

Adding Analysis Task

Wednesday, July 29

1

1st R3BRoot Development Workshop
July 28 - 30, 2015
GSI, Darmstadt



- Create analysis task for calculation of time calibration parameters of TOF detector
- input: TOF raw items (TDC distribution)
- output: TDC \rightarrow time [ns] conversion parameters



Create class

- r3broot/tof
- Choose class name: R3B + Detector + Task
 - ➔ R3BTofFillTcal
- Create header (.h) and source (.cxx) files — names are identical to class name — in the detector sub-folder
- Class inherits from FairTask
 - ➔ #include "FairTask.h"
 - ➔ ...
 - ➔ class R3BTofFillTcal : public FairTask {
 - ➔ };



Header guard

- `#ifndef _R3BTOF_FILLTCAL_`
- `#define _R3BTOF_FILLTCAL_`
- ...
- ...
- ...
- `#endif`



Header file

- Default constructor `R3BTofFillTcal();`
- Standard constructor `R3BTofFillTcal(const char* taskName, const Int_t verbose);`
- virtual destructor: `virtual ~R3BTofFillTcal();`
- Overridden functions (public)
 - ➔ `InitStatus Init(); // Initialisation`
 - ➔ `void Exec(Option_t *option); // Implementation of event loop`
 - ➔ `void FinishEvent(); // End of each event`
 - ➔ `void FinishTask(); // End of task`



Data members

- Forward declaration (later add #include to source file)
 - ➔ class TClonesArray;
 - ➔ class TH1F;
- ...
- ...
- private:
 - ➔ Input / output data
 - TClonesArray* fRawData;
 - ➔ Additional data members
 - TH1F *fh_tdc[16];
 - Int_t fNevents;



Needed for CINT dictionary

- Header file: `ClassDef(R3BTofFillTcal, 1)`
- Source file: `ClassImp(R3BTofFillTcal)`



Implement constructor

- `#include "R3BTofFillTcal.h"`

- `// Default constructor`

```
R3BTofFillTcal::R3BTofFillTcal()
```

```
: FairTask("R3BTofFillTcal", 1),
```

```
  fRawData(NULL),
```

```
  fNevents(0)
```

```
{
```

```
}
```




Implement constructor

- // Standard constructor

```
R3BTofFillTcal::R3BTofFillTcal(const char *name, const Int_t verbose)
```

```
: FairTask(name, verbose),
```

```
  fRawData(NULL),
```

```
  fNevents(0)
```

```
{
```

```
}
```



Implement destructor

- // Destructor

```
R3BTofFillTcal::~~R3BTofFillTcal()
```

```
{
```

```
}
```



Implement Initialisation

- #include "FairRootManager.h"
- #include "TClonesArray.h"
- #include "TH1F.h"
- #include "TString.h"
- ...
- InitStatus R3BTofFillTcal::Init()
- {
 - ➔ FairRootManager *mgr = FairRootManager::Instance(); // singleton class
 - ➔ fRawData = (TClonesArray*) mgr->GetObject("TofRawItem");
 - ➔ if(! fRawData)
 - ➔ return kFATAL;
 - ➔ for(Int_t i = 0; i < 16; i++) {
 - ➔ fh_tdc[i] = new TH1F(Form("h_tdc_%d", i), Form("TDC distribution channel %d", i), 128, 0., 4096.);
 - ➔ }
 - ➔ return kSUCCESS;
- }
- // Add NULL checks
- kERROR kFATAL



Event loop implementation

- #include "R3BTofRawItem.h"
- ...
- void R3BTofFillTcal::Exec(Option_t *option)
- {
 - ➔ // Loop over input data
 - ➔ R3BTofRawItem *item;
 - ➔ Int_t nItems = fRawData->GetEntriesFast();
 - ➔ for(Int_t i = 0; i < nItems; i++)
 - ➔ {
 - item = (R3BTofRawItem*) fRawData->At(i);
 - if(NULL == item) { continue; }
 - fh_tdc[item->GetChannelId()]->Fill(item->GetTdc());
 - ➔ }
- }



Finish event

- `void R3BTofFillTcal::FinishEvent()`
- {
 ➔ `fNevents += 1;`
- }



Finish task

- #include "FairLogger.h"
- ...
- void R3BTofFillTcal::FinishTask()
- {
 - ➔ LOG(INFO) << "Number of entries: " << fh_tdc[15]->GetEntries() << FairLogger::endl;
 - ➔ for(Int_t i = 0; i < 16; i++) {
 - ➔ fh_tdc[i]->Write();
 - ➔ }
- }



Compilation

- TofLinkDef.h (for ROOT CINT):
`#pragma link C++ class R3BTofFillTcal+;`



Compilation (2)

- CMakeLists.txt:
 - ➔ 1. set(SRCS
...
R3BTofFillTcal.cxx)
 - ➔ 2. Check if additional include directories are needed
 - add them to set(INCLUDE_DIRECTORIES)
 - ➔ 3. Check if additional library dependencies are needed
 - add them to set(DEPENDENCIES)



Compile

- Recompile:
 - ➔ `cd "BUILD_DIR"`
 - ➔ `./config.sh`
 - ➔ `make -j4`



Execute task

- r3broot/macros/r3b/unpack/tof
- In the steering macro:

```
R3BTofFillTcal *tofFillTcal = new R3BTofFillTcal("TofFillTcal", 1);
```

```
run->AddTask(tofFillTcal);
```

```
...
```

```
run->Init();
```

```
run->Run(0, nEvents);
```