LCIO 2.0

Status and open issues

Frank Gaede, DESY

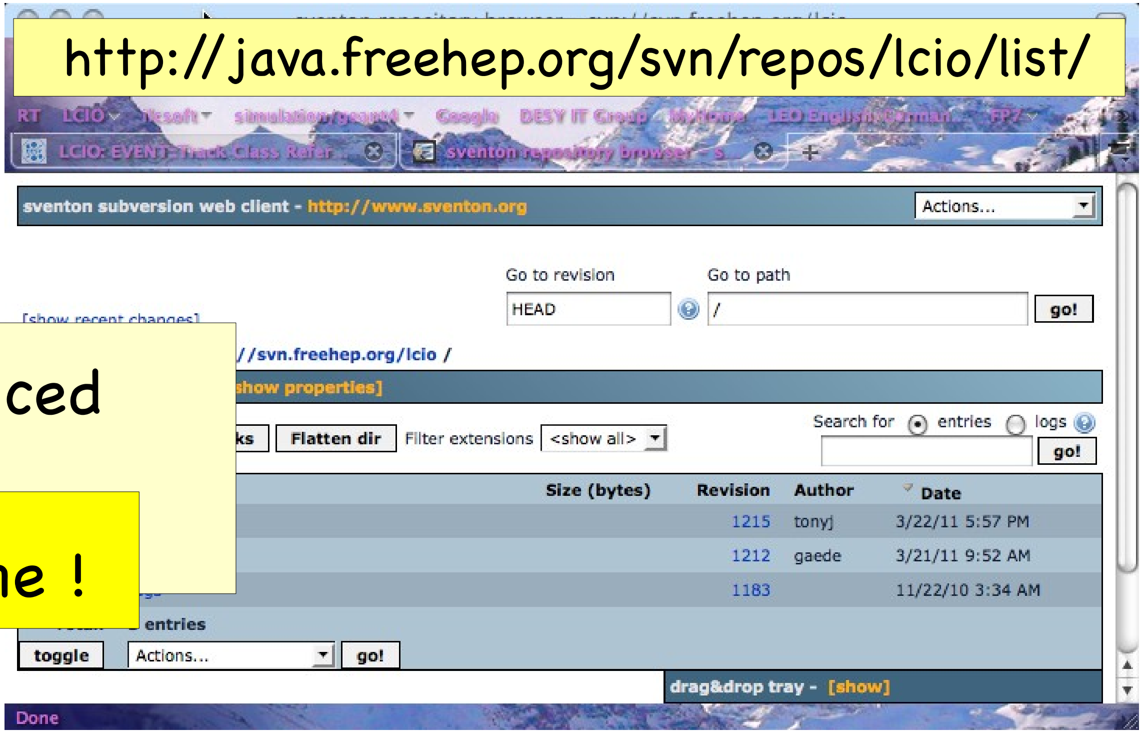
Software Common Task Group

Meeting, Sep 8, 2011

LCIO repository moved to SVN

svn webinterface:

http://java.freehep.org/svn/repos/lcio/list/



sventon subversion web client - http://www.sventon.org

Go to revision HEAD

Go to path /

[show recent changes]

//svn.freehep.org/lcio /

show properties]

Flatten dir

Filter extensions <show all>

Search for ● entries ○ logs

Size (bytes)	Revision	Author	Date
	1215	tonyj	3/22/11 5:57 PM
	1212	gaede	3/21/11 9:52 AM
	1183		11/22/10 3:34 AM

entries

need to be announced
- manual
- web page

Done !

checkout released versions:

svn co svn://svn.freehep.org/lcio/tags/v01-60 v01-60

checkout HEAD version:

svn co svn://svn.freehep.org/lcio/trunk trunk

old CVS still works for checkout of released versions !

LCIO 2.0 - new features

- LCIO 2.0 (AKNA LCIOv2) is planned for some time now
- goal is to improve LCIO while still being backward compatible
- planned/requested features:
- **direct access to events -> Done**
- **partial reading of events -> postponed**
- **splitting of events over files -> postponed**
- **storing of (arbitrary) user classes -> currently not planned**
- **simplify using LCIO with ROOT -> Done**
  - (ROOT macros, TTreeViewer, I/O (?) ,...)
- **improving the event data model -> Done**
  - (1d,2d hits, tracks/trajectories)

Frank Gaede, Software CTG Meeting, Sep 8, 2011

cleanup of build systems

- C++
  - remove old Makefiles - have CMake only

Done !

- Java
  - remove old ant scripts
  - have Maven only
  - -> include Maven in release
    - no dependency for C++

Done !?

