

# HowTo (for reconstruction)

**Structure**

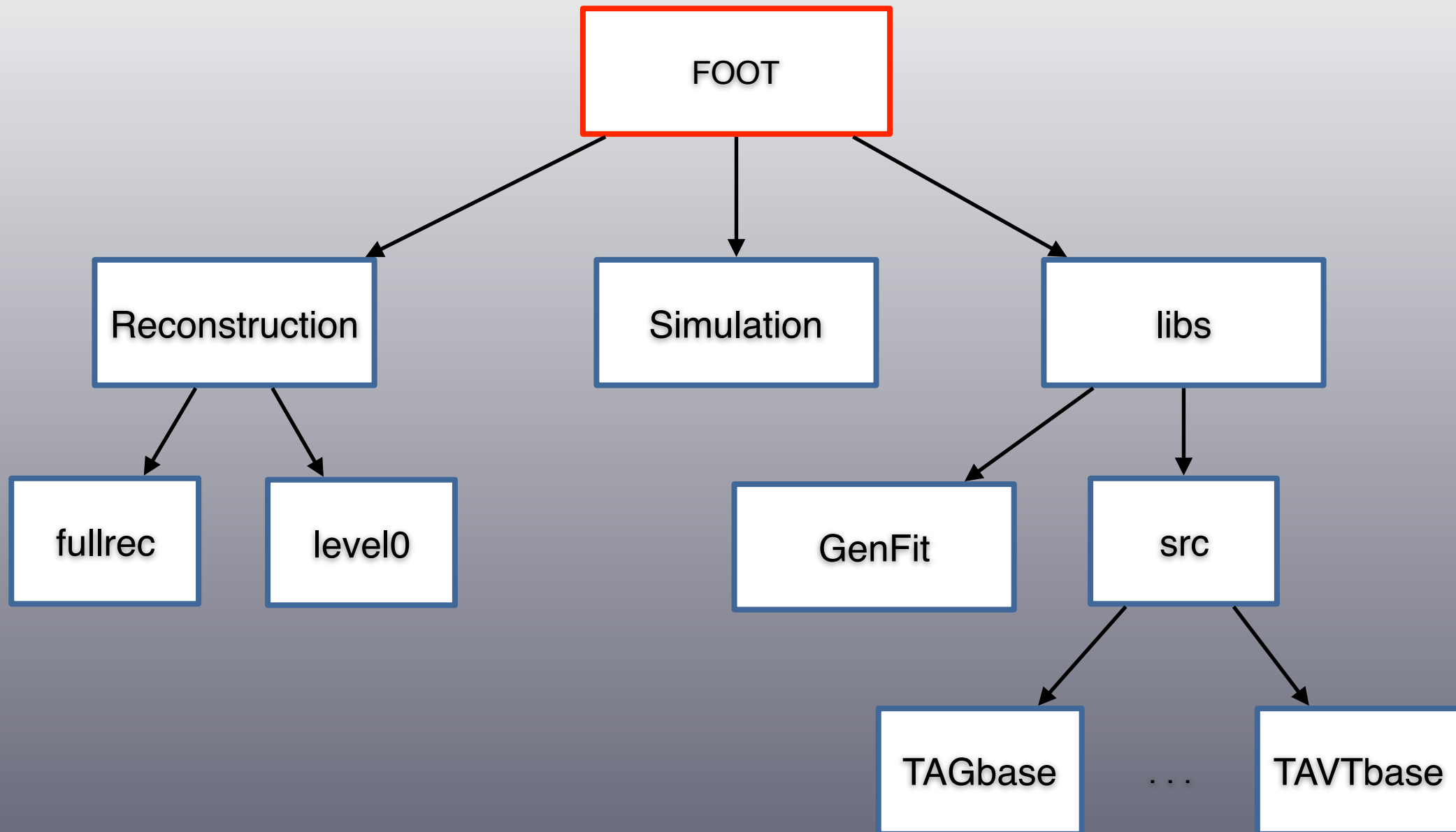
**CMake**

**Local Reconstruction**

**Conclusions**

**ToDo**

# Code Structure



# CMake (i)

## • Install:

- *mkdir* build
- *cd* build
- *cmake* pathToFootSrc -DCMAKE\_BUILD\_TYPE=Debug

## • Make:

- *make* in build (or any sub-directory)

## • Config:

- *source* setupFoot.(c)sh
- Macros, config and data files copied in the corresponding build folder
- rootlogon.C and setup files copied with the right path in build, remove any local myLogin.(c)sh files

