

ROOT

File Management

- Creating a new root file:

```
TFile *f = new TFile("demo.root", "recreate")
```

- Opening an existing file:

```
TFile f("demo.root")
```

- Closing a file:

```
f->Close()
```

OR

```
f.Close()
```

- Accessing contents in a file:

```
f.ls()
```

- Current path:

```
gDirectory->pwd()
```

- Change Directory:

```
f.cd()
```

- Making Directory and going in it:

```
f.mkdir("my-dir")
```

```
f.cd("my-dir")
```

- GUI view of directory:

```
new TBrowser
```

- Deleting Directory

```
f.delete("my-dir")
```


Creating TTree

- a container to handle large data in an organised way

- Creating a TTree:

```
TTree t("t", "my tree")
```

```
t.Print()
```

- Linking variable to TTree:

```
Int_t money-A, money-B;
```

```
t.Branch("mA", &money-A, "money-A/I")
```

```
money-A=10;
```

```
t.Fill();
```

```
t.Scan("money-A")
```

- Linking an array to TTree

```
Int_t arr[3] = {0};
```

```
t.Branch("A", arr, "myarr[3]/I")
```

```
arr[0]=11; arr[1]=12; arr[2]=13
```

```
t.Fill()
```

- Filling TTree with variable size array

```
Int_t n_max = 100;
```

```
Int_t energy[n_max] = {0};
```

```
Int_t n;
```

```
t.Branch("Size", &n, "n/I")
```

t.Branch("energy", energy, "Penergy [uJ/E^2]");

n=1;

energy[0] = 21;

t.Fill();

Linking struct and TTree

```
struct particle_info {  
    float energy;  
    float position[3];  
};
```

3;

particle_info p1;

t.Branch("particle detail", &p1, "e/F: pos[3]/F^2");

p1.energy = 30.36;

p1.position[0] = 231.7; p1.position[1] = 93.5;

p1.position[2] = 77; t.Fill();

t.Scan("1*");

Histogram

• Drawing a histogram:

h0("h0", "histo0", 100, -10, 10)

~~h0.Draw()~~
h0.FillRandom("gaus", 100)

h0.Draw() OR h0 → Draw()

• Writing it into file f:

h0.Write()

<u>Class Name</u>	<u>Functionality</u>
TObject	Provides default behaviour for all objects in Root system eg) object I/O, printing etc.
TSystem	Class defining interface to the OS.
DataLoader	To load datasets
TChain	Collection of files containing TTree objects
Factory	It contains implemented MVA methods