

Adding Data Classes

Tuesday, July 28

2

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

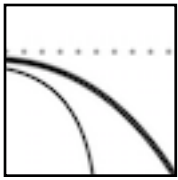


- Create a c++ class for storing the raw data items of TOF detector in output file
- and integrate it into TOF unpacker



Create class

- Choose location:
 - ➔ r3bdata/tofData
- Choose class name:
 - ➔ R3B + Detector + Data level: R3BTofRawItem
- Create header and source files (.h, .cxx, names are like class name)



Header file

- Add header guard
 - ➔ `#ifndef _R3BTOF_RAWITEM_`
 - ➔ `#define _R3BTOF_RAWITEM_`
 - ➔ ...
 - ➔ ...
 - ➔ `#endif`



Header file (2)

- Add class declaration

- ➔ #include "TObject.h"

- ➔ ...

- ➔ class R3BTofRawItem : public TObject {

- public:

- // Constructor / destructor

- private:

- // Data members

- public:

- // Public accessors

- ➔ };



Header file (3)

- Data members
 - ➔ private:
 - Int_t fChannelID;
 - Int_t fClock;
 - Int_t fTDC;



Constructor / destructor

- public:
 - ➔ `R3BTofRawItem(); // Default constructor`
 - ➔ `R3BTofRawItem(Int_t channelID, Int_t clock, Int_t tdc); // Standard constructor`
 - ➔ `virtual ~R3BTofRawItem();`



Accessors

- public:
 - ➔ `inline Int_t GetChannelID() const { return fChannelID; }`
 - ➔ ...
 - ➔ `inline void SetChannelID(Int_t channelID) { fChannelID = channelID; }`
 - ➔ ...



CINT Dictionary

- Header file: `ClassDef(R3BTofRawItem, 1)`
- Source file: `ClassImp(R3BTofRawItem)`



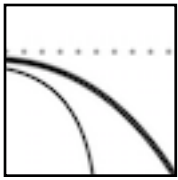
Source file

- #include "R3BTofRawItem.h"
- ...
- R3BTofRawItem::R3BTofRawItem(Int_t channelID, Int_t clock, Int_t tdc)
- : fChannelID(channelID),
- fClock(clock),
- fTDC(tdc)
- {
- }



Source file

- `R3BTofRawItem::~~R3BTofRawItem()`
- `{`
- `}`



Compilation

- R3BDataLinkDef.h
 - ➔ `#pragma link C++ class R3BTofRawItem+;`



Compilation (2)

- CMakeLists.txt
 - ➔ set(SRCS
 - ...
 - tofData/R3BTofRawItem.cxx)



Compile

- `cd "BUILDDIR"`
- `./config.sh`
- `nice make -j4`



Use the data class

- r3broot/tof
- In CMakeLists.txt:
 - ➔ add `${R3BROOT_SOURCE_DIR}/r3bdata/tofData` to `set(INCLUDE_DIRECTORIES ...`
 - ➔ add `R3BData` to `set(DEPENDENCIES ...`
- In Source file (`R3BTofUnpack.cxx`):
 - ➔ `#include "R3BTofRawItem.h"`



Fill data in R3BTofUnpack

- r3broot/tof
- In class declaration (R3BTofUnpack.h file)

➔ class TClonesArray;

private:

```
TClonesArray* fRawData;  
Int_t fNHits;
```




- In class implementation (R3BTofUnpack.cxx file)

- ➔ #include "TClonesArray.h"

- ➔ #include "FairRootManager.h"

1. R3BTofUnpack::R3BTofUnpack(.....) — initialize data members

```
fRawData(new TClonesArray("R3BTofRawItem"),  
fNHits(0))
```

2. R3BTofUnpack::Register() — register in output file

```
FairRootManager* fMan = FairRootManager::Instance();  
fMan->Register("TofRawItem", "Tof", fRawData, kTRUE);
```

3. R3BTofUnpack::Reset() — clear at the end of each event

```
fRawData->Clear();  
fNHits = 0;
```

4. R3BTofUnpack::DoUnpack(.....) — fill data

```
nHits++;  
c_leading =  
"  
new ((*fRawData)[fNHits]) R3BTofRawItem(i, c_leading, t_leading);  
fNHits++;
```



- In class implementation (R3BTofUnpack.cxx file)
-::Init()
- {
- Register();
- }