

The Event Reconstruction Chain

Code Structure

Campaign manager

Foot Global Parameters

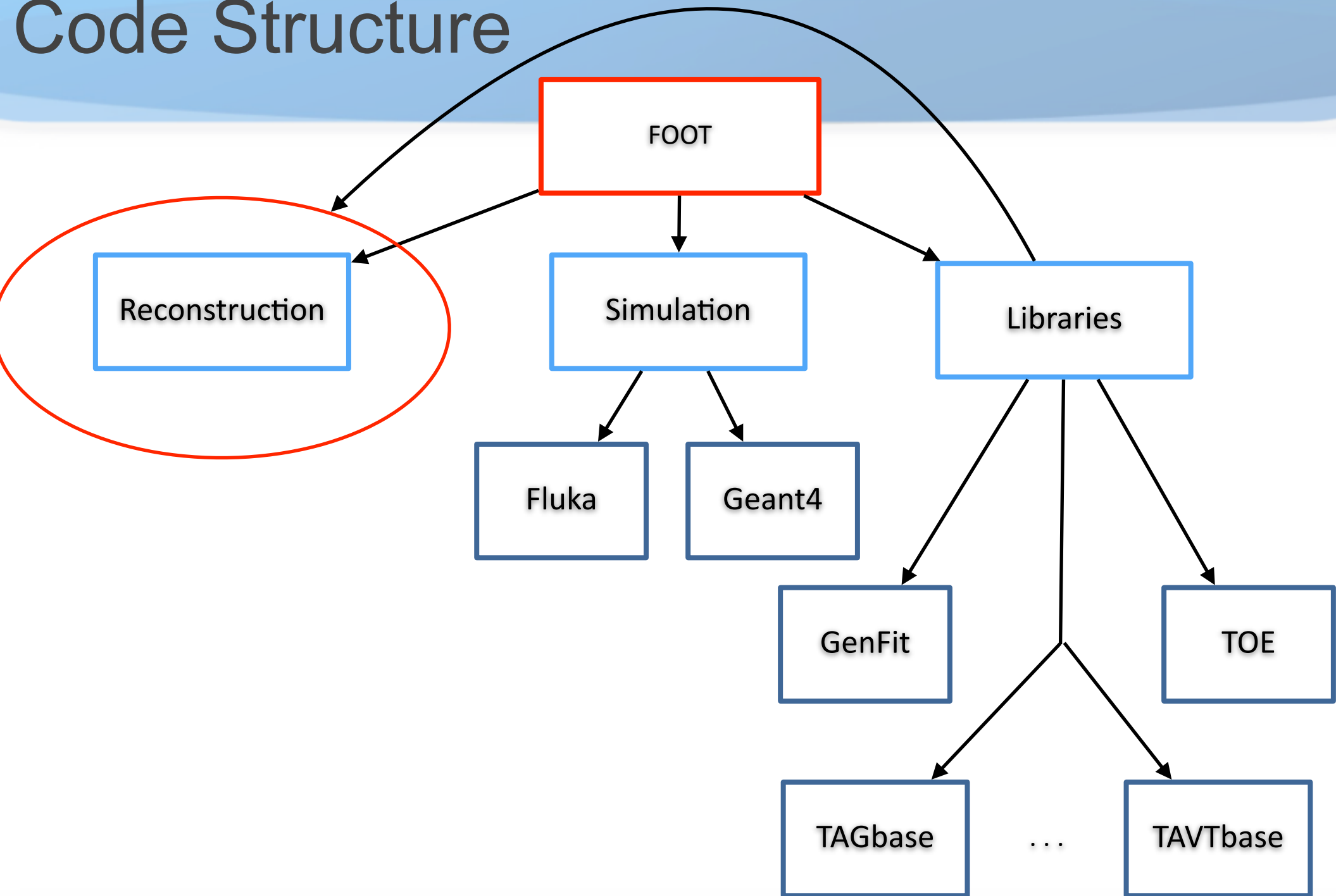
Run manager

Executables

Event Display

Analysis framework

Code Structure



Campaign Manager

Campaign manager (i)

□ Master file: ./cammaps/FOOT.map

```
// List of Campaigns
```

```
CamNumber: 0  
CamName: "STD"  
CamDataMC: 1  
CamDate: ""  
CamSum: "Test using standard input files"
```

```
CamNumber: 1  
CamName: "12C_200"  
CamDataMC: 1  
CamDate: ""  
CamSum: "Simulation data for 12C on 12C (5mm target) @ 200MeV/u"
```

```
CamNumber: 2  
CamName: "160_200"  
CamDataMC: 1  
CamDate: ""  
CamSum: "Simulation data for 160 on 12C (run 6) or C2H4 (run 1-5) for 5mm target thickness @ 200MeV/u"
```

```
CamNumber: 3  
CamName: "GSI"  
CamDataMC: 0  
CamDate: "5-8 April 2019"  
CamSum: "Test experiment with ST+BM+VTX+TW..."
```

```
CamNumber: 4  
CamName: "HE_MC"  
CamDataMC: 1  
CamDate: "5-8 January 2020"  
CamSum: "MC data for HE reaction produced by FLUKA"
```

```
CamNumber: 5  
CamName: "H_MC"  
CamDataMC: 1  
CamDate: ""  
CamSum: "Simulation data for 1H on C2H4 for 5mm target thickness @ 200MeV/u (run 1) and @ 700MeV/u (run 2)"
```

- Campaign name with associated id
- MC data flag
- Date of data taking or production
- Summary of the campaign

➔ e.g.: load campaign file in ./cammaps/GSI.cam when option: `-exp GSI`

Campaign manager (ii)

- Real data structure: ./cammaps/GSI.cam

```
// Campaign file
CamName: "GSI"
RunNumber: 2187-2261
NumberDevices: 6

DetectorName: "FOOT"
NumberFiles: 1
"./geomaps/GSI/FOOT.geo": 2210; 2239

DetectorName: "ST"
NumberFiles: 4
"./geomaps/GSI/TASTdetector.geo": -1
"./config/GSI/TASTdetector.cfg": -1
"./config/GSI/WDChannelMap.map": -1
"./calib/GSI/WDTimeCalibration/tcalib.dat": 2187;
2190; 2191; 2192; 2193; 2200; 2202; 2203; 2204;
2205; 2206; 2208; 2209; 2210; 2211; 2212; 2213;
2214; 2216; 2217; 2218; 2219; 2220; 2221; 2222;
2223; 2224; 2225; 2226; 2227; 2228; 2229; 2230;
2231; 2232; 2233; 2234; 2235; 2236; 2237; 2238;
2239; 2240; 2241; 2242; 2243; 2244; 2245; 2246;
2247; 2248; 2249; 2250; 2251; 2252; 2253; 2254;
2255; 2256; 2257; 2258; 2259; 2260; 2261; 2262;
2263; 2264; 2270; 2271; 2272;

DetectorName: "BM"
NumberFiles: 4
"./geomaps/GSI/TABMdetector.geo": -1
"./config/GSI/TABMdetector.cfg": -1
"./config/GSI/TABMdetector.map": -1
"./calib/GSI/TABM_T0_Calibration.cal": 2210; 2239

DetectorName: "TG"
NumberFiles: 1
"./geomaps/GSI/TAGdetector.geo": 2187; 2210
```

```
DetectorName: "VT"
NumberFiles: 3
"./geomaps/GSI/TAVTdetector.map": -1
"./config/GSI/TAVTdetector.geo": -1
"./config/GSI/TAVTdetector.map": -1

DetectorName: "TW"
NumberFiles: 7
"./geomaps/GSI/TATWdetector.geo": -1
"./config/GSI/TATW_BBparameters.cfg": 2187; 2210
"./config/GSI/TATWChannelMapXML.map": -1
"./config/GSI/TATWbarsMapStatus.map": -1
"./calib/GSI/TATW_Energy_Calibration.cal": -1
"./calib/GSI/TATW_Tof_Calibration.cal": -1
"./calib/GSI/TATWEnergyTuning.cal": -1
```

- Campaign name with associated run
- Detector name with associated file number and name
- Name following with the corresponding run number validity (-1 no dependence)

Campaign manager (iii)

□ Run dependency: (as example)

```
// Campaign file
CamName: "GSI"
RunNumber: 2187-2261
NumberDevices: 6

. . .

DetectorName: "BM"
NumberFiles: 4
"./geomaps/GSI/TABMdetector.geo": -1
"./config/GSI/TABMdetector.cfg": -1
"./config/GSI/TABMdetector.map": -1
"./calib/GSI/TABM_T0_Calibration.cal": 2210; 2239
. . .
```

- Config file ./config/GSI/TABM_T0_Calibration_2210.cal for BM is valid for runs 2210-2238
- Config file ./config/GSI/TABM_T0_Calibration_2239.cal for BM is valid for runs 2239-2261

Campaign manager (iv)

□ MC Structure: ./cammapp/12C_200.cam

```
// Campaign file
CamName: "12C_200"
RunNumber: 1
NumberDevices: 10

DetectorName: "FOOT"
NumberFiles: 1
"./geomaps/12C_200/FOOT.geo": -1

DetectorName: "DI"
NumberFiles: 1
"./geomaps/12C_200/TADIdetector.geo": -1

DetectorName: "ST"
NumberFiles: 2
"./geomaps/12C_200/TASTdetector.geo": -1
"./config/12C_200/TASTdetector.cfg": -1

DetectorName: "BM"
NumberFiles: 2
"./geomaps/12C_200/TABMdetector.geo": -1
"./config/12C_200/TABMdetector.cfg": -1

DetectorName: "TG"
NumberFiles: 1
"./geomaps/12C_200/TAGdetector.geo": -1
```

```
DetectorName: "VT"
NumberFiles: 2
"./geomaps/12C_200/TAVTdetector.geo": -1
"./config/12C_200/TAVTdetector.cfg": -1

DetectorName: "IT"
NumberFiles: 2
"./geomaps/12C_200/TAITdetector.geo": -1
"./config/12C_200/TAITdetector.cfg": -1

DetectorName: "MSD"
NumberFiles: 1
"./geomaps/12C_200/TAMSDdetector.geo": -1

DetectorName: "TW"
NumberFiles: 5
"./geomaps/12C_200/TATWdetector.geo": -1
"./config/12C_200/TATW_BBparameters.cfg": -1
"./config/12C_200/TATWbarsMapStatus.map": -1
"./calib/12C_200/TATW_Energy_Calibration.cal": -1
"./calib/12C_200/TATW_Tof_Calibration.cal": -1

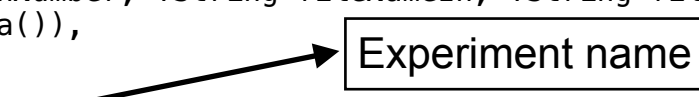
DetectorName: "CA"
NumberFiles: 1
"./geomaps/12C_200/TACAdetector.geo": -1
```

➔ Campaign manager will return the name of the parameter files with the right name with respect to campaign and run number for a given detector.

