Offline Analysis DataBase

Andreas Morsch
ALICE Offline Week
9/03/2011

Purpose

- The Offline Analysis Data Base (*OADB*) will contain data needed for the end user analysis of ESDs and AODs. Examples of such data are
  - Acceptance and efficiency correction maps
  - PID response functions and resolutions.
  - Centrality Histograms and Scaling Factors (PbPb)
  - Alignment Matrices (PHOS, EMCAL)
- The *OADB* should be available for local analysis and on the GRID.
- The data base should provide versioning and validity periods.

Requirements

- The *OADB* should be available for local analysis and on the GRID.
- The data base should provide versioning and validity periods.

Implementation (1)

- The *OADB* will implemented as ROOT-files residing in an *svn* repository as part of *AliRoot* providing versioning per se.
- The root directory structure of the repository is given by the different physics working groups:
  - OADB/PWG1/
  - OADB/PWG2/
  - OADB/COMMON/Centrality/data
  - ...
- Subdirectory names hint to the purpose of the data stored:
  - OADB/PWG1/PID/TOF

Implementation (2)

- Depending on the *OADB* update frequency, the deployment can be synchronized or not with the analysis tags. The requested *OADB* will be unpacked in the job scratch space and made available as a local directory.
- The data objects themselves are stored in a file containing one or several objects of type *AliOADBContainer*

Updating and Validity Periods

- Most of the data contained in the *OADB* will span several runs or even LHC running periods.
- Each category should have one person responsible for keeping the file up to data. This comprises adding new objects and redefining the validity ranges.
- Post *svn commit action* will write an object with the revision number to the root file.
  - Under development

AliOADBContainer

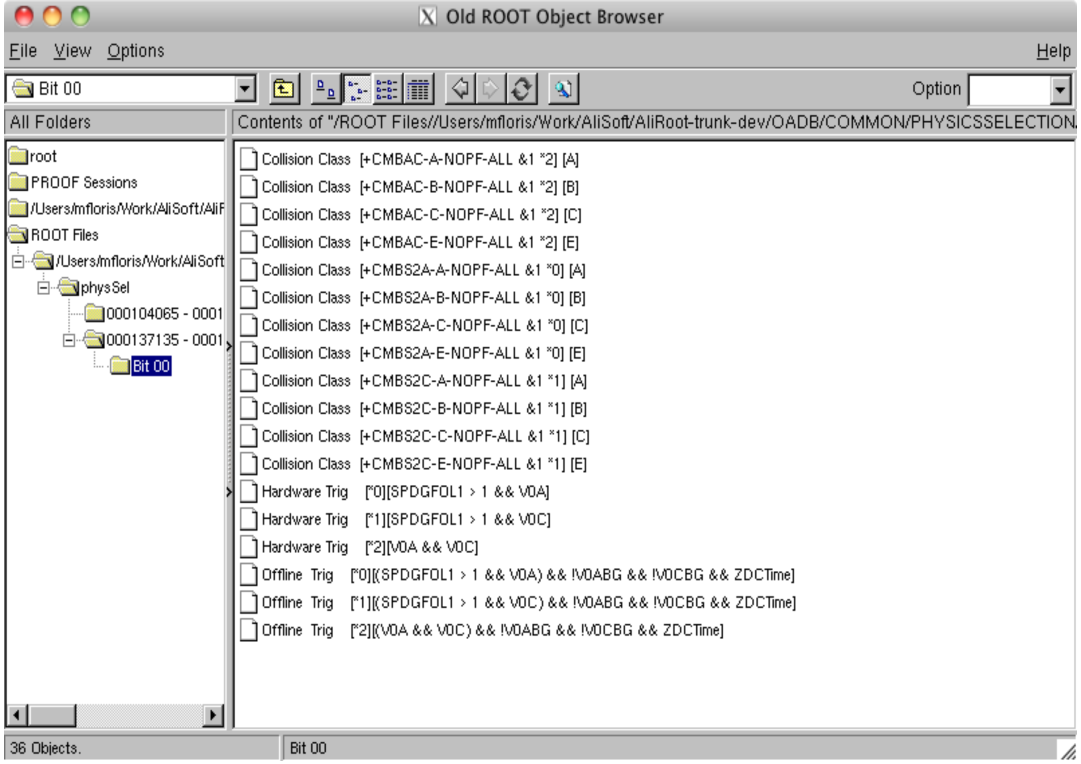
- Composite container class
  - TObjArray to hold the Data needed for analysis
    - TArrayI for the run ranges
    - Same index for range and data
  - TList for the default Objects
    - TObjects can be found via their name
- Services
  - Appending, updating deleting, … of objects
  - Consistency checks
  - Debug information

```
AliOADBContainer();
  AliOADBContainer(char* name);
  virtual ~AliOADBContainer();
  AliOADBContainer(const AliOADBContainer& cont);
  AliOADBContainer& operator=(const
AliOADBContainer& cont);
// Object adding and removal
  void   AppendObject(TObject* obj, Int_t lower,
Int_t upper);
  void   UpdateObject(Int_t index, TObject* obj,
Int_t lower, Int_t upper);
  void   RemoveObject(Int_t index);
  void   AddDefaultObject(TNamed* obj);
  void   CleanDefaultList();
  TList* GetDefaultList() const {return
fDefaultList;}
```

```
// I/O
  void  WriteToFile(char* fname)  const;
  Int_t InitFromFile(char* fname, char* key);
// Getters
  Int_t GetNumberOfEntries()    const {return
fEntries;}
  Int_t LowerLimit(Int_t idx)   const {return
fLowerLimits[idx];}
  Int_t UpperLimit(Int_t idx)   const {return
fUpperLimits[idx];}
  TObject* GetObject(Int_t run, char* def = "") const;
  TObject* GetObjectByIndex(Int_t run) const;
  TObject* GetDefaultObject(char* key)
  {return(fDefaultList->FindObject(key));}
// Debugging
  void List();
// Browsable
  virtual Bool_t IsFolder() const { return kTRUE; }
  void Browse(TBrowser *b);
```

