BEE

a C++ library for event analysis in HERA-B

Started by C. Borgmeier and I. Siccama,
continued by T. Glebe

Contents

1 BEE — *Initialize non-library projects.* 5

2 Clue — *The BEE event classes.* 7
- 2.1 PkgIni — *Initialization of BEE package* 9
- 2.2 ROOTini — *Class to check whether the ROOT system is initialized* 10
- 2.3 RsegC — *HERA-B component id* 10
- 2.4 RecoSegment — *Reconstructed track class* 13
- 2.5 RecoTrack — *reconstructed track class* 19
- 2.6 RecoVertex — *Reconstructed vertex* 24
- 2.7 VDS_Track — *Track in vertex detector* 27
- 2.8 RICH_Track — *Track in RICH detector* 28
- 2.9 OTR_SpacePoint — *Space point in outer tracker* 28
- 2.10 MuonTrack — *Track in MUON detector* 29
- 2.11 TrackClones — *Clone class* 30
- 2.12 EcalHit — *Photon candidates found by ECAL* 31
- 2.13 MC_Track — *Monte Carlo Track* 35
- 2.14 MC_Vertex — *Monte Carlo Vertex* 41
- 2.15 StreamObject — *write/read ROOT files* 43
- 2.16 RtEvent — *event class* 44
- 2.17 Wire — *Target wire class* 48
- 2.18 Topp — *TOPP table entry* 49
- 2.19 GTopp — *Representation of a Topp table entry* 51

3 Gate 54
- 3.1 GClonesArray — *Array of similar objects.* 54
- 3.2 GObjArray — *Array of different objects.* 56
- 3.3 GObjArrayIterator — *GObjectArray Iterator class* 58
- 3.4 ConstGObjArrayIterator — *ConstGObjectArray Iterator class* 59

4 Geometry — *Point/Vector classes* 62
- 4.1 ThreeVector — *ROOT based implementation* 62
- 4.2 ThreeVector — *CLHEP implementation* 64
- 4.3 Point3D — *ROOT based implementation* 65

4.4 Point3D — *CLHEP implementation* 67
4.5 LorentzVector — *ROOT based implementation* 68
4.6 LorentzVector — *CLHEP implementation* 70

5 ARTE — *The C++ Arte Table interface* 71
5.1 TObject — *Dummy base class* 73
5.2 ArtePointerBase — *Base class for ArtePointers* 75
5.3 ArteLink — *single link class* 77
5.4 ArteVector — *Link vector* 78
5.5 ArteRelVector — *Arte Relation Vector* 78
5.6 ArtePointer — *Arte pointer class* 79
5.7 ArteRelationPointer — *Arte Relation Pointer* 81
5.8 ArteInvOneRelationPointer — *ArteInvOneRelationPointer* 82
5.9 ArteManRelationPointer — *ArteManRelationPointer* ... 83
5.10 ArteInvManRelationPointer — *ArteInvManRelationPointer* 83
5.11 ArteRelationIterator — *ArteRelationIterator* 84
5.12 ArteTableIterator — *Arte Table Iterator class* 86
5.13 ArteTable — *Arte Table class* 88

Class Graph 91

BEE

BEE provides a basic set of event reconstruction and physics analysis functionality.

1 BEE

Initialize non-library projects.

Names

extern const PkgIni **BEESteer** — *Global object which holds XCVS information for **BEE** project.*

extern const PkgIni **ARTESteer** — *Global object which holds XCVS information for **ARTE** project.*

extern const PkgIni **CERNSteer** — *Global object which holds XCVS information for **CERN** project.*

extern const PkgIni **DocSteer** — *Global object which holds XCVS information for **doc** project.*

extern const PkgIni **GCombinerSteer** — *Global object which holds XCVS information for **gcombiner** project.*

extern const PkgIni **GRelationSteer** — *Global object which holds XCVS information for **grelation** project.*

extern const PkgIni **InterfacesSteer** — *Global object which holds XCVS information for **interfaces** project.*

extern const PkgIni

SMatrixSteer	*Global object which holds XCVS information for **smatrix** project.*

Author: T. Glebe

2 Clue

The BEE event classes.

Names

— Global Utility Functions —

const ThreeVector
WireDirection (const Wire& rhs)
return unit vector of Wire direction

const Point3D
Propagate (const RecoSegment& rs, float z)
calculate impact point at z

const double
distance (const ThreeVector& r1,
const ThreeVector& m1,
const ThreeVector& r2,
const ThreeVector& m2)
dist between two rays (unsigned)

const double
distance (const RecoSegment& rseg,
const Wire& wire)
distance between RecoSegment and Wire

const double
distance (const Wire& wire,
const RecoSegment& rseg)
distance between RecoSegment and Wire

2.1 class **PkgIni** *Initialization of BEE package* 9

extern const PkgIni
ClueSteer *Global CLUE object which holds XCVS information.*

2.2 class **ROOTini** *Class to check whether the ROOT system is initialized* . . . 10

2.3 class **RsegC** : public TObject

HERA-B component id 10

2.4 class **RecoSegment** : public TObject, public TrackIf
Reconstructed track class 13

2.5 class **RecoTrack** : public RecoSegment
reconstructed track class 19

2.6 class **RecoVertex** : public TObject, public VertexIf
Reconstructed vertex 24

2.7 class **VDS_Track** : public RecoSegment
Track in vertex detector 27

2.8 class **RICH_Track** : public RecoSegment
Track in RICH detector 28

2.9 class **OTR_SpacePoint** : public RecoSegment
Space point in outer tracker . 28

2.10 class **MuonTrack** : public RecoSegment
Track in MUON detector 29

2.11 class **TrackClones** : public TObject
Clone class 30

2.12 class **EcalHit** : public TObject, public TrackIf
Photon candidates found by ECAL 31

2.13 class **MC_Track** : public TObject, public TrackIf
Monte Carlo Track 35

2.14 class **MC_Vertex** : public TObject, public VertexIf
Monte Carlo Vertex 41

2.15 class **StreamObject** : public TObject
write/read ROOT files 43

2.16 class **RtEvent** : public TObject
event class 44

2.17 class **Wire** : public TObject
Target wire class 48

2.18 class **Topp** : public TObject, public ToppIf
TOPP table entry 49

2.19 class **GTopp** : public ToppIf
Representation of a Topp table entry 51

Clue provides the classes which are needed for the event structure plus basic classes which are needed for analyzation. The project was started by C. Borgmeier and I. Siccama.

Author: T. Glebe

2.1 class **PkgIni**

Initialization of BEE package

Public Members

— Constructors —

	PkgIni ()	*for ROOT backward compatibility*
	PkgIni (const char* nam, const char* vers, const char* dat)	
	~**PkgIni** ()	

— Access methods —

const char* const	**version** () const	*get string with version number*
const char* const	**name** () const	*get string with package name*
const char* const	**date** () const	*get birthday & time of package version*
const bool	**debug** ()	*toggle debug flag*
bool	**isDebug** () const	*return debug flag*

— Expert functions —

std::ostream&	**print** (std::ostream& os) const	*used by operator<<()*

Class to hold the XCVS information of a BEE package. Each project should create a global PkgIni object which is initialized with the prjects XCVS information.

Author: T. Glebe

2.2

class **ROOTini**

Class to check whether the ROOT system is initialized

Public Members

ROOTini ()

Class to check whether the ROOT system is initialized. If not, initialize it.

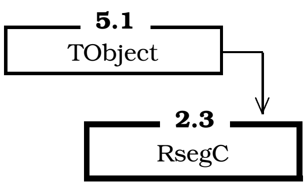
Author: T. Glebe

2.3

class **RsegC** : public TObject

HERA-B component id

Inheritance



Public Members

	— Constructors —	
	RsegC ()	
	RsegC (const unsigned int rhs)	
	~**RsegC** ()	
RsegC&	**operator=** (const RsegC& rhs)	
	— Access methods —	
unsigned int	**cmp** () const	*return bit pattern*
bool	**vxd** () const	*bit 0 (0x1): Vertex Detector*
bool	**patt** () const	*bit 1 (0x2): Pattern Tracker*
bool	**magt** () const	*bit 2 (0x4): Magnet Tracker*
bool	**muon** () const	*bit 3 (0x8): Muon Detector*
bool	**ecal** () const	*bit 4 (0x10): Ecal*
bool	**tar** () const	*bit 5 (0x20): Target Constraint*
bool	**trit** () const	*bit 6 (0x40): Trigger Tracker, betw RICH and ECAL*
bool	**bit7** () const	*bit 7 (0x80): not assigned yet*
bool	**slte** () const	*bit 8 (0x100): match SLT trigger tracks*
bool	**sltmu** () const	*bit 9 (0x200): match SLT trigger tracks*
bool	**bit10** () const	*bit 10 (0x400): not assigned yet*
bool	**bit11** () const	*bit 11 (0x800): not assigned yet*
bool	**bit12** () const	*bit 12 (0x1000): not assigned yet*
bool	**bit13** () const	*bit 13 (0x2000): not assigned yet*
bool	**bit14** () const	*bit 14 (0x4000): not assigned yet*
bool	**bit15** () const	*bit 15 (0x8000): not assigned yet*

bool	**bit16** () const	*bit 16 (0x10000): not assigned yet*
bool	**bit17** () const	*bit 17 (0x20000): not assigned yet*
bool	**bit18** () const	*bit 18 (0x40000): Match SVD-PC (marple RSEG only)*
bool	**bit19** () const	*bit 19 (0x80000): not assigned yet*
bool	**clonerm** () const	*bit 20 (0x100000): Used by Cloneremove.C*
bool	**bit21** () const	*bit 21 (0x200000): not assigned yet*
bool	**bit22** () const	*bit 22 (0x400000): not assigned yet*
bool	**bit23** () const	*bit 23 (0x800000): not assigned yet*
bool	**slt** () const	*bit 24 (0x1000000): SLT seeded segment*
bool	**trd** () const	*bit 25 (0x2000000): TRD straw device type*
bool	**rich** () const	*bit 26 (0x4000000): Cherenkov Counter device type*
bool	**ptc** () const	*bit 27 (0x8000000): HiPT Chamber device type*
bool	**msg** () const	*bit 28 (0x10000000): MSGC device type*
bool	**wch** () const	*bit 29 (0x20000000): honeycomb wire chamber device type*
bool	**sil** () const	*bit 30 (0x40000000): silicon strip device type*
bool	**bitsign** () const	*bit 31 (0x80000000): Kill bit, set if segment is deleted*

— Operators —

bool **operator==** (unsigned int rhs) const

bool **operator!=** (unsigned int rhs) const

const RsegC&

operator—= (const unsigned int rhs)

const RsegC&
operator&= (const unsigned int rhs)

— Expert functions —

std::ostream&
print (std::ostream& os) const

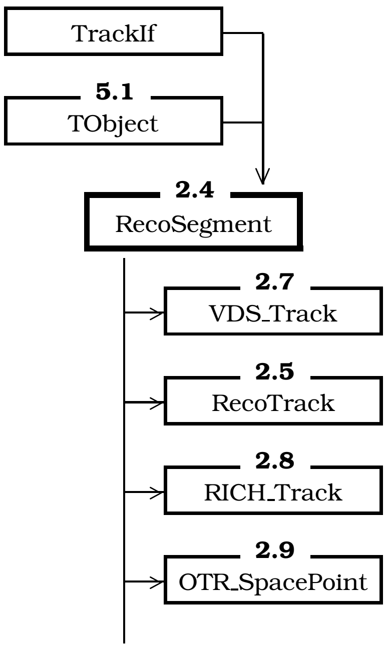
Author: T. Glebe

2.4

class **RecoSegment** : public TObject, public TrackIf

Reconstructed track class

Inheritance



2.10
MuonTrack

Public Members

— Constructors —

RecoSegment ()

