The 2nd part

LARG HEC TestBeam software in Athena framework

Oct. 12, 2001

CERN

Contact nkanaya@uvic.ca

Purpose for LARG HEC Tutorial

What we can do in the Athena Framework?

In the Athena Framework, one can reconstruct signal and produce a standard ntuple, which is the same as the one produced by the hec_adc framework.

And also, you can add your own code to LArHEC TB software, and get the histogram / ntuple you want.

At the end of this tutorial, you should know

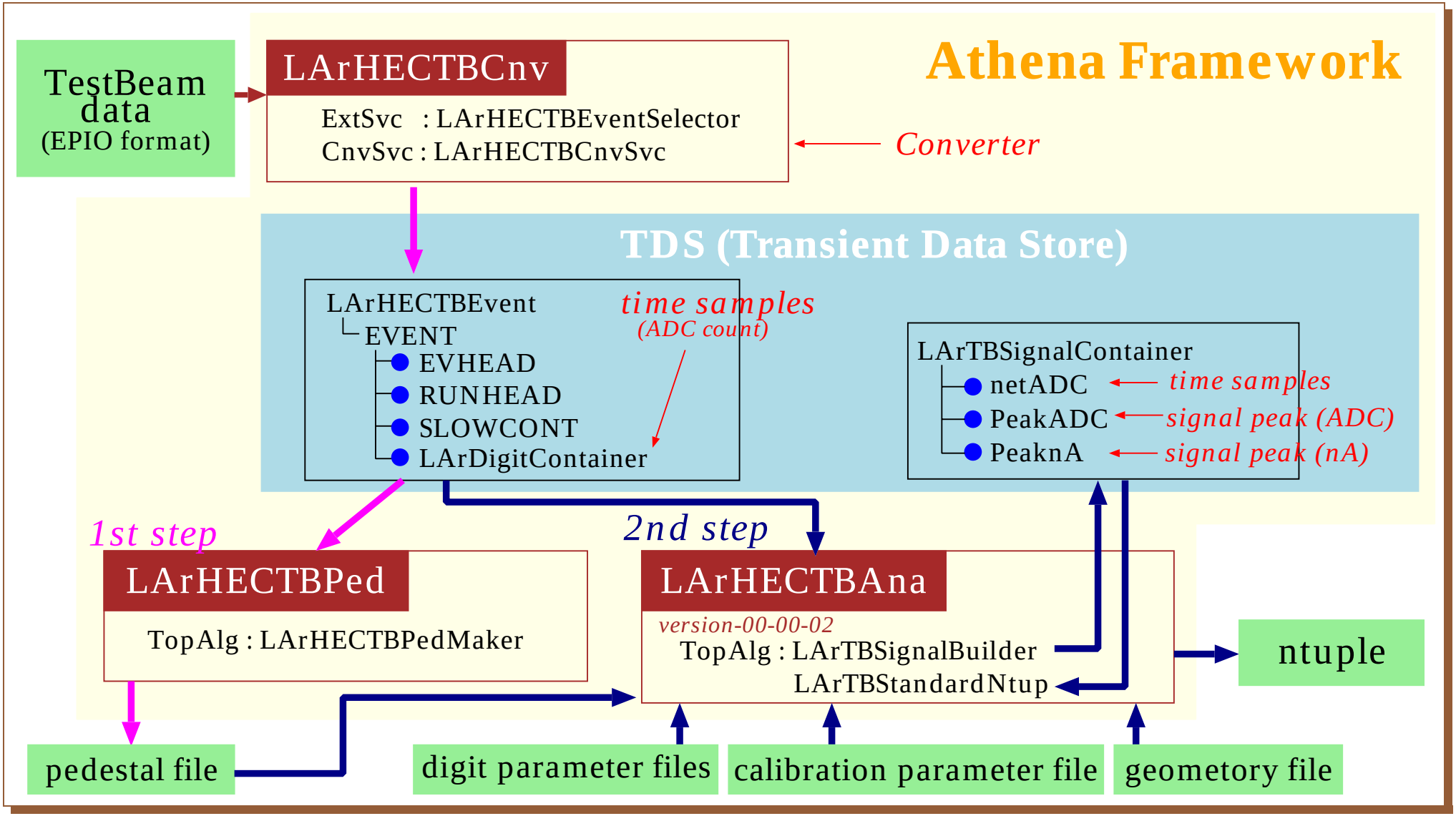
- **How to execute LArHEC TB software**
- **About HEC TB software and data structure in Athena.**
- **How to add your own analysis code in LArHEC TB software**

