

Jan 17, 15 12:37	Analyzer.h	Page 1/3
------------------	------------	----------

```

////////////////////////////////////
// This class has been automatically generated on
// Mon Nov 25 10:41:51 2013 by ROOT version 5.34/09
// from TTree EventTree/The Tree with the variable used to performe the calculat
ion of energy deposition on the HEPD detector
// found on file: Electron5MeV_4M.root
////////////////////////////////////

#ifndef Analyzer_h
#define Analyzer_h

#include <TROOT.h>
#include <TChain.h>
#include <TFile.h>
#include <TH1F.h>
#include <TH2F.h>
#include <TProfile.h>
#include <vector>

// Header file for the classes stored in the TTree if any.
#include "RootEvent.hh"
#include "RootTrack.hh"
#include "RootVertex.hh"
#include "CaloRootHit.hh"
#include "TrackerRootHit.hh"
#include <TObject.h>
#include <TVector3.h>

class Analyzer {
public :
    TTree          *fTree;    //!

```

Jan 17, 15 12:37	Analyzer.h	Page 2/3
------------------	------------	----------

```

{
    // if parameter tree is not specified (or zero), connect the file
    // used to generate this class and read the Tree.
    TFile *f = (TFile*)gROOT->GetListOfFiles()->FindObject(fileName);
    if (!f || !f->IsOpen()) {
        f = new TFile(fileName);
    }
    fileName.Append("./HEPD");
    TDirectory *dir = (TDirectory*)f->Get(fileName);
    dir->GetObject("EventTree",fTree);
    Init(fTree);
}

Int_t Analyzer::GetEntry(Long64_t entry)
{
    // Read contents of entry.
    if (!fTree) return 0;
    return fTree->GetEntry(entry);
}

Long64_t Analyzer::LoadTree(Long64_t entry)
{
    // Set the environment to read one entry
    if (!fTree) return -5;
    Long64_t centry = fTree->LoadTree(entry);
    if (centry < 0) return centry;
    if (fTree->GetTreeNumber() != fCurrent) {
        fCurrent = fTree->GetTreeNumber();
        Notify();
    }
    return centry;
}

void Analyzer::Init(TTree *tree)
{
    // The Init() function is called when the selector needs to initialize
    // a new tree or chain. Typically here the branch addresses and branch
    // pointers of the tree will be set.
    // It is normally not necessary to make changes to the generated
    // code, but the routine can be extended by the user if needed.
    // Init() will be called many times when running on PROOF
    // (once per file to be processed).

    // Set branch addresses and branch pointers
    Event = 0;
    if (!tree) return;
    fTree = tree;
    fCurrent = -1;

    fTree->SetBranchAddress("Event", &Event, &b_Event);
    Notify();
}

Bool_t Analyzer::Notify()
{
    // The Notify() function is called when a new file is opened. This
    // can be either for a new TTree in a TChain or when when a new TTree
    // is started when using PROOF. It is normally not necessary to make changes
    // to the generated code, but the routine can be extended by the
    // user if needed. The return value is currently not used.

    return kTRUE;
}

void Analyzer::Show(Long64_t entry)
{
    // Print contents of entry.
    // If entry is not specified, print current entry

```

Jan 17, 15 12:37

Analyzer.h

Page 3/3

```
    if (!fTree) return;  
    fTree->Show(entry);  
}  
#endif // #ifdef Analyzer_cxx
```