RecoSegment (const RecoSegment& rhs)

˜**RecoSegment** ()

const RecoSegment&
operator= (const RecoSegment& rhs)

— ARTE only functions —

template <class T> void
convert (T rtra) *fill RecoSegment from RSEG or RTRA*

RecoSegment (const ArtePointer<RTRA>& r)
create object from RTRA table

RecoSegment (const ArtePointer<RSEG>& r)
create object from RSEG table

ArtePointer <RTRA>
rtra () const *return RTRA pointer*

void **set_rtra** (const ArtePointer<RTRA>& rtra)
set RTRA pointer

unsigned int
rtra () const *not available in ARTE*

— Operators —

bool **operator==** (const TrackIf& rhs) const
compare TrackIf pointers

bool **operator==** (const RecoSegment& rhs) const
compare Rtra value

bool **operator**< (const RecoSegment& rhs) const
compare Rtra value

— TrackIf methods —

float **x** () const *x of track*

float **y** () const *y of track*

float **x** (float z) const *linear extrapolation of x to z*

float	**y** (float z) const	*linear extrapolation of y to z*
float	**z** () const	*z of track*
float	**zorig** () const	*in case of propagation: z before propagation*
float	**tx** () const	*dx/dz slope of track (t_x)*
float	**ty** () const	*dy/dz slope of track (t_y)*
float	**p** () const	*momentum of track (p)*
float	**pt** () const	*transverse Track momentum* $p_t = p \cdot \sqrt{(t_x^2 + t_y^2)/(t_x^2 + t_y^2 + 1)}$
float	**pz** () const	*momentum along z:* $p/\sqrt{t_x^2 + t_y^2 + 1}$
float	**chi2** () const	*χ^2 of track fit*
unsigned	short int **ndf** () const	*degrees of freedom*
float	**phi** () const	*azimuthal angle ϕ [deg]*
float	**theta** () const	*polar angle $\theta = \cos^{-1}(e_z)$ [deg]*
float	**eta** () const	*rapidity $\eta = -\log\tan(\theta/2.)$ rapidity*
int	**charge** () const	*particle charge: -1 for neg. +1 for pos. & neutrals*
virtual float	**energy** (double mass = 0.) const	$E = \sqrt{m^2 + p^2}$
float	**xf** (double mass = 0.) const	$x_f = (p_z - E(m) \cdot \sqrt{1 - \frac{2m_p}{920GeV/c + m_p}})/m_p$, m_p *= Proton mass,* m *= Particle mass*
float	**rap** (double mass = 0.) const	*Rapidity* $y = \frac{1}{2}\ln(\frac{E+p_z}{E-p_z})$
	— Access methods —	
float	**trprob** () const	*upper tail prob. of track χ^2*
void	**set_chi2** (const float chi2)	*set χ^2*
void	**set_nf** (const UShort_t nf)	

		set degrees of freedom
float	**pinv** () const	*inverse momentum of track*
float	**px** () const	$t_x \cdot p_z$
float	**py** () const	$t_y \cdot p_z$
RsegC	**cmp** () const	*Component id*
int	**isRICH** () const	*is RICH component id set?*
int	**isOTR** () const	*is OTR component id set?*
int	**isVDS** () const	*is VDS component id set?*
int	**isECAL** () const	*is ECAL component id set?*
void	**set_x** (const float x)	*set x parameter*
void	**set_y** (const float y)	*set y parameter*
void	**set_z** (const float z)	*set z parameter*
void	**set_tx** (const float tx)	*set t_x parameter*
void	**set_ty** (const float ty)	*set t_y parameter*
void	**set_p** (const float p)	*set momentum p*
float	**cov_x** (double dz=0.) const	*get* $\sigma_x^2\|_{z+dz}$
float	**cov_x_y** (double dz=0.) const	*get* $\sigma_x \times \sigma_y\|_{z+dz}$
float	**cov_y** (double dz=0.) const	*get* $\sigma_y^2\|_{z+dz}$
float	**cov_x_tx** (double dz=0.) const	*get* $\sigma_x \times \sigma_{t_x}\|_{z+dz}$
float	**cov_y_tx** (double dz=0.) const	*get* $\sigma_y \times \sigma_{t_x}\|_{z+dz}$
float	**cov_tx** () const	*get* $\sigma_{t_x}^2$
float	**cov_x_ty** (double dz=0.) const	*get* $\sigma_x \times \sigma_{t_x}\|_{z+dz}$
float	**cov_y_ty** (double dz=0.) const	

		get $\sigma_y \times \sigma_{t_y}\|_{z+dz}$
float	**cov_tx_ty** () const	*get* $\sigma_{t_x} \times \sigma_{t_y}$
float	**cov_ty** () const	*get* $\sigma_{t_y}^2$
float	**cov_x_p** (double dz=0.) const	*get* $\sigma_x \times \sigma_p\|_{z+dz}$
float	**cov_y_p** (double dz=0.) const	*get* $\sigma_y \times \sigma_p\|_{z+dz}$
float	**cov_tx_p** () const	*get* $\sigma_{t_x} \times \sigma_p$
float	**cov_ty_p** () const	*get* $\sigma_{t_y} \times \sigma_p$
float	**cov_p** () const	*get* σ_p^2
void	**set_cov_x** (const float c)	*set* σ_x^2
void	**set_cov_x_y** (const float c)	*set* $\sigma_x \times \sigma_y$
void	**set_cov_y** (const float c)	*set* σ_y^2
void	**set_cov_x_tx** (const float c)	*set* $\sigma_x \times \sigma_{t_x}$
void	**set_cov_y_tx** (const float c)	*set* $\sigma_y \times \sigma_{t_x}$
void	**set_cov_tx** (const float c)	*set* $\sigma_{t_x}^2$
void	**set_cov_x_ty** (const float c)	*set* $\sigma_x \times \sigma_{t_x}$
void	**set_cov_y_ty** (const float c)	*set* $\sigma_y \times \sigma_{t_y}$
void	**set_cov_tx_ty** (const float c)	*set* $\sigma_{t_x} \times \sigma_{t_y}$
void	**set_cov_ty** (const float c)	*set* $\sigma_{t_y}^2$
void	**set_cov_x_p** (const float c)	*set* $\sigma_x \times \sigma_p$
void	**set_cov_y_p** (const float c)	*set* $\sigma_y \times \sigma_p$
void	**set_cov_tx_p** (const float c)	

set $\sigma_{t_x} \times \sigma_p$

void **set_cov_ty_p** (const float c)
set $\sigma_{t_y} \times \sigma_p$

void **set_cov_p** (const float c)
set σ_p^2

— Expert functions —

void **collect** (vector<TrackIf*>& c) const
collect pointers

void **propagate** (const double zz)
propagate track to z

void **sPinvToP** () $\sigma(Q/p) \rightarrow \sigma(p)$

void **sPToPinv** () $\sigma(p) \rightarrow \sigma(Q/p)$

SVector <double, 3>
xvec () const $\vec{x} = (x,\ y,\ z)$

SVector <double, 3>
tvec () const $\vec{t} = (t_x,\ t_y,\ 1)$

SVector <double, 3>
evec () const $\vec{e} = (e_x,\ e_y,\ ez)$ *unit vector along* $\vec{p}$

SVector <double, 3>
pvec () const $\vec{p} = (p_x,\ p_y,\ p_z)$

bool **FillCINV** () *fill COV_ matrix and compute inverse cov. matrix*

const SMatrix <double, 5> &
COV () const *return covariance matrix **Cov***

const SMatrix <double, 5> &
CINV () const *return inverse covariance matrix* $\boldsymbol{Cov}^{-1}$

virtual bool
isValid () const *is track valid?*

bool **PrepareForVertexing** ()
Prepare reconstructed track for vertexing with SVertex package: propagate track parameters and errors to $z = 0$, *fill inverse covariance matrix*

virtual void

valid () *mark track as valid*

virtual void
invalid () *mark track as invalid*

std::ostream&
print (std::ostream&) const
used by operator<<()

std::ostream&
print_cov (std::ostream&) const
print covariance matrix

— GRelation relations —

GRelationList <SPatternMatch>
pmatch *RecoTrack → PatternMatch*

const GRelationList <SPatternMatch> &
Pmatch () const

GRelationList <SPatternMatch> &
Pmatch ()

GRelationList <SPatternMatch> &
container (const SPatternMatch& p)

void **Print** (Option_t *option="")
only available in ROOT

Reconstructed track class. In contrast to `ARTE`, the momentum covariance matrix elements are computed with the momentum p and not with Q/p. The reason is that they enter in this way into the vertexing (Vt based vertexing codes). The transformation $cov_{ij}(Q/p) \rightarrow cov_{ij}(p)$ is done before writing the `BEE` files in the `ARTE` project `cluearte`. The `RecoSegment` memberfunctions `sPinvToP()` and `sPtoPinv()` do the appropriate transformation in the respective direction.

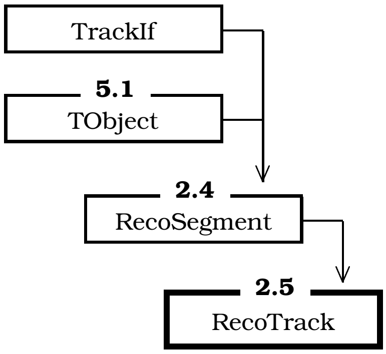
Author: C. Borgmeier, T. Glebe

2.5

class **RecoTrack** : public RecoSegment

reconstructed track class

Inheritance



Public Members

	— Constructors —	
	RecoTrack ()	
	RecoTrack (const RecoTrack& rt)	
	~**RecoTrack** ()	
	RecoTrack (const ArtePointer<RTRA>& rtra, double mass = 0)	*not available in ROOT*
	RecoTrack (const ArtePointer<RSEG>& rseg)	*not available in ROOT*
const RecoTrack&	**operator=** (const RecoTrack& rhs)	
	— Access methods —	
int	**pid** () const	*particle identification flag*
void	**set_pid** (int pid)	*set particle identification flag*
float	**le** () const	*global likelihood of electron hypothesis*
float	**lmu** () const	*global likelihood of muon hypothesis*
float	**lpi** () const	*global likelihood of pion hypothesis*
float	**lk** () const	*global likelihood of kaon hypothesis*

float	**lp** () const	*global likelihood of proton hypothesis*
int	**hsi** () const	*number of hits in the SVD*
int	**hin** () const	*number of hits in the inner tracker*
int	**hou** () const	*number of hits in the outer tracker*
int	**hri** () const	*number of hits in the RICH*
int	**hca** () const	*number of hits in the calorimeter*
int	**htr** () const	*number of hits in the TRD*
int	**hmu** () const	*number of hits in the muon chambers*
int	**fit** () const	*fit flag*
float	**mass** () const	*particle mass* m
void	**set_mass** (const float newMass)	*set particle mass* m
float	**energy** (double mass = 0.) const	$E = \sqrt{m^2 + p^2}$
float	**ebrems** () const	*possible Bremsstrahlung correction*
void	**set_ebrems** (const float ener)	*set bremsstrahlungs energy*
void	**ebrems_recovery** ()	*perform bremsstrahlungs recovery*
float	**cer** () const	*Cerenkov angle of associated RICH ring*
int	**mommask** () const	

