An introduction to ROOT I/O and trees

- Lecture 8 of MNXB01
  - Inspired by René Brun's 2007 summer student lectures
- Outline
  - ROOT I/O
  - ROOT trees

Introduction to ROOT

Summer Students Lecture

10 July 2007

René Brun

CERN/PH/SFT

http://root.cern.ch

Questions about exercises?

Text vs object oriented I/O

- What is the advantage of OO I/O?

OO I/O

- The class can provide methods to read and write data (dataformat)
  - We can encapsulate also the I/O

Implementation in ROOT

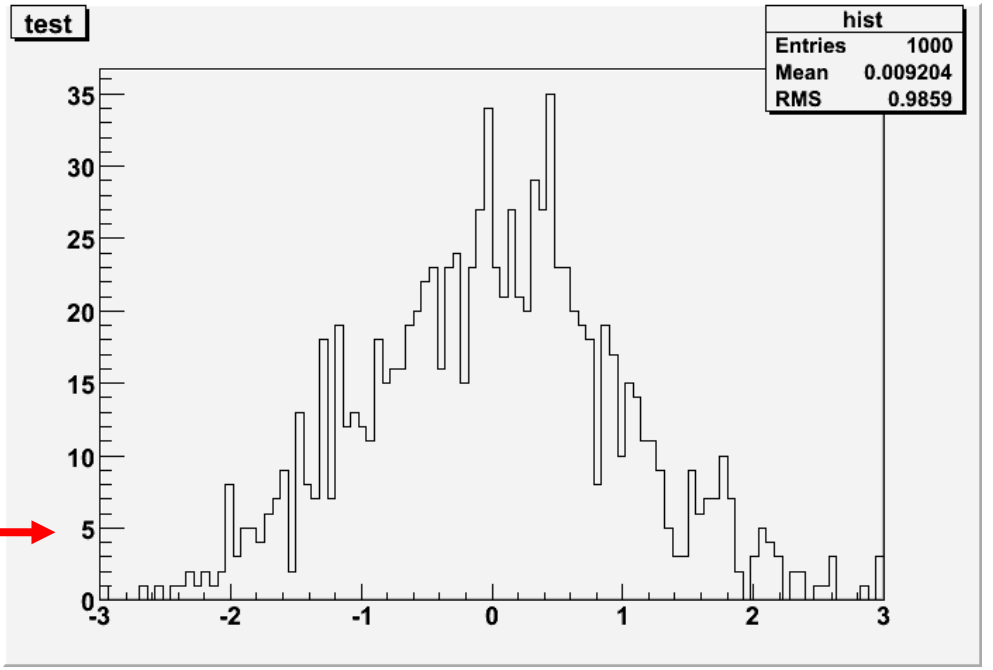
TFile / TDirectory

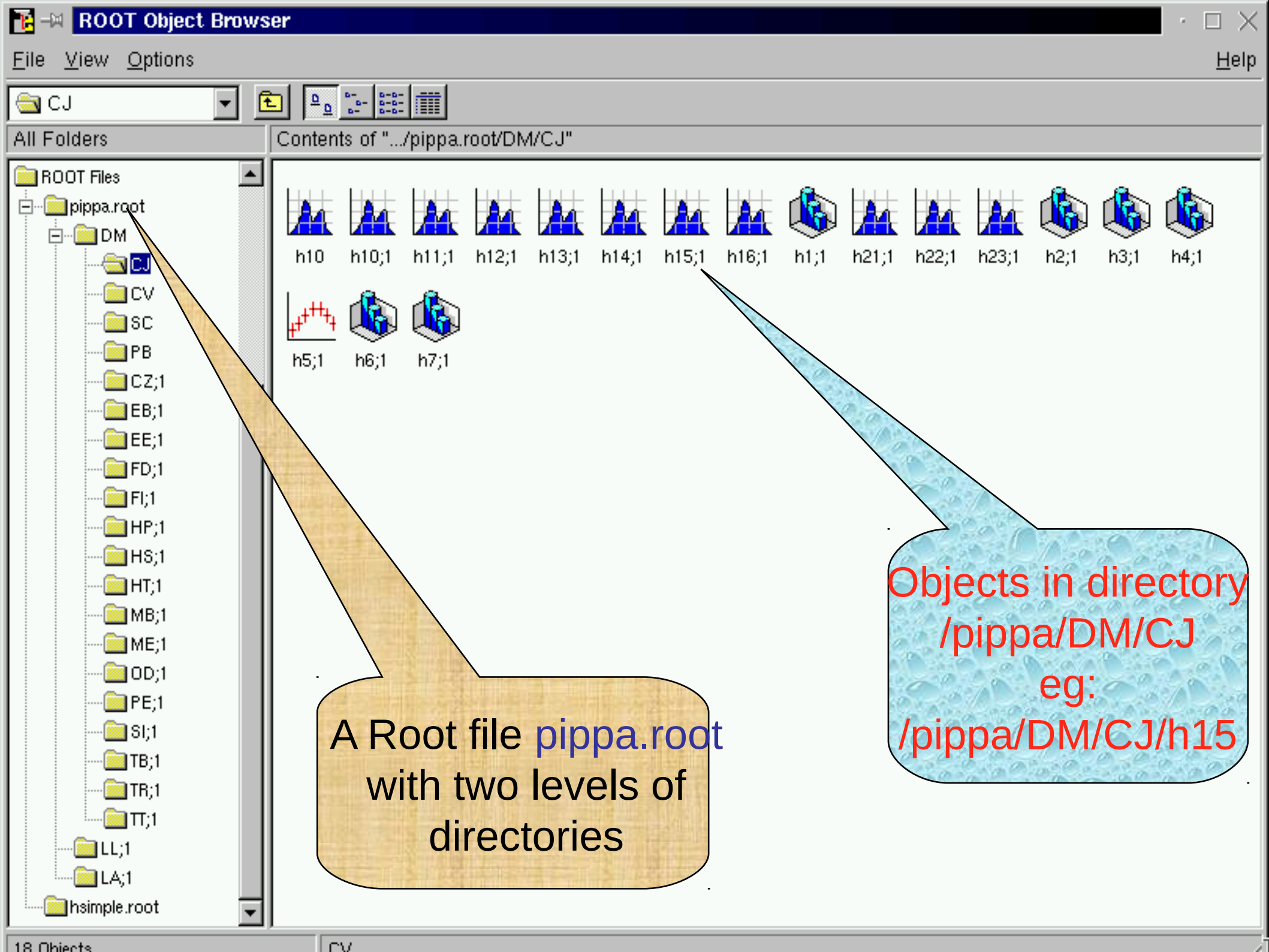
- A TFile object may be divided in a hierarchy of directories, like a Unix file system.
- Two I/O modes are supported
  - Key-mode (TKey). An object is identified by a name (key), like files in a Unix directory. OK to support up to a few thousand objects, like histograms, geometries, mag fields, etc.
  - TTree-mode to store event data, when the number of events may be millions, billions.

Example of key mode

```
void keywrite() {
   TFile f("keymode.root","new");
   TH1F h("hist","test",100,-3,3);
   h.FillRandom("gaus",1000);
   h.Write()
}
```

```
void keyRead() {
   TFile f("keymode.root");
   TH1F *h = (TH1F*)f.Get("hist");;
   h.Draw();
}
```




ROOT Object Browser
File View Options Help
CJ
All Folders
Contents of ".../pippa.root/DM/CJ"
ROOT Files
pippa.root
DM
CJ
CV
SC
PB
CZ;1
EB;1
EE;1
FD;1
FI;1
HP;1
HS;1
HT;1
MB;1
ME;1
OD;1
PE;1
SI;1
TB;1
TR;1
TT;1
LL;1
LA;1
hsimple.root
h10
h10;1
h11;1
h12;1
h13;1
h14;1
h15;1
h16;1
h1;1
h21;1
h22;1
h23;1
h2;1
h3;1
h4;1
h5;1
h6;1
h7;1
A Root file pippa.root with two levels of directories
Objects in directory /pippa/DM/CJ eg: /pippa/DM/CJ/h15
18 Objects
CV

