# CLAS12TOOL is now CLAS12ROOT

Derek Glazier

13/11/19

# Clas12root Download

You will first  need hipo

git clone --recurse-submodules https://github.com/gavalian/hipo

git clone --recurse-submodules https://github.com/jeffersonlab/clas12root.git

cd clas12root

git checkout tutorial

setenv CLAS12ROOT $PWD

setenv PATH "$PATH":"$CLAS12ROOT/bin"

setenv HIPO /Where/did/I/put/my/hipo

./installC12Root

Get file from ifarm : ~devita/skim.hipo
Or http://nuclear.gla.ac.uk/~dglazier/skim.hipo

# CLAS12ROOT on ifarm

source /group/clas12/packages/setup.csh
(or setup.sh for bash)

module load clas12/pro

# Why CLAS12ROOT ?

ROOT provides many useful tools for data analysis

CLAS12 HiPO file is compact

```
npcglazier2.dglazier> ls -lh /work/jlab/clas12data/v16/skim9_5038.hipo
-rw-r--r-- 1 dglazier dglazier 3.7G Aug 30 10:25 /work/jlab/clas12data/v16/skim9_5038.hipo
npcglazier2.dglazier> ls dst2root/skim9_5038.hipo.root  -lh
-rw-r--r-- 1 dglazier dglazier 5.7G Oct 29 16:40 dst2root/skim9_5038.hipo.root
```

and fast to read  (7.5 Gb ROOT 7.0 Gb Hipo SSD)

```
treeLZ4->Draw("sqrt(px*px+py*py+pz*pz)*sqrt(vx*vx+vy*vy+vz*vz)
                *status/4000.0*pid>>(200,-10000,10000)","","hist");
```

| ROOT TTREE | ROOT DATAFRAME | HIPO CLAS12ROOT | HIPO CLAS12ROOT dev |
|---|---|---|---|
| 438 s | 223 s | 40 s | 28 s |

And is the default format for DSTs,
no overhead for converting to ROOT

# CLAS12ROOT @JeffersonLab Github

JeffersonLab / **clas12root**

Unwatch 6 | Star 0 | Fork 1

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Security | Insights | Settings

CLAS12 analysis tools for HIPO data using C++ and ROOT

Edit

Manage topics

**16** commits | **1** branch | **1** release | **2** contributors

Branch: **master** | **New pull request** | **Create new file** | **Upload files** | **Find file** | **Clone or download**

**dglazier** Merge pull request #2 from dglazier/master ··· Latest commit cd92b82 on 30 Aug

| | | |
|---|---|---|
| Clas12Banks | Fix FTBasedPid so clas12reader::getByID is based on FT based Pid if c… | 2 months ago |
| Clas12Root | add ftb particle to bank hist | 2 months ago |
| RunRoot | refactor from clas12tool to clas12root | 3 months ago |
| cmake | #1 | 3 months ago |
| docs | #1 | 3 months ago |
| hipo4 | #1 | 3 months ago |

# CLAS12ROOT @JeffersonLab Instructions

## Clas12Root

Data Analysis Tools for hipo4 data format.

Examples are given for running in interactive ROOT sessions and ROOT-Jupyter notebooks.

### Clas12Banks -> Clas12Root

NEW We now use an external hipo4 repository. This must be pointed at with the variable HIPO when installing. The files from hipo/hipo4 will be copied here to Hipo4.

For Hipo library see https://github.com/gavalian/hipo

The Clas12Banks implementation can be used independent of ROOT, although currently ROOT dictionaries are created for the classes via cmake (this could be removed). This defines the specific CLAS12 DST banks and provides an interface to the data.

