

HepMC Reference Manual

2.03.06

Generated by Doxygen 1.5.1-3

Wed Mar 12 13:08:08 2008

Contents

1	HepMC Directory Hierarchy	1
1.1	HepMC Directories	1
2	HepMC Namespace Index	3
2.1	HepMC Namespace List	3
3	HepMC Hierarchical Index	5
3.1	HepMC Class Hierarchy	5
4	HepMC Class Index	7
4.1	HepMC Class List	7
5	HepMC File Index	9
5.1	HepMC File List	9
6	HepMC Directory Documentation	11
6.1	/home/cepa01/garren/lcg/hepmc/HepMC-2.03.06/examples/ Directory Reference	11
6.2	/home/cepa01/garren/lcg/hepmc/HepMC-2.03.06/fio/ Directory Reference	12
6.3	/home/cepa01/garren/lcg/hepmc/HepMC-2.03.06/HepMC/ Directory Reference .	13
6.4	/home/cepa01/garren/lcg/hepmc/HepMC-2.03.06/src/ Directory Reference	14
6.5	/home/cepa01/garren/lcg/hepmc/HepMC-2.03.06/test/ Directory Reference	15
7	HepMC Namespace Documentation	17
7.1	CLHEP Namespace Reference	17
7.2	detail Namespace Reference	18
7.3	HepMC Namespace Reference	19
7.4	HepMC::detail Namespace Reference	27
8	HepMC Class Documentation	29
8.1	HepMC::CommonIO Class Reference	29
8.2	HepMC::detail::disable_if<, > Struct Template Reference	35

8.3	HepMC::detail::disable_if< false, T > Struct Template Reference	36
8.4	HepMC::detail::enable_if<, > Struct Template Reference	37
8.5	HepMC::detail::enable_if< true, T > Struct Template Reference	38
8.6	HepMC::Flow Class Reference	39
8.7	HepMC::FourVector Class Reference	46
8.8	HepMC::GenEvent Class Reference	54
8.9	HepMC::GenEvent::particle_const_iterator Class Reference	72
8.10	HepMC::GenEvent::particle_iterator Class Reference	75
8.11	HepMC::GenEvent::vertex_const_iterator Class Reference	79
8.12	HepMC::GenEvent::vertex_iterator Class Reference	82
8.13	HepMC::GenParticle Class Reference	86
8.14	HepMC::GenVertex Class Reference	95
8.15	HepMC::GenVertex::edge_iterator Class Reference	109
8.16	HepMC::GenVertex::particle_iterator Class Reference	113
8.17	HepMC::GenVertex::vertex_iterator Class Reference	116
8.18	HepMC::HeavyIon Class Reference	121
8.19	HepMC::HEPEVT_Wrapper Class Reference	129
8.20	HepMC::IO_Ascii Class Reference	142
8.21	HepMC::IO_AsciiParticles Class Reference	147
8.22	HepMC::IO_BaseClass Class Reference	151
8.23	HepMC::IO_ExtendedAscii Class Reference	155
8.24	HepMC::IO_GenEvent Class Reference	162
8.25	HepMC::IO_HEPEVT Class Reference	170
8.26	HepMC::IO_HERWIG Class Reference	176
8.27	HepMC::IO_PDG_ParticleDataTable Class Reference	183
8.28	HepMC::detail::is_arithmetic< T > Struct Template Reference	186
8.29	HepMC::detail::is_arithmetic< char > Struct Template Reference	187
8.30	HepMC::detail::is_arithmetic< double > Struct Template Reference	188
8.31	HepMC::detail::is_arithmetic< float > Struct Template Reference	189
8.32	HepMC::detail::is_arithmetic< int > Struct Template Reference	190
8.33	HepMC::detail::is_arithmetic< long > Struct Template Reference	191
8.34	HepMC::detail::is_arithmetic< long double > Struct Template Reference	192
8.35	HepMC::detail::is_arithmetic< short > Struct Template Reference	193
8.36	HepMC::detail::is_arithmetic< signed char > Struct Template Reference	194
8.37	HepMC::detail::is_arithmetic< unsigned char > Struct Template Reference	195
8.38	HepMC::detail::is_arithmetic< unsigned int > Struct Template Reference	196

8.39	HepMC::detail::is_arithmetic< unsigned long > Struct Template Reference	197
8.40	HepMC::detail::is_arithmetic< unsigned short > Struct Template Reference	198
8.41	IsEventGood Class Reference	199
8.42	IsFinalState Class Reference	200
8.43	IsGoodEvent Class Reference	201
8.44	IsGoodEventMyPythia Class Reference	202
8.45	IsPhoton Class Reference	203
8.46	IsStateFinal Class Reference	204
8.47	IsW_Boson Class Reference	205
8.48	HepMC::ParticleData Class Reference	206
8.49	HepMC::ParticleDataTable Class Reference	213
8.50	HepMC::PdfInfo Class Reference	220
8.51	HepMC::Polarization Class Reference	225
8.52	HepMC::TempParticleMap Class Reference	229
8.53	HepMC::ThreeVector Class Reference	232
8.54	HepMC::WeightContainer Class Reference	237
9	HepMC File Documentation	243
9.1	CommonIO.cc File Reference	243
9.2	CommonIO.h File Reference	244
9.3	CompareGenEvent.cc File Reference	245
9.4	CompareGenEvent.h File Reference	246
9.5	enable_if.h File Reference	247
9.6	example_BuildEventFromScratch.cc File Reference	248
9.7	example_EventSelection.cc File Reference	249
9.8	example_MyHerwig.cc File Reference	250
9.9	example_MyPythia.cc File Reference	251
9.10	example_MyPythiaOnlyToHepMC.cc File Reference	252
9.11	example_MyPythiaRead.cc File Reference	253
9.12	example_MyPythiaWithEventSelection.cc File Reference	254
9.13	example_PythiaParticle.cc File Reference	255
9.14	example_ReadMyPythia.cc File Reference	256
9.15	example_UsingIterators.cc File Reference	257
9.16	Flow.cc File Reference	258
9.17	Flow.h File Reference	259
9.18	GenEvent.cc File Reference	260
9.19	GenEvent.h File Reference	261

9.20	GenParticle.cc File Reference	262
9.21	GenParticle.h File Reference	263
9.22	GenVertex.cc File Reference	264
9.23	GenVertex.h File Reference	265
9.24	HeavyIon.h File Reference	266
9.25	HEPEVT_Wrapper.cc File Reference	267
9.26	HEPEVT_Wrapper.h File Reference	268
9.27	HepMC_CLHEP20.h File Reference	270
9.28	HerwigWrapper.h File Reference	271
9.29	HerwigWrapper6_4.h File Reference	272
9.30	initPythia.cc File Reference	289
9.31	IO_Ascii.cc File Reference	290
9.32	IO_Ascii.h File Reference	291
9.33	IO_AsciiParticles.cc File Reference	292
9.34	IO_AsciiParticles.h File Reference	293
9.35	IO_BaseClass.h File Reference	294
9.36	IO_ExtendedAscii.cc File Reference	295
9.37	IO_ExtendedAscii.h File Reference	296
9.38	IO_GenEvent.cc File Reference	297
9.39	IO_GenEvent.h File Reference	298
9.40	IO_HEPEVT.cc File Reference	299
9.41	IO_HEPEVT.h File Reference	300
9.42	IO_HERWIG.cc File Reference	301
9.43	IO_HERWIG.h File Reference	302
9.44	IO_PDG_ParticleDataTable.cc File Reference	303
9.45	IO_PDG_ParticleDataTable.h File Reference	304
9.46	is_arithmetic.h File Reference	305
9.47	IsGoodEvent.h File Reference	306
9.48	list_of_examples.cc File Reference	307
9.49	ParticleData.cc File Reference	308
9.50	ParticleData.h File Reference	309
9.51	ParticleDataTable.h File Reference	310
9.52	PdfInfo.h File Reference	311
9.53	Polarization.cc File Reference	312
9.54	Polarization.h File Reference	313
9.55	PythiaHelper.h File Reference	314

9.56 PythiaWrapper.h File Reference	315
9.57 PythiaWrapper5_720.h File Reference	316
9.58 PythiaWrapper6_152.h File Reference	324
9.59 PythiaWrapper6_152_WIN32.h File Reference	331
9.60 PythiaWrapper6_2.h File Reference	332
9.61 PythiaWrapper6_2_WIN32.h File Reference	340
9.62 SearchVector.cc File Reference	341
9.63 SearchVector.h File Reference	342
9.64 SimpleVector.h File Reference	343
9.65 TempParticleMap.h File Reference	344
9.66 testHepMCIteration.h File Reference	345
9.67 testHerwigCopies.cc File Reference	346
9.68 testPrintBug.cc File Reference	347
9.69 testPythiaCopies.cc File Reference	348
9.70 testSimpleVector.cc File Reference	349
9.71 VectorConversion.h File Reference	350
9.72 Version.h File Reference	351
9.73 WeightContainer.h File Reference	352
10 HepMC Example Documentation	353
10.1 example_BuildEventFromScratch.cc	353
10.2 example_EventSelection.cc	356
10.3 example_MyHerwig.cc	358
10.4 example_MyPythia.cc	360
10.5 example_MyPythiaOnlyToHepMC.cc	362
10.6 example_MyPythiaRead.cc	363
10.7 example_MyPythiaWithEventSelection.cc	365
10.8 example_PythiaParticle.cc	367
10.9 example_UsingIterators.cc	369
10.10testHerwigCopies.cc	372
10.11testPythiaCopies.cc	374
10.12VectorConversion.h	376

Chapter 1

HepMC Directory Hierarchy

1.1 HepMC Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

examples	11
fio	12
HepMC	13
src	14
test	15

Chapter 2

HepMC Namespace Index

2.1 HepMC Namespace List

Here is a list of all namespaces with brief descriptions:

CLHEP	17
detail	18
HepMC	19
HepMC::detail	27

Chapter 3

HepMC Hierarchical Index

3.1 HepMC Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

HepMC::CommonIO	29
HepMC::detail::disable_if<, >	35
HepMC::detail::disable_if< false, T >	36
HepMC::detail::enable_if<, >	37
HepMC::detail::enable_if< true, T >	38
HepMC::Flow	39
HepMC::FourVector	46
HepMC::GenEvent	54
HepMC::GenEvent::particle_const_iterator	72
HepMC::GenEvent::particle_iterator	75
HepMC::GenEvent::vertex_const_iterator	79
HepMC::GenEvent::vertex_iterator	82
HepMC::GenParticle	86
HepMC::GenVertex	95
HepMC::GenVertex::edge_iterator	109
HepMC::GenVertex::particle_iterator	113
HepMC::GenVertex::vertex_iterator	116
HepMC::HeavyIon	121
HepMC::HEPEVT_Wrapper	129
HepMC::IO_BaseClass	151
HepMC::IO_Ascii	142
HepMC::IO_AsciiParticles	147
HepMC::IO_ExtendedAscii	155
HepMC::IO_GenEvent	162
HepMC::IO_HEPEVT	170
HepMC::IO_HERWIG	176
HepMC::IO_PDG_ParticleDataTable	183
HepMC::detail::is_arithmetic< T >	186
HepMC::detail::is_arithmetic< char >	187
HepMC::detail::is_arithmetic< double >	188
HepMC::detail::is_arithmetic< float >	189
HepMC::detail::is_arithmetic< int >	190
HepMC::detail::is_arithmetic< long >	191

HepMC::detail::is_arithmetic< long double >	192
HepMC::detail::is_arithmetic< short >	193
HepMC::detail::is_arithmetic< signed char >	194
HepMC::detail::is_arithmetic< unsigned char >	195
HepMC::detail::is_arithmetic< unsigned int >	196
HepMC::detail::is_arithmetic< unsigned long >	197
HepMC::detail::is_arithmetic< unsigned short >	198
IsEventGood	199
IsFinalState	200
IsGoodEvent	201
IsGoodEventMyPythia	202
IsPhoton	203
IsStateFinal	204
IsW_Boson	205
HepMC::ParticleData	206
HepMC::ParticleDataTable	213
HepMC::PdfInfo	220
HepMC::Polarization	225
HepMC::TempParticleMap	229
HepMC::ThreeVector	232
HepMC::WeightContainer	237

Chapter 4

HepMC Class Index

4.1 HepMC Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

HepMC::CommonIO	29
HepMC::detail::disable_if <, > (Internal - used by SimpleVector to decide if a class is arithmetic)	35
HepMC::detail::disable_if < false, T > (Internal - used by SimpleVector to decide if a class is arithmetic)	36
HepMC::detail::enable_if <, > (Internal - used to decide if a class is arithmetic) . .	37
HepMC::detail::enable_if < true, T > (Internal - use if class T is arithmetic) . . .	38
HepMC::Flow (The flow object)	39
HepMC::FourVector (FourVector (p.46) is a simple representation of a physics 4 vector)	46
HepMC::GenEvent (The GenEvent (p.54) class is the core of HepMC (p.19)) . .	54
HepMC::GenEvent::particle_const_iterator (Const particle iterator)	72
HepMC::GenEvent::particle_iterator (Non-const particle iterator)	75
HepMC::GenEvent::vertex_const_iterator (Const vertex iterator)	79
HepMC::GenEvent::vertex_iterator (Non-const vertex iterator)	82
HepMC::GenParticle (The GenParticle (p. 86) class contains information about generated particles)	86
HepMC::GenVertex (GenVertex (p.95) contains information about decay vertices)	95
HepMC::GenVertex::edge_iterator (Edge iterator)	109
HepMC::GenVertex::particle_iterator (Particle iterator)	113
HepMC::GenVertex::vertex_iterator (Vertex iterator)	116
HepMC::HeavyIon (The HeavyIon (p. 121) class stores information about heavy ions)	121
HepMC::HEPEVT_Wrapper (Generic Wrapper for the fortran HEPEVT common block)	129
HepMC::IO_Ascii (IO_Ascii (p. 142) is used to read or write from an ascii file) . .	142
HepMC::IO_AsciiParticles (Event input/output in ascii format for eye and machine reading)	147
HepMC::IO_BaseClass (All input/output classes inherit from IO_BaseClass (p. 151))	151
HepMC::IO_ExtendedAscii (IO_ExtendedAscii (p.155) also deals with HeavyIon (p.121) and PdfInfo (p.220))	155
HepMC::IO_GenEvent (IO_GenEvent (p.162) also deals with HeavyIon (p.121) and PdfInfo (p.220))	162

HepMC::IO_HEPEVT (HEPEVT IO class)	170
HepMC::IO_HERWIG (IO_HERWIG (p. 176) is used to get Herwig information)	176
HepMC::IO_PDG_ParticleDataTable (Example ParticleDataTable (p. 213) IO method)	183
HepMC::detail::is_arithmetic< T > (Undefined and therefore non-arithmetic)	186
HepMC::detail::is_arithmetic< char > (Character is arithmetic)	187
HepMC::detail::is_arithmetic< double > (Double is arithmetic)	188
HepMC::detail::is_arithmetic< float > (Float is arithmetic)	189
HepMC::detail::is_arithmetic< int > (Int is arithmetic)	190
HepMC::detail::is_arithmetic< long > (Long is arithmetic)	191
HepMC::detail::is_arithmetic< long double > (Long double is arithmetic)	192
HepMC::detail::is_arithmetic< short > (Short is arithmetic)	193
HepMC::detail::is_arithmetic< signed char > (Signed character is arithmetic)	194
HepMC::detail::is_arithmetic< unsigned char > (Unsigned character is arithmetic)	195
HepMC::detail::is_arithmetic< unsigned int > (Unsigned int is arithmetic)	196
HepMC::detail::is_arithmetic< unsigned long > (Unsigned long is arithmetic)	197
HepMC::detail::is_arithmetic< unsigned short > (Unsigned short is arithmetic)	198
IsEventGood (Example class)	199
IsFinalState (Test class)	200
IsGoodEvent	201
IsGoodEventMyPythia (Example class)	202
IsPhoton (Example class)	203
IsStateFinal (Example class)	204
IsW_Boson (Example class)	205
HepMC::ParticleData (Example ParticleData (p. 206) class)	206
HepMC::ParticleDataTable (Example ParticleDataTable (p. 213) class)	213
HepMC::PdfInfo (The PdfInfo (p. 220) class stores PDF information)	220
HepMC::Polarization (The Polarization (p. 225) class stores theta and phi for a GenParticle (p. 86))	225
HepMC::TempParticleMap (TempParticleMap (p. 229) is a temporary GenParticle* container used during input)	229
HepMC::ThreeVector (ThreeVector (p. 232) is a simple representation of a position or displacement 3 vector)	232
HepMC::WeightContainer (Container for the Weights associated with an event or vertex)	237

Chapter 5

HepMC File Index

5.1 HepMC File List

Here is a list of all files with brief descriptions:

CommonIO.cc	243
CommonIO.h	244
CompareGenEvent.cc	245
CompareGenEvent.h	246
enable_if.h	247
example_BuildEventFromScratch.cc	248
example_EventSelection.cc	249
example_MyHerwig.cc	250
example_MyPythia.cc	251
example_MyPythiaOnlyToHepMC.cc	252
example_MyPythiaRead.cc	253
example_MyPythiaWithEventSelection.cc	254
example_PythiaParticle.cc	255
example_ReadMyPythia.cc	256
example_UsingIterators.cc	257
Flow.cc	258
Flow.h	259
GenEvent.cc	260
GenEvent.h	261
GenParticle.cc	262
GenParticle.h	263
GenVertex.cc	264
GenVertex.h	265
HeavyIon.h	266
HEPEVT_Wrapper.cc	267
HEPEVT_Wrapper.h	268
HepMC_CLHEP20.h	270
HerwigWrapper.h	271
HerwigWrapper6_4.h	272
initPythia.cc	289
IO_Ascii.cc	290
IO_Ascii.h	291
IO_AsciiParticles.cc	292

IO_AsciiParticles.h	293
IO_BaseClass.h	294
IO_ExtendedAscii.cc	295
IO_ExtendedAscii.h	296
IO_GenEvent.cc	297
IO_GenEvent.h	298
IO_HEPEVT.cc	299
IO_HEPEVT.h	300
IO_HERWIG.cc	301
IO_HERWIG.h	302
IO_PDG_ParticleDataTable.cc	303
IO_PDG_ParticleDataTable.h	304
is_arithmetic.h	305
IsGoodEvent.h	306
list_of_examples.cc	307
ParticleData.cc	308
ParticleData.h	309
ParticleDataTable.h	310
PdfInfo.h	311
Polarization.cc	312
Polarization.h	313
PythiaHelper.h	314
PythiaWrapper.h	315
PythiaWrapper5_720.h	316
PythiaWrapper6_152.h	324
PythiaWrapper6_152_WIN32.h	331
PythiaWrapper6_2.h	332
PythiaWrapper6_2_WIN32.h	340
SearchVector.cc	341
SearchVector.h	342
SimpleVector.h	343
TempParticleMap.h	344
testHepMCIteration.h	345
testHerwigCopies.cc	346
testPrintBug.cc	347
testPythiaCopies.cc	348
testSimpleVector.cc	349
VectorConversion.h	350
Version.h	351
WeightContainer.h	352

Chapter 6

HepMC Directory Documentation

6.1 /home/cepa01/garren/lcg/hepmc/HepMC-2.03.06/examples/ Directory Reference

Files

- file `example_BuildEventFromScratch.cc`
- file `example_EventSelection.cc`
- file `example_MyHerwig.cc`
- file `example_MyPythia.cc`
- file `example_MyPythiaOnlyToHepMC.cc`
- file `example_MyPythiaRead.cc`
- file `example_MyPythiaWithEventSelection.cc`
- file `example_PythiaParticle.cc`
- file `example_ReadMyPythia.cc`
- file `example_UsingIterators.cc`
- file `initPythia.cc`
- file `list_of_examples.cc`
- file `PythiaHelper.h`
- file `testHerwigCopies.cc`
- file `testPythiaCopies.cc`
- file `VectorConversion.h`

6.2 /home/cepa01/garren/lcg/hepmc/HepMC-2.03.06/fio/ Directory Reference

Files

- file **HEPEVT_Wrapper.cc**
- file **IO_HEPEVT.cc**
- file **IO_HERWIG.cc**

6.3 /home/cepa01/garren/lcg/hepmc/HepMC- 2.03.06/HepMC/ Directory Reference

Files

- file **CommonIO.h**
- file **CompareGenEvent.h**
- file **enable_if.h**
- file **Flow.h**
- file **GenEvent.h**
- file **GenParticle.h**
- file **GenVertex.h**
- file **HeavyIon.h**
- file **HEPEVT_Wrapper.h**
- file **HepMC_CLHEP20.h**
- file **HerwigWrapper.h**
- file **HerwigWrapper6_4.h**
- file **IO_Ascii.h**
- file **IO_AsciiParticles.h**
- file **IO_BaseClass.h**
- file **IO_ExtendedAscii.h**
- file **IO_GenEvent.h**
- file **IO_HEPEVT.h**
- file **IO_HERWIG.h**
- file **IO_PDG_ParticleDataTable.h**
- file **is_arithmetic.h**
- file **ParticleData.h**
- file **ParticleDataTable.h**
- file **PdfInfo.h**
- file **Polarization.h**
- file **PythiaWrapper.h**
- file **PythiaWrapper5_720.h**
- file **PythiaWrapper6_152.h**
- file **PythiaWrapper6_152_WIN32.h**
- file **PythiaWrapper6_2.h**
- file **PythiaWrapper6_2_WIN32.h**
- file **SearchVector.h**
- file **SimpleVector.h**
- file **TempParticleMap.h**
- file **Version.h**
- file **WeightContainer.h**

6.4 /home/cepa01/garren/lcg/hepmc/HepMC-2.03.06/src/ Directory Reference

Files

- file **CommonIO.cc**
- file **CompareGenEvent.cc**
- file **Flow.cc**
- file **GenEvent.cc**
- file **GenParticle.cc**
- file **GenVertex.cc**
- file **IO_Ascii.cc**
- file **IO_AsciiParticles.cc**
- file **IO_ExtendedAscii.cc**
- file **IO_GenEvent.cc**
- file **IO_PDG_ParticleDataTable.cc**
- file **ParticleData.cc**
- file **Polarization.cc**
- file **SearchVector.cc**

6.5 /home/cepa01/garren/lcg/hepmc/HepMC-2.03.06/test/ Directory Reference

Files

- file **IsGoodEvent.h**
- file **testHepMCIteration.h**
- file **testPrintBug.cc**
- file **testSimpleVector.cc**

Chapter 7

HepMC Namespace Documentation

7.1 CLHEP Namespace Reference

7.1.1 Detailed Description

CLHEP (p. 17) Vector classes are used in one of the examples

7.2 detail Namespace Reference

7.2.1 Detailed Description

internal namespace

7.3 HepMC Namespace Reference

Classes

- class **CommonIO**
- class **Flow**

The flow object.

- class **GenEvent**

*The **GenEvent** (p. 54) class is the core of **HepMC** (p. 19).*

- class **GenParticle**

*The **GenParticle** (p. 86) class contains information about generated particles.*

- class **GenVertex**

***GenVertex** (p. 95) contains information about decay vertices.*

- class **HeavyIon**

*The **HeavyIon** (p. 121) class stores information about heavy ions.*

- class **HEPEVT_Wrapper**

*Generic Wrapper for the fortran **HEPEVT** common block.*

- class **IO_Ascii**

***IO_Ascii** (p. 142) is used to read or write from an ascii file.*

- class **IO_AsciiParticles**

event input/output in ascii format for eye and machine reading

- class **IO_BaseClass**

*all input/output classes inherit from **IO_BaseClass** (p. 151)*

- class **IO_ExtendedAscii**

***IO_ExtendedAscii** (p. 155) also deals with **HeavyIon** (p. 121) and **PdfInfo** (p. 220).*

- class **IO_GenEvent**

***IO_GenEvent** (p. 162) also deals with **HeavyIon** (p. 121) and **PdfInfo** (p. 220).*

- class **IO_HEPEVT**

***HEPEVT** IO class.*

- class **IO_HERWIG**

***IO_HERWIG** (p. 176) is used to get Herwig information.*

- class **IO_PDG_ParticleDataTable**

*an example **ParticleDataTable** (p. 213) IO method*

- class **ParticleData**

*an example **ParticleData** (p. 206) class*

- class **ParticleDataTable**
*an example **ParticleDataTable** (p. 213) class*
- class **PdfInfo**
*The **PdfInfo** (p. 220) class stores PDF information.*
- class **Polarization**
*The **Polarization** (p. 225) class stores theta and phi for a **GenParticle** (p. 86).*
- class **FourVector**
***FourVector** (p. 46) is a simple representation of a physics 4 vector.*
- class **ThreeVector**
***ThreeVector** (p. 232) is a simple representation of a position or displacement 3 vector.*
- class **TempParticleMap**
TempParticleMap** (p. 229) is a temporary **GenParticle container used during input.*
- class **WeightContainer**
Container for the Weights associated with an event or vertex.

Namespaces

- namespace **detail**

Enumerations

- enum **known_io** {
 gen = 1, **ascii**, **extascii**, **ascii_pdt**,
 extascii_pdt }
*The **known_io** enum is used to track which type of input is being read.*
- enum **IteratorRange** {
 parents, **children**, **family**, **ancestors**,
 descendants, **relatives** }
type of iteration

Functions

- bool **compareGenEvent** (**GenEvent** *, **GenEvent** *)
- bool **compareSignalProcessVertex** (**GenEvent** *, **GenEvent** *)
- bool **compareBeamParticles** (**GenEvent** *, **GenEvent** *)
- bool **compareWeights** (**GenEvent** *, **GenEvent** *)
- bool **compareVertices** (**GenEvent** *, **GenEvent** *)
- bool **compareParticles** (**GenEvent** *, **GenEvent** *)
- bool **compareVertex** (**GenVertex** *v1, **GenVertex** *v2)

- template<class InputIterator, class OutputIterator, class Predicate> void **copy_if** (InputIterator first, InputIterator last, OutputIterator out, Predicate pred)
define the type of iterator to use
- double **lifetime_from_width** (double width)
set lifetime from width
- bool **not_in_vector** (std::vector< **HepMC::GenParticle** * > *, **GenParticle** *)
returns true if it cannot find GenParticle in the vector*
- std::vector< **HepMC::GenParticle** * >::iterator **already_in_vector** (std::vector< **HepMC::GenParticle** * > *, **GenParticle** *)
- void **version** ()
*print **HepMC** (p. 19) version*
- void **writeVersion** (std::ostream &os)
*write **HepMC** (p. 19) version to os*
- std::string **versionName** ()
*return **HepMC** (p. 19) version*
- std::ostream & **operator<<** (std::ostream &ostr, const **Flow** &f)
for printing
- std::ostream & **operator<<** (std::ostream &ostr, const **GenParticle** &part)
print particle
- std::ostream & **operator<<** (std::ostream &ostr, const **GenVertex** &vtx)
print vertex information
- std::ostream & **operator<<** (std::ostream &ostr, const **ParticleData** &pdata)
write to ostr
- std::ostream & **operator<<** (std::ostream &ostr, const **Polarization** &polar)
print polarization information
- bool **not_in_vector** (std::vector< **GenParticle** * > *v, **GenParticle** *p)
- std::vector< **HepMC::GenParticle** * >::iterator **already_in_vector** (std::vector< **GenParticle** * > *v, **GenParticle** *p)
*returns true if **GenParticle** (p. 86) is in the vector*

Variables

- static const double **HepMC_hbarc**
*hbar * c -> calculated with units of [mm*GeV]*
- static const double **HepMC_pi** = 3.14159265358979323846

7.3.1 Detailed Description

All classes in the **HepMC** (p. 19) packages are in the **HepMC** (p. 19) namespace

7.3.2 Enumeration Type Documentation

7.3.2.1 `enum HepMC::IteratorRange`

type of iteration

Enumerator:

parents
children
family
ancestors
descendants
relatives

Definition at line 35 of file GenVertex.h.

7.3.2.2 `enum HepMC::known_io`

The `known_io` enum is used to track which type of input is being read.

Enumerator:

gen
ascii
extascii
ascii_pdt
extascii_pdt

Definition at line 22 of file CommonIO.h.

7.3.3 Function Documentation

7.3.3.1 `std::vector<HepMC::GenParticle*>::iterator HepMC::already_in_vector (std::vector< GenParticle * > * v, GenParticle * p)`

returns true if **GenParticle** (p. 86) is in the vector

Definition at line 18 of file SearchVector.cc.

References p.

7.3.3.2 `std::vector<HepMC::GenParticle*>::iterator HepMC::already_in_vector (std::vector< HepMC::GenParticle * > *, GenParticle *)`

Returns the index of a `GenParticle*` within a vector. Returns -1 if `GenParticle*` is not in the vector.

Referenced by `not_in_vector()`, `HepMC::GenVertex::remove_particle_in()`, and `HepMC::GenVertex::remove_particle_out()`.

7.3.3.3 `double HepMC::clifetime_from_width (double width)`

set lifetime from width

if you want to instantiate the particle lifetime from its width, use this static method inside the constructor:

Examples:

`example_BuildEventFromScratch.cc`.

Definition at line 110 of file `ParticleData.cc`.

References `HepMC_hbarc`.

Referenced by `main()`, and `HepMC::IO_PDG_ParticleDataTable::read_entry()`.

7.3.3.4 `bool HepMC::compareBeamParticles (GenEvent *, GenEvent *)`

Definition at line 77 of file `CompareGenEvent.cc`.

References `HepMC::GenEvent::beam_particles()`.

Referenced by `compareGenEvent()`.

7.3.3.5 `bool HepMC::compareGenEvent (GenEvent *, GenEvent *)`

Examples:

`testHerwigCopies.cc`, and `testPythiaCopies.cc`.

Definition at line 16 of file `CompareGenEvent.cc`.

References `HepMC::GenEvent::alphaQCD()`, `HepMC::GenEvent::alphaQED()`, `compareBeamParticles()`, `compareParticles()`, `compareSignalProcessVertex()`, `compareVertices()`, `compareWeights()`, `HepMC::GenEvent::event_number()`, `HepMC::GenEvent::event_scale()`, `HepMC::GenEvent::heavy_ion()`, `HepMC::GenEvent::mpi()`, `HepMC::GenEvent::pdf_info()`, `HepMC::GenEvent::random_states()`, and `HepMC::GenEvent::signal_process_id()`.

Referenced by `main()`.

7.3.3.6 `bool HepMC::compareParticles (GenEvent *, GenEvent *)`

Definition at line 109 of file `CompareGenEvent.cc`.

References `HepMC::GenEvent::particles_begin()`, `HepMC::GenEvent::particles_end()`, and `HepMC::GenEvent::particles_size()`.

Referenced by `compareGenEvent()`.

7.3.3.7 bool HepMC::compareSignalProcessVertex (GenEvent *, GenEvent *)

Definition at line 64 of file CompareGenEvent.cc.

References HepMC::GenEvent::signal_process_vertex().

Referenced by compareGenEvent().

7.3.3.8 bool HepMC::compareVertex (GenVertex * v1, GenVertex * v2)

Definition at line 152 of file CompareGenEvent.cc.

References HepMC::GenVertex::barcode(), HepMC::GenVertex::particles_in_const_begin(), HepMC::GenVertex::particles_in_const_end(), HepMC::GenVertex::particles_in_size(), HepMC::GenVertex::particles_out_const_begin(), HepMC::GenVertex::particles_out_const_end(), HepMC::GenVertex::particles_out_size(), and HepMC::GenVertex::position().

Referenced by compareVertices().

7.3.3.9 bool HepMC::compareVertices (GenEvent *, GenEvent *)

Definition at line 131 of file CompareGenEvent.cc.

References HepMC::GenEvent::barcode_to_vertex(), compareVertex(), v, HepMC::GenEvent::vertices_begin(), HepMC::GenEvent::vertices_end(), and HepMC::GenEvent::vertices_size().

Referenced by compareGenEvent().

7.3.3.10 bool HepMC::compareWeights (GenEvent *, GenEvent *)

Definition at line 92 of file CompareGenEvent.cc.

References HepMC::WeightContainer::size(), and HepMC::GenEvent::weights().

Referenced by compareGenEvent().

7.3.3.11 template<class InputIterator, class OutputIterator, class Predicate> void HepMC::copy_if (InputIterator *first*, InputIterator *last*, OutputIterator *out*, Predicate *pred*) [inline]

define the type of iterator to use

Examples:

example_UsingIterators.cc.

Definition at line 50 of file GenEvent.h.

Referenced by main().

7.3.3.12 bool HepMC::not_in_vector (std::vector< GenParticle * > * v, GenParticle * p)

Definition at line 11 of file SearchVector.cc.

References already_in_vector(), and p.