ROOT uses 2 tricks (1/2)

- All objects that should be written derive from the same base class Tobject
  https://root.cern.ch/doc/master/classTObject.html
  - In that way we have a common set of methods
  - Also, we can use a common set of containers

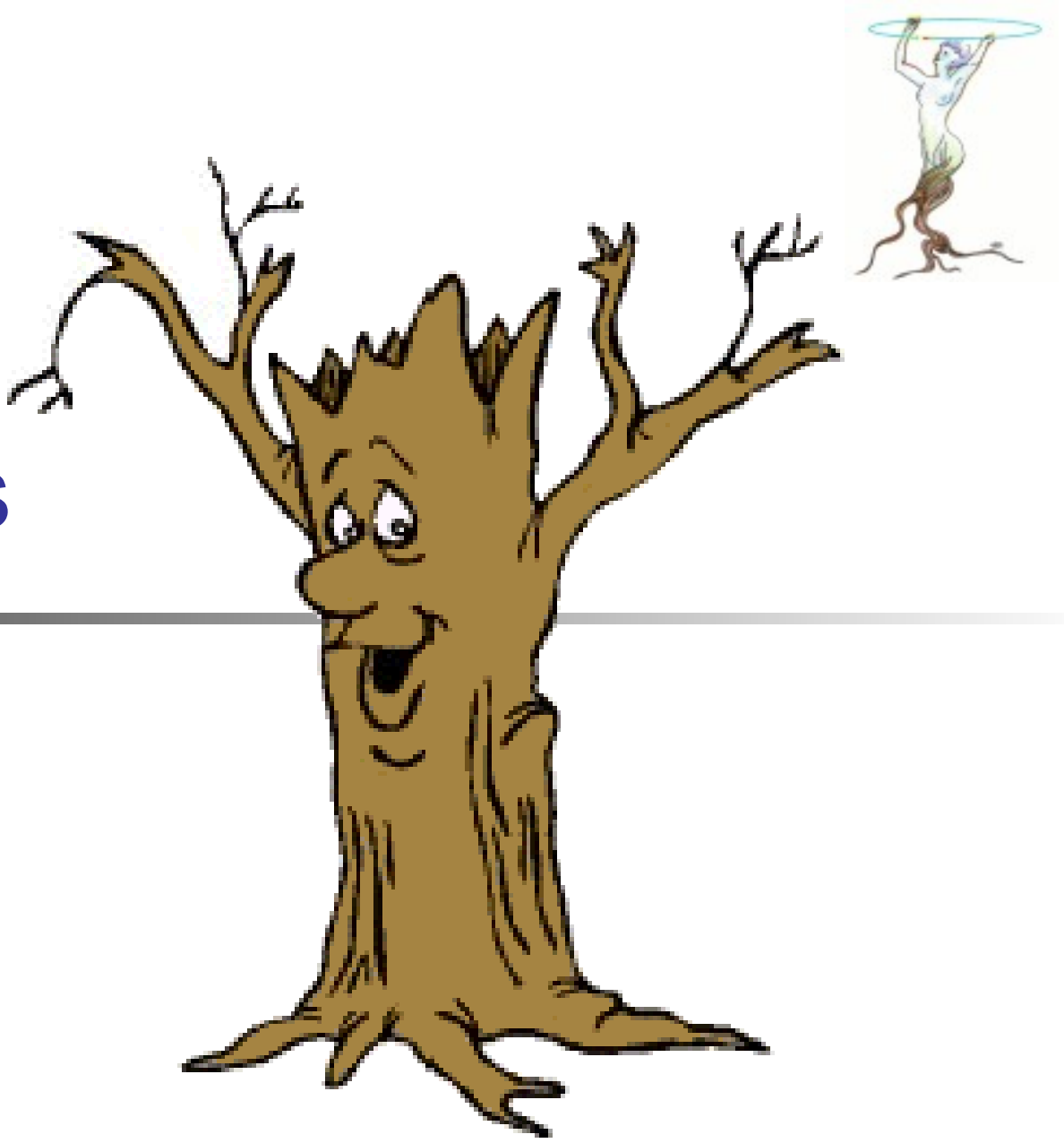
ROOT uses 2 tricks (2/2)

- One has to include some macros that generates the necessary functions/streamers for each object
- In class description:
  ClassDef(MyEvent, 1); //1=version
- In class implementation:
  ClassImp(MyEvent) //no semi-colon!