For actual Clas12Banks definitions see HIPO4 DSTs

### Also see c++ function for accessing banks "Cheat sheet" AccesssingBankDataInCpp.txt in the top level directory.

The Clas12Root package depends on both Hipo and Clas12Banks. This provides ROOT-like analysis tools for operating on clas12 hipo DSTs.

```
HipoDraw
HipoTreeMaker
HipoProof
```

### To Download

git clone --recurse-submodules https://github.com/jeffersonlab/clas12root.git

cd clas12root

### To setup Run ROOT

for cshrc

setenv CLAS12ROOT $PWD (the actual path can be added in your bashrc or tchrc)

setenv PATH "$PATH":"$CLAS12ROOT/bin"

setenv HIPO /Where/is/hipo

or for bash

export CLAS12ROOT=$PWD

export PATH="$PATH":"$CLAS12ROOT/bin"

export HIPO=/Where/is/hipo

### To install

installC12Root

If there are issues with cmake and your ROOTSYS you can try using the local FindROOT file. Edit the CMakeList.txt files removing the lines with comment ##USEROOTSYS and uncomment the line

```
######include("cmake/FindROOT.cmake")
```

**Note just to help confuse you the first letter in bank items have been capitilised so where you might expect to use REC::Particle::pz, or particle->getpz(), you will find Pz will work better, similarly Time, Energy...You can check the bank header files e.g. Clas12Banks4/particle.h and look at the get function declarations e.g. getPz() rather than getpz() **

### interactive root session

To start an interactive session with pre-loaded Clas12Root use clas12root instead of root on the command line.

### Ex 0 Plotting an item from any bank

This is faster than the particle draw as it only requires the reading of 1 bank.

```
clas12root

BankHist bankDraw("/WHERE/IS/MY/HIPO/file.hipo");
bankDraw.Hist1D("REC::Particle::Px",100,0,10,"")->Draw()
bankDraw.Hist1D("REC::Scintillator::Time",1000,0,200,"")->Draw()
```

You can group histograms together for lazy execution if they all come from the same bank.

```
    bankDraw.Hist1D("REC::CovMat::C11",100,0,1,"")
    bankDraw.Hist1D("REC::CovMat::C22",100,0,1,"")
    bankDraw.Hist1D("REC::CovMat::C33",100,0,1,"")
    bankDraw.Hist1D("REC::CovMat::C44",100,0,1,"")
    bankDraw.Hist1D("REC::CovMat::C55",100,0,1,"")->Draw("(3x2)")
```

### Ex 1 Looping over events and getting particle information

The clas12reader class performs the correlation of particle and detector inofmation (aka reverse l
over particles you are looping over region_particle (see Clas12Banks for full reference). Each reg
own definition of a region_particle so it will only return meaningful data (i.e. a CD particle will retu
information). In addition the getTime, getPath, getDetEnergy functions have predefined meaning
getTime returns FTCAL time, for FD it returns FTOF1A if it exists, if not it will try FTOF1B, FTOF2

###NEW you can add hipo file tags to clas12reader e.g.

```
clas12reader c12("file.hipo",{0,1,6});
```

You can inspect the code $CLAS12ROOT/RunRoot/Ex1_CLAS12Reader.C for more guidance or

To run:

```
clas12root $CLAS12ROOT/RunRoot/Ex1_CLAS12Reader.C+ --in=/WHERE/IS/MY/HIPO/file.
```

Note the use of the + sign after the macro name. This compiles the script meaning it will run muct

#### Jupyter

To install rootbooks see https://root.cern.ch/how/how-create-rootbook

mkdir myNotebooks

cp -r $CLAS12ROOT/RunRoot/jupy myNotebooks/.

cd myNotebooks/jupy

Start a ROOT note book :

```
root --notebook
```

Click on the notebook CLAS12Reader3Pl.ipynb and follow the tutorial

### Ex 2 Drawing particle histograms from hipo files

```
particleDraw /WHERE/IS/MY/HIPO/file.hipo
```

Or chain together files with wildcard, note the ' '

```
particleDraw '/WHERE/IS/MY/HIPO/file_*.hipo'
```

You will get an interactive ROOT prompt where you can draw histograms:

```
ParticleHist [0] hists.Hist1D("P.P",100,0,10,"P.P")
ParticleHist [1] hists.Hist2D("P.P:P.DetEnergy",100,0,10,100,0,5,"P.P")->Draw("(2x1)")
```

Note you only have to call draw once, and then it only has to loop over the date once. The option (2x1) specifies the dimensions of the pads in the produced canvas, the parenthesis is required.