# CMake (ii)

## Targets

```
FOOT:
--> Including cmake additional path: /Users/finck/Hadron/FOOT/trunk_cmake/libs/cmake/Modules/
--> Found ROOT 6.04/02 in /Users/finck/Root/root6
libs:
  genfit2:
  src:
  src/TAGbase:
  src/TATRbase:
  src/TAIRbase:
  src/TABMbase:
  src/TAVTbase:
  src/TAITbase:
  src/TAMSDbase:
  src/TAGmclib:
  src/TATWbase:
  src/TACAbase:
  src/TAGfoot:
Reconstruction:
  level0:
Simulation:
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/finck/Hadron/FOOT/build
[ 32%] Built target genfit2
[ 46%] Built target TAGbase
[ 50%] Built target TAGmclib
[ 52%] Built target TAIRbase
[ 54%] Built target TATRbase
[ 59%] Built target TABMbase
[ 73%] Built target TAVTbase
[ 80%] Built target TAITbase
[ 85%] Built target TAMSDbase
[ 90%] Built target TATWbase
[ 93%] Built target TACAbase
[ 97%] Built target TAGfoot
[ 98%] Built target RecoTools
[ 98%] Built target DecodeMC
[ 98%] Built target Txt2Root
[100%] Built target makeGeo
```

➔ Compile all targets at once with all dependencies

# MC Local reconstruction (i)

## • Run local MC with Macro for Vertex

```
void FillMCVertex(EVENT_STRUCT *myStr) {  
  
    /*Ntupling the MC Vertex information*/  
    TAGparaDsc* vtGeo      = new TAGparaDsc(TAVTparGeo::GetDefParaName(), new TAVTparGeo());  
    TAVTparGeo* geomap    = (TAVTparGeo*) vtGeo->Object();           // Geo parameters  
    geomap->InitGeo();  
  
    TAGdataDsc* vtRaw      = new TAGdataDsc("vtRaw", new TAVTntuRaw());  
    TAGdataDsc* vtClus     = new TAGdataDsc("vtClus", new TAVTntuCluster());  
    TAGdataDsc* vtTrck     = new TAGdataDsc("vtTrck", new TAVTntuTrack());           // Containers  
    TAGdataDsc* vtVtx      = new TAGdataDsc("vtVtx", new TAVTntuVertex());  
  
    TAGparaDsc* vtConf     = new TAGparaDsc("vtConf", new TAVTparConf());  
    TAVTparConf* parconf   = (TAVTparConf*) vtConf->Object();       // Config parameters  
    parconf->FromFile("./config/TAVTdetector.cfg");  
  
    TAVTparConf::SetHistoMap();  
    vtActRaw = new TAVTactNtuMC("vtActRaw", vtRaw, vtGeo, myStr);  
    vtActRaw->CreateHistogram();  
  
    vtActClus = new TAVTactNtuClusterF("vtActCluster", vtRaw, vtClus, vtConf, vtGeo);  
    vtActClus->CreateHistogram();  
  
    vtActTrck = new TAVTactNtuTrackF("vtActTrck", vtClus, vtTrck, vtConf, vtGeo); // Actions  
    vtActTrck->CreateHistogram();  
  
    vtActVtx = new TAVTactNtuVertex("vtActVtx", vtTrck, vtVtx, vtConf, vtGeo);  
    vtActVtx->CreateHistogram();  
  
    outFile->SetupElementBranch(vtRaw, TAVTntuRaw::GetBranchName());  
    outFile->SetupElementBranch(vtClus, TAVTntuCluster::GetBranchName()); // set branch in tree  
    outFile->SetupElementBranch(vtTrck, TAVTntuTrack::GetBranchName());  
}
```

# MC Local reconstruction (ii)

```
void ReadVtxRawMC(TString name = "12C_400_vtx.root")
{
    TGeoVolume* top = gGeoManager->MakeBox("TOPPER", gGeoManager->GetMedium("AIR"), 25., 25., 120.);
    gGeoManager->SetTopVolume(top);
    GlobalPar::Instance();
    Materials* listMaterials = new Materials() ;
    TAGroot tagr;
    EVENT_STRUCT evStr;
    TFile* f = new TFile(name.Data());
    TTree* tree = (TTree*)gDirectory->Get("EventTree"); // Declaration
    Evento *ev = new Evento();
    ev->FindBranches(tree,&evStr);
    outFile = new TAGactTreeWriter("outFile");

    FillMCVertex(&evStr);

    tagr.AddRequiredItem("vtActCluster");
    tagr.AddRequiredItem("vtActTrck");
    tagr.AddRequiredItem("vtActVtx");
    tagr.AddRequiredItem("outFile");
    tagr.Print();

    if (outFile->Open(nameOut.Data(), "RECREATE")) return; // tree file writer
    vtActRaw->SetHistogramDir(outFile->File());
    vtActClus->SetHistogramDir(outFile->File());
    vtActTrck->SetHistogramDir(outFile->File()); // output of histograms
    vtActVtx->SetHistogramDir(outFile->File());

    tagr.BeginEventLoop();
    Long64_t nentries = tree->GetEntries();
    for (Long64_t ientry = 0; ientry < nentries; ientry++) { // loop
        tree->GetEntry(ientry);
        tagr.NextEvent();
    }
}
```



