

Introduction to ROOT

March 2021 J. Smallcombe

What is ROOT?

“ROOT is an object-oriented framework aimed at solving the data analysis challenges of high-energy physics.”

What is ROOT? (Box of tricks)

C++ based library & program

Flexible maths and physics tools all in one place :

Histograms & graphing tools

TTrees (object based data files)

Fit minimisers

Maths & statistics functions

GUIs

A Library or a Program

Use ROOT classes in compiled programs

Individual Object Classes

Entire ROOT system

Convenient working with data saved in ROOT format
or to export data to ROOT format

Use ROOT as a program

Powerful command line interpreter

Work live (on the fly) or with scripts

Free combination of scripts and GUIs

Easy “Compile” scripts within root for speed

Sources of help examples:

Detailed Class Documentation

<https://root.cern.ch/doc/master/classTH1.html>

Introductory Guides and Tutorials

<https://root.cern.ch/howtos>

Well Commented Example Codes

https://root.cern/doc/master/group__Tutorials.html

Active Forum Moderated by ROOT Lead programmers

<https://root-forum.cern.ch/>

Running ROOT (From the command line)

root

Start a new ROOT iterative session

root file.root

Start a new ROOT iterative session and open file.root if it exists.

root rootscript.C

Start a new ROOT iterative session and execute ROOT command list in rootscript.C

root codename.cpp

root codename.h

Start a new ROOT iterative session, load and interpret (parse) code, execute function **codename()** if it exists.

Basic Commands Within ROOT

.q

Quit current ROOT iterative session.

.x rootscript.C

Execute ROOT commands in rootscript.C

.L codename.cpp/h

Load and interpret code.

.x codename.cpp/h

Load and interpret code **AND** execute function **codename()** if it exists.

Basic Commands Within ROOT

.q

Quit

.x

Execute

“script” - A list of commands to executed immediately (as if they had been entered manually into the ROOT interactive session)

.L

Load

“code” (macro) - Compilable C++ code consisting of functions & classes, which may be used in ROOT (after being loaded and parsed by the interpreter)

.x

Load

codeName (macro) - Load and execute

Basic Commands Within ROOT

double VariableName=1.2

Create a local instance of a standard **C** variable type double (decimal number)

TExample VariableName(constructor,inputs)

Create a local instance of an object of the class/type TExample.

VariableName.SomeFunction();

Call a TExample class function on a local instance **VariableName**.

TExample *PointerName = new TExample(inputs)

Create a instance of a TExample object on the heap memory **and** save the address of the object to a new pointer **PointerName**

PointerName->OtherFunction()

Call a TExample class function of a instance pointed to by **PointerName**

Demo 0

Some ROOT Classes

TH1 - 1-dimensional histogram

TH2 - 2-dimensional histogram

TF1 - Holder of mathematical functions used for fitting, drawing etc

TCanvas - A graphical environment

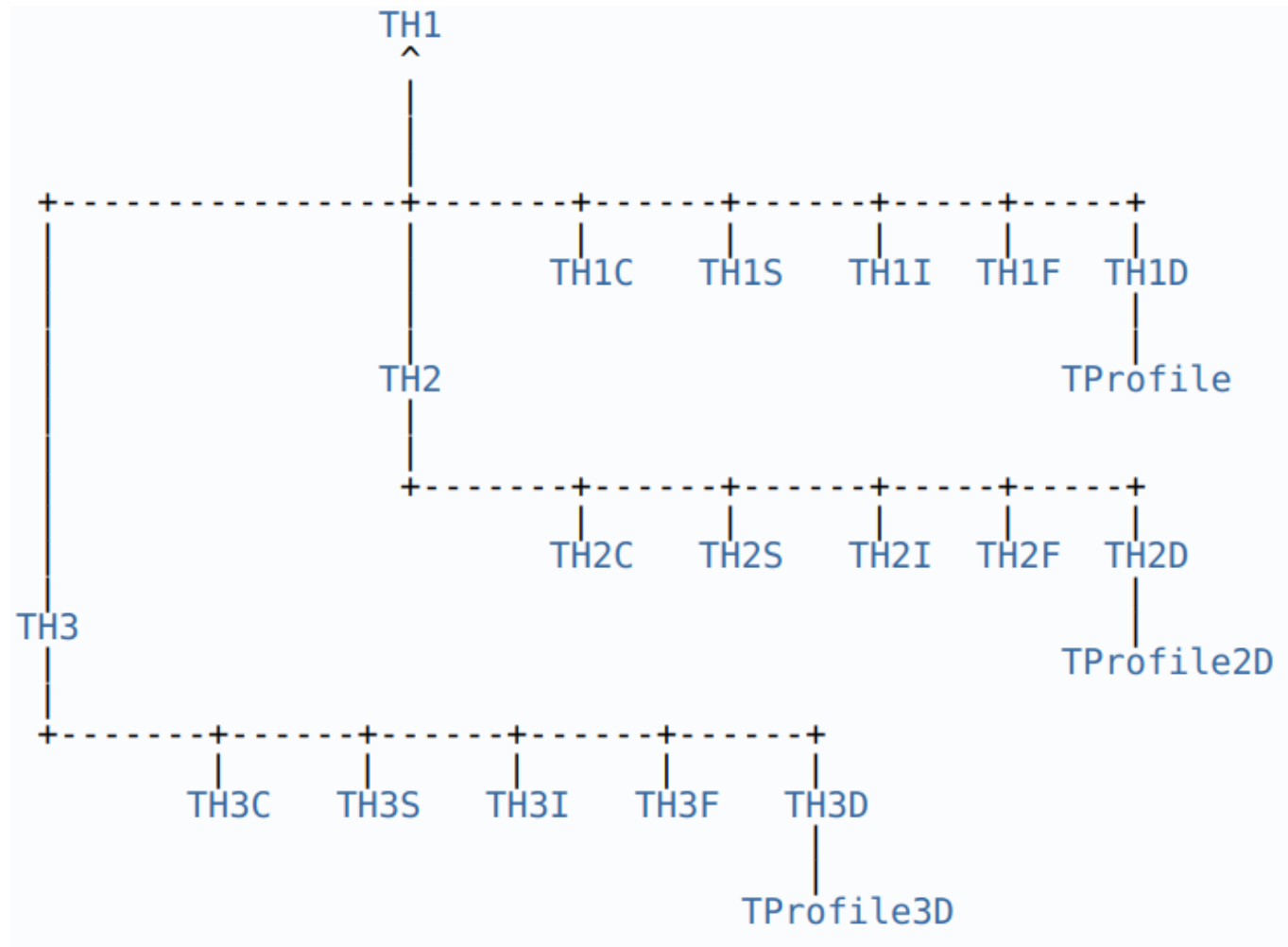
TBrowser - A file browser (with graphical window)