7.3.3.13 `bool HepMC::not_in_vector (std::vector< HepMC::GenParticle * > *, GenParticle *)`

returns true if it cannot find GenParticle* in the vector

Referenced by HepMC::Flow::connected_partners(), and HepMC::Flow::dangling_connected_partners().

7.3.3.14 `std::ostream& HepMC::operator<< (std::ostream & ostr, const Polarization & polar)`

print polarization information

Definition at line 107 of file Polarization.cc.

References HepMC::Polarization::phi(), and HepMC::Polarization::theta().

7.3.3.15 `std::ostream& HepMC::operator<< (std::ostream & ostr, const ParticleData & pdata)`

write to ostr

Definition at line 94 of file ParticleData.cc.

References HepMC::ParticleData::charge(), HepMC::ParticleData::clifetime(), HepMC::ParticleData::mass(), HepMC::ParticleData::name(), HepMC::ParticleData::pdg_id(), and HepMC::ParticleData::spin().

7.3.3.16 `std::ostream& HepMC::operator<< (std::ostream & ostr, const GenVertex & vtx)`

print vertex information

Definition at line 440 of file GenVertex.cc.

References HepMC::GenVertex::barcode(), HepMC::GenVertex::position(), and HepMC::FourVector::x().

7.3.3.17 `std::ostream& HepMC::operator<< (std::ostream & ostr, const GenParticle & part)`

print particle

Definition at line 189 of file GenParticle.cc.

References HepMC::GenVertex::barcode(), HepMC::GenParticle::barcode(), HepMC::FourVector::e(), HepMC::GenParticle::end_vertex(), HepMC::GenParticle::momentum(), HepMC::GenParticle::pdg_id(), HepMC::FourVector::px(), HepMC::FourVector::py(), HepMC::FourVector::pz(), and HepMC::GenParticle::status().

7.3.3.18 `std::ostream& HepMC::operator<< (std::ostream & ostr, const Flow & f)`

for printing

Definition at line 190 of file Flow.cc.

References HepMC::Flow::m_icode.

7.3.3.19 void HepMC::version () [inline]

print **HepMC** (p. 19) version

Definition at line 26 of file Version.h.

References versionName().

7.3.3.20 std::string HepMC::versionName () [inline]

return **HepMC** (p. 19) version

Definition at line 21 of file Version.h.

Referenced by version(), HepMC::IO_GenEvent::write_event(), HepMC::IO_ExtendedAscii::write_event(), HepMC::IO_AsciiParticles::write_event(), HepMC::IO_Ascii::write_event(), and writeVersion().

7.3.3.21 void HepMC::writeVersion (std::ostream & os) [inline]

write **HepMC** (p. 19) version to os

Definition at line 32 of file Version.h.

References versionName().

Referenced by HepMC::GenEvent::print_version().

7.3.4 Variable Documentation

7.3.4.1 const double HepMC::HepMC_hbarc [static]

Initial value:

$$(6.6260755e-34 * (1.e-6/1.60217733e-19) / (2*3.14159265358979323846)) \\ * (2.99792458e+8 * 1000.) * 1.e+3$$

hbar * c -> calculated with units of [mm*GeV]

Definition at line 56 of file ParticleData.h.

Referenced by clifetime_from_width(), HepMC::ParticleData::set_width(), and HepMC::ParticleData::width().

7.3.4.2 const double HepMC::HepMC_pi = 3.14159265358979323846 [static]

Definition at line 19 of file Polarization.h.

7.4 HepMC::detail Namespace Reference

Classes

- struct **enable_if**
internal - used to decide if a class is arithmetic
- struct **enable_if**< **true**, **T** >
internal - use if class T is arithmetic
- struct **disable_if**
internal - used by SimpleVector to decide if a class is arithmetic
- struct **disable_if**< **false**, **T** >
internal - used by SimpleVector to decide if a class is arithmetic
- struct **is_arithmetic**
undefined and therefore non-arithmetic
- struct **is_arithmetic**< **char** >
character is arithmetic
- struct **is_arithmetic**< **unsigned char** >
unsigned character is arithmetic
- struct **is_arithmetic**< **signed char** >
signed character is arithmetic
- struct **is_arithmetic**< **short** >
short is arithmetic
- struct **is_arithmetic**< **unsigned short** >
unsigned short is arithmetic
- struct **is_arithmetic**< **int** >
int is arithmetic
- struct **is_arithmetic**< **unsigned int** >
unsigned int is arithmetic
- struct **is_arithmetic**< **long** >
long is arithmetic
- struct **is_arithmetic**< **unsigned long** >
unsigned long is arithmetic
- struct **is_arithmetic**< **float** >
float is arithmetic
- struct **is_arithmetic**< **double** >

double is arithmetic

- struct **is_arithmetic**< long double >
long double is arithmetic

Chapter 8

HepMC Class Documentation

8.1 HepMC::CommonIO Class Reference

```
#include <CommonIO.h>
```

Public Member Functions

- **CommonIO** ()
- **~CommonIO** ()
- **std::string IO_GenEvent_Key** () const
- **std::string IO_GenEvent_End** () const
- **std::string IO_Ascii_Key** () const
- **std::string IO_Ascii_End** () const
- **std::string IO_ExtendedAscii_Key** () const
- **std::string IO_ExtendedAscii_End** () const
- **int io_type** () const
get IO type
- **void write_IO_GenEvent_Key** (std::ostream &)
- **void write_IO_GenEvent_End** (std::ostream &)
- **void write_IO_Ascii_Key** (std::ostream &)
- **void write_IO_Ascii_End** (std::ostream &)
- **void write_IO_ExtendedAscii_Key** (std::ostream &)
- **void write_IO_ExtendedAscii_End** (std::ostream &)
- **void write_IO_Ascii_PDT_Key** (std::ostream &)
- **void write_IO_Ascii_PDT_End** (std::ostream &)
- **void write_IO_ExtendedAscii_PDT_Key** (std::ostream &)
- **void write_IO_ExtendedAscii_PDT_End** (std::ostream &)
- **int find_file_type** (std::istream &)
look for line type (key)
- **int find_end_key** (std::istream &)
look for line type (key)
- **bool read_io_ascii** (std::istream *is, **GenEvent** *evt)

- `bool read_io_extendedascii (std::istream *is, GenEvent *evt)`
- `bool read_io_genevent (std::istream *is, GenEvent *evt)`
- `bool read_io_particle_data_table (std::istream *, ParticleDataTable *)`

Protected Member Functions

- `HeavyIon * read_heavy_ion (std::istream *)`
- `PdfInfo * read_pdf_info (std::istream *)`
- `GenParticle * read_particle (std::istream *, TempParticleMap &)`
- `GenVertex * read_vertex (std::istream *, TempParticleMap &)`
- `ParticleData * read_particle_data (std::istream *, ParticleDataTable *)`

8.1.1 Detailed Description

Definition at line 24 of file CommonIO.h.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 HepMC::CommonIO::CommonIO () [inline]

Definition at line 104 of file CommonIO.h.

8.1.2.2 HepMC::CommonIO::~CommonIO () [inline]

Definition at line 29 of file CommonIO.h.

8.1.3 Member Function Documentation

8.1.3.1 std::string HepMC::CommonIO::IO_GenEvent_Key () const [inline]

Definition at line 32 of file CommonIO.h.

8.1.3.2 std::string HepMC::CommonIO::IO_GenEvent_End () const [inline]

Definition at line 33 of file CommonIO.h.

8.1.3.3 std::string HepMC::CommonIO::IO_Ascii_Key () const [inline]

Definition at line 35 of file CommonIO.h.

8.1.3.4 std::string HepMC::CommonIO::IO_Ascii_End () const [inline]

Definition at line 36 of file CommonIO.h.

8.1.3.5 std::string HepMC::CommonIO::IO_ExtendedAscii_Key () const [inline]

Definition at line 38 of file CommonIO.h.

8.1.3.6 `std::string HepMC::CommonIO::IO_ExtendedAscii_End () const`
[inline]

Definition at line 39 of file CommonIO.h.

8.1.3.7 `int HepMC::CommonIO::io_type () const` [inline]

get IO type

Definition at line 41 of file CommonIO.h.

Referenced by HepMC::IO_GenEvent::fill_next_event().

8.1.3.8 `void HepMC::CommonIO::write_IO_GenEvent_Key (std::ostream &)`
[inline]

Definition at line 118 of file CommonIO.h.

Referenced by HepMC::IO_GenEvent::write_event().

8.1.3.9 `void HepMC::CommonIO::write_IO_GenEvent_End (std::ostream &)`
[inline]

Definition at line 121 of file CommonIO.h.

Referenced by HepMC::IO_GenEvent::write_end_listing().

8.1.3.10 `void HepMC::CommonIO::write_IO_Ascii_Key (std::ostream &)`
[inline]

Definition at line 124 of file CommonIO.h.

8.1.3.11 `void HepMC::CommonIO::write_IO_Ascii_End (std::ostream &)`
[inline]

Definition at line 127 of file CommonIO.h.

Referenced by HepMC::IO_Ascii::write_end_listing().

8.1.3.12 `void HepMC::CommonIO::write_IO_ExtendedAscii_Key (std::ostream &)` [inline]

Definition at line 130 of file CommonIO.h.

Referenced by HepMC::IO_ExtendedAscii::write_event().

8.1.3.13 `void HepMC::CommonIO::write_IO_ExtendedAscii_End (std::ostream &)` [inline]

Definition at line 133 of file CommonIO.h.

Referenced by HepMC::IO_ExtendedAscii::write_end_listing().

8.1.3.14 `void HepMC::CommonIO::write_IO_Ascii_PDT_Key (std::ostream &)`
`[inline]`

Definition at line 136 of file CommonIO.h.

8.1.3.15 `void HepMC::CommonIO::write_IO_Ascii_PDT_End (std::ostream &)`
`[inline]`

Definition at line 139 of file CommonIO.h.

8.1.3.16 `void HepMC::CommonIO::write_IO_ExtendedAscii_PDT_Key`
`(std::ostream &) [inline]`

Definition at line 142 of file CommonIO.h.

8.1.3.17 `void HepMC::CommonIO::write_IO_ExtendedAscii_PDT_End`
`(std::ostream &) [inline]`

Definition at line 145 of file CommonIO.h.

8.1.3.18 `int HepMC::CommonIO::find_file_type (std::istream &)`

look for line type (key)

Definition at line 17 of file CommonIO.cc.

References HepMC::ascii, HepMC::ascii_pdt, HepMC::extascii, HepMC::extascii_pdt, and HepMC::gen.

Referenced by HepMC::IO_GenEvent::fill_next_event(), HepMC::IO_ExtendedAscii::fill_next_event(), HepMC::IO_Ascii::fill_next_event(), HepMC::IO_ExtendedAscii::fill_particle_data_table(), and HepMC::IO_Ascii::fill_particle_data_table().

8.1.3.19 `int HepMC::CommonIO::find_end_key (std::istream &)`

look for line type (key)

Definition at line 44 of file CommonIO.cc.

References HepMC::ascii, HepMC::ascii_pdt, HepMC::extascii, HepMC::extascii_pdt, and HepMC::gen.

Referenced by HepMC::IO_GenEvent::fill_next_event(), HepMC::IO_ExtendedAscii::fill_next_event(), HepMC::IO_Ascii::fill_next_event(), HepMC::IO_ExtendedAscii::fill_particle_data_table(), and HepMC::IO_Ascii::fill_particle_data_table().

8.1.3.20 `bool HepMC::CommonIO::read_io_ascii (std::istream * is, GenEvent * evt)`

Definition at line 77 of file CommonIO.cc.

References HepMC::GenVertex::add_particle_in(), HepMC::GenEvent::add_vertex(), HepMC::GenEvent::barcode_to_vertex(), HepMC::TempParticleMap::end_vertex(), HepMC::Temp-

ParticleMap::order_begin(), HepMC::TempParticleMap::order_end(), p, read_vertex(), HepMC::GenEvent::set_event_number(), HepMC::GenEvent::set_random_states(), HepMC::GenEvent::set_signal_process_id(), HepMC::GenEvent::set_signal_process_vertex(), v, and HepMC::GenEvent::weights().

Referenced by HepMC::IO_GenEvent::fill_next_event(), and HepMC::IO_Ascii::fill_next_event().

8.1.3.21 bool HepMC::CommonIO::read_io_extendedascii (std::istream * *is*, GenEvent * *evt*)

Definition at line 132 of file CommonIO.cc.

References HepMC::GenVertex::add_particle_in(), HepMC::GenEvent::add_vertex(), HepMC::GenEvent::barcode_to_vertex(), HepMC::TempParticleMap::end_vertex(), HepMC::TempParticleMap::order_begin(), HepMC::TempParticleMap::order_end(), p, read_heavy_ion(), read_pdf_info(), read_vertex(), HepMC::GenEvent::set_alphaQCD(), HepMC::GenEvent::set_alphaQED(), HepMC::GenEvent::set_beam_particles(), HepMC::GenEvent::set_event_number(), HepMC::GenEvent::set_event_scale(), HepMC::GenEvent::set_heavy_ion(), HepMC::GenEvent::set_mpi(), HepMC::GenEvent::set_pdf_info(), HepMC::GenEvent::set_random_states(), HepMC::GenEvent::set_signal_process_id(), HepMC::GenEvent::set_signal_process_vertex(), v, and HepMC::GenEvent::weights().

Referenced by HepMC::IO_GenEvent::fill_next_event(), and HepMC::IO_ExtendedAscii::fill_next_event().

8.1.3.22 bool HepMC::CommonIO::read_io_genevent (std::istream * *is*, GenEvent * *evt*)

Definition at line 203 of file CommonIO.cc.

References HepMC::GenVertex::add_particle_in(), HepMC::GenEvent::add_vertex(), HepMC::GenEvent::barcode_to_vertex(), HepMC::TempParticleMap::end_vertex(), HepMC::TempParticleMap::order_begin(), HepMC::TempParticleMap::order_end(), p, read_heavy_ion(), read_pdf_info(), read_vertex(), HepMC::GenEvent::set_alphaQCD(), HepMC::GenEvent::set_alphaQED(), HepMC::GenEvent::set_beam_particles(), HepMC::GenEvent::set_event_number(), HepMC::GenEvent::set_event_scale(), HepMC::GenEvent::set_heavy_ion(), HepMC::GenEvent::set_mpi(), HepMC::GenEvent::set_pdf_info(), HepMC::GenEvent::set_random_states(), HepMC::GenEvent::set_signal_process_id(), HepMC::GenEvent::set_signal_process_vertex(), v, and HepMC::GenEvent::weights().

Referenced by HepMC::IO_GenEvent::fill_next_event().

8.1.3.23 bool HepMC::CommonIO::read_io_particle_data_table (std::istream *, ParticleDataTable *)

ParticleDataTable (p. 213) is deprecated. We include this method for reading old files which may have **ParticleData** (p. 206) information.

Definition at line 278 of file CommonIO.cc.

References read_particle_data().

Referenced by HepMC::IO_ExtendedAscii::fill_particle_data_table(), and HepMC::IO_Ascii::fill_particle_data_table().

8.1.3.24 **HeavyIon * HepMC::CommonIO::read_heavy_ion** (std::istream *) [protected]

Definition at line 286 of file CommonIO.cc.

Referenced by read_io_extendedascii(), and read_io_genevent().

8.1.3.25 **PdfInfo * HepMC::CommonIO::read_pdf_info** (std::istream *) [protected]

Definition at line 312 of file CommonIO.cc.

Referenced by read_io_extendedascii(), and read_io_genevent().

8.1.3.26 **GenParticle * HepMC::CommonIO::read_particle** (std::istream *, TempParticleMap &) [protected]

Definition at line 370 of file CommonIO.cc.

References HepMC::TempParticleMap::addEndParticle(), HepMC::ascii, p, and HepMC::Flow::set_icode().

Referenced by read_vertex().

8.1.3.27 **GenVertex * HepMC::CommonIO::read_vertex** (std::istream *, TempParticleMap &) [protected]

Definition at line 334 of file CommonIO.cc.

References read_particle(), and v.

Referenced by read_io_ascii(), read_io_extendedascii(), and read_io_genevent().

8.1.3.28 **ParticleData * HepMC::CommonIO::read_particle_data** (std::istream *, ParticleDataTable *) [protected]

ParticleDataTable (p.213) is deprecated. We include this method for reading old files which may have **ParticleData** (p.206) information.

Definition at line 419 of file CommonIO.cc.

References HepMC::ParticleDataTable::insert().

Referenced by read_io_particle_data_table().

The documentation for this class was generated from the following files:

- **CommonIO.h**
- **CommonIO.cc**

8.2 HepMC::detail::disable_if<, > Struct Template Reference

internal - used by SimpleVector to decide if a class is arithmetic

```
#include <enable_if.h>
```

8.2.1 Detailed Description

```
template<bool, class> struct HepMC::detail::disable_if<, >
```

internal - used by SimpleVector to decide if a class is arithmetic

Definition at line 33 of file enable_if.h.

The documentation for this struct was generated from the following file:

- enable_if.h

8.3 HepMC::detail::disable_if< false, T > Struct Template Reference

internal - used by SimpleVector to decide if a class is arithmetic

```
#include <enable_if.h>
```

Public Types

- **typedef T type**
check type of class T

8.3.1 Detailed Description

```
template<class T> struct HepMC::detail::disable_if< false, T >
```

internal - used by SimpleVector to decide if a class is arithmetic

Definition at line 38 of file enable_if.h.

8.3.2 Member Typedef Documentation

8.3.2.1 `template<class T> typedef T HepMC::detail::disable_if< false, T >::type`

check type of class T

Definition at line 40 of file enable_if.h.

The documentation for this struct was generated from the following file:

- `enable_if.h`

8.4 HepMC::detail::enable_if<, > Struct Template Reference

internal - used to decide if a class is arithmetic

```
#include <enable_if.h>
```

8.4.1 Detailed Description

```
template<bool, class> struct HepMC::detail::enable_if<, >
```

internal - used to decide if a class is arithmetic

Definition at line 17 of file enable_if.h.

The documentation for this struct was generated from the following file:

- enable_if.h

8.5 HepMC::detail::enable_if< true, T > Struct Template Reference

internal - use if class T is arithmetic

```
#include <enable_if.h>
```

Public Types

- **typedef T type**
check type of class T

8.5.1 Detailed Description

```
template<class T> struct HepMC::detail::enable_if< true, T >
```

internal - use if class T is arithmetic

Definition at line 22 of file enable_if.h.

8.5.2 Member Typedef Documentation

8.5.2.1 `template<class T> typedef T HepMC::detail::enable_if< true, T >::type`

check type of class T

Definition at line 24 of file enable_if.h.

The documentation for this struct was generated from the following file:

- `enable_if.h`

8.6 HepMC::Flow Class Reference

The flow object.

```
#include <Flow.h>
```

Public Types

- `typedef std::map< int, int >::iterator iterator`
iterator for flow pattern container
- `typedef std::map< int, int >::const_iterator const_iterator`
const iterator for flow pattern container

Public Member Functions

- **Flow** (**GenParticle** *particle_owner=0)
default constructor
- **Flow** (const **Flow** &)
copy
- virtual ~**Flow** ()
- void **swap** (**Flow** &other)
swap
- **Flow** & **operator=** (const **Flow** &)
make a copy
- bool **operator==** (const **Flow** &a) const
equality
- bool **operator!=** (const **Flow** &a) const
inequality
- void **print** (std::ostream &ostr=std::cout) const
*print **Flow** (p. 39) information to ostr*
- std::vector< **HepMC::GenParticle** * > **connected_partners** (int code, int code_index=1, int num_indices=2) const
- std::vector< **HepMC::GenParticle** * > **dangling_connected_partners** (int code, int code_index=1, int num_indices=2) const
- const **GenParticle** * **particle_owner** () const
*find particle owning this **Flow** (p. 39)*
- int **icode** (int code_index=1) const
flow code
- **Flow** **set_icode** (int code_index, int code)

set flow code

- **Flow** **set_unique_icode** (int code_index=1)

set unique flow code

- bool **empty** () const

return true if there is no flow container

- int **size** () const

size of flow pattern container

- void **clear** ()

clear flow patterns

- bool **erase** (int code_index)

empty flow pattern container

- **iterator** **begin** ()

beginning of flow pattern container

- **iterator** **end** ()

end of flow pattern container

- **const_iterator** **begin** () const

beginning of flow pattern container

- **const_iterator** **end** () const

end of flow pattern container

Protected Member Functions

- void **connected_partners** (std::vector< **HepMC::GenParticle** * > *output, int code, int code_index, int num_indices) const

for internal use only

- void **dangling_connected_partners** (std::vector< **HepMC::GenParticle** * > *output, std::vector< **HepMC::GenParticle** * > *visited_particles, int code, int code_index, int num_indices) const

for internal use only

Friends

- std::ostream & **operator**<< (std::ostream &ostr, const **Flow** &f)

for printing

8.6.1 Detailed Description

The flow object.

The particle's flow object keeps track of an arbitrary number of flow patterns within a graph (i.e. color flow, charge flow, lepton number flow, ...) **Flow** (p. 39) patterns are coded with an integer, in the same manner as in Herwig.

Definition at line 66 of file Flow.h.

8.6.2 Member Typedef Documentation

8.6.2.1 `typedef std::map<int,int>::iterator HepMC::Flow::iterator`

iterator for flow pattern container

Definition at line 126 of file Flow.h.

8.6.2.2 `typedef std::map<int,int>::const_iterator HepMC::Flow::const_iterator`

const iterator for flow pattern container

Definition at line 128 of file Flow.h.

8.6.3 Constructor & Destructor Documentation

8.6.3.1 `HepMC::Flow::Flow (GenParticle * particle_owner = 0)`

default constructor

Definition at line 13 of file Flow.cc.

8.6.3.2 `HepMC::Flow::Flow (const Flow &)`

copy

copies both the m_icode AND the m_particle_owner

Definition at line 17 of file Flow.cc.

8.6.3.3 `HepMC::Flow::~Flow () [virtual]`

Definition at line 24 of file Flow.cc.

8.6.4 Member Function Documentation

8.6.4.1 `void HepMC::Flow::swap (Flow & other)`

swap

Definition at line 28 of file Flow.cc.

References m_icode, and m_particle_owner.

Referenced by HepMC::GenParticle::swap().

8.6.4.2 Flow & HepMC::Flow::operator= (const Flow &) [inline]

make a copy

copies only the `m_icode` ... not the `particle_owner` this is intuitive behaviour so you can do `oneparticle->flow() = otherparticle->flow()`

Definition at line 201 of file `Flow.h`.

References `m_icode`.

8.6.4.3 bool HepMC::Flow::operator== (const Flow & a) const [inline]

equality

equivalent flows have the same flow codes for all `flow_numbers` (i.e. their `m_icode` maps are identical), but they need not have the same `m_particle` owner

Definition at line 192 of file `Flow.h`.

References `m_icode`.

8.6.4.4 bool HepMC::Flow::operator!= (const Flow & a) const [inline]

inequality

Definition at line 198 of file `Flow.h`.

8.6.4.5 void HepMC::Flow::print (std::ostream & ostr = std::cout) const

print **Flow** (p. 39) information to `ostr`

Definition at line 34 of file `Flow.cc`.

8.6.4.6 std::vector< GenParticle * > HepMC::Flow::connected_partners (int code, int code_index = 1, int num_indices = 2) const

returns all connected particles which have "code" in any of the `num_indices` beginning with index `code_index`.

Returns all flow partners which have "code" in any of the `num_indices` beginning with index `code_index`. `m_particle_owner` is included in the result. Return is by value since the set should never be very big. EXAMPLE: if you want to find all flow partners that have the same code in indices 2,3,4 as particle `p` has in index 2, you would use: `set<GenParticle*> result = p->flow().connected_partners(p->flow().icode(2),2,3);`

Definition at line 38 of file `Flow.cc`.

References `icode()`.

8.6.4.7 std::vector< GenParticle * > HepMC::Flow::dangling_connected_partners (int code, int code_index = 1, int num_indices = 2) const

same as `connected_partners`, but returns only those particles which are connected to ≤ 1 other particles (i.e. the flow line "dangles" at these particles)

Definition at line 108 of file `Flow.cc`.

References icode().

8.6.4.8 `const GenParticle * HepMC::Flow::particle_owner () const` [inline]

find particle owning this **Flow** (p. 39)

Definition at line 160 of file Flow.h.

8.6.4.9 `int HepMC::Flow::icode (int code_index = 1) const` [inline]

flow code

Definition at line 163 of file Flow.h.

Referenced by `connected_partners()`, `dangling_connected_partners()`, and `HepMC::GenParticle::flow()`.

8.6.4.10 `Flow HepMC::Flow::set_icode (int code_index, int code)` [inline]

set flow code

Definition at line 167 of file Flow.h.

Referenced by `HepMC::IO_GenEvent::read_particle()`, `HepMC::CommonIO::read_particle()`, and `HepMC::GenParticle::set_flow()`.

8.6.4.11 `Flow HepMC::Flow::set_unique_icode (int code_index = 1)` [inline]

set unique flow code

use this method if you want to assign a unique flow code, but do not want the burden of choosing it yourself

Definition at line 171 of file Flow.h.

Referenced by `HepMC::GenParticle::set_flow()`.

8.6.4.12 `bool HepMC::Flow::empty () const` [inline]

return true if there is no flow container

Definition at line 177 of file Flow.h.

8.6.4.13 `int HepMC::Flow::size () const` [inline]

size of flow pattern container

Definition at line 178 of file Flow.h.

8.6.4.14 `void HepMC::Flow::clear ()` [inline]

clear flow patterns

Definition at line 179 of file Flow.h.

8.6.4.15 `bool HepMC::Flow::erase (int code_index)` [inline]

empty flow pattern container

Definition at line 180 of file Flow.h.

8.6.4.16 `Flow::iterator HepMC::Flow::begin ()` [inline]

beginning of flow pattern container

Definition at line 183 of file Flow.h.

8.6.4.17 `Flow::iterator HepMC::Flow::end ()` [inline]

end of flow pattern container

Definition at line 184 of file Flow.h.

8.6.4.18 `Flow::const_iterator HepMC::Flow::begin () const` [inline]

beginning of flow pattern container

Definition at line 185 of file Flow.h.

8.6.4.19 `Flow::const_iterator HepMC::Flow::end () const` [inline]

end of flow pattern container

Definition at line 186 of file Flow.h.

8.6.4.20 `void HepMC::Flow::connected_partners (std::vector< HepMC::GenParticle * > * output, int code, int code_index, int num_indices) const` [protected]

for internal use only

protected: for recursive use by `Flow::connected_partners()` (p. 42)

Definition at line 60 of file Flow.cc.

References `HepMC::GenParticle::end_vertex()`, `HepMC::family`, `HepMC::not_in_vector()`, `p`, `HepMC::GenVertex::particles_begin()`, `HepMC::GenVertex::particles_end()`, and `HepMC::GenParticle::production_vertex()`.

8.6.4.21 `void HepMC::Flow::dangling_connected_partners (std::vector< HepMC::GenParticle * > * output, std::vector< HepMC::GenParticle * > * visited_particles, int code, int code_index, int num_indices) const` [protected]

for internal use only

protected: for recursive use by `Flow::dangling_connected_partners` (p. 42)

Definition at line 123 of file Flow.cc.

References HepMC::GenParticle::end_vertex(), HepMC::family, HepMC::not_in_vector(), p, HepMC::GenVertex::particles_begin(), HepMC::GenVertex::particles_end(), and HepMC::GenParticle::production_vertex().

8.6.5 Friends And Related Function Documentation

8.6.5.1 std::ostream& operator<< (std::ostream & *ostr*, const Flow & *f*) [friend]

for printing

Definition at line 190 of file Flow.cc.

The documentation for this class was generated from the following files:

- **Flow.h**
- **Flow.cc**

8.7 HepMC::FourVector Class Reference

FourVector (p. 46) is a simple representation of a physics 4 vector.

```
#include <SimpleVector.h>
```

Public Member Functions

- **FourVector** (double xin, double yin, double zin, double tin=0)
constructor requiring at least x, y, and z
- **FourVector** (double t)
constructor requiring only t
- **FourVector** ()
- template<class T> **FourVector** (const T &v, typename detail::disable_if< detail::is_arithmetic< T >::value, void >::type *=0)
- **FourVector** (const **FourVector** &v)
copy constructor
- void **swap** (**FourVector** &other)
swap
- double **px** () const
return px
- double **py** () const
return py
- double **pz** () const
return pz
- double **e** () const
return E
- double **x** () const
return x
- double **y** () const
return y
- double **z** () const
return z
- double **t** () const
return t
- double **m2** () const
Invariant mass squared.

- double **m** () const
*Invariant mass. If **m2()** (p. 50) is negative then $-\sqrt{-m2()}$ is returned.*
- double **perp2** () const
Transverse component of the spatial vector squared.
- double **perp** () const
Transverse component of the spatial vector (R in cylindrical system).
- double **mag** () const
Magnitude of the spatial vector.
- double **theta** () const
The polar angle.
- double **phi** () const
The azimuth angle.
- double **rho** () const
spatial vector component magnitude
- **FourVector** & **operator=** (const **FourVector** &)
make a copy
- bool **operator==** (const **FourVector** &) const
equality
- bool **operator!=** (const **FourVector** &) const
inequality
- double **pseudoRapidity** () const
Returns the pseudo-rapidity, i.e. $-\ln(\tan(\theta/2))$.
- double **eta** () const
Pseudorapidity (of the space part).
- void **set** (double x, double y, double z, double t)
set x, y, z, and t
- void **setX** (double x)
set x
- void **setY** (double y)
set y
- void **setZ** (double z)
set z
- void **setT** (double t)
set t

- void **setPx** (double x)
set px
- void **setPy** (double y)
set py
- void **setPz** (double z)
set pz
- void **setE** (double t)
set E

8.7.1 Detailed Description

FourVector (p. 46) is a simple representation of a physics 4 vector.

For compatibility with existing code, the basic expected geometrical access methods are provided. Also, there is a templated constructor that will take another vector (HepLorentzVector, GenVector, ...) which must have the following methods: **x()** (p. 50), **y()** (p. 50), **z()** (p. 50), **t()** (p. 50).

Examples:

VectorConversion.h.

Definition at line 42 of file SimpleVector.h.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 HepMC::FourVector::FourVector (double *xin*, double *yin*, double *zin*, double *tin* = 0) [inline]

constructor requiring at least x, y, and z

Definition at line 47 of file SimpleVector.h.

8.7.2.2 HepMC::FourVector::FourVector (double *t*) [inline]

constructor requiring only t

Definition at line 51 of file SimpleVector.h.

8.7.2.3 HepMC::FourVector::FourVector () [inline]

Definition at line 54 of file SimpleVector.h.

8.7.2.4 template<class T> HepMC::FourVector::FourVector (const T & *v*, typename detail::disable_if< detail::is_arithmetic< T >::value, void >::type * = 0) [inline]

templated constructor this is used ONLY if T is not arithmetic

Definition at line 60 of file SimpleVector.h.

8.7.2.5 HepMC::FourVector::FourVector (const FourVector & v) [inline]

copy constructor

Definition at line 65 of file SimpleVector.h.

8.7.3 Member Function Documentation

8.7.3.1 void HepMC::FourVector::swap (FourVector & other)

swap

Referenced by HepMC::GenVertex::swap(), and HepMC::GenParticle::swap().

8.7.3.2 double HepMC::FourVector::px () const [inline]

return px

Definition at line 70 of file SimpleVector.h.

Referenced by main(), HepMC::operator<<(), HepMC::GenParticle::print(), and HepMC::IO_HEPEVT::write_event().

8.7.3.3 double HepMC::FourVector::py () const [inline]

return py

Definition at line 71 of file SimpleVector.h.

Referenced by main(), HepMC::operator<<(), HepMC::GenParticle::print(), and HepMC::IO_HEPEVT::write_event().

8.7.3.4 double HepMC::FourVector::pz () const [inline]

return pz

Definition at line 72 of file SimpleVector.h.

Referenced by main(), HepMC::operator<<(), HepMC::GenParticle::print(), and HepMC::IO_HEPEVT::write_event().

8.7.3.5 double HepMC::FourVector::e () const [inline]

return E

Definition at line 73 of file SimpleVector.h.

Referenced by main(), HepMC::operator<<(), HepMC::GenParticle::print(), and HepMC::IO_HEPEVT::write_event().

8.7.3.6 double HepMC::FourVector::x () const [inline]

return x

Definition at line 75 of file SimpleVector.h.

Referenced by main(), HepMC::operator<<(), HepMC::GenVertex::point3d(), and HepMC::GenVertex::print().

8.7.3.7 double HepMC::FourVector::y () const [inline]

return y

Definition at line 76 of file SimpleVector.h.

Referenced by main(), HepMC::GenVertex::point3d(), and HepMC::GenVertex::print().