Needed for Booter class

// Declaration

// tree file writer

// output of histograms

// loop

# MC Local reconstruction (iii)

## Running ReadVtxRawMC.C:

```
-----  
| Welcome to ROOT 6.04/02                               http://root.cern.ch |  
|                                                         (c) 1995-2014, The ROOT Team |  
| Built for macosx64                                     |  
| From tag v6-04-02, 14 July 2015                       |  
| Try '.help', '.demo', '.license', '.credits', '.quit'/'.'q' |  
-----
```

```
*****  
Messages from rootlogon.C
```

```
*****  
Load libraries for level0
```

```
*****
```

- Add /Users/finck/Hadron/F00T/build/Reconstruction/level0/ to the macros path
- Add /Users/finck/Hadron/F00T/build/libs/lib/TABMbase/ to the include path
- Add /Users/finck/Hadron/F00T/build/libs/lib/TAMSDbase/ to the include path
- Add /Users/finck/Hadron/F00T/build/libs/lib/TAGfoot/ to the include path
- Add /Users/finck/Hadron/F00T/build/libs/lib/TAIRbase/ to the include path
- Add /Users/finck/Hadron/F00T/build/libs/lib/TATWbase/ to the include path
- Add /Users/finck/Hadron/F00T/build/libs/lib/TACAbase/ to the include path
- Add /Users/finck/Hadron/F00T/build/libs/lib/TAGbase/ to the include path
- Add /Users/finck/Hadron/F00T/build/libs/lib/TAGmclib/ to the include path
- Add /Users/finck/Hadron/F00T/build/libs/lib/TAITbase/ to the include path
- Add /Users/finck/Hadron/F00T/build/libs/lib/TAVTbase/ to the include path
- Add GenFit to the include path

```
root [0]
```

```
Processing ReadVtxRawMC.C+...
```

```
Info in <TMacOSXSystem::ACLic>: creating shared library /Users/finck/Hadron/F00T/build/Reconstruction/  
level0/ReadVtxRawMC_C.so
```

```
Info in <TGeoManager::TGeoManager>: Geometry genfitGeom, GENFIT geometry created
```

```
Info in <TGeoManager::SetTopVolume>: Top volume is TOPPER. Master volume is TOPPER
```

```
Info in <TGeoNavigator::BuildCache>: --- Maximum geometry depth set to 100
```