Momentum determination flag:
1: estimated by REFIT on all hits assigned,
2: estimated by REFIT with rejection of hits in magnet,
3: estimated in VDS-ECAL match,
4: obtained during magnet tracking,
5: target constrain of OTR-PC segment,
6: PC-kick in the magnet is used for estimation,
7: estimated in VDS-RICH match,
8: estimated in VDS-PC match,
9: assigned by CATS for VDS segment

const ArtePointer <Topp>
topp () const *link to Topp table*

void **set_topp** (const ArtePointer<Topp>& topp)
set link to Topp table

— Expert functions —

std::ostream&
print (std::ostream& os) const
used by operator<<()

void **print_all** (std::ostream& os) const
print all available information

void **printMC** () const *print related MC tracks*

— CLUE relations —

ClueRelationList <MC_Track>
mtrack *RecoTrack ↔ MC_Track relation*

const ClueRelationList <MC_Track> &
mTrack () const

ClueRelationList <MC_Track> &
mTrack ()

ClueRelationList <MC_Track> &
container (const ArtePointer<MC_Track>& p)

ClueRelationList <TrackClones>

clones *RecoTrack ↔ TrackClones*

const ClueRelationList <TrackClones> &
Clones () const

ClueRelationList <TrackClones> &
Clones ()

ClueRelationList <TrackClones> &
container (const ArtePointer<TrackClones>& p)

ClueRelationList <RecoVertex>
rvertex *RecoTrack ↔ RecoVertex*

const ClueRelationList <RecoVertex> &
rVertex () const

ClueRelationList <RecoVertex> &
rVertex ()

ClueRelationList <RecoVertex> &
container (const ArtePointer<RecoVertex>& p)

ClueRelationList <EcalHit>
ehit *RecoTrack ↔ EcalHit*

const ClueRelationList <EcalHit> &
Ehit () const

ClueRelationList <EcalHit> &
Ehit ()

ClueRelationList <EcalHit> &
container (const ArtePointer<EcalHit>& p)

void **RelStreamer** (TBuffer& R__b,
const Version_t& StreamVersion)
only availabe in ROOT: call Streamer of relation container

void **Print** (Option_t* o="")
only available in ROOT

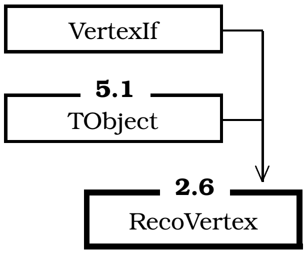
Author: C. Borgmeier, T. Glebe

2.6

class **RecoVertex** : public TObject, public VertexIf

Reconstructed vertex

Inheritance



Public Members

— Constructors —

	RecoVertex ()	*default constructor*
	RecoVertex (const RecoVertex& rhs)	*copy constructor*
	˜**RecoVertex** ()	*destructor*
	RecoVertex (const ArtePointer<RVER>& rver)	*from ArtePointer*
const RecoVertex&	**operator=** (const RecoVertex& rhs)	

— Operators —

bool	**operator**< (const RecoVertex& rhs) const	$z_1 < z_2$

— VertexIf access methods —

float	**vx** () const	x *of vertex*
float	**vy** () const	y *of vertex*
float	**vz** () const	z *of vertex*
float	**chi2** () const	χ^2 *of vertex fit*
unsigned short int	**ndf** () const	*degrees of freedom of vertex fit*
float	**prob** () const	*probability of the vertex fit*
unsigned short int		

	ntracks () const	*no of tracks in vertex*
const TrackIf*	**track** (unsigned int i) const	*returns pointer to track in vertex (slow access via relation!)*
const TrackIf*&	**track** (unsigned int i)	*returns null pointer + prints warning*
float	**vtx_cov_x** () const	σ_x^2 *of vertex*
float	**vtx_cov_y** () const	σ_y^2 *of vertex*
float	**vtx_cov_z** () const	σ_z^2 *of vertex*
	— Access methods —	
bool	**valid** () const	*is vertex valid?*
void	**invalid** ()	*mark vertex as invalid*
void	**set_valid** ()	*mark vertex as valid*
float	**xerr** () const	*error in* x
float	**yerr** () const	*error in* y
float	**zerr** () const	*error in* z
float	**mass** () const	*invariant mass in vertex*
SVector <double, 3>	**vpos** () const	$\vec{v} = (x,\ y,\ z)$
const SMatrix <double, 3> &	**VCOV** () const	*vertex cov. matrix*
std::ostream&	**print** (std::ostream& os) const	*used by operator<<()*
std::ostream&	**print_cov** (std::ostream& os) const	*print covariance matrix*
void	**set_chi2** (float chi)	
void	**set_ndf** (int nd)	
void	**set_mass** (float ms)	

float **vtx_cov_x_y** () const
$\sigma_x \times \sigma_y$ of vertex

float **vtx_cov_x_z** () const
$\sigma_x \times \sigma_z$ of vertex

float **vtx_cov_y_z** () const
$\sigma_y \times \sigma_z$ of vertex

void **set_vtx_cov_x** (const float c)
set σ_x^2 of vertex

void **set_vtx_cov_x_y** (const float c)
set $\sigma_x \times \sigma_y$ of vertex

void **set_vtx_cov_y** (const float c)
set σ_y^2 of vertex

void **set_vtx_cov_x_z** (const float c)
set $\sigma_x \times \sigma_z$ of vertex

void **set_vtx_cov_y_z** (const float c)
set $\sigma_y \times \sigma_z$ of vertex

void **set_vtx_cov_z** (const float c)
set σ_z^2 of vertex

void **set_xyz** (const float x, const float y, const float z)
set vertex position

— CLUE relations —

ClueRelationList <RecoTrack>
out *RecoVertex ↔ RecoTrack (relations to outgoing tracks)*

const ClueRelationList <RecoTrack> &
Out () const

ClueRelationList <RecoTrack> &
Out ()

ClueRelationList <RecoTrack> &
container (const ArtePointer<RecoTrack>& p)

void **RelStreamer** (TBuffer& R__b, const Version_t& StreamVersion)
only availabe in ROOT: call Streamer of relation container

void **Clear** (Option_t *option="")
only available in ROOT

void **Print** (Option_t *option="")

only available in ROOT

void **Paint** (Option_t *option="")
only available in ROOT

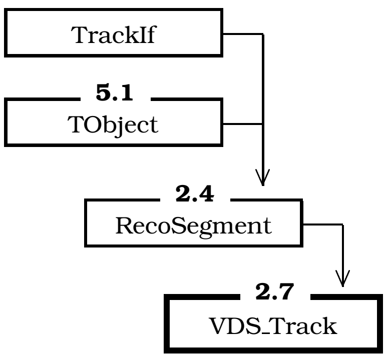
Author: C. Borgmeier, T. Glebe

2.7

class **VDS_Track** : public RecoSegment

Track in vertex detector

Inheritance



Public Members

VDS_Track ()

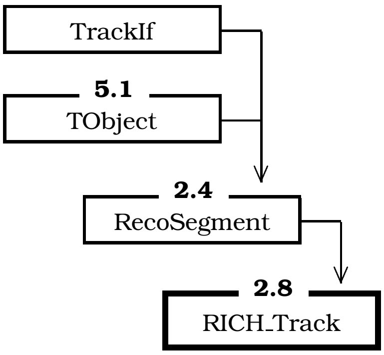
VDS_Track (const ArtePointer<RSEG>& rseg)

Author: C. Borgmeier, T. Glebe

2.8 class **RICH_Track** : public RecoSegment

Track in RICH detector

Inheritance



Public Members

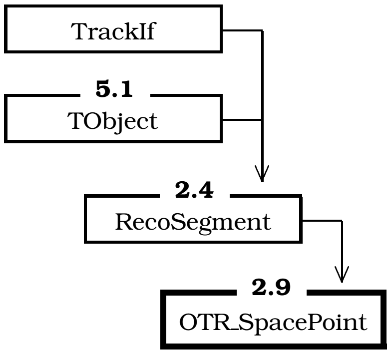
	RICH_Track ()
	RICH_Track (const ArtePointer<RSEG>& rseg)
float	**radius**
float	**signalPhotons**
float	**backgndPhotons**
float	**fraction**

Author: C. Borgmeier, T. Glebe

2.9 class **OTR_SpacePoint** : public RecoSegment

Space point in outer tracker

Inheritance



Public Members

OTR_SpacePoint ()

OTR_SpacePoint (const ArtePointer<RSEG>& rseg)

int **nHits**

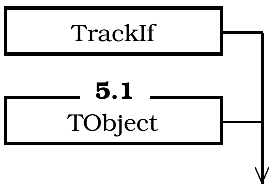
Author: C. Borgmeier, T. Glebe

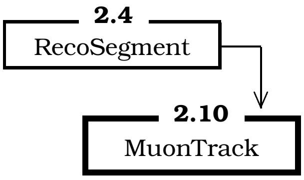
2.10

class **MuonTrack** : public RecoSegment

Track in MUON detector

Inheritance





Public Members

MuonTrack ()

MuonTrack (const ArtePointer<RSEG>& rseg)

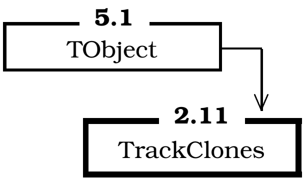
Author: C. Borgmeier, T. Glebe

2.11

class **TrackClones** : public TObject

Clone class

Inheritance



Public Members

— Constructors —

TrackClones ()

TrackClones (const TrackClones& rhs)

TrackClones (const RsegC& t)

~**TrackClones** ()

const TrackClones&

operator= (const TrackClones& rhs)

— Access methods —

RsegC **type** () const *type corresponds to rseg->cmp; type of incompatibility*

— CLUE relations —

ClueRelationList <RecoTrack>
rtrack *TrackClones ↔ RecoTrack*

const ClueRelationList <RecoTrack> &
rTrack () const

ClueRelationList <RecoTrack> &
rTrack ()

ClueRelationList <RecoTrack> &
container (const ArtePointer<RecoTrack>& p)

void **RelStreamer** (TBuffer& R__b,
const Version_t& StreamVersion)
only availabe in ROOT: call Streamer of relation container

Class for tracks sharing segments.