8.7.3.8 double HepMC::FourVector::z () const [inline]

return z

Definition at line 77 of file SimpleVector.h.

Referenced by main(), HepMC::GenVertex::point3d(), and HepMC::GenVertex::print().

8.7.3.9 double HepMC::FourVector::t () const [inline]

return t

Definition at line 78 of file SimpleVector.h.

Referenced by main(), and HepMC::GenVertex::print().

8.7.3.10 double HepMC::FourVector::m2 () const

Invariant mass squared.

Referenced by main().

8.7.3.11 double HepMC::FourVector::m () const

Invariant mass. If **m2()** (p. 50) is negative then -sqrt(-m2()) is returned.

Referenced by main().

8.7.3.12 double HepMC::FourVector::perp2 () const

Transverse component of the spatial vector squared.

Referenced by main().

8.7.3.13 double HepMC::FourVector::perp () const

Transverse component of the spatial vector (R in cylindrical system).

Referenced by main().

8.7.3.14 double HepMC::FourVector::mag () const

Magnitude of the spatial vector.

Referenced by main().

8.7.3.15 double HepMC::FourVector::theta () const

The polar angle.

Referenced by main().

8.7.3.16 double HepMC::FourVector::phi () const

The azimuth angle.

Referenced by main().

8.7.3.17 double HepMC::FourVector::rho () const

spatial vector component magnitude

Referenced by main().

8.7.3.18 FourVector& HepMC::FourVector::operator= (const FourVector &) const

make a copy

8.7.3.19 bool HepMC::FourVector::operator== (const FourVector &) const

equality

8.7.3.20 bool HepMC::FourVector::operator!= (const FourVector &) const

inequality

8.7.3.21 double HepMC::FourVector::pseudoRapidity () const

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

Referenced by main().

8.7.3.22 double HepMC::FourVector::eta () const

Pseudorapidity (of the space part).

Referenced by main().

8.7.3.23 void HepMC::FourVector::set (double *x*, double *y*, double *z*, double *t*)

set *x*, *y*, *z*, and *t*

Referenced by `main()`.

8.7.3.24 void HepMC::FourVector::setX (double *x*) [inline]

set *x*

Definition at line 103 of file `SimpleVector.h`.

Referenced by `main()`.

8.7.3.25 void HepMC::FourVector::setY (double *y*) [inline]

set *y*

Definition at line 104 of file `SimpleVector.h`.

Referenced by `main()`.

8.7.3.26 void HepMC::FourVector::setZ (double *z*) [inline]

set *z*

Definition at line 105 of file `SimpleVector.h`.

Referenced by `main()`.

8.7.3.27 void HepMC::FourVector::setT (double *t*) [inline]

set *t*

Definition at line 106 of file `SimpleVector.h`.

Referenced by `main()`.

8.7.3.28 void HepMC::FourVector::setPx (double *x*) [inline]

set *px*

Definition at line 108 of file `SimpleVector.h`.

Referenced by `main()`.

8.7.3.29 void HepMC::FourVector::setPy (double *y*) [inline]

set *py*

Definition at line 109 of file `SimpleVector.h`.

Referenced by `main()`.

8.7.3.30 void HepMC::FourVector::setPz (double *z*) [inline]

set pz

Definition at line 110 of file SimpleVector.h.

Referenced by main().

8.7.3.31 void HepMC::FourVector::setE (double *t*) [inline]

set E

Definition at line 111 of file SimpleVector.h.

Referenced by main().

The documentation for this class was generated from the following file:

- **SimpleVector.h**

8.8 HepMC::GenEvent Class Reference

The **GenEvent** (p. 54) class is the core of **HepMC** (p. 19).

```
#include <GenEvent.h>
```

Public Member Functions

- **GenEvent** (int signal_process_id=0, int event_number=0, **GenVertex** *signal_vertex=0, const **WeightContainer** &weights=std::vector< double >(), const std::vector< long > &randomstates=std::vector< long >())
*default constructor creates null pointers to **HeavyIon** (p. 121) and **PdfInfo** (p. 220)*
- **GenEvent** (int signal_process_id, int event_number, **GenVertex** *signal_vertex, const **WeightContainer** &weights, const std::vector< long > &randomstates, const **HeavyIon** &ion, const **PdfInfo** &pdf)
*explicit constructor that takes **HeavyIon** (p. 121) and **PdfInfo** (p. 220)*
- **GenEvent** (const **GenEvent** &inevent)
deep copy
- **GenEvent** & operator= (const **GenEvent** &inevent)
make a deep copy
- virtual ~**GenEvent** ()
deletes all vertices/particles in this evt
- void **swap** (**GenEvent** &other)
swap
- void **print** (std::ostream &ostr=std::cout) const
dumps to ostr
- void **print_version** (std::ostream &ostr=std::cout) const
dumps release version to ostr
- **GenParticle** * **barcode_to_particle** (int barCode) const
assign a barcode to a particle
- **GenVertex** * **barcode_to_vertex** (int barCode) const
assign a barcode to a vertex
- int **signal_process_id** () const
unique signal process id
- int **event_number** () const
event number
- int **mpi** () const
number of multi parton interactions

- double **event_scale** () const
energy scale, see hep-ph/0109068
- double **alphaQCD** () const
QCD coupling, see hep-ph/0109068.
- double **alphaQED** () const
- **GenVertex * signal_process_vertex** () const
pointer to the vertex containing the signal process
- bool **valid_beam_particles** () const
test to see if we have two valid beam particles
- std::pair< **HepMC::GenParticle ***, **HepMC::GenParticle *** > **beam_particles** () const
pair of pointers to the two incoming beam particles
- **WeightContainer & weights** ()
*direct access to **WeightContainer** (p. 237)*
- const **WeightContainer & weights** () const
*direct access to **WeightContainer** (p. 237)*
- **HeavyIon *const heavy_ion** () const
*access the **HeavyIon** (p. 121) container if it exists*
- **HeavyIon * heavy_ion** ()
- **PdfInfo *const pdf_info** () const
*access the **PdfInfo** (p. 220) container if it exists*
- **PdfInfo * pdf_info** ()
- std::vector< long > **random_states** () const
vector of integers containing information about the random state
- void **set_signal_process_id** (int id)
set unique signal process id
- void **set_event_number** (int eventno)
set event number
- void **set_mpi** (int)
Use this to set the number of multi parton interactions in each event.
- void **set_event_scale** (double scale)
set energy scale
- void **set_alphaQCD** (double a)
set QCD coupling

- void **set_alphaQED** (double a)
set QED coupling
- void **set_signal_process_vertex** (GenVertex *)
set pointer to the vertex containing the signal process
- bool **set_beam_particles** (GenParticle *, GenParticle *)
set incoming beam particles
- bool **set_beam_particles** (std::pair< HepMC::GenParticle *, HepMC::GenParticle * > const &)
use a pair of GenParticle's to set incoming beam particles*
- void **set_random_states** (const std::vector< long > &randomstates)
provide random state information
- void **set_heavy_ion** (const HeavyIon &ion)
*provide a pointer to the **HeavyIon** (p. 121) container*
- void **set_pdf_info** (const PdfInfo &p)
*provide a pointer to the **PdfInfo** (p. 220) container*
- int **particles_size** () const
how many particle barcodes exist?
- bool **particles_empty** () const
return true if there are no particle barcodes
- int **vertices_size** () const
how many vertex barcodes exist?
- bool **vertices_empty** () const
return true if there are no vertex barcodes
- bool **add_vertex** (GenVertex *vtx)
adds to evt and adopts
- bool **remove_vertex** (GenVertex *vtx)
erases vtx from evt
- void **clear** ()
empties the entire event
- **vertex_const_iterator** **vertices_begin** () const
begin vertex iteration
- **vertex_const_iterator** **vertices_end** () const
end vertex iteration
- **vertex_iterator** **vertices_begin** ()

begin vertex iteration

- **vertex_iterator vertices_end ()**
end vertex iteration
- **particle_const_iterator particles_begin () const**
begin particle iteration
- **particle_const_iterator particles_end () const**
end particle iteration
- **particle_iterator particles_begin ()**
begin particle iteration
- **particle_iterator particles_end ()**
end particle iteration

Protected Member Functions

- **bool set_barcode (GenParticle *p, int suggested_barcode=0)**
*set the barcode - intended for use by **GenParticle** (p. 86)*
- **bool set_barcode (GenVertex *v, int suggested_barcode=0)**
*set the barcode - intended for use by **GenVertex** (p. 95)*
- **void remove_barcode (GenParticle *p)**
*intended for use by **GenParticle** (p. 86)*
- **void remove_barcode (GenVertex *v)**
*intended for use by **GenVertex** (p. 95)*
- **void delete_all_vertices ()**
delete all vertices owned by this event

Friends

- class **GenParticle**
- class **GenVertex**
- class **vertex_const_iterator**
- class **vertex_iterator**
- class **particle_const_iterator**
- class **particle_iterator**

Classes

- class **particle_const_iterator**
const particle iterator
- class **particle_iterator**
non-const particle iterator
- class **vertex_const_iterator**
const vertex iterator
- class **vertex_iterator**
non-const vertex iterator

8.8.1 Detailed Description

The **GenEvent** (p. 54) class is the core of **HepMC** (p. 19).

HepMC::GenEvent (p. 54) contains information about generated particles. **GenEvent** (p. 54) is structured as a set of vertices which contain the particles.

Examples:

`example_BuildEventFromScratch.cc`, `example_EventSelection.cc`, `example_MyHerwig.cc`, `example_MyPythia.cc`, `example_MyPythiaOnlyToHepMC.cc`, `example_MyPythiaRead.cc`, `example_MyPythiaWithEventSelection.cc`, `example_PythiaParticle.cc`, `example_UsingIterators.cc`, `testHerwigCopies.cc`, and `testPythiaCopies.cc`.

Definition at line 142 of file `GenEvent.h`.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 **HepMC::GenEvent::GenEvent** (int *signal_process_id* = 0, int *event_number* = 0, **GenVertex** * *signal_vertex* = 0, const **WeightContainer** & *weights* = std::vector< double >(), const std::vector< long > & *randomstates* = std::vector< long >())

default constructor creates null pointers to **HeavyIon** (p. 121) and **PdfInfo** (p. 220)

This constructor only allows null pointers to **HeavyIon** (p. 121) and **PdfInfo** (p. 220)

note: default values for `m_event_scale`, `m_alphaQCD`, `m_alphaQED` are as suggested in hep-ph/0109068, "Generic Interface..."

Definition at line 18 of file `GenEvent.cc`.

8.8.2.2 **HepMC::GenEvent::GenEvent** (int *signal_process_id*, int *event_number*, **GenVertex** * *signal_vertex*, const **WeightContainer** & *weights*, const std::vector< long > & *randomstates*, const **HeavyIon** & *ion*, const **PdfInfo** & *pdf*)

explicit constructor that takes **HeavyIon** (p. 121) and **PdfInfo** (p. 220)

GenEvent (p. 54) makes its own copy of **HeavyIon** (p. 121) and **PdfInfo** (p. 220)

note: default values for `m_event_scale`, `m_alphaQCD`, `m_alphaQED` are as suggested in `hep-ph/0109068`, "Generic Interface..."

Definition at line 46 of file `GenEvent.cc`.

8.8.2.3 HepMC::GenEvent::GenEvent (const GenEvent & *inevent*)

deep copy

deep copy - makes a copy of all vertices!

Definition at line 75 of file `GenEvent.cc`.

References `add_vertex()`, `beam_particles()`, `GenParticle`, `GenVertex`, `p`, `particles_begin()`, `particles_end()`, `random_states()`, `set_beam_particles()`, `set_random_states()`, `set_signal_process_vertex()`, `signal_process_vertex()`, `v`, `vertices_begin()`, `vertices_end()`, and `weights()`.

8.8.2.4 HepMC::GenEvent::~~GenEvent () [virtual]

deletes all vertices/particles in this evt

Deep destructor. deletes all vertices/particles in this evt

Definition at line 172 of file `GenEvent.cc`.

References `delete_all_vertices()`.

8.8.3 Member Function Documentation

8.8.3.1 GenEvent & HepMC::GenEvent::operator= (const GenEvent & *inevent*)

make a deep copy

best practices implementation

Definition at line 183 of file `GenEvent.cc`.

References `swap()`.

8.8.3.2 void HepMC::GenEvent::swap (GenEvent & *other*)

swap

Definition at line 143 of file `GenEvent.cc`.

References `m_alphaQCD`, `m_alphaQED`, `m_beam_particle_1`, `m_beam_particle_2`, `m_event_number`, `m_event_scale`, `m_heavy_ion`, `m_mpi`, `m_particle_barcodes`, `m_pdf_info`, `m_random_states`, `m_signal_process_id`, `m_signal_process_vertex`, `m_vertex_barcodes`, `m_weights`, `HepMC::WeightContainer::swap()`, `vertices_begin()`, and `vertices_end()`.

Referenced by `operator=()`.

8.8.3.3 void HepMC::GenEvent::print (std::ostream & *ostr* = std::cout) const

dumps to `ostr`

dumps the content of this event to ostr to dump to cout use: `event.print()`; if you want to write this event to file `outfile.txt` you could use: `std::ofstream outfile("outfile.txt"); event.print(outfile);`

Examples:

`example_BuildEventFromScratch.cc`, `example_MyHerwig.cc`, `testHerwig-Copies.cc`, and `testPythiaCopies.cc`.

Definition at line 191 of file `GenEvent.cc`.

References `alphaQCD()`, `alphaQED()`, `HepMC::GenVertex::barcode()`, `beam_particles()`, `HepMC::WeightContainer::end()`, `event_number()`, `event_scale()`, `particles_size()`, `signal_process_id()`, `signal_process_vertex()`, `HepMC::WeightContainer::size()`, `vertices_end()`, `vertices_size()`, and `weights()`.

Referenced by `main()`.

8.8.3.4 `void HepMC::GenEvent::print_version (std::ostream & ostr = std::cout) const`

dumps release version to ostr

Definition at line 246 of file `GenEvent.cc`.

References `HepMC::writeVersion()`.

8.8.3.5 `GenParticle * HepMC::GenEvent::barcode_to_particle (int barCode) const [inline]`

assign a barcode to a particle

Each vertex or particle has a barcode, which is just an integer which uniquely identifies it inside the event (i.e. there is a one to one mapping between particle memory addresses and particle barcodes... and the same applied for vertices).

The value of a barcode has NO MEANING and NO ORDER! For the user's convenience, when an event is read in via an `IO_method` from an indexed list (like the HEPEVT common block), then the index will become the barcode for that particle.

Particle barcodes are always positive integers. The barcodes are chosen and set automatically when a vertex or particle comes under the ownership of an event (i.e. it is contained in an event).

Definition at line 618 of file `GenEvent.h`.

8.8.3.6 `GenVertex * HepMC::GenEvent::barcode_to_vertex (int barCode) const [inline]`

assign a barcode to a vertex

Each vertex or particle has a barcode, which is just an integer which uniquely identifies it inside the event (i.e. there is a one to one mapping between particle memory addresses and particle barcodes... and the same applied for vertices).

The value of a barcode has NO MEANING and NO ORDER! For the user's convenience, when an event is read in via an `IO_method` from an indexed list (like the HEPEVT common block), then the index will become the barcode for that particle.

Vertex barcodes are always negative integers. The barcodes are chosen and set automatically when a vertex or particle comes under the ownership of an event (i.e. it is contained in an event).

Definition at line 638 of file GenEvent.h.

Referenced by HepMC::compareVertices(), HepMC::CommonIO::read_io_ascii(), HepMC::CommonIO::read_io_extendedascii(), and HepMC::CommonIO::read_io_genevent().

8.8.3.7 int HepMC::GenEvent::signal_process_id () const [inline]

unique signal process id

The integer ID that uniquely specifies this signal process, i.e. MSUB in Pythia. It is necessary to package this with each event rather than with the run because many processes may be generated within one run.

Definition at line 522 of file GenEvent.h.

Referenced by HepMC::compareGenEvent(), print(), HepMC::IO_GenEvent::write_event(), HepMC::IO_ExtendedAscii::write_event(), HepMC::IO_AsciiParticles::write_event(), and HepMC::IO_Ascii::write_event().

8.8.3.8 int HepMC::GenEvent::event_number () const [inline]

event number

Examples:

`example_EventSelection.cc`, `example_MyPythiaRead.cc`, `testHerwigCopies.cc`, and `testPythiaCopies.cc`.

Definition at line 525 of file GenEvent.h.

Referenced by HepMC::compareGenEvent(), main(), print(), HepMC::IO_HEPEVT::write_event(), HepMC::IO_GenEvent::write_event(), HepMC::IO_ExtendedAscii::write_event(), HepMC::IO_AsciiParticles::write_event(), and HepMC::IO_Ascii::write_event().

8.8.3.9 int HepMC::GenEvent::mpi () const [inline]

number of multi parton interactions

Returns the number of multi parton interactions in the event. This number is -1 if it is not set.

Definition at line 529 of file GenEvent.h.

Referenced by HepMC::compareGenEvent(), HepMC::IO_GenEvent::write_event(), and HepMC::IO_ExtendedAscii::write_event().

8.8.3.10 double HepMC::GenEvent::event_scale () const [inline]

energy scale, see hep-ph/0109068

Definition at line 531 of file GenEvent.h.

Referenced by HepMC::compareGenEvent(), print(), HepMC::IO_GenEvent::write_event(), HepMC::IO_ExtendedAscii::write_event(), HepMC::IO_AsciiParticles::write_event(), and HepMC::IO_Ascii::write_event().

8.8.3.11 double HepMC::GenEvent::alphaQCD () const [inline]

QCD coupling, see hep-ph/0109068.

Definition at line 533 of file GenEvent.h.

Referenced by HepMC::compareGenEvent(), print(), HepMC::IO_GenEvent::write_event(), HepMC::IO_ExtendedAscii::write_event(), HepMC::IO_AsciiParticles::write_event(), and HepMC::IO_Ascii::write_event().

8.8.3.12 double HepMC::GenEvent::alphaQED () const [inline]

QED coupling, see hep-ph/0109068

Definition at line 535 of file GenEvent.h.

Referenced by HepMC::compareGenEvent(), print(), HepMC::IO_GenEvent::write_event(), HepMC::IO_ExtendedAscii::write_event(), HepMC::IO_AsciiParticles::write_event(), and HepMC::IO_Ascii::write_event().

8.8.3.13 GenVertex * HepMC::GenEvent::signal_process_vertex () const [inline]

pointer to the vertex containing the signal process

returns a (mutable) pointer to the signal process vertex

Definition at line 537 of file GenEvent.h.

Referenced by HepMC::compareSignalProcessVertex(), GenEvent(), print(), HepMC::IO_GenEvent::write_event(), HepMC::IO_ExtendedAscii::write_event(), HepMC::IO_AsciiParticles::write_event(), and HepMC::IO_Ascii::write_event().

8.8.3.14 bool HepMC::GenEvent::valid_beam_particles () const

test to see if we have two valid beam particles

Definition at line 485 of file GenEvent.cc.

References p, particles_begin(), and particles_end().

8.8.3.15 std::pair< HepMC::GenParticle *, HepMC::GenParticle * > HepMC::GenEvent::beam_particles () const [inline]

pair of pointers to the two incoming beam particles

Definition at line 659 of file GenEvent.h.

Referenced by HepMC::compareBeamParticles(), GenEvent(), print(), HepMC::IO_GenEvent::write_event(), and HepMC::IO_ExtendedAscii::write_event().

8.8.3.16 WeightContainer & HepMC::GenEvent::weights () [inline]

direct access to **WeightContainer** (p.237)

direct access to the weights container is allowed. Thus you can use `myevt.weights()[2]`; to access element 2 of the weights. or use `myevt.weights().push_back(mywgt);` to add an element. and you can set the weights with `myevt.weights() = myvector;`

Definition at line 542 of file GenEvent.h.

Referenced by `HepMC::compareWeights()`, `GenEvent()`, `print()`, `HepMC::CommonIO::read_io_ascii()`, `HepMC::CommonIO::read_io_extendedascii()`, `HepMC::CommonIO::read_io_genevent()`, `HepMC::IO_GenEvent::write_event()`, `HepMC::IO_ExtendedAscii::write_event()`, `HepMC::IO_AsciiParticles::write_event()`, and `HepMC::IO_Ascii::write_event()`.

8.8.3.17 `const WeightContainer & HepMC::GenEvent::weights () const [inline]`

direct access to **WeightContainer** (p. 237)

Definition at line 544 of file GenEvent.h.

8.8.3.18 `HeavyIon *const HepMC::GenEvent::heavy_ion () const [inline]`

access the **HeavyIon** (p. 121) container if it exists

Definition at line 547 of file GenEvent.h.

Referenced by `HepMC::compareGenEvent()`, `HepMC::IO_GenEvent::write_event()`, and `HepMC::IO_ExtendedAscii::write_event()`.

8.8.3.19 `HeavyIon * HepMC::GenEvent::heavy_ion () [inline]`

Definition at line 550 of file GenEvent.h.

8.8.3.20 `PdfInfo *const HepMC::GenEvent::pdf_info () const [inline]`

access the **PdfInfo** (p. 220) container if it exists

Definition at line 553 of file GenEvent.h.

Referenced by `HepMC::compareGenEvent()`, `HepMC::IO_GenEvent::write_event()`, and `HepMC::IO_ExtendedAscii::write_event()`.

8.8.3.21 `PdfInfo * HepMC::GenEvent::pdf_info () [inline]`

Definition at line 556 of file GenEvent.h.

8.8.3.22 `std::vector< long > HepMC::GenEvent::random_states () const [inline]`

vector of integers containing information about the random state

Vector of integers which specify the random number generator's state for this event. It is left to the generator to make use of this. We envision a vector of `RndmStatesTags` to be included with a run class which would specify the meaning of the `random_states`.

Definition at line 564 of file GenEvent.h.

Referenced by `HepMC::compareGenEvent()`, `GenEvent()`, `HepMC::IO_GenEvent::write_event()`, `HepMC::IO_ExtendedAscii::write_event()`, `HepMC::IO_AsciiParticles::write_event()`, and `HepMC::IO_Ascii::write_event()`.

8.8.3.23 `void HepMC::GenEvent::set_signal_process_id (int id) [inline]`

set unique signal process id

Examples:

`example_MyHerwig.cc`, `example_MyPythia.cc`, `example_MyPythiaRead.cc`, `example_PythiaParticle.cc`, and `testHerwigCopies.cc`.

Definition at line 567 of file `GenEvent.h`.

Referenced by `main()`, `HepMC::CommonIO::read_io_ascii()`, `HepMC::CommonIO::read_io_extendedascii()`, and `HepMC::CommonIO::read_io_genevent()`.

8.8.3.24 `void HepMC::GenEvent::set_event_number (int eventno) [inline]`

set event number

Examples:

`example_MyHerwig.cc`, `example_MyPythia.cc`, `example_MyPythiaRead.cc`, `example_PythiaParticle.cc`, and `testHerwigCopies.cc`.

Definition at line 570 of file `GenEvent.h`.

Referenced by `HepMC::IO_HERWIG::fill_next_event()`, `HepMC::IO_HEPEVT::fill_next_event()`, `main()`, `HepMC::CommonIO::read_io_ascii()`, `HepMC::CommonIO::read_io_extendedascii()`, and `HepMC::CommonIO::read_io_genevent()`.

8.8.3.25 `void HepMC::GenEvent::set_mpi (int) [inline]`

Use this to set the number of multi parton interactions in each event.

Examples:

`example_MyPythia.cc`, `example_MyPythiaOnlyToHepMC.cc`, `example_MyPythiaWithEventSelection.cc`, and `testPythiaCopies.cc`.

Definition at line 574 of file `GenEvent.h`.

Referenced by `main()`, `HepMC::CommonIO::read_io_extendedascii()`, and `HepMC::CommonIO::read_io_genevent()`.

8.8.3.26 `void HepMC::GenEvent::set_event_scale (double scale) [inline]`

set energy scale

Definition at line 578 of file `GenEvent.h`.

Referenced by `HepMC::CommonIO::read_io_extendedascii()`, and `HepMC::CommonIO::read_io_genevent()`.

8.8.3.27 void HepMC::GenEvent::set_alphaQCD (double *a*) [inline]

set QCD coupling

Definition at line 580 of file GenEvent.h.

Referenced by HepMC::CommonIO::read_io_extendedascii(), and HepMC::CommonIO::read_io_genevent().

8.8.3.28 void HepMC::GenEvent::set_alphaQED (double *a*) [inline]

set QED coupling

Definition at line 582 of file GenEvent.h.

Referenced by HepMC::CommonIO::read_io_extendedascii(), and HepMC::CommonIO::read_io_genevent().

8.8.3.29 void HepMC::GenEvent::set_signal_process_vertex (GenVertex *) [inline]

set pointer to the vertex containing the signal process

Examples:

example_BuildEventFromScratch.cc.

Definition at line 584 of file GenEvent.h.

References add_vertex().

Referenced by HepMC::IO_HERWIG::fill_next_event(), GenEvent(), main(), HepMC::CommonIO::read_io_ascii(), HepMC::CommonIO::read_io_extendedascii(), and HepMC::CommonIO::read_io_genevent().

8.8.3.30 bool HepMC::GenEvent::set_beam_particles (GenParticle * *bp1*, GenParticle * *bp2*)

set incoming beam particles

construct the beam particle information using pointers to **GenParticle** (p.86) returns false if either GenParticle* is null

Definition at line 503 of file GenEvent.cc.

Referenced by HepMC::IO_HERWIG::fill_next_event(), HepMC::IO_HEPEVT::fill_next_event(), GenEvent(), HepMC::CommonIO::read_io_extendedascii(), HepMC::CommonIO::read_io_genevent(), and set_beam_particles().

8.8.3.31 bool HepMC::GenEvent::set_beam_particles (std::pair< HepMC::GenParticle *, HepMC::GenParticle * > const & *bp*)

use a pair of GenParticle*'s to set incoming beam particles

construct the beam particle information using a std::pair of pointers to **GenParticle** (p.86) returns false if either GenParticle* is null

Definition at line 512 of file GenEvent.cc.

References `set_beam_particles()`.

8.8.3.32 `void HepMC::GenEvent::set_random_states (const std::vector< long > & randomstates) [inline]`

provide random state information

Definition at line 595 of file GenEvent.h.

Referenced by `GenEvent()`, `HepMC::CommonIO::read_io_ascii()`, `HepMC::CommonIO::read_io_extendedascii()`, and `HepMC::CommonIO::read_io_genevent()`.

8.8.3.33 `void HepMC::GenEvent::set_heavy_ion (const HeavyIon & ion) [inline]`

provide a pointer to the **HeavyIon** (p.121) container

Definition at line 589 of file GenEvent.h.

Referenced by `HepMC::CommonIO::read_io_extendedascii()`, and `HepMC::CommonIO::read_io_genevent()`.

8.8.3.34 `void HepMC::GenEvent::set_pdf_info (const PdfInfo & p) [inline]`

provide a pointer to the **PdfInfo** (p.220) container

Definition at line 592 of file GenEvent.h.

References `p`.

Referenced by `HepMC::CommonIO::read_io_extendedascii()`, and `HepMC::CommonIO::read_io_genevent()`.

8.8.3.35 `int HepMC::GenEvent::particles_size () const [inline]`

how many particle barcodes exist?

Definition at line 645 of file GenEvent.h.

Referenced by `HepMC::compareParticles()`, `print()`, and `HepMC::IO_AsciiParticles::write_event()`.

8.8.3.36 `bool HepMC::GenEvent::particles_empty () const [inline]`

return true if there are no particle barcodes

Definition at line 648 of file GenEvent.h.

Referenced by `delete_all_vertices()`.

8.8.3.37 `int HepMC::GenEvent::vertices_size () const [inline]`

how many vertex barcodes exist?

Definition at line 651 of file GenEvent.h.

Referenced by HepMC::compareVertices(), print(), HepMC::IO_GenEvent::write_event(), HepMC::IO_ExtendedAscii::write_event(), HepMC::IO_AsciiParticles::write_event(), and HepMC::IO_Ascii::write_event().

8.8.3.38 bool HepMC::GenEvent::vertices_empty () const [inline]

return true if there are no vertex barcodes

Definition at line 654 of file GenEvent.h.

Referenced by delete_all_vertices().

8.8.3.39 bool HepMC::GenEvent::add_vertex (GenVertex * vtx)

adds to evt and adopts

returns true if successful - generally will only return false if the inserted vertex is already included in the event.

Examples:

example_BuildEventFromScratch.cc.

Definition at line 252 of file GenEvent.cc.

References HepMC::GenVertex::barcode(), HepMC::GenVertex::parent_event(), remove_vertex(), and HepMC::GenVertex::set_parent_event_().

Referenced by HepMC::IO_HERWIG::build_end_vertex(), HepMC::IO_HEPEVT::build_end_vertex(), HepMC::IO_HERWIG::build_production_vertex(), HepMC::IO_HEPEVT::build_production_vertex(), HepMC::IO_HERWIG::fill_next_event(), HepMC::IO_HEPEVT::fill_next_event(), GenEvent(), main(), HepMC::CommonIO::read_io_ascii(), HepMC::CommonIO::read_io_extendedascii(), HepMC::CommonIO::read_io_genevent(), and set_signal_process_vertex().

8.8.3.40 bool HepMC::GenEvent::remove_vertex (GenVertex * vtx)

erases vtx from evt

this removes vtx from the event but does NOT delete it. returns True if an entry vtx existed in the table and was erased

Definition at line 275 of file GenEvent.cc.

References HepMC::GenVertex::barcode(), HepMC::GenVertex::parent_event(), and HepMC::GenVertex::set_parent_event_().

Referenced by add_vertex().

8.8.3.41 void HepMC::GenEvent::clear ()

empties the entire event

remove all information from the event deletes all vertices/particles in this evt

Definition at line 283 of file GenEvent.cc.

References delete_all_vertices().

8.8.3.42 `vertex_const_iterator HepMC::GenEvent::vertices_begin () const`
[inline]

begin vertex iteration

Examples:

`example_UsingIterators.cc.`

Definition at line 294 of file GenEvent.h.

Referenced by HepMC::compareVertices(), GenEvent(), main(), swap(), HepMC::IO_HEPEVT::write_event(), HepMC::IO_GenEvent::write_event(), HepMC::IO_ExtendedAscii::write_event(), and HepMC::IO_Ascii::write_event().

8.8.3.43 `vertex_const_iterator HepMC::GenEvent::vertices_end () const`
[inline]

end vertex iteration

Examples:

`example_UsingIterators.cc.`

Definition at line 298 of file GenEvent.h.

Referenced by HepMC::compareVertices(), GenEvent(), main(), print(), swap(), HepMC::IO_HEPEVT::write_event(), HepMC::IO_GenEvent::write_event(), HepMC::IO_ExtendedAscii::write_event(), and HepMC::IO_Ascii::write_event().

8.8.3.44 `vertex_iterator HepMC::GenEvent::vertices_begin ()` [inline]

begin vertex iteration

Definition at line 351 of file GenEvent.h.

8.8.3.45 `vertex_iterator HepMC::GenEvent::vertices_end ()` [inline]

end vertex iteration

Definition at line 355 of file GenEvent.h.

8.8.3.46 `particle_const_iterator HepMC::GenEvent::particles_begin () const`
[inline]

begin particle iteration

Examples:

`example_BuildEventFromScratch.cc`, `example_EventSelection.cc`, `example_MyPythiaWithEventSelection.cc`, and `example_UsingIterators.cc.`

Definition at line 413 of file GenEvent.h.

Referenced by `HepMC::compareParticles()`, `GenEvent()`, `main()`, `IsGoodEvent::operator()`, `IsGoodEventMyPythia::operator()`, `IsEventGood::operator()`, `valid_beam_particles()`, and `HepMC::IO_AsciiParticles::write_event()`.

8.8.3.47 `particle_const_iterator HepMC::GenEvent::particles_end () const`
[inline]

end particle iteration

Examples:

`example_BuildEventFromScratch.cc`, `example_EventSelection.cc`, `example_MyPythiaWithEventSelection.cc`, and `example_UsingIterators.cc`.

Definition at line 417 of file `GenEvent.h`.

Referenced by `HepMC::compareParticles()`, `GenEvent()`, `main()`, `IsGoodEvent::operator()`, `IsGoodEventMyPythia::operator()`, `IsEventGood::operator()`, `valid_beam_particles()`, and `HepMC::IO_AsciiParticles::write_event()`.

8.8.3.48 `particle_iterator HepMC::GenEvent::particles_begin ()` [inline]

begin particle iteration

Definition at line 466 of file `GenEvent.h`.