# MC Local reconstruction (iv)

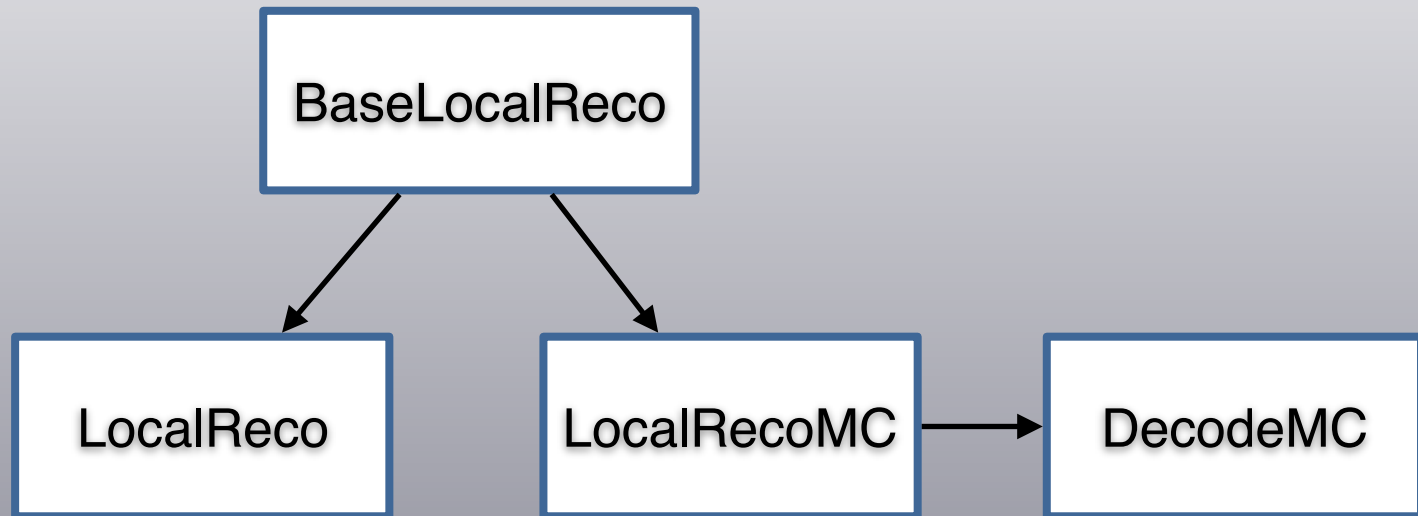
```
GeoTrafo action actGeoTrafo found
TAGroot:
Known Actions:
  name          type
  actGeoTrafo   TAGgeoTrafo
  outFile       TAGactTreeWriter
  vtActRaw      TAVTactNtuMC
  vtActCluster  TAVTactNtuClusterF
  vtActTrck     TAVTactNtuTrackF
  vtActVtx      TAVTactNtuVertex
Known ParaDsc's:
  name          type
  vtGeo         TAVTparGeo
  vtConf        TAVTparConf
Known DataDsc's:
  name          type          produced by
  vtRaw         TAVTntuRaw      vtActRaw
  vtClus        TAVTntuCluster  vtActCluster
  vtTrck        TAVTntuTrack    vtActTrck
  vtVtx         TAVTntuVertex   vtActVtx
Required Actions:
  vtActCluster  TAVTactNtuClusterF
  vtActTrck     TAVTactNtuTrackF
  vtActVtx      TAVTactNtuVertex
  outFile       TAGactTreeWriter
Required DataDsc's:
  Beginning the Event Loop
Running against ntuple with:: 10000 entries!
Loaded Event:: 0
Loaded Event:: 100
Loaded Event:: 200
Loaded Event:: 300
Loaded Event:: 400
Loaded Event:: 500
TAGactTreeWriter: TAGactTreeWriter 'outFile'
branch name          index/type  bt bid  tot size file size  comp  frac
tree:tree            -- tree ----  0.0k   0.0k   1.00
Real time 0:00:05, CP time 5.690
root [1]
```



# Local reconstruction

(new framework)

• Reconstruction/level0:



- The reconstruction actions are common to MC and real data
- Dedicated classes for raw actions MC and real data
- DecodeMC is available as macro or as compiled executable
- Move RecoL0 and ReoTools in fullrec folder (executable: DecodeGlb)

# MC Local reconstruction (i)

## BaseLocalReco (i)

```
class BaseLocalReco : public TNamed // using TNamed for the in/out files
{
public:
    ///! default constructor
    BaseLocalReco(TString fileNameIn, TString fileNameOut);

    virtual ~BaseLocalReco();

    ///! Create raw/rec action
    virtual void CreateRecAction();
    virtual void CreateRawAction() { return; }

    ///! Add raw/rec required items
    virtual void AddRawRequiredItem() { return; }
    virtual void AddRecRequiredItem();

    ///! Set raw/rec histogram directory
    virtual void SetRawHistogramDir() { return; }
    virtual void SetRecHistogramDir();

    ///! Loop events
    virtual void LoopEvent(Int_t /*nEvents*/) { return; }
    virtual void BeforeEventLoop();
    virtual void AfterEventLoop();

    ///! Open/Close File In/Out
    virtual void OpenFileIn() { return; }
    virtual void CloseFileIn() { return; }
    virtual void OpenFileOut();
    virtual void CloseFileOut();

    ///! Create branch in tree
    virtual void SetTreeBranches();
};
```

# MC Local reconstruction (ii)

## → BaseLocalReco (ii)

```
#!/ Enable detectors
void EnableTr() { fFlagTr = true; }
void EnableBm() { fFlagBm = true; }
void EnableVtx() { fFlagVtx = true; }
void EnableIt() { fFlagIt = true; }
void EnableMsd() { fFlagMsd = true; }
void EnableTw() { fFlagTw = true; }
void EnableCa() { fFlagCa = true; }

// Enable global
void EnableTree() { fFlagTree = true; }
void DisableTree(){ fFlagTree = false; }

void EnableHisto() { fFlagHisto = true; }
void DisableHisto() { fFlagHisto = false; }

void EnableVtxTrack() { fVtxTrackFlag = true; }
void DisableVtxTrack() { fVtxTrackFlag = false; }
};
```

