RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

HLT – Status

Calibration

Jochen Thäder
Kirchhoff Institut für Physik
Ruprecht-Karls-Universität Heidelberg

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Status

- Calibration Components for
  - DiMuon
  - HLT
  - PHOS
  - TPC
  - TRD

- Base Processing Class:
  `AliHLTCalibrationProcessor`

- „HLT →FXS → Shuttle“ HLT Component
  `FXSSubscriberComponent`

→ Input

- **Input**
  - **Raw data**
    - **Coming over DDLs**
  - **Reconstructed data (cluster, tracks)**
    - **Created in HLT**
  - **Runtime parameters**
    - **Coming from DCS (live update via pendelino interface)**
  - **HCDB**
    - **Direct copy of OCDB (update at start of run via taxi interface)**

... Processing ...

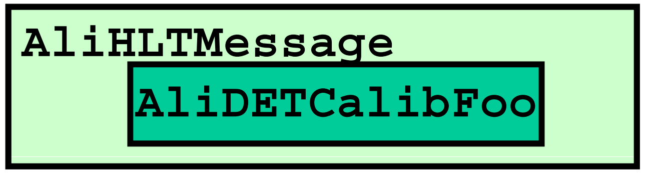
- **Processing**
  - **In dedicated HLT Calibration Components**
  - **Inherit from base Class**

    `AliHLTCalibrationProcessor`

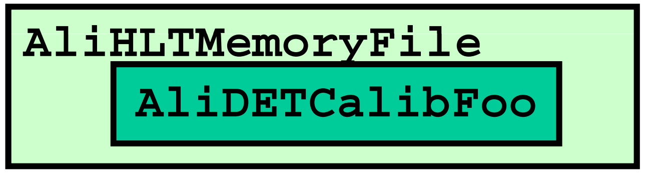
  - **Distinguish between two modes**
    - **Processing**
    - **Ship Data to FXS**

Output → (1)

- Output
  - Any selfdefined structures
    - Detector exprts have to take care to rootify data in Preprocessor / Monitoring
  - ROOT TObjects
    - Send as AliHLTMessage inside HLT



    - Send as AliHLTMemoryFile in Memory to FXSSubscriber



Output → (2)

- Where <u>can</u> output go to?
  - FXS ->Shuttle -> Preprocessor -> OCDB
    - Written as data blob (binary / Root file)
      - AliHLTMemoryFile
  - TCP–port -> Monitoring / Visualisation
    - HOMER readable format
      - AliHLTMessage
      - Any selfdefined structures
  - DAQ (via HLT Output data) -> Storage / DQM
    - Inside the HLT output Block
    - HOMER readable format
      - AliHLTMessage
      - Any selfdefined structures

ROOT addons !

- **AliHLTMessage**
  - **Custom ROOT TMessage (same behavoir)**
  - **TObject „wrapped“ inside**
    - **Used between the nodes**
    - **In HOMER datastructures**

- **AliHLTMemoryFile**
  - **Custom ROOT TFile (same behavoir)**
  - **ROOT File „written“ to Memory**

    (→ HLT Components never write to d isk)
  - **Used to send Calibration TObjects from proccessing node to FXS node**

`AliHLTCalibrationProcessor`

- **All Calibration components inherit from**
  `AliHLTCalibrationProcessor`
- **Takes care of necessary formating/headers for sending to FXS**
- **2 Main processing „user“ functions**
  - **`ProcessCalibration()`**
    - **Processes data on event basis**
    - **Fills detector calibration objects**
  - **`ShipDataToFXS()`**
    - **Called on END_OF_RUN**
    - **Called on “-eventmodulo X“ -> send every X event**
    - **Can perform additional analysis**
    - **Sends Calibration objects to FXS**

→ to HLT FXS

- **Has been installed / tested**
  - **Data is shipped from HLT Chain to FXS in testmode**

- **Two Push Functions for**
  - **TObjects**

```
Int_t AliHLTCalibrationProcessor::PushToFXS(
TObject* pObject, const char* pDetector,
const char* pFileID, const char* pDDLNumber = "");
```