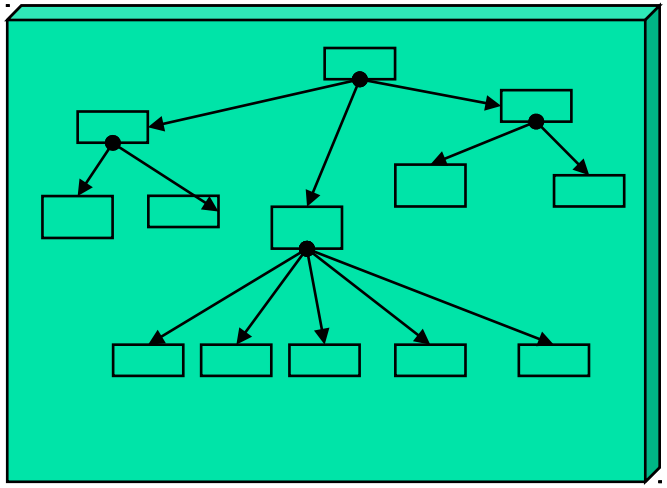
Self-describing files

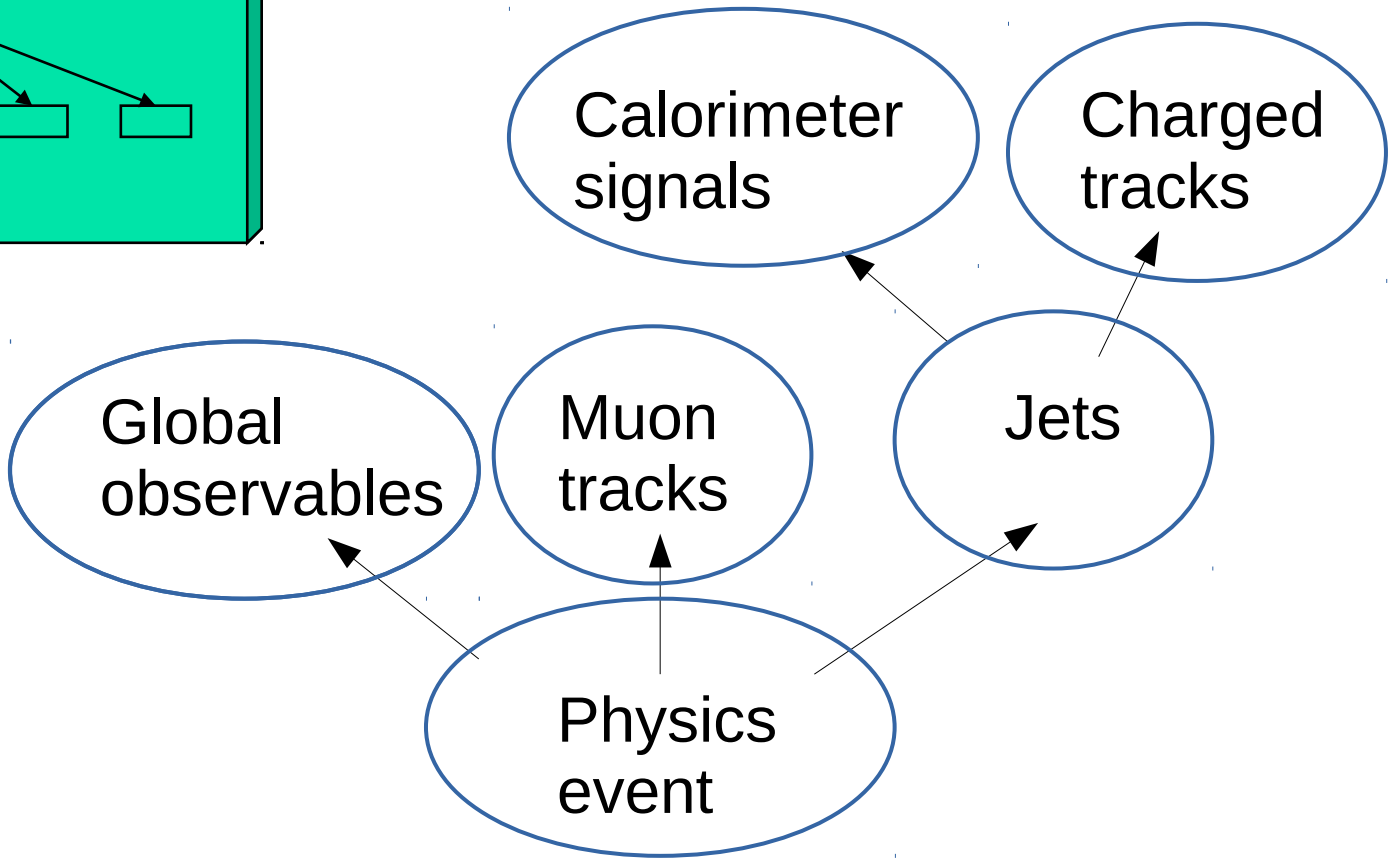
- Dictionary for persistent classes written to the file.
- ROOT files can be read by foreign readers
- Support for Backward and Forward compatibility (one can bump class version)
- Files created in 2001 must be readable in 2015
- Classes (data objects) for all objects in a file can be regenerated via TFile::MakeProject