# MC Local reconstruction (iii)

## LocalRecoMC

```
class LocalRecoMC : public BaseLocalReco
{
public:
    ///! default constructor
    LocalRecoMC(TString fileNameIn, TString fileNameout);

    virtual ~LocalRecoMC();

    ///! Add required items
    virtual void AddRawRequiredItem();

    ///! Create raw data action
    virtual void CreateRawAction();

    ///! Set raw histogram directory
    virtual void SetRawHistogramDir();

    ///! Open File
    virtual void OpenFileIn();

    ///! Close File in
    virtual void CloseFileIn();

    ///! Loop events
    virtual void LoopEvent(Int_t nEvents);
};
```

- Implemented the MC specific methods (in virtual in base class)

# MC Local reconstruction (iv)

## Macro: DecodeMC

```
void DecodeMC(TString name = "12C_400_vtx.root")
{
  GlobalPar::Instance();
  GlobalPar::GetPar()->Print();

  Int_t pos = name.Last('.');
  TString nameOut = name(0, pos);
  nameOut.Append("_Out.root");

  LocalRecoMC* locRec = new LocalRecoMC(name, nameOut);

  // global setting
  //locRec->EnableTree();
  locRec->EnableHisto();
  // set detecteors
  locRec->EnableVtx();
  locRec->EnableVtxTrack();
  locRec->EnableIt();
  locRec->EnableMsd();

  TStopwatch watch;
  watch.Start();

  locRec->BeforeEventLoop();
  locRec->LoopEvent(1);
  locRec->AfterEventLoop();

  watch.Print();
}
```

- For the moment only VTX-IT-MSD implemented
- Need digitizer for BM, TW and CAL (template for TW and CAL)

# MC Local reconstruction (v)

## Executable: DecodeMC (i)

```
int main (int argc, char *argv[])
{
    TString in("12C_400_vtx.root");
    Int_t pos = in.Last('.');
    TString out = in(0, pos);
    out.Append("_Out.root");

    Bool_t st = false;
    Bool_t bm = false;
    Bool_t vt = false;
    Bool_t it = false;
    Bool_t msd = false;
    Bool_t tof = false;
    Bool_t cal = false;
    Bool_t ntu = false;
    Bool_t his = false;
    Int_t nTotEv = 500;

    for (int i = 0; i < argc; i++){
        if(strcmp(argv[i], "-out") == 0) { out =TString(argv[++i]); } // Root file name for output
        if(strcmp(argv[i], "-in") == 0) { in = TString(argv[++i]); } // Root file in input
        if(strcmp(argv[i], "-nev") == 0) { nTotEv = atoi(argv[++i]); } // Number of events
        if(strcmp(argv[i], "-st") == 0) { st = true; } // enable start counter
        if(strcmp(argv[i], "-bm") == 0) { bm = true; } // enable beam monitor
        if(strcmp(argv[i], "-vt") == 0) { vt = true; } // enable vertex
        if(strcmp(argv[i], "-it") == 0) { it = true; } // enable inner tracker
        if(strcmp(argv[i], "-msd") == 0) { msd = true; } // enable multi strip detector
        if(strcmp(argv[i], "-tof") == 0) { tof = true; } // enable TOF
        if(strcmp(argv[i], "-cal") == 0) { cal = true; } // enable calorimeter
        if(strcmp(argv[i], "-ntu") == 0) { ntu = true; } // enable tree filling
        if(strcmp(argv[i], "-his") == 0) { his = true; } // enable histogramming

        if(strcmp(argv[i], "-help") == 0) {
            ...
        }
    }
}
```

# MC Local reconstruction (vi)

## Executable: DecodeMC (ii)

```
TApplication::CreateApplication();

GlobalPar::Instance();
GlobalPar::GetPar()->Print();
LocalRecoMC* locRec = new LocalRecoMC(in, out);

// global setting
if (ntu)
    locRec->EnableTree();
if (his)
    locRec->EnableHisto();

// set detecteors
if (vt) {
    locRec->EnableVtx();
    locRec->EnableVtxTrack();
}
if (it)
    locRec->EnableIt();
if (msd)
    locRec->EnableMsd();
...
locRec->BeforeEventLoop();
locRec->LoopEvent(nTotEv);
locRec->AfterEventLoop();

return 0;
}
```