8.8.3.49 `particle_iterator HepMC::GenEvent::particles_end ()` [inline]

end particle iteration

Definition at line 470 of file `GenEvent.h`.

8.8.3.50 `bool HepMC::GenEvent::set_barcode (GenParticle * p, int suggested_barcode = 0)` [protected]

set the barcode - intended for use by **GenParticle** (p. 86)

Definition at line 347 of file `GenEvent.cc`.

References `p`.

Referenced by `HepMC::GenVertex::set_parent_event_()`, `HepMC::GenVertex::suggest_barcode()`, and `HepMC::GenParticle::suggest_barcode()`.

8.8.3.51 `bool HepMC::GenEvent::set_barcode (GenVertex * v, int suggested_barcode = 0)` [protected]

set the barcode - intended for use by **GenVertex** (p. 95)

Definition at line 418 of file `GenEvent.cc`.

References `v`.

8.8.3.52 `void HepMC::GenEvent::remove_barcode (GenParticle * p)` [inline, protected]

intended for use by **GenParticle** (p. 86)

Definition at line 599 of file GenEvent.h.

References p.

Referenced by `HepMC::GenParticle::set_end_vertex_()`, `HepMC::GenVertex::set_parent_event_()`, `HepMC::GenParticle::set_production_vertex_()`, `HepMC::GenParticle::~GenParticle()`, and `HepMC::GenVertex::~GenVertex()`.

8.8.3.53 `void HepMC::GenEvent::remove_barcode (GenVertex * v)` [inline, protected]

intended for use by **GenVertex** (p. 95)

Definition at line 602 of file GenEvent.h.

References v.

8.8.3.54 `void HepMC::GenEvent::delete_all_vertices ()` [protected]

delete all vertices owned by this event

deletes all vertices in the vertex container (i.e. all vertices owned by this event) The vertices are the "owners" of the particles, so as we delete the vertices, the vertex destructors are automatically deleting their particles.

Definition at line 316 of file GenEvent.cc.

References `particles_empty()`, and `vertices_empty()`.

Referenced by `clear()`, and `~GenEvent()`.

8.8.4 Friends And Related Function Documentation

8.8.4.1 `friend class GenParticle` [friend]

Definition at line 143 of file GenEvent.h.

Referenced by `GenEvent()`.

8.8.4.2 `friend class GenVertex` [friend]

Definition at line 144 of file GenEvent.h.

Referenced by `GenEvent()`.

8.8.4.3 `friend class vertex_const_iterator` [friend]

Definition at line 292 of file GenEvent.h.

Referenced by `HepMC::GenEvent::vertex_iterator::operator vertex_const_iterator()`.

8.8.4.4 friend class vertex_iterator [friend]

Definition at line 349 of file GenEvent.h.

8.8.4.5 friend class particle_const_iterator [friend]

Definition at line 411 of file GenEvent.h.

Referenced by HepMC::GenEvent::particle_iterator::operator particle_const_iterator().

8.8.4.6 friend class particle_iterator [friend]

Definition at line 464 of file GenEvent.h.

The documentation for this class was generated from the following files:

- **GenEvent.h**
- **GenEvent.cc**

8.9 HepMC::GenEvent::particle_const_iterator Class Reference

const particle iterator

```
#include <GenEvent.h>
```

Public Member Functions

- **particle_const_iterator** (const std::map< int, **HepMC::GenParticle** * >::const_iterator &i)
iterate over particles
- **particle_const_iterator** ()
- **particle_const_iterator** (const **particle_const_iterator** &i)
copy constructor
- virtual ~**particle_const_iterator** ()
- **particle_const_iterator** & operator= (const **particle_const_iterator** &i)
make a copy
- **GenParticle** * operator * (void) const
*return a pointer to **GenParticle** (p. 86)*
- **particle_const_iterator** & operator++ (void)
Pre-fix increment.
- **particle_const_iterator** operator++ (int)
Post-fix increment.
- bool operator== (const **particle_const_iterator** &a) const
equality
- bool operator!= (const **particle_const_iterator** &a) const
inequality

Protected Attributes

- std::map< int, **HepMC::GenParticle** * >::const_iterator **m_map_iterator**
*const iterator to the **GenParticle** (p. 86) map*

8.9.1 Detailed Description

const particle iterator

HepMC::GenEvent::particle_const_iterator (p. 72) is used to iterate over all particles in the event.

Examples:

`example_BuildEventFromScratch.cc`, `example_EventSelection.cc`, and `example_MyPythiaWithEventSelection.cc`.

Definition at line 375 of file `GenEvent.h`.

8.9.2 Constructor & Destructor Documentation

8.9.2.1 `HepMC::GenEvent::particle_const_iterator::particle_const_iterator (const std::map< int, HepMC::GenParticle * >::const_iterator & i) [inline]`

iterate over particles

Definition at line 380 of file `GenEvent.h`.

8.9.2.2 `HepMC::GenEvent::particle_const_iterator::particle_const_iterator () [inline]`

Definition at line 383 of file `GenEvent.h`.

8.9.2.3 `HepMC::GenEvent::particle_const_iterator::particle_const_iterator (const particle_const_iterator & i) [inline]`

copy constructor

Definition at line 385 of file `GenEvent.h`.

8.9.2.4 `virtual HepMC::GenEvent::particle_const_iterator::~~particle_const_iterator () [inline, virtual]`

Definition at line 387 of file `GenEvent.h`.

8.9.3 Member Function Documentation

8.9.3.1 `particle_const_iterator& HepMC::GenEvent::particle_const_iterator::operator= (const particle_const_iterator & i) [inline]`

make a copy

Definition at line 389 of file `GenEvent.h`.

References `m_map_iterator`.

8.9.3.2 `GenParticle* HepMC::GenEvent::particle_const_iterator::operator * (void) const [inline]`

return a pointer to `GenParticle` (p. 86)

Definition at line 393 of file `GenEvent.h`.

References `m_map_iterator`.

8.9.3.3 `particle_const_iterator& HepMC::GenEvent::particle_const_iterator::operator++ (void) [inline]`

Pre-fix increment.

Definition at line 396 of file GenEvent.h.

References `m_map_iterator`.

8.9.3.4 `particle_const_iterator HepMC::GenEvent::particle_const_iterator::operator++ (int) [inline]`

Post-fix increment.

Definition at line 399 of file GenEvent.h.

8.9.3.5 `bool HepMC::GenEvent::particle_const_iterator::operator== (const particle_const_iterator & a) const [inline]`

equality

Definition at line 402 of file GenEvent.h.

References `m_map_iterator`.

8.9.3.6 `bool HepMC::GenEvent::particle_const_iterator::operator!= (const particle_const_iterator & a) const [inline]`

inequality

Definition at line 405 of file GenEvent.h.

References `m_map_iterator`.

8.9.4 Member Data Documentation

8.9.4.1 `std::map<int,HepMC::GenParticle*>::const_iterator HepMC::GenEvent::particle_const_iterator::m_map_iterator [protected]`

const iterator to the **GenParticle** (p.86) map

Definition at line 409 of file GenEvent.h.

Referenced by `operator*()`, `operator!=()`, `operator++()`, `operator=()`, and `operator==()`.

The documentation for this class was generated from the following file:

- **GenEvent.h**

8.10 HepMC::GenEvent::particle_iterator Class Reference

non-const particle iterator

```
#include <GenEvent.h>
```

Public Member Functions

- **particle_iterator** (const std::map< int, **HepMC::GenParticle** * >::iterator &i)
iterate over particles
- **particle_iterator** ()
- **particle_iterator** (const **particle_iterator** &i)
copy constructor
- virtual ~**particle_iterator** ()
- **particle_iterator** & **operator**= (const **particle_iterator** &i)
make a copy
- **operator particle_const_iterator** () const
const particle iterator
- **GenParticle** * **operator** * (void) const
*return pointer to **GenParticle** (p. 86)*
- **particle_iterator** & **operator**++ (void)
Pre-fix increment.
- **particle_iterator** **operator**++ (int)
Post-fix increment.
- bool **operator**== (const **particle_iterator** &a) const
equality
- bool **operator**!= (const **particle_iterator** &a) const
inequality

Protected Attributes

- std::map< int, **HepMC::GenParticle** * >::iterator **m_map_iterator**
*iterator for **GenParticle** (p. 86) map*

8.10.1 Detailed Description

non-const particle iterator

HepMC::GenEvent::particle_iterator (p.75) is used to iterate over all particles in the event.

Examples:

`example_UsingIterators.cc.`

Definition at line 426 of file GenEvent.h.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 `HepMC::GenEvent::particle_iterator::particle_iterator (const std::map< int, HepMC::GenParticle * >::iterator & i) [inline]`

iterate over particles

Definition at line 431 of file GenEvent.h.

8.10.2.2 `HepMC::GenEvent::particle_iterator::particle_iterator () [inline]`

Definition at line 433 of file GenEvent.h.

8.10.2.3 `HepMC::GenEvent::particle_iterator::particle_iterator (const particle_iterator & i) [inline]`

copy constructor

Definition at line 435 of file GenEvent.h.

8.10.2.4 `virtual HepMC::GenEvent::particle_iterator::~particle_iterator () [inline, virtual]`

Definition at line 436 of file GenEvent.h.

8.10.3 Member Function Documentation

8.10.3.1 `particle_iterator& HepMC::GenEvent::particle_iterator::operator= (const particle_iterator & i) [inline]`

make a copy

Definition at line 438 of file GenEvent.h.

References `m_map_iterator`.

8.10.3.2 `HepMC::GenEvent::particle_iterator::operator particle_const_iterator () const [inline]`

const particle iterator

Definition at line 443 of file GenEvent.h.

References `m_map_iterator`, and `HepMC::GenEvent::particle_const_iterator`.

8.10.3.3 GenParticle* HepMC::GenEvent::particle_iterator::operator * (void) const [inline]

return pointer to **GenParticle** (p. 86)

Definition at line 446 of file GenEvent.h.

References m_map_iterator.

8.10.3.4 particle_iterator& HepMC::GenEvent::particle_iterator::operator++ (void) [inline]

Pre-fix increment.

Definition at line 449 of file GenEvent.h.

References m_map_iterator.

8.10.3.5 particle_iterator HepMC::GenEvent::particle_iterator::operator++ (int) [inline]

Post-fix increment.

Definition at line 452 of file GenEvent.h.

8.10.3.6 bool HepMC::GenEvent::particle_iterator::operator== (const particle_iterator & a) const [inline]

equality

Definition at line 455 of file GenEvent.h.

References m_map_iterator.

8.10.3.7 bool HepMC::GenEvent::particle_iterator::operator!= (const particle_iterator & a) const [inline]

inequality

Definition at line 458 of file GenEvent.h.

References m_map_iterator.

8.10.4 Member Data Documentation

8.10.4.1 std::map<int,HepMC::GenParticle*>::iterator HepMC::Gen- Event::particle_iterator::m_map_iterator [protected]

iterator for **GenParticle** (p. 86) map

Definition at line 462 of file GenEvent.h.

Referenced by operator *(), operator particle_const_iterator(), operator !=(), operator ++(), operator ==(), and operator =().

The documentation for this class was generated from the following file:

- `GenEvent.h`

8.11 HepMC::GenEvent::vertex_const_iterator Class Reference

const vertex iterator

```
#include <GenEvent.h>
```

Public Member Functions

- **vertex_const_iterator** (const std::map< int, **HepMC::GenVertex** *, std::greater< int > >::const_iterator &i)
constructor requiring vertex information
- **vertex_const_iterator** ()
- **vertex_const_iterator** (const **vertex_const_iterator** &i)
copy constructor
- virtual ~**vertex_const_iterator** ()
- **vertex_const_iterator** &operator= (const **vertex_const_iterator** &i)
make a copy
- **GenVertex** *operator* (void) const
*return a pointer to a **GenVertex** (p. 95)*
- **vertex_const_iterator** &operator++ (void)
Pre-fix increment.
- **vertex_const_iterator** operator++ (int)
Post-fix increment.
- bool operator== (const **vertex_const_iterator** &a) const
equality
- bool operator!= (const **vertex_const_iterator** &a) const
inequality

Protected Attributes

- std::map< int, **HepMC::GenVertex** *, std::greater< int > >::const_iterator **m_map_iterator**
const iterator to a vertex map

8.11.1 Detailed Description

const vertex iterator

HepMC::GenEvent::vertex_const_iterator (p. 79) is used to iterate over all vertices in the event.

Definition at line 256 of file GenEvent.h.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 `HepMC::GenEvent::vertex_const_iterator::vertex_const_iterator (const std::map< int, HepMC::GenVertex *, std::greater< int > >::const_iterator & i) [inline]`

constructor requiring vertex information

Definition at line 261 of file GenEvent.h.

8.11.2.2 `HepMC::GenEvent::vertex_const_iterator::vertex_const_iterator () [inline]`

Definition at line 265 of file GenEvent.h.

8.11.2.3 `HepMC::GenEvent::vertex_const_iterator::vertex_const_iterator (const vertex_const_iterator & i) [inline]`

copy constructor

Definition at line 267 of file GenEvent.h.

8.11.2.4 `virtual HepMC::GenEvent::vertex_const_iterator::~~vertex_const_iterator () [inline, virtual]`

Definition at line 269 of file GenEvent.h.

8.11.3 Member Function Documentation

8.11.3.1 `vertex_const_iterator& HepMC::GenEvent::vertex_const_iterator::operator= (const vertex_const_iterator & i) [inline]`

make a copy

Definition at line 271 of file GenEvent.h.

References `m_map_iterator`.

8.11.3.2 `GenVertex* HepMC::GenEvent::vertex_const_iterator::operator * (void) const [inline]`

return a pointer to a **GenVertex** (p.95)

Definition at line 274 of file GenEvent.h.

References `m_map_iterator`.

8.11.3.3 `vertex_const_iterator& HepMC::GenEvent::vertex_const_iterator::operator++ (void) [inline]`

Pre-fix increment.

Definition at line 276 of file GenEvent.h.

References `m_map_iterator`.

8.11.3.4 `vertex_const_iterator HepMC::GenEvent::vertex_const_iterator::operator++ (int) [inline]`

Post-fix increment.

Definition at line 279 of file GenEvent.h.

8.11.3.5 `bool HepMC::GenEvent::vertex_const_iterator::operator== (const vertex_const_iterator & a) const [inline]`

equality

Definition at line 282 of file GenEvent.h.

References `m_map_iterator`.

8.11.3.6 `bool HepMC::GenEvent::vertex_const_iterator::operator!= (const vertex_const_iterator & a) const [inline]`

inequality

Definition at line 285 of file GenEvent.h.

References `m_map_iterator`.

8.11.4 Member Data Documentation

8.11.4.1 `std::map<int,HepMC::GenVertex*,std::greater<int> >::const_iterator HepMC::GenEvent::vertex_const_iterator::m_map_iterator [protected]`

const iterator to a vertex map

Definition at line 290 of file GenEvent.h.

Referenced by `operator*()`, `operator!=()`, `operator++()`, `operator=()`, and `operator==()`.

The documentation for this class was generated from the following file:

- **GenEvent.h**

8.12 HepMC::GenEvent::vertex_iterator Class Reference

non-const vertex iterator

```
#include <GenEvent.h>
```

Public Member Functions

- **vertex_iterator** (const std::map< int, **HepMC::GenVertex** *, std::greater< int > >::iterator &i)
constructor requiring vertex information
- **vertex_iterator** ()
- **vertex_iterator** (const **vertex_iterator** &i)
copy constructor
- virtual ~**vertex_iterator** ()
- **vertex_iterator** & **operator**= (const **vertex_iterator** &i)
make a copy
- **operator vertex_const_iterator** () const
const vertex iterator
- **GenVertex** * **operator** * (void) const
*return a pointer to a **GenVertex** (p. 95)*
- **vertex_iterator** & **operator**++ (void)
Pre-fix increment.
- **vertex_iterator** **operator**++ (int)
Post-fix increment.
- bool **operator**== (const **vertex_iterator** &a) const
equality
- bool **operator**!= (const **vertex_iterator** &a) const
inequality

Protected Attributes

- std::map< int, **HepMC::GenVertex** *, std::greater< int > >::iterator **m_map_iterator**
iterator to the vertex map

8.12.1 Detailed Description

non-const vertex iterator

HepMC::GenEvent::vertex_iterator (p.82) is used to iterate over all vertices in the event.

Examples:

`example_UsingIterators.cc.`

Definition at line 308 of file GenEvent.h.

8.12.2 Constructor & Destructor Documentation

8.12.2.1 HepMC::GenEvent::vertex_iterator::vertex_iterator (const std::map< int, HepMC::GenVertex *, std::greater< int > >::iterator & i) [inline]

constructor requiring vertex information

Definition at line 313 of file GenEvent.h.

8.12.2.2 HepMC::GenEvent::vertex_iterator::vertex_iterator () [inline]

Definition at line 317 of file GenEvent.h.

8.12.2.3 HepMC::GenEvent::vertex_iterator::vertex_iterator (const vertex_iterator & i) [inline]

copy constructor

Definition at line 319 of file GenEvent.h.

8.12.2.4 virtual HepMC::GenEvent::vertex_iterator::~~vertex_iterator () [inline, virtual]

Definition at line 320 of file GenEvent.h.

8.12.3 Member Function Documentation

8.12.3.1 vertex_iterator& HepMC::GenEvent::vertex_iterator::operator= (const vertex_iterator & i) [inline]

make a copy

Definition at line 322 of file GenEvent.h.

References `m_map_iterator`.

8.12.3.2 HepMC::GenEvent::vertex_iterator::operator vertex_const_iterator () const [inline]

const vertex iterator

Definition at line 327 of file GenEvent.h.

References `m_map_iterator`, and `HepMC::GenEvent::vertex_const_iterator`.

8.12.3.3 `GenVertex* HepMC::GenEvent::vertex_iterator::operator * (void) const`
[inline]

return a pointer to a **GenVertex** (p.95)

Definition at line 330 of file GenEvent.h.

References `m_map_iterator`.

8.12.3.4 `vertex_iterator& HepMC::GenEvent::vertex_iterator::operator++ (void)`
[inline]

Pre-fix increment.

Definition at line 333 of file GenEvent.h.

References `m_map_iterator`.

8.12.3.5 `vertex_iterator HepMC::GenEvent::vertex_iterator::operator++ (int)`
[inline]

Post-fix increment.

Definition at line 336 of file GenEvent.h.

8.12.3.6 `bool HepMC::GenEvent::vertex_iterator::operator== (const
vertex_iterator & a) const` [inline]

equality

Definition at line 339 of file GenEvent.h.

References `m_map_iterator`.

8.12.3.7 `bool HepMC::GenEvent::vertex_iterator::operator!= (const
vertex_iterator & a) const` [inline]

inequality

Definition at line 342 of file GenEvent.h.

References `m_map_iterator`.

8.12.4 Member Data Documentation

8.12.4.1 `std::map<int,HepMC::GenVertex*,std::greater<int> >::iterator
HepMC::GenEvent::vertex_iterator::m_map_iterator` [protected]

iterator to the vertex map

Definition at line 347 of file GenEvent.h.

Referenced by operator `*`(), operator `vertex_const_iterator()`, operator `!=()`, operator `++()`, operator `=()`, and operator `==()`.

The documentation for this class was generated from the following file:

- **GenEvent.h**

8.13 HepMC::GenParticle Class Reference

The **GenParticle** (p.86) class contains information about generated particles.

```
#include <GenParticle.h>
```

Public Member Functions

- **GenParticle** (void)
default constructor
- **GenParticle** (const **FourVector** &momentum, int pdg_id, int status=0, const **Flow** &its-flow=**Flow**(), const **Polarization** &polar=**Polarization**(0, 0))
constructor requires momentum and particle ID
- **GenParticle** (const **GenParticle** &inparticle)
shallow copy.
- virtual ~**GenParticle** ()
- void **swap** (**GenParticle** &other)
swap
- **GenParticle** & **operator=** (const **GenParticle** &inparticle)
- bool **operator==** (const **GenParticle** &) const
check for equality
- bool **operator!=** (const **GenParticle** &) const
check for inequality
- void **print** (std::ostream &ostr=std::cout) const
dump this particle's full info to ostr
- **operator HepMC::FourVector** () const
conversion operator
- **FourVector** **momentum** () const
standard 4 momentum
- int **pdg_id** () const
particle ID
- int **status** () const
HEPEVT decay status.
- **Flow** **flow** () const
particle flow
- int **flow** (int code_index) const
particle flow index

- **Polarization** **polarization** () const
polarization information
- **GenVertex** * **production_vertex** () const
pointer to the production vertex
- **GenVertex** * **end_vertex** () const
pointer to the decay vertex
- **GenEvent** * **parent_event** () const
pointer to the event that owns this particle
- double **generated_mass** () const
mass as generated
- double **generatedMass** () const
generatedMass() (p. 92) is included for backwards compatibility with **CLHEP** (p. 17) **HepMC** (p. 19)
- int **barcode** () const
particle barcode
- bool **suggest_barcode** (int the_bar_code)
In general there is no reason to "suggest_barcode".
- void **set_momentum** (const **FourVector** &vec4)
set standard 4 momentum
- void **set_pdg_id** (int id)
set particle ID
- void **set_status** (int status=0)
set decay status
- void **set_flow** (const **Flow** &f)
set particle flow
- void **set_flow** (int code_index, int code=0)
- void **set_polarization** (const **Polarization** &pol=**Polarization**(0, 0))
set polarization
- void **set_generated_mass** (const double &m)
define the actual generated mass
- void **setGeneratedMass** (const double &m)
setGeneratedMass() (p. 93) is included for backwards compatibility with **CLHEP** (p. 17) **HepMC** (p. 19)

Protected Member Functions

- void **set_production_vertex_** (**GenVertex** *productionvertex=0)
set production vertex
- void **set_end_vertex_** (**GenVertex** *decayvertex=0)
set decay vertex
- void **set_barcode_** (int the_bar_code)
*for use by **GenEvent** (p. 54) only*

Friends

- class **GenVertex**
- class **GenEvent**
- std::ostream & **operator<<** (std::ostream &, const **GenParticle** &)
print particle

8.13.1 Detailed Description

The **GenParticle** (p. 86) class contains information about generated particles.

HepMC::GenParticle (p. 86) contains momentum, generated mass, particle ID, decay status, flow, polarization, pointers to production and decay vertices and a unique barcode identifier.

Examples:

example_BuildEventFromScratch.cc, and **example_UsingIterators.cc**.

Definition at line 55 of file **GenParticle.h**.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 **HepMC::GenParticle::GenParticle** (void)

default constructor

Definition at line 14 of file **GenParticle.cc**.

8.13.2.2 **HepMC::GenParticle::GenParticle** (const **FourVector** & *momentum*, int *pdg_id*, int *status* = 0, const **Flow** & *itsflow* = **Flow**(), const **Polarization** & *polar* = **Polarization**(0, 0))

constructor requires momentum and particle ID

Definition at line 23 of file **GenParticle.cc**.

References **set_flow**().

8.13.2.3 HepMC::GenParticle::GenParticle (const GenParticle & *inparticle*)

shallow copy.

Shallow copy: does not copy the vertex pointers (note - impossible to copy vertex pointers which having the vertex and particles in/out point-back to one another – unless you copy the entire tree – which we don't want to do)

Definition at line 37 of file GenParticle.cc.

References barcode(), set_end_vertex_(), set_production_vertex_(), and suggest_barcode().

8.13.2.4 HepMC::GenParticle::~~GenParticle () [virtual]

Definition at line 58 of file GenParticle.cc.

References parent_event(), and HepMC::GenEvent::remove_barcode().

8.13.3 Member Function Documentation

8.13.3.1 void HepMC::GenParticle::swap (GenParticle & *other*)

swap

Definition at line 63 of file GenParticle.cc.

References m_barcode, m_end_vertex, m_flow, m_generated_mass, m_momentum, m_pdg_id, m_polarization, m_production_vertex, m_status, HepMC::Polarization::swap(), HepMC::Flow::swap(), and HepMC::FourVector::swap().

Referenced by operator=().

8.13.3.2 GenParticle & HepMC::GenParticle::operator= (const GenParticle & *inparticle*)

shallow.

Shallow: does not copy the vertex pointers (note - impossible to copy vertex pointers which having the vertex and particles in/out point-back to one another – unless you copy the entire tree – which we don't want to do)

Definition at line 77 of file GenParticle.cc.

References swap().

8.13.3.3 bool HepMC::GenParticle::operator== (const GenParticle &) const

check for equality

consistent with the definition of the copy constructor as a shallow constructor,.. this operator does not test the vertex pointers. Does not compare barcodes.

Definition at line 89 of file GenParticle.cc.

References generated_mass(), m_flow, momentum(), pdg_id(), polarization(), and status().

8.13.3.4 `bool HepMC::GenParticle::operator!= (const GenParticle &) const`

check for inequality

Definition at line 102 of file GenParticle.cc.

8.13.3.5 `void HepMC::GenParticle::print (std::ostream & ostr = std::cout) const`

dump this particle's full info to ostr

Dump this particle's full info to ostr, where by default particle.print(); will dump to cout.

Definition at line 106 of file GenParticle.cc.

References HepMC::GenVertex::barcode(), barcode(), HepMC::FourVector::e(), end_vertex(), momentum(), pdg_id(), polarization(), production_vertex(), HepMC::FourVector::px(), HepMC::FourVector::py(), HepMC::FourVector::pz(), and status().

8.13.3.6 `HepMC::GenParticle::operator HepMC::FourVector () const` [inline]

conversion operator

Definition at line 171 of file GenParticle.h.

8.13.3.7 `FourVector HepMC::GenParticle::momentum () const` [inline]

standard 4 momentum

Definition at line 174 of file GenParticle.h.

Referenced by HepMC::operator<<(), operator==(), and print().

8.13.3.8 `int HepMC::GenParticle::pdg_id () const` [inline]

particle ID

Definition at line 177 of file GenParticle.h.

Referenced by HepMC::operator<<(), operator==(), and print().

8.13.3.9 `int HepMC::GenParticle::status () const` [inline]

HEPEVT decay status.

Definition at line 179 of file GenParticle.h.

Referenced by HepMC::operator<<(), operator==(), and print().

8.13.3.10 `Flow HepMC::GenParticle::flow () const` [inline]

particle flow

Definition at line 186 of file GenParticle.h.

8.13.3.11 `int HepMC::GenParticle::flow (int code_index) const [inline]`

particle flow index

Definition at line 188 of file GenParticle.h.

References HepMC::Flow::icode().

8.13.3.12 `Polarization HepMC::GenParticle::polarization () const [inline]`

polarization information

Definition at line 191 of file GenParticle.h.

Referenced by operator==(), and print().

8.13.3.13 `GenVertex * HepMC::GenParticle::production_vertex () const [inline]`

pointer to the production vertex

Definition at line 181 of file GenParticle.h.

Referenced by HepMC::GenVertex::add_particle_out(), HepMC::Flow::connected_partners(), HepMC::Flow::dangling_connected_partners(), parent_event(), print(), and HepMC::GenVertex::remove_particle().

8.13.3.14 `GenVertex * HepMC::GenParticle::end_vertex () const [inline]`

pointer to the decay vertex

Definition at line 184 of file GenParticle.h.

Referenced by HepMC::GenVertex::add_particle_in(), HepMC::Flow::connected_partners(), HepMC::Flow::dangling_connected_partners(), HepMC::operator<<(), parent_event(), print(), and HepMC::GenVertex::remove_particle().

8.13.3.15 `GenEvent * HepMC::GenParticle::parent_event () const`

pointer to the event that owns this particle

Definition at line 123 of file GenParticle.cc.

References end_vertex(), HepMC::GenVertex::parent_event(), and production_vertex().

Referenced by set_end_vertex_(), set_production_vertex_(), suggest_barcode(), and ~GenParticle().

8.13.3.16 `double HepMC::GenParticle::generated_mass () const`

mass as generated

Because of precision issues, the generated mass is not always the same as the mass calculated from the momentum 4 vector. If the generated mass has been set, then **generated_mass()** (p.91) returns that value. If the generated mass has not been set, then **generated_mass()** (p.91) returns the mass calculated from the momentum 4 vector.

Definition at line 233 of file GenParticle.cc.

Referenced by `generatedMass()`, and `operator==()`.

8.13.3.17 `double HepMC::GenParticle::generatedMass () const [inline]`

generatedMass() (p. 92) is included for backwards compatibility with **CLHEP** (p. 17) **HepMC** (p. 19)

Definition at line 116 of file `GenParticle.h`.

References `generated_mass()`.

8.13.3.18 `int HepMC::GenParticle::barcode () const [inline]`

particle barcode

The barcode is the particle's reference number, every vertex in the event has a unique barcode. Particle barcodes are positive numbers, vertex barcodes are negative numbers.

Definition at line 215 of file `GenParticle.h`.

Referenced by `GenParticle()`, `HepMC::operator<<()`, `print()`, `set_end_vertex_()`, and `set_production_vertex_()`.

8.13.3.19 `bool HepMC::GenParticle::suggest_barcode (int the_bar_code)`

In general there is no reason to "suggest_barcode".

allows a barcode to be suggested for this particle. In general it is better to let the event pick the barcode for you, which is automatic. Returns TRUE if the suggested barcode has been accepted (i.e. the suggested barcode has not already been used in the event, and so it was used). Returns FALSE if the suggested barcode was rejected, or if the particle is not yet part of an event, such that it is not yet possible to know if the suggested barcode will be accepted).

Definition at line 153 of file `GenParticle.cc`.

References `parent_event()`, `HepMC::GenEvent::set_barcode()`, and `set_barcode_()`.

Referenced by `GenParticle()`.

8.13.3.20 `void HepMC::GenParticle::set_momentum (const FourVector & vec4) [inline]`

set standard 4 momentum

Definition at line 194 of file `GenParticle.h`.

8.13.3.21 `void HepMC::GenParticle::set_pdg_id (int id) [inline]`

set particle ID

Definition at line 197 of file `GenParticle.h`.

8.13.3.22 `void HepMC::GenParticle::set_status (int status = 0) [inline]`

set decay status

Definition at line 199 of file GenParticle.h.

8.13.3.23 void HepMC::GenParticle::set_flow (const Flow & f) [inline]

set particle flow

Definition at line 201 of file GenParticle.h.

Referenced by GenParticle().

8.13.3.24 void HepMC::GenParticle::set_flow (int code_index, int code = 0) [inline]

set particle flow index

Definition at line 203 of file GenParticle.h.

References HepMC::Flow::set_icode(), and HepMC::Flow::set_unique_icode().

8.13.3.25 void HepMC::GenParticle::set_polarization (const Polarization & pol = Polarization(0, 0)) [inline]

set polarization

Definition at line 212 of file GenParticle.h.

8.13.3.26 void HepMC::GenParticle::set_generated_mass (const double & m)

define the actual generated mass

If you do not call **set_generated_mass()** (p.93), then **generated_mass()** (p.91) will simply return the mass calculated from **momentum()** (p.90)

Definition at line 237 of file GenParticle.cc.

Referenced by setGeneratedMass().

8.13.3.27 void HepMC::GenParticle::setGeneratedMass (const double & m) [inline]

setGeneratedMass() (p.93) is included for backwards compatibility with CLHEP (p.17) **HepMC** (p.19)

Definition at line 140 of file GenParticle.h.

References set_generated_mass().

8.13.3.28 void HepMC::GenParticle::set_production_vertex_ (GenVertex * productionvertex = 0) [protected]

set production vertex

Definition at line 129 of file GenParticle.cc.

References barcode(), parent_event(), and HepMC::GenEvent::remove_barcode().

Referenced by `HepMC::GenVertex::add_particle_out()`, `GenParticle()`, and `HepMC::GenVertex::remove_particle()`.

8.13.3.29 `void HepMC::GenParticle::set_end_vertex_ (GenVertex * decayvertex = 0) [protected]`

set decay vertex

Definition at line 142 of file `GenParticle.cc`.

References `barcode()`, `parent_event()`, and `HepMC::GenEvent::remove_barcode()`.

Referenced by `HepMC::GenVertex::add_particle_in()`, `GenParticle()`, and `HepMC::GenVertex::remove_particle()`.

8.13.3.30 `void HepMC::GenParticle::set_barcode_ (int the_bar_code) [inline, protected]`

for use by **GenEvent** (p. 54) only

Definition at line 217 of file `GenParticle.h`.

Referenced by `suggest_barcode()`.

8.13.4 Friends And Related Function Documentation

8.13.4.1 `friend class GenVertex [friend]`

Definition at line 57 of file `GenParticle.h`.

8.13.4.2 `friend class GenEvent [friend]`

Definition at line 58 of file `GenParticle.h`.

8.13.4.3 `std::ostream& operator<< (std::ostream & ostr, const GenParticle & part) [friend]`

print particle

Definition at line 189 of file `GenParticle.cc`.