Remember at the end you can save all the histograms to file :

```
ParticleHist [0] hists.Save("HistFile.root");
```

Instead of drawing histograms interactively at the prompt you may give predefined histograms via a script e.g. :

```
particleDraw /WHERE/IS/MY/HIPO/file.hipo Ex2_HipoDraw.C
```

See $CLAS12ROOT/RunRoot/Ex2_HipoDraw.C for details.

There are predefined aliases for DST bank detector layers :

```
ECIN. , ECOUT. , PCAL. , FTOF1A. , FTOF1B. , FTOF2. , CTOF. , CND1. , CND2. , CND3. , FTCAL. , FTHODO.
e.g. ECIN.Energy  , HTCC.Nphe , DC.TrCh12 , CTOF.Time
```

The REC::Particle bank should be directly accessed with

```
PBANK.
e.g. PBANK.Pid , PBANK.Px
```

The FT based equivalent PID varaibles can be accessed from the particle bank by

```
e.g. PBANK.FTBPid , PBANK.FTBBeta
```

The region particle has derived quantities such as theta and phi as well as selected variabels for a particle for example time from a particular ToF layer. Note the order of precedence for the FD is TOF1B, TOF1A, TOF2, PCAL and DeltaEnergy is the corresponding timing detector energy. These should be accessed with

```
P.
e.g. P.Theta , P.P , P.Phi , P.Region , P.Time , P.DetEnergy , P.DeltaEnergy , P.Path , P.Pid , P.Calc
e.g. P.Region==FT,  P.Region==FD
```

For REC::EVNT use (adding FTB for RECFT::EVNT banks)

```
e.g. EVNT.StartTime or EVNT.FTBStartTime
```

For Run::config

```
e.g. RUN.Trigger
```

#### Jupyter

Go to directory containing notebooks e.g. $CLAS12ROOT/RunRoot/jupy

Start a ROOT note book :

```
root --notebook
```

Click on the notebook HipoDraw.ipynb and follow the tutorial

### Ex 4 Filtering and Skimming into a ROOT ntuple (tree)

```
particleTree  /WHERE/IS/MY/HIPO/file.hipo /OUTPUT/tree.root Ex4_TreeMaker.C
```

Or chain together files with wildcard, note the ' '

```
particleTree  '/WHERE/IS/MY/HIPO/file_*.hipo' /OUTPUT/tree.root Ex4_TreeMaker.C
```

The script $CLAS12ROOT/RunRoot/Ex4_TreeMaker.C defines which branches are to be written and what cuts to put on the event topology. You can copy and edit this file to do what you want. The branches should use the conventions above for accessing different bank items e.g.

```
treemaker.Branch("P.Time/F"); //create tree with time branch
treemaker.Branch("PBANK.Px/F"); //create tree with particle Px branch
```

You can perform some arithmetic and define a new branch e.g.

```
        treemaker.Branch("P.Time-EVNT.StartTime/F","Time"); //branch name Time
treemaker.Branch("P.Time-EVNT.FTBStartTime/F","FTBTime"); //branch name FTBTime

treemaker.AddExactPid(11,1); //filter events with exactly 1 e-
treemaker.AddAtLeastPid(211,1);//and at least 1 pi+
treemaker.AddZeroOfRestPid(); //and zero of any other particle type (default is any)
```

#### Jupyter

Go to directory containing notebooks e.g. $CLAS12ROOT/RunRoot/jupy

Start a ROOT note book :

```
root --notebook
```

Click on the notebook HipoToRootTree.ipynb and follow the tutorial

### Ex 3 Using HipoSelector & PROOFLite

This assumes you are aware of and understand the ROOT TSelector and PROOF scheme. See https://root.cern.ch/proof.