Physics selection OADB

- Several parameters hardcoded in the physics selection
- Solution: Move configuration to the OADB
  - 2 Classes, Browsable:
    - *AliPhysicsSelectionOADB*
      - Trigger Class AliVEvent::EOfflineTriggerTypes
      - Online/Offline trigger condition (strings)
    - *AliFillingSchemeOADB*
      - BX ids per beam side
  - Trigger conditions parsed using tokens defined in *AliTriggerAnalysis*
  - MC: same conditions, skips trigger class names

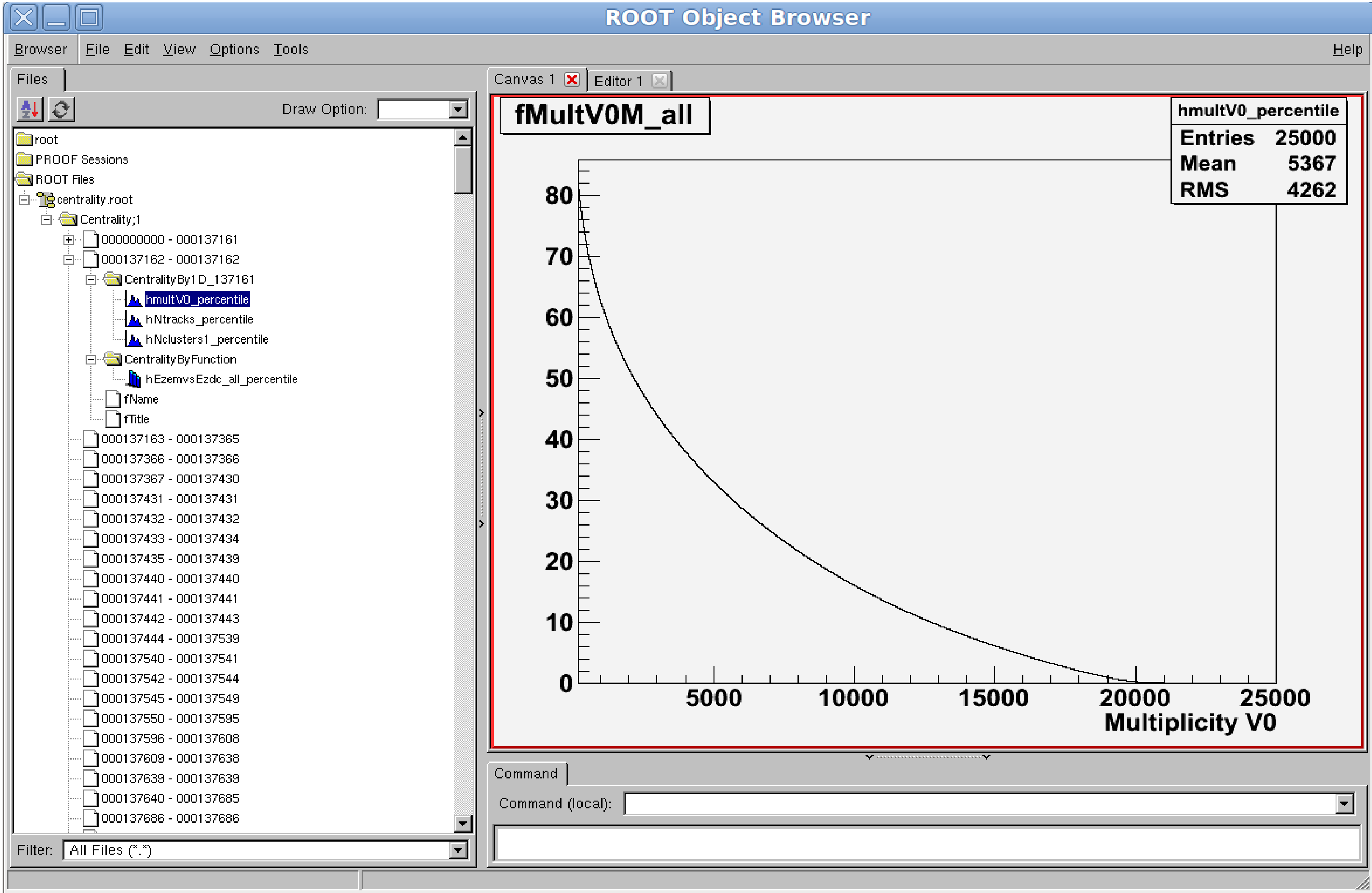


```
Contents of "/ROOT Files//Users/mfloris/Work/AliSoft/AliRoot-trunk-dev/OADB/COMMON/PHYSICSSELECTION
Collision Class  [+CMBAC-A-NOPF-ALL &1 *2] [A]
Collision Class  [+CMBAC-B-NOPF-ALL &1 *2] [B]
Collision Class  [+CMBAC-C-NOPF-ALL &1 *2] [C]
