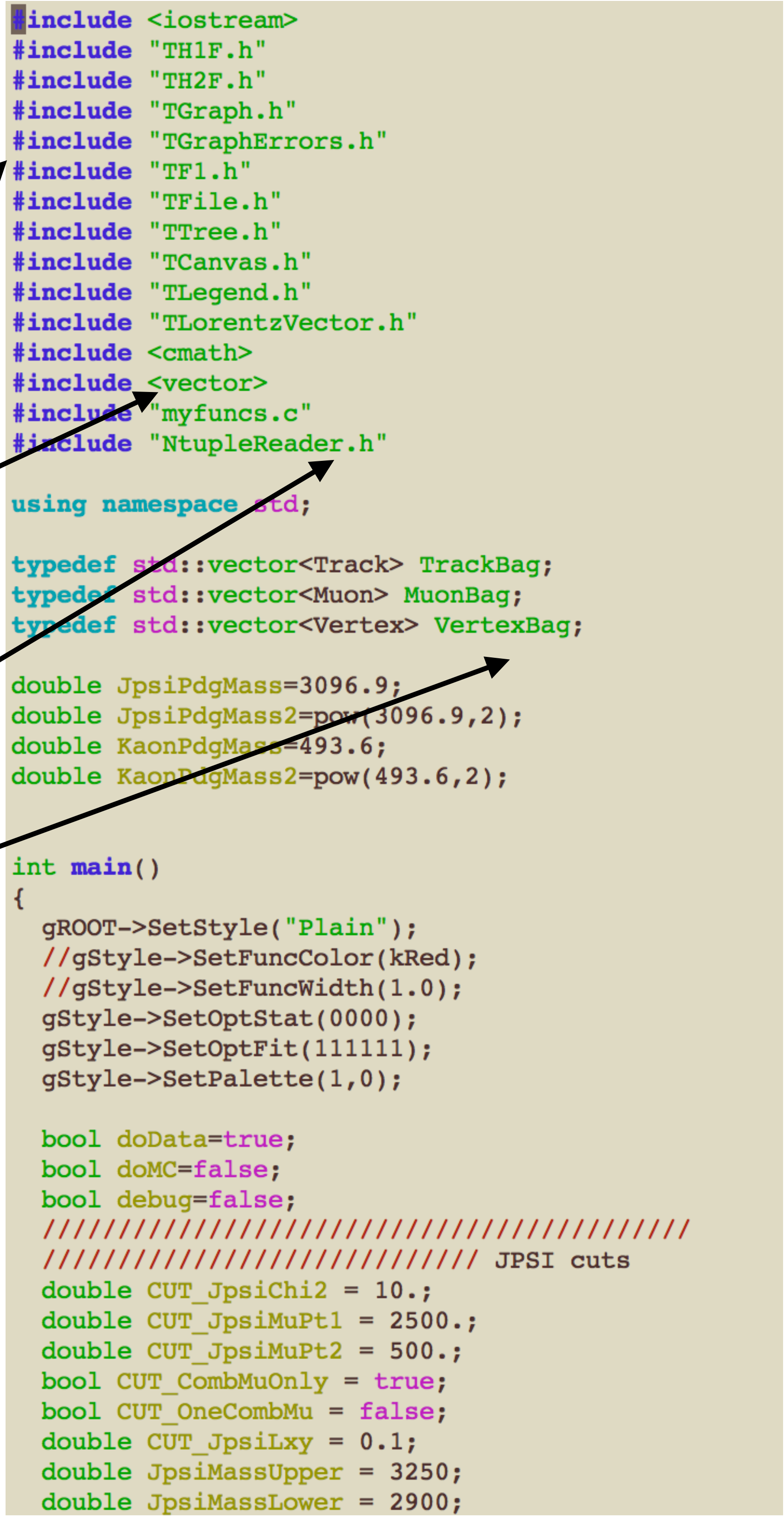
# Macros

- Working with the ROOT prompt is not the most efficient way!
- A ROOT macro is a series of C++ directives that ROOT executes (or rather interprets, by means of CINT)
- Macros can be divided into named and unnamed
- They have to be contained either in brackets {} [unnamed macros] or be more complex (i.e contain functions - and at least one with the same name as the macro itself) [named macros]
- We can use a macro to call another macro (to call another macro, to…)
- The simplest unnamed macro ROOT understands and executes: {}

https://root.cern.ch/working-macros

# Compiled C++ code

- There is a library for all ROOT objects that can be linked into a C++ program
- Makes execution much faster



```
#include <iostream>
#include "TH1F.h"
#include "TH2F.h"
#include "TGraph.h"
#include "TGraphErrors.h"
#include "TF1.h"
#include "TFile.h"
#include "TTree.h"
#include "TCanvas.h"
#include "TLegend.h"
#include "TLorentzVector.h"
#include <cmath>
#include <vector>
#include "myfuncs.c"
#include "NtupleReader.h"

using namespace std;

typedef std::vector<Track> TrackBag;
typedef std::vector<Muon> MuonBag;
typedef std::vector<Vertex> VertexBag;

double JpsiPdgMass=3096.9;
double JpsiPdgMass2=pow(3096.9,2);
double KaonPdgMass=493.6;
double KaonPdgMass2=pow(493.6,2);


int main()
{
  gROOT->SetStyle("Plain");
  //gStyle->SetFuncColor(kRed);
  //gStyle->SetFuncWidth(1.0);
  gStyle->SetOptStat(0000);
  gStyle->SetOptFit(111111);
  gStyle->SetPalette(1,0);

  bool doData=true;
  bool doMC=false;
  bool debug=false;
  //////////////////////////////////////////
  ///////////////////////////// JPSI cuts
  double CUT_JpsiChi2 = 10.;
  double CUT_JpsiMuPt1 = 2500.;
  double CUT_JpsiMuPt2 = 500.;
  bool CUT_CombMuOnly = true;
  bool CUT_OneCombMu = false;
  double CUT_JpsiLxy = 0.1;
  double JpsiMassUpper = 3250;
  double JpsiMassLower = 2900;
```

ROOT libraries

STL libraries

User classes

typedefs

- More on that to follow…