Create a HipoSelector (similar to tree->MakeSelector("mySelector");), using the executable makeHipoSelector :

```
makeHipoSelector mySelector
```

You should use some meaningful name rather than mySelector. Edit it to perform the tasks you would like. But use the ProcessEvent function instead of the Process function as you would in a TSelector. You can use the _c12 clas12reader object to access all the data as shown in Ex1_CLAS12Reader.C

e.g.

```
Bool_t HipoFileSelector::ProcessEvent(){

    _hist1->Fill(_c12->head()->getStartTime());
            return kTRUE;
}
```

You may also add event selections as above using the AddFilter function

```
    void testSelector::AddFilter(){
        _c12->addExactPid(11,1);    //exactly 1 electron
    }
```

To execute (note the + is important) :

```
clas12proof 4 mySelector.C+ Ex3_ProofLite.C
```

Note 4 = number of workers used, you should change this to however many you would like.

Note mySelector is hard-coded in Ex3_ProofLite.C so for your own selector you should copy and edit this script.

As a more complete example you can check testSelector in RunRoot which implements the particle analysis and histogramming from Ex1. This can be run with Ex3b_TestSelector.C once you change the HipoChain files :

```
clas12proof 8 RunRoot/testSelector.C+  RunRoot/Ex3b_TestSelector.C
```

# CLAS12ROOT @JeffersonLab Notebooks

## Creating Root tree files with hiporoot::ParticleTree

First load the classes into the notebook

In [ ]:
```
gROOT->ProcessLine(".x $CLAS12ROOT/RunRoot/LoadClas12Root.C");
```

Create the tree maker with the full path to the hipo file you want to analyse. You may also give wildcard (*) arguments .

In [ ]:
```
ParticleTree treemaker("/WHERE/IS/MY/HIPO/file.hipo","/WHERE/SHOULD/I/PUT/MY/tree.root");
//ParticleTree treemaker("/work/jlab/clas12data/dst_skim4_5038.hipo","tree.root");
```

Create some tree branches. There are predefined aliases for DST bank detector layers :

```
      ECIN. , ECOUT. , PCAL. , FTOF1A. , FTOF1B. , FTOF2. , CTOF. , CND1. , CND2. , CND3. ,
  FTCAL. , FTHODO. , HTCC. , LTCC. , DC. , CVT.
      e.g. ECIN.Energy  , HTCC.Nphe , DC.TrChi2 , CTOF.Time
```

The REC::Particle bank should be directly accessed with

```
      PBANK.
      e.g. PBANK.Pid , PBANK.Px
```

Ther REC::EVNT bank :

```
      EVNT.
      e.g. EVNT.StartTime,
```

The region particle should be accessed with

```
      P.
      e.g. P.Theta , P.P , P.Phi , P.Region , P.Time , P.DetEnergy , P.DeltaEnergy , P.Path
  , P.Pid , P.CalcMass
```

### Create branches

Here we can create tree branches correlating particle and detector information.

I can make a branch for any DST bank item using the aliases above.

I can choose standard ROOT branch type e.g /F for float /I for int /D for double,....

In [ ]:
```
treemaker.Branch("PBANK.Px/F");
treemaker.Branch("PBANK.Py/F");
treemaker.Branch("PBANK.Pz/F");
treemaker.Branch("PBANK.Vx/F");
treemaker.Branch("PBANK.Vy/F");
treemaker.Branch("PBANK.Vz/F");
treemaker.Branch("PBANK.Pid/I");
treemaker.Branch("P.Time/F");
treemaker.Branch("EVNT.StartTime/F");
```

## Drawing histograms with hiporoot::ParticleHist

First load the classes into the notebook

In [ ]:
```
gROOT->ProcessLine(".x $CLAS12ROOT/RunRoot/LoadClas12Root.C");
```

Create the histogram maker with the full path to the hipo file you want to analyse. You may also give wildcard (*) arguments .