Contents

Section 1

About LArHEC TestBeam Software in athena

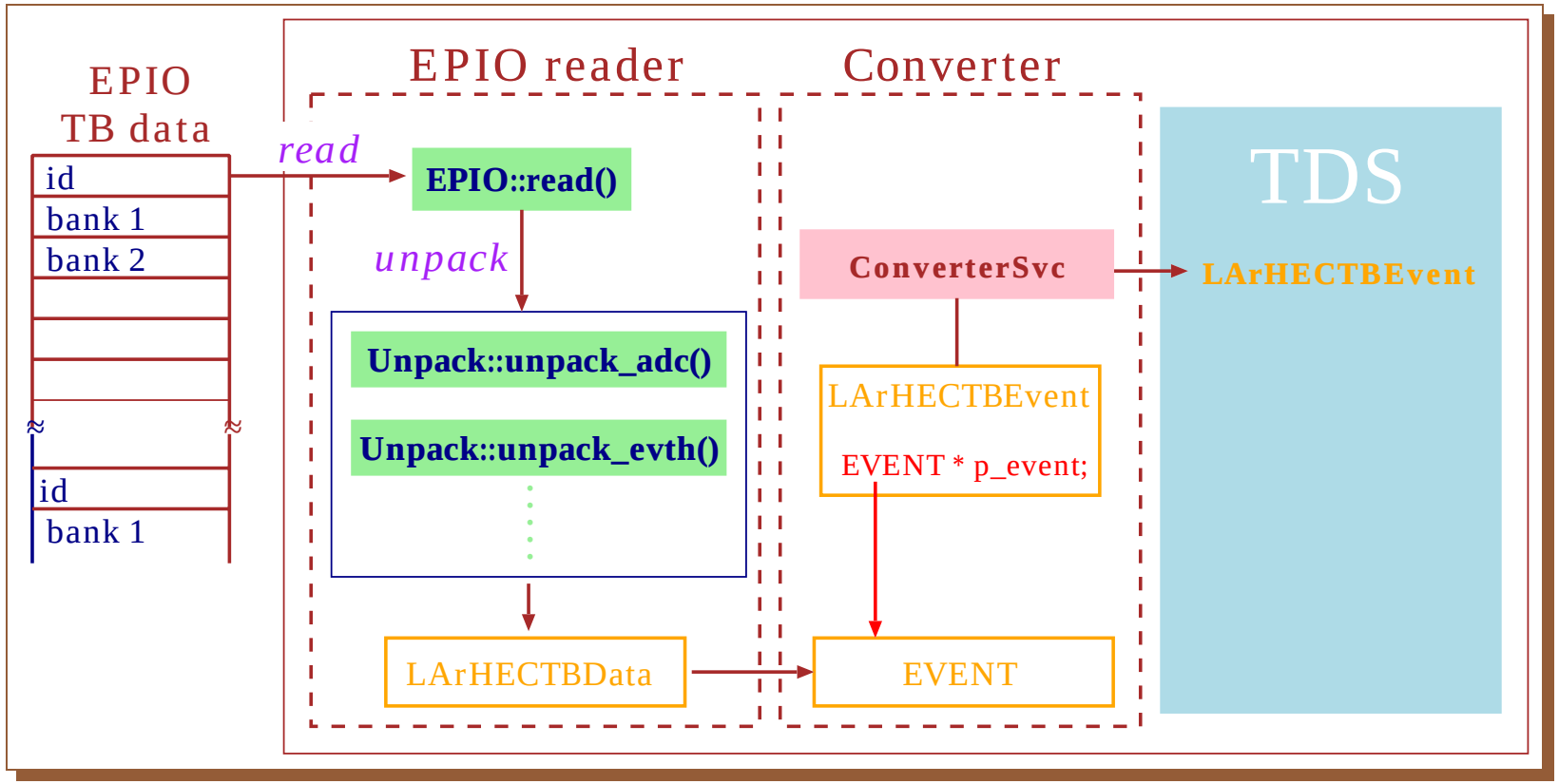
LAr HEC TestBeam packages in athena



LArHECTBCnv

Convert TB data from EPIO format to TDS

- Read EPIO data
- Unpack bank
- Create a data object to be recorded in TDS
- Record it in TDS



LARHECTBEvent

TB data recorded in TDS by LArHECTBCnv

It consists of the **EVENT**-type pointer and member functions.