The documentation for this class was generated from the following files:

- **GenParticle.h**
- **GenParticle.cc**

8.14 HepMC::GenVertex Class Reference

GenVertex (p. 95) contains information about decay vertices.

```
#include <GenVertex.h>
```

Public Types

- typedef std::vector< **HepMC::GenParticle** * >::const_iterator **particles_in_const_iterator**
const iterator for incoming particles
- typedef std::vector< **HepMC::GenParticle** * >::const_iterator **particles_out_const_iterator**
const iterator for outgoing particles

Public Member Functions

- **GenVertex** (const **FourVector** &position=**FourVector**(0, 0, 0, 0), int id=0, const **WeightContainer** &weights=std::vector< double >())
default constructor
- **GenVertex** (const **GenVertex** &invertex)
shallow copy
- virtual ~**GenVertex** ()
- void **swap** (**GenVertex** &other)
swap
- **GenVertex** & **operator=** (const **GenVertex** &invertex)
shallow
- bool **operator==** (const **GenVertex** &a) const
equality
- bool **operator!=** (const **GenVertex** &a) const
inequality
- void **print** (std::ostream &ostr=std::cout) const
print vertex information
- double **check_momentum_conservation** () const
|Sum (mom_in-mom_out)|
- void **add_particle_in** (**GenParticle** *inparticle)
add incoming particle
- void **add_particle_out** (**GenParticle** *outparticle)
add outgoing particle

- **GenParticle * remove_particle (GenParticle *particle)**
remove a particle
- **operator HepMC::FourVector () const**
conversion operator
- **operator HepMC::ThreeVector () const**
conversion operator
- **GenEvent * parent_event () const**
pointer to the event that owns this vertex
- **ThreeVector point3d () const**
vertex position
- **FourVector position () const**
vertex position and time
- **void set_position (const FourVector &position=FourVector(0, 0, 0, 0))**
set vertex position and time
- **int id () const**
vertex ID
- **void set_id (int id)**
set vertex ID
- **int barcode () const**
unique identifier
- **bool suggest_barcode (int the_bar_code)**
In general there is no reason to "suggest_barcode".
- **WeightContainer & weights ()**
direct access to the weights container is allowed.
- **const WeightContainer & weights () const**
const direct access to the weights container
- **particles_in_const_iterator particles_in_const_begin () const**
begin iteration of incoming particles
- **particles_in_const_iterator particles_in_const_end () const**
end iteration of incoming particles
- **particles_out_const_iterator particles_out_const_begin () const**
begin iteration of outgoing particles
- **particles_out_const_iterator particles_out_const_end () const**

end iteration of outgoing particles

- **int particles_in_size () const**
number of incoming particles
- **int particles_out_size () const**
number of outgoing particles
- **vertex_iterator vertices_begin (IteratorRange range=relatives)**
begin vertex range
- **vertex_iterator vertices_end (IteratorRange)**
end vertex range
- **particle_iterator particles_begin (IteratorRange range=relatives)**
begin particle range
- **particle_iterator particles_end (IteratorRange)**
end particle range

Protected Member Functions

- **void set_parent_event_ (GenEvent *evt)**
set parent event
- **void set_barcode_ (int the_bar_code)**
set identifier
- **void change_parent_event_ (GenEvent *evt)**
for use with swap
- **int edges_size (IteratorRange range=family) const**
size
- **edge_iterator edges_begin (IteratorRange range=family) const**
begin range
- **edge_iterator edges_end (IteratorRange) const**
end range
- **void delete_adopted_particles ()**
for internal use only
- **void remove_particle_in (GenParticle *)**
for internal use only - remove particle from incoming list
- **void remove_particle_out (GenParticle *)**
for internal use only - remove particle from outgoing list

Friends

- class **GenEvent**
- class **edge_iterator**
- class **vertex_iterator**
- class **particle_iterator**
- `std::ostream & operator<< (std::ostream &, const GenVertex &)`
print vertex information

Classes

- class **edge_iterator**
edge iterator
- class **particle_iterator**
particle iterator
- class **vertex_iterator**
vertex iterator

8.14.1 Detailed Description

GenVertex (p. 95) contains information about decay vertices.

HepMC::GenVertex (p. 95) contains the position in space and time of a decay. It also contains lists of incoming and outgoing particles.

Examples:

`example_BuildEventFromScratch.cc.`

Definition at line 47 of file `GenVertex.h`.

8.14.2 Member Typedef Documentation

8.14.2.1 `typedef std::vector<HepMC::GenParticle*>::const_iterator HepMC::GenVertex::particles_in_const_iterator`

const iterator for incoming particles

Definition at line 130 of file `GenVertex.h`.

8.14.2.2 `typedef std::vector<HepMC::GenParticle*>::const_iterator HepMC::GenVertex::particles_out_const_iterator`

const iterator for outgoing particles

Definition at line 133 of file `GenVertex.h`.

8.14.3 Constructor & Destructor Documentation

8.14.3.1 HepMC::GenVertex::GenVertex (const FourVector & *position* = FourVector(0, 0, 0, 0), int *id* = 0, const WeightContainer & *weights* = std::vector< double >())

default constructor

Definition at line 14 of file GenVertex.cc.

8.14.3.2 HepMC::GenVertex::GenVertex (const GenVertex & *invertex*)

shallow copy

Shallow copy: does not copy the FULL list of particle pointers. Creates a copy of - invertex

- outgoing particles of invertex, but sets the decay vertex of these particles to NULL
- all incoming particles which do not have a creation vertex. (i.e. it creates copies of all particles which it owns) (note - impossible to copy the FULL list of particle pointers while having the vertex and particles in/out point-back to one another – unless you copy the entire tree – which we don't want to do)

Definition at line 23 of file GenVertex.cc.

References `add_particle_in()`, `add_particle_out()`, `barcode()`, `particles_in_const_begin()`, `particles_in_const_end()`, `particles_out_const_begin()`, `particles_out_const_end()`, and `suggest_barcode()`.

8.14.3.3 HepMC::GenVertex::~~GenVertex () [virtual]

Definition at line 63 of file GenVertex.cc.

References `delete_adopted_particles()`, `parent_event()`, and `HepMC::GenEvent::remove_barcode()`.

8.14.4 Member Function Documentation

8.14.4.1 void HepMC::GenVertex::swap (GenVertex & *other*)

swap

Definition at line 71 of file GenVertex.cc.

References `m_barcode`, `m_event`, `m_id`, `m_particles_in`, `m_particles_out`, `m_position`, `m_weights`, `HepMC::WeightContainer::swap()`, and `HepMC::FourVector::swap()`.

Referenced by `operator=()`.

8.14.4.2 GenVertex & HepMC::GenVertex::operator= (const GenVertex & *invertex*)

shallow

Shallow: does not copy the FULL list of particle pointers. Creates a copy of - invertex

- outgoing particles of `invertex`, but sets the decay vertex of these particles to `NULL`
- all incoming particles which do not have a creation vertex.
- it does not alter `*this`'s `m_event` (!) (i.e. it creates copies of all particles which it owns) (note - impossible to copy the FULL list of particle pointers while having the vertex and particles in/out point-back to one another – unless you copy the entire tree – which we don't want to do)

Definition at line 82 of file `GenVertex.cc`.

References `swap()`.

8.14.4.3 `bool HepMC::GenVertex::operator==(const GenVertex & a) const`

equality

Returns true if the positions and the particles in the lists of `a` and `this` are identical. Does not compare barcodes. Note that it is impossible for two vertices to point to the same particle's address, so we need to do more than just compare the particle pointers

Definition at line 103 of file `GenVertex.cc`.

References `particles_in_const_begin()`, `particles_in_const_end()`, `particles_in_size()`, `particles_out_const_begin()`, `particles_out_const_end()`, `particles_out_size()`, and `position()`.

8.14.4.4 `bool HepMC::GenVertex::operator!=(const GenVertex & a) const`

inequality

Definition at line 140 of file `GenVertex.cc`.

8.14.4.5 `void HepMC::GenVertex::print(std::ostream & ostr = std::cout) const`

print vertex information

Definition at line 145 of file `GenVertex.cc`.

References `barcode()`, `HepMC::WeightContainer::end()`, `id()`, `particles_in_const_begin()`, `particles_in_const_end()`, `particles_out_const_begin()`, `particles_out_const_end()`, `position()`, `HepMC::WeightContainer::size()`, `HepMC::FourVector::t()`, `weights()`, `HepMC::FourVector::x()`, `HepMC::FourVector::y()`, and `HepMC::FourVector::z()`.

Referenced by `HepMC::IO_HERWIG::build_production_vertex()`.

8.14.4.6 `double HepMC::GenVertex::check_momentum_conservation() const`

$|\text{Sum}(\text{mom_in} - \text{mom_out})|$

finds the difference between the total momentum out and the total momentum in vectors, and returns the magnitude of this vector i.e. returns $|\text{vec}\{\text{p_in}\} - \text{vec}\{\text{p_out}\}|$

Definition at line 253 of file `GenVertex.cc`.

References `particles_in_const_begin()`, `particles_in_const_end()`, `particles_out_const_begin()`, and `particles_out_const_end()`.

8.14.4.7 void HepMC::GenVertex::add_particle_in (GenParticle * *inparticle*)

add incoming particle

Examples:

example_BuildEventFromScratch.cc.

Definition at line 273 of file GenVertex.cc.

References HepMC::GenParticle::end_vertex(), remove_particle_in(), and HepMC::GenParticle::set_end_vertex_().

Referenced by HepMC::IO_HERWIG::build_end_vertex(), HepMC::IO_HEPEVT::build_end_vertex(), HepMC::IO_HERWIG::build_production_vertex(), HepMC::IO_HEPEVT::build_production_vertex(), HepMC::IO_HERWIG::fill_next_event(), GenVertex(), main(), HepMC::CommonIO::read_io_ascii(), HepMC::CommonIO::read_io_extendedascii(), and HepMC::CommonIO::read_io_genevent().

8.14.4.8 void HepMC::GenVertex::add_particle_out (GenParticle * *outparticle*)

add outgoing particle

Examples:

example_BuildEventFromScratch.cc.

Definition at line 284 of file GenVertex.cc.

References HepMC::GenParticle::production_vertex(), remove_particle_out(), and HepMC::GenParticle::set_production_vertex_().

Referenced by HepMC::IO_HERWIG::build_end_vertex(), HepMC::IO_HEPEVT::build_end_vertex(), HepMC::IO_HERWIG::build_production_vertex(), HepMC::IO_HEPEVT::build_production_vertex(), HepMC::IO_HERWIG::fill_next_event(), HepMC::IO_HEPEVT::fill_next_event(), GenVertex(), and main().

8.14.4.9 GenParticle * HepMC::GenVertex::remove_particle (GenParticle * *particle*)

remove a particle

remove_particle finds *particle in the in and/or out list and removes it from these lists ... it DOES NOT DELETE THE PARTICLE or its relations. You could delete the particle too as follows: delete vtx->remove_particle(particle);

this finds *particle in the in and/or out list and removes it from these lists ... it DOES NOT DELETE THE PARTICLE or its relations. you could delete the particle too as follows: delete vtx->remove_particle(particle); or if the particle has an end vertex, you could: delete vtx->remove_particle(particle)->end_vertex(); which would delete the particle's end vertex, and thus would also delete the particle, since the particle would be owned by the end vertex.

Definition at line 295 of file GenVertex.cc.

References HepMC::GenParticle::end_vertex(), HepMC::GenParticle::production_vertex(), remove_particle_in(), remove_particle_out(), HepMC::GenParticle::set_end_vertex_(), and HepMC::GenParticle::set_production_vertex_().

8.14.4.10 HepMC::GenVertex::operator HepMC::FourVector () const [inline]

conversion operator

Definition at line 366 of file GenVertex.h.

References position().

8.14.4.11 HepMC::GenVertex::operator HepMC::ThreeVector () const [inline]

conversion operator

Definition at line 368 of file GenVertex.h.

References point3d().

8.14.4.12 GenEvent * HepMC::GenVertex::parent_event () const [inline]

pointer to the event that owns this vertex

Definition at line 372 of file GenVertex.h.

Referenced by HepMC::GenEvent::add_vertex(), HepMC::GenParticle::parent_event(), HepMC::GenEvent::remove_vertex(), suggest_barcode(), and ~GenVertex().

8.14.4.13 ThreeVector HepMC::GenVertex::point3d () const [inline]

vertex position

Definition at line 374 of file GenVertex.h.

References HepMC::FourVector::x(), HepMC::FourVector::y(), and HepMC::FourVector::z().

Referenced by operator HepMC::ThreeVector().

8.14.4.14 FourVector HepMC::GenVertex::position () const [inline]

vertex position and time

Definition at line 370 of file GenVertex.h.

Referenced by HepMC::IO_HERWIG::build_end_vertex(), HepMC::IO_HEPEVT::build_end_vertex(), HepMC::IO_HERWIG::build_production_vertex(), HepMC::IO_HEPEVT::build_production_vertex(), HepMC::compareVertex(), operator HepMC::FourVector(), HepMC::operator<<(), operator==(), print(), and set_position().

8.14.4.15 void HepMC::GenVertex::set_position (const FourVector & position = FourVector(0, 0, 0, 0)) [inline]

set vertex position and time

Definition at line 388 of file GenVertex.h.

References position().

Referenced by HepMC::IO_HERWIG::build_end_vertex(), HepMC::IO_HEPEVT::build_end_vertex(), HepMC::IO_HERWIG::build_production_vertex(), and HepMC::IO_HEPEVT::build_production_vertex().

8.14.4.16 int HepMC::GenVertex::id () const [inline]

vertex ID

we don't define what you use the id for – but we imagine, for example it might code the meaning of the **weights()** (p. 103)

Definition at line 378 of file GenVertex.h.

Referenced by `print()`.

8.14.4.17 void HepMC::GenVertex::set_id (int *id*) [inline]

set vertex ID

Definition at line 392 of file GenVertex.h.

8.14.4.18 int HepMC::GenVertex::barcode () const [inline]

unique identifier

The barcode is the vertex's reference number, every vertex in the event has a unique barcode. Vertex barcodes are negative numbers, particle barcodes are positive numbers.

Definition at line 380 of file GenVertex.h.

Referenced by `HepMC::GenEvent::add_vertex()`, `HepMC::compareVertex()`, `GenVertex()`, `HepMC::operator<<()`, `print()`, `HepMC::GenParticle::print()`, `HepMC::GenEvent::print()`, `HepMC::GenEvent::remove_vertex()`, `set_parent_event_()`, `HepMC::IO_GenEvent::write_event()`, `HepMC::IO_ExtendedAscii::write_event()`, `HepMC::IO_AsciiParticles::write_event()`, and `HepMC::IO_Ascii::write_event()`.

8.14.4.19 bool HepMC::GenVertex::suggest_barcode (int *the_bar_code*)

In general there is no reason to "suggest_barcode".

allows a barcode to be suggested for this vertex. In general it is better to let the event pick the barcode for you, which is automatic. Returns TRUE if the suggested barcode has been accepted (i.e. the suggested barcode has not already been used in the event, and so it was used). Returns FALSE if the suggested barcode was rejected, or if the vertex is not yet part of an event, such that it is not yet possible to know if the suggested barcode will be accepted).

Definition at line 363 of file GenVertex.cc.

References `parent_event()`, `HepMC::GenEvent::set_barcode()`, and `set_barcode_()`.

Referenced by `GenVertex()`.

8.14.4.20 WeightContainer & HepMC::GenVertex::weights () [inline]

direct access to the weights container is allowed.

Definition at line 383 of file GenVertex.h.

Referenced by `print()`.

8.14.4.21 `const WeightContainer & HepMC::GenVertex::weights () const` [inline]

const direct access to the weights container

Definition at line 385 of file GenVertex.h.

8.14.4.22 `GenVertex::particles_in_const_iterator HepMC::GenVertex::particles_in_const_begin () const` [inline]

begin iteration of incoming particles

Definition at line 399 of file GenVertex.h.

Referenced by `check_momentum_conservation()`, `HepMC::compareVertex()`, `GenVertex()`, `operator==()`, `print()`, and `set_parent_event_()`.

8.14.4.23 `GenVertex::particles_in_const_iterator HepMC::GenVertex::particles_in_const_end () const` [inline]

end iteration of incoming particles

Definition at line 404 of file GenVertex.h.

Referenced by `check_momentum_conservation()`, `HepMC::compareVertex()`, `GenVertex()`, `operator==()`, `print()`, and `set_parent_event_()`.

8.14.4.24 `GenVertex::particles_out_const_iterator HepMC::GenVertex::particles_out_const_begin () const` [inline]

begin iteration of outgoing particles

Definition at line 409 of file GenVertex.h.

Referenced by `check_momentum_conservation()`, `HepMC::compareVertex()`, `GenVertex()`, `operator==()`, `print()`, and `set_parent_event_()`.

8.14.4.25 `GenVertex::particles_out_const_iterator HepMC::GenVertex::particles_out_const_end () const` [inline]

end iteration of outgoing particles

Definition at line 414 of file GenVertex.h.

Referenced by `check_momentum_conservation()`, `HepMC::compareVertex()`, `GenVertex()`, `operator==()`, `print()`, and `set_parent_event_()`.

8.14.4.26 `int HepMC::GenVertex::particles_in_size () const` [inline]

number of incoming particles

Definition at line 418 of file GenVertex.h.

Referenced by `HepMC::compareVertex()`, and `operator==()`.

8.14.4.27 int HepMC::GenVertex::particles_out_size () const [inline]

number of outgoing particles

Definition at line 422 of file GenVertex.h.

Referenced by HepMC::compareVertex(), and operator==().

8.14.4.28 void HepMC::GenVertex::set_parent_event_ (GenEvent * evt) [protected]

set parent event

only the **GenEvent** (p. 54) (friend) is allowed to set the parent_event, and barcode. It is done automatically anytime you add a vertex to an event

Definition at line 388 of file GenVertex.cc.

References barcode(), particles_in_const_begin(), particles_in_const_end(), particles_out_const_begin(), particles_out_const_end(), HepMC::GenEvent::remove_barcode(), and HepMC::GenEvent::set_barcode().

Referenced by HepMC::GenEvent::add_vertex(), and HepMC::GenEvent::remove_vertex().

8.14.4.29 void HepMC::GenVertex::set_barcode_ (int the_bar_code) [inline, protected]

set identifier

Definition at line 381 of file GenVertex.h.

Referenced by suggest_barcode().

8.14.4.30 void HepMC::GenVertex::change_parent_event_ (GenEvent * evt) [protected]

for use with swap

Definition at line 419 of file GenVertex.cc.

8.14.4.31 int HepMC::GenVertex::edges_size (IteratorRange range = family) const [protected]

size

Definition at line 583 of file GenVertex.cc.

References HepMC::children, HepMC::family, and HepMC::parents.

8.14.4.32 GenVertex::edge_iterator HepMC::GenVertex::edges_begin (IteratorRange range = family) const [inline, protected]

begin range

Definition at line 440 of file GenVertex.h.

Referenced by HepMC::GenVertex::vertex_iterator::vertex_iterator().

8.14.4.33 `GenVertex::edge_iterator HepMC::GenVertex::edges_end
(IteratorRange) const` [inline, protected]

end range

Definition at line 445 of file GenVertex.h.

Referenced by `HepMC::GenVertex::vertex_iterator::operator++()`, and `HepMC::GenVertex::vertex_iterator::vertex_iterator()`.

8.14.4.34 `GenVertex::vertex_iterator HepMC::GenVertex::vertices_begin
(IteratorRange range = relatives)` [inline]

begin vertex range

Definition at line 468 of file GenVertex.h.

References `vertex_iterator`.

8.14.4.35 `GenVertex::vertex_iterator HepMC::GenVertex::vertices_end
(IteratorRange)` [inline]

end vertex range

Definition at line 474 of file GenVertex.h.

References `vertex_iterator`.

8.14.4.36 `GenVertex::particle_iterator HepMC::GenVertex::particles_begin
(IteratorRange range = relatives)` [inline]

begin particle range

Definition at line 489 of file GenVertex.h.

References `particle_iterator`.

Referenced by `HepMC::Flow::connected_partners()`, and `HepMC::Flow::dangling_connected_partners()`.

8.14.4.37 `GenVertex::particle_iterator HepMC::GenVertex::particles_end
(IteratorRange)` [inline]

end particle range

Definition at line 494 of file GenVertex.h.

References `particle_iterator`.

Referenced by `HepMC::Flow::connected_partners()`, and `HepMC::Flow::dangling_connected_partners()`.

8.14.4.38 `void HepMC::GenVertex::delete_adopted_particles()` [protected]

for internal use only

deletes all particles which this vertex owns to be used by the vertex destructor and `operator=`

Definition at line 329 of file GenVertex.cc.

Referenced by `~GenVertex()`.

8.14.4.39 void HepMC::GenVertex::remove_particle_in (GenParticle *) [protected]

for internal use only - remove particle from incoming list

this finds *particle in m_particles_in and removes it from that list

Definition at line 317 of file GenVertex.cc.

References HepMC::already_in_vector().

Referenced by `add_particle_in()`, and `remove_particle()`.

8.14.4.40 void HepMC::GenVertex::remove_particle_out (GenParticle *) [protected]

for internal use only - remove particle from outgoing list

this finds *particle in m_particles_out and removes it from that list

Definition at line 323 of file GenVertex.cc.

References HepMC::already_in_vector().

Referenced by `add_particle_out()`, and `remove_particle()`.

8.14.5 Friends And Related Function Documentation

8.14.5.1 friend class GenEvent [friend]

Definition at line 51 of file GenVertex.h.

8.14.5.2 friend class edge_iterator [friend]

Definition at line 206 of file GenVertex.h.

8.14.5.3 friend class vertex_iterator [friend]

Definition at line 285 of file GenVertex.h.

Referenced by `vertices_begin()`, and `vertices_end()`.

8.14.5.4 friend class particle_iterator [friend]

Definition at line 333 of file GenVertex.h.

Referenced by `particles_begin()`, and `particles_end()`.

8.14.5.5 `std::ostream& operator<< (std::ostream & ostr, const GenVertex & vtx)` [friend]

print vertex information

Definition at line 440 of file GenVertex.cc.

The documentation for this class was generated from the following files:

- **GenVertex.h**
- **GenVertex.cc**

8.15 HepMC::GenVertex::edge_iterator Class Reference

edge iterator

```
#include <GenVertex.h>
```

Public Member Functions

- **edge_iterator** ()
- **edge_iterator** (const **GenVertex** &vtx, **IteratorRange** range=family)
used to set limits on the iteration
- **edge_iterator** (const **edge_iterator** &p)
copy
- virtual **~edge_iterator** ()
- **edge_iterator** & **operator=** (const **edge_iterator** &p)
make a copy
- **GenParticle** * **operator *** (void) const
return a pointer to a particle
- **edge_iterator** & **operator++** (void)
Pre-fix increment.
- **edge_iterator** **operator++** (int)
Post-fix increment.
- bool **operator==** (const **edge_iterator** &a) const
equality
- bool **operator!=** (const **edge_iterator** &a) const
inequality
- bool **is_parent** () const
true if parent of root vtx
- bool **is_child** () const
true if child of root vtx
- const **GenVertex** * **vertex_root** () const
root vertex of this iteration

8.15.1 Detailed Description

edge iterator

iterate over the family of edges connected to m_vertex begins with parents (incoming particles) then children (outgoing) This is not a recursive iterator ... it is a building block for the public

iterators and is intended for internal use only. The acceptable Iterator Ranges are: family, parents, children

Definition at line 172 of file GenVertex.h.

8.15.2 Constructor & Destructor Documentation

8.15.2.1 HepMC::GenVertex::edge_iterator::edge_iterator ()

Definition at line 462 of file GenVertex.cc.

8.15.2.2 HepMC::GenVertex::edge_iterator::edge_iterator (const GenVertex & *vx*, IteratorRange *range* = family)

used to set limits on the iteration

Definition at line 466 of file GenVertex.cc.

References HepMC::ancestors, HepMC::children, HepMC::descendants, HepMC::family, HepMC::GenVertex::m_particles_in, HepMC::GenVertex::m_particles_out, and HepMC::parents.

8.15.2.3 HepMC::GenVertex::edge_iterator::edge_iterator (const edge_iterator & *p*)

copy

Definition at line 517 of file GenVertex.cc.

References p.

8.15.2.4 HepMC::GenVertex::edge_iterator::~~edge_iterator () [virtual]

Definition at line 521 of file GenVertex.cc.

8.15.3 Member Function Documentation

8.15.3.1 GenVertex::edge_iterator & HepMC::GenVertex::edge_iterator::operator= (const edge_iterator & *p*)

make a copy

Definition at line 523 of file GenVertex.cc.

References p.

8.15.3.2 GenParticle * HepMC::GenVertex::edge_iterator::operator * (void) const

return a pointer to a particle

Definition at line 533 of file GenVertex.cc.

8.15.3.3 GenVertex::edge_iterator & HepMC::GenVertex::edge_iterator::operator++ (void)

Pre-fix increment.

Definition at line 538 of file GenVertex.cc.

References HepMC::family, HepMC::GenVertex::m_particles_in, HepMC::GenVertex::m_particles_out, and HepMC::parents.

8.15.3.4 GenVertex::edge_iterator HepMC::GenVertex::edge_iterator::operator++ (int)

Post-fix increment.

Definition at line 566 of file GenVertex.cc.

8.15.3.5 bool HepMC::GenVertex::edge_iterator::operator== (const edge_iterator & a) const [inline]

equality

Definition at line 426 of file GenVertex.h.

8.15.3.6 bool HepMC::GenVertex::edge_iterator::operator!= (const edge_iterator & a) const [inline]

inequality

Definition at line 431 of file GenVertex.h.

8.15.3.7 bool HepMC::GenVertex::edge_iterator::is_parent () const

true if parent of root vtx

Definition at line 573 of file GenVertex.cc.

Referenced by HepMC::GenVertex::particle_iterator::advance_to_first_(), and HepMC::GenVertex::vertex_iterator::follow_edge_().

8.15.3.8 bool HepMC::GenVertex::edge_iterator::is_child () const

true if child of root vtx

Definition at line 578 of file GenVertex.cc.

8.15.3.9 const GenVertex * HepMC::GenVertex::edge_iterator::vertex_root () const [inline]

root vertex of this iteration

Definition at line 436 of file GenVertex.h.

The documentation for this class was generated from the following files:

- `GenVertex.h`
- `GenVertex.cc`

8.16 HepMC::GenVertex::particle_iterator Class Reference

particle iterator

```
#include <GenVertex.h>
```

Public Member Functions

- **particle_iterator** ()
- **particle_iterator** (**GenVertex** &vertex_root, **IteratorRange** range)
used to set limits on the iteration
- **particle_iterator** (const **particle_iterator** &)
copy
- virtual **~particle_iterator** ()
- **particle_iterator** & **operator=** (const **particle_iterator** &)
make a copy
- **GenParticle** * **operator *** (void) const
return a pointer to a particle
- **particle_iterator** & **operator++** (void)
Pre-fix increment.
- **particle_iterator** **operator++** (int)
Post-fix increment.
- bool **operator==** (const **particle_iterator** &) const
equality
- bool **operator!=** (const **particle_iterator** &) const
inequality

Protected Member Functions

- **GenParticle** * **advance_to_first_** ()
"first" particle

8.16.1 Detailed Description

particle iterator

Iterates over all particles connected via a graph. by iterating through all vertices in the m_range. For each vertex it returns orphaned parent particles (i.e. parents without production vertices) then children ... in this way each particle is associated to exactly one vertex and so it is returned exactly once. Is made friend so that it can access protected edge iterator

Examples:

example_UsingIterators.cc.

Definition at line 306 of file GenVertex.h.

8.16.2 Constructor & Destructor Documentation**8.16.2.1 HepMC::GenVertex::particle_iterator::particle_iterator ()**

Definition at line 826 of file GenVertex.cc.

8.16.2.2 HepMC::GenVertex::particle_iterator::particle_iterator (GenVertex & vertex_root, IteratorRange range)

used to set limits on the iteration

Definition at line 828 of file GenVertex.cc.

References `advance_to_first_()`, `HepMC::family`, and `HepMC::GenVertex::vertex_iterator::range()`.

8.16.2.3 HepMC::GenVertex::particle_iterator::particle_iterator (const particle_iterator &)

copy

Definition at line 842 of file GenVertex.cc.

8.16.2.4 HepMC::GenVertex::particle_iterator::~~particle_iterator () [virtual]

Definition at line 847 of file GenVertex.cc.

8.16.3 Member Function Documentation**8.16.3.1 GenVertex::particle_iterator & HepMC::GenVertex::particle_iterator::operator= (const particle_iterator &)**

make a copy

Definition at line 850 of file GenVertex.cc.

References `m_edge`, and `m_vertex_iterator`.

8.16.3.2 GenParticle * HepMC::GenVertex::particle_iterator::operator * (void) const

return a pointer to a particle

Definition at line 857 of file GenVertex.cc.

8.16.3.3 GenVertex::particle_iterator & HepMC::GenVertex::particle_iterator::operator++ (void)

Pre-fix increment.

Definition at line 862 of file GenVertex.cc.

References `advance_to_first_()`, and `HepMC::GenVertex::vertex_iterator::range()`.

8.16.3.4 GenVertex::particle_iterator HepMC::GenVertex::particle_iterator::operator++ (int)

Post-fix increment.

Definition at line 881 of file GenVertex.cc.

8.16.3.5 bool HepMC::GenVertex::particle_iterator::operator== (const particle_iterator &) const [inline]

equality

Definition at line 479 of file GenVertex.h.

8.16.3.6 bool HepMC::GenVertex::particle_iterator::operator!= (const particle_iterator &) const [inline]

inequality

Definition at line 484 of file GenVertex.h.

8.16.3.7 GenParticle * HepMC::GenVertex::particle_iterator::advance_to_first_() [protected]

"first" particle

if the current edge is not a suitable return value (because it is a parent of the vertex root that itself belongs to a different vertex) it advances to the first suitable return value

Definition at line 888 of file GenVertex.cc.

References `HepMC::GenVertex::edge_iterator::is_parent()`, `HepMC::GenVertex::vertex_iterator::range()`, and `HepMC::relatives`.

Referenced by `operator++()`, and `particle_iterator()`.

The documentation for this class was generated from the following files:

- **GenVertex.h**
- **GenVertex.cc**

8.17 HepMC::GenVertex::vertex_iterator Class Reference

vertex iterator

```
#include <GenVertex.h>
```

Public Member Functions

- **vertex_iterator** ()
- **vertex_iterator** (**GenVertex** &vtx_root, **IteratorRange** range)
used to set limits on the iteration
- **vertex_iterator** (**GenVertex** &vtx_root, **IteratorRange** range, std::set< const **HepMC::GenVertex** * > &visited_vertices)
next constructor is intended for internal use only
- **vertex_iterator** (const **vertex_iterator** &v_iter)
copy
- virtual ~**vertex_iterator** ()
- **vertex_iterator** & **operator**= (const **vertex_iterator** &)
make a copy
- **GenVertex** * **operator** * (void) const
return a pointer to a vertex
- **vertex_iterator** & **operator**++ (void)
Pre-fix increment.
- **vertex_iterator** **operator**++ (int)
Post-fix increment.
- bool **operator**== (const **vertex_iterator** &) const
equality
- bool **operator**!= (const **vertex_iterator** &) const
inequality
- **GenVertex** * **vertex_root** () const
vertex that this iterator begins from
- **IteratorRange** **range** () const
iterator range
- void **copy_with_own_set** (const **vertex_iterator** &v_iter, std::set< const **HepMC::GenVertex** * > &visited_vertices)
intended for internal use only.

Protected Member Functions

- **GenVertex * follow_edge_ ()**
non-null if recursive iter. created
- **void copy_recursive_iterator_ (const vertex_iterator *recursive_v_iter)**
copy recursive iterator

8.17.1 Detailed Description

vertex iterator

Iterates over all vertices connected via a graph to this vertex. this is made friend to that it can access protected edge iterator the range can be IteratorRange= (parents, children, family, ancestors, descendants, relatives) example for range=descendants the iterator will return all vertices which are children (connected by an outgoing particle edge), grandchildren, great-grandchildren, etc. of this vertex In all cases the iterator always returns this vertex (returned last). The algorithm is accomplished by converting the graph to a tree (by "chopping" the edges connecting to an already visited vertex) and returning the vertices in POST ORDER traversal.

Definition at line 236 of file GenVertex.h.

8.17.2 Constructor & Destructor Documentation

8.17.2.1 HepMC::GenVertex::vertex_iterator::vertex_iterator ()

Definition at line 595 of file GenVertex.cc.

Referenced by copy_recursive_iterator_(), and follow_edge_().

