Local Reconstruction

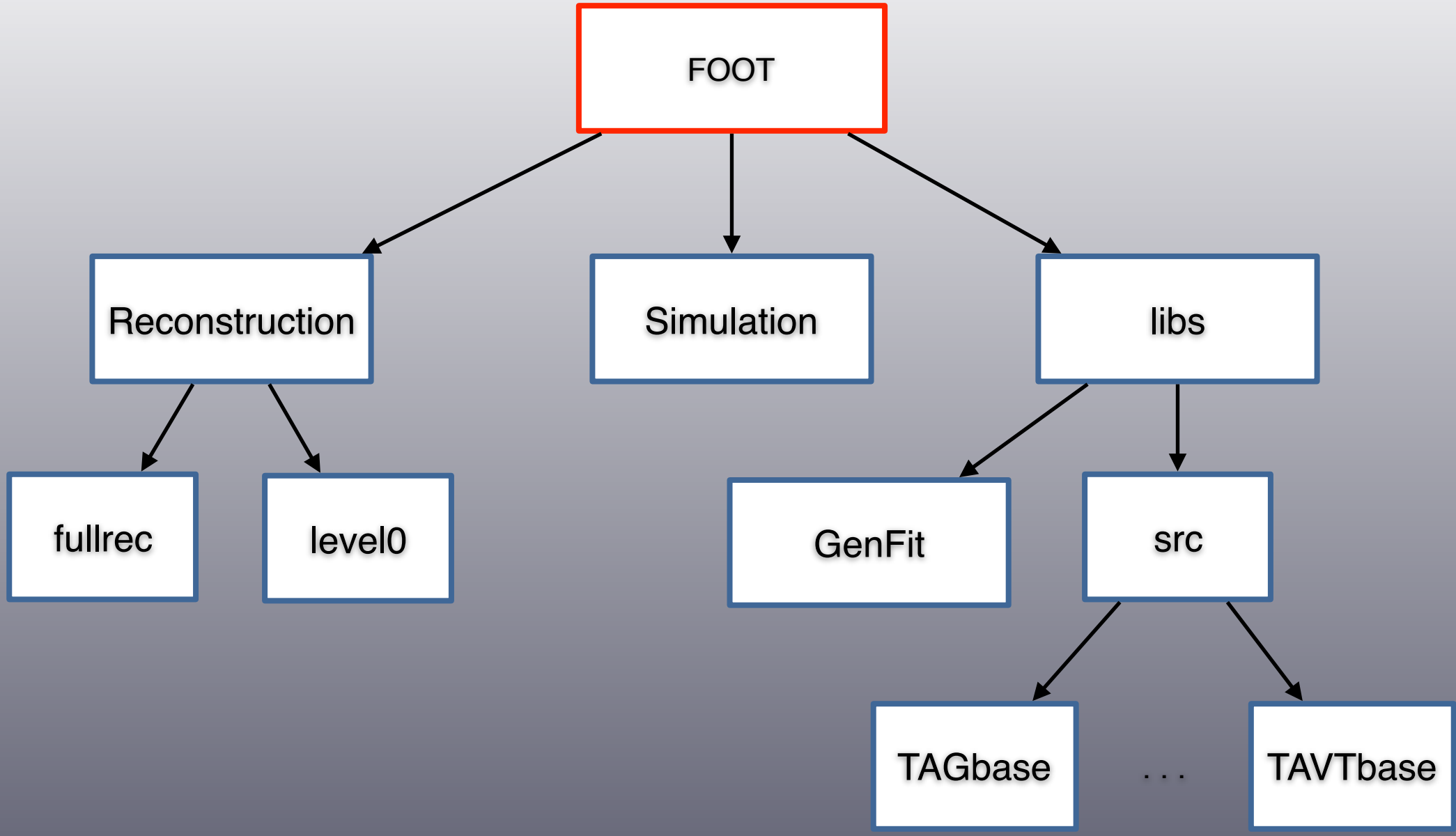
Structure

Local Reconstruction

Conclusions

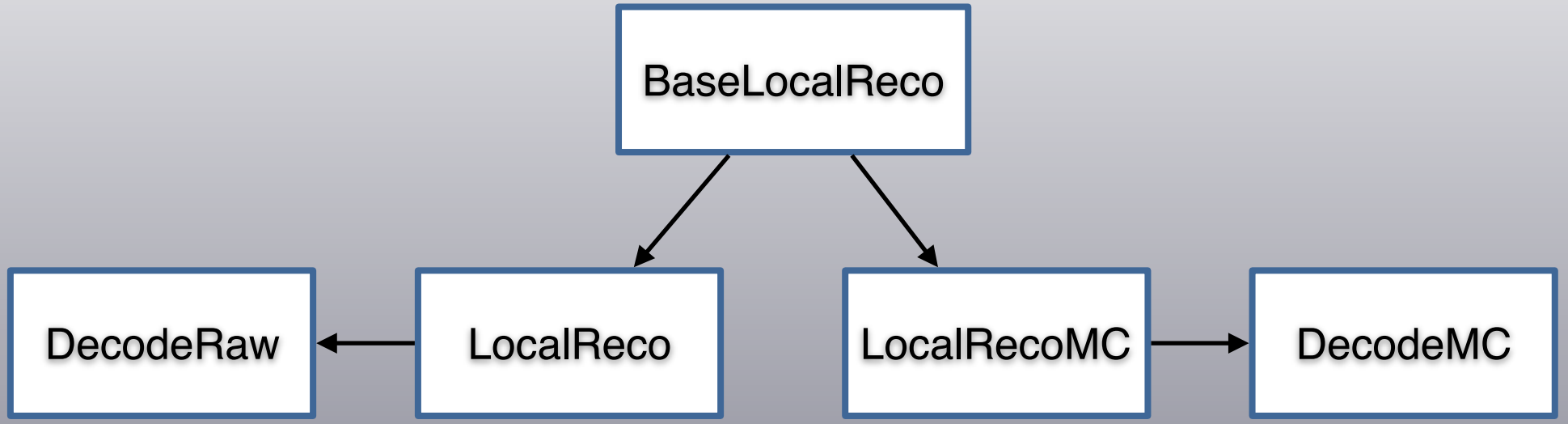
Outlook

Code Structure



Local reconstruction
(new framework)

- Reconstruction/level0:



- The reconstruction actions are common to MC and real data
- Dedicated classes for actions MC and real data
- DecodeMC is available as macro or as compiled executable
- DecodeRaw not available for the moment

Local reconstruction

- 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 CreateRawAction()            { return; }
   virtual void CreateRecAction();

   //! 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 (i)

- 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 (ii)

- 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

MC Local reconstruction (iii)

- 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 (iv)

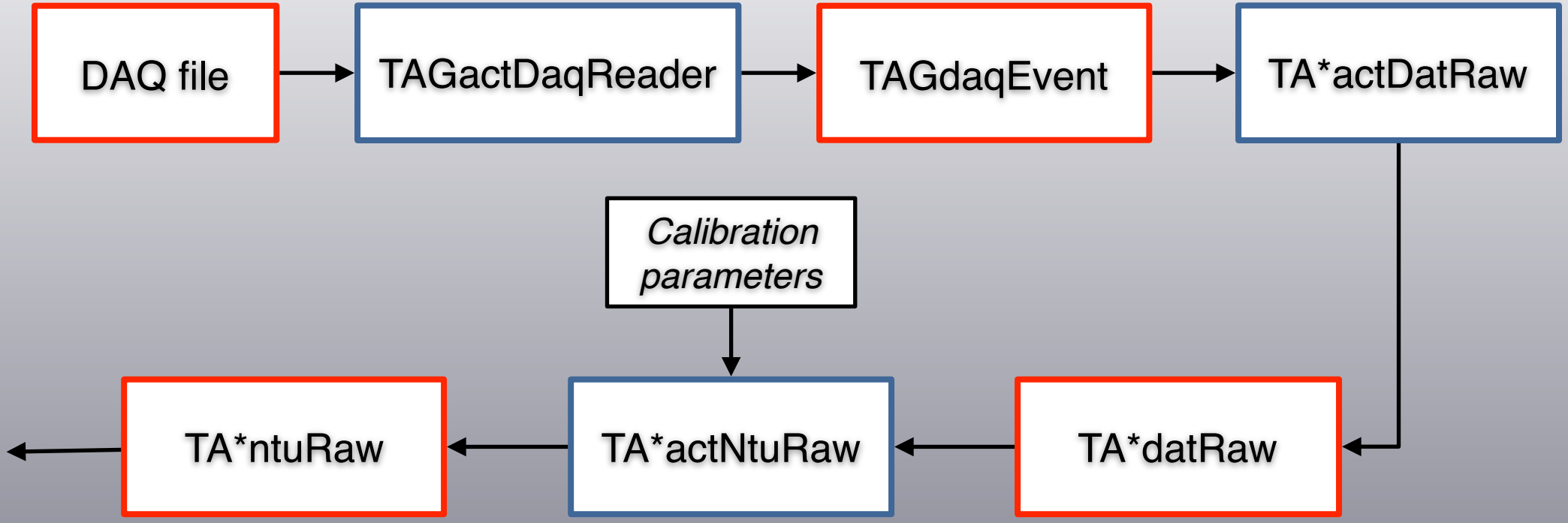
- Executable: DecodeMC (ii)

```
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

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) {
     …
   }
  …
}
//_____________________________________________________________
void LocalReco::OpenFileIn()
{
   fActEvtReader = new TAGactDaqReader("daqAct");

   fActEvtReader->SetupChannel(fpDaqEvent);
   fActEvtReader->Open(GetName());
}
```

Conclusion

- Need inputs from DAQ/detectors people for raw data format
- Framework ready for local reconstruction (Reconstruction/level0)

Outlook (i)

- 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:
   vector<UInt_t>  fData;           // data vector (common for all SE's)
   vector<Int_t>   fOffset;         // offset table for SE's
   ClassDef(TAGdaqEvent,2)
};
```

Outlook (ii)

- TAGactDaqReader

```
/*-----------------------------------------+-------------------------------*/
/* TEMPLATE CLASS
   Need real implementation
*/

#include "TAGactionFile.hxx"

//##############################################################################

class TAGactDaqReader : public TAGactionFile {
  public:
    explicit        TAGactDaqReader(const char* name=0);
    virtual         ~TAGactDaqReader();

    void            SetupChannel(TAGdataDsc* p_data);

    virtual Int_t   Open(const TString& name, Option_t* option=0);
    virtual void    Close();
    virtual Bool_t  IsOpen() const;
    virtual Bool_t  Process();

  private:
…
    ClassDef(TAGactDaqReader,0)
};
```

- Define format with DAQ/detector people