  - **Selfdefined structures**

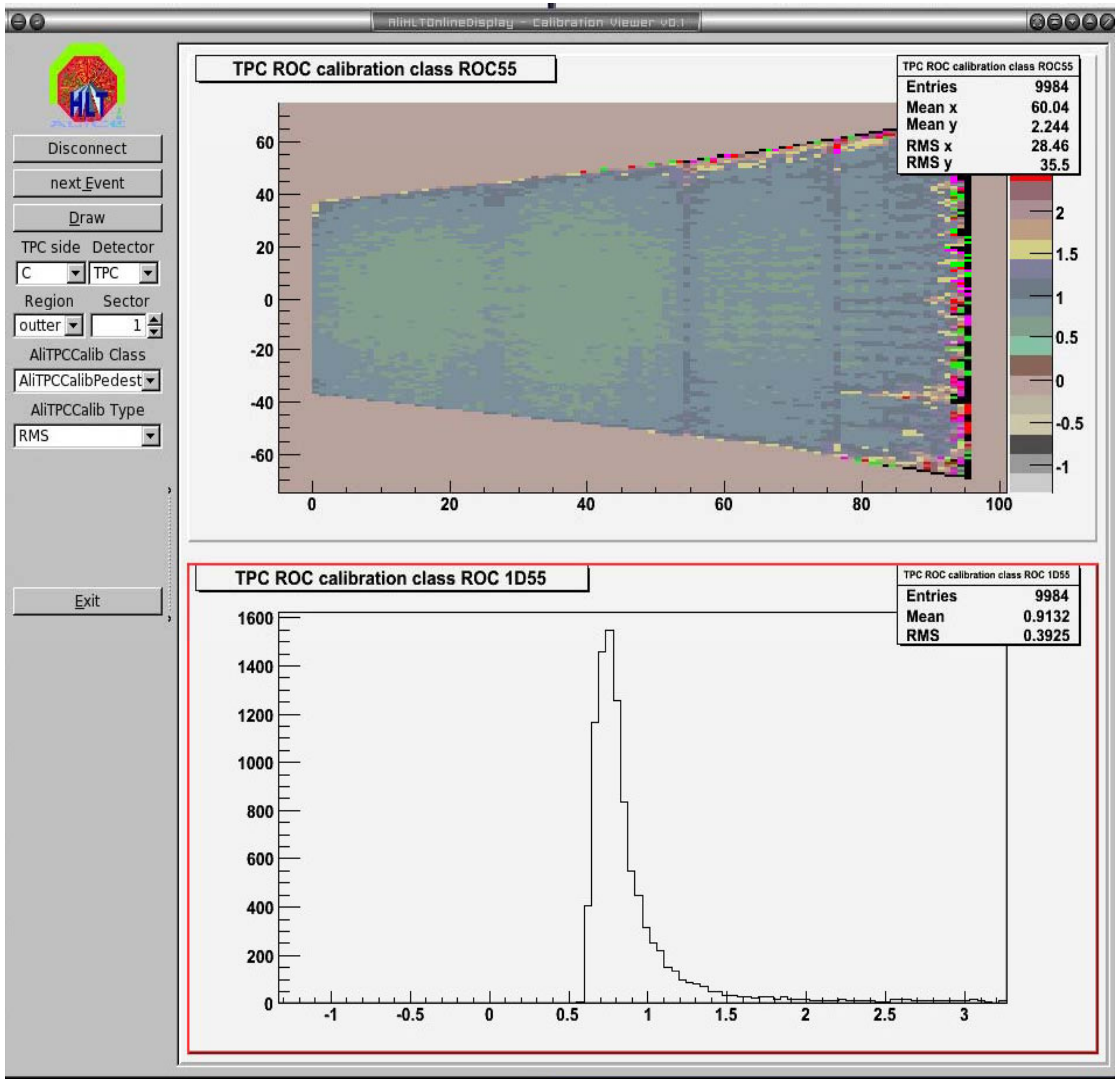
```
Int_t AliHLTCalibrationProcessor::PushToFXS(
void* pBuffer, int iSize, const char* pDetector,
const char* pFileID, const char* pDDLNumber = "");
```

TPC - Calibration

- **HLT Components were tested :**
  - **In HLT online Framework**
    - **Data from TPC Comissioning 2006**
    - **Was run on final FEPs**
    - **More than 1 million events processed**
  - **During TPC Commissioning in June 2007**
    - **Signal Calibration**
    - **Pedestal Calibration**
    - **All 8 Sectors (0,1,3,4,9,10,12,13)**
    - **HLT Calibration Viewer**