TFile - Use to connect to .root data files

TTree - Flexible (object based) data list

Histogram Classes TH1/TH2/TH3

Data of a TH1 type objects primarily consist of an array of bin content values.



Class contains many useful functions for the direct probing and manipulation of the data e.g. Addition, Statistics, Rebinning.

Histogram Classes TH1/TH2/TH3

Data of a TH1 type objects primarily consist of an array of bin content values.

TH1C : histograms with one byte per channel. Maximum bin content = 127

TH1S : histograms with one short per channel. Maximum bin content = 32767

TH1I : histograms with one int per channel. Maximum bin content = 2147483647

TH1F : histograms with one float per channel. Maximum precision 7 digits

TH1D : histograms with one double per channel. Maximum precision 14 digits

TH1D 10,000 bins, 8 byte each 0.08 MB

TH3C 1000x1000x1000 bins 1 byte each 1 GB

Class contains many useful functions for the direct probing and manipulation of the data e.g. Addition, Statistics, Rebinning.

Histogram Classes TH1/TH2/TH3

Typical constructor:

TH1D("name", "title", Nb , Min , Max)

Name - ROOT special name

Title - Display Only
(optional "title;x-title;y-title")

Nb - Number of bins (integer)

Min - Lowest edge (double)

Max - Highest edge (double)



" " - Use quotes around string type input,
ROOT typically uses *const char** not c++ *strings*

ROOT “name”

TH1D(“name”, “title”, Nb , Min , Max)

Many classes inherit from “TNamed”

Object names stored in “*internal ROOT memory management system*”.

Object instance accessible by name indirectly.

Names as pointer in interpreter mode.

ROOT “name”

The ROOT name of an object is a plain text reference to the object.

This is NOT the same thing as the chosen name for a variable in code.

```
TH1D hist1(“MyHist”, “My Title”,100,0,100);
```

```
hist1 = A TH1D type object
```

```
MyHist = A plain text reference to the object hist1
```


ROOT “name”

The ROOT name of an object is a plain text reference to the object.

This is NOT the same thing as the chosen name for a variable in code.

However,
The same characters ***may*** be entered for both
(And frequently are)

```
TH1D hist1(“hist1”, “My Title”,100,0,100);
```

Histogram Classes TH1/TH2/TH3

Typical constructor:

TH1D("name", "title", Nb , Min , Max)

Name - ROOT special name

Title - Display Only
(optional "title;x-title;y-title")

Nb - Number of bins (integer)

Min - Lowest edge (double)

Max - Highest edge (double)



" " - Use quotes around string type input,
ROOT typically uses *const char** not c++ *strings*

Summary

- ROOT is mostly a big library of objects but with a powerful interpreter (and some cooky memory management)
- Can code directly into interpreter, run simple scripts or load complete c++ codes
- ROOT Object “name” variables act as special pointers within the ROOT system
- TH1 useful histogram objects