ROOT Trees



What is a tree?



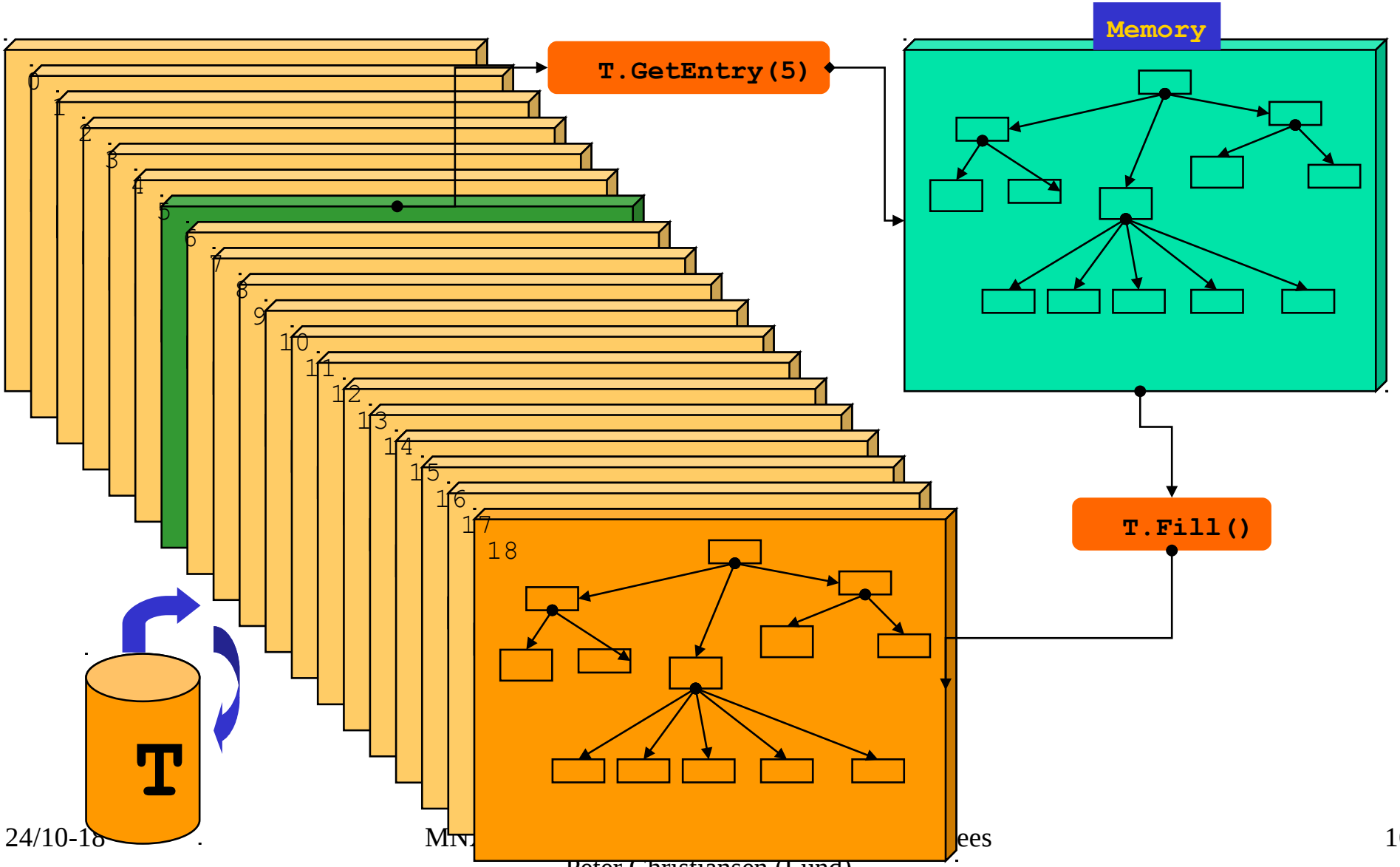


Why Trees ?

- Trees have been designed to support very large collections of objects. The overhead in memory is in general less than 4 bytes per entry.
- Trees allow direct and random access to any entry (sequential access is the best)
- Trees have branches and leaves. One can read a subset of all branches.
- High level functions like TTree::Draw loop on all entries with selection expressions.
- Trees can be browsed via TBrowser
- Trees can be analyzed via TTreeViewer

Memory <--> Tree

Each Node is a branch in the Tree



Writing/Reading a Tree

Event.h

```
class Event : public Something  {
 Header                fHeader;
 std::list<Vertex*>    fVertices;
 std::vector<Track>    fTracks;
 TOF                   fTOF;
 Calor                *fCalor;
}
```

