

Update of evaluation of **ROOT I/O for LCIO**

Frank Gaede
DESY

ILD Software WG Meeting
December 12, 2009

Outline

- introduction
- short summary of status in September
- new developments since then
- current status and plans

reminder: why look into ROOT I/O

- ROOT I/O (RIO) is used by all LHC experiments and most other ongoing HEP experiments
 - -> large user community & experience
- possibly could provide missing features:
 - direct access to events
 - splitting of events over files (sim, rec, dst)
 - storage of generic user classes in LCIO files
 - Note: these can also be implemented in an improved SIO
- former GLD groups heavily use ROOT and ROOT macros for physics analysis
 - -> would like to use LCIO classes in ROOT macros !

status in september I

- created dictionary for (almost) all classes of LCIO – including complete Event Data Model
- thanks to ROOT team for their help and for adding some features to ROOT 5.24.00
- with this dictionary, one can:
 - write LCIO objects to ROOT TTree's
 - use LCIO objects/classes in ROOT macros
- e.g. can create **.rlcio** files from existing DSTs :
 - copyfix DST01-06_ppr004_bbc_sdu.**slcio** DST01-06_ppr004_bbc_sdu.**rlcio**
 - dumpevent DST01-06_ppr004_bbc_sdu.**rlcio** 42
 - anajob DST01-06_ppr004_bbc_sdu.**rlcio**
 - **note: main application code is unmodified wrt. current LCIO !**

status in september II

- cannot have pointers between branches w/o using TObject & TRefs -> store complete LCEvent in one branch
- observations:
- **pros:**
- can use unmodified user code w/ new persistency
- rather simply to add user classes (no streamer writing)
- can use LCIO in ROOT macros
- direct access
- **cons:**
- no branches (slow !)
- no use of TBrowser and TTreeView
- no split files
- no Java interface

activities since september

- working meetings at CERN and DESY (FG, A.Naumann)
- identified solutions for pointers between branches (index based)
- fixed some bugs issues in ROOT (cint, TBrowser,...)
- identified 'improvements' on LCIO classes for use w/ ROOT:
 - made LCollectionVec template (need type of LCOjects)
 - use inclusion of `std::vector<LCOject*>` rather than inheritance in LCollectionVec (ROOT browser parser problem)
- implemented index based pointers
 - (`hash(colName) << 32 + index`)
 - ignored for now for SIO files – could be used for splitting events over files and partial event reading in .slcio files
- implemented streaming mode – needed for Marlin

current status

- can write LCEvents split over branches (one per LCCollection)
- can use TBrowser (and to some extent TTreeViewer)
 - even w/o dictionary
- need ROOT 5.25.04 (or higher)
- use with Marlin currently under testing
- plan 'experimental release' soon
 - -> would like to get some feed back from users

```

std::vector<TBranch*> branches ;

//***** define and read ROOT branches *****

IMPL::LCEventImpl* evt =0 ;
TBranch* bevt = t->GetBranch("LCEvent") ;
bevt->SetAddress( &evt ) ;
branches.push_back( bevt ) ;

evt->setCurrentEvent( evt ) ;

IMPL::LCCollectionVec* col = new IMPL::LCCollectionVec ;
branches.push_back( t->GetBranch("MCParticlesSkimmed") ) ;
branches.back()->SetAddress( &col ) ;

IMPL::LCCollectionVec* cpfo = new IMPL::LCCollectionVec ;
branches.push_back( t->GetBranch("PandoraPF0s") ) ;
branches.back()->SetAddress( &cpfo ) ;

IMPL::LCCollectionVec* crml = new IMPL::LCCollectionVec ;
branches.push_back( t->GetBranch("RecoMCTruthLink") ) ;
branches.back()->SetAddress( &crml ) ;

evt->addCollection( col , "MCParticlesSkimmed" ) ;
evt->addCollection( cpfo , "PandoraPF0s" ) ;
evt->addCollection( crml , "RecoMCTruthLink" ) ;

int nBranches = branches.size() ;
int nevt = t->GetEntries();

for (Int_t i = 0; i < nevt ; i++) {

    Long64_t tentry = t->LoadTree(i);

    for (Int_t k = 0; k < nBranches ; k++) {
        int nbyte = branches[k]->GetEntry(tentry);
    }

//***** plain LCIO code below this line *****

int nMCP = col->getNumberOfElements() ;
double eMCP = 0.0 ;
for(int j=0;j<nMCP ;++j){

    EVENT::MCParticle* mcp = dynamic_cast<EVENT::MCParticle*>( col->getElementAt(j) ) ;
-- readevent.C 37% (71,50) CVS:1.1.2.6[rio_v00-00] (C++/l Abbrev)
(No changes need to be saved)