Campaign manager (v)

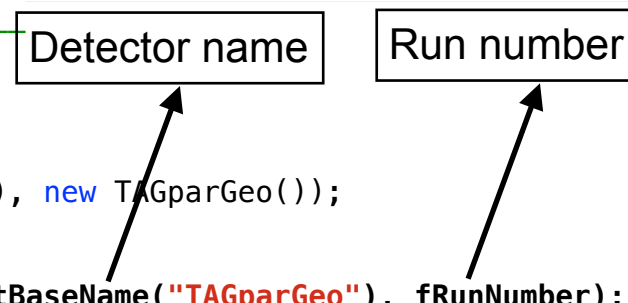
□ Example of implementation:

```
BaseReco::BaseReco(TString expName, Int_t runNumber, TString fileNameIn, TString fileNameout)
: TNamed(fileNameIn.Data(), fileNameout.Data()),
. . .
// load campaign file
fCampaignManager = new TAGcampaignManager(expName);
fCampaignManager->FromFile();
. . .
}
```



```
//
void BaseReco::ReadParFiles()
{
. . .
if (GlobalPar::GetPar()->IncludeTG()) {
    fpParGeoG = new TAGparaDsc(TAGparGeo::GetDefParaName(), new TAGparGeo());
    TAGparGeo* parGeo = (TAGparGeo*)fpParGeoG->Object();

    TString parFileName = fCampaignManager->GetCurGeoFile(FootBaseName("TAGparGeo"), fRunNumber);
    parGeo->FromFile(parFileName.Data());
}
. . .
```



➔ Campaign manager from detector name and the run number will compose the name of the different cfg, map, cal, geo files.

Name Manager

TAGnameManager (i)

- Action names: TAGnameManager::GetActionDscName(className)

		Raw
TAGactTreeWriter	evtActWriter	
TAGactDaqReader	daqActReader	
TAGactNtuEvent	evtActEvent	
TAGactWDreader	wdActreader	
TASTactNtuHit	stActHit	
TABMactNtuRaw	bmActRaw	
TABMactNtuHit	bmActHit	
TAVTactNtuHit	vtActHit	
TAMSDactNtuRaw	msActRaw	
TAMSDactNtuHit	msActHit	
TATWactNtuHit	twActHit	
TACAactNtuHit	caActHit	
TABMactNtuTrack	bmActTrack	
TAVTactNtuCluster	vtActClus	
TAVTactNtuTrackF	vtActTrack	
TAVTactNtuVertexPD	vtActVertex	
TAMSDactNtuCluster	msActClus	
TAMSDactNtuPoint	msActPoint	
TATWactNtuPoint	twActPoint	
TACAactNtuClusterP	caActClus	

		MC
TAGactTreeReader	evtActReader	
TASTactNtuHitMC	stActHitMc	
TABMactNtuHitMC	bmActHitMc	
TAVTactNtuHitMC	vtActHitMc	
TAMSDactNtuHitMC	msActHitMc	
TATWactNtuHitMC	twActHitMc	
TACAactNtuHitMC	caActHitMc	

→ name: detector base+"Act"+action type
(TABMactNtuRaw → **bmActRaw**)

Alias:

```
#define FootParaDscName(className) TAGnameManager::GetParaDscName(className).Data()  
#define FootActionDscName(className) TAGnameManager::GetActionDscName(className).Data()  
#define FootDataDscMcName(className) TAGnameManager::GetDataDscMcName(className).Data()
```

TAGnameManager (ii)

- Container names: TAGnameManager::GetActionDscName(className)

TAGdaqEvent	daqEvent	Raw
TAGntuEvent	evtEvent	
TACAntuRaw	caRaw	
TASTntuRaw	stRaw	
TATWntuRaw	twRaw	
TAWDntuTrigger	wdTrigger	
TASTntuHit	stHit	
TABMntuRaw	bmRaw	
TABMntuHit	bmHit	
TAVTntuHit	vtHit	
TAMSDntuRaw	msRaw	
TAMSDntuHit	msHit	
TATWntuHit	twHit	
TACAntuHit	caHit	
TABMntuTrack	bmTrack	
TAVTntuTrack	vtTrack	
TAVTntuVertex	vtVertex	
TAVTntuCluster	vtClus	
TAMSDntuCluster	msClus	
TAMSDntuPoint	msPoint	
TATWntuPoint	twPoint	
TACAntuCluster	caClus	

TAMCntuEvent	mcEvent	MC
TAMCntuPart	mcPart	
TAMCntuHit	Hit	
TAMCntuHit	Hit	
TAMCntuHit	Hit	
TAMCntuHit	Hit	
TAMCntuHit	Hit	
TAMCntuHit	Hit	

➔ Same class leads to same name

- Special method for MC hit container:

TAGnameManager::GetDataDsc**Mc**Name(idx)

Idx = { kST, kBM, kVTX, kITR, kMSD, kTW, kCAL}

TAMCntuHit	stHitMc
TAMCntuHit	bmHitMc
TAMCntuHit	vtHitMc
TAMCntuHit	msHitMc
TAMCntuHit	twHitMc
TAMCntuHit	caHitMc

➔ name: detector base+container type

(TABMntuRaw → **bmRaw**)

➔ Changing a class name (include 'ntu' in name)

TAGnameManager (iii)

- Interface:
 - In base classes (TAGparaDsc, TAGdataDsc, TAGaction)

```
class TAGparaDsc : public TAGnamed {
public:
    //! fail bit
    enum { kValid = BIT(15)
};

    explicit          TAGparaDsc(const char* name, TAGpara* p_para=0);
explicit          TAGparaDsc(TAGpara* p_para=0);
    virtual          ~TAGparaDsc();
    . . .
};

TAGparaDsc::TAGparaDsc(TAGpara* p_para)
: TAGnamed("", 0),
  fpObject(0),
  fpObjectClass(0),
  fpConsumerList(0)
{
    if (!gTAGroot) Fatal("TAGparaDsc()", "TAGroot not instantiated");
    SetBit(kMustCleanup);

    SetName(gTAGroot->DefaultParaDscName(p_para->ClassName()));
    . . .
}
```

- Add new constructor (same way for the TAGdataDsc & TAGaction)
 - ➔ Overloading TAGroot::DefaultParaDscName() method

TAGnameManager (iv)

- Interface:
 - Using the default name methods in TAGroot

```
class TAGroot : public TAGobject {
public:
    TAGroot();
    virtual ~TAGroot();
    . . .
    // Default action name
    const char* DefaultActionName();
    // Default data dsc name
    const char* DefaultDataDscName();
    // Default parameter dsc name
    const char* DefaultParaDscName();
    . . .
    // Default action name
    const char* DefaultActionName(const char* className);
    // Default data dsc name
    const char* DefaultDataDscName(const char* className);
    // Default parameter dsc name
    const char* DefaultParaDscName(const char* className);
    . . .
};

const char* TAGroot::DefaultActionName(const char* className)
{
    return Form("%s", FootParaDscName(className));
}
. . .
```

➔ No need to set the name, generated via TAGnameManager class

Implementation (i)

- Interface in Reco classes:

- Parameters:

```
fpParGeoBm = new TAGparaDsc("bmGeo", new TABMparGeo());
```



```
fpParGeoBm = new TAGparaDsc(new TABMparGeo());
```

➔ Just removing parameter Dsc name

- Data containers:

```
fpNtuTrackBm = new TAGdataDsc("bmTrack", new TABMntuTrack());
```



```
fpNtuTrackBm = new TAGdataDsc(new TABMntuTrack());
```

➔ Just removing data Dsc name

Implementation (ii)

- Reminder, does not work in case of TAMCntuHit:

```
fpNtuMcBm = new TAGdataDsc("bmMc", new TAMCntuHit());
```



```
fpNtuMcBm = new TAGdataDsc(FootDataDscMcName(kBM), new TAMCntuHit());
```

- Actions:

```
fActClusVtx = new TAVTactNtuCluster("vtActClus", fpNtuHitVtx, fpNtuClusVtx, fpParConfVtx, fpParGeoVtx);
```



```
fActClusVtx = new TAVTactNtuCluster("vtActClus", fpNtuHitVtx, fpNtuClusVtx, fpParConfVtx, fpParGeoVtx);
```



- Cannot simply remove name, need to modify all mother classes



```
const Char_t* name = FootActionDscName("TAVTactNtuCluster");  
fActClusVtx = new TAVTactNtuCluster(name, fpNtuHitVtx, fpNtuClusVtx, fpParConfVtx, fpParGeoVtx);
```

Implementation (iii)