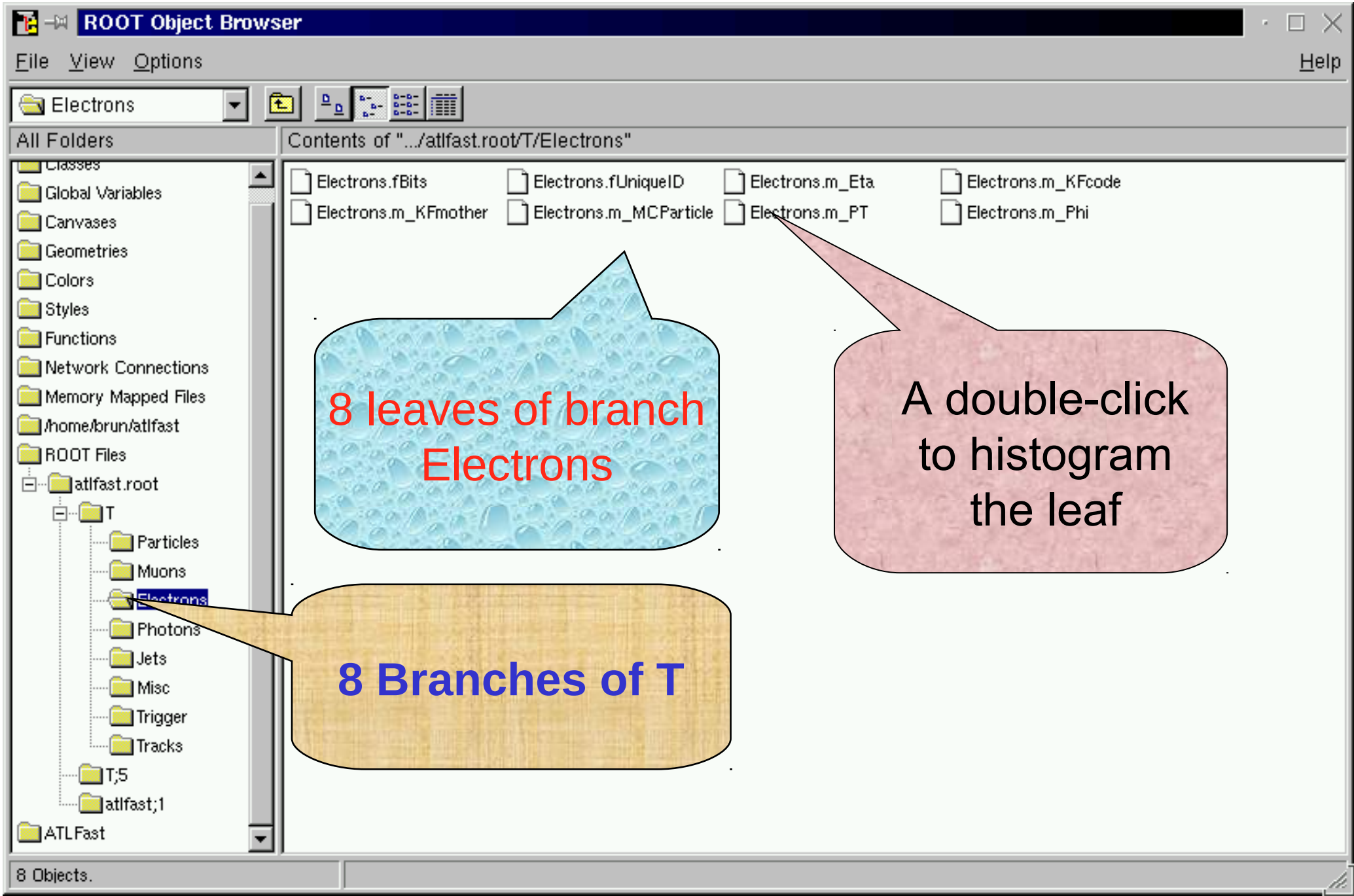
Write.C

```
main() {
   Event *event = 0;
   TFile f("demo.root","recreate");
   int split = 99;  //maximum split
   TTree *T = new TTree("T","demo Tree");
   T->Branch("event",&event,split);
   for (int ev=0;ev<1000;ev++) {
      event = new Event(…);
      T->Fill();
      delete event;
   }
   T->AutoSave();
}
```