Collision Class  [+CMBAC-E-NOPF-ALL &1 *2] [E]
Collision Class  [+CMBS2A-A-NOPF-ALL &1 *0] [A]
Collision Class  [+CMBS2A-B-NOPF-ALL &1 *0] [B]
Collision Class  [+CMBS2A-C-NOPF-ALL &1 *0] [C]
Collision Class  [+CMBS2A-E-NOPF-ALL &1 *0] [E]
Collision Class  [+CMBS2C-A-NOPF-ALL &1 *1] [A]
Collision Class  [+CMBS2C-B-NOPF-ALL &1 *1] [B]
Collision Class  [+CMBS2C-C-NOPF-ALL &1 *1] [C]
Collision Class  [+CMBS2C-E-NOPF-ALL &1 *1] [E]
Hardware Trig   [*0][SPDGFOL1 > 1 && V0A]
Hardware Trig   [*1][SPDGFOL1 > 1 && V0C]
Hardware Trig   [*2][V0A && V0C]
Offline  Trig   [*0][(SPDGFOL1 > 1 && V0A) && !V0ABG && !V0CBG && ZDCTime]
Offline  Trig   [*1][(SPDGFOL1 > 1 && V0C) && !V0ABG && !V0CBG && ZDCTime]
Offline  Trig   [*2][(V0A && V0C) && !V0ABG && !V0CBG && ZDCTime]
```