- Branch name:

```
fActEvtWriter->SetupElementBranch(fpNtuMcEvt, TAMCntuEvent::GetBranchName());
```



```
fActEvtWriter->SetupElementBranch(fpNtuMcEvt);
```

```
void TAGactTreeWriter::SetupElementBranch(TAGdataDsc* p_data, Int_t i_size, Int_t i_compress)
{
    TAGdata* obj = p_data->Object();
    TString name(obj->ClassName());
    const char* branch = TAGnameManager::GetBranchName(name);
    TAGactTreeWriterBranch* p_chan = new TAGactTreeWriterBranch(p_data, branch, i_size, i_compress, kFALSE);

    AddDataIn(p_data, "TAGdata"); // ??? be more specific ???
    fpBranchList->Add(p_chan);
    return;
}
```


Implementation (iv)

- Branch name:

```
fActEvtReader->SetupBranch(fpNtuTrackBm, TABMntuTrack::GetBranchName());
```



```
fActEvtReader->SetupBranch(fpNtuTrackBm);
```

```
void TAGactTreeReader::SetupBranch(TAGdataDsc* p_data)
{
    TAGdata* obj = p_data->Object();
    TString name(obj->ClassName());
    const char* branch = TAGnameManager::GetBranchName(name);

    TAGactTreeReaderBranch* p_chan = new TAGactTreeReaderBranch(p_data, branch);
    AddDataOut(p_data, "TAGdata"); // ??? be more specific ???
    fpBranchList->Add(p_chan);
    return;
}
```

Implementation (v)

- Branch name:
 - Special case for TAMCntuHit

```
fActEvtWriter->SetupElementBranch(fpNtuMcSt, TAMCntuHit::GetStcBranchName());
```



```
fActEvtWriter->SetupElementBranch(fpNtuMcSt, FootBranchMcName(kST));
```

Global Parameters

Foot Global Parameters (i)

□ File: FootGlobal.par (i)

```
...
IncludeKalman:      n
IncludeTOE:        n
IncludeStraight:   n
FromLocalReco:    n
```

Global Reconstruction Options

```
Kalman Mode:      ON
Tracking Systems Considered: VI IT MSD
Reverse Tracking: false
```

```
...
##### Options for reconstruction #####
```

```
EnableTree:      y
EnableHisto:     y
EnableTracking:  y

EnableSaveHits:  n
EnableRootObject: n
EnableRegionMc:  n
EnableElecNoiseMc: n
```

Reconstruction Options

```
EnableTree:      enable tree filling
EnableHisto:    enable control histograms
EnableTracking: enable tracking actions
EnableSaveHits : enable saving hits in tree
EnableRootObject: enable reading from root object
EnableRegionMc: enable region reading
EnableElecNoiseMc: enable electronics noise in MC
```

Enable Glb Reco from level0 trees

Foot Global Parameters (i)

□ File: FootGlobal.par (ii)

```
##### END - Options for reconstruction
```

IncludeDI:	n
IncludeST:	y
IncludeBM:	y
IncludeTG:	y
IncludeVT:	y
IncludeIT:	n
IncludeMSD:	n
IncludeTW:	y
IncludeCA:	n

Detector to switch on/off

- Enable reconstruction of the different detectors
- Other options are depreciated or only use for global Genfit reconstruction

Foot Global Parameters (iii)

□ Debug Level

```
. . .  
Debug: 0  
ClassDebugLevel: TAVTparGeo 1  
. . .  
ClassDebugLevel: TAVTactBaseNtuMC 2  
. . .
```

□ Examples: TAVTbaseParGeo/TAVTactBaseNtuMC

```
Bool_t TAVTbaseParGeo::FromFile(const TString& name)  
{  
    . . .  
    ReadItem(fSensorsN);  
    if(FootDebugLevel(1))  
        cout << endl << "Sensors number " << fSensorsN << endl;  
    . . .  
}
```

```
void TAVTactBaseNtuMC::ComputeNoiseLevel()  
{  
    . . .  
    FootDebug(1, "ComputeNoiseLevel()", Form("Number of noise pixels %d\n", fNoisyPixelsN));  
    . . .  
}
```

➡ Add class name and debug level for each class

Foot Global Parameters (iv)

□ TAGrunInfo (i):

```
struct GlobalParameter_t : public TObject {  
    . . .  
    Bool_t EnableLocalReco;  
    Bool_t EnableTree;  
    Bool_t EnableHisto;  
    Bool_t EnableSaveHits;  
    Bool_t EnableTracking;  
    Bool_t EnableRootObject;  
    Bool_t IncludeKalman;  
    Bool_t IncludeTOE;  
    Bool_t IncludeDI;  
    Bool_t IncludeST;  
    Bool_t IncludeBM;  
    Bool_t IncludeTG;  
    Bool_t IncludeVT;  
    Bool_t IncludeIT;  
    Bool_t IncludeMSD;  
    Bool_t IncludeTW;  
    Bool_t IncludeCA;  
    ClassDef(GlobalParameter_t,1)  
};  
  
class TAGrunInfo : public TAGobject {  
private:  
    TString          fsCam;          // campaign name  
    Short_t         fiRun;          // run number  
    GlobalParameter_t fGlobalParameter; // global parameters  
    . . .  
};
```

➔ Add all reconstruction parameters in run info

Foot Global Parameters (v)

□ TAGrunInfo : printout (ii)

```
KEY: TAGrunInfo runinfo;1
root [3] runinfo->Print()
Run info:          cam: 12C_200/  run:    1
Global info:
  EnableLocalReco: 0
  EnableTree: 1
  EnableHisto: 1
  EnableTracking: 1
  EnableSaveHits: 0
  EnableRootObject: 0

  IncludeKalman: 0
  IncludeTOE: 1

  IncludeDI: 1
  IncludeST: 1
  IncludeBM: 1
  IncludeTG: 1
  IncludeVT: 1
  IncludeIT: 1
  IncludeMSD: 1
  IncludeTW: 1
  IncludeCA: 0
```

➔ Run info saved at each stage, MC, local and global reconstruction

Run manager

Run Manager (i)

Run manager

- Import detector out column from log, example: CNAO2023.run

```
# Run file
CamName: "CNAO2023"
RunNumber: 5957-6313;

. . .

# run      starttime          stoptime          Daq Evt  Duration  DAQ Rate   Run type  Comment
#          (s)              (s)              (#)      (s)      (Hz)
5957 "2023-10-27 23:56:13"  "2023-10-27 23:59:19"  79889    186      429        7        "None"
5958 "2023-10-27 23:59:50"  "2023-10-28 00:00:46"  26711    56       476        7        "None"
. . .

6312 "2023-11-08 03:42:38"  "2023-11-08 04:17:36"  193105   2098     92         5        "large TDC window no clipping"
6313 "2023-11-08 04:19:43"  "2023-11-08 04:20:36"  12128    53       228        3        "Pedestal"
```

Detectors Out

"BM VT IT MSD"
"BM VT IT MSD"

"None"
"None"

→ Check if detector really present, if not excluded in Global parameter

```
Bool_t TAGrunManager::ConditionChecks(Int_t runNumber, TAGparGeo* parGeo)
{
/ Check if a detector is off in a given run
  vector<TString> list = TAGrecoManager::GetPar()->DectIncluded();
  for (vector<TString>::const_iterator it = list.begin(); it != list.end(); ++it) {
    TString str = *it;

    if (IsDetectorOff(str)) {
      Warning("Checks()", "the detector %s is NOT referenced in this run, Disable it",
str.Data());
      if (str == "ST")
        TAGrecoManager::GetPar()->IncludeST(false);
      if (str == "BM")
        TAGrecoManager::GetPar()->IncludeBM(false);
    }
  }
}
```

```
IncludeDI:      n
IncludeST:      n
IncludeBM:      n
IncludeTG:      n
IncludeVT:    y
IncludeIT:      n
IncludeMSD:     n
IncludeTW:      n
IncludeCA:      n
```

```
Warning in <TAGrunManager::Checks(): the detector VT is NOT referenced in this run, Disable it
```

Run Manager (ii)

- TAGrunManager:

```
class TAGrunManager : public TAGaction {
public:
    /*!
     \struct TypeParameter_t
     \brief Run type parameters
     */
    struct TypeParameter_t : public TNamed {
        Int_t      TypeId;          ///< Index type run
        TString    TypeName;       ///< Type of run name
        TString    Trigger;        ///< Trigger type
        TString    Beam;           ///< Beam element
        Float_t    BeamEnergy;     ///< Beam energy per nucleon
        Float_t    BeamEnergy2;   ///< Upper beam energy per nucleon
        TString    Target;         ///< Target element
        Float_t    TargetSize;     ///< Target size
        vector<string> DetectorOut; ///< Dectector not present
        Int_t      TotalEvts;      ///< Total event number
        TString    Comments;       ///< Comments
        TString    MagnetFlag;     ///< Magnet on (yes, no)
    };
};
```

```
/*!
 \struct RunParameter_t
 \brief Run parameters
 */
struct RunParameter_t : public TNamed {
    Int_t      RunId;          ///< Index type run
    TString    StartTime;     ///< Run starting time
    TString    StopTime;      ///< Run stopping time
    Int_t      DaqEvts;       ///< DAQ event number in run
    Int_t      Duration;      ///< Run duration
    Int_t      DaqRate;       ///< DAQ rate number in run
    Int_t      RunType;       ///< Run type
    TString    Comments;      ///< Comments
    vector<string> DetectorOut; ///< Detector not present
};
. . .
```

- Handle type of run and run information

Run Manager (iii)

- Output when running the decoding

```
Current run number: 6144
  Type: 4
  Daq events: 207 334
  Duration: 2538 s [00h:42m:18s]
  Daq Rate: 81 Hz
  Comments: pure frag
  DetectorOut: None
```

```
Current type:
  Type index: 4
  Type name: Carbon target & frag
  Main Trigger: Fragmentation
  Type Beam: 12C
  Beam Energy: 200.0 MeV/u
  Type Target: Carbon
  Magnet ON: Yes
  Target Size: 0.50 cm
  Total Events: 2 072 468
  Comments: None
```