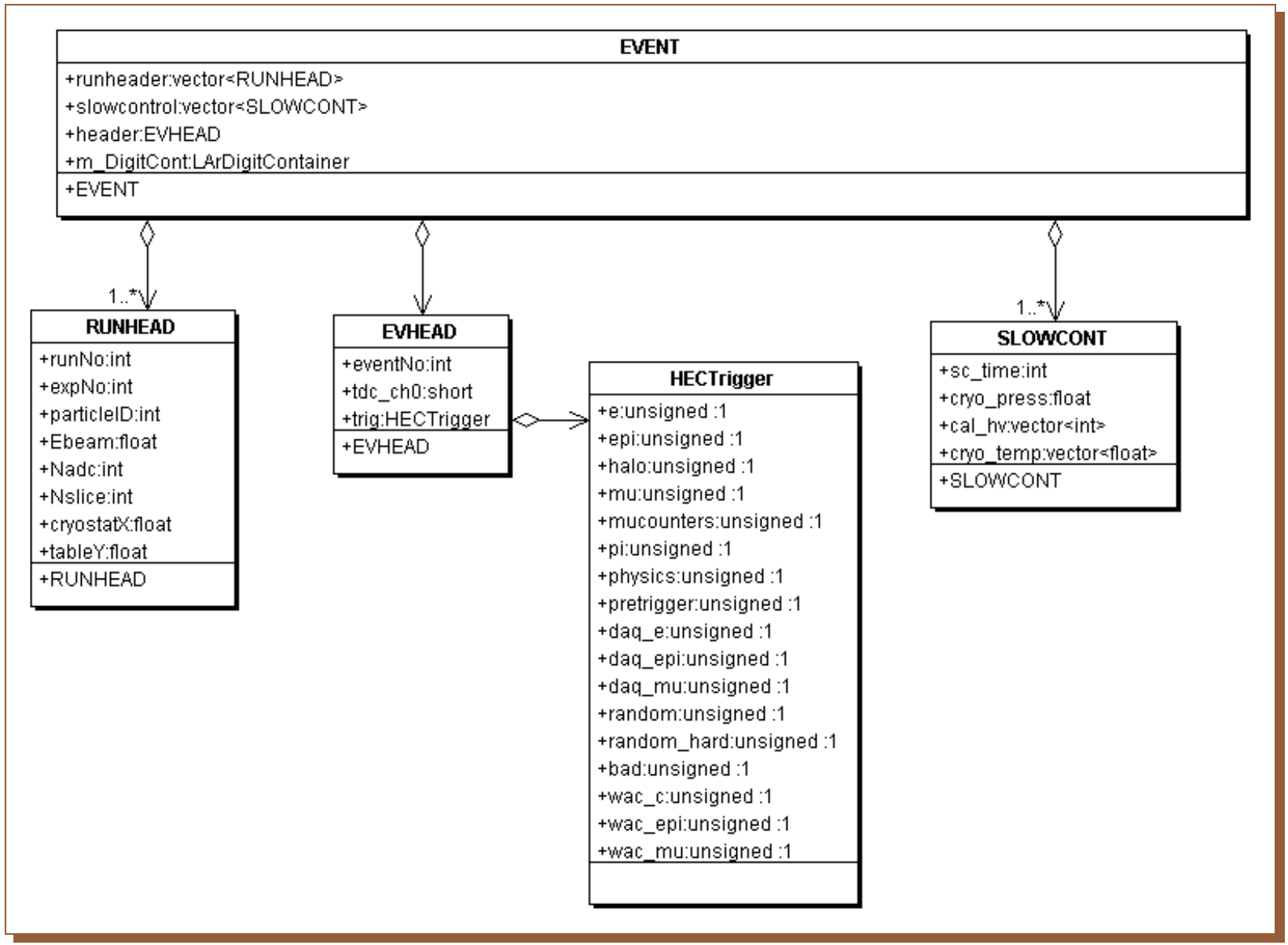
LArHECTBEvent.h

```
public :
  inline int event_number() const { return p_event->header.eventNo; }
  inline short tdc_count() const { return p_event->header.tdc_ch0; }
  inline HECTrigger * trigger_status() const { return p_event->header.trig; }
  inline RUNHEAD * run_header() const { return p_event->runheader; }
  inline SLOWCONT * slow_control() const { return p_event->slowcontrol; }
  inline EVENT * LArHECTB_event() const { return p_event; }
private :
  EVENT * p_event;
```

† Comments

RUNHEAD and SLOWCONT are usually included in the first event only.