```

example of ROOT macro with LCIO

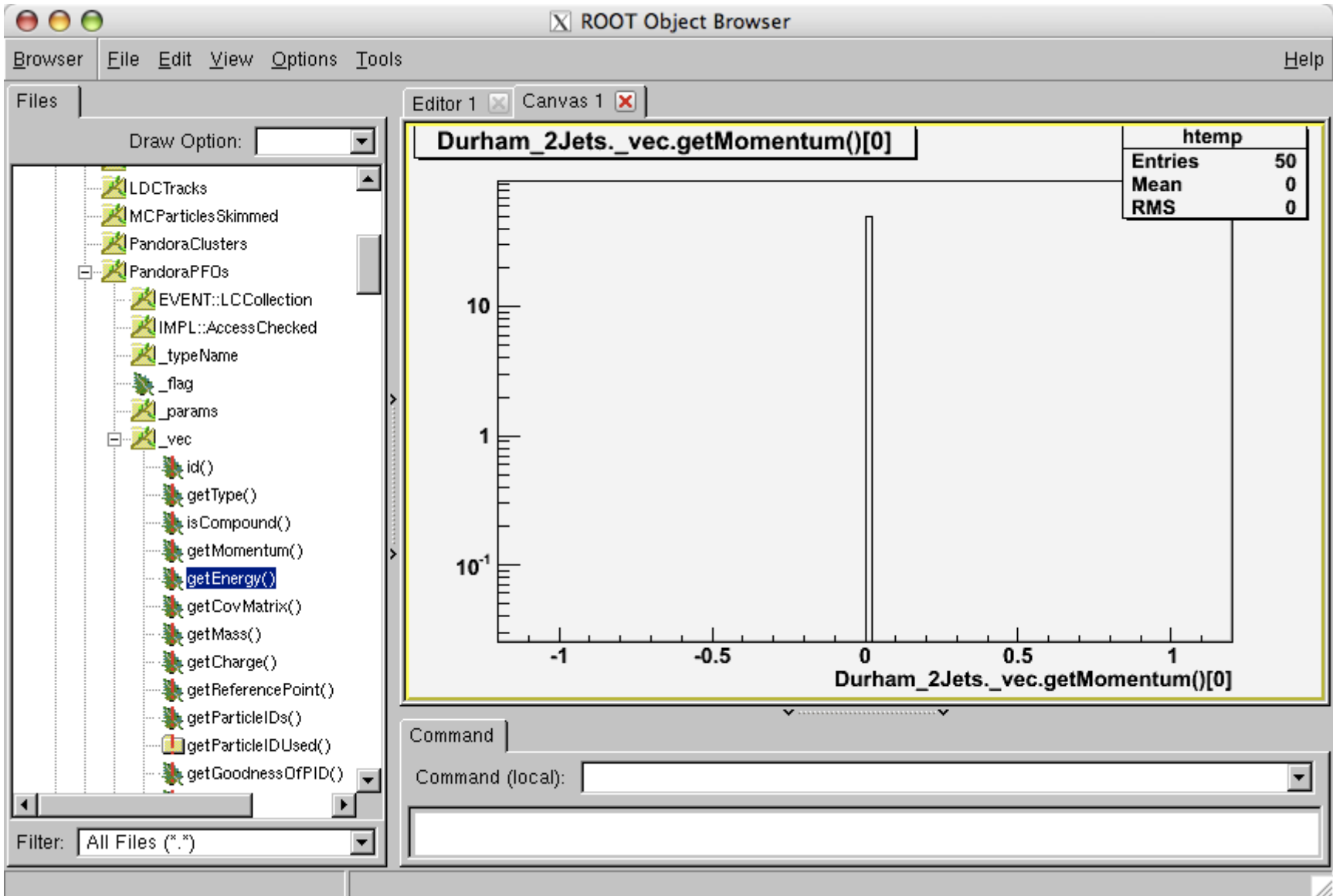
- boiler plate code for opening and reading ROOT file
- plain LCIO code section
-> can be copied into Marlin processor after rapid prototyping in ROOT