- Print out information about the given run and type of the run

Executables

Local Reconstruction

□ Command line:

```
. . .  
EnableTree:      y  
EnableHisto:    y  
EnableTracking:  y  
  
EnableSaveHits: n  
EnableRootObject: n  
  
. . .
```

Set global parameters



➔ from MC hits:

```
DecodeGlb -in 12C_C_200_1.root -out 12C_C_200_L0Out.root -nev 10000  
-exp 12C_200 -run 1 -mc
```

➔ from raw hits:

```
DecodeGlb -in data/data_built.2211.physics_foot.daq.VTX.1.dat  
-out 12C_C_200_L0Out.root -nev 10000 -exp GSI -run 2211
```

Global TOE Reconstruction (i)

□ Command line:

```
. . .  
IncludeKalman:  n  
IncludeTOE:    y  
FromLocalReco: n  
. . .
```

Will be put off anyhow by executable

Will be put on anyhow by executable

➔ from MC hits: `DecodeGlb -in 12C_C_200_1.root`
`-out 12C_C_200_1_GlbOut.root -nev 10000 -exp 12C_200 -run 1 -mc`

```
. . .  
IncludeKalman:  n  
IncludeTOE:    y  
FromLocalReco: y  
. . .
```

➔ from L0 reco: `DecodeGlb -in 12C_C_200_L0Out.root`
`-out 12C_C_200_1_GlbOut.root -nev 10000 -exp 12C_200 -run 1 -mc`

Global GenFit Reconstruction (i)

□ Command line:

```
. . .  
IncludeKalman:  y  
IncludeTOE:    n  
EnableLocalReco: n  
. . .
```

Will be put on anyhow by executable

Will be put off anyhow by executable

➔ from raw hits: `DecodeGlb -in data/data_built.2211.physics_foot.daq.VTX.1.dat -out 12C_C_200_1_GlbOut.root -nev 10000 -exp GSI -run 2211`

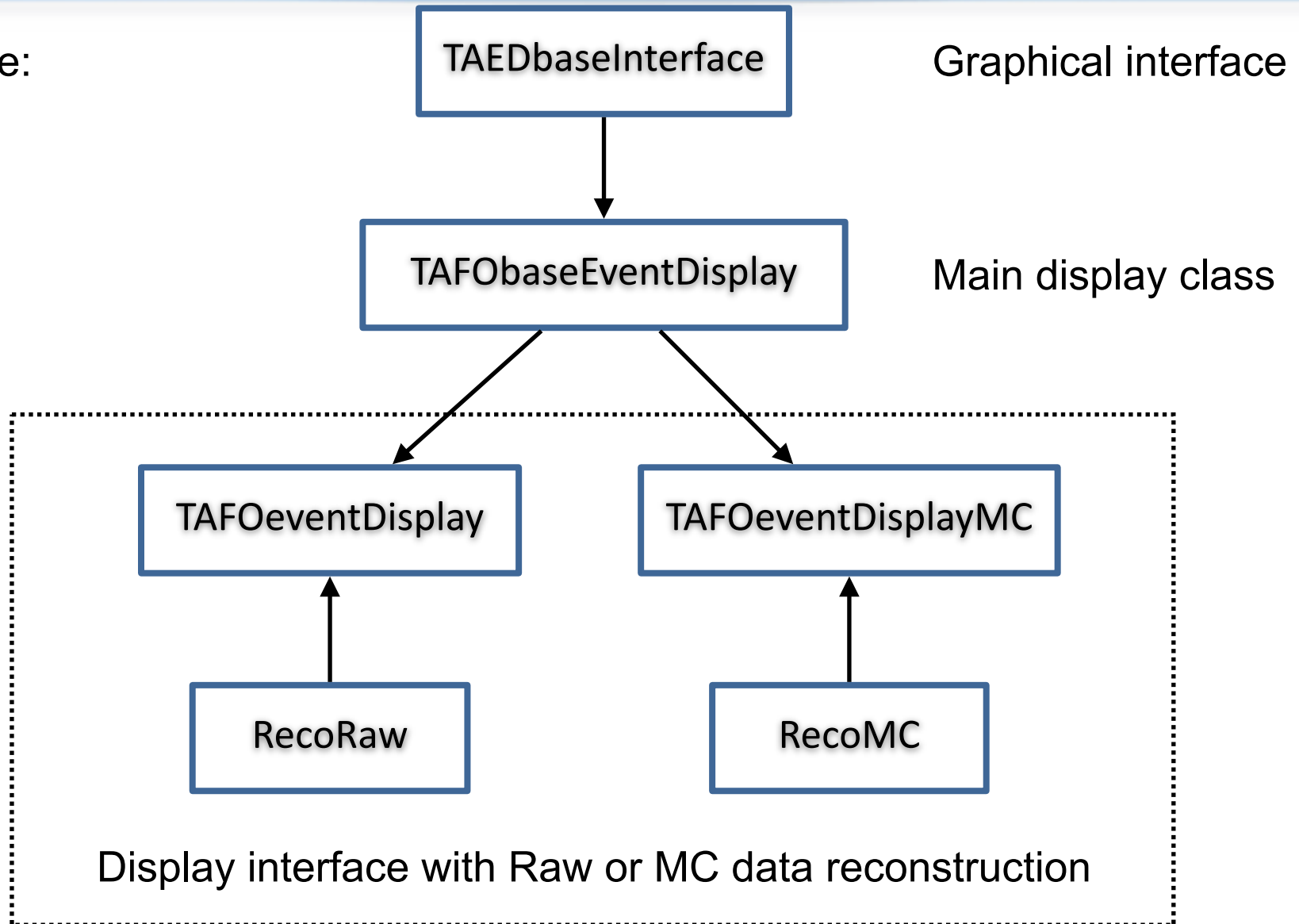
Change input file, remove -mc flag

➔ Still not fully tested cos don't have reliable raw data

Event Display

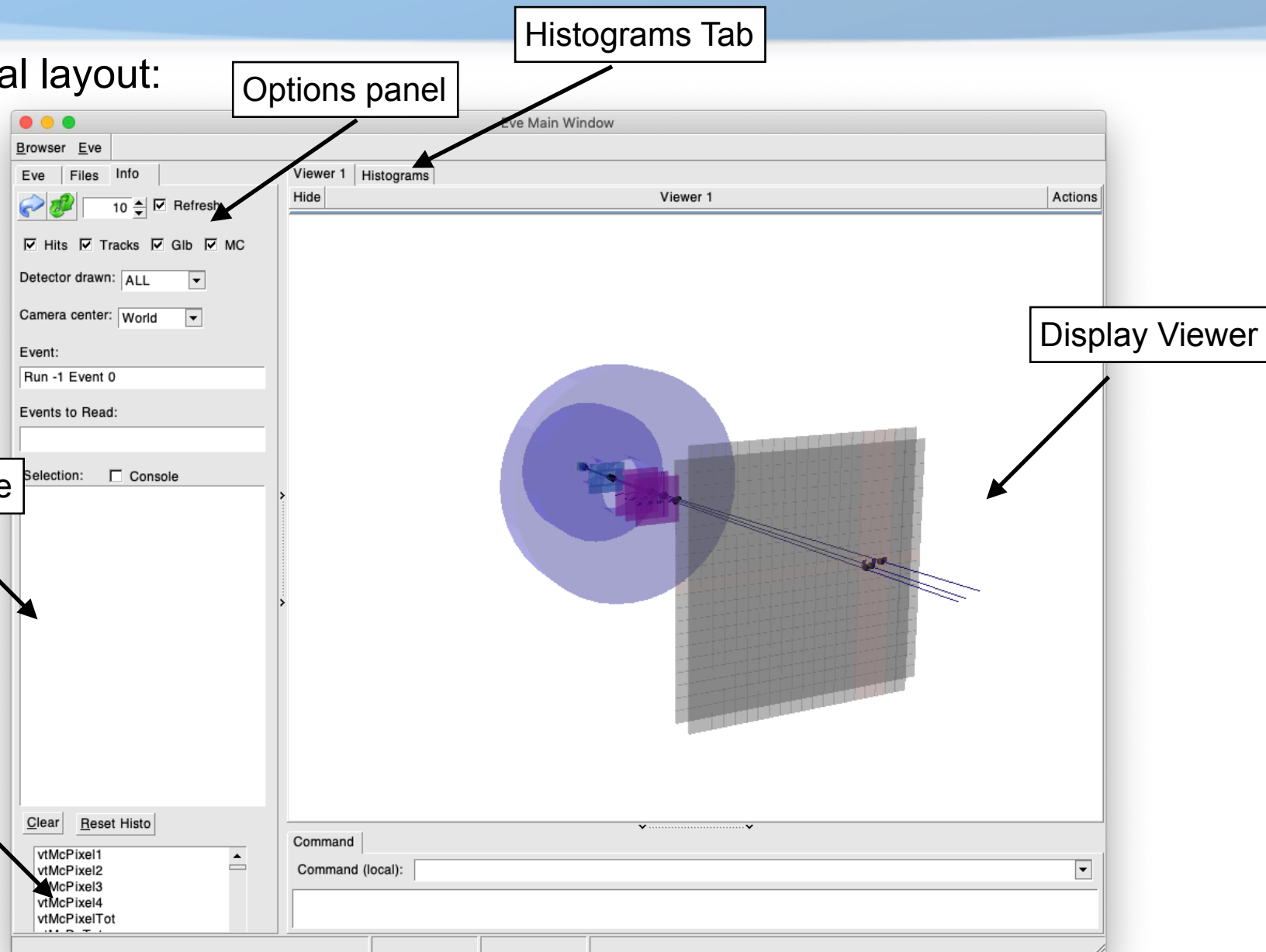
Event Display

□ Scheme:



Event Display (i)

□ General layout:



Event Display (ii)

Navigation

The screenshot shows the 'Eve Main' window with a toolbar containing four icons: a blue right-pointing arrow, a blue left-pointing arrow, a green circular arrow, and a red cube icon. A red oval highlights these four icons. Arrows from text boxes point to each icon: 'forward' points to the right arrow, 'backward (only MC)' points to the left arrow, 'go to event' points to the red cube icon, and 'loop over events' points to the green circular arrow. Below the toolbar, there are checkboxes for 'Hits', 'Tracks', and 'Glu', a 'Detector drawn' dropdown set to 'ALL', a 'Camera center' dropdown set to 'World', an 'Event:' field containing 'Run -1 Event 56', an 'Events to Read:' field, and a 'Selection:' section with a 'Console' checkbox.

forward

backward (only MC)

go to event

loop over events

Eve Main

Browser Eve

Eve Files Info

Viewer 1 Histograms

Hide

56 Refresh

Hits Tracks Glu

Detector drawn: ALL

Camera center: World

Event:

Run -1 Event 56

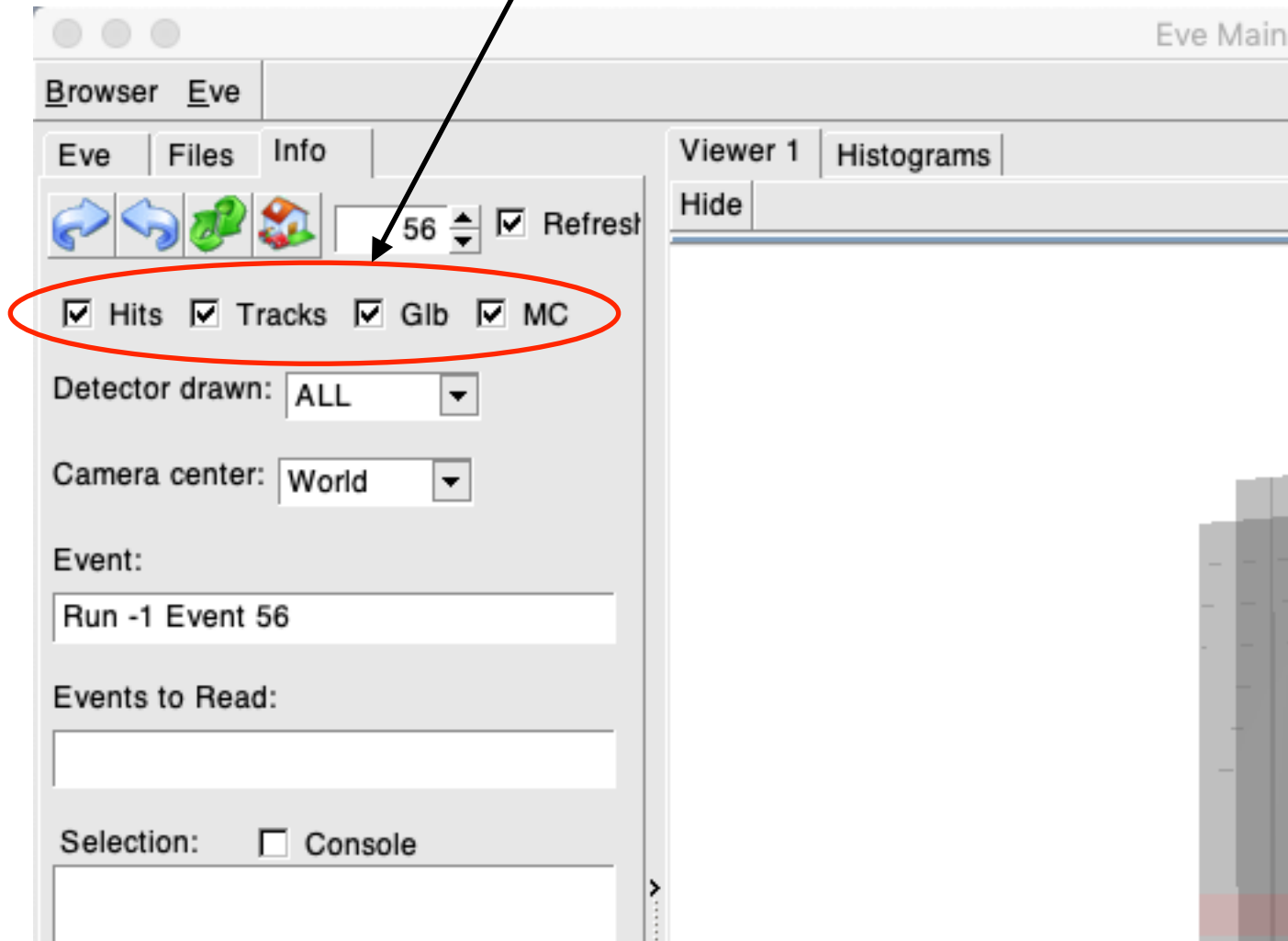
Events to Read:

Selection: Console

Event Display (iii)

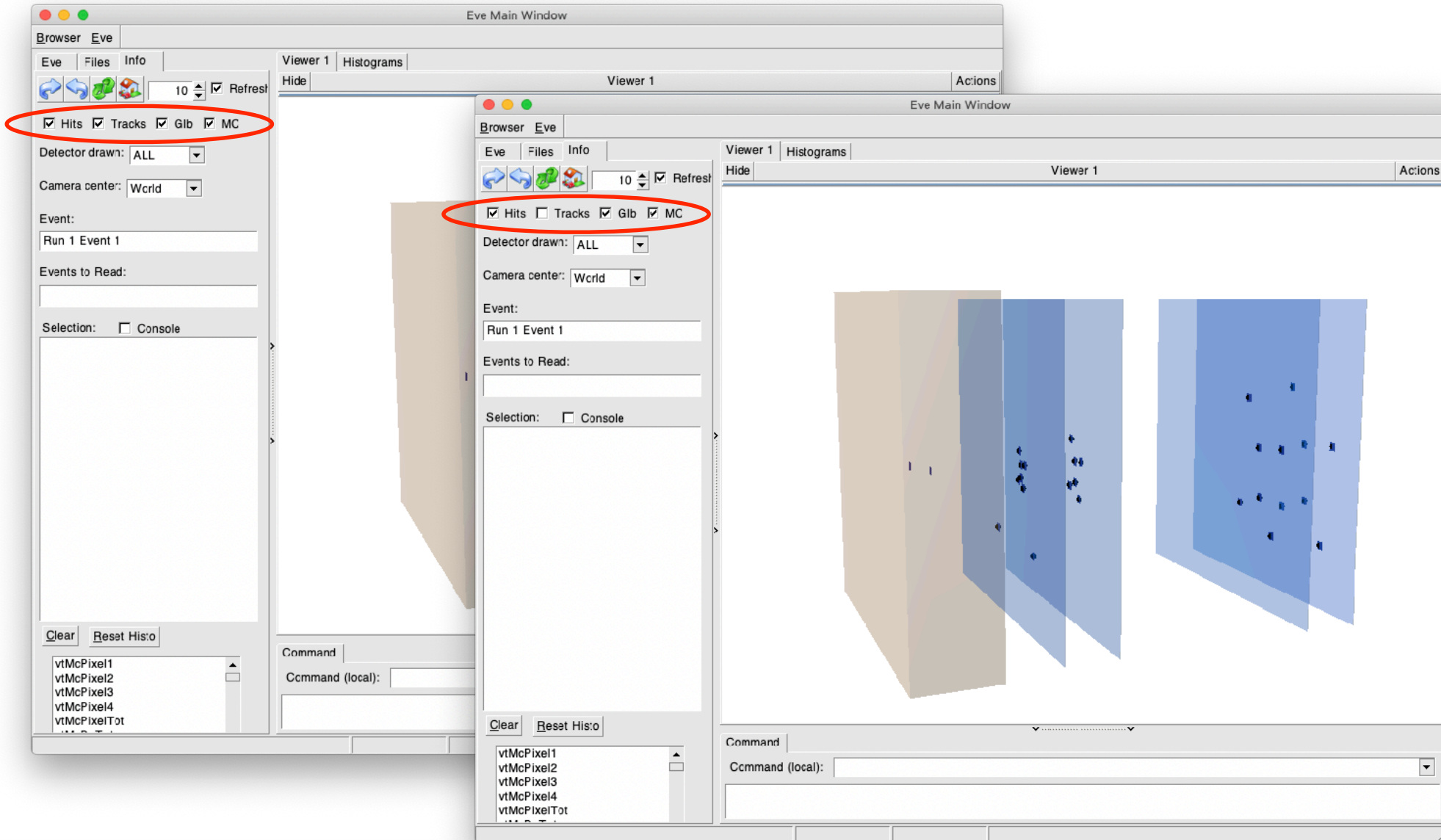
□ Drawing (i)

Toggle hits - tracks - global tracks - MC hits



Event Display (iv)