- Status:
  - Implementation almost finished
  - Validation/debugging on CAF in progress
  - TODO: Fill OADB, tests over the full period

Michele Floris, JFGO

Centrality



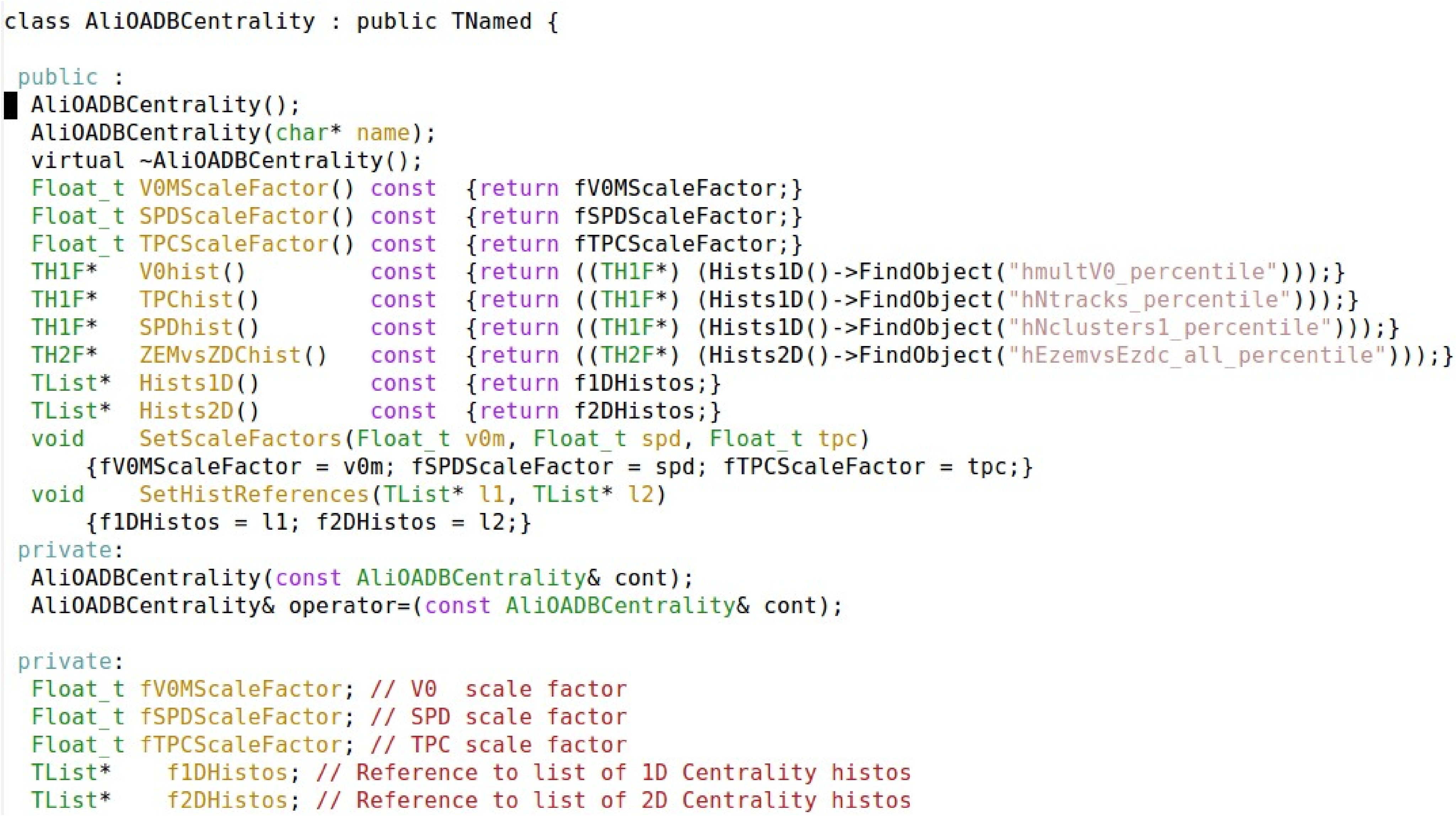
```
root
PROOF Sessions
ROOT Files
centrality.root
  Centrality;1
    000000000 - 000137161
    000137162 - 000137162
      CentralityBy1D_137161
        hmultV0_percentile
        hNtracks_percentile
        hNclusters1_percentile
      CentralityByFunction
        hEzemvsEzdc_all_percentile
      fName
      fTitle
    000137163 - 000137365
    000137366 - 000137366
    000137367 - 000137430
    000137431 - 000137431
    000137432 - 000137432
    000137433 - 000137434
    000137435 - 000137439
    000137440 - 000137440
    000137441 - 000137441
    000137442 - 000137443
    000137444 - 000137539
    000137540 - 000137541
    000137542 - 000137544
    000137545 - 000137549
    000137550 - 000137595
    000137596 - 000137608
    000137609 - 000137638
    000137639 - 000137639
    000137640 - 000137685
    000137686 - 000137686
```