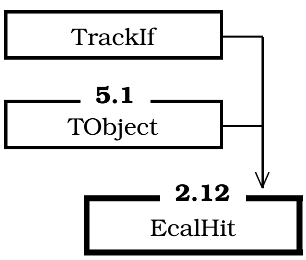
Author: C. Borgmeier, T. Glebe

2.12

class **EcalHit** : public TObject, public TrackIf

Photon candidates found by ECAL

Inheritance



Public Members

	— Constructors —	
	EcalHit ()	
	EcalHit (const ArtePointer<RCCL>& rccl)	*only available with ARTE*
	EcalHit (const EcalHit& rhs)	
	~**EcalHit** ()	
EcalHit&	**operator=** (const EcalHit& rhs)	
	— TrackIf methods —	
float	**x** () const	x *position of cluster*
float	**y** () const	y *position of cluster*
float	**z** () const	z *position of cluster*
float	**x** (float z) const	*Track* x *position at* z
float	**y** (float z) const	*Track* y *position at* z
float	**tx** () const	t_x *of cluster calculated from point* x_0
float	**ty** () const	t_y *of cluster calculated from point* x_0
float	**p** () const	*Photon track momentum* $\|\vec{p}\| = E_{cluster}$
float	**pt** () const	*transverse Photon track momentum* $p_t = p \cdot \sqrt{(t_x^2 + t_y^2)/(t_x^2 + t_y^2 + 1)}$
float	**pz** () const	*Photon track momentum along* z, $p_z = p/\sqrt{t_x^2 + t_y^2 + 1}$
float	**phi** () const	*azimuthal angle* ϕ *[deg]*
float	**theta** () const	*polar angle* $\theta = \cos^{-1}(e_z)$ *[deg]*
float	**eta** () const	*pseudorapidity* $\eta = -\log\tan(\theta/2.)$
int	**charge** () const	*Photon charge: 0*
float	**energy** (double rm = 0.) const	*Ecal cluster energy* $E_{cluster}$ *(rm argument has no effect)*
float	**xf** (double rm = 0.) const	

		Feynman Variable $x_F = \frac{E+p_z}{(E+p_z)_{max}}$
float	**rap** (double rm = 0.) const	*Rapidity* $y = \frac{1}{2}\ln(\frac{E+p_z}{E-p_z})$
SVector <double, 3>	**xvec** () const	$\vec{x} = (x,\ y,\ z)$
SVector <double, 3>	**tvec** () const	$\vec{t} = (t_x,\ t_y,\ 1)$
SVector <double, 3>	**evec** () const	$\vec{e} = (e_x,\ e_y,\ ez)$ *unit vector along* $\vec{p}$
SVector <double, 3>	**pvec** () const	$\vec{p} = (p_x,\ p_y,\ p_z)$
bool	**isValid** () const	*is ECAL cluster valid?*
const SMatrix <double, 5> &	**COV** () const	*return 0 matrix*
const SMatrix <double, 5> &	**CINV** () const	*return 0 matrix*
float	**cov_x** (double dz=0.) const	*get* $\sigma_x^2\|_{z+dz}$
float	**cov_y** (double dz=0.) const	*get* $\sigma_y^2\|_{z+dz}$
float	**cov_tx** () const	*get* $\sigma_{t_x}^2 = \sigma_x^2/(z-z_0)^2$
float	**cov_ty** () const	*get* $\sigma_{t_y}^2 = \sigma_y^2/(z-z_0)^2$
float	**cov_p** () const	*get* σ_E^2
void	**propagate** (const double dz)	*propagate track + errors*
	— dummy functions —	
float	**chi2** () const	*always return 0*
unsigned short int	**ndf** () const	*always return 0*
	— EcalHit access functions —	
float	**zorig** () const	*return Z before propagation*
float	**cov_x_tx** (double dz=0.) const	*get* $\sigma_x \times \sigma_{t_x}\|_{z+dz}$
float	**cov_y_ty** (double dz=0.) const	

get $\sigma_y \times \sigma_{t_y}|_{z+dz}$

— Operators —

bool	**operator==** (const TrackIf& rhs) const	*comparison between Track objects*

— Expert methods —

void	**valid** ()	*mark cluster as valid*
void	**invalid** ()	*mark cluster as invalid*
void	**collect** (vector<TrackIf*>& c) const	*collect TrackIf pointers*
std::ostream&	**print** (std::ostream&) const	*used by operator<<()*

— EcalHit Access methods —

unsigned int	**width** () const	*width of cluster*
float	**x_c** () const	x_c*: cluster* x
float	**y_c** () const	y_c*: cluster* y
float	**z_c** () const	z_c*: cluster* z
float	**px** () const	$t_x \cdot p_z$
float	**py** () const	$t_y \cdot p_z$
float	**Ez** () const	$E/\sqrt{t_x^2 + t_y^2 + 1}$
float	**Et** () const	$E \cdot \sqrt{(t_x^2 + t_y^2)/(t_x^2 + t_y^2 + 1)}$
SVector <double, 3>	**xc_vec** () const	$\vec{x} = (x_c,\ y_c,\ z_c)$
SVector <double, 3>	**Evec** () const	$\vec{E} = (E_x,\ E_y,\ E_z)$
void	**calc_track** (const SVector<double, 3>& x0)	*calc track parameters & cov. matrix using photon origin* $\vec{x}_0$
bool	**calc_CINV** ()	*calculate inverse cov. matrix*

— CLUE relations —

ClueRelationList <RecoTrack>	**rtrack**	*EcalHit* $\leftrightarrow$ *RecoTrack*

const ClueRelationList <RecoTrack> &

rTrack () const

ClueRelationList <RecoTrack> &
rTrack ()

ClueRelationList <RecoTrack> &
container (const ArtePointer<RecoTrack>& p)

void **RelStreamer** (TBuffer& R__b,
const Version_t& StreamVersion)
only availabe in ROOT: call Streamer of relation container

— ROOT only functions —

void **Print** (Option_t * = "")

void **Paint** (Option_t *option)

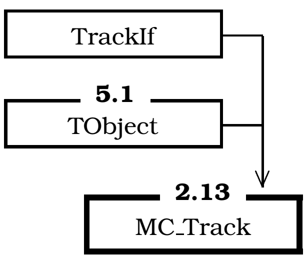
Author: T. Glebe

2.13

class **MC_Track** : public TObject, public TrackIf

Monte Carlo Track

Inheritance



Public Members

— Constructors —

MC_Track ()

MC_Track (const MC_Track& rhs)

MC_Track (const ArtePointer<MTRA>& t)
constructor reading from ARTE table

~**MC_Track** ()

MC_Track&
operator= (const MC_Track&)

— TrackIf Access methods —

float	**x** () const	*Track x position*
float	**y** () const	*Track y position*
float	**x** (float z) const	*Track x position at z*
float	**y** (float z) const	*Track y position at z*
float	**z** () const	*Track z position*
float	**tx** () const	*Track slope t_x*
float	**ty** () const	*Track slope t_y*
float	**p** () const	*Track momentum p*
float	**pt** () const	*transverse Track momentum p_t ($\perp$ to z)*
float	**pz** () const	*Track momentum along z*
float	**chi2** () const	*dummy function: returns 0*
unsigned short int	**ndf** () const	*dummy function: returns 0*
float	**phi** () const	*azimuthal angle ϕ [deg]*
float	**theta** () const	*polar angle $\theta = \cos^{-1}(e_z)$ [deg]*
float	**eta** () const	*pseudorapidity $\eta = -\log\tan(\theta/2.)$*
int	**charge** () const	*track charge*
float	**energy** (double rm = 0.) const	*track energy $E = \sqrt{m^2+p^2}$, rm is ignored*
float	**xf** (double rm = 0.) const	*Feynman Variable $x_F = \frac{E+p_z}{(E+p_z)_{max}}$, rm is ignored*
float	**rap** (double rm = 0.) const	

		Rapidity $y = \frac{1}{2}\ln(\frac{E+p_z}{E-p_z})$, *rm is ignored*
SVector <double, 3>	**xvec** () const	$\vec{x} = (x,\ y,\ z)$
SVector <double, 3>	**tvec** () const	$\vec{t} = (t_x,\ t_y,\ 1)$
SVector <double, 3>	**evec** () const	$\vec{e} = (e_x,\ e_y,\ ez)$ *unit vector along* $\vec{p}$
SVector <double, 3>	**pvec** () const	$\vec{p} = (p_x,\ p_y,\ p_z)$
const SMatrix <double, 5> &	**COV** () const	*dummy function: returns zero matrix*
const SMatrix <double, 5> &	**CINV** () const	*dummy function: returns zero matrix*
float	**cov_x** (double dz=0.) const	*dummy function: return 0*
float	**cov_y** (double dz=0.) const	*dummy function: return 0*
float	**cov_tx** () const	*dummy function: return 0*
float	**cov_ty** () const	*dummy function: return 0*
float	**cov_p** () const	*dummy function: return 0*
bool	**isValid** () const	*dummy function: always return true*
	— Operators —	
bool	**operator==** (const TrackIf& rhs) const	*comparison between Track objects*
bool	**operator==** (const GTopp& rhs) const	*compare with GTOPP*
	— Expert methods —	
void	**valid** ()	*dummy function: do nothing*
void	**invalid** ()	*dummy function: do nothing*
void	**collect** (vector<TrackIf*>& c) const	

collect TrackIf pointers

void **propagate** (const double dz)
dummy function: do nothing

std::ostream&
print (std::ostream&) const
used by operator<<()

— Access methods —

float **mass** () const *return topp mass m*

const char* const
name () const *return topp name*

unsigned int
flag () const *MC flag: 0: just created; not yet processed*
1: expected to decay according to QMTRA_LENG
2: decayed resonance or electromagnetic decay (production vertex = decay vertex)
3: decay with non-zero track length
4: track ends in a electromagnetic or hadronic interaction
5: track leaves the detector volume
6: track stops inside detector volume
7: track curls up inside detector volume
8: track stopped by user

const ToppIf&
topp () const *link to Topp table*

const ArtePointer <MC_Vertex>
bVertex () const *pointer to begin vertex*

void **set_bVertex** (const ArtePointer<MC_Vertex>& p)
set pointer to begin vertex

const ArtePointer <MC_Track>
mother () const *pointer to begin vertex*

void **set_mother** (const ArtePointer<MC_Track>& t) *set pointer to begin vertex*

2.13.1 void **print_decay** (unsigned int level, char opt = ' ', const ArtePointer<MC_Track> mtra = 0, unsigned int lvl=0) const
print MC decay chain 40

unsigned int
outsize () const *no of outgoing tracks*

— CLUE relations —

ClueRelationList <RecoTrack>
rtrack *MC_Track ↔ RecoTrack relation*

const ClueRelationList <RecoTrack> &
rTrack () const

ClueRelationList <RecoTrack> &
rTrack ()

ClueRelationList <RecoTrack> &
container (const ArtePointer<RecoTrack>& p)

ClueRelationList <MC_Vertex>
endVtx *MC_Track ↔ MC_Vertex relation (end vertices)*

const ClueRelationList <MC_Vertex> &
EndVtx () const

ClueRelationList <MC_Vertex> &
EndVtx ()

ClueRelationList <MC_Vertex> &
container (const ArtePointer<MC_Vertex>& p)

ClueRelationList <MC_Track>
endTrack *MC_Track ↔ MC_Track relation (outgoing tracks)*

const ClueRelationList <MC_Track> &
EndTrack () const

ClueRelationList <MC_Track> &
EndTrack ()

ClueRelationList <MC_Track> &
container (const ArtePointer<MC_Track>& p)

— ROOT only functions —

void **RelStreamer** (TBuffer& R__b, const Version_t& StreamVersion)
call Streamer of relation container

void **Print** (Option_t *option="")

2.13.2 virtual void **Paint** (Option_t *option="")
Do not call this function directly, use Draw() instead . . . 41

Monte Carlo Track. In case of prompt decays (e.g. $\phi \rightarrow \mathrm{K}^{+} + \mathrm{K}^{-}$) no begin- or end-vertices exist in ARTE (to reduce event size). Therefore we need in addition to begin- and end-vertex relations relations to mother- and outgoing tracks. In case of decays which have a travel distance this introduces some redundancy: the mother & outgoing tracks can be accessed either directly or via the begin- and end-vertices.