Class Diagram in LARHECTBEvent



LArHECTBPed

Calculate pedestals and write them in an ascii file

Top Algorithm : LArHECTBPedMaker.cxx

- produce pedestals and their rms for each cell.
- calculate mean and rms using all events in a given run.
- possibility to apply σ_{cut} in order to skip noisy channel in pedestal calculation.
- Done using the first N events.

<LArHECTBPedMaker.cxx>

execute ( )

retrieve LArDigitContainer

```
for ( ifirst !=  ilast ; ifirst ++ ) {
 if ( nevents < N) {
                    }
 else {
                    }
}
```

calculate σ_{cut} , μ_{cut} using the first N events

calculate σ , μ using the remainning events with $|x - \mu_{cut}| < \sigma_{cut} * n$

finalize ( )

iostream out ;

out $\ll \mu \ll \sigma \ll N_{used} \ll \mu_{cut} \ll \sigma_{cut} \ll$ endl;

- Following variables can be changed by *jobOptions.txt*.
  - The first time sample # to be used.
  - The last time sample # to be used.
  - The number of events used for cut condition
  - The number of sigma for event selection
  - File name
- Output format

μ , σ , N , μ_{cut} , σ_{cut} for each cell

About jobOptions.txt

jobOptions.txt

```
#include "jobOptions_PedMaker.txt"
MessageSvc.OutputLevel      = 2;
ApplicationMgr.EvtMax = 10;
```

Message Stream Ouput Level

Number of events processed

jobOptions_PedMaker.txt

```
ApplicationMgr.DLLs += { "StoreGate", "LArHECTBPed" ,
"LArHECTBCnv","LArBookkeeping" };  (1)

ApplicationMgr.TopAlg += { "LArHECTBPedMaker/LArHECPed" };(2)

ApplicationMgr.ExtSvc += { "StoreGateSvc", "LArHECTBCnvSvc" ,
"LArBookkeepingSvc" , "LArHECTBEventSelector/EventSelector" };  (3)

EventPersistencySvc.CnvServices     = { "LArHECTBCnvSvc" };  (4)

LArHECPed.FirstSlice = 0;
LArHECPed.LastSlice = 0;
LArHECPed.Nsigma = 3;
LArHECPed.Nevent = 500;
LArHECPed.OutputFileName = "ped_r10053.dat" ;

EventSelector.RunNb = { "10053" };
```

properties

(5)

Run number

(1) **Shared libraries used**

(2) **Top Algorithm name(*.cxx)**

mandatory

(3) **Service names executed**

(4) **Converter service name**

(5) Properties in LArHECTBPed

- the first slice used for pedestal calculation
- the last "
- The number of sigma used for event selection
- The number of events used for μ_{cut} and σ_{cut}
- Output file name

LArBookkeepingSvc

~nkanaya/maxidisk/uvic/data/run_10053.dat

LArHECTBAna

Reconstruct signal

There are two TopAlgorithms and four subAlgorithms.