hmultV0_percentile	
Entries	25000
Mean	5367
RMS	4262

Command (local):

Filter: All Files (*.*)

AliOADBCentrality

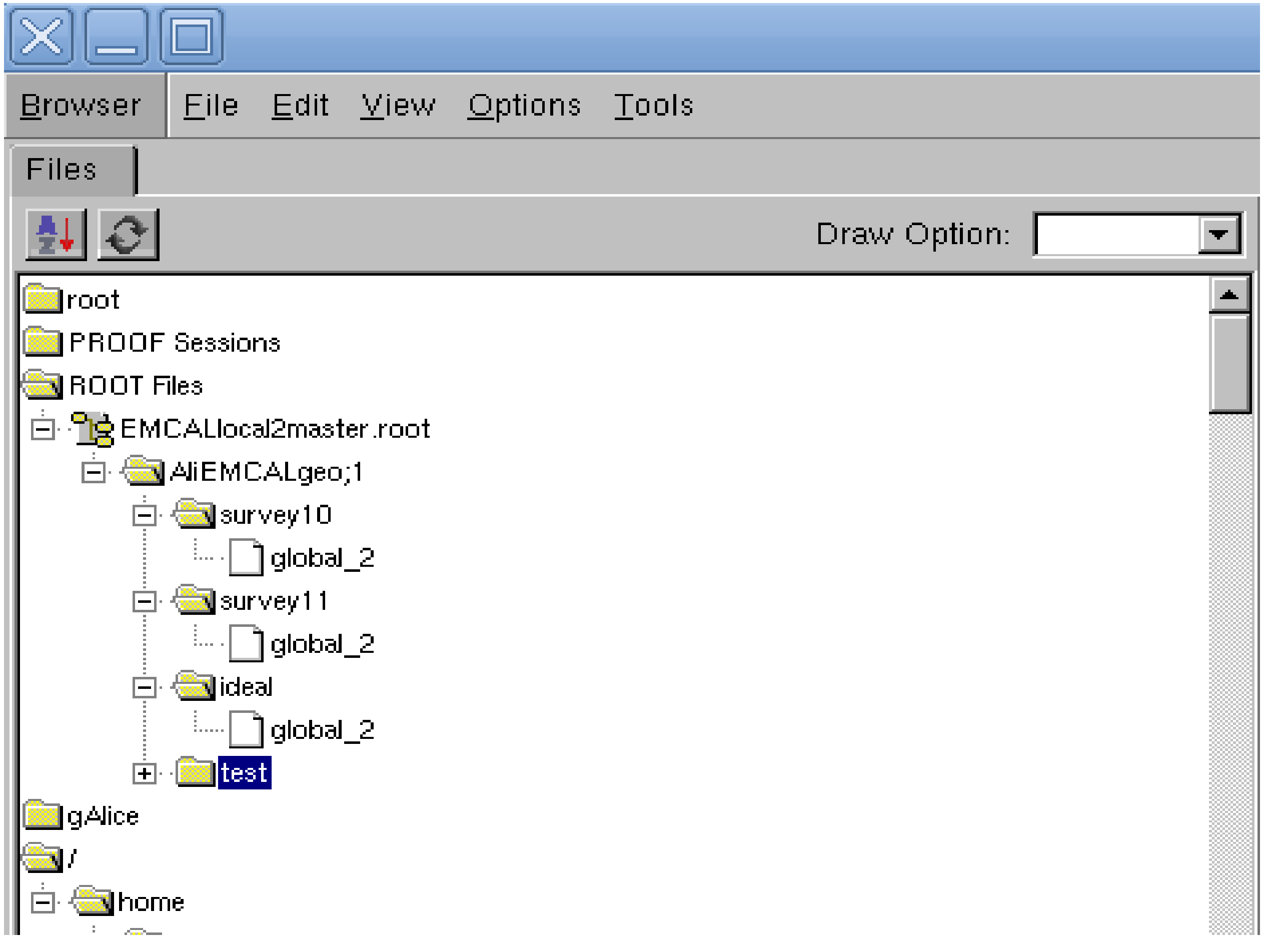


```
class AliOADBCentrality : public TNamed {

 public :
  AliOADBCentrality();
  AliOADBCentrality(char* name);
  virtual ~AliOADBCentrality();
  Float_t V0MScaleFactor() const  {return fV0MScaleFactor;}
  Float_t SPDScaleFactor() const  {return fSPDScaleFactor;}
  Float_t TPCScaleFactor() const  {return fTPCScaleFactor;}
  TH1F*   V0hist()         const  {return ((TH1F*) (Hists1D()->FindObject("hmultV0_percentile")));}
  TH1F*   TPChist()        const  {return ((TH1F*) (Hists1D()->FindObject("hNtracks_percentile")));}
  TH1F*   SPDhist()        const  {return ((TH1F*) (Hists1D()->FindObject("hNclusters1_percentile")));}
  TH2F*   ZEMvsZDChist()   const  {return ((TH2F*) (Hists2D()->FindObject("hEzemvsEzdc_all_percentile")));}
  TList*  Hists1D()        const  {return f1DHistos;}
  TList*  Hists2D()        const  {return f2DHistos;}
  void    SetScaleFactors(Float_t v0m, Float_t spd, Float_t tpc)
      {fV0MScaleFactor = v0m; fSPDScaleFactor = spd; fTPCScaleFactor = tpc;}
  void    SetHistReferences(TList* l1, TList* l2)
      {f1DHistos = l1; f2DHistos = l2;}
 private:
  AliOADBCentrality(const AliOADBCentrality& cont);
  AliOADBCentrality& operator=(const AliOADBCentrality& cont);

 private:
  Float_t fV0MScaleFactor; // V0  scale factor
  Float_t fSPDScaleFactor; // SPD scale factor
  Float_t fTPCScaleFactor; // TPC scale factor
  TList*    f1DHistos; // Reference to list of 1D Centrality histos
  TList*    f2DHistos; // Reference to list of 2D Centrality histos
```

AliEMCALgeo



```
root
PROOF Sessions
ROOT Files
  EMCALlocal2master.root
    AliEMCALgeo;1
      survey10
        global_2
      survey11
        global_2
      ideal
        global_2
      test
gAlice
/
  home
```

To Do

- AliOADBManager
  - Single access point
  - I/O
  - Caching
  - Clean-up