Author: C. Borgmeier, T. Glebe

2.13.1

void **print_decay** (unsigned int level, char opt = ' ', const ArtePointer<MC_Track> mtra = 0, unsigned int lvl=0) const

print MC decay chain

print MC decay chain. Options for additional information: f: print track flag m: print ArtePointer<MC_Track> r: print ArtePointer<RecoTrack> (matches to reconstructed tracks)

2.13.2

virtual void **Paint** (Option_t *option="")

Do not call this function directly, use Draw() instead

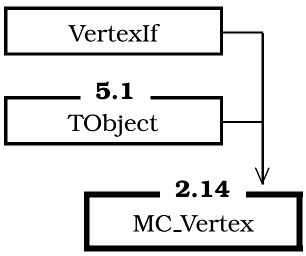
Do not call this function directly, use Draw() instead. Before you can draw, create a TView.

2.14

class **MC_Vertex** : public TObject, public VertexIf

Monte Carlo Vertex

Inheritance



Public Members

MC_Vertex ()

MC_Vertex (const MC_Vertex& rhs)

MC_Vertex (const ArtePointer<MVER>& mv)
only available in ARTE

~**MC_Vertex** ()

MC_Vertex&
operator= (const MC_Vertex&)

std::ostream&
print (std::ostream&) const
used by operator<<()

— VertexIf access methods —

float	**vx** () const	x *of vertex*
float	**vy** () const	y *of vertex*
float	**vz** () const	z *of vertex*

float **chi2** () const *dummy function: return 0*

unsigned short int
ndf () const *dummy function: return 0*

float **prob** () const *dummy function: return 1*

unsigned short int
ntracks () const *no of tracks in vertex*

const TrackIf*
track (unsigned int i) const
returns pointer to track in vertex (slow access via relation!)

const TrackIf*&
track (unsigned int i)
returns null pointer + prints warning

SVector <double, 3>
vpos () const $\vec{x} = (x,\ y,\ z)$

const SMatrix <double, 3> &
VCOV () const *dummy function: return zero matrix*

float **vtx_cov_x** () const
dummy function: return 0

float **vtx_cov_y** () const
dummy function: return 0

float **vtx_cov_z** () const
dummy function: return 0

— Access methods —

const ArtePointer <MC_Track>
mother () const *return pointer to mother track*

void **set_mother** (const ArtePointer<MC_Track>& p)
set pointer to mother track

— CLUE relations —

ClueRelationList <MC_Track>
out *MC_Vertex ↔ MC_Track (relations to outgoing tracks)*

const ClueRelationList <MC_Track> &
Out () const

ClueRelationList <MC_Track> &

Out ()

ClueRelationList <MC_Track> &
container (const ArtePointer<MC_Track>& p)

void **RelStreamer** (TBuffer& R__b, const Version_t& StreamVersion)
only availabe in ROOT: call Streamer of relation container

void **Print** (Option_t *option="")
only available in ROOT

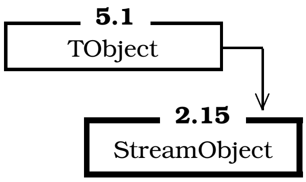
void **Paint** (Option_t *option="")
only available in ROOT

Author: C. Borgmeier, T. Glebe

2.15 class **StreamObject** : public TObject

write/read ROOT files

Inheritance



Public Members

StreamObject ()

StreamObject (RtEvent*)

Helper class to write and read CLUE events from/to ROOT files.

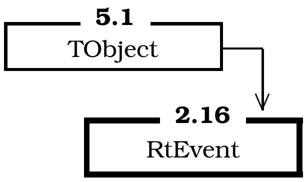
Author: C. Borgmeier, T. Glebe

2.16

class **RtEvent** : public TObject

event class

Inheritance



Public Members

		— Constructors —		
		RtEvent ()		
	virtual	~**RtEvent** ()		
		— Access methods —		
	long int	**unixtime** () const	*seconds since 1.1.70 0:00 somewhere in the U.S.*	
	int	**evt** () const	*event number*	
	int	**run** () const	*run number*	
2.16.1	int	**exp** () const	*experiment number*	47
	float	**vfld** () const	*version of the mag field MC and reco*	
	int	**flagdst** () const	*event classification flag from/to DST*	
	int	**dbmajor** () const	*major key for constants on data base*	
	int	**dbminor** () const	*minor key for constants on data base*	

	int	**flag** () const	*event classification flag from/to EDIR if no EDIR: same as flagdst*	
	unsigned int	**bx** () const	*bx number from DIHD*	
2.16.2	int	**tcode** () const	*Trigger code*	47
	float	**ir** () const	*Interaction rate*	
	float	**bg** () const	*Background rate*	
	float	**rad0** () const	*Cherenkov radius for* $\beta = 1$	
	int	**hvds** () const	*Hits in VDS*	
	int	**hwch** () const	*Hits in wire chambers*	
	int	**hmsg** () const	*Hits in MSG microstrip chambers*	
	int	**hptc** () const	*Hits in PTC chambers (pi-pi trigger)*	
	int	**hrich** () const	*Hits in Cerenkov counter*	
	int	**htrd** () const	*Hits in forward TRD*	
	int	**hcal** () const	*Hits in electromagnetic calorimeter*	
	int	**hmuo** () const	*Hits in muon outer chambers*	
	int	**hmux** () const	*Hits in muon central chambers*	

GClonesArray <Topp*> *
topp () const *get Topp table pointer*

GClonesArray <RecoTrack*> *
tracks () const *get RecoTrack table pointer*

GClonesArray <VDS_Track*> *
vds () const

GClonesArray <OTR_SpacePoint*> *
otr () const

GClonesArray <RICH_Track*> *
rich () const

GClonesArray <EcalHit*> *
ecal () const

GClonesArray <MuonTrack*> *

muon () const

GClonesArray <RecoVertex*> *
vertices () const

GClonesArray <Wire*> *
wires () const

GClonesArray <TrackClones*> *
clones () const

GClonesArray <MC_Track*> *
mcTracks () const

GClonesArray <MC_Vertex*> *
mcVertices () const

— Event algorithms —

bool **prepareRecoTracksforVertexing** ()
Prepare reconstructed tracks for vertexing with SVertex package: transform $\sigma^2(1/p) \rightarrow \sigma^2(p)$, *propagate RecoTracks to* $z = 0$, *fill inverse covariance matrix*

— Expert functions —

std::ostream&
print (std::ostream& os) const
used in operator<<()

bool **FillTopp** () *fill Topp table from ARTE TOPP*

bool **FillHeader** () *fill event header from ARTE EVHD, DIHD*

void **unlock** () *initialize objects contained in GClonesArrays*

— ROOT only functions —

void **Clear** (Option_t * = ””)

void **Delete** (Option_t * = ””)

void **Print** (Option_t *options=””)

void	**Clear** ()	*replacement of ROOT Clear()*
void	**Delete** ()	*replacement of ROOT Delete()*
vector <float>	**user**	*user space for ROOT file*
void	**StreamUser** (TBuffer &R__b)	*only availabe in ROOT*

Holds information of ARTE in C++ objects. The Track lists can also be accessed via the ARTE table interface.

Author: C. Borgmeier, T. Glebe

2.16.1

int **exp** () const

experiment number

experiment number.
exp > 0: real data
exp < 0: Monte Carlo; specifies the lab:
-1: DESY HH
-2: DESY Zeuthen
...: not yet specified

2.16.2

int **tcode** () const

Trigger code

Trigger code.
0: (FLT) FLT trigger
2: (VME) Standalone trigger
4: (LEMO1) ECAL Internal LED calibration

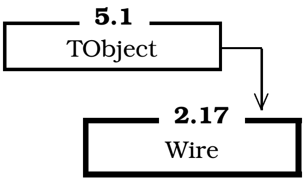
5: (LEMO2) Front Panel trigger for test
6: (LEMO3) Front Panel trigger for test
7: (LEMO4) Front Panel trigger for test
8: FLT + RANDOM
9: (RANDOM) Random trigger
10: VME + RANDOM
12: LEMO1 + RANDOM
13: LEMO2 + RANDOM
14: LEMO3 + RANDOM
15: LEMO4 + RANDOM

2.17

class **Wire** : public TObject

Target wire class

Inheritance



Public Members

— Constructors —

Wire ()

Wire (const ArtePointer<GTAR>& wire)
only available in ARTE

Wire (const ArtePointer<RTAR>& wire)
only available in ARTE

Wire (const Wire& rhs)

~**Wire** ()

— Access methods —

float **x** () const *x of wire center*

float	**y** () const	*y of wire center*
float	**z** () const	*z of wire center*
float	**dx** () const	*size in x-direction*
float	**dy** () const	*size in y-direction*
float	**dz** () const	*size in z-direction*
SVector <double, 3>	**xvec** () const	*position vector* $\vec{x} = (x,\ y,\ z)$
SVector <double, 3>	**dxvec** () const	*size vector* $\vec{dx} = (dx,\ dy,\ dz)$
SVector <double, 3>	**dvec** () const	*direction vector* $\vec{d} = (1,\ 0,\ 0)$ *or* $\vec{d} = (0,\ 1,\ 0)$

— Expert functions —

std::ostream&	**print** (std::ostream&) const	*used by operator<<()*

— ROOT only functions —

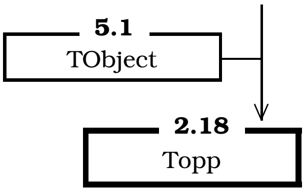
void	**Print** (Option_t *options="")
void	**Paint** (Option_t *option="")

Author: T. Glebe

2.18 class **Topp** : public TObject, public ToppIf

TOPP table entry

Inheritance



Public Members

— Constructors —

Topp ()

Topp (const ArtePointer<TOPP>& topp)
only available with ARTE

const Topp&
operator= (const ArtePointer<TOPP>& topp)

— ToppIf Access methods —

const char* const	**name** () const	*particle name*
float	**mass** () const	*mass*
float	**wid** () const	*width*
float	**trbw** () const	*truncate Breit-Wigner at mass - trbw*
float	**life** () const	*life time*
int	**chrg** () const	*3 * charge*
bool	**valid** () const	*Do we represent a valid Topp entry?*
const ToppIf&	**anti** () const	*get anti particle*
int	**lund** () const	*LUND code*

— Access Methods —

int	**colo** () const	*colour information: 0 = colour singlet; 2 = gluon; 1 = quark or anti-diquark; -1 = anti-quark or diquark*
int	**deca** () const	*Lund decay flag (0 = no decay)*
int	**gean** () const	*GEANT code*

const ArtePointer <Topp>

anti_ptr () const *anti-particle row number*

void **set_anti** (const ArtePointer<Topp>& topp) *set anti-particle row number*

— Operators —

bool **operator==** (const ToppIf& rhs) const

— Expert functions —

std::ostream& **print** (std::ostream&) const *used by operator<<()*

— ROOT only functions —

void **Print** (Option_t *option="")