The screenshot shows the ROOT Object Browser interface. On the left, a tree view displays the object hierarchy, with `PandoraPFOs._vec.getEnergy()` selected. The main canvas displays a histogram of the energy values. The y-axis is logarithmic, ranging from 1 to 10^3 . The x-axis is linear, ranging from 0 to 100. A statistics table in the top right corner provides the following data:

htemp	
Entries	6753
Mean	3.453
RMS	6.229

At the bottom of the window, there is a Command input field and a Command (local) dropdown menu.

simple histograms (from scalars) via mouse click !



```
LCEvent->Draw("Durham_2Jets._vec.getMomentum()[0]")
```

does not work !?

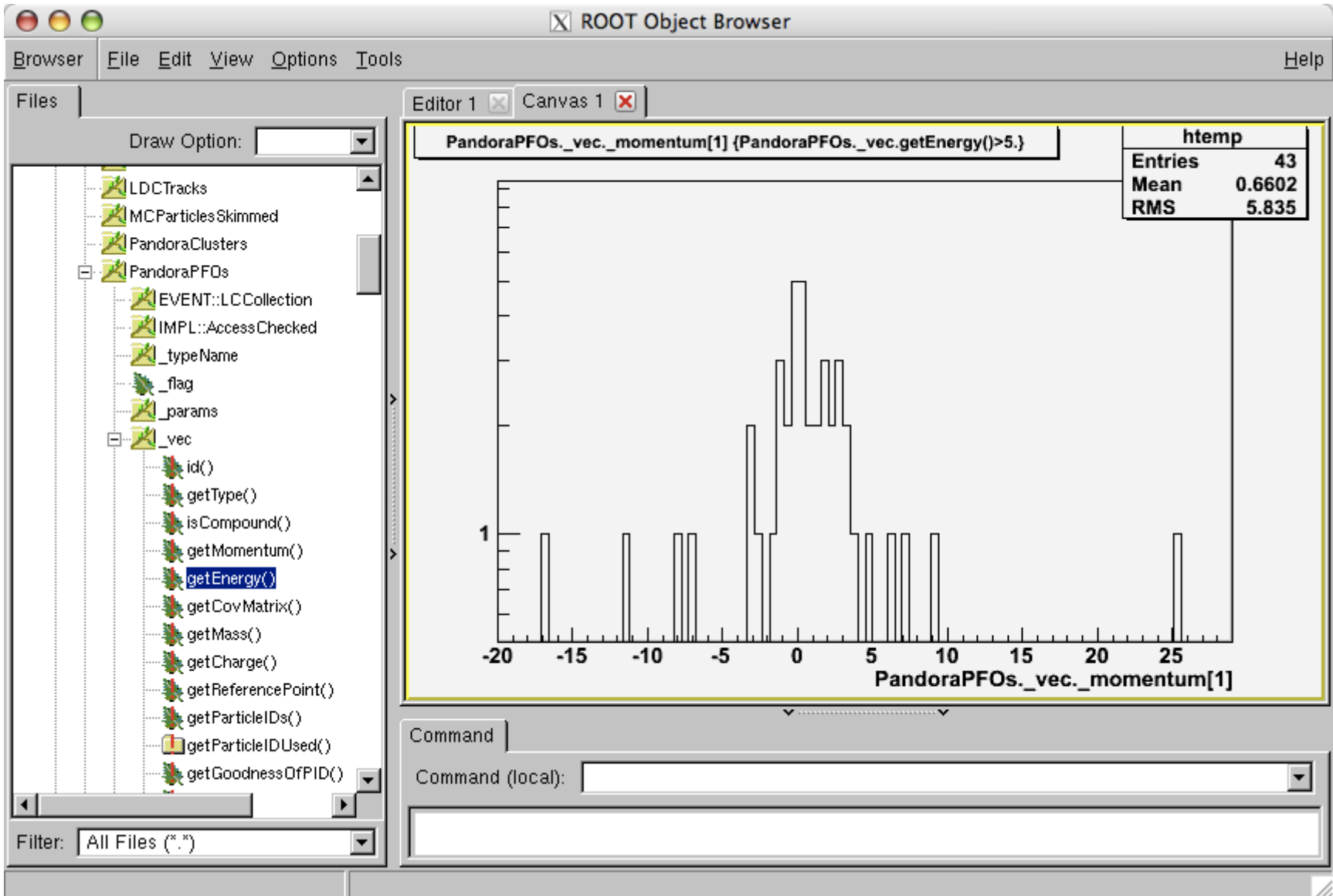
The screenshot shows the ROOT Object Browser interface. On the left, a tree view displays the object hierarchy, with `getEnergy()` selected under the `_vec` object. The main canvas displays a histogram titled `Durham_2Jets._vec._momentum[0]`. The x-axis is labeled `Durham_2Jets._vec._momentum[0]` and ranges from -200 to 200. The y-axis is on a logarithmic scale from 1 to 10. A statistics table in the top right corner shows the following data:

htemp	
Entries	150
Mean	-1.108
RMS	99.95

At the bottom, there is a Command input field with the text `Command (local):` and a dropdown menu.

`LCEvent->Draw("Durham_2Jets._vec._momentum[0]")`

does work !?



LCEvent->Draw("PandoraPFOs._vec._momentum[1]","PandoraPFOs._vec.getEnergy()>5.")

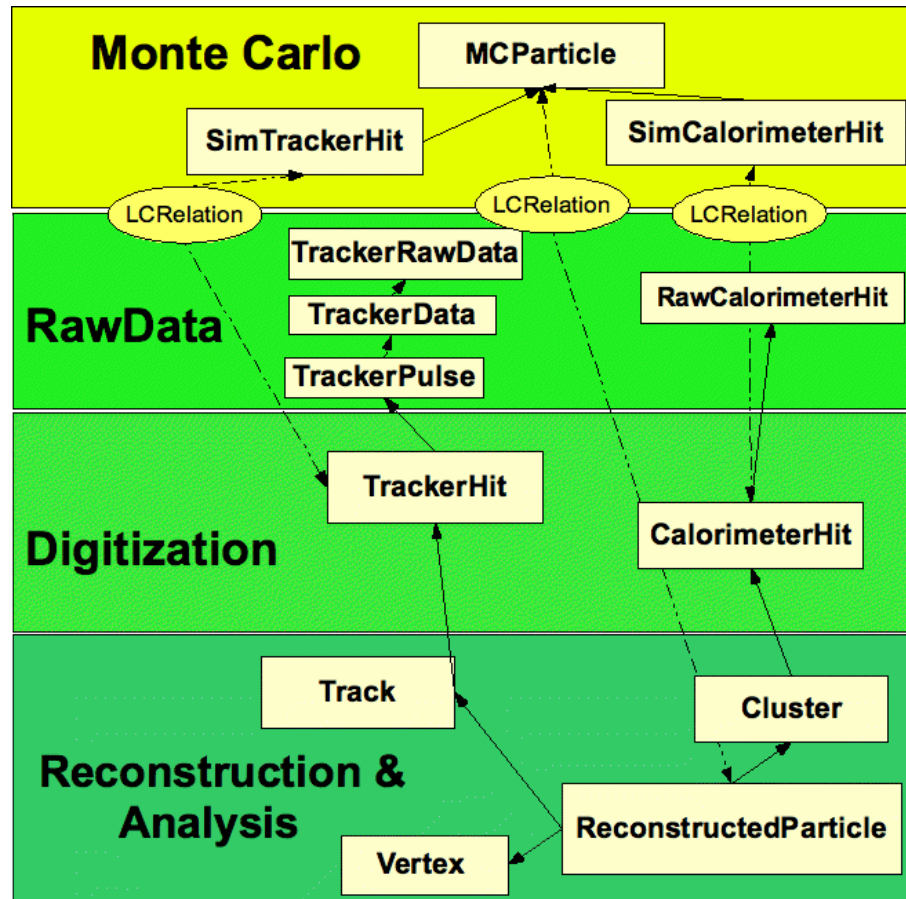
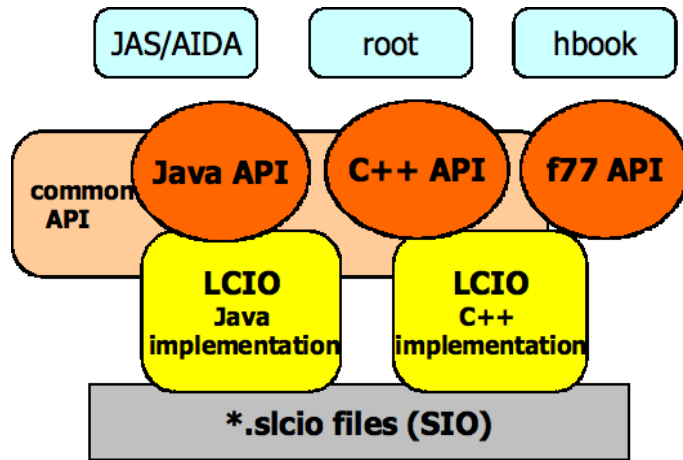
-> simple 'analysis' of LCIO DSTs with Ttree::Draw possible 12