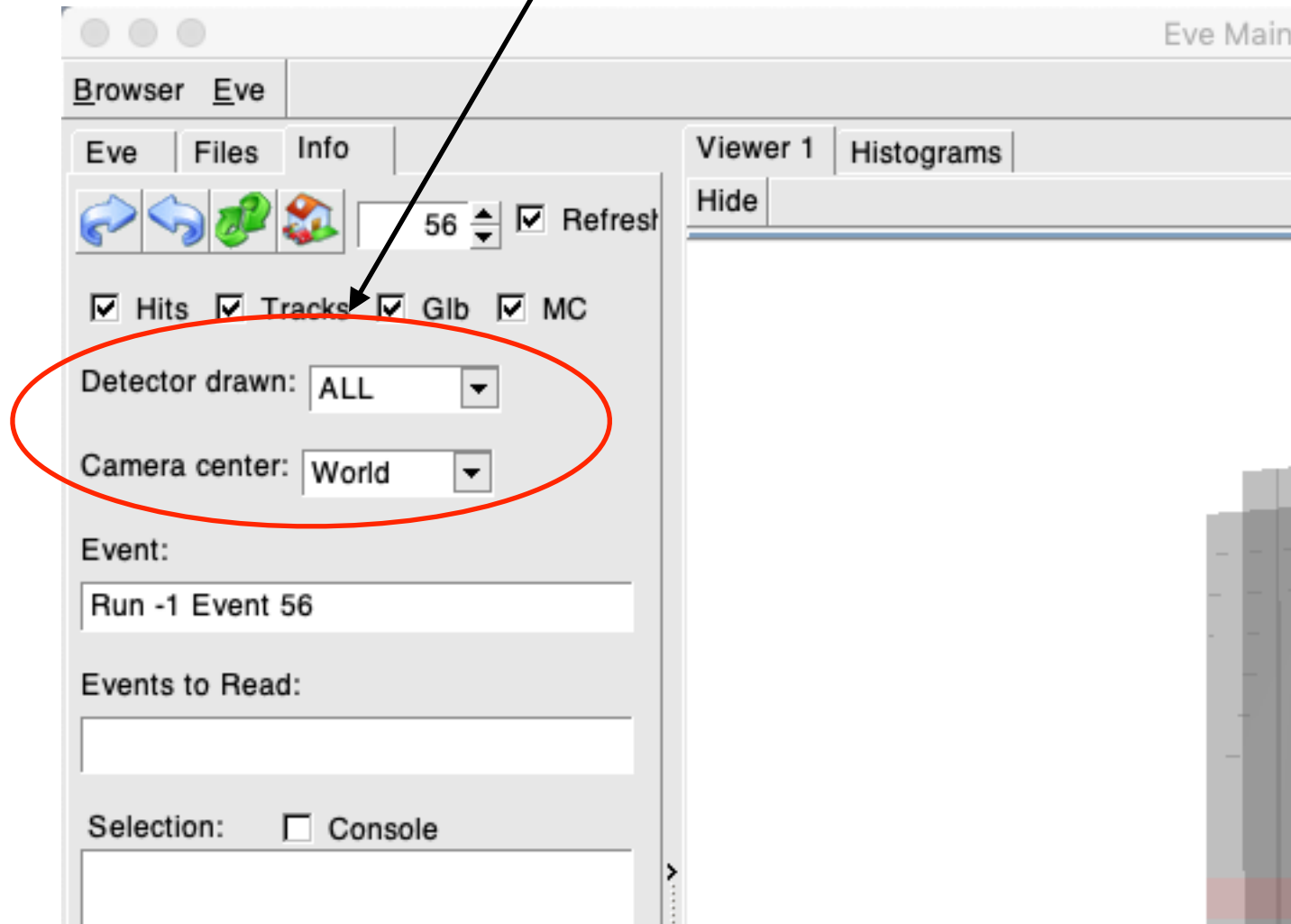
□ Drawing (ii)



Event Display (v)

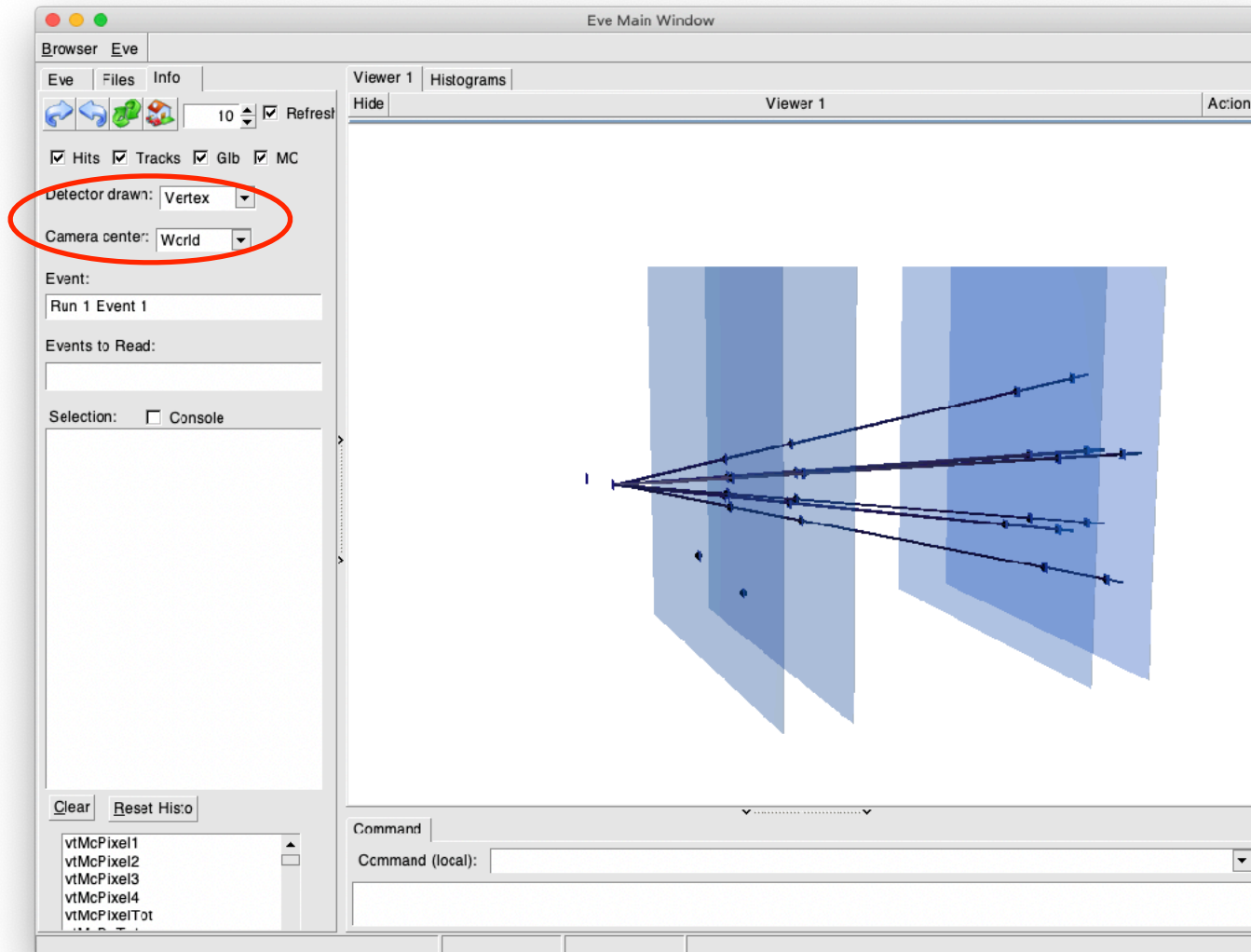
- Drawing (iii)

Menu: toggle selected detector to draw / center of camera



Event Display (vi)

□ Drawing (iv)



Event Display (vii)

□ Frame info

if activated, info also printed on your terminal

The screenshot shows the Event Display (vii) interface. The main window displays a 3D detector model with several blue planes representing detector layers. A track is shown as a series of blue lines connecting hit points on these planes. The interface includes a menu bar (Browser, Eve), a toolbar with navigation icons, and a control panel with checkboxes for Hits, Tracks, Glb, and MC. The Detector draw and Camera center are set to ALL and Wcrlid, respectively. The Event field shows 'Run 1 Event 1'. The Events to Read field is empty. The Selection field has a checkbox for Console. The terminal window at the bottom left displays the following track information:

```
Track # 5 (valid: 1):  
with 4 clusters  
for plane 3  
at position: (-0.632 -0.558  
with 7 pixels  
for plane 2  
at position: (-0.583 -0.479  
with 1 pixels  
for plane 1  
at position: (-0.395 -0.191  
with 10 pixels  
for plane 0  
at position: (-0.344 -0.112  
with 4 pixels
```

Annotations include: 'if activated, info also printed on your terminal' pointing to the terminal window; 'alt+mouse left' pointing to the track lines; and 'Track informations' pointing to the terminal window content.

- All hit, cluster and tracks are clickable

Event Display (viii)

□ Histograms (i)

