Cluearte

Cluearte is the ARTE interface to the BEE analysis library.

T. Glebe

Contents

1 Read/Write BEE events in ARTE

Names

1.1 int **clueWriteEvent** (const TableFillBase& tf)

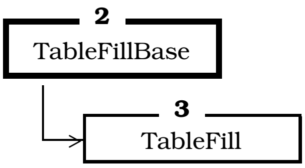
Write current ARTE event into a CLUE file

Write current ARTE event into a CLUE file. The filename is given by /CLUE/SETFILENAME

2 class **TableFillBase**

BEE event filling interface.

Inheritance



Public Members

virtual bool
EventSelect () const
returns false in case event should be skipped

virtual bool
Fill () const
fill all tables, called in clueWriteEvent()

virtual bool
FillRSEG () const
fill from RSEG table

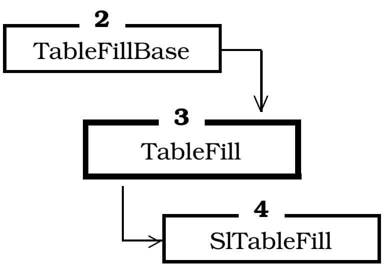
Abstract class which defines the cluearte interface for user defined filling procedures. You can define your own BEE event filling routines by inheriting a class from `TableFillBase` and overloading the pure virtual functions of the base class.

Author: T. Glebe

3 class **TableFill** : public TableFillBase

BEE event filling: 1 to 1 copy

Inheritance



Public Members

— Constructors —

TableFill ()

TableFill (double e)
set track energy threshold

virtual ~**TableFill** ()

— Functions —

virtual bool
EventSelect () const
returns false in case event should be skipped

virtual bool
Fill () const *fill all tables, called in clueWriteEvent()*

virtual bool
FillRSEG () const
fill from RSEG tables

virtual bool
FillWire () const *fill Wire table*

virtual bool

FillRecoTrack () const
fill RecoTrack table

virtual bool
FillRecoVertex () const
fill RecoVertex table

virtual bool
FillMCtrack () const
fill MC tracks

virtual bool
FillMCvertex () const
fill MC vertices

virtual bool
FillMisc () const *fill other information*

virtual bool
FillSelectedMCtrack () const
fill MC tracks if above energy threshold

virtual bool
FillSelectedMCvertex () const
fill MC vertices if tracks above threshold

— Access methods —

inline void
set_min_energy (double e)
MC: set min. energy of particles to be copied

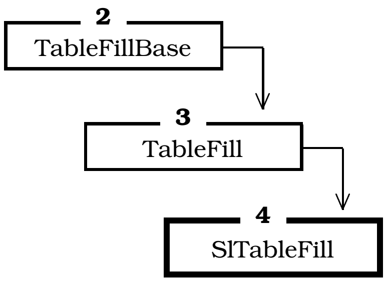
Class which defines a way to fill a BEE event. The TableFillBase interface class is used. The TableFill class performs a 1:1 copy of ARTE tables to BEE tables. In case of writing MC truth information, an energy threshold for the MC tracks (MTRA in ARTE) can be set. Then only those MTRA tracks are stored in BEE, which have an energy above the threshold. This gives the user the possiblity to optionally avoid low-energetic tracks from shower simulations. This can reduce the size of an BEE file considerably.

Author: T. Glebe

4 class **SlTableFill** : public TableFill

Single Lepton data BEE event filling

Inheritance



Public Members

— Constructors —

SlTableFill ()

virtual ˜**SlTableFill** ()

— Functions —

virtual bool **EventSelect** () const
call anl_evselect() from ARTE/analysis

virtual bool **FillRecoTrack** () const
use anl_TrSelect()

virtual bool **FillRecoVertex** () const
do not save RVER's

virtual bool **FillMisc** () const *fill other information*

Class which defines a way to fill a BEE event for a Single Lepton data analysis. The TableFill class is used as a base class in order to reuse the filling of MC truth information from the `TableFill` class.