In [ ]:
```
ParticleHist hists("/WHERE/IS/MY/HIPO/file.hipo");
//ParticleHist hists("/work/jlab/clas12data/dst_skim4_5038.hipo");
```

Turn on javascript ROOT for interactive histograms

In [ ]:
```
%jsroot
```

Draw some histograms. There are predefined aliases for DST bank detector layers :

```
      ECIN. , ECOUT. , PCAL. , FTOF1A. , FTOF1B. , FTOF2. , CTOF. , CND1. , CND2. , CND3. ,
  FTCAL. , FTHODO. , HTCC. , LTCC. , DC. , CVT.
      e.g. ECIN.Energy  , HTCC.Nphe , DC.TrChi2 , CTOF.Time
```

The REC::Particle bank should be directly accessed with

```
      PBANK.
      e.g. PBANK.Pid , PBANK.Px
```

The region particle should be accessed with

```
      P.
      e.g. P.Theta , P.P , P.Phi , P.Region , P.Time , P.DetEnergy , P.DeltaEnergy , P.Path
  , P.Pid , P.CalcMass
```

### Drawing hists 1

First draw a 1D histogram of the time difference between FTOF1A and FTOF1B

Second, draw a 2D hist of the time difference versus the particle theta, with colour map.

Plot the 2 histograms side-by-side (2x1)

In [ ]:
```
hists.Hist1D("FTOF1B.Time-FTOF2.Time",1000,-5,5,"FTOF1B.Time-FTOF2.Time");
hists.Hist2D("FTOF1B.Time-FTOF2.Time:P.Theta*TMath::RadToDeg()",50,-5,5,50,0,40,"FTOF1B.Time-FTO
F2.Time")->Draw("(2x1)col1");
```

### Drawing hists 2

Now draw the $\theta$ versus $\phi$ distributions for different particle types

In [ ]:
```
hists.Hist2D("P.Theta*TMath::RadToDeg():P.Phi*TMath::RadToDeg()",180,0,180,180,-180,180,"P.Pid==
```

# CLAS12ROOT @JeffersonLab Forum

https://clas12.discourse.group/c/clas12root

Clas12Root | all tags | Latest | Top | + New Topic

| Topic | Replies | Views | Activity |
|---|---|---|---|
| Reading Bank information in clas12root | 3 | 20 | 2d |
| Same information in CND1 and FTOF1A | 4 | 15 | 2d |
| last visit | | | |
| Using FTbased Pid | 0 | 13 | Aug 30 |
| Restructuring Clas12Tool and Hipo | 6 | 33 | Aug 15 |
| Compile a project with g++ and Clas12Tool as a library | 1 | 27 | Aug 13 |
| Clas12tool on OSX | 2 | 29 | Aug 1 |
| Cherenkov # photoelectrons | 1 | 17 | Jul 26 |
| About the Clas12Root category | 0 | 13 | Jul 26 |

# HIPO @gavalian Github

gavalian / **hipo**

Watch 0 | Star 0 | Fork 1

Code | Issues 3 | Pull requests 0 | Projects 0 | Wiki | Security | Insights

High Performance Output Data format for experimental Physics

**20** commits | **1** branch | **0** releases | **2** contributors

Branch: **master** | **New pull request** | **Create new file** | **Upload files** | **Find file** | **Clone or download**

**Gagik Gavalian** and **Gagik Gavalian** improved benchmark printout — Latest commit 99aed92 4 days ago

| | | |
|---|---|---|
| examples | improved benchmark printout | 4 days ago |
| hipo4 | inlining some of the get functions. | 4 days ago |
| lz4 @ 798301b | first commit to the repository | 5 months ago |
| python | added python interface for reading hipo files with ctypes | 12 days ago |
| .gitmodules | first commit to the repository | 5 months ago |
| Makefile | added python interface for reading hipo files with ctypes | 12 days ago |
| README.md | updated the readme file | 5 months ago |