- DecodeMC -in mc.root -out mc\_out.root -vt -his -nev 1000

# MC Local reconstruction (vii)

## Running: DecodeMC (i)

```
DecodeMC -in 12C_400_vtx.root -out 12C_400_vtx_out.root -vt -nev 1000 -his
```

```
Info in <TGeoManager::TGeoManager>: Geometry FOOT, FOOT Geometry created
```

```
Warning in <TClass::Init>: no dictionary for class ROOT::TIOFeatures is available
```

```
No GeoTrafo action actGeoTrafo available yet
```

```
TAGroot:
```

```
Known Actions:
```

name	type
locRecFile	TAGactTreeWriter
vtActNtu	TAVTactNtuMC
vtActClus	TAVTactNtuClusterF
vtActTrack	TAVTactNtuTrackF
vtActVtx	TAVTactNtuVertexPD

```
Known ParaDsc's:
```

name	type
vtGeo	TAVTparGeo
vtConf	TAVTparConf

```
Known DataDsc's:
```

name	type	produced by
vtRaw	TAVTntuRaw	vtActNtu
vtClus	TAVTntuCluster	vtActClus
vtTrack	TAVTntuTrack	vtActTrack
vtVtx	TAVTntuVertex	vtActVtx

```
Required Actions:
```

vtActNtu	TAVTactNtuMC
locRecFile	TAGactTreeWriter
vtActClus	TAVTactNtuClusterF
vtActTrack	TAVTactNtuTrackF
vtActVtx	TAVTactNtuVertexPD



# MC Local reconstruction (viii)

## Running DecodeMC (ii)

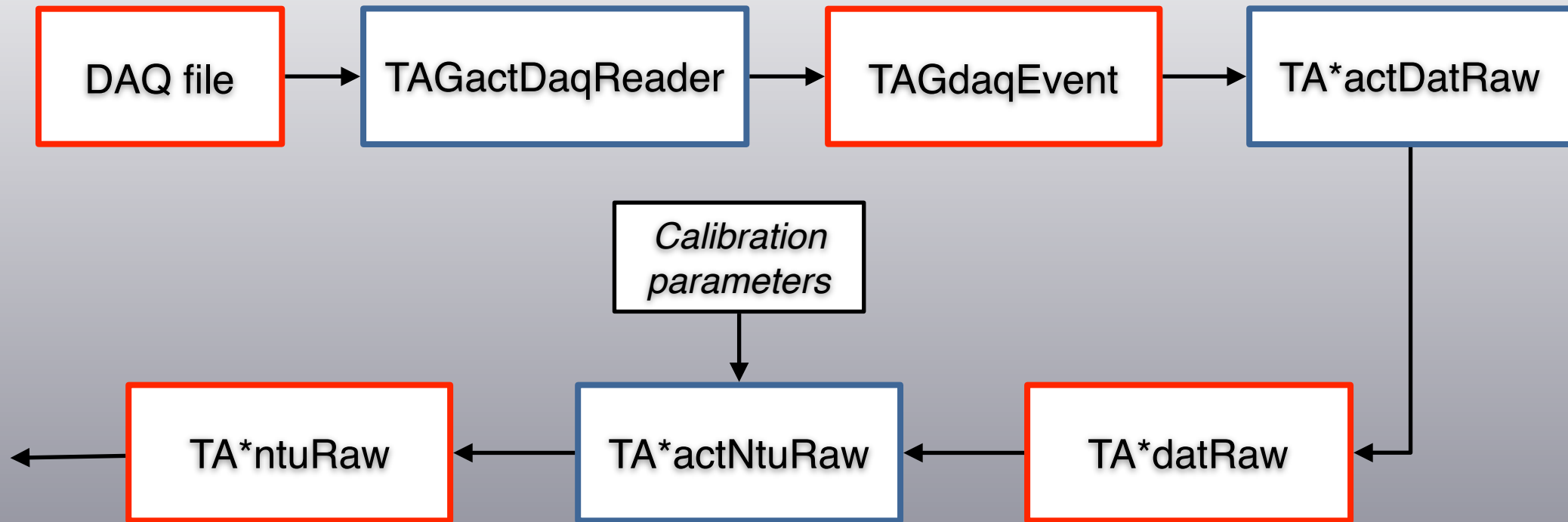
```
Loaded Event:: 0
Loaded Event:: 100
Loaded Event:: 200
Loaded Event:: 300
Loaded Event:: 400
Loaded Event:: 500
Loaded Event:: 600
Loaded Event:: 700
Loaded Event:: 800
Loaded Event:: 900
TAGactTreeWriter: TAGactTreeWriter 'locRecFile'
branch name          index/type  bt bid  tot size file size  comp  frac
tree:tree            -- tree ----  0.0k   0.0k  1.00
Real time 0:00:11, CP time 11.480
```