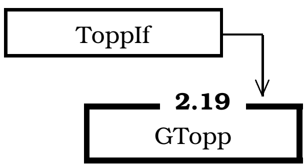
Author: C. Borgmeier, T. Glebe

2.19

class **GTopp** : public ToppIf

Representation of a Topp table entry

Inheritance



Public Members

— Constructors —

GTopp ()

GTopp (const GTopp& rhs)

GTopp (const char* name)

	GTopp (const char* name, int chrg)	
virtual	~**GTopp** ()	
	— ToppIf access methods —	
const char* const	**name** () const	*particle name*
float	**mass** () const	*mass*
float	**wid** () const	*width*
float	**trbw** () const	*truncate Breit-Wigner at mass - trbw*
float	**life** () const	*life time*
int	**chrg** () const	*3 * charge*
const ToppIf&	**anti** () const	*access to anti-particle*
int	**lund** () const	*LUND code*
	— Access methods —	
const ToppIf&	**topp** () const	*topp table pointer*
bool	**valid** () const	*Do we represent a valid Topp entry?*
	— Factory functions —	
SP_sp	**swith** (const SProperty<TrackIf, bool>& rhs) const	*create a SPattern object*
PMC_sp	**createPattern** () const	
PMC_sp	**with** (const SProperty<TrackIf, bool>& rhs) const	
template <unsigned int NTR> typename DP_trait <NTR> ::DPMC_sp	**decaysTo** (const SPatternExpression<NTR>& rhs) const	*create a SDecayPatternMatch-Container object*
	— Operators —	
bool	**operator==** (const ToppIf& rhs) const	
bool	**operator==** (const GTopp& rhs) const	
bool	**operator!=** (const GTopp& rhs) const	

bool **operator**< (const GTopp& rhs) const

— Expert functions —

void **set_topp** (const ArtePointer<Topp>& topp)
set topp table pointer

void **set_anti** (const GTopp& anti)
set link to anti GTopp object

std::ostream&
print (std::ostream& os) const
used by operator<<()

3 Gate

Names

extern const PkgIni **GateSteer** *Global Gate object which holds XCVS information.*

3.1 template <class T> class **GClonesArray** : public TClonesArray *Array of similar objects.* 54

3.2 template <class T> class **GObjArray** : public TObjArray *Array of different objects.* ... 56

3.3 template <class T> class **GObjArrayIterator** : public TObject *GObjectArray Iterator class* . 58

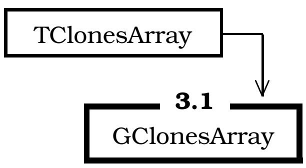
3.4 template <class T> class **ConstGObjArrayIterator** : public TObject *ConstGObjectArray Iterator class* 59

STL like ROOT containers

3.1 template <class T> class **GClonesArray** : public TClonesArray

Array of similar objects.

Inheritance



Public Members

— Constructors —

GClonesArray ()

GClonesArray (unsigned int newsize)

GClonesArray (const GClonesArray<T>& rhs)

~**GClonesArray** ()

GClonesArray <T> &
operator= (const GClonesArray<T>& rhs)

— Access methods —

iterator **begin** ()

iterator **end** ()

const_iterator
begin () const

const_iterator
end () const

inline size_type
size () const

inline bool
empty () const

T **push_back** (T obj)

inline void
pop_back ()

inline void
erase (iterator position)

void **erase** (iterator p1, iterator p2)

inline T& **GetAt** (int idx)

inline const value_type
operator[] (int idx) const

void **set** (int idx, T v)

void **print** (std::ostream& os) const
used by operator<<()

void **saveObjects** (GClonesArray<T>& newArray) const

void **copyObjectsFrom** (const GObjArray<T>& oldArray)

inline void
Print (Option_t* ="")

GClonesArray is a template wrapper for the ROOT TClonesArray class. This class is only available in ROOT. Outside ROOT, GClonesArray is typedef'd to a STL vector.

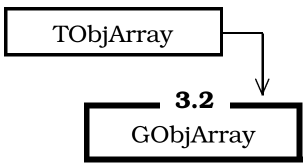
Author: C. Borgmeier

3.2

template <class T> class **GObjArray** : public TObjArray

Array of different objects.

Inheritance



Public Members

— Constructors —

GObjArray ()

GObjArray (const GObjArray<T>& rhs)

~**GObjArray** ()

GObjArray <T> &
operator= (const GObjArray<T>& rhs)

— Access methods —

iterator **begin** ()

inline iterator

end ()

const_iterator
begin () const

inline const_iterator
end () const

inline size_type
size () const

inline bool
empty () const

inline void
push_back (value_type obj)

inline void
pop_back ()

inline void
erase (iterator position)

void **erase** (iterator p1, iterator p2)

inline void
clear ()

inline const value_type
operator[] (int idx) const

inline value_type
GetAt (int idx)

inline const value_type
GetAt (int idx) const

inline void
set (int idx, value_type v)

void **print** (std::ostream& os) const
used by operator<<()

void **Print** (Option_t* ="")

void **saveObjects** (GObjArray<T>& newArray) const

GObjArray is a template wrapper for the ROOT TObjArray class. This class is only available in ROOT. Outside ROOT, GObjarray is typedef'd to a STL vector.

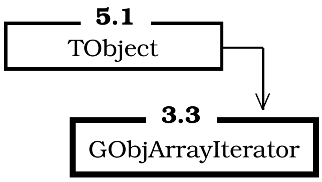
Author: C. Borgmeier

3.3

template <class T> class GObjArrayIterator : public TObject

GObjectArray Iterator class

Inheritance



Public Members

— Constructors —

GObjArrayIterator ()

GObjArrayIterator (TObjArray& array, Int_t idx)

GObjArrayIterator (const GObjArrayIterator<T>& rhs)

const GObjArrayIterator <T> & **operator=** (const GObjArrayIterator<T>& rhs)

— Operators —

T **operator*** ()

const T **operator*** () const

GObjArrayIterator <T> & **operator++** ()

GObjArrayIterator <T> **operator++** (int)

GObjArrayIterator <T> & **operator–** ()

GObjArrayIterator <T>

operator– (int)

GObjArrayIterator <T>
operator+ (int i) const

inline bool
operator!= (const ConstGObjArrayIterator<T>& rhs) const

inline bool
operator== (const GObjArrayIterator<T>& rhs) const

inline bool
operator< (const GObjArrayIterator<T>& rhs) const

— Expert functions —

void **Print** (Option_t* o="")

void **print** (std::ostream& os) const

Protected Members

void **increment** ()

inline void
decrement ()

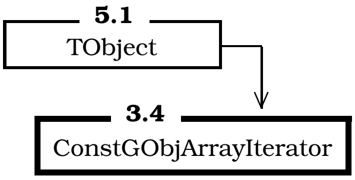
Author: C. Borgmeier, T. Glebe

3.4

template <class T> class **ConstGObjArrayIterator** : public TObject

ConstGObjectArray Iterator class

Inheritance



Public Members

— Constructors —

ConstGObjArrayIterator ()

ConstGObjArrayIterator (const ConstGObjArrayIterator<T>& rhs)

ConstGObjArrayIterator (const GObjArrayIterator<T>& rhs)

ConstGObjArrayIterator (const TObjArray& array, Int_t idx)
this would better be protected.

— Operators —

const ConstGObjArrayIterator <T> &
operator= (const ConstGObjArrayIterator<T>& rhs)

const ConstGObjArrayIterator <T> &
operator= (const GObjArrayIterator<T>& rhs)

T **operator*** ()

const T **operator*** () const

ConstGObjArrayIterator <T> &
operator++ ()

ConstGObjArrayIterator <T>
operator++ (int)

ConstGObjArrayIterator <T> &
operator– ()

ConstGObjArrayIterator <T>

operator-- (int)

ConstGObjArrayIterator <T>
operator+ (int i) const

int **operator!=** (const ConstGObjArrayIterator<T>& rhs) const

int **operator==** (const ConstGObjArrayIterator<T>& rhs) const

inline bool
operator< (const GObjArrayIterator<T>& rhs) const

void **print** (std::ostream& os) const

Protected Members

inline void
increment ()

inline void
decrement ()

Author: C. Borgmeier, T. Glebe

4 Geometry

Point/Vector classes

Names

extern const PkgIni **GeomSteer** *Global Geom object which holds XCVS information.*

4.1 namespace CLUE class **ThreeVector** : public TVector3 *ROOT based implementation* 62

4.2 namespace CLUE class **ThreeVector** : public TObject, public Hep3Vector *CLHEP implementation* 64

4.3 namespace CLUE class **Point3D** : public ThreeVector *ROOT based implementation* 65

4.4 namespace CLUE class **Point3D** : public TObject, public HepPoint3D *CLHEP implementation* 67

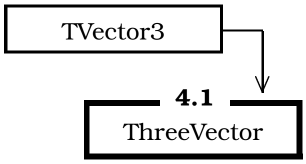
4.5 namespace CLUE class **LorentzVector** : public TLorentzVector *ROOT based implementation* 68

4.6 namespace CLUE class **LorentzVector** : public HepLorentzVector *CLHEP implementation* 70

4.1 namespace CLUE class **ThreeVector** : public TVector3

ROOT based implementation

Inheritance



Public Members

— Constructors —

ThreeVector ()

ThreeVector (const ThreeVector& rhs)

ThreeVector (const TVector3& rhs)

ThreeVector (const double x, const double y, const double z)

const ThreeVector&
operator= (const ThreeVector& rhs)

— Access methods —

inline double
x () const — *X() can be used, too*

inline double
y () const — *Y() can be used, too*

inline double
z () const — *Z() can be used, too*

inline void
set_x (const double x) — *set x-component in cartesian coordinates*

inline void
set_y (const double y) — *set y-component in cartesian coordinates*

inline void
set_z (const double z) — *set z-component in cartesian coordinates*

inline ThreeVector

unit () const — *Unit vector parallel to this.*

virtual void
Paint (Option_t *option="")
only available in ROOT

ThreeVector - ROOT based implementation for usage with ROOT. For documentation of TVector3 member functions see the ROOT reference manual.

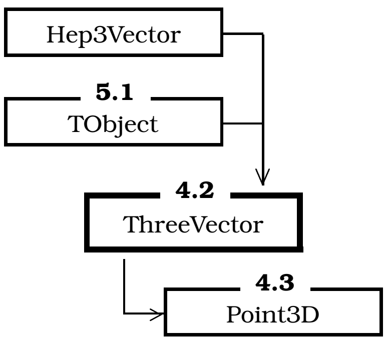
Author: T. Glebe

4.2

namespace CLUE class **ThreeVector** : public TObject, public Hep3Vector

CLHEP implementation

Inheritance



Public Members

— Constructors —

ThreeVector ()

ThreeVector (const ThreeVector& rhs)

ThreeVector (const Hep3Vector& rhs)

ThreeVector (const double x, const double y, const double z)

— Access methods —

inline double
X () const — *x() can be used, too*

inline double
Y () const — *y() can be used, too*

inline double
Z () const — *z() can be used, too*

inline void
set_x (const double x)
set x-component in cartesian coordinates

inline void
set_y (const double y)
set y-component in cartesian coordinates

inline void
set_z (const double z)
set z-component in cartesian coordinates

ThreeVector - CLHEP based implementation for usage outside ROOT. For documentation of Hep3Vector member functions see CLHEP documentation.