# BankHist – drawing items from a bank

Create a new ROOT macro : BankHist.C

```
{
  BankHist bankDraw("/where/is/myHipo.hipo");

  bankDraw.Hist1D("REC::Particle::Pid",10000,-3000,3000,"")->Draw();

}
```

Run :

clas12root BankHist.C

# BankHist – drawing items from a bank

Create a new ROOT macro : BankHist.C

```
{
  BankHist bankDraw("/where/is/myHipo.hipo");

  bankDraw.Hist2D("REC::Particle::Px:REC::Particle::Py",100,-3,3,100,-3,3,"");

  bankDraw.Hist1D("REC::Particle::Pid",10000,-3000,3000,"")->Draw("colz");

}
```

Standard ROOT TH options

Run :

clas12root BankHist.C

# BankHist – drawing items from a bank

Create a new ROOT macro : BankHist.C

```
{
  BankHist bankDraw("/where/is/myHipo.hipo");

  bankDraw.Hist2D("REC::Particle::Px:REC::Particle::Py",100,-3,3,100,-3,3,"");

  bankDraw.Hist1D("REC::Particle::Pid",10000,-3000,3000,"");

  bankDraw.Hist1D("sqrt(REC::Particle::Px*REC::Particle::Px"
                  "+ REC::Particle::Py*REC::Particle::Py"
                  "+ REC::Particle::Pz*REC::Particle::Pz)"
                 ,200,0,10,"")->Draw("(3x1)colz");

  bankDraw.Save("BankHists.root");
}
```

Run :

clas12root BankHist.C

# ParticleDraw - Drawing particle banks Region particle functions

In addition to actual banks the particle object has some additional functions wehre some choices have been made depending on which region they are in

| C++ code | particleDraw | |
|---|---|---|
| particles[i]→getPid(); | P.Pid | |
| particles[i]→getTime(); | P.Time | |
| particles[i]→getPath(); | P.Path | |
| particles[i]→getDeltaEnergy(); | P.DeltaEnergy | |
| particles[i]→getSector(); | P.Sector | |
| particles[i]→getRegion(); | P.Region | (FD, CD, FT) |
| particles[i]→getTheta(); | P.Theta | |
| particles[i]→getPhi(); | P.Phi | |
| particles[i]→getP(); | P.P | |

For FT Time, Path, comes from FTCAL, DeltaEnergy from FTHODO
For FD, Time, Path, DeltaEnergy comes from FTOF1B, FTOF1A, FTOF2, PCAL in order of preference
For CD Time, Path, DeltaEnergy comes from CTOF, then CND if no CTOF

# ParticleDraw – Drawing particle banks with Correlated detector banks

## Drawing particle e- momentum

particleDraw /*where*/is/my/hipo/file.hipo

```
hists.Hist1D("P.P",200,0,10,"P.Pid==11")->Draw();
```

* run from script or type in interactive ROOT command line

# ParticleDraw – Drawing particle banks with Correlated detector banks

Drawing particle e- momentum using REC::Particle => PBANK

particleDraw */where/*is/my/hipo/file.hipo

```
hists.Hist1D("sqrt(PBANK.Px*PBANK.Px+PBANK.Py*PBANK.Py+PBANK.Pz*PBANK.Pz)"
             ,200,0,10,"P.Pid==11")->Draw("");
```

* run from script or type in interactive ROOT command line

# ParticleDraw – Drawing particle banks with Time

Bank aliases

e.g. FTOF1A, CTOF, CND1, CND2, CND3, FTCAL, …..

particleDraw /*where*/is/my/hipo/file.hipo ParticleHist.C

```
hists.Hist1D("P.Time-EVNT.StartTime",1000,0,100,"P.Time");
hists.Hist1D("FTOF1A.Time-FTOF1B.Time",1000,-10,10,
             "FTOF1A.Time&&FTOF1B.Time");
hists.Hist1D("CTOF.Time-EVNT.StartTime",1000,0,100,"CTOF.Time");
hists.Hist1D("EVNT.FTBStartTime-EVNT.StartTime",1000,-200,200,
             "EVNT.FTBStartTime")->Draw("(2x2)");
```