Summary & Outlook

- an experimental 'prototype' implementation of ROOT I/O for LCIO DST classes (tracks, clusters, reco particles and vertices) exists in C++
- event split into branches/collections
- simple use of browser possible
- currently finalized also for streaming mode (Marlin)
- plan 'expert test release' soon (EOW?)
 - cvs branch 'rio_v00-00'
- Outlook
 - need evaluation by users and experts, further improvements
 - a reasonable solution for Java yet has to be found
 - also started to look into improving SIO (direct access,...)

additional material

LCIO: persistency & event data model



- joined DESY and SLAC project - first presented @ CHEP 2003
- provides **persistency (I/O)** and an **event data model (EDM)** to ILC detector R&D community
- features:
 - Object I/O (w/ pointer chasing)
 - schema evolution
 - compressed records
 - hierarchical data model
 - decoupled from I/O by interfaces
 - C++, Java (and Fortran)
 - some generic user object I/O

LCIO is used by ILD, SID, Calice, EUPixelTelescope, LCTPC,...

LCIO philosophy

- LCIO provides an EDM in terms of an **abstract interface** for reading and a **default implementation** for writing LCIO files
- user code is **completely decoupled from actual persistency**
- persistency is provided through – hand written – streamer classes in a non-intrusive way
- currently (since 2003) we provide SIO persistency which is packaged w/ LCIO for user convenience
- -> **only LCIO needed to read any LCIO file (no dictionary needed)**

SIO persistency & possible improvements

- SIO: Simple Input Output
 - originally developed at SLAC – now maintained by LCIO team
 - provides OO persistency with built-in compression incl. pointers within one record (one event)
 - LCIO events have to be read completely
 - no splitting of events over files (sim, rec, dst)
 - no direct access to events
 - only via 'fast skip' or creation of TOC on open (slow)
 - splitting and direct access could be added with some effort or be using a more powerful I/O package

ROOT I/O features

- automatic generation of streamer code for almost arbitrary C++ classes
- automatic schema evolution
- no need to inherit from TObject or include ClassDef(version) macros
 - -> almost non intrusive
- pointers within one branch
- compression
- direct access, TBrowser, TTreeBrowser,...
- ...many more ... (see <http://root.cern.ch/drupal>)