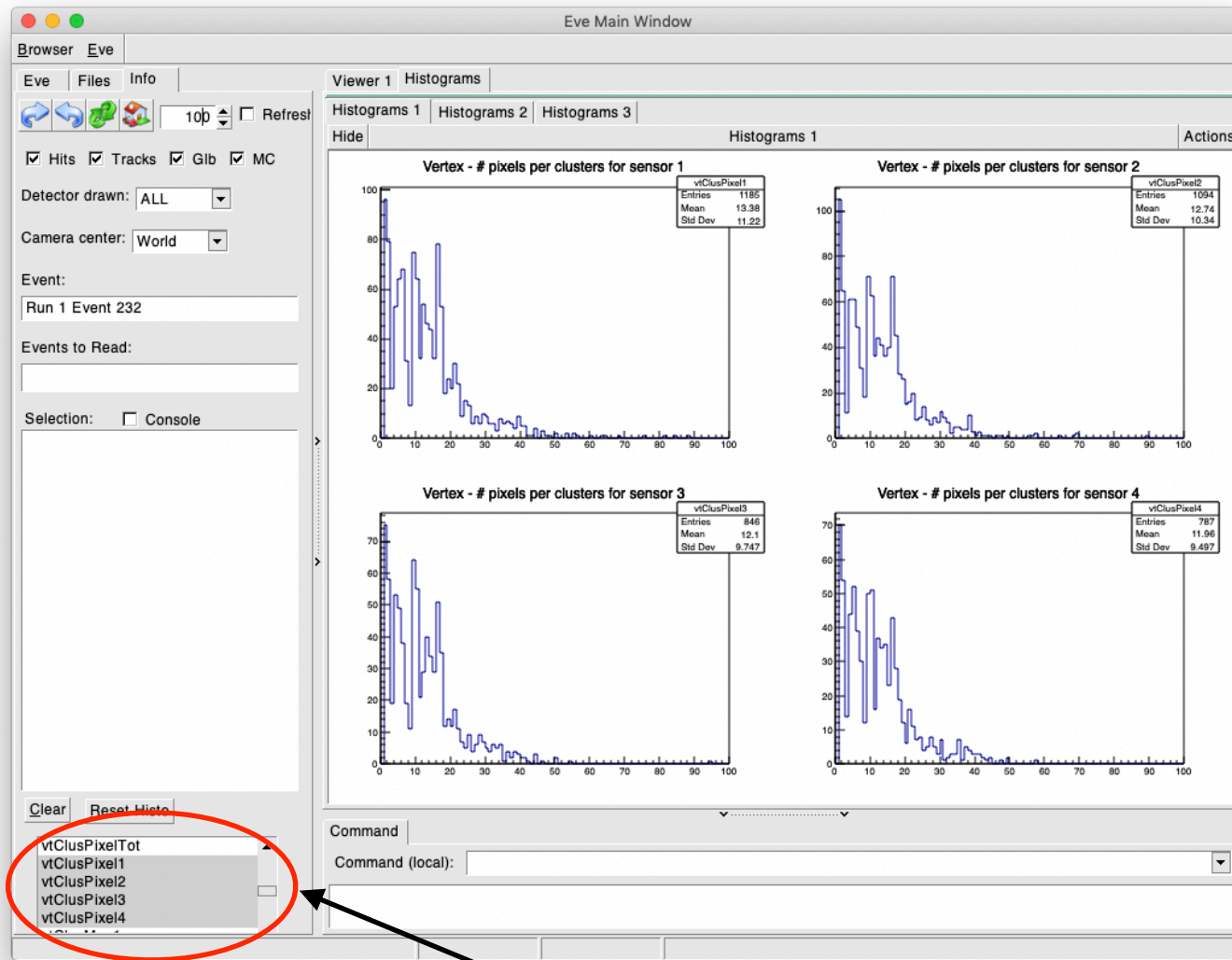
The screenshot shows the 'Eve Main Window' interface. On the left, there is a control panel with various options: 'Eve', 'Files', 'Info', 'Viewer 1', 'Histograms', 'Hide', 'Viewer 1', and 'Actions'. Below these are navigation icons, a 'Refresh' button, and checkboxes for 'Hits', 'Tracks', 'Glb', and 'MC'. There are also dropdown menus for 'Detector drawn' (set to 'ALL') and 'Camera center' (set to 'World'). The 'Event' field shows 'Run 1 Event 1'. Below that, there is a 'Selection' section with a 'Console' checkbox and a text area containing track information:

```
Track # 5 (valid: 1):  
with 4 clusters  
for plane 3  
at position: (-0.632 -0.558  
with 7 pixels  
for plane 2  
at position: (-0.583 -0.479  
with 1 pixels  
for plane 1  
at position: (-0.395 -0.191  
with 10 pixels  
for plane 0  
at position: (-0.344 -0.112  
with 4 pixels
```

At the bottom left, there is a list of histograms: 'vtMcPixel1', 'vtMcPixel2', 'vtMcPixel3', 'vtMcPixel4', and 'vtMCPixelTot'. This list is circled in red. A callout box with an arrow points to this list, containing the text 'List of histograms, clickable'. Below the histogram list are 'Clear' and 'Reset Histo' buttons. At the bottom right, there is a 'Command' field and a 'Command (local):' dropdown menu.

Event Display (ix)

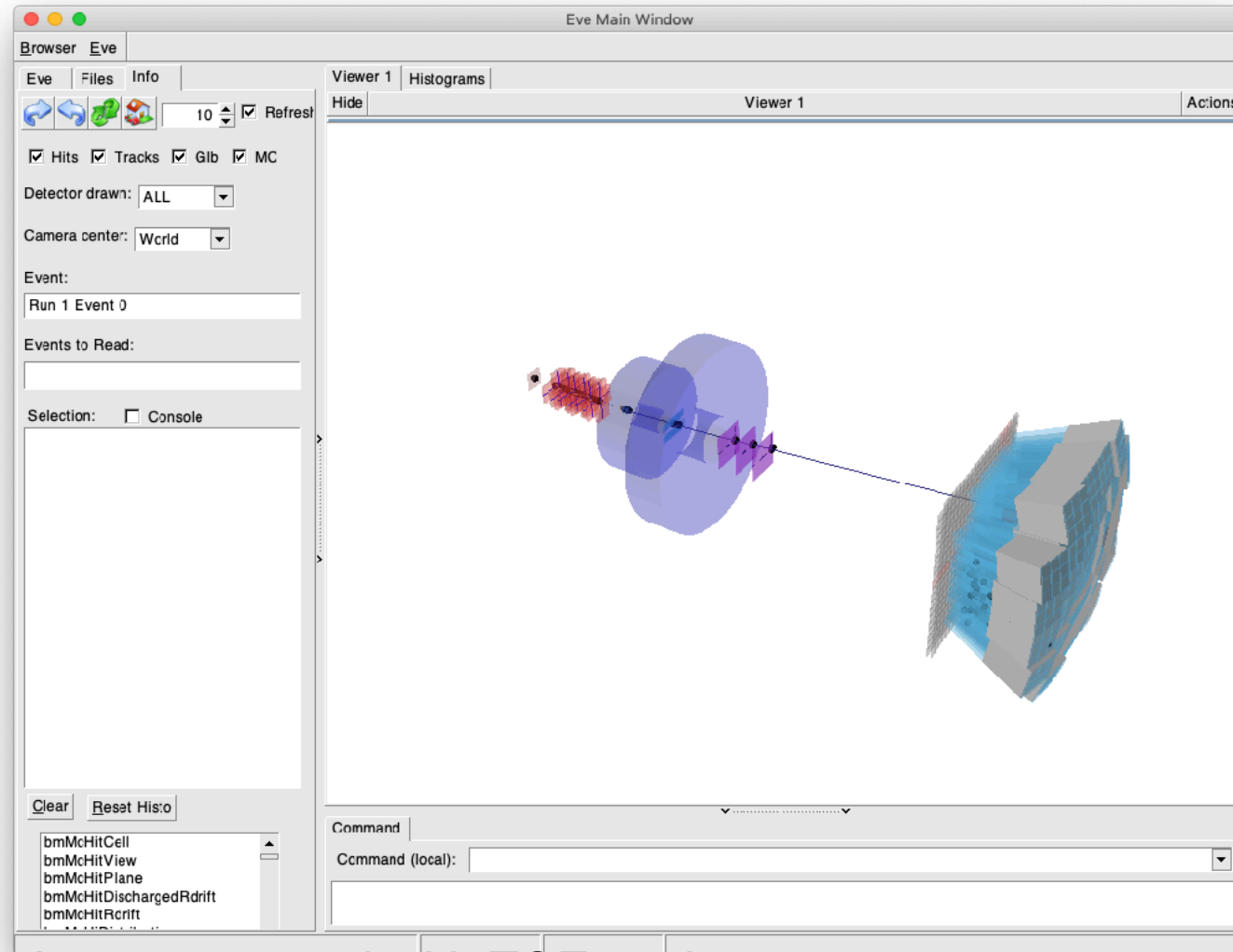
□ Histograms (ii)



Selected histograms

Event Display (x)

□ Global tracks



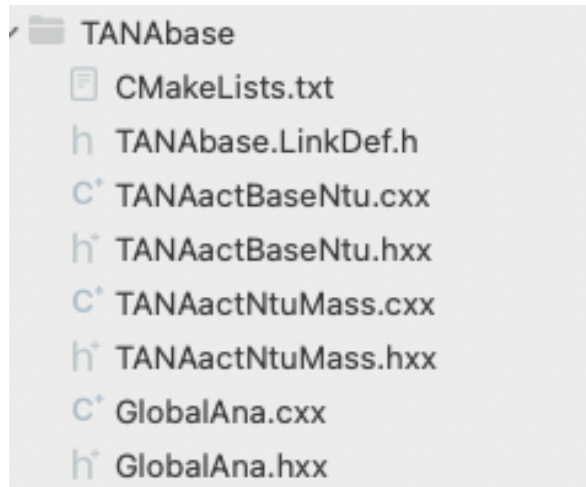
- Global track reconstructed with TOE package

Analysis framework

Classes

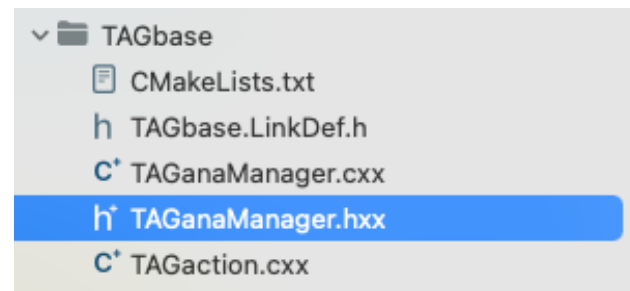
□ Analysis folder:

- Folder added:



□ Library folder:

- New analysis manager class:

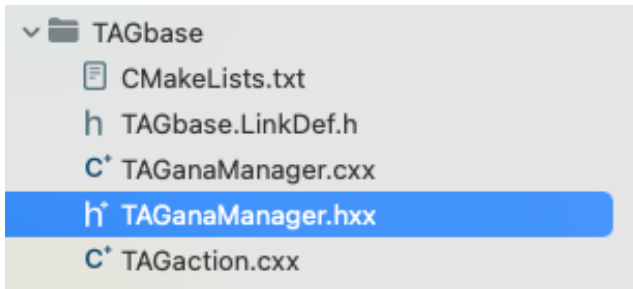


- Base class for analysis: TANAactBaseNtu
- Example of analysis class: TANAactNtuMass
- Master class managing analysis: GlobalAna