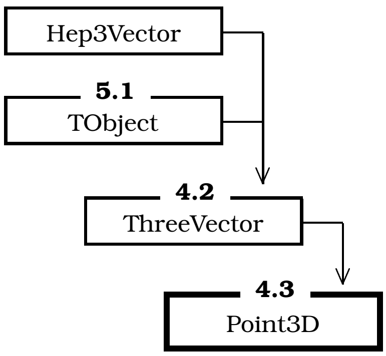
Author: C. Borgmeier

4.3

namespace CLUE class **Point3D** : public ThreeVector

ROOT based implementation

Inheritance



Public Members

— Constructors —

Point3D ()

Point3D (const Point3D& rhs)

Point3D (const ThreeVector& rhs)

Point3D (const TVector3& rhs)

Point3D (const double x, const double y, const double z)

const Point3D& **operator=** (const Point3D& rhs)

Point3D - ROOT based implementation for use with ROOT For member functions see ThreeVector documentation.

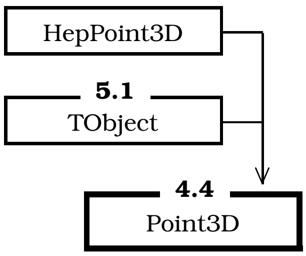
Author: T. Glebe

4.4

namespace CLUE class **Point3D** : public TObject, public HepPoint3D

CLHEP implementation

Inheritance



Public Members

— Constructors —

Point3D ()

Point3D (const Point3D& rhs)

Point3D (const HepPoint3D& rhs)

Point3D (const double x, const double y, const double z)

const Point3D&
operator= (const Point3D& rhs)

— Access methods —

inline double
X () const *x() can be used, too*

inline double
Y () const *y() can be used, too*

inline double
Z () const *z() can be used, too*

inline void
set_x (const double x)
set x-component in cartesian coordinates

inline void

set_y (const double y)
set y-component in cartesian coordinates

inline void
set_z (const double z)
set z-component in cartesian coordinates

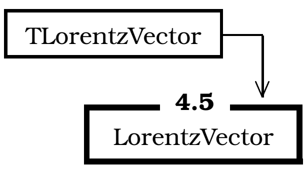
Point3D - CLHEP based implementation for use outside ROOT For HepPoint3D member functions see CLHEP documentation.

Author: C. Borgmeier

4.5 namespace CLUE class **LorentzVector** : public TLorentzVector

ROOT based implementation

Inheritance



Public Members

— Constructors —

LorentzVector ()

LorentzVector (const LorentzVector& rhs)

LorentzVector (const TLorentzVector& rhs)

LorentzVector (const ThreeVector& rhs, double E)

LorentzVector (const double px,
const double py,
const double pz,
const double E)

const LorentzVector&
operator= (const LorentzVector& rhs)

— Access methods —

inline ThreeVector
vect () const *get spatial component*

inline double
e () const *get 4th component (energy)*

inline double
mag () const *Invariant mass. If mag2() is negative then -sqrt(-mag2()) is returned.*

inline double
mag2 () const *Invariant mass squared.*

inline double
rapidity () const *Returns the rapidity, i.e.* $0.5 \cdot \ln((E + p_z)/(E - p_z))$

inline double
pseudoRapidity () const
Returns the pseudo-rapidity, i.e. $-\ln(\tan(\theta/2))$

— Operators —

inline LorentzVector
operator+ (const LorentzVector &rhs) const

inline LorentzVector
operator- (const LorentzVector &rhs) const

LorentzVector - ROOT based implementation for usage with ROOT. For documentation of TLorentzVector member functions see the ROOT reference manual.

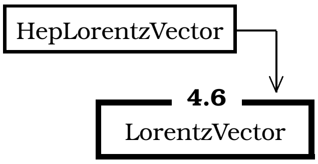
Author: T. Glebe

4.6

namespace CLUE class **LorentzVector** : public HepLorentzVector

CLHEP implementation

Inheritance



Public Members

— Constructors —

LorentzVector ()

LorentzVector (const LorentzVector& rhs)

LorentzVector (const HepLorentzVector& rhs)

LorentzVector (const Hep3Vector& rhs,
double E)

LorentzVector (const double px,
const double py,
const double pz,
const double E)

const LorentzVector&
operator= (const LorentzVector& rhs)

LorentzVector - CLHEP based implementation for usage without ROOT. For documentation of HepLorentzVector member functions see CLHEP documentation

Author: T. Glebe

5 ARTE

The C++ Arte Table interface

Names

5.1 class **TObject** *Dummy base class* 73

— typedef's from ROOT Rtypes.h —

typedef char
Char_t

typedef unsigned char
UChar_t

typedef short
Short_t

typedef unsigned short
UShort_t

typedef int
Int_t

typedef unsigned int
UInt_t

typedef int
Seek_t

typedef long
Long_t

typedef unsigned long
ULong_t

typedef float
Float_t

typedef double
Double_t

typedef char
Text_t

typedef unsigned char
Bool_t

typedef unsigned char

Byte_t

typedef short
Version_t

typedef const char
Option_t

typedef int
Ssiz_t

typedef float
Real_t

typedef void
(*VoidFuncPtr_t) ()

5.2 class **ArtePointerBase**
Base class for ArtePointers .. 75

5.3 class **ArteLink** *single link class* 77

5.4 class **ArteVector** : public ArteLink
Link vector 78

5.5 template <class T> class
ArteRelVector : public ArteVector
Arte Relation Vector 78

5.6 template <class T> class
ArtePointer : public ArtePointerBase
Arte pointer class 79

5.7 template <class Relation, class Result> class
ArteRelationPointer : public ArtePointerBase
Arte Relation Pointer 81

5.8 template <class Relation, class Result> class
ArteInvOneRelationPointer : public ArteRelation-Pointer<Relation, Result>
ArteInvOneRelationPointer .. 82

5.9 template <class Relation, class Result> class
ArteManRelationPointer : public ArteRelation-Pointer<Relation, Result>
ArteManRelationPointer 83

5.10 template <class Relation, class Result> class

ArteInvManRelationPointer : public ArteRelation-Pointer<Relation, Result>
ArteInvManRelationPointer .. 83

5.11 template <class Result> class **ArteRelationIterator** : public ArtePointerBase
ArteRelationIterator 84

5.12 template <class T> class **ArteTableIterator** : public TObject
Arte Table Iterator class 86

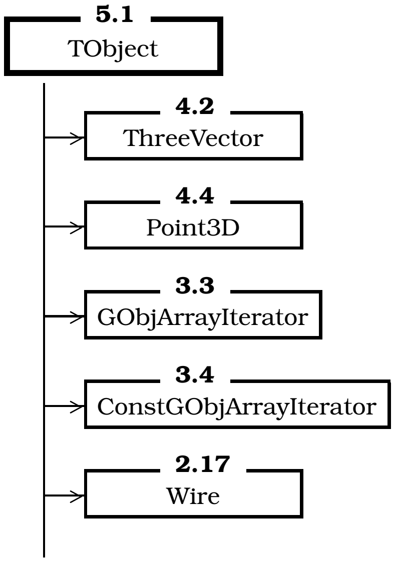
5.13 template <class T> class **ArteTable** : public TObject
Arte Table class 88

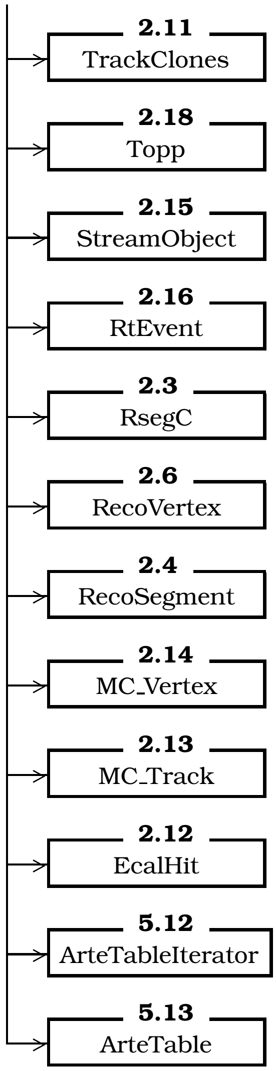
5.1

class **TObject**

Dummy base class

Inheritance





Public Members

	TObject ()
TObject*	**Clone** (void)

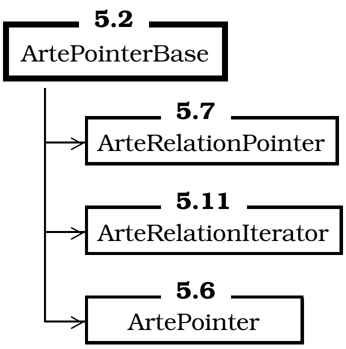
TObject is the base class for all classes which should be integrated into ROOT RTTI. For sake of clean code, a dummy base class is used in case no ROOT is present.

Author: T. Glebe

5.2 class **ArtePointerBase**

Base class for ArtePointers

Inheritance



Public Members

— Constructors —

ArtePointerBase ()

ArtePointerBase (UShort_t i)

ArtePointerBase (const ArtePointerBase& rhs)

const ArtePointerBase&
operator= (const ArtePointerBase& rhs)

— Access methods —

inline UShort_t
index () const

inline void
setInt (const UShort_t index)

inline bool

null () const

— Operators —

inline bool
operator< (const ArtePointerBase& rhs) const

inline bool
operator> (const ArtePointerBase& rhs) const

inline int **operator+** (const ArtePointerBase& rhs) const

inline int **operator-** (const ArtePointerBase& rhs) const

inline bool
operator== (const ArtePointerBase& rhs) const

inline bool
operator!= (const ArtePointerBase& rhs) const

inline ArtePointerBase&
operator++ ()

inline ArtePointerBase&
operator– ()

inline ArtePointerBase
operator++ (int)

inline ArtePointerBase
operator– (int)

— Expert functions —

inline void
print (std::ostream& os) const
used by operator<<()

Protected Members

inline UShort_t&
indexReference ()
Returns the private data.

inline void
increment ()

inline void
decrement ()

const ArtePointerBase&

operator+= (int offset)

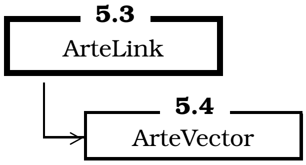
const ArtePointerBase&
operator-= (int offset)

Author: C. Borgmeier, T. Glebe

5.3 class **ArteLink**

single link class

Inheritance



Public Members

inline int&
getRef ()

inline int&
getInt ()

inline int operator
() () — *only available in ARTE compatibility mode*

inline void*&
getVec ()

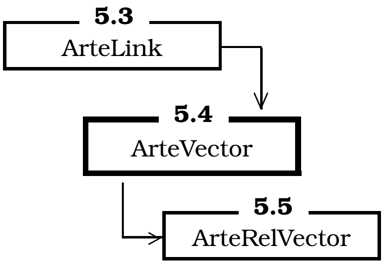
Author: C. Borgmeier

5.4

class **ArteVector** : public ArteLink

Link vector

Inheritance



Public Members

inline const int& operator
() ()

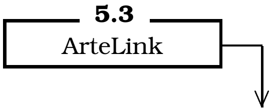
Author: C. Borgmeier

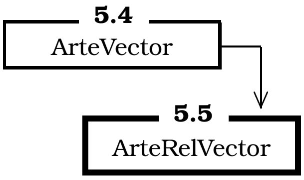
5.5

template <class T> class **ArteRelVector** : public ArteVector

Arte Relation Vector

Inheritance





Public Members

	ArteRelVector ()
	~**ArteRelVector** ()
void	**clear** ()

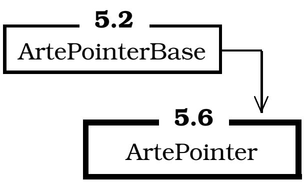
Author: C. Borgmeier

5.6

template <class T> class **ArtePointer** : public ArtePointerBase

Arte pointer class

Inheritance



Public Members

	— Constructors —
	ArtePointer ()
	ArtePointer (const UShort_t index)
explicit	**ArtePointer** (const ArtePointerBase& rhs)

this is a conversion between ArtePointers!!