## Rootfile output:

```
root [1] .ls
TFile**          12C_400_vtx_out.root
TFile*           12C_400_vtx_out.root
KEY: TH1F        vtMcPixel1;1    Vertex - MC # pixels per clusters for sensor 1
KEY: TH1F        vtMcPixel2;1    Vertex - MC # pixels per clusters for sensor 2
KEY: TH1F        vtMcPixel3;1    Vertex - MC # pixels per clusters for sensor 3
KEY: TH1F        vtMcPixel4;1    Vertex - MC # pixels per clusters for sensor 4
KEY: TH1F        vtMcPixelTot;1  Vertex - MC # total pixels per clusters
...
KEY: TH1F        vtVtxPosZ;1     Vertex position at Z
KEY: TH2F        vtVtxPosXY;1    Vertex position at XY
```

# Rawdata Local reconstruction (i)

• Scheme:



- TAGactDaqReader: interface with DAQ in TAGdaq folder
- To pack/unpack DAQ event need package in TAGdaqApi
- TAGactDaqReader and TAGdaqEvent only template (compilable)
- TA\*actDatRaw and TA\*datRaw exist for some detectors (in not compilable status)

# Rawdata Local reconstruction (ii)

## LocalReco (i)

```
//  
void LocalReco::CreateRawAction()  
{  
    fpDaqEvent    = new TAGdataDsc("daqEvt", new TAGdaqEvent());  
    if (fFlagVtx) {  
        fpDatRawVtx    = new TAGdataDsc("vtDat", new TAVTdatRaw());  
        fpNtuRawVtx    = new TAGdataDsc("vtRaw", new TAVTntuRaw());  
        // skip dat raw and go directly to raw data ?  
        fActDatRawVtx = new TAVTactDaqRaw("vtAcDat", fpDatRawVtx, fpDaqEvent, fpParGeoVtx);  
        if (fFlagHisto)  
            fActDatRawVtx->CreateHisto();  
  
        fActNtuRawVtx = new TAVTactNtuRaw("vtActNtu", fpNtuRawVtx, fpDatRawVtx, fpParGeoVtx);  
        // or  
        fActNtuRawVtx = new TAVTactNtuRaw("vtActNtu", fpNtuRawVtx, fpDaqEvent, fpParGeoVtx);  
        if (fFlagHisto)  
            fActNtuRawVtx->CreateHisto();  
    }  
  
    if (fFlagIt) {  
        ...  
    }  
    ...  
}
```

# Rawdata Local reconstruction (iii)

## LocalReco (ii)

```
//  
void LocalReco::OpenFileIn()  
{  
    fActEvtReader = new TAGactDaqReader("daqAct");  
  
    fActEvtReader->SetupChannel(fpDaqEvent);  
    fActEvtReader->Open(GetName());  
}
```

- LocalReco is not compilable since dat raw containers and actions are not compilable (need implementation), not included in CMakeFiles.txt
- DAQ event and reader just templates
- Create TAGdaqApi to provide pack/unpack event library from DAQ people

# Conclusion

- Need inputs from DAQ/detectors people for Digitizer and raw data format
- Framework ready for local reconstruction (Reconstruction/level0)
- Framework for global reconstruction still working (Reconstruction/fullrec)

# ToDo (MC)

→ Digitizer: TABMdigitizer, TATWdigitizer and TACAdigitizer

```
class TACAntuRaw;
class TACAntuHit; // Template for CAL same for BM and TOF

// -----
class TACAdigitizer : public TAGbaseDigitizer { // need to inherit from base class
public:
    TACAdigitizer(TACAntuRaw* p_datraw);
    ~TACAdigitizer();
    ...
    // method to be implemented
    Bool_t      Process(Double_t edep, Double_t x0, Double_t y0, Double_t zin=0,
                       Double_t zout=0, Double_t time = 0, Int_t sensorId = 0);
    ...
    TACAntuHit* GetCurrentHit()      { return fCurrentHit; } // current hit
private:
    TACAntuRaw*  fpNtuRaw;
    ...
    TACAntuHit*  fCurrentHit;
};
```