Analysis Manager

Library folder:

- New analysis manager class:



- ./config/expname/TANAdetector.cfg

```
// +-----+
// +-----+
// This is a Configuration File for FOOT analysis
// Campaign GSI2021
// +-----+
// Parameters for Analysis
// +-----+
MassReso:    1
PtReso:      0
Efficiency:  0
```

```
class TAGanaManager : public TAGparTools {
protected:
    TString    fkDefaultName; ///< default analysis file

    /*!
     \struct AnalysisParameter_t
     \brief Analysis parameters
     */
    struct GlbAnaParameter_t : public TObject {
        Bool_t    MassResoFlag;          ///< Mass resolution flag
        Bool_t    PtResoFlag;           ///< Momentum resolution flag
        Bool_t    EfficiencyFlag;       ///< Efficiency studies flag
    };

    GlbAnaParameter_t fAnalysisParameter; ///< analysis parameters
};
```

- ➔ Name of the different analysis modules
- ➔ Could add as many needed

- ➔ File only on GSI2021 and CNAO2023

Analysis Class (i)

□ Base analysis class: TANAactBaseNtu

```
class TANAactBaseNtu : public TAGaction {  
public:  
    explicit TANAactBaseNtu(const char* name           = 0,  
                           TAGdataDsc* p_irntutrack = 0,  
                           TAGparaDsc* p_geomap      = 0);  
  
    virtual ~TANAactBaseNtu();  
  
    // Create histograms  
    virtual void CreateHistogram() { return; }  
  
    // Action  
    virtual Bool_t Action() { return true; }  
  
private:  
    TAGdataDsc*   fpNtuTrack;           ///< input global tracks  
    TAGgeoTrafo*  fpFootGeo;           ///< First geometry transformer  
    TAGparaDsc*   fpGeoMapG;          ///< geometry para dsc  
  
    Float_t       fBeamEnergyTarget;   ///< Beam energy at target  
    Float_t       fBeamEnergy;         ///< Beam energy  
    Float_t       fBeamA;              ///< Beam atomic mass  
    Float_t       fBeamZ;              ///< Beam atomic number  
    ...  
}
```

- Contains the global track container, target/beam and FOOT geometry
- Histograms creation and action method are virtual

Analysis Class (ii)

□ Analysis class template: TANAactNtuMass

```
class TANAactNtuMass : public TANAactBaseNtu {  
public:  
    explicit TANAactNtuMass(const char* name          = 0,  
                           TAGdataDsc* p_irntutrack = 0,  
                           TAGparaDsc* p_geomap     = 0);  
  
    virtual ~TANAactNtuMass();  
  
    // Create histograms  
    void CreateHistogram();  
  
    // Action  
    Bool_t Action();  
  
    ClassDef(TANAactNtuMass, 0)  
};
```

- ➔ Inherits from base class
- ➔ Could add as much containers and descriptors needed
- ➔ Will save automatically the created histograms
(not foreseen saving in a tree, but could be done by passing the tree of TAGactTreeWriter as argument)

Global Analysis Class (i)

Global analysis class: GlobalAna

```
class GlobalAna : public TNamed // using TNamed for the in/out files
{
public:
    // default constructor
    GlobalAna(TString expName, Int_t runNumber, TString fileNameIn, TString fileNameout, Bool_t isMC = false);

    // default destructor
    virtual ~GlobalAna();

    // Read parameters
    void ReadParFiles();

    // Create raw action
    virtual void CreateAnaAction();

    // Add required items
    virtual void AddRequiredItem();

    // Set histogram directory
    virtual void SetHistogramDir();

    // Loop events
    virtual void LoopEvent(Int_t nEvents);

    // Begin loop
    virtual void BeforeEventLoop();

    // End loop
    virtual void AfterEventLoop();

    // Open File Out
    virtual void OpenFileOut();

    // Close File Out
    virtual void CloseFileOut();

    // Create L0 branch in tree
    virtual void SetTreeBranches();

    // Goto Event
    virtual Bool_t GoEvent(Int_t iEvent);
};
```

➔ Base on the structure of BaseReco class

Global Analysis Class (ii)

Global analysis class: GlobalAna

```
//  
//! Read parameters files  
void GlobalAna::ReadParFiles()  
{  
    Int_t Z_beam = 0;  
    Int_t A_beam = 0;  
    TString ion_name;  
    Float_t kinE_beam = 0.;  
  
    // Read Trafo file  
    TString parFileName = fCampManager->GetCurGeoFile(FootBaseName("TAGgeoTrafo"), fRunNumber);  
    fpFootGeo->FromFile(parFileName);  
  
    . . .  
  
    // initialise par files for start counter  
    if (TAGrecoManager::GetPar()->IncludeST() || TAGrecoManager::GetPar()->IncludeTW() || TAGrecoManager::GetPar()->IncludeCA()) {  
        fpParGeoSt = new TAGparaDsc(new TASTparGeo());  
        TASTparGeo* parGeo = (TASTparGeo*)fpParGeoSt->Object();  
        TString parFileName = fCampManager->GetCurGeoFile(FootBaseName("TASTparGeo"), fRunNumber);  
        parGeo->FromFile(parFileName.Data());  
  
        fpParConfSt = new TAGparaDsc(new TASTparConf());  
        TASTparConf* parConf = (TASTparConf*)fpParConfSt->Object();  
        parFileName = fCampManager->GetCurConfFile(FootBaseName("TASTparGeo"), fRunNumber);  
        parConf->FromFile(parFileName.Data());  
    }  
  
    . . .  
}
```

➔ Read all geomaps/config files for all included detectors

Global Analysis Class (iii)

Global analysis class: GlobalAna

```
//  
//! Set L0 tree branches for reading back  
void GlobalAna::SetTreeBranches()  
{  
    const Char_t* name = FootActionDscName("TAGactTreeReader");  
    fActEvtReader = new TAGactTreeReader(name);  
  
    if (TAGrecoManager::GetPar()->IncludeST()) {  
        fpNtuHitSt = new TAGdataDsc(new TASTntuHit());  
        fActEvtReader->SetupBranch(fpNtuHitSt);  
  
        if (fFlagMC) {  
            fpNtuMcSt = new TAGdataDsc(FootDataDscMcName(kST), new TAMCntuHit());  
            fActEvtReader->SetupBranch(fpNtuMcSt, FootBranchMcName(kST));  
        }  
    }  
  
    if (TAGrecoManager::GetPar()->IncludeBM()) {  
        fpNtuTrackBm = new TAGdataDsc(new TABMntuTrack());  
        fActEvtReader->SetupBranch(fpNtuTrackBm);  
        if (fFlagMC) {  
            fpNtuMcBm = new TAGdataDsc(FootDataDscMcName(kBM), new TAMCntuHit());  
            fActEvtReader->SetupBranch(fpNtuMcBm, FootBranchMcName(kBM));  
        }  
    }  
    . . .  
}
```

➔ Read all containers (excluded all hits except for ST) for the included detectors

Global Analysis Class (iv)

□ Global analysis class: GlobalAna

```
//-----  
//! Create reconstruction actions  
void GlobalAna::CreateAnaAction()  
{  
    // place here your beloved analysis class  
    if ((TAGrecoManager::GetPar()->IncludeTOE() || TAGrecoManager::GetPar()->IncludeKalman())) {  
        if (fAnaManager->GetAnalysisPar().MassResoFlag)  
            fActGlbAna = new TANAactNtuMass("anaActMass", fpNtuGlbTrack, fpParGeoG);  
    }  
}  
//-----  
//! Add required reconstruction actions in list  
void GlobalAna::AddRequiredItem()  
{  
    // Add the required analysis class  
    if ((TAGrecoManager::GetPar()->IncludeTOE() || TAGrecoManager::GetPar()->IncludeKalman())) {  
        if (fAnaManager->GetAnalysisPar().MassResoFlag)  
            gTAGroot->AddRequiredItem("anaActMass");  
    }  
}
```

- ➔ Create and require the dedicated class analysis when flag is on in the analysis manager
- ➔ Can have more than one analysis module

Global Analysis executable

Global analysis main: DecoceGlbAnalysis

```
for (int i = 0; i < argc; i++){
    if(strcmp(argv[i],"-out") == 0) { out =TString(argv[++i]); } // Raw file name for output
    if(strcmp(argv[i],"-in") == 0) { in = TString(argv[++i]); } // Root file in input
    if(strcmp(argv[i],"-exp") == 0) { exp = TString(argv[++i]); } // extention for config/geomap files
    if(strcmp(argv[i],"-nev") == 0) { nTotEv = atoi(argv[++i]); } // Number of events to be analyzed
    if(strcmp(argv[i],"-nsk") == 0) { nSkipEv = atoi(argv[++i]); } // Number of events to be skip
    if(strcmp(argv[i],"-run") == 0) { runNb = atoi(argv[++i]); } // Run Number
    if(strcmp(argv[i],"-mc") == 0) { mc = true; } // reco from MC local reco data

    . . .
}

TApplication::CreateApplication();

TAGrecoManager::Instance(exp);
TAGrecoManager::GetPar()->FromFile();
TAGrecoManager::GetPar()->Print();

// check input file exists
if(in.IsNull() || gSystem->AccessPathName(in.Data())) {
    Error("main()", "Input file does not exist or is null");
    exit(-1);
}

GlobalAna* glbAna = new GlobalAna(exp, runNb, in, out, mc);
. . .
}
```

➔ Copy of the DecodeGlb main, take same arguments

```
DecodeGlbAnalysis -in run4287_GlbS_70kEvts_Out.root -out MassAnalysis.root -exp GSI2021 -run 4287
```

End