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

HLT – Calibration Viewer



TPC side: C
Detector: TPC
Region: outter
Sector: 1
AliTPCCalib Class: AliTPCCalibPedest
AliTPCCalib Type: RMS

TPC ROC calibration class ROC55

Entries	9984
Mean x	60.04
Mean y	2.244
RMS x	28.46
RMS y	35.5

TPC ROC calibration class ROC 1D55

Entries	9984
Mean	0.9132
RMS	0.3925

Outlook

- Further Tests with Calibration Compents
- Testing of various sub detector calibrations
- Finializing live DCS data in calibration

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

CalibrationComponent

- **Components have to implement:**
  - `ProcessCalibration()`
  - `ShipDataToFXS()`
  - `InitCalibration()` **(optional)**
  - `DeinitCalibration()` **(optional)**
  - `ScanArgument()` **(optional)**
  - **Normal HLT Steer Component functions**

Is it difficult? ...... NO !!

Remember: If questions appear... HLT core is always willing to help!!

CalibrationComponent – Details (1)

- **Initialization at beginning of run**
  - **Invoke Worker class(es)**
  - `InitCalibration()`

- **Deinitialization after run**
  - **Cleanup before leave !**
  - `DeinitCalibration()`

- **Read in Component arguments**
  - `ScanArgument()`

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Example using the TPC (Init / Deinit)

```
Int_t AliHLTTPCCalibPedestalComponent::InitCalibration()  {
  // see header file for class documentation

  // ** Create pedestal calibration
  if ( fCalibPedestal )
    return EINPROGRESS;

  fCalibPedestal = new AliTPCCalibPedestal();
  ...

  return 0;
}

Int_t AliHLTTPCCalibPedestalComponent::DeinitCalibration() {
  // see header file for class documentation

  if ( fCalibPedestal )
    delete fCalibPedestal;
  fCalibPedestal = NULL;
  ...

  return 0;
}
```

CalibrationComponent – Details (2)

- **Process Calibration**
  - Check input data
  - Process the input data
  - Fill histograms
  - Push data to TCP-port / DAQ
- **Ship data to FXS**
  - Call additional “analyze“ functions
  - Push data to FXS

Example using the TPC (Process)

```
Int_t AliHLTTPCCalibPedestalComponent::ProcessCalibration( const AliHLTComponentEventData& evtData,
                                                          AliHLTComponentTriggerData& trigData ) {
  // see header file for class documentation

  ...

  iter = GetFirstInputBlock( AliHLTTPCDefinitions::fgkDDLPackedRawDataType );

  while ( iter != NULL ) {
    ...

    // ** Init TPCRawStream
    fRawReader->SetMemory( reinterpret_cast<UChar_t*>( iter->fPtr ), iter->fSize );
    fRawReader->SetEquipmentID(DDLid);

    fRawStream = new AliTPCRawStream( fRawReader );
    fRawStream->SetOldRCUFormat( fRCUFormat );

    // ** Process actual Pedestal Calibration - Fill histograms
    fCalibPedestal->ProcessEvent( fRawStream );

    // ** Delete TPCRawStream

    iter = GetNextInputBlock();

  } //  while ( iter != NULL ) {

  // ** Get output specification
  fSpecification = AliHLTTPCDefinitions::EncodeDataSpecification( slice, slice, fMinPatch, fMaxPatch );

  // ** PushBack data to shared memory ...
  PushBack( (TObject*) fCalibPedestal,
    AliHLTTPCDefinitions::fgkCalibPedestalDataType, fSpecification);

  return 0;
}
```

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Example using the TPC (Ship)

```
Int_t
  AliHLTTPCCalibPedestalComponent::ShipDataToFXS(
  const AliHLTComponentEventData& evtData,
  AliHLTComponentTriggerData& trigData ) {
  // see header file for class documentation

  if ( fEnableAnalysis )
     fCalibPedestal->Analyse();

  // ** PushBack data to FXS
 PushToFXS( (TObject*) fCalibPedestal, "TPC",
  "pedestals"/*, DDLNumber optional*/ );

  return 0;
}
```