8.17.2.2 HepMC::GenVertex::vertex_iterator::vertex_iterator (GenVertex & vtx_root, IteratorRange range)

used to set limits on the iteration

Definition at line 600 of file GenVertex.cc.

References HepMC::GenVertex::edges_begin(), HepMC::GenVertex::edges_end(), and follow_edge_().

8.17.2.3 HepMC::GenVertex::vertex_iterator::vertex_iterator (GenVertex & vtx_root, IteratorRange range, std::set< const HepMC::GenVertex * > & visited_vertices)

next constructor is intended for internal use only

Definition at line 616 of file GenVertex.cc.

References HepMC::GenVertex::edges_begin(), HepMC::GenVertex::edges_end(), and follow_edge_().

8.17.2.4 **HepMC::GenVertex::vertex_iterator::vertex_iterator (const vertex_iterator & *v_iter*)**

copy

Definition at line 633 of file GenVertex.cc.

8.17.2.5 **HepMC::GenVertex::vertex_iterator::~~vertex_iterator ()** [virtual]

Definition at line 640 of file GenVertex.cc.

8.17.3 Member Function Documentation

8.17.3.1 **GenVertex::vertex_iterator & HepMC::GenVertex::vertex_iterator::operator= (const vertex_iterator &)**

make a copy

Definition at line 645 of file GenVertex.cc.

References `copy_recursive_iterator_()`, `m_edge`, `m_it_owns_set`, `m_range`, `m_recursive_iterator`, `m_vertex`, and `m_visited_vertices`.

8.17.3.2 **GenVertex * HepMC::GenVertex::vertex_iterator::operator * (void) const**

return a pointer to a vertex

Definition at line 682 of file GenVertex.cc.

8.17.3.3 **GenVertex::vertex_iterator & HepMC::GenVertex::vertex_iterator::operator++ (void)**

Pre-fix increment.

Definition at line 697 of file GenVertex.cc.

References `HepMC::GenVertex::edges_end()`, and `follow_edge_()`.

8.17.3.4 **GenVertex::vertex_iterator HepMC::GenVertex::vertex_iterator::operator++ (int)**

Post-fix increment.

Definition at line 739 of file GenVertex.cc.

8.17.3.5 **bool HepMC::GenVertex::vertex_iterator::operator== (const vertex_iterator &) const** [inline]

equality

Definition at line 450 of file GenVertex.h.

8.17.3.6 `bool HepMC::GenVertex::vertex_iterator::operator!= (const vertex_iterator &) const [inline]`

inequality

Definition at line 455 of file GenVertex.h.

8.17.3.7 `GenVertex * HepMC::GenVertex::vertex_iterator::vertex_root () const [inline]`

vertex that this iterator begins from

Definition at line 460 of file GenVertex.h.

8.17.3.8 `IteratorRange HepMC::GenVertex::vertex_iterator::range () const [inline]`

iterator range

Definition at line 464 of file GenVertex.h.

Referenced by `HepMC::GenVertex::particle_iterator::advance_to_first_()`, `HepMC::GenVertex::particle_iterator::operator++()`, and `HepMC::GenVertex::particle_iterator::particle_iterator()`.

8.17.3.9 `void HepMC::GenVertex::vertex_iterator::copy_with_own_set (const vertex_iterator & v_iter, std::set< const HepMC::GenVertex * > & visited_vertices)`

intended for internal use only.

intended for internal use only. (use with care!) this is the same as the operator= method, but it allows the user to specify which set container m_visited_vertices points to. in all cases, this vertex will NOT own its set.

Definition at line 746 of file GenVertex.cc.

References `copy_recursive_iterator_()`, `m_edge`, `m_range`, `m_recursive_iterator`, and `m_vertex`.

8.17.3.10 `GenVertex * HepMC::GenVertex::vertex_iterator::follow_edge_ () [protected]`

non-null if recursive iter. created

Definition at line 769 of file GenVertex.cc.

References `HepMC::family`, `HepMC::GenVertex::edge_iterator::is_parent()`, and `vertex_iterator()`.

Referenced by `operator++()`, and `vertex_iterator()`.

8.17.3.11 `void HepMC::GenVertex::vertex_iterator::copy_recursive_iterator_ (const vertex_iterator * recursive_v_iter) [protected]`

copy recursive iterator

Definition at line 805 of file GenVertex.cc.

References `copy_recursive_iterator_()`, `m_edge`, `m_it_owns_set`, `m_range`, `m_recursive_iterator`, `m_vertex`, `m_visited_vertices`, and `vertex_iterator()`.

Referenced by `copy_recursive_iterator_()`, `copy_with_own_set()`, and `operator=()`.

The documentation for this class was generated from the following files:

- **GenVertex.h**
- **GenVertex.cc**

8.18 HepMC::HeavyIon Class Reference

The **HeavyIon** (p.121) class stores information about heavy ions.

```
#include <HeavyIon.h>
```

Public Member Functions

- **HeavyIon** ()
default constructor
- **HeavyIon** (int nh, int np, int nt, int nc, int ns, int nsp, int nnw=0, int nwn=0, int nwnw=0, float im=0., float pl=0., float ec=0., float s=0.)
The first 6 values must be provided.
- **~HeavyIon** ()
- **HeavyIon** (**HeavyIon** const &orig)
copy constructor
- **HeavyIon** & **operator=** (**HeavyIon** const &rhs)
make a copy
- void **swap** (**HeavyIon** &other)
*swap two **HeavyIon** (p.121) objects*
- bool **operator==** (const **HeavyIon** &) const
check for equality
- bool **operator!=** (const **HeavyIon** &) const
check for inequality
- int **Ncoll_hard** () const
Number of hard scatterings.
- int **Npart_proj** () const
Number of projectile participants.
- int **Npart_targ** () const
Number of target participants.
- int **Ncoll** () const
Number of NN (nucleon-nucleon) collisions.
- int **spectator_neutrons** () const
Number of spectator neutrons.
- int **spectator_protons** () const
Number of spectator protons.
- int **N_Nwounded_collisions** () const

Number of N-Nwounded collisions.

- **int Nwounded_N_collisions () const**
Number of Nwounded-N collisons.
- **int Nwounded_Nwounded_collisions () const**
Number of Nwounded-Nwounded collisions.
- **float impact_parameter () const**
Impact Parameter(in fm) of collision.
- **float event_plane_angle () const**
Azimuthal angle of event plane.
- **float eccentricity () const**
- **float sigma_inel_NN () const**
nucleon-nucleon inelastic (including diffractive) cross-section
- **void set_Ncoll_hard (const int &i)**
set number of hard scatterings
- **void set_Npart_proj (const int &i)**
set number of projectile participants
- **void set_Npart_targ (const int &i)**
set number of target participants
- **void set_Ncoll (const int &i)**
set number of NN (nucleon-nucleon) collisions
- **void set_spectator_neutrons (const int &i)**
set number of spectator neutrons
- **void set_spectator_protons (const int &i)**
set number of spectator protons
- **void set_N_Nwounded_collisions (const int &i)**
set number of N-Nwounded collisions
- **void set_Nwounded_N_collisions (const int &i)**
set number of Nwounded-N collisons
- **void set_Nwounded_Nwounded_collisions (const int &i)**
set number of Nwounded-Nwounded collisions
- **void set_impact_parameter (const float &f)**
set Impact Parameter in fm
- **void set_event_plane_angle (const float &f)**
set azimuthal angle of event plane

- void **set_eccentricity** (const float &f)
set eccentricity of participating nucleons in the transverse plane
- void **set_sigma_inel_NN** (const float &f)
set nucleon-nucleon inelastic cross-section

8.18.1 Detailed Description

The **HeavyIon** (p. 121) class stores information about heavy ions.

HepMC::HeavyIon (p. 121) provides additional information storage for Heavy Ion generators in **GenEvent** (p. 54). Creation and use of this information is optional.

Definition at line 45 of file HeavyIon.h.

8.18.2 Constructor & Destructor Documentation

8.18.2.1 HepMC::HeavyIon::HeavyIon () [inline]

default constructor

Definition at line 51 of file HeavyIon.h.

8.18.2.2 HepMC::HeavyIon::HeavyIon (int *nh*, int *np*, int *nt*, int *nc*, int *ns*, int *nsp*, int *nnw* = 0, int *nwn* = 0, int *nwnw* = 0, float *im* = 0., float *pl* = 0., float *ec* = 0., float *s* = 0.) [inline]

The first 6 values must be provided.

Required members are the number of hard scatterings, the number of projectile participants. the number of target participants. the number of nucleon-nucleon collisions, the number of spectator neutrons, and the number of spectator protons.

Definition at line 168 of file HeavyIon.h.

8.18.2.3 HepMC::HeavyIon::~~HeavyIon () [inline]

Definition at line 72 of file HeavyIon.h.

8.18.2.4 HepMC::HeavyIon::HeavyIon (HeavyIon const & *orig*) [inline]

copy constructor

Definition at line 186 of file HeavyIon.h.

8.18.3 Member Function Documentation

8.18.3.1 **HeavyIon & HepMC::HeavyIon::operator= (HeavyIon const & *rhs*)** [inline]

make a copy

Definition at line 202 of file HeavyIon.h.

References swap().

8.18.3.2 **void HepMC::HeavyIon::swap (HeavyIon & *other*)** [inline]

swap two **HeavyIon** (p. 121) objects

Definition at line 209 of file HeavyIon.h.

References m_eccentricity, m_event_plane_angle, m_impact_parameter, m_N_Nwounded_collisions, m_Ncoll, m_Ncoll_hard, m_Npart_proj, m_Npart_targ, m_Nwounded_N_collisions, m_Nwounded_Nwounded_collisions, m_sigma_inel_NN, m_spectator_neutrons, and m_spectator_protons.

Referenced by operator=().

8.18.3.3 **bool HepMC::HeavyIon::operator== (const HeavyIon &) const** [inline]

check for equality

equality requires that each member match

Definition at line 226 of file HeavyIon.h.

References eccentricity(), event_plane_angle(), impact_parameter(), N_Nwounded_collisions(), Ncoll(), Ncoll_hard(), Npart_proj(), Npart_targ(), Nwounded_N_collisions(), Nwounded_Nwounded_collisions(), sigma_inel_NN(), spectator_neutrons(), and spectator_protons().

8.18.3.4 **bool HepMC::HeavyIon::operator!= (const HeavyIon &) const** [inline]

check for inequality

any nonmatching member generates inequality

Definition at line 244 of file HeavyIon.h.

8.18.3.5 **int HepMC::HeavyIon::Ncoll_hard () const** [inline]

Number of hard scatterings.

Definition at line 87 of file HeavyIon.h.

Referenced by operator==(), HepMC::IO_GenEvent::write_heavy_ion(), and HepMC::IO_ExtendedAscii::write_heavy_ion().

8.18.3.6 **int HepMC::HeavyIon::Npart_proj () const** [inline]

Number of projectile participants.

Definition at line 89 of file HeavyIon.h.

Referenced by `operator==()`, `HepMC::IO_GenEvent::write_heavy_ion()`, and `HepMC::IO_ExtendedAscii::write_heavy_ion()`.

8.18.3.7 `int HepMC::HeavyIon::Npart_targ () const [inline]`

Number of target participants.

Definition at line 91 of file HeavyIon.h.

Referenced by `operator==()`, `HepMC::IO_GenEvent::write_heavy_ion()`, and `HepMC::IO_ExtendedAscii::write_heavy_ion()`.

8.18.3.8 `int HepMC::HeavyIon::Ncoll () const [inline]`

Number of NN (nucleon-nucleon) collisions.

Definition at line 93 of file HeavyIon.h.

Referenced by `operator==()`, `HepMC::IO_GenEvent::write_heavy_ion()`, and `HepMC::IO_ExtendedAscii::write_heavy_ion()`.

8.18.3.9 `int HepMC::HeavyIon::spectator_neutrons () const [inline]`

Number of spectator neutrons.

Definition at line 95 of file HeavyIon.h.

Referenced by `operator==()`, `HepMC::IO_GenEvent::write_heavy_ion()`, and `HepMC::IO_ExtendedAscii::write_heavy_ion()`.

8.18.3.10 `int HepMC::HeavyIon::spectator_protons () const [inline]`

Number of spectator protons.

Definition at line 97 of file HeavyIon.h.

Referenced by `operator==()`, `HepMC::IO_GenEvent::write_heavy_ion()`, and `HepMC::IO_ExtendedAscii::write_heavy_ion()`.

8.18.3.11 `int HepMC::HeavyIon::N_Nwounded_collisions () const [inline]`

Number of N-Nwounded collisions.

Definition at line 99 of file HeavyIon.h.

Referenced by `operator==()`, `HepMC::IO_GenEvent::write_heavy_ion()`, and `HepMC::IO_ExtendedAscii::write_heavy_ion()`.

8.18.3.12 `int HepMC::HeavyIon::Nwounded_N_collisions () const [inline]`

Number of Nwounded-N collisons.

Definition at line 101 of file HeavyIon.h.

Referenced by `operator==()`, `HepMC::IO_GenEvent::write_heavy_ion()`, and `HepMC::IO_ExtendedAscii::write_heavy_ion()`.

8.18.3.13 `int HepMC::HeavyIon::Nwounded_Nwounded_collisions () const` `[inline]`

Number of Nwounded-Nwounded collisions.

Definition at line 103 of file `HeavyIon.h`.

Referenced by `operator==()`, `HepMC::IO_GenEvent::write_heavy_ion()`, and `HepMC::IO_ExtendedAscii::write_heavy_ion()`.

8.18.3.14 `float HepMC::HeavyIon::impact_parameter () const` `[inline]`

Impact Parameter(in fm) of collision.

Definition at line 105 of file `HeavyIon.h`.

Referenced by `operator==()`, `HepMC::IO_GenEvent::write_heavy_ion()`, and `HepMC::IO_ExtendedAscii::write_heavy_ion()`.

8.18.3.15 `float HepMC::HeavyIon::event_plane_angle () const` `[inline]`

Azimuthal angle of event plane.

Definition at line 107 of file `HeavyIon.h`.

Referenced by `operator==()`, `HepMC::IO_GenEvent::write_heavy_ion()`, and `HepMC::IO_ExtendedAscii::write_heavy_ion()`.

8.18.3.16 `float HepMC::HeavyIon::eccentricity () const` `[inline]`

eccentricity of participating nucleons in the transverse plane (as in phobos nucl-ex/0510031)

Definition at line 110 of file `HeavyIon.h`.

Referenced by `operator==()`, `HepMC::IO_GenEvent::write_heavy_ion()`, and `HepMC::IO_ExtendedAscii::write_heavy_ion()`.

8.18.3.17 `float HepMC::HeavyIon::sigma_inel_NN () const` `[inline]`

nucleon-nucleon inelastic (including diffractive) cross-section

Definition at line 112 of file `HeavyIon.h`.

Referenced by `operator==()`, `HepMC::IO_GenEvent::write_heavy_ion()`, and `HepMC::IO_ExtendedAscii::write_heavy_ion()`.

8.18.3.18 `void HepMC::HeavyIon::set_Ncoll_hard (const int & i) [inline]`

set number of hard scatterings

Definition at line 116 of file `HeavyIon.h`.

8.18.3.19 void HepMC::HeavyIon::set_Npart_proj (const int & i) [inline]

set number of projectile participants

Definition at line 118 of file HeavyIon.h.

8.18.3.20 void HepMC::HeavyIon::set_Npart_targ (const int & i) [inline]

set number of target participants

Definition at line 120 of file HeavyIon.h.

8.18.3.21 void HepMC::HeavyIon::set_Ncoll (const int & i) [inline]

set number of NN (nucleon-nucleon) collisions

Definition at line 122 of file HeavyIon.h.

**8.18.3.22 void HepMC::HeavyIon::set_spectator_neutrons (const int & i)
[inline]**

set number of spectator neutrons

Definition at line 124 of file HeavyIon.h.

8.18.3.23 void HepMC::HeavyIon::set_spectator_protons (const int & i) [inline]

set number of spectator protons

Definition at line 126 of file HeavyIon.h.

**8.18.3.24 void HepMC::HeavyIon::set_N_Nwounded_collisions (const int & i)
[inline]**

set number of N-Nwounded collisions

Definition at line 128 of file HeavyIon.h.

**8.18.3.25 void HepMC::HeavyIon::set_Nwounded_N_collisions (const int & i)
[inline]**

set number of Nwounded-N collisions

Definition at line 130 of file HeavyIon.h.

**8.18.3.26 void HepMC::HeavyIon::set_Nwounded_Nwounded_collisions (const
int & i) [inline]**

set number of Nwounded-Nwounded collisions

Definition at line 132 of file HeavyIon.h.

8.18.3.27 `void HepMC::HeavyIon::set_impact_parameter (const float & f)`
[inline]

set Impact Parameter in fm

Definition at line 135 of file HeavyIon.h.

8.18.3.28 `void HepMC::HeavyIon::set_event_plane_angle (const float & f)`
[inline]

set azimuthal angle of event plane

Definition at line 137 of file HeavyIon.h.

8.18.3.29 `void HepMC::HeavyIon::set_eccentricity (const float & f)` [inline]

set eccentricity of participating nucleons in the transverse plane

Definition at line 139 of file HeavyIon.h.

8.18.3.30 `void HepMC::HeavyIon::set_sigma_inel_NN (const float & f)` [inline]

set nucleon-nucleon inelastic cross-section

Definition at line 141 of file HeavyIon.h.

The documentation for this class was generated from the following file:

- **HeavyIon.h**

8.19 HepMC::HEPEVT_Wrapper Class Reference

Generic Wrapper for the fortran HEPEVT common block.

```
#include <HEPEVT_Wrapper.h>
```

Static Public Member Functions

- static void **print_hepevt** (std::ostream &ostr=std::cout)
write information from HEPEVT common block
- static void **print_hepevt_particle** (int index, std::ostream &ostr=std::cout)
write particle information to ostr
- static bool **is_double_precision** ()
True if common block uses double.
- static bool **check_hepevt_consistency** (std::ostream &ostr=std::cout)
check for problems with HEPEVT common block
- static void **zero_everything** ()
set all entries in HEPEVT to zero
- static int **event_number** ()
event number
- static int **number_entries** ()
num entries in current evt
- static int **status** (int index)
status code
- static int **id** (int index)
PDG particle id.
- static int **first_parent** (int index)
index of 1st mother
- static int **last_parent** (int index)
index of last mother
- static int **number_parents** (int index)
number of parents
- static int **first_child** (int index)
index of 1st daughter
- static int **last_child** (int index)
index of last daughter

- static int **number_children** (int index)
number of children
- static double **px** (int index)
X momentum.
- static double **py** (int index)
Y momentum.
- static double **pz** (int index)
Z momentum.
- static double **e** (int index)
Energy.
- static double **m** (int index)
generated mass
- static double **x** (int index)
X Production vertex.
- static double **y** (int index)
Y Production vertex.
- static double **z** (int index)
Z Production vertex.
- static double **t** (int index)
production time
- static void **set_event_number** (int evtno)
set event number
- static void **set_number_entries** (int noentries)
set number of entries in HEPEVT
- static void **set_status** (int index, int status)
set particle status
- static void **set_id** (int index, int id)
set particle ID
- static void **set_parents** (int index, int firstparent, int lastparent)
define parents of a particle
- static void **set_children** (int index, int firstchild, int lastchild)
define children of a particle
- static void **set_momentum** (int index, double px, double py, double pz, double e)
set particle momentum

- static void **set__mass** (int index, double mass)
set particle mass
- static void **set__position** (int index, double x, double y, double z, double t)
set particle production vertex
- static unsigned int **sizeof__int** ()
size of integer in bytes
- static unsigned int **sizeof__real** ()
size of real in bytes
- static int **max__number__entries** ()
size of common block
- static void **set__sizeof__int** (unsigned int)
define size of integer
- static void **set__sizeof__real** (unsigned int)
define size of real
- static void **set__max__number__entries** (unsigned int)
define size of common block

Static Protected Member Functions

- static double **byte__num__to__double** (unsigned int)
navigate a byte array
- static int **byte__num__to__int** (unsigned int)
navigate a byte array
- static void **write__byte__num** (double, unsigned int)
pretend common block is an array of bytes
- static void **write__byte__num** (int, unsigned int)
pretend common block is an array of bytes
- static void **print__legend** (std::ostream &ostr=std::cout)
print output legend

8.19.1 Detailed Description

Generic Wrapper for the fortran HEPEVT common block.

This class is intended for static use only - it makes no sense to instantiate it.

Definition at line 130 of file HEPEVT_Wrapper.h.

8.19.2 Member Function Documentation

8.19.2.1 `void HepMC::HEPEVT_Wrapper::print_hepevt (std::ostream & ostr = std::cout) [static]`

write information from HEPEVT common block

dumps the content of this HEPEVT event to ostr (Width is 80)

Examples:

`example_MyHerwig.cc.`

Definition at line 27 of file HEPEVT_Wrapper.cc.

References `event_number()`, `is_double_precision()`, `max_number_entries()`, `number_entries()`, `print_hepevt_particle()`, `print_legend()`, `sizeof_int()`, and `sizeof_real()`.

Referenced by `main()`.

8.19.2.2 `void HepMC::HEPEVT_Wrapper::print_hepevt_particle (int index, std::ostream & ostr = std::cout) [static]`

write particle information to ostr

dumps the content HEPEVT particle entry *i* (Width is 120) here *i* is the C array index (i.e. it starts at 0 ... whereas the fortran array index starts at 1) So if there's 100 particles, the last valid index is 100-1=99

Definition at line 68 of file HEPEVT_Wrapper.cc.

References `e()`, `first_child()`, `first_parent()`, `last_child()`, `last_parent()`, `m()`, `px()`, `py()`, `pz()`, `status()`, `t()`, `x()`, `y()`, and `z()`.

Referenced by `check_hepevt_consistency()`, and `print_hepevt()`.

8.19.2.3 `bool HepMC::HEPEVT_Wrapper::is_double_precision () [inline, static]`

True if common block uses double.

Definition at line 337 of file HEPEVT_Wrapper.h.

References `sizeof_real()`.

Referenced by `print_hepevt()`.

8.19.2.4 `bool HepMC::HEPEVT_Wrapper::check_hepevt_consistency (std::ostream & ostr = std::cout) [static]`

check for problems with HEPEVT common block

This method inspects the HEPEVT common block and looks for inconsistencies in the mother/daughter pointers

Definition at line 88 of file HEPEVT_Wrapper.cc.

References `event_number()`, `first_child()`, `first_parent()`, `last_child()`, `last_parent()`, `m()`, `number_entries()`, `print_hepevt_particle()`, and `print_legend()`.

8.19.2.5 void HepMC::HEPEVT_Wrapper::zero_everything () [static]

set all entries in HEPEVT to zero

Definition at line 212 of file HEPEVT_Wrapper.cc.

References `max_number_entries()`, `set_children()`, `set_event_number()`, `set_id()`, `set_mass()`, `set_momentum()`, `set_number_entries()`, `set_parents()`, `set_position()`, and `set_status()`.

8.19.2.6 int HepMC::HEPEVT_Wrapper::event_number () [inline, static]

event number

Definition at line 343 of file HEPEVT_Wrapper.h.

References `byte_num_to_int()`.

Referenced by `HepMC::IO_HERWIG::build_end_vertex()`, `HepMC::IO_HEPEVT::build_end_vertex()`, `HepMC::IO_HERWIG::build_production_vertex()`, `HepMC::IO_HEPEVT::build_production_vertex()`, `check_hepevt_consistency()`, `HepMC::IO_HERWIG::fill_next_event()`, `HepMC::IO_HEPEVT::fill_next_event()`, and `print_hepevt()`.

8.19.2.7 int HepMC::HEPEVT_Wrapper::number_entries () [inline, static]

num entries in current evt

Definition at line 346 of file HEPEVT_Wrapper.h.

References `byte_num_to_int()`, `max_number_entries()`, and `sizeof_int()`.

Referenced by `check_hepevt_consistency()`, `HepMC::IO_HERWIG::fill_next_event()`, `HepMC::IO_HEPEVT::fill_next_event()`, `first_child()`, `first_parent()`, `last_child()`, `last_parent()`, `print_hepevt()`, `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, and `HepMC::IO_HERWIG::repair_hepevt()`.

8.19.2.8 int HepMC::HEPEVT_Wrapper::status (int *index*) [inline, static]

status code

Definition at line 353 of file HEPEVT_Wrapper.h.

References `byte_num_to_int()`, and `sizeof_int()`.

Referenced by `HepMC::IO_HERWIG::build_particle()`, `HepMC::IO_HEPEVT::build_particle()`, `HepMC::IO_HERWIG::fill_next_event()`, `print_hepevt_particle()`, `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, and `HepMC::IO_HERWIG::repair_hepevt()`.

8.19.2.9 int HepMC::HEPEVT_Wrapper::id (int *index*) [inline, static]

PDG particle id.

Definition at line 356 of file HEPEVT_Wrapper.h.

References `byte_num_to_int()`, `max_number_entries()`, and `sizeof_int()`.

Referenced by `HepMC::IO_HERWIG::build_particle()`, `HepMC::IO_HEPEVT::build_particle()`, `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, and `HepMC::IO_HERWIG::repair_hepevt()`.

8.19.2.10 `int HepMC::HEPEVT_Wrapper::first_parent (int index)` [inline, static]

index of 1st mother

Definition at line 362 of file HEPEVT_Wrapper.h.

References `byte_num_to_int()`, `max_number_entries()`, `number_entries()`, and `sizeof_int()`.

Referenced by `HepMC::IO_HERWIG::build_production_vertex()`, `HepMC::IO_HEPEVT::build_production_vertex()`, `check_hepevt_consistency()`, `HepMC::IO_HERWIG::fill_next_event()`, `last_parent()`, `number_parents()`, `print_hepevt_particle()`, `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, and `HepMC::IO_HERWIG::repair_hepevt()`.

8.19.2.11 `int HepMC::HEPEVT_Wrapper::last_parent (int index)` [inline, static]

index of last mother

Definition at line 370 of file HEPEVT_Wrapper.h.

References `byte_num_to_int()`, `first_parent()`, `max_number_entries()`, `number_entries()`, and `sizeof_int()`.

Referenced by `HepMC::IO_HERWIG::build_production_vertex()`, `HepMC::IO_HEPEVT::build_production_vertex()`, `check_hepevt_consistency()`, `number_parents()`, `print_hepevt_particle()`, `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, and `HepMC::IO_HERWIG::repair_hepevt()`.

8.19.2.12 `int HepMC::HEPEVT_Wrapper::number_parents (int index)` [inline, static]

number of parents

Definition at line 388 of file HEPEVT_Wrapper.h.

References `first_parent()`, and `last_parent()`.

Referenced by `HepMC::IO_HERWIG::build_production_vertex()`, and `HepMC::IO_HEPEVT::build_production_vertex()`.

8.19.2.13 `int HepMC::HEPEVT_Wrapper::first_child (int index)` [inline, static]

index of 1st daughter

Definition at line 394 of file HEPEVT_Wrapper.h.

References `byte_num_to_int()`, `max_number_entries()`, `number_entries()`, and `sizeof_int()`.

Referenced by `HepMC::IO_HERWIG::build_end_vertex()`, `HepMC::IO_HEPEVT::build_end_vertex()`, `check_hepevt_consistency()`, `last_child()`, `number_children()`, `print_hepevt_particle()`, `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, and `HepMC::IO_HERWIG::repair_hepevt()`.

8.19.2.14 `int HepMC::HEPEVT_Wrapper::last_child (int index)` [inline, static]

index of last daughter

Definition at line 402 of file HEPEVT_Wrapper.h.

References `byte_num_to_int()`, `first_child()`, `max_number_entries()`, `number_entries()`, and `sizeof_int()`.

Referenced by `HepMC::IO_HERWIG::build_end_vertex()`, `HepMC::IO_HEPEVT::build_end_vertex()`, `check_hepevt_consistency()`, `number_children()`, `print_hepevt_particle()`, `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, and `HepMC::IO_HERWIG::repair_hepevt()`.

8.19.2.15 `int HepMC::HEPEVT_Wrapper::number_children (int index)` [inline, static]

number of children

Definition at line 420 of file HEPEVT_Wrapper.h.

References `first_child()`, and `last_child()`.

Referenced by `HepMC::IO_HERWIG::build_end_vertex()`, and `HepMC::IO_HEPEVT::build_end_vertex()`.

8.19.2.16 `double HepMC::HEPEVT_Wrapper::px (int index)` [inline, static]

X momentum.

Definition at line 427 of file HEPEVT_Wrapper.h.

References `byte_num_to_double()`, `max_number_entries()`, `sizeof_int()`, and `sizeof_real()`.

Referenced by `HepMC::IO_HERWIG::build_particle()`, `HepMC::IO_HEPEVT::build_particle()`, `print_hepevt_particle()`, and `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`.

8.19.2.17 `double HepMC::HEPEVT_Wrapper::py (int index)` [inline, static]

Y momentum.

Definition at line 433 of file HEPEVT_Wrapper.h.

References `byte_num_to_double()`, `max_number_entries()`, `sizeof_int()`, and `sizeof_real()`.

Referenced by `HepMC::IO_HERWIG::build_particle()`, `HepMC::IO_HEPEVT::build_particle()`, `print_hepevt_particle()`, and `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`.

8.19.2.18 `double HepMC::HEPEVT_Wrapper::pz (int index)` [inline, static]

Z momentum.

Definition at line 440 of file HEPEVT_Wrapper.h.

References `byte_num_to_double()`, `max_number_entries()`, `sizeof_int()`, and `sizeof_real()`.

Referenced by `HepMC::IO_HERWIG::build_particle()`, `HepMC::IO_HEPEVT::build_particle()`, `print_hepevt_particle()`, and `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`.

8.19.2.19 `double HepMC::HEPEVT_Wrapper::e (int index)` [inline, static]

Energy.

Definition at line 446 of file `HEPEVT_Wrapper.h`.

References `byte_num_to_double()`, `max_number_entries()`, `sizeof_int()`, and `sizeof_real()`.

Referenced by `HepMC::IO_HERWIG::build_particle()`, `HepMC::IO_HEPEVT::build_particle()`, `print_hepevt_particle()`, and `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`.

8.19.2.20 `double HepMC::HEPEVT_Wrapper::m (int index)` [inline, static]

generated mass

Definition at line 452 of file `HEPEVT_Wrapper.h`.

References `byte_num_to_double()`, `max_number_entries()`, `sizeof_int()`, and `sizeof_real()`.

Referenced by `HepMC::IO_HERWIG::build_particle()`, `HepMC::IO_HEPEVT::build_particle()`, `check_hepevt_consistency()`, `print_hepevt_particle()`, and `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`.

8.19.2.21 `double HepMC::HEPEVT_Wrapper::x (int index)` [inline, static]

X Production vertex.

Definition at line 458 of file `HEPEVT_Wrapper.h`.

References `byte_num_to_double()`, `max_number_entries()`, `sizeof_int()`, and `sizeof_real()`.

Referenced by `HepMC::IO_HERWIG::build_end_vertex()`, `HepMC::IO_HEPEVT::build_end_vertex()`, `HepMC::IO_HERWIG::build_production_vertex()`, `HepMC::IO_HEPEVT::build_production_vertex()`, `print_hepevt_particle()`, and `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`.

8.19.2.22 `double HepMC::HEPEVT_Wrapper::y (int index)` [inline, static]

Y Production vertex.

Definition at line 465 of file `HEPEVT_Wrapper.h`.

References `byte_num_to_double()`, `max_number_entries()`, `sizeof_int()`, and `sizeof_real()`.

Referenced by `HepMC::IO_HERWIG::build_end_vertex()`, `HepMC::IO_HEPEVT::build_end_vertex()`, `HepMC::IO_HERWIG::build_production_vertex()`, `HepMC::IO_HEPEVT::build_production_vertex()`, `print_hepevt_particle()`, and `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`.

8.19.2.23 `double HepMC::HEPEVT_Wrapper::z (int index)` [inline, static]

Z Production vertex.

Definition at line 472 of file HEPEVT_Wrapper.h.

References `byte_num_to_double()`, `max_number_entries()`, `sizeof_int()`, and `sizeof_real()`.

Referenced by `HepMC::IO_HERWIG::build_end_vertex()`, `HepMC::IO_HEPEVT::build_end_vertex()`, `HepMC::IO_HERWIG::build_production_vertex()`, `HepMC::IO_HEPEVT::build_production_vertex()`, `print_hepevt_particle()`, and `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`.

8.19.2.24 `double HepMC::HEPEVT_Wrapper::t (int index)` [inline, static]

production time

Definition at line 479 of file HEPEVT_Wrapper.h.

References `byte_num_to_double()`, `max_number_entries()`, `sizeof_int()`, and `sizeof_real()`.