ArtePointer (const ArtePointer<T>& index)

const self&
operator= (const self& rhs)

— Operators —

inline T& **operator*** () const

inline T* **operator->** () const

inline bool
operator> (const ArtePointer<T>& rhs) const

inline bool
operator< (const ArtePointer<T>& rhs) const

inline int **operator-** (const ArtePointer<T>& rhs) const

inline bool
operator!= (const ArtePointer<T>& rhs) const

inline bool
operator== (const ArtePointer<T>& rhs) const

— ARTE compatibility —

inline ArtePointer <T> &
operator++ ()

inline ArtePointer <T>
operator++ (int)

— ARTE only functions —

inline static unsigned int
nrow () *Returns the number of entries in the corresponding Arte table.*

inline static self
append (const_reference e)

inline static void*
getLink ()

inline static int&
getIntLink ()

inline static ArteLink&
getLinkLink ()

inline static int

ncol () *Return the width of the Arte table.*

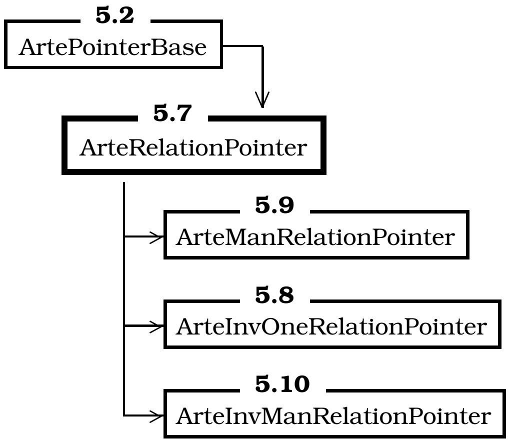
Author: C. Borgmeier, T. Glebe

5.7

template <class Relation, class Result> class **ArteRelationPointer** : public ArtePointerBase

Arte Relation Pointer

Inheritance



Public Members

— Constructors —

ArteRelationPointer ()

ArteRelationPointer (const ArteRelationPointer<Relation, Result>& arg)

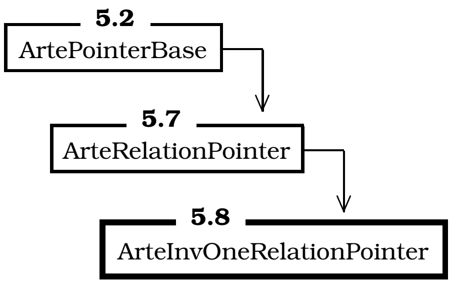
Author: C. Borgmeier

5.8

template <class Relation, class Result> class **ArteInvOneRelationPointer** : public ArteRelationPointer<Relation, Result>

ArteInvOneRelationPointer

Inheritance



Public Members

iterator **begin** () const

iterator **end** () const

inline static ArteRelVector <Relation> & **getLink** ()

inline static ArteRelVector <Relation> & **getLinkLink** () *only in ARTE compat. mode*

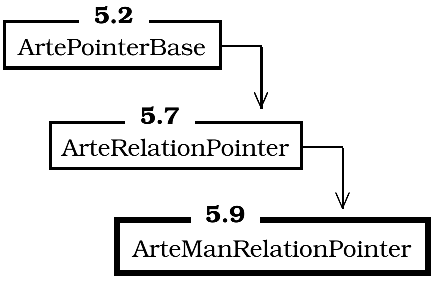
Author: C. Borgmeier

5.9

template <class Relation, class Result> class **ArteManRelationPointer** : public ArteRelationPointer<Relation, Result>

ArteManRelationPointer

Inheritance



Public Members

- iterator **begin** () const
- iterator **end** () const
- inline static ArteRelVector <Relation> & **getLink** ()
- inline static ArteRelVector <Relation> & **getLinkLink** () *only in ARTE compat. mode*

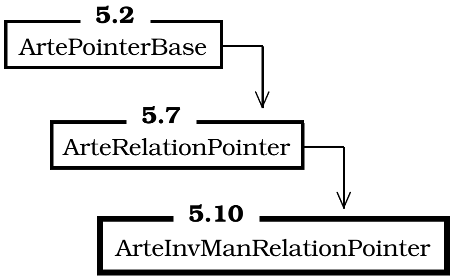
Author: C. Borgmeier

5.10

template <class Relation, class Result> class **ArteInvManRelationPointer** : public ArteRelationPointer<Relation, Result>

ArteInvManRelationPointer

Inheritance



Public Members

iterator **begin** () const

iterator **end** () const

inline static ArteRelVector <Relation> & **getLink** ()

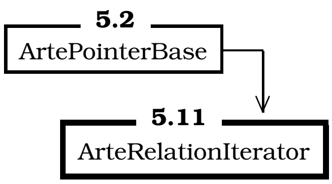
Author: C. Borgmeier

5.11

template <class Result> class **ArteRelationIterator** : public ArtePointerBase

ArteRelationIterator

Inheritance



Public Members

— Constructors —

ArteRelationIterator ()

template <class Relation>
ArteRelationIterator (const ArteInvOneRelationPointer<Relation, Result>& x)

template <class Relation>
ArteRelationIterator (const ArteManRelationPointer<Relation, Result>& x)

template <class Relation>
ArteRelationIterator (const ArteInvManRelationPointer<Relation, Result>& x)

— Relation structures —

struct **ArteOneToManyRelation**

struct **ArteManyToManyRelation**

struct **ArteBackwardsManyToManyRelation**

— Operators —

template <class Relation> const ArteRelationIterator <Result> &
operator= (const ArteInvOneRelationPointer<Relation, Result>& x)

template <class Relation> const ArteRelationIterator <Result> &
operator= (const ArteManRelationPointer<Relation, Result>& x)

template <class Relation> const ArteRelationIterator <Result> &
operator= (const ArteInvManRelationPointer<Relation, Result>& x)

inline ArtePointer <Result>
operator* () const

operator ArtePointer<**Result**> () const *only in ARTE compat. mode*

self& **operator++** ()

self **operator++** (int)

inline bool **operator!=** (const ArteRelationIterator<Result>& rhs)

inline bool **operator==** (const ArteRelationIterator<Result>& rhs)

— ARTE compatibility mode —

inline int **valid** () const

inline void **next** ()

Protected Members

ArtePointer <Result> **lookup** () const

void **followLink** ()

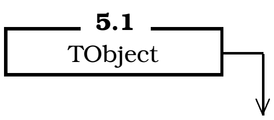
Author: C. Borgmeier, T. Glebe

5.12

template <class T> class **ArteTableIterator** : public TObject

Arte Table Iterator class

Inheritance



5.12 ArteTableIterator

Public Members

— Constructors —

ArteTableIterator ()

ArteTableIterator (const ArtePointer<T>& rhs)

ArteTableIterator (const ArteTableIterator<T>& rhs)

— Operators —

inline ArtePointer <T> &
operator* ()

inline const ArtePointer <T> &
operator* () const

self& **operator++** ()

inline self
operator++ (int)

self& **operator–** ()

inline self
operator– (int)

inline const self&
operator+= (int offset)

inline const self&
operator-= (int offset)

inline bool
operator!= (const ArteTableIterator<T>& rhs) const

inline bool
operator== (const ArteTableIterator<T>& rhs) const

inline bool
operator< (const ArteTableIterator<T>& rhs) const

inline bool

operator> (const ArteTableIterator<T>& rhs) const

inline int **operator-** (const ArteTableIterator<T>& rhs) const

inline ArteTableIterator <T>
operator+ (int rhs) const

inline ArteTableIterator <T>
operator- (int rhs) const

inline void
print (std::ostream& os) const

— Arte compatibility mode —

operator ArtePointer<**T**> () const

inline int **valid** () const

ArtePointer <T>
next ()

inline static unsigned int
nrow ()

inline T* **operator-**> () const

Protected Members

inline unsigned int
index () const *public only in ARTE compat. mode*

ArteTableIterator (unsigned int i)
forbidden constructor

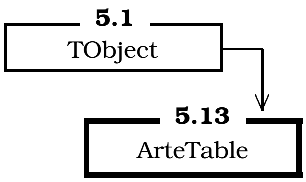
Author: C. Borgmeier, T. Glebe

5.13

template <class T> class **ArteTable** : public TObject

Arte Table class

Inheritance



Public Members

— Constructors —

ArteTable ()

— Access methods —

static std::string
getName () *not available in CINT*

static inline iterator
begin ()

static inline iterator
end ()

static unsigned int
size ()

static ArtePointer <T>
push_back (const T& e)

static ArtePointer <T>
allocate (unsigned int n)

void **print** (std::ostream& os) const

virtual void
Print (Option_t* = "")
only available in ROOT

inline static bool
isArteTable ()

inline static unsigned int
nrow () *only in ARTE compat. mode*

static void*
getLink () *private in Arte compat. mode*

— ARTE only functions —

inline static void

book (int num = 1)

inline static bool
booked ()

inline static void
free ()

inline static void
clear ()

inline static int
ncol () *Return the width of the Arte table*

Author: C. Borgmeier, T. Glebe

Class Graph

2.1 PkgIni 9

2.2 ROOTini 10

2.19 GTopp 51

3.1 GClonesArray 54

3.2 GObjArray 56

4.1 ThreeVector 62

- 4.5 LorentzVector 68
- 4.6 LorentzVector 70
- 5.1 TObject 73
  - 4.2 ThreeVector 64
    - 4.3 Point3D 65
  - 4.4 Point3D 67
  - 3.3 GObjArrayIterator 58
  - 3.4 ConstGObjArrayIterator 59
  - 2.17 Wire 48
  - 2.11 TrackClones 30
  - 2.18 Topp 49
  - 2.15 StreamObject 43
  - 2.16 RtEvent 44
  - 2.3 RsegC 10

- **2.6** RecoVertex 24
- **2.4** RecoSegment 13
  - **2.7** VDS_Track 27
  - **2.5** RecoTrack 19
  - **2.8** RICH_Track 28
  - **2.9** OTR_SpacePoint 28
  - **2.10** MuonTrack 29
- **2.14** MC_Vertex 41
- **2.13** MC_Track 35
- **2.12** EcalHit 31
- **5.12** ArteTableIterator 86
- **5.13** ArteTable 88

5.2 ArtePointerBase 75
- **5.7** ArteRelationPointer 81
  - **5.9** ArteManRelationPointer 83

5.8 ArteInvOneRelationPointer 82
5.10 ArteInvManRelationPointer 83
5.11 ArteRelationIterator 84
5.6 ArtePointer 79

5.3 ArteLink 77
5.4 ArteVector 78
5.5 ArteRelVector 78