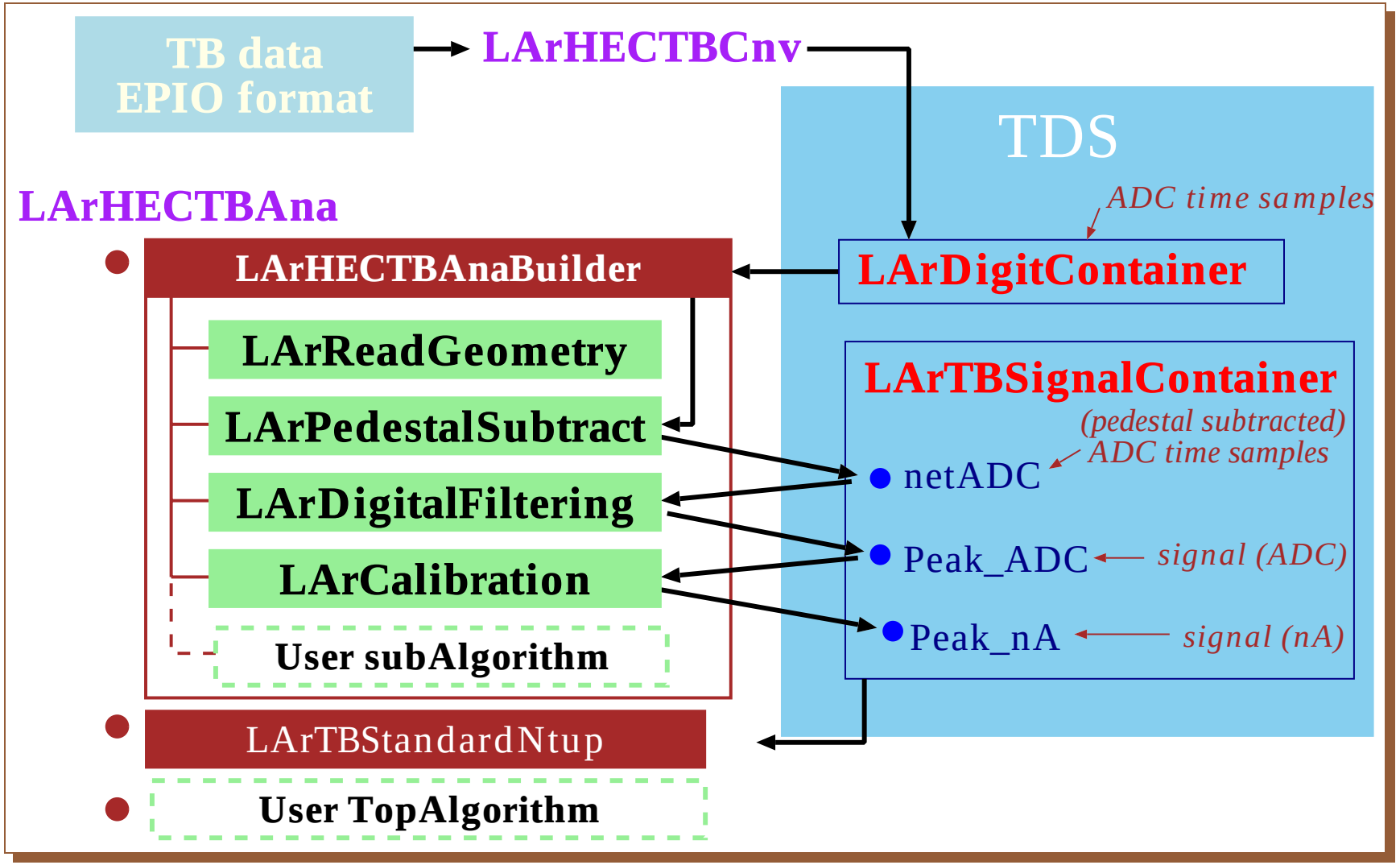
Top Algorithm : LArTBSignalBuilder.cxx

→ *reconstruct a signal*

Sub Algorithm

- LArReadGeometry.cxx — *read geometry file*
- LArPedestalSubtract.cxx — *do pedestal subtraction*
- LArDigitalFiltering.cxx — *find signal peak and its time for given cell*
- LArCalibration.cxx — *convert adc to nA*

LArHECTBAna (continue)



LArHECTBAna (continue 2)

Top Algorithm : LArTBStandardNtup.cxx

- run header ntuple (ID = 100)

variable	contents
hec_runno	run number
hec_runpd	run period number
hec_beame	beam energy
hec_noevt	number of event
hec_parttype	article type (1=e, 2=μ, 3=π)
hec_ctyox	cryostat position in x
hec_tabley	table position in y
hec_peakf	peak finding method (2=degital filtering)
hec_eunit	units of energy (1 = energy)
hec_cells_used	the number of cells used
hec_ped_rms	run pedestal rms for each channel
hec_ieta	eta value for each channel
hec_iphi	phi value for each channel
hec_iz	z value for each channel
hec_ic	adc channel number

- event ntuple (ID = 101)

variable	contents
hec_evetno	event number
hec_trig	tirgger flag array
hec_nchan	number of good channels
hec_signal (hec_nchan)	signal for each channel

- slow control ntuple (ID = 102)

variable	contents
hec_adc_used	number of used channels
hec_lartemp	liquid argon temperature
hec_press	pressure