Referenced by `HepMC::IO_HERWIG::build_end_vertex()`, `HepMC::IO_HEPEVT::build_end_vertex()`, `HepMC::IO_HERWIG::build_production_vertex()`, `HepMC::IO_HEPEVT::build_production_vertex()`, `print_hepevt_particle()`, and `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`.

8.19.2.25 `void HepMC::HEPEVT_Wrapper::set_event_number (int evtno)` [inline, static]

set event number

Definition at line 486 of file HEPEVT_Wrapper.h.

References `write_byte_num()`.

Referenced by `HepMC::IO_HEPEVT::write_event()`, and `zero_everything()`.

8.19.2.26 `void HepMC::HEPEVT_Wrapper::set_number_entries (int noentries)` [inline, static]

set number of entries in HEPEVT

Definition at line 489 of file HEPEVT_Wrapper.h.

References `sizeof_int()`, and `write_byte_num()`.

Referenced by `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, `HepMC::IO_HEPEVT::write_event()`, and `zero_everything()`.

8.19.2.27 `void HepMC::HEPEVT_Wrapper::set_status (int index, int status)` [inline, static]

set particle status

Definition at line 492 of file HEPEVT_Wrapper.h.

References `max_number_entries()`, `sizeof_int()`, and `write_byte_num()`.

Referenced by `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, `HepMC::IO_HEPEVT::write_event()`, `zero_everything()`, and `HepMC::IO_HERWIG::zero_hepevt_entry()`.

8.19.2.28 `void HepMC::HEPEVT_Wrapper::set_id (int index, int id)` [inline, static]

set particle ID

Definition at line 498 of file HEPEVT_Wrapper.h.

References `max_number_entries()`, `sizeof_int()`, and `write_byte_num()`.

Referenced by `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, `HepMC::IO_HERWIG::repair_hepevt()`, `HepMC::IO_HEPEVT::write_event()`, `zero_everything()`, and `HepMC::IO_HERWIG::zero_hepevt_entry()`.

8.19.2.29 `void HepMC::HEPEVT_Wrapper::set_parents (int index, int firstparent, int lastparent)` [inline, static]

define parents of a particle

Definition at line 504 of file HEPEVT_Wrapper.h.

References `max_number_entries()`, `sizeof_int()`, and `write_byte_num()`.

Referenced by `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, `HepMC::IO_HERWIG::repair_hepevt()`, `HepMC::IO_HEPEVT::write_event()`, `zero_everything()`, and `HepMC::IO_HERWIG::zero_hepevt_entry()`.

8.19.2.30 `void HepMC::HEPEVT_Wrapper::set_children (int index, int firstchild, int lastchild)` [inline, static]

define children of a particle

Definition at line 514 of file HEPEVT_Wrapper.h.

References `max_number_entries()`, `sizeof_int()`, and `write_byte_num()`.

Referenced by `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, `HepMC::IO_HERWIG::repair_hepevt()`, `HepMC::IO_HEPEVT::write_event()`, `zero_everything()`, and `HepMC::IO_HERWIG::zero_hepevt_entry()`.

8.19.2.31 `void HepMC::HEPEVT_Wrapper::set_momentum (int index, double px, double py, double pz, double e)` [inline, static]

set particle momentum

Definition at line 524 of file HEPEVT_Wrapper.h.

References `max_number_entries()`, `sizeof_int()`, `sizeof_real()`, and `write_byte_num()`.

Referenced by `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, `HepMC::IO_HEPEVT::write_event()`, `zero_everything()`, and `HepMC::IO_HERWIG::zero_hepevt_entry()`.

8.19.2.32 `void HepMC::HEPEVT_Wrapper::set_mass (int index, double mass)` [inline, static]

set particle mass

Definition at line 538 of file HEPEVT_Wrapper.h.

References `max_number_entries()`, `sizeof_int()`, `sizeof_real()`, and `write_byte_num()`.

Referenced by `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, `HepMC::IO_HEPEVT::write_event()`, `zero_everything()`, and `HepMC::IO_HERWIG::zero_hepevt_entry()`.

8.19.2.33 `void HepMC::HEPEVT_Wrapper::set_position (int index, double x, double y, double z, double t)` [inline, static]

set particle production vertex

Definition at line 545 of file `HEPEVT_Wrapper.h`.

References `max_number_entries()`, `sizeof_int()`, `sizeof_real()`, and `write_byte_num()`.

Referenced by `HepMC::IO_HERWIG::remove_gaps_in_hepevt()`, `HepMC::IO_HEPEVT::write_event()`, `zero_everything()`, and `HepMC::IO_HERWIG::zero_hepevt_entry()`.

8.19.2.34 `unsigned int HepMC::HEPEVT_Wrapper::sizeof_int ()` [inline, static]

size of integer in bytes

Definition at line 225 of file `HEPEVT_Wrapper.h`.

Referenced by `e()`, `first_child()`, `first_parent()`, `id()`, `last_child()`, `last_parent()`, `m()`, `number_entries()`, `print_hepevt()`, `px()`, `py()`, `pz()`, `set_children()`, `set_id()`, `set_mass()`, `set_momentum()`, `set_number_entries()`, `set_parents()`, `set_position()`, `set_status()`, `status()`, `t()`, `x()`, `y()`, and `z()`.

8.19.2.35 `unsigned int HepMC::HEPEVT_Wrapper::sizeof_real ()` [inline, static]

size of real in bytes

Definition at line 227 of file `HEPEVT_Wrapper.h`.

Referenced by `e()`, `is_double_precision()`, `m()`, `print_hepevt()`, `px()`, `py()`, `pz()`, `set_mass()`, `set_momentum()`, `set_position()`, `t()`, `x()`, `y()`, and `z()`.

8.19.2.36 `int HepMC::HEPEVT_Wrapper::max_number_entries ()` [inline, static]

size of common block

Definition at line 229 of file `HEPEVT_Wrapper.h`.

Referenced by `e()`, `first_child()`, `first_parent()`, `id()`, `last_child()`, `last_parent()`, `m()`, `number_entries()`, `print_hepevt()`, `px()`, `py()`, `pz()`, `set_children()`, `set_id()`, `set_mass()`, `set_momentum()`, `set_parents()`, `set_position()`, `set_status()`, `t()`, `HepMC::IO_HEPEVT::write_event()`, `x()`, `y()`, `z()`, `zero_everything()`, and `HepMC::IO_HERWIG::zero_hepevt_entry()`.

8.19.2.37 `void HepMC::HEPEVT_Wrapper::set_sizeof_int (unsigned int)`
`[inline, static]`

define size of integer

Definition at line 232 of file HEPEVT_Wrapper.h.

8.19.2.38 `void HepMC::HEPEVT_Wrapper::set_sizeof_real (unsigned int)`
`[inline, static]`

define size of real

Examples:

`example_MyHerwig.cc`, `example_MyPythia.cc`, `example_MyPythiaOnlyToHepMC.cc`, `example_MyPythiaRead.cc`, `example_MyPythiaWithEventSelection.cc`, `example_PythiaParticle.cc`, `testHerwigCopies.cc`, and `testPythiaCopies.cc`.

Definition at line 242 of file HEPEVT_Wrapper.h.

Referenced by `main()`.

8.19.2.39 `void HepMC::HEPEVT_Wrapper::set_max_number_entries (unsigned int)` `[inline, static]`

define size of common block

Examples:

`example_MyHerwig.cc`, `example_MyPythia.cc`, `example_MyPythiaOnlyToHepMC.cc`, `example_MyPythiaRead.cc`, `example_MyPythiaWithEventSelection.cc`, `example_PythiaParticle.cc`, `testHerwigCopies.cc`, and `testPythiaCopies.cc`.

Definition at line 251 of file HEPEVT_Wrapper.h.

Referenced by `main()`.

8.19.2.40 `double HepMC::HEPEVT_Wrapper::byte_num_to_double (unsigned int)` `[inline, static, protected]`

navigate a byte array

Definition at line 255 of file HEPEVT_Wrapper.h.

References `hepevt`, and `hepevt_bytes_allocation`.

Referenced by `e()`, `m()`, `px()`, `py()`, `pz()`, `t()`, `x()`, `y()`, and `z()`.

8.19.2.41 `int HepMC::HEPEVT_Wrapper::byte_num_to_int (unsigned int)`
`[inline, static, protected]`

navigate a byte array

Definition at line 273 of file HEPEVT_Wrapper.h.

References `hepevt`, and `hepevt_bytes_allocation`.

Referenced by `event_number()`, `first_child()`, `first_parent()`, `id()`, `last_child()`, `last_parent()`, `number_entries()`, and `status()`.

8.19.2.42 `void HepMC::HEPEVT_Wrapper::write_byte_num (double, unsigned int)` [`inline`, `static`, `protected`]

pretend common block is an array of bytes

Definition at line 295 of file `HEPEVT_Wrapper.h`.

References `hepevt`, and `hepevt_bytes_allocation`.

Referenced by `set_children()`, `set_event_number()`, `set_id()`, `set_mass()`, `set_momentum()`, `set_number_entries()`, `set_parents()`, `set_position()`, and `set_status()`.

8.19.2.43 `void HepMC::HEPEVT_Wrapper::write_byte_num (int, unsigned int)` [`inline`, `static`, `protected`]

pretend common block is an array of bytes

Definition at line 312 of file `HEPEVT_Wrapper.h`.

References `hepevt`, and `hepevt_bytes_allocation`.

8.19.2.44 `void HepMC::HEPEVT_Wrapper::print_legend (std::ostream & ostr = std::cout)` [`static`, `protected`]

print output legend

Definition at line 55 of file `HEPEVT_Wrapper.cc`.

Referenced by `check_hepevt_consistency()`, and `print_hepevt()`.

The documentation for this class was generated from the following files:

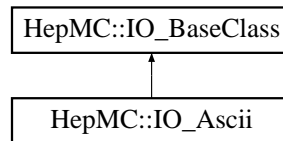
- `HEPEVT_Wrapper.h`
- `HEPEVT_Wrapper.cc`

8.20 HepMC::IO_Ascii Class Reference

IO_Ascii (p. 142) is used to read or write from an ascii file.

```
#include <IO_Ascii.h>
```

Inheritance diagram for HepMC::IO_Ascii::



Public Member Functions

- **IO_Ascii** (const char *filename="IO_Ascii.dat", std::ios::openmode mode=std::ios::out)
constructor requiring a file name and std::ios mode
- virtual **~IO_Ascii** ()
- void **write_event** (const **GenEvent** *evt)
write this event
- bool **fill_next_event** (**GenEvent** *evt)
get the next event
- void **write_particle_data_table** (const **ParticleDataTable** *)
*write this **ParticleDataTable** (p. 213)*
- bool **fill_particle_data_table** (**ParticleDataTable** *)
*fill this **ParticleDataTable** (p. 213)*
- void **write_comment** (const std::string comment)
- int **rdstate** () const
check the state of the IO stream
- void **clear** ()
clear the IO stream
- void **print** (std::ostream &ostr=std::cout) const
write to ostr

Protected Member Functions

- void **write_vertex** (**GenVertex** *)
write vertex information
- void **write_particle** (**GenParticle** *p)
write particle information

- void **write_particle_data** (const **ParticleData** *d)
write ParticleDataTable (p. 213) information
- bool **write_end_listing** ()
write end tag
- void **output** (const double &)
write double
- void **output** (const int &)
write int
- void **output** (const long int &)
write long int
- void **output** (const char &)
write a single character

8.20.1 Detailed Description

IO_Ascii (p. 142) is used to read or write from an ascii file.

Strategy for reading or writing events/particleData as machine readable ascii to a file. When instantiating, the mode of file to be created must be specified.

Definition at line 64 of file **IO_Ascii.h**.

8.20.2 Constructor & Destructor Documentation

8.20.2.1 HepMC::IO_Ascii::IO_Ascii (const char * *filename* = "IO_Ascii.dat",
std::ios::openmode *mode* = std::ios::out)

constructor requiring a file name and std::ios mode

Definition at line 16 of file **IO_Ascii.cc**.

8.20.2.2 HepMC::IO_Ascii::~~IO_Ascii () [virtual]

Definition at line 39 of file **IO_Ascii.cc**.

References **write_end_listing**() .

8.20.3 Member Function Documentation

8.20.3.1 void HepMC::IO_Ascii::write_event (const **GenEvent** * *evt*) [virtual]

write this event

Writes evt to m_file. It does NOT delete the event after writing.

Implements **HepMC::IO_BaseClass** (p. 152).

Definition at line 54 of file IO_Ascii.cc.

References HepMC::GenEvent::alphaQCD(), HepMC::GenEvent::alphaQED(), HepMC::GenVertex::barcode(), HepMC::WeightContainer::begin(), HepMC::WeightContainer::end(), HepMC::GenEvent::event_number(), HepMC::GenEvent::event_scale(), output(), HepMC::GenEvent::random_states(), HepMC::GenEvent::signal_process_id(), HepMC::GenEvent::signal_process_vertex(), HepMC::WeightContainer::size(), v, HepMC::versionName(), HepMC::GenEvent::vertices_begin(), HepMC::GenEvent::vertices_end(), HepMC::GenEvent::vertices_size(), HepMC::GenEvent::weights(), and write_vertex().

8.20.3.2 bool HepMC::IO_Ascii::fill_next_event (GenEvent * *evt*) [virtual]

get the next event

Implements **HepMC::IO_BaseClass** (p. 152).

Definition at line 103 of file IO_Ascii.cc.

References HepMC::ascii, HepMC::CommonIO::find_end_key(), HepMC::CommonIO::find_file_type(), and HepMC::CommonIO::read_io_ascii().

8.20.3.3 void HepMC::IO_Ascii::write_particle_data_table (const ParticleDataTable *) [virtual]

write this **ParticleDataTable** (p. 213)

Implements **HepMC::IO_BaseClass** (p. 152).

Definition at line 181 of file IO_Ascii.cc.

References HepMC::ParticleDataTable::begin(), HepMC::ParticleDataTable::end(), write_end_listing(), and write_particle_data().

8.20.3.4 bool HepMC::IO_Ascii::fill_particle_data_table (ParticleDataTable *) [virtual]

fill this **ParticleDataTable** (p. 213)

Implements **HepMC::IO_BaseClass** (p. 153).

Definition at line 201 of file IO_Ascii.cc.

References HepMC::ascii_pdt, HepMC::CommonIO::find_end_key(), HepMC::CommonIO::find_file_type(), HepMC::CommonIO::read_io_particle_data_table(), and HepMC::ParticleDataTable::set_description().

8.20.3.5 void HepMC::IO_Ascii::write_comment (const std::string *comment*)

insert a comment directly into the output file — normally you only want to do this at the beginning or end of the file. All comments are preceded with "HepMC:IO_Ascii-COMMENT\n"

Definition at line 166 of file IO_Ascii.cc.

References write_end_listing().

8.20.3.6 int HepMC::IO_Ascii::rdstate () const [inline]

check the state of the IO stream

Definition at line 125 of file IO_Ascii.h.

8.20.3.7 void HepMC::IO_Ascii::clear () [inline]

clear the IO stream

Definition at line 126 of file IO_Ascii.h.

8.20.3.8 void HepMC::IO_Ascii::print (std::ostream & *ostr* = std::cout) const [virtual]

write to ostr

Reimplemented from **HepMC::IO_BaseClass** (p.153).

Definition at line 44 of file IO_Ascii.cc.

8.20.3.9 void HepMC::IO_Ascii::write_vertex (GenVertex *) [protected]

write vertex information

Definition at line 249 of file IO_Ascii.cc.

References output(), v, and write_particle().

Referenced by write_event().

8.20.3.10 void HepMC::IO_Ascii::write_particle (GenParticle * *p*) [protected]

write particle information

Definition at line 295 of file IO_Ascii.cc.

References output(), and p.

Referenced by write_vertex().

8.20.3.11 void HepMC::IO_Ascii::write_particle_data (const ParticleData * *d*) [protected]

write **ParticleDataTable** (p.213) information

Definition at line 319 of file IO_Ascii.cc.

References HepMC::ParticleData::charge(), HepMC::ParticleData::clifetime(), HepMC::ParticleData::mass(), HepMC::ParticleData::name(), output(), HepMC::ParticleData::pdg_id(), and HepMC::ParticleData::spin().

Referenced by write_particle_data_table().

8.20.3.12 bool HepMC::IO_Ascii::write_end_listing () [protected]

write end tag

Definition at line 337 of file IO_Ascii.cc.

References HepMC::CommonIO::write_IO_Ascii_End().

Referenced by write_comment(), write_particle_data_table(), and ~IO_Ascii().

8.20.3.13 void HepMC::IO_Ascii::output (const double &) [inline, protected]

write double

Definition at line 115 of file IO_Ascii.h.

Referenced by write_event(), write_particle(), write_particle_data(), and write_vertex().

8.20.3.14 void HepMC::IO_Ascii::output (const int &) [inline, protected]

write int

Definition at line 122 of file IO_Ascii.h.

8.20.3.15 void HepMC::IO_Ascii::output (const long int &) [inline, protected]

write long int

Definition at line 123 of file IO_Ascii.h.

8.20.3.16 void HepMC::IO_Ascii::output (const char &) [inline, protected]

write a single character

Definition at line 124 of file IO_Ascii.h.

The documentation for this class was generated from the following files:

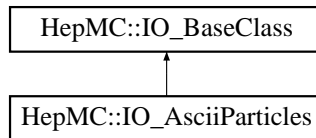
- IO_Ascii.h
- IO_Ascii.cc

8.21 HepMC::IO_AsciiParticles Class Reference

event input/output in ascii format for eye and machine reading

```
#include <IO_AsciiParticles.h>
```

Inheritance diagram for HepMC::IO_AsciiParticles::



Public Member Functions

- **IO_AsciiParticles** (const char *filename="IO_AsciiParticles.dat", std::ios::openmode mode=std::ios::out)
constructor requiring a file name and std::ios mode
- virtual **~IO_AsciiParticles** ()
- void **write_event** (const **GenEvent** *evt)
write this event
- bool **fill_next_event** (**GenEvent** *evt)
get the next event
- void **write_particle_data_table** (const **ParticleDataTable** *)
write this ParticleDataTable (p. 213)
- bool **fill_particle_data_table** (**ParticleDataTable** *)
fill this ParticleDataTable (p. 213)
- void **write_comment** (const std::string comment)
- void **setPrecision** (int iprec)
set output precision
- int **rdstate** () const
check the state of the IO stream
- void **clear** ()
clear the IO stream
- void **print** (std::ostream &ostr=std::cout) const
write to ostr

Protected Member Functions

- bool **write_end_listing** ()
write end tag

8.21.1 Detailed Description

event input/output in ascii format for eye and machine reading

Strategy for reading or writing events/particleData as machine readable ascii to a file. When instantiating, the mode of file to be created must be specified.

Examples:

example_PythiaParticle.cc.

Definition at line 54 of file IO_AsciiParticles.h.

8.21.2 Constructor & Destructor Documentation

8.21.2.1 HepMC::IO_AsciiParticles::IO_AsciiParticles (const char * *filename* = "IO_AsciiParticles.dat", std::ios::openmode *mode* = std::ios::out)

constructor requiring a file name and std::ios mode

Definition at line 18 of file IO_AsciiParticles.cc.

8.21.2.2 HepMC::IO_AsciiParticles::~~IO_AsciiParticles () [virtual]

Definition at line 47 of file IO_AsciiParticles.cc.

8.21.3 Member Function Documentation

8.21.3.1 void HepMC::IO_AsciiParticles::write_event (const GenEvent * *evt*) [virtual]

write this event

Implements **HepMC::IO_BaseClass** (p.152).

Definition at line 64 of file IO_AsciiParticles.cc.

References HepMC::GenEvent::alphaQCD(), HepMC::GenEvent::alphaQED(), HepMC::GenVertex::barcode(), HepMC::WeightContainer::begin(), HepMC::WeightContainer::end(), HepMC::GenEvent::event_number(), HepMC::GenEvent::event_scale(), HepMC::GenEvent::particles_begin(), HepMC::GenEvent::particles_end(), HepMC::GenEvent::particles_size(), HepMC::GenEvent::random_states(), HepMC::GenEvent::signal_process_id(), HepMC::GenEvent::signal_process_vertex(), HepMC::WeightContainer::size(), HepMC::versionName(), HepMC::GenEvent::vertices_size(), and HepMC::GenEvent::weights().

8.21.3.2 bool HepMC::IO_AsciiParticles::fill_next_event (GenEvent * *evt*) [virtual]

get the next event

Implements **HepMC::IO_BaseClass** (p.152).

Definition at line 180 of file IO_AsciiParticles.cc.

8.21.3.3 void HepMC::IO_AsciiParticles::write_particle_data_table (const ParticleDataTable *) [inline, virtual]

write this **ParticleDataTable** (p. 213)

Implements **HepMC::IO_BaseClass** (p. 152).

Definition at line 106 of file IO_AsciiParticles.h.

8.21.3.4 bool HepMC::IO_AsciiParticles::fill_particle_data_table (ParticleDataTable *) [inline, virtual]

fill this **ParticleDataTable** (p. 213)

Implements **HepMC::IO_BaseClass** (p. 153).

Definition at line 107 of file IO_AsciiParticles.h.

8.21.3.5 void HepMC::IO_AsciiParticles::write_comment (const std::string comment)

insert a comment directly into the output file — normally you only want to do this at the beginning or end of the file. All comments are preceded with "HepMC::IO_AsciiParticles-COMMENT\n"

Definition at line 203 of file IO_AsciiParticles.cc.

References `write_end_listing()`.

8.21.3.6 void HepMC::IO_AsciiParticles::setPrecision (int iprec) [inline]

set output precision

Definition at line 100 of file IO_AsciiParticles.h.

8.21.3.7 int HepMC::IO_AsciiParticles::rdstate () const [inline]

check the state of the IO stream

Definition at line 98 of file IO_AsciiParticles.h.

8.21.3.8 void HepMC::IO_AsciiParticles::clear () [inline]

clear the IO stream

Definition at line 99 of file IO_AsciiParticles.h.

8.21.3.9 void HepMC::IO_AsciiParticles::print (std::ostream & ostr = std::cout) const [virtual]

write to ostr

Reimplemented from **HepMC::IO_BaseClass** (p. 153).

Definition at line 54 of file IO_AsciiParticles.cc.

8.21.3.10 `bool HepMC::IO_AsciiParticles::write_end_listing ()` [protected]

write end tag

Definition at line 218 of file `IO_AsciiParticles.cc`.

Referenced by `write_comment()`.

The documentation for this class was generated from the following files:

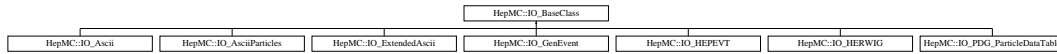
- `IO_AsciiParticles.h`
- `IO_AsciiParticles.cc`

8.22 HepMC::IO_BaseClass Class Reference

all input/output classes inherit from **IO_BaseClass** (p. 151)

```
#include <IO_BaseClass.h>
```

Inheritance diagram for HepMC::IO_BaseClass::



Public Member Functions

- virtual **~IO_BaseClass** ()
- virtual void **write_event** (const **GenEvent** *)=0
*write this **GenEvent** (p. 54)*
- virtual bool **fill_next_event** (**GenEvent** *)=0
*fill this **GenEvent** (p. 54)*
- virtual void **write_particle_data_table** (const **ParticleDataTable** *)=0
*write this **ParticleDataTable** (p. 213)*
- virtual bool **fill_particle_data_table** (**ParticleDataTable** *)=0
*fill this **ParticleDataTable** (p. 213)*
- virtual void **print** (std::ostream &ostr=std::cout) const
write output to ostr
- **GenEvent** * **read_next_event** ()
do not over-ride
- **ParticleDataTable** * **read_particle_data_table** ()
do not over-ride
- virtual **GenEvent** *& **operator>>** (**GenEvent** *&)
the same as read_next_event
- virtual const **GenEvent** *& **operator<<** (const **GenEvent** *&)
the same as write_event
- virtual **GenEvent** *& **operator<<** (**GenEvent** *&)
the same as write_event
- virtual **ParticleDataTable** *& **operator>>** (**ParticleDataTable** *&)
the same as read_particle_data_table
- virtual const **ParticleDataTable** *& **operator<<** (const **ParticleDataTable** *&)
the same as write_particle_data_table

- virtual **ParticleDataTable** *& **operator<<** (**ParticleDataTable** *&)
the same as write_particle_data_table

8.22.1 Detailed Description

all input/output classes inherit from **IO_BaseClass** (p. 151)

If you want to write a new IO class, then inherit from this class and re-define **read_event()** and **write_event()** (p. 152)

Definition at line 35 of file **IO_BaseClass.h**.

8.22.2 Constructor & Destructor Documentation

8.22.2.1 virtual **HepMC::IO_BaseClass::~IO_BaseClass** () [inline, virtual]

Definition at line 37 of file **IO_BaseClass.h**.

8.22.3 Member Function Documentation

8.22.3.1 virtual void **HepMC::IO_BaseClass::write_event** (const **GenEvent** *) [pure virtual]

write this **GenEvent** (p. 54)

Implemented in **HepMC::IO_Ascii** (p. 143), **HepMC::IO_AsciiParticles** (p. 148), **HepMC::IO_ExtendedAscii** (p. 157), **HepMC::IO_GenEvent** (p. 165), and **HepMC::IO_HEPEVT** (p. 172).

Referenced by **operator<<()**.

8.22.3.2 virtual bool **HepMC::IO_BaseClass::fill_next_event** (**GenEvent** *) [pure virtual]

fill this **GenEvent** (p. 54)

Implemented in **HepMC::IO_Ascii** (p. 144), **HepMC::IO_AsciiParticles** (p. 148), **HepMC::IO_ExtendedAscii** (p. 157), **HepMC::IO_GenEvent** (p. 165), **HepMC::IO_HEPEVT** (p. 171), and **HepMC::IO_HERWIG** (p. 178).

Referenced by **read_next_event()**.

8.22.3.3 virtual void **HepMC::IO_BaseClass::write_particle_data_table** (const **ParticleDataTable** *) [pure virtual]

write this **ParticleDataTable** (p. 213)

Implemented in **HepMC::IO_Ascii** (p. 144), **HepMC::IO_AsciiParticles** (p. 149), **HepMC::IO_ExtendedAscii** (p. 157), and **HepMC::IO_GenEvent** (p. 165).

Referenced by **operator<<()**.

8.22.3.4 virtual bool HepMC::IO_BaseClass::fill_particle_data_table (ParticleDataTable *) [pure virtual]

fill this ParticleDataTable (p. 213)

Implemented in HepMC::IO_Ascii (p. 144), HepMC::IO_AsciiParticles (p. 149), HepMC::IO_ExtendedAscii (p. 158), HepMC::IO_GenEvent (p. 166), and HepMC::IO_PDG_ParticleDataTable (p. 184).

Referenced by read_particle_data_table().

8.22.3.5 void HepMC::IO_BaseClass::print (std::ostream & ostr = std::cout) const [inline, virtual]

write output to ostr

Reimplemented in HepMC::IO_Ascii (p. 145), HepMC::IO_AsciiParticles (p. 149), HepMC::IO_ExtendedAscii (p. 158), HepMC::IO_GenEvent (p. 166), HepMC::IO_HEPEVT (p. 172), HepMC::IO_HERWIG (p. 178), and HepMC::IO_PDG_ParticleDataTable (p. 184).

Definition at line 117 of file IO_BaseClass.h.

8.22.3.6 GenEvent * HepMC::IO_BaseClass::read_next_event () [inline]

do not over-ride

creates a new event and fills it by calling the sister method read_next_event(GenEvent*)

Examples:

example_MyHerwig.cc, example_MyPythia.cc, example_MyPythiaOnlyToHepMC.cc, example_MyPythiaRead.cc, example_MyPythiaWithEventSelection.cc, example_PythiaParticle.cc, testHerwigCopies.cc, and testPythiaCopies.cc.

Definition at line 87 of file IO_BaseClass.h.

References fill_next_event().

Referenced by main(), and operator>>().

8.22.3.7 ParticleDataTable * HepMC::IO_BaseClass::read_particle_data_table () [inline]

do not over-ride

creates a new particle data table and fills it by calling the sister method read_particle_data_table(ParticleDataTable*)

Definition at line 103 of file IO_BaseClass.h.

References fill_particle_data_table().

Referenced by operator>>().

8.22.3.8 `GenEvent *& HepMC::IO_BaseClass::operator>> (GenEvent *&)`
[inline, virtual]

the same as `read_next_event`

Definition at line 121 of file `IO_BaseClass.h`.

References `read_next_event()`.

8.22.3.9 `const GenEvent *& HepMC::IO_BaseClass::operator<< (const GenEvent *&)` [inline, virtual]

the same as `write_event`

Definition at line 126 of file `IO_BaseClass.h`.

References `write_event()`.

8.22.3.10 `GenEvent *& HepMC::IO_BaseClass::operator<< (GenEvent *&)`
[inline, virtual]

the same as `write_event`

Definition at line 132 of file `IO_BaseClass.h`.

References `write_event()`.

8.22.3.11 `ParticleDataTable *& HepMC::IO_BaseClass::operator>> (ParticleDataTable *&)` [inline, virtual]

the same as `read_particle_data_table`

Definition at line 137 of file `IO_BaseClass.h`.

References `read_particle_data_table()`.

8.22.3.12 `const ParticleDataTable *& HepMC::IO_BaseClass::operator<< (const ParticleDataTable *&)` [inline, virtual]

the same as `write_particle_data_table`

Definition at line 143 of file `IO_BaseClass.h`.

References `write_particle_data_table()`.

8.22.3.13 `ParticleDataTable *& HepMC::IO_BaseClass::operator<< (ParticleDataTable *&)` [inline, virtual]

the same as `write_particle_data_table`

Definition at line 149 of file `IO_BaseClass.h`.

References `write_particle_data_table()`.

The documentation for this class was generated from the following file:

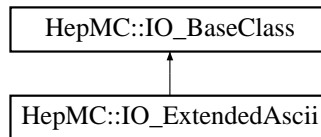
- `IO_BaseClass.h`

8.23 HepMC::IO_ExtendedAscii Class Reference

IO_ExtendedAscii (p. 155) also deals with **HeavyIon** (p. 121) and **PdfInfo** (p. 220).

```
#include <IO_ExtendedAscii.h>
```

Inheritance diagram for HepMC::IO_ExtendedAscii::



Public Member Functions

- **IO_ExtendedAscii** (const char *filename="IO_ExtendedAscii.dat", std::ios::openmode mode=std::ios::out)
constructor requiring a file name and std::ios mode
- virtual ~**IO_ExtendedAscii** ()
- void **write_event** (const **GenEvent** *evt)
write this event
- bool **fill_next_event** (**GenEvent** *evt)
get the next event
- void **write_particle_data_table** (const **ParticleDataTable** *)
*write this **ParticleDataTable** (p. 213)*
- bool **fill_particle_data_table** (**ParticleDataTable** *)
*fill this **ParticleDataTable** (p. 213)*
- void **write_comment** (const std::string comment)
- int **rdstate** () const
check the state of the IO stream
- void **clear** ()
clear the IO stream
- void **print** (std::ostream &ostr=std::cout) const
write to ostr

Protected Member Functions

- void **write_vertex** (**GenVertex** *)
write vertex information

- void **write_beam_particles** (std::pair< **HepMC::GenParticle** *, **HepMC::GenParticle** * >)
write beam particle information
- void **write_heavy_ion** (**HeavyIon** *)
write heavy ion information
- void **write_pdf_info** (**PdfInfo** *)
write PDF information
- void **write_particle** (**GenParticle** *p)
write particle information
- void **write_particle_data** (const **ParticleData** *d)
write particle data information
- bool **write_end_listing** ()
write end tag
- void **output** (const double &)
write double
- void **output** (const float &)
write float
- void **output** (const int &)
write int
- void **output** (const long int &)
write long int
- void **output** (const char &)
write a single character

8.23.1 Detailed Description

IO_ExtendedAscii (p. 155) also deals with **HeavyIon** (p. 121) and **PdfInfo** (p. 220).

event input/output in ascii format for machine reading extended format contains **HeavyIon** (p. 121) and **PdfInfo** (p. 220) classes

Definition at line 65 of file **IO_ExtendedAscii.h**.

8.23.2 Constructor & Destructor Documentation

8.23.2.1 **HepMC::IO_ExtendedAscii::IO_ExtendedAscii** (const char * *filename* = "IO_ExtendedAscii.dat", std::ios::openmode *mode* = std::ios::out)

constructor requiring a file name and std::ios mode

Definition at line 19 of file **IO_ExtendedAscii.cc**.

8.23.2.2 HepMC::IO_ExtendedAscii::~~IO_ExtendedAscii () [virtual]

Definition at line 42 of file IO_ExtendedAscii.cc.