# ToDo (MC)

Digitizer: TABMdigitizer, TATWdigitizer and TACAdigitizer

```
class TACAntuRaw;  
class TACAntuHit; // Template for CAL same for BM and TOF  
  
// -----  
class TACAdigitizer : public TAGbaseDigitizer { // need to inherit from base class  
public:  
    TACAdigitizer(TACAntuRaw* p_datraw);  
    ~TACAdigitizer();  
    ...  
    // method to be implemented  
    Bool_t Process(Double_t edep, Double_t x0, Double_t y0, Double_t zin=0,  
                  Double_t zout=0, Double_t time=0, Int_t sensorId=0) = 0;  
    ...  
    TACAntuHit* GetHit();  
private:  
    TACAntuRaw* fData;  
    ...  
    TACAntuHit* fCurrentHit;  
};
```

```
class TAGbaseDigitizer : public TAGObject {  
public:  
    TAGbaseDigitizer();  
    virtual ~TAGbaseDigitizer();  
    // purely virtual method  
    virtual Bool_t Process( Double_t edep, Double_t x0, Double_t y0, Double_t zin = 0,  
                           Double_t zout = 0, Double_t time = 0, Int_t sensorId = 0) = 0;  
  
    ClassDef(TAGbaseDigitizer,0)  
};
```

# ToDo (MC)

Digitizer: TABMdigitizer, TATWdigitizer and TACAdigitizer

```
class TACAntuRaw;
class TACAntuHit; // Template for CAL same for BM and TOF

// -----
class TACAdigitizer : public TAGbaseDigitizer { // need to inherit from base class
public:
    TACAdigitizer(TACAntuRaw* p_datraw);
    ~TACAdigitizer();
    ...
    // method to be implemented
    Bool_t Process(Double_t edep, Double_t x0, Double_t y0, Double_t zin=0,
                  Double_t zout=0, Double_t time=0, Int_t sensorId=0) = 0;
    ...
    TACAntuHit* GetHit() const;
private:
    TACAntuRaw* fData;
    ...
    TACAntuHit* fCurrentHit;
};

class TAGbaseDigitizer : public TAGObject {
public:
    TAGbaseDigitizer();
    virtual ~TAGbaseDigitizer();
    // purely virtual method
    virtual Bool_t Process( Double_t edep, Double_t x0, Double_t y0, Double_t zin = 0,
                          Double_t zout = 0, Double_t time = 0, Int_t sensorId = 0) = 0;

    ClassDef(TAGbaseDigitizer,0)
};
```

- Some work done for BM
- First version for CAL and TOF exists



# ToDo (rawdata)

## • TAGdaqEvent

```
/*-----*/
/* TEMPLATE CLASS
   Need real implementation
*/
...
class TAGdaqEvent : public TAGdata {
public:
    TAGdaqEvent();
    virtual      ~TAGdaqEvent();

    Int_t        NSubEvent()          const { return (Int_t)fOffset.size(); }
    Int_t        SubEventType(Int_t i_ind) const { return (fData[fOffset[i_ind]+1]>>16) & 0xffff;; }
    Int_t        SubEventSubType(Int_t i_ind) const { return fData[fOffset[i_ind]+1] & 0xffff;; }
    Int_t        SubEventProcId(Int_t i_ind) const { return (fData[fOffset[i_ind]+2]>>16) & 0xffff;; }
    Int_t        SubEventProcType(Int_t i_ind) const { return fData[fOffset[i_ind]+2] & 0xff;; }
    Int_t        SubEventCrate(Int_t i_ind) const { return fData[fOffset[i_ind]+2] & 0xff; }
    Int_t        SubEventSize(Int_t i_ind) const { return fData[fOffset[i_ind]]; }
    const UInt_t* SubEventData(Int_t i_ind) const { return &fData[0] + fOffset[i_ind]+3; }
    virtual Bool_t NeedAutoDelete() const;
    virtual void   Clear(Option_t* opt="");
    virtual void   ToStream(ostream& os = cout, Option_t* option = "") const;
private:
    void          SetupOffset();
private:
    // NOTE: CUSTOM STREAMER
    vector<UInt_t> fData;          // data vector (common for all SE's)
    vector<Int_t>  fOffset;       // offset table for SE's
    ClassDef(TAGdaqEvent,2)
};
```

- Define format with DAQ/detector people

# ToDo

• Clustering classes needed:

- TATWntuCluster, TACAantuCluster with TATWactNtuCluster, TACAactNtuCluster
- Need to reformat cluster definition for MSD