* run from script or type in interactive ROOT command line

# ParticleDraw – Drawing particle banks In 2D

Bank aliases

e.g. FTOF1A, ECIN , ECOUT, PCAL …..

particleDraw /*where*/is/my/hipo/file.hipo ParticleHist2.C

```
hists.Hist2D("PBANK.P:HTCC.Nphe",200,0,10,50,0,50,"");
hists.Hist2D("PBANK.Beta:FTOF1A.Path/(FTOF1A.Time-EVNT.StartTime)",
             200,0,1,100,0,100,"");
hists.Hist2D("P.P : ECIN.Energy",200,0,10,100,0,1,"");
hists.Hist2D("P.DetEnergy : ECOUT.Energy + ECIN.Energy +PCAL.Energy",
             200,0,3,100,0,3,"P.Pid==11&&P.Region==FD")->Draw("(2x2)col1");
```

* run from script or type in interactive ROOT command line

# Clas12root event loops

This is not a recommended framework for full CLAS12 data analysis

Analysis should be performed in a more general framework which can handle the
large data volumes in a reproducable manner

The following just shows how you may use parts of clas12root in such a framework

# Clas12root event loops
## Loop over files

You can do this however you like!

Here I use clas12root::HipoChain normally used for clas12proof

```
HipoChain chain;
chain.Add("/WHERE/IS/MY/HIPO/file.hipo");


//loop over files
for(int ifile=0;ifile<chain.GetNFiles();++ifile){
  clas12reader c12{chain.GetFileName(ifile).Data()};
```

* Note, each file gets a new reader

This can be made nicer...

# Clas12root event loops
## create standard ROOT stuff

```
//create particles before looping to be more efficient
TLorentzVector p4_gamma1;
TLorentzVector p4_gamma2;

//Create histogram
TH1F hmass("pi0mass","Invariant Mass to 2#gamma",100,0,0.6);
TH1F htime("DeltaTime","Time difference of 2#gamma",100,-10,10);
```

* You might prefer to use the ROOT GenVector  4-vector, it is faster

# Clas12root event loops
## Configuring the reader

```
c12.addExactPid(11,1);     //exactly 1 electron
c12.addExactPid(211,1);     //exactly 1 pi+
c12.addExactPid(-211,1);     //exactly 1 pi-
c12.addExactPid(2212,1);     //exactly 1 proton
c12.addExactPid(22,2);     //exactly 2 gamma

c12.addZeroOfRestPid();  //nothing else
c12.useFTBased(); //and use the Pids from RECFT
```

Also at least option….

```
c12.addAtLeastPid(22,2);     //at least 2 gamma
```

# Clas12root event loops
## Loop over events and get particles

```
//loop over all events in the file
while(c12.next()==true){

  if(c12.getDetParticles().empty())
    continue;

   auto electron=c12.getByID(11)[0];
   auto gamma1=c12.getByID(22)[0];
   auto gamma2=c12.getByID(22)[1];
   auto proton=c12.getByID(2212)[0];
   auto pip=c12.getByID(211)[0];
   auto pim=c12.getByID(-211)[0];
```

# Clas12root event loops

## Do ROOT stuff

```
p4_gamma1.SetXYZM(gamma1->par()->getPx(),gamma1->par()->getPy(),
                  gamma1->par()->getPz(),0);
p4_gamma2.SetXYZM(gamma2->par()->getPx(),gamma2->par()->getPy(),
                  gamma2->par()->getPz(),0);

auto pi0 = p4_gamma1 + p4_gamma2;

//Fill histograms if gammas are in FD
if(gamma1->getRegion()==FD && gamma1->getRegion()==FD){
  hmass.Fill(pi0.M());
  htime.Fill(gamma1->getTime() - gamma2->getTime() );
}
```

# Clas12root other stuff

Filter events into ROOT ntuples

particleTree

Run multicore over multiple files

HipoSelector and HipoChain