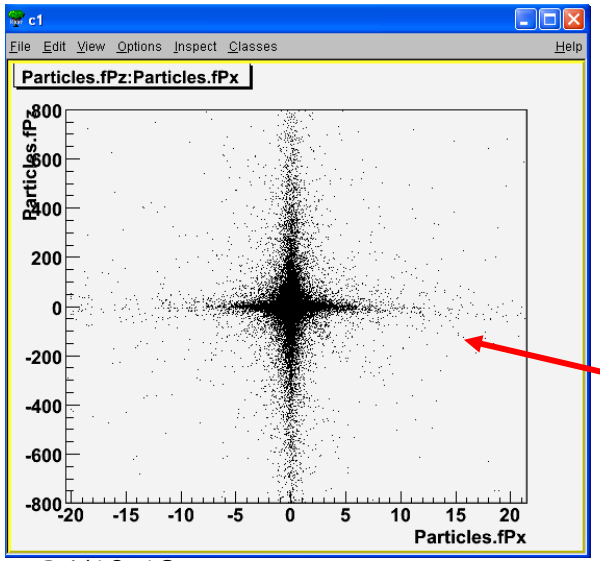
Read.C

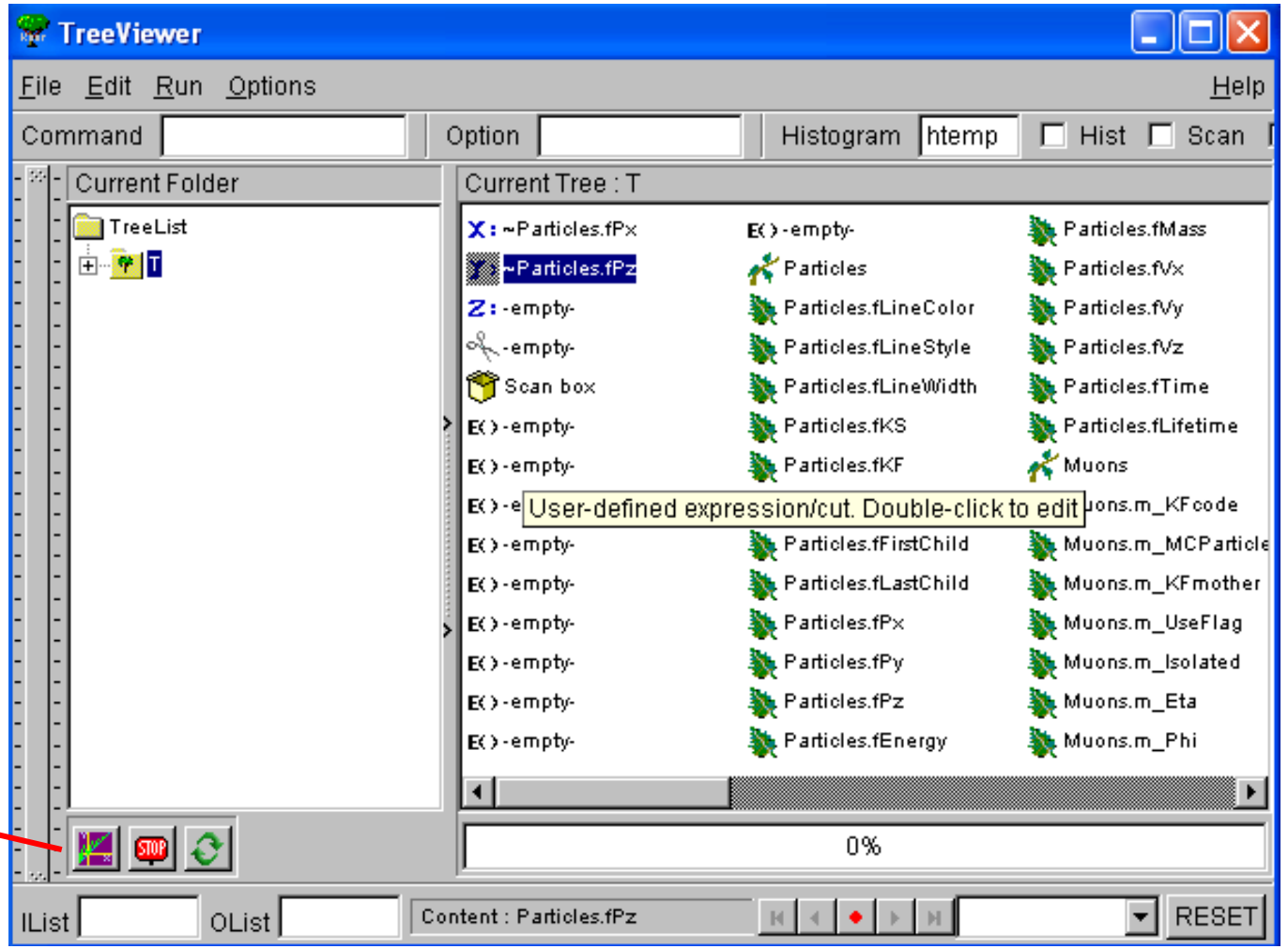
```
main() {
   Event *event = 0;
   TFile f("demo.root");
   TTree *T = (TTree*)f.Get"T");
   T->SetBranchAddress("event",&event);
   Long64_t N = T->GetEntries();
   for (Long64_t ev=0;ev<N;ev++) {
      T->GetEntry(ev);
      // do something with event
   }
}
```

Browsing a TTree with TBrowser



The TTreeViewer





Example

Go through last exercises