Author: T. Glebe

5 class **PID_data**

helper class to keep PID information

Public Members

	— Constructors —	
	PID_data ()	
	— Class Members —	
void	**clear** ()	*set all data to 0*
	global member data	
int	**PID**	*particle identification flag*
float	**Lelectron**	*global electron likelihood*
float	**Lmuon**	*global muon likelihood*
float	**Lpion**	*global pion likelihood*
float	**Lkaon**	*global kaon likelihood*
float	**Lproton**	*global proton likelihood*

Author: T. Glebe

6 PID functions

Names

6.1 void **PID_init** ()

Initialize HBOOK histos for PID monitoring

Initialize HBOOK histos for PID monitoring. This function should be called in the ARTE usinit() function.

Author: T. Glebe

6.2 const PID_data& **checkPID** (const ArtePointer<RTRA>& rtra)

Strategy: Basic PID-device is the RICH detector

Strategy: Basic PID-device is the RICH detector. If information from RISE is present, it will be used. Otherwise likelihoods from RITER are used. In case neither leads to a positive identification despite the fact that a ring is associated to the track, some private code is used. (The code in many cases gives results similar to the output of RISE.) Rare light particles (e,μ) need a RICH likelihood above 0.99 to be identified. Pions are identified (essentially by default) if the likelihood exceeds 0.3. Kaons are identified if the likelihood is larger than 0.6 and protons are required to have a likelihood larger than 0.3. If hits in the muon system are present, then a particle with muon-likelihood above 0.0 is identified as muon. Also, if no RICH information is present, a track with hits in the muon system is identified as a muon.

Author: Michael Schmelling

7 Functions needed to select proper photon candidates

Names

7.1

void **fTransport** (const ArtePointer<RTRA>& rtra, float z, float &x, float &y)

Track propagation

Track propagation.

Return (x, y) of a track transported to z (behind magnet) by calculating momentum kick in magnet.

Author: M. Schmelling

7.2

void **EHIT_init** ()

Histo initialization

Histo initialization.

Initialize HBOOK histograms to monitor photon selection code. A call to this function should be placed in the ARTE usinit() function.

Author: M. Schmelling

7.3

void **selectRCCL** ()

Rough RCCL selection

Rough RCCL selection.

A rough selection of RCCL clusters according to their width, esum and asymmetry. The RCCL::flag field is set to 0, because it will be used by the photon selection code. A selected RCCL cluster gets a flag = 1, a rejected one has flag = 0.

Author: M. Schmelling

7.4

void **assignClusters** (const ArtePointer<RecoTrack>& rtrack)

Assign clusters to tracks

Assign clusters to tracks.

Transport track behind magnet and look for matching ECAL cluster within a radius of 3 cm around track impact at $z = 1344$. Add energy of clusters up and calculate E/p. Store E/p value temporarily in $mass$ field of track (will be resetted later).

Author: M. Schmelling, adopted to BEE by T. Glebe

7.5

void **findBremsstrahlung** ()

Identify bremsstrahlung

Identify bremsstrahlung.

Straight line exprapolation of track to ECAL and search for bremsstrahlungs cluster. Store bremsstrahlungs energy in Ebrem field of RecoTrack. Function must be called after RecoTrack table is filled.

Author: M. Schmelling, adopted to BEE by T. Glebe

7.6

void **fillEcalHitTable** ()

Fill EcalHit table

Fill EcalHit table.

Fill EcalHit table, set relations between RecoTracks and remaining EcalHit objects, and perform a final PID assignment based on E/p information. For Electron candidates, perform bremsstrahlung recovery. This functions resets the RecoTrack::mass field (which was used for E/p storage). This function must be called after `findBremsstrahlung()`.

Author: M. Schmelling, adopted to BEE by T. Glebe

Class Graph