References `write_end_listing()`.

8.23.3 Member Function Documentation**8.23.3.1 void HepMC::IO_ExtendedAscii::write_event (const GenEvent * evt) [virtual]**

write this event

Writes evt to m_file. It does NOT delete the event after writing.

Implements **HepMC::IO_BaseClass** (p. 152).

Definition at line 57 of file IO_ExtendedAscii.cc.

References `HepMC::GenEvent::alphaQCD()`, `HepMC::GenEvent::alphaQED()`, `HepMC::GenVertex::barcode()`, `HepMC::GenEvent::beam_particles()`, `HepMC::WeightContainer::begin()`, `HepMC::WeightContainer::end()`, `HepMC::GenEvent::event_number()`, `HepMC::GenEvent::event_scale()`, `HepMC::GenEvent::heavy_ion()`, `HepMC::GenEvent::mpi()`, `output()`, `HepMC::GenEvent::pdf_info()`, `HepMC::GenEvent::random_states()`, `HepMC::GenEvent::signal_process_id()`, `HepMC::GenEvent::signal_process_vertex()`, `HepMC::WeightContainer::size()`, `v`, `HepMC::versionName()`, `HepMC::GenEvent::vertices_begin()`, `HepMC::GenEvent::vertices_end()`, `HepMC::GenEvent::vertices_size()`, `HepMC::GenEvent::weights()`, `write_beam_particles()`, `write_heavy_ion()`, `HepMC::CommonIO::write_IO_ExtendedAscii_Key()`, `write_pdf_info()`, and `write_vertex()`.

8.23.3.2 bool HepMC::IO_ExtendedAscii::fill_next_event (GenEvent * evt) [virtual]

get the next event

Implements **HepMC::IO_BaseClass** (p. 152).

Definition at line 111 of file IO_ExtendedAscii.cc.

References `HepMC::extascii`, `HepMC::CommonIO::find_end_key()`, `HepMC::CommonIO::find_file_type()`, and `HepMC::CommonIO::read_io_extendedascii()`.

8.23.3.3 void HepMC::IO_ExtendedAscii::write_particle_data_table (const ParticleDataTable *) [virtual]

write this **ParticleDataTable** (p. 213)

Implements **HepMC::IO_BaseClass** (p. 152).

Definition at line 188 of file IO_ExtendedAscii.cc.

References `HepMC::ParticleDataTable::begin()`, `HepMC::ParticleDataTable::end()`, `write_end_listing()`, and `write_particle_data()`.

8.23.3.4 `bool HepMC::IO_ExtendedAscii::fill_particle_data_table (ParticleDataTable *) [virtual]`

fill this **ParticleDataTable** (p. 213)

Implements **HepMC::IO_BaseClass** (p. 153).

Definition at line 208 of file `IO_ExtendedAscii.cc`.

References `HepMC::extascii_pdt`, `HepMC::CommonIO::find_end_key()`, `HepMC::CommonIO::find_file_type()`, `HepMC::CommonIO::read_io_particle_data_table()`, and `HepMC::ParticleDataTable::set_description()`.

8.23.3.5 `void HepMC::IO_ExtendedAscii::write_comment (const std::string comment)`

insert a comment directly into the output file — normally you only want to do this at the beginning or end of the file. All comments are preceded with "HepMC::IO_ExtendedAscii-COMMENT\n"

Definition at line 173 of file `IO_ExtendedAscii.cc`.

References `write_end_listing()`.

8.23.3.6 `int HepMC::IO_ExtendedAscii::rdstate () const [inline]`

check the state of the IO stream

Definition at line 140 of file `IO_ExtendedAscii.h`.

8.23.3.7 `void HepMC::IO_ExtendedAscii::clear () [inline]`

clear the IO stream

Definition at line 141 of file `IO_ExtendedAscii.h`.

8.23.3.8 `void HepMC::IO_ExtendedAscii::print (std::ostream & ostr = std::cout) const [virtual]`

write to ostr

Reimplemented from **HepMC::IO_BaseClass** (p. 153).

Definition at line 47 of file `IO_ExtendedAscii.cc`.

8.23.3.9 `void HepMC::IO_ExtendedAscii::write_vertex (GenVertex *) [protected]`

write vertex information

Definition at line 256 of file `IO_ExtendedAscii.cc`.

References `output()`, `v`, and `write_particle()`.

Referenced by `write_event()`.

8.23.3.10 void HepMC::IO_ExtendedAscii::write_beam_particles (std::pair< HepMC::GenParticle *, HepMC::GenParticle * >) [protected]

write beam particle information

Definition at line 302 of file IO_ExtendedAscii.cc.

References output(), and p.

Referenced by write_event().

8.23.3.11 void HepMC::IO_ExtendedAscii::write_heavy_ion (HeavyIon *) [protected]

write heavy ion information

Definition at line 319 of file IO_ExtendedAscii.cc.

References HepMC::HeavyIon::eccentricity(), HepMC::HeavyIon::event_plane_angle(), HepMC::HeavyIon::impact_parameter(), HepMC::HeavyIon::N_Nwounded_collisions(), HepMC::HeavyIon::Ncoll(), HepMC::HeavyIon::Ncoll_hard(), HepMC::HeavyIon::Npart_proj(), HepMC::HeavyIon::Npart_targ(), HepMC::HeavyIon::Nwounded_N_collisions(), HepMC::HeavyIon::Nwounded_Nwounded_collisions(), output(), HepMC::HeavyIon::sigma_inel_NN(), HepMC::HeavyIon::spectator_neutrons(), and HepMC::HeavyIon::spectator_protons().

Referenced by write_event().

8.23.3.12 void HepMC::IO_ExtendedAscii::write_pdf_info (PdfInfo *) [protected]

write PDF information

Definition at line 363 of file IO_ExtendedAscii.cc.

References HepMC::PdfInfo::id1(), HepMC::PdfInfo::id2(), output(), HepMC::PdfInfo::pdf1(), HepMC::PdfInfo::pdf2(), HepMC::PdfInfo::scalePDF(), HepMC::PdfInfo::x1(), and HepMC::PdfInfo::x2().

Referenced by write_event().

8.23.3.13 void HepMC::IO_ExtendedAscii::write_particle (GenParticle * p) [protected]

write particle information

Definition at line 395 of file IO_ExtendedAscii.cc.

References output(), and p.

Referenced by write_vertex().

8.23.3.14 void HepMC::IO_ExtendedAscii::write_particle_data (const ParticleData * d) [protected]

write particle data information

Definition at line 420 of file IO_ExtendedAscii.cc.

References HepMC::ParticleData::charge(), HepMC::ParticleData::clifetime(), HepMC::ParticleData::mass(), HepMC::ParticleData::name(), output(), HepMC::ParticleData::pdg_id(), and HepMC::ParticleData::spin().

Referenced by write_particle_data_table().

8.23.3.15 bool HepMC::IO_ExtendedAscii::write_end_listing () [protected]

write end tag

Definition at line 438 of file IO_ExtendedAscii.cc.

References HepMC::CommonIO::write_IO_ExtendedAscii_End().

Referenced by write_comment(), write_particle_data_table(), and ~IO_ExtendedAscii().

8.23.3.16 void HepMC::IO_ExtendedAscii::output (const double &) [inline, protected]

write double

Definition at line 123 of file IO_ExtendedAscii.h.

Referenced by write_beam_particles(), write_event(), write_heavy_ion(), write_particle(), write_particle_data(), write_pdf_info(), and write_vertex().

8.23.3.17 void HepMC::IO_ExtendedAscii::output (const float &) [inline, protected]

write float

Definition at line 130 of file IO_ExtendedAscii.h.

8.23.3.18 void HepMC::IO_ExtendedAscii::output (const int &) [inline, protected]

write int

Definition at line 137 of file IO_ExtendedAscii.h.

8.23.3.19 void HepMC::IO_ExtendedAscii::output (const long int &) [inline, protected]

write long int

Definition at line 138 of file IO_ExtendedAscii.h.

8.23.3.20 void HepMC::IO_ExtendedAscii::output (const char &) [inline, protected]

write a single character

Definition at line 139 of file IO_ExtendedAscii.h.

The documentation for this class was generated from the following files:

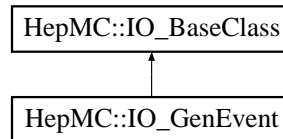
- `IO_ExtendedAscii.h`
- `IO_ExtendedAscii.cc`

8.24 HepMC::IO_GenEvent Class Reference

IO_GenEvent (p. 162) also deals with **HeavyIon** (p. 121) and **PdfInfo** (p. 220).

```
#include <IO_GenEvent.h>
```

Inheritance diagram for HepMC::IO_GenEvent::



Public Member Functions

- **IO_GenEvent** (const char *filename="IO_GenEvent.dat", std::ios::openmode mode=std::ios::out)
constructor requiring a file name and std::ios mode
- **IO_GenEvent** (std::istream &)
constructor requiring an input stream
- **IO_GenEvent** (std::ostream &)
constructor requiring an output stream
- virtual **~IO_GenEvent** ()
- void **write_event** (const **GenEvent** *evt)
write this event
- bool **fill_next_event** (**GenEvent** *evt)
get the next event
- void **write_particle_data_table** (const **ParticleDataTable** *)
*write this **ParticleDataTable** (p. 213)*
- bool **fill_particle_data_table** (**ParticleDataTable** *)
*fill this **ParticleDataTable** (p. 213)*
- void **write_comment** (const std::string comment)
- int **rdstate** () const
check the state of the IO stream
- void **clear** ()
clear the IO stream
- void **print** (std::ostream &ostr=std::cout) const
write to ostr

Protected Member Functions

- void **write_vertex** (**GenVertex** *)
write vertex information
- void **write_beam_particles** (std::pair< **HepMC::GenParticle** *, **HepMC::GenParticle** * >)
write beam particle information
- void **write_heavy_ion** (**HeavyIon** *)
write heavy ion information
- void **write_pdf_info** (**PdfInfo** *)
write PDF information
- void **write_particle** (**GenParticle** *p)
write particle information
- void **write_particle_data** (const **ParticleData** *d)
write particle data information
- **GenVertex** * **read_vertex** (**TempParticleMap** &particle_to_end_vertex)
read vertex information
- **GenParticle** * **read_particle** (**TempParticleMap** &particle_to_end_vertex)
*read **GenParticle** (p. 86) information*
- **ParticleData** * **read_particle_data** (**ParticleDataTable** *)
read particle data table information
- **HeavyIon** * **read_heavy_ion** ()
read heavy ion information
- **PdfInfo** * **read_pdf_info** ()
read PDF information
- bool **write_end_listing** ()
write end tag
- void **output** (const double &)
write double
- void **output** (const float &)
write float
- void **output** (const int &)
write int
- void **output** (const long &)
write long

- void **output** (const char &)
write a single character

8.24.1 Detailed Description

IO_GenEvent (p. 162) also deals with **HeavyIon** (p. 121) and **PdfInfo** (p. 220).

event input/output in ascii format for machine reading extended format contains **HeavyIon** (p. 121) and **PdfInfo** (p. 220) classes

Strategy for reading or writing events using iostreams When instantiating with a file name, the mode of file to be created must be specified. Options are: std::ios::in open file for input std::ios::out open file for output std::ios::trunc erase old file when opening (i.e. ios::out|iostrunc removes oldfile, and creates a new one for output) std::ios::app append output to end of file for the purposes of this class, simultaneous input and output mode (std::ios::in | std::ios::out) is not allowed.

Event listings are preceded by the key: "HepMC::IO_GenEvent-START_EVENT_LISTING\n" and terminated by the key: "HepMC::IO_GenEvent-END_EVENT_LISTING\n" **GenParticle** (p. 86) Data tables are preceded by the key: "HepMC::IO_GenEvent-START_PARTICLE_DATA\n" and terminated by the key: "HepMC::IO_GenEvent-END_PARTICLE_DATA\n" Comments are allowed. They need not be preceded by anything, though if a comment is written using write_comment(const string) then it will be preceded by "HepMC::IO_GenEvent-COMMENT\n" Each event, vertex, particle, particle data, heavy ion, or pdf info line is preceded by "E ", "V ", "P ", "D ", "H ", "F " respectively. ExtendedAscii ignores particle data blocks Comments may appear anywhere in the file – so long as they do not contain any of the start/stop keys.

Examples:

example_EventSelection.cc, **example_MyPythia.cc**, **example_MyPythia-Read.cc**, and **example_UsingIterators.cc**.

Definition at line 65 of file IO_GenEvent.h.

8.24.2 Constructor & Destructor Documentation

8.24.2.1 HepMC::IO_GenEvent::IO_GenEvent (const char * *filename* = "IO_GenEvent.dat", std::ios::openmode *mode* = std::ios::out)

constructor requiring a file name and std::ios mode

Definition at line 19 of file IO_GenEvent.cc.

8.24.2.2 HepMC::IO_GenEvent::IO_GenEvent (std::istream &)

constructor requiring an input stream

Definition at line 56 of file IO_GenEvent.cc.

8.24.2.3 HepMC::IO_GenEvent::IO_GenEvent (std::ostream &)

constructor requiring an output stream

Definition at line 65 of file IO_GenEvent.cc.

8.24.2.4 HepMC::IO_GenEvent::~~IO_GenEvent () [virtual]

Definition at line 81 of file IO_GenEvent.cc.

References `write_end_listing()`.

8.24.3 Member Function Documentation

8.24.3.1 void HepMC::IO_GenEvent::write_event (const GenEvent * *evt*) [virtual]

write this event

Writes *evt* to output stream. It does NOT delete the event after writing.

Implements **HepMC::IO_BaseClass** (p. 152).

Definition at line 96 of file IO_GenEvent.cc.

References `HepMC::GenEvent::alphaQCD()`, `HepMC::GenEvent::alphaQED()`, `HepMC::GenVertex::barcode()`, `HepMC::GenEvent::beam_particles()`, `HepMC::WeightContainer::begin()`, `HepMC::WeightContainer::end()`, `HepMC::GenEvent::event_number()`, `HepMC::GenEvent::event_scale()`, `HepMC::GenEvent::heavy_ion()`, `HepMC::GenEvent::mpi()`, `output()`, `HepMC::GenEvent::pdf_info()`, `HepMC::GenEvent::random_states()`, `HepMC::GenEvent::signal_process_id()`, `HepMC::GenEvent::signal_process_vertex()`, `HepMC::WeightContainer::size()`, `v`, `HepMC::versionName()`, `HepMC::GenEvent::vertices_begin()`, `HepMC::GenEvent::vertices_end()`, `HepMC::GenEvent::vertices_size()`, `HepMC::GenEvent::weights()`, `write_beam_particles()`, `write_heavy_ion()`, `HepMC::CommonIO::write_IO_GenEvent_Key()`, `write_pdf_info()`, and `write_vertex()`.

8.24.3.2 bool HepMC::IO_GenEvent::fill_next_event (GenEvent * *evt*) [virtual]

get the next event

Implements **HepMC::IO_BaseClass** (p. 152).

Definition at line 150 of file IO_GenEvent.cc.

References `HepMC::ascii`, `HepMC::ascii_pdt`, `HepMC::extascii`, `HepMC::extascii_pdt`, `HepMC::CommonIO::find_end_key()`, `HepMC::CommonIO::find_file_type()`, `HepMC::gen`, `HepMC::CommonIO::io_type()`, `HepMC::CommonIO::read_io_ascii()`, `HepMC::CommonIO::read_io_extendedascii()`, and `HepMC::CommonIO::read_io_genevent()`.

8.24.3.3 void HepMC::IO_GenEvent::write_particle_data_table (const ParticleDataTable *) [inline, virtual]

write this **ParticleDataTable** (p. 213)

Implements **HepMC::IO_BaseClass** (p. 152).

Definition at line 203 of file IO_GenEvent.h.

8.24.3.4 `bool HepMC::IO_GenEvent::fill_particle_data_table (ParticleDataTable *)` `[inline, virtual]`

fill this **ParticleDataTable** (p. 213)

Implements **HepMC::IO_BaseClass** (p. 153).

Definition at line 204 of file `IO_GenEvent.h`.

8.24.3.5 `void HepMC::IO_GenEvent::write_comment (const std::string comment)`

insert a comment directly into the output file — normally you only want to do this at the beginning or end of the file. All comments are preceded with "HepMC::IO_GenEvent-COMMENT\n"

Definition at line 228 of file `IO_GenEvent.cc`.

References `write_end_listing()`.

8.24.3.6 `int HepMC::IO_GenEvent::rdstate () const` `[inline]`

check the state of the IO stream

Definition at line 186 of file `IO_GenEvent.h`.

Referenced by `main()`.

8.24.3.7 `void HepMC::IO_GenEvent::clear ()` `[inline]`

clear the IO stream

Definition at line 195 of file `IO_GenEvent.h`.

8.24.3.8 `void HepMC::IO_GenEvent::print (std::ostream & ostr = std::cout)`
`const` `[virtual]`

write to ostr

Reimplemented from **HepMC::IO_BaseClass** (p. 153).

Definition at line 86 of file `IO_GenEvent.cc`.

8.24.3.9 `void HepMC::IO_GenEvent::write_vertex (GenVertex *)` `[protected]`

write vertex information

Definition at line 243 of file `IO_GenEvent.cc`.

References `output()`, `v`, and `write_particle()`.

Referenced by `write_event()`.

8.24.3.10 `void HepMC::IO_GenEvent::write_beam_particles (std::pair< HepMC::GenParticle *, HepMC::GenParticle * >)` `[protected]`

write beam particle information

Definition at line 289 of file `IO_GenEvent.cc`.

References `output()`, and `p`.

Referenced by `write_event()`.

8.24.3.11 `void HepMC::IO_GenEvent::write_heavy_ion (HeavyIon *)` [protected]

write heavy ion information

Definition at line 306 of file `IO_GenEvent.cc`.

References `HepMC::HeavyIon::eccentricity()`, `HepMC::HeavyIon::event_plane_angle()`, `HepMC::HeavyIon::impact_parameter()`, `HepMC::HeavyIon::N_Nwounded_collisions()`, `HepMC::HeavyIon::Ncoll()`, `HepMC::HeavyIon::Ncoll_hard()`, `HepMC::HeavyIon::Npart_proj()`, `HepMC::HeavyIon::Npart_targ()`, `HepMC::HeavyIon::Nwounded_N_collisions()`, `HepMC::HeavyIon::Nwounded_Nwounded_collisions()`, `output()`, `HepMC::HeavyIon::sigma_inel_NN()`, `HepMC::HeavyIon::spectator_neutrons()`, and `HepMC::HeavyIon::spectator_protons()`.

Referenced by `write_event()`.

8.24.3.12 `void HepMC::IO_GenEvent::write_pdf_info (PdfInfo *)` [protected]

write PDF information

Definition at line 350 of file `IO_GenEvent.cc`.

References `HepMC::PdfInfo::id1()`, `HepMC::PdfInfo::id2()`, `output()`, `HepMC::PdfInfo::pdf1()`, `HepMC::PdfInfo::pdf2()`, `HepMC::PdfInfo::scalePDF()`, `HepMC::PdfInfo::x1()`, and `HepMC::PdfInfo::x2()`.

Referenced by `write_event()`.

8.24.3.13 `void HepMC::IO_GenEvent::write_particle (GenParticle * p)` [protected]

write particle information

Definition at line 382 of file `IO_GenEvent.cc`.

References `output()`, and `p`.

Referenced by `write_vertex()`.

8.24.3.14 `void HepMC::IO_GenEvent::write_particle_data (const ParticleData * d)` [protected]

write particle data information

Definition at line 407 of file `IO_GenEvent.cc`.

References `HepMC::ParticleData::charge()`, `HepMC::ParticleData::clifetime()`, `HepMC::ParticleData::mass()`, `HepMC::ParticleData::name()`, `output()`, `HepMC::ParticleData::pdg_id()`, and `HepMC::ParticleData::spin()`.

8.24.3.15 **GenVertex*** **HepMC::IO_GenEvent::read_vertex** (**TempParticleMap** & *particle_to_end_vertex*) [protected]

read vertex information

8.24.3.16 **GenParticle *** **HepMC::IO_GenEvent::read_particle** (**TempParticleMap** & *particle_to_end_vertex*) [protected]

read **GenParticle** (p. 86) information

Definition at line 473 of file `IO_GenEvent.cc`.

References `HepMC::TempParticleMap::addEndParticle()`, `p`, and `HepMC::Flow::set_icode()`.

8.24.3.17 **ParticleData *** **HepMC::IO_GenEvent::read_particle_data** (**ParticleDataTable ***) [protected]

read particle data table information

Definition at line 515 of file `IO_GenEvent.cc`.

References `HepMC::ParticleDataTable::insert()`.

8.24.3.18 **HeavyIon *** **HepMC::IO_GenEvent::read_heavy_ion** () [protected]

read heavy ion information

Definition at line 425 of file `IO_GenEvent.cc`.

8.24.3.19 **PdfInfo *** **HepMC::IO_GenEvent::read_pdf_info** () [protected]

read PDF information

Definition at line 451 of file `IO_GenEvent.cc`.

8.24.3.20 **bool** **HepMC::IO_GenEvent::write_end_listing** () [protected]

write end tag

Definition at line 537 of file `IO_GenEvent.cc`.

References `HepMC::CommonIO::write_IO_GenEvent_End()`.

Referenced by `write_comment()`, and `~IO_GenEvent()`.

8.24.3.21 **void** **HepMC::IO_GenEvent::output** (**const double** &) [inline, protected]

write double

Definition at line 141 of file `IO_GenEvent.h`.

Referenced by `write_beam_particles()`, `write_event()`, `write_heavy_ion()`, `write_particle()`, `write_particle_data()`, `write_pdf_info()`, and `write_vertex()`.

8.24.3.22 void HepMC::IO_GenEvent::output (const float &) [inline, protected]

write float

Definition at line 150 of file IO_GenEvent.h.

8.24.3.23 void HepMC::IO_GenEvent::output (const int &) [inline, protected]

write int

Definition at line 159 of file IO_GenEvent.h.

8.24.3.24 void HepMC::IO_GenEvent::output (const long &) [inline, protected]

write long

Definition at line 168 of file IO_GenEvent.h.

8.24.3.25 void HepMC::IO_GenEvent::output (const char &) [inline, protected]

write a single character

Definition at line 177 of file IO_GenEvent.h.

The documentation for this class was generated from the following files:

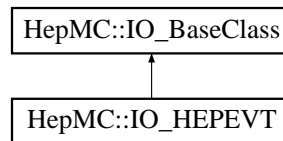
- **IO_GenEvent.h**
- **IO_GenEvent.cc**

8.25 HepMC::IO_HEPEVT Class Reference

HEPEVT IO class.

```
#include <IO_HEPEVT.h>
```

Inheritance diagram for HepMC::IO_HEPEVT::



Public Member Functions

- **IO_HEPEVT** ()
- virtual **~IO_HEPEVT** ()
- bool **fill_next_event** (**GenEvent** *)
*fill this **GenEvent** (p. 54)*
- void **write_event** (const **GenEvent** *)
*write this **GenEvent** (p. 54)*
- void **print** (std::ostream &ostr=std::cout) const
write output to ostr
- bool **trust_both_mothers_and_daughters** () const
default is false
- bool **trust_mothers_before_daughters** () const
default is true
- bool **print_inconsistency_errors** () const
default is true
- bool **trust_beam_particles** () const
default is true
- void **set_trust_mothers_before_daughters** (bool b=1)
define mother daughter trust rules
- void **set_trust_both_mothers_and_daughters** (bool b=0)
define mother daughter trust rules
- void **set_print_inconsistency_errors** (bool b=1)
- void **set_trust_beam_particles** (bool b=true)
declare whether or not beam particles exist

Protected Member Functions

- **GenParticle * build_particle** (int index)
*create a **GenParticle** (p. 86)*
- **void build_production_vertex** (int i, std::vector< **HepMC::GenParticle** * > &hepevt_particle, **GenEvent** *evt)
create a production vertex
- **void build_end_vertex** (int i, std::vector< **HepMC::GenParticle** * > &hepevt_particle, **GenEvent** *evt)
create an end vertex
- **int find_in_map** (const std::map< **HepMC::GenParticle** *, int > &m, **GenParticle** *p) const
find this particle in the particle map

8.25.1 Detailed Description

HEPEVT IO class.

IO class for reading the standard HEPEVT common block.

Examples:

`example_MyPythia.cc`, `example_MyPythiaOnlyToHepMC.cc`, `example_MyPythiaRead.cc`, `example_MyPythiaWithEventSelection.cc`, `example_PythiaParticle.cc`, and `testPythiaCopies.cc`.

Definition at line 40 of file `IO_HEPEVT.h`.

8.25.2 Constructor & Destructor Documentation

8.25.2.1 **HepMC::IO_HEPEVT::IO_HEPEVT** ()

Definition at line 12 of file `IO_HEPEVT.cc`.

8.25.2.2 **HepMC::IO_HEPEVT::~~IO_HEPEVT** () [virtual]

Definition at line 18 of file `IO_HEPEVT.cc`.

8.25.3 Member Function Documentation

8.25.3.1 **bool HepMC::IO_HEPEVT::fill_next_event** (**GenEvent** *) [virtual]

fill this **GenEvent** (p. 54)

Implements **HepMC::IO_BaseClass** (p. 152).

Definition at line 31 of file `IO_HEPEVT.cc`.

References `HepMC::GenVertex::add_particle_out()`, `HepMC::GenEvent::add_vertex()`, `build_end_vertex()`, `build_particle()`, `build_production_vertex()`, `HepMC::HEPEVT_Wrapper::event_number()`, `HepMC::HEPEVT_Wrapper::number_entries()`, `HepMC::GenEvent::set_beam_particles()`, `HepMC::GenEvent::set_event_number()`, and `trust_beam_particles()`.

8.25.3.2 `void HepMC::IO_HEPEVT::write_event (const GenEvent *) [virtual]`

write this **GenEvent** (p.54)

Implements **HepMC::IO_BaseClass** (p.152).

Definition at line 110 of file `IO_HEPEVT.cc`.

References `HepMC::FourVector::e()`, `HepMC::GenEvent::event_number()`, `find_in_map()`, `HepMC::HEPEVT_Wrapper::max_number_entries()`, `p`, `HepMC::FourVector::px()`, `HepMC::FourVector::py()`, `HepMC::FourVector::pz()`, `HepMC::HEPEVT_Wrapper::set_children()`, `HepMC::HEPEVT_Wrapper::set_event_number()`, `HepMC::HEPEVT_Wrapper::set_id()`, `HepMC::HEPEVT_Wrapper::set_mass()`, `HepMC::HEPEVT_Wrapper::set_momentum()`, `HepMC::HEPEVT_Wrapper::set_number_entries()`, `HepMC::HEPEVT_Wrapper::set_parents()`, `HepMC::HEPEVT_Wrapper::set_position()`, `HepMC::HEPEVT_Wrapper::set_status()`, `v`, `HepMC::GenEvent::vertices_begin()`, and `HepMC::GenEvent::vertices_end()`.

8.25.3.3 `void HepMC::IO_HEPEVT::print (std::ostream & ostr = std::cout) const [virtual]`

write output to ostr

Reimplemented from **HepMC::IO_BaseClass** (p.153).

Definition at line 20 of file `IO_HEPEVT.cc`.

8.25.3.4 `bool HepMC::IO_HEPEVT::trust_both_mothers_and_daughters () const [inline]`

default is false

Definition at line 120 of file `IO_HEPEVT.h`.

8.25.3.5 `bool HepMC::IO_HEPEVT::trust_mothers_before_daughters () const [inline]`

default is true

Definition at line 123 of file `IO_HEPEVT.h`.

8.25.3.6 `bool HepMC::IO_HEPEVT::print_inconsistency_errors () const [inline]`

default is true

Definition at line 126 of file `IO_HEPEVT.h`.

8.25.3.7 `bool HepMC::IO_HEPEVT::trust_beam_particles () const [inline]`

default is true

Definition at line 138 of file IO_HEPEVT.h.

Referenced by fill_next_event().

8.25.3.8 `void HepMC::IO_HEPEVT::set_trust_mothers_before_daughters (bool b = 1) [inline]`

define mother daughter trust rules

Definition at line 132 of file IO_HEPEVT.h.

8.25.3.9 `void HepMC::IO_HEPEVT::set_trust_both_mothers_and_daughters (bool b = 0) [inline]`

define mother daughter trust rules

Definition at line 129 of file IO_HEPEVT.h.

8.25.3.10 `void HepMC::IO_HEPEVT::set_print_inconsistency_errors (bool b = 1) [inline]`

Since HEPEVT has bi-directional pointers, it is possible that the mother/daughter pointers are inconsistent (though physically speaking this should never happen). In practise it happens often. When a conflict occurs (i.e. when mother/daughter pointers are in disagreement, where an empty (0) pointer is not considered a disagreement) an error is printed. These errors can be turned off with: `myio_hepevt.set_print_inconsistency_errors(0)`; but it is **STRONGLY** recommended that you print the HEPEVT common and understand the inconsistency **BEFORE** you turn off the errors. The messages are there for a reason [remember, there is no message printed when the information is missing, ... only when is it inconsistent. User beware.] You can inspect the HEPEVT common block for inconsistencies with **HEPEVT_Wrapper::check_hepevt_consistency()** (p. 132)

There is a switch controlling whether the mother pointers or the daughters are to be trusted. For example, in Pythia the mother information is always correctly included, but the daughter information is often left unfilled: in this case we want to trust the mother pointers and not necessarily the daughters. [THIS IS THE DEFAULT]. Unfortunately the reverse happens for the stdhep(2001) translation of Isajet, so we need an option to toggle the choices.

Definition at line 135 of file IO_HEPEVT.h.

8.25.3.11 `void HepMC::IO_HEPEVT::set_trust_beam_particles (bool b = true) [inline]`

declare whether or not beam particles exist

Definition at line 141 of file IO_HEPEVT.h.

8.25.3.12 **GenParticle * HepMC::IO_HEPEVT::build_particle (int *index*)** [protected]

create a **GenParticle** (p. 86)

Builds a particle object corresponding to index in HEPEVT

Definition at line 325 of file IO_HEPEVT.cc.

References HepMC::HEPEVT_Wrapper::e(), HepMC::HEPEVT_Wrapper::id(), HepMC::HEPEVT_Wrapper::m(), p, HepMC::HEPEVT_Wrapper::px(), HepMC::HEPEVT_Wrapper::py(), HepMC::HEPEVT_Wrapper::pz(), and HepMC::HEPEVT_Wrapper::status().

Referenced by fill_next_event().

8.25.3.13 **void HepMC::IO_HEPEVT::build_production_vertex (int *i*, std::vector< HepMC::GenParticle * > & *hepevt_particle*, GenEvent * *evt*)** [protected]

create a production vertex

for particle in HEPEVT with index *i*, build a production vertex if appropriate, and add that vertex to the event

Definition at line 191 of file IO_HEPEVT.cc.

References HepMC::GenVertex::add_particle_in(), HepMC::GenVertex::add_particle_out(), HepMC::GenEvent::add_vertex(), HepMC::HEPEVT_Wrapper::event_number(), HepMC::HEPEVT_Wrapper::first_parent(), HepMC::HEPEVT_Wrapper::last_parent(), HepMC::HEPEVT_Wrapper::number_parents(), p, HepMC::GenVertex::position(), HepMC::GenVertex::set_position(), HepMC::HEPEVT_Wrapper::t(), HepMC::HEPEVT_Wrapper::x(), HepMC::HEPEVT_Wrapper::y(), and HepMC::HEPEVT_Wrapper::z().

Referenced by build_end_vertex(), and fill_next_event().

8.25.3.14 **void HepMC::IO_HEPEVT::build_end_vertex (int *i*, std::vector< HepMC::GenParticle * > & *hepevt_particle*, GenEvent * *evt*)** [protected]

create an end vertex

for particle in HEPEVT with index *i*, build an end vertex if appropriate, and add that vertex to the event

Definition at line 257 of file IO_HEPEVT.cc.

References HepMC::GenVertex::add_particle_in(), HepMC::GenVertex::add_particle_out(), HepMC::GenEvent::add_vertex(), build_production_vertex(), HepMC::HEPEVT_Wrapper::event_number(), HepMC::HEPEVT_Wrapper::first_child(), HepMC::HEPEVT_Wrapper::last_child(), HepMC::HEPEVT_Wrapper::number_children(), p, HepMC::GenVertex::position(), HepMC::GenVertex::set_position(), HepMC::HEPEVT_Wrapper::t(), HepMC::HEPEVT_Wrapper::x(), HepMC::HEPEVT_Wrapper::y(), and HepMC::HEPEVT_Wrapper::z().

Referenced by fill_next_event().

8.25.3.