About jobOptions.txt

LArHECTBAna_jobOptions_SignalBuilder.txt → LArHECTBAna_jobOptions.txt

```
ApplicationMgr.DLLs += { "StoreGate", "LArHECTBPed" ,
"LArHECTBCnv","LArBookkeeping" , "HbookCnv" };                    (1)
ApplicationMgr.TopAlg += { "LArTBSignalBuilder/LArBuilder",
                           "LArTBStandardNtup/ LArNtup" };        (2)
ApplicationMgr.ExtSvc += { "StoreGateSvc", "LArHECTBCnvSvc" ,
"LArBookkeepingSvc" , "LArHECTBEventSelector/EventSelector" };
EventPersistencySvc.CnvServices     = { "LArHECTBCnvSvc" };
EventSelector.RunNb = { "10053" };
-----------------------------------------------------------------
LArBuilder.ProcessNames = { "LArReadGeometry/LArGeo" ,
                            "LArPedestalSubtract/LArPed" ,
                            "LArDigitalFiltering/LArDig" ,
                            "LArCalibration/LArCal" };             (3)
LArPed.PedestalFileName = " ...../ped_r10053.dat";                 (4)
-----------------------------------------------------------------
ApplicationMgr.HistogramPersistency="HBOOK";
NTupleSvc.Output  = { "FILE1 DATAFILE='hec_adc.ntp' OPT='NEW'"};   (5)
LArNtup.Energy_unit = "nA";     (6)
```

(1) **For a histgram/ntuple**

(2) **Top Algorithm name(*.cxx)**) mandatory

(3) **subAlgorithm name(*.cxx)**

(4) **input file name**

(5) **for ntuple**

(6) **fill signal in unit of "nA"**

LArBookkeeping

- **Tasks**
  - manage data stored at different places (HPSS, Castor).
  - visualize and edit run information.
- **LArBookkeeping based on mySQL**
- **web interface available in** *http://larbookkeeping.in2p3.fr*
  - shifter interface
  - user interface
- *Interface to Athena available via*
  - *LArBookkeepingSvc*
  - you only have to select run in jobOptions.txt

EventSelector.RunNb = { *"10053"* } ;

data file staged according to run number.

Section 2

How to execute LArHEC TB software

How to execute a package?

There are two ways to execute Athena, which depends on what you want to do.

(1) Execute Athena without building

If you don’t modify any existing package, you should not check out the package you want to use. Binding necessary shared libraries at run time is sufficient.

(such a work is performed by CMT according to *requirements* file.)

(2) Execute Athena with building

If you want to change a package, you have to check out the package you want to modify. You can produce your shared library in your own directory, and bind it at run time.

Exercise 1

Execute LArHEC code without building

General setup for LArHEC

A template package (TestRelease) is prepared for users.

execute LArHECTB code without building

```
atlas> goto_build [1]
atlas> vi requirements [2]
atlas> cmt config [3]
atlas> source setup.sh [4]
atlas> gmake [5]
atlas> goto_run [6]
atlas> cp $LARHECTBPEDROOT/share/*.txt . [7]
```

† Comments

- [2] add the following lines

```
use LArHECTBPed LArHECTBPed-00-* LArCalorimeter/LArTestBeam
use LArHECTBAna LArHECTBAna-00-* LArCalorimeter/LArTestBeam
```

- [7] `$LARHECTBPEDROOT` is set automatically by setup.sh script.

Exercise 2

Produce a pedestal and a standard ntuple

- Run LArHECTBPed to create pedestal file
  - `athena LArHECTBPed_jobOptions.txt`
- Do you see the pedestal file?
- Execute LArHECTBAna, and
  - Produce the standard ntuple
  - `athena LArHECTBAna_jobOptions.txt`
  - Have a look at the standard ntuple “hec_adc.ntp”

Section 3

How to add your code

Exercise 3

Execute LArHECTBAna with building

If you want to change LArHECTBAna :

- add your own analysis code
- modify existing code

you should check it out, edit code and build the package.

copy LArHECTBAna package without building

```
atlas> .  $LArTutorial/scripts/Setup_HECexample.sh [1]
```

goto your work area : $HOME / maxidisk / Tutorial / LArCalorimter

```
atlas> cp -r $LArTutorial/code/LArCalorimeter/LArTestBeam . [2]
atlas> goto_build [3]
atlas> cmt broadcast cmt config [4]
atlas> cmt broadcast gmake [5]
atlas> goto_source [6]
```

(1) Edit LArHECUserHist.cxx

- Search FIXME (two parts)
- book a histogram and fill the average of first 3 samples

Congratulations !

You successfully finished the tutorial