  - -> Maven plugin for creating header files only once
    - interesting for developers - (no rebuild after install)

extensions of MCParticle

- add spin information:
  - float[3] getSpin()
- add color flow information
  - int[2] getColorFlow()
    - are these pointers to other MCParticles (indices) ?
- -> both copied from stdhep/HepEvt4 as written by Whizzard

Done !

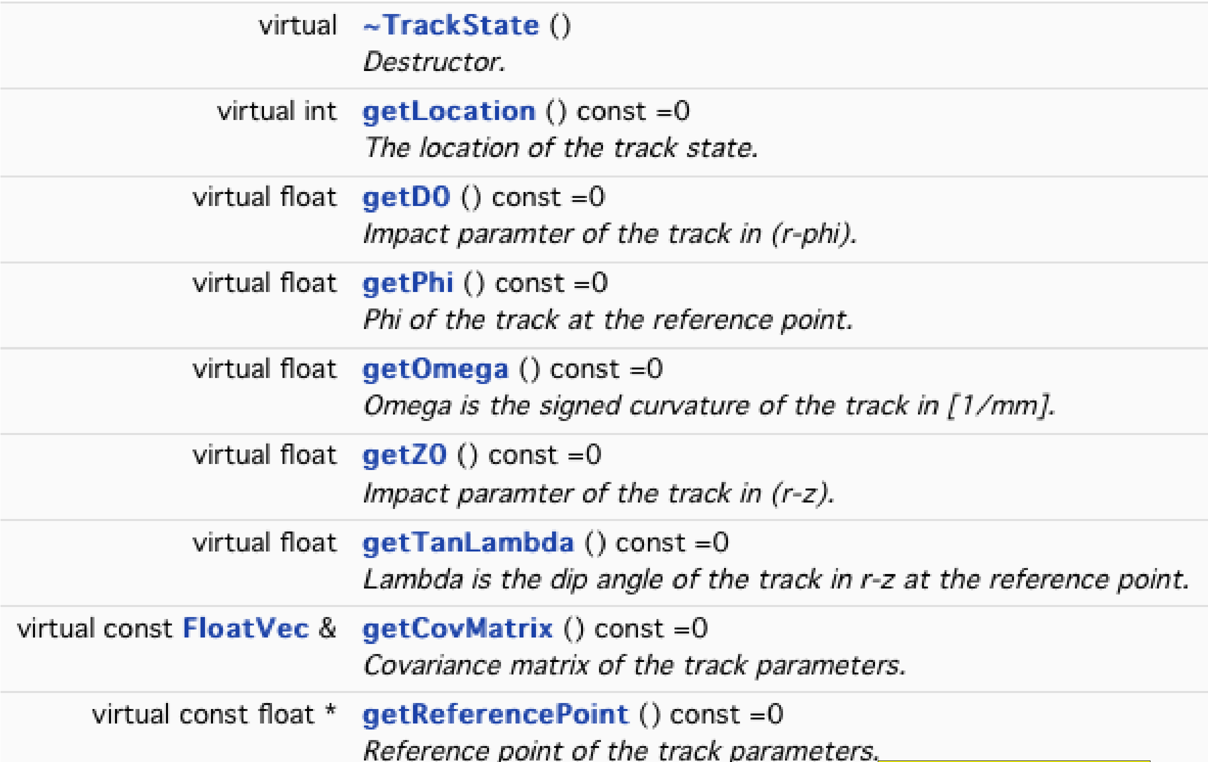
- user request:
  - have simProcessId for particles that decayed in simulator
  - -> will use lower 16 bits of SimStatus word + collection parameters: SimProcessID, SimProcessName
  - short getSimProcessID()
    - need to define details of processIDs
    - implement this in Mokka and SLIC the same way

should be postponed to next minor release !

Frank Gaede, Software CTG Meeting, Sep 8, 2011

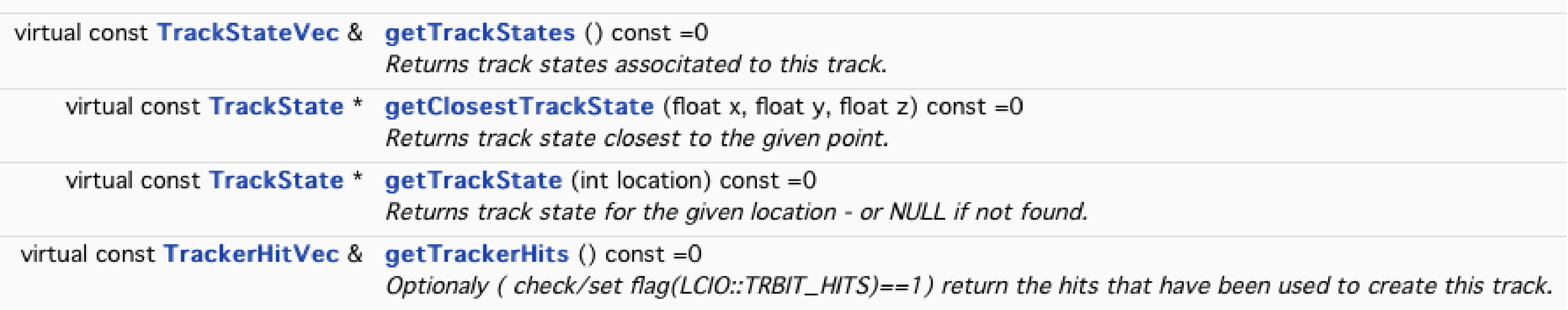
Track with multipleTrackStates

- Track now has multiple TrackStates
- canonical TSs:
  - TrackState::AtIP, AtFirstHit, AtLastHit, AtCalo, AtVertex, AtOther
- TS returned either by
  - identifier
  - or closest to given point
- mostly backward compatible (isReferencePointPCA dropped)



virtual	~TrackState () *Destructor.*
virtual int	getLocation () const =0 *The location of the track state.*
virtual float	getD0 () const =0 *Impact paramter of the track in (r-phi).*
virtual float	getPhi () const =0 *Phi of the track at the reference point.*
virtual float	getOmega () const =0 *Omega is the signed curvature of the track in [1/mm].*
virtual float	getZ0 () const =0 *Impact paramter of the track in (r-z).*
virtual float	getTanLambda () const =0 *Lambda is the dip angle of the track in r-z at the reference point.*
virtual const FloatVec &	getCovMatrix () const =0 *Covariance matrix of the track parameters.*
virtual const float *	getReferencePoint () const =0 *Reference point of the track parameters.*

Done !



The tracks that have been combined to this track.

virtual const TrackStateVec &	getTrackStates () const =0 *Returns track states associtated to this track.*
virtual const TrackState *	getClosestTrackState (float x, float y, float z) const =0 *Returns track state closest to the given point.*
virtual const TrackState *	getTrackState (int location) const =0 *Returns track state for the given location - or NULL if not found.*
virtual const TrackerHitVec &	getTrackerHits () const =0 *Optionaly (check/set flag(LCIO::TRBIT_HITS)==1) return the hits that have been used to create this track.*

Tracker-and CalorimeterHit

- canonical way of accessing layer number:
  - local to sub detector (inside-out, starting from 0)
  - getLayerNumber(), setLayerNumber()
    - filled from cellIDs after reading, write to cellID
  - need convention: string "layer" in CellIDEncoding
  - if "layer" not present - layerNum = -1 (deal with this in Marlin/org.lcsim)
  - will update SLIC and Mokka accordingly

droped - layer needs to be accessed through cellID !

- add cellIDs to TrackerHit:
  - getCellID0(), getCellID1() (-> same as in CalorimeterHit)
  - use cellID for consistency w/ CaloHit - even though there are no cells
  - drop old 'type' word and replace getType() with access to cellID["type"]

Done !

- question: convention for subdetectorIDs in cellIDs ?
  - -> this will probably have to be done on a per concept (detector) basis
  - -> need convention for ILD for DBD reconstruction

Done !

Frank Gaede, Software CTG Meeting, Sep 8, 2011

additional extensions

- to Cluster add
- float getEnergyError()

Done !

- float getTime() <- new request from ILD/CLIC

to be done ?

- to SimCalorimeterHit optionally add the position where the energy deposition (step) occurred:

Done ! v01-60

- float[3] getStepPosition(int i)
  - only if flag LCIO.CHBIT_STEP==1
    - useful for detailed simulation studies of edge effects in calorimeter cells or MAPS digitization

needed asap for DHCAL !!

1d and 2d TrackerHits

- agreed to have two new TrackerHit classes :
- TrackerHitPlanar
  - x, y, z - 'space point'
  - u(theta, phi) , v(theta, phi) – measurement directions (spanning vectors in the plane)
  - du, dv - measurement errors
  - -> to be used for 1d and 2d

Done !

- TrackerHitCylindrical
  - x, y, z - 'space point'
  - R, Xc, Yc – cylinder parameters (parallel to z)
  - dphi, dz - measurement errors
  - -> to be used for 1d and 2d

Done !

- these also implement the TrackerHit interface (x,y,z, cov) for backward compatibility and code reusability
- some testing done within MarlinTrk and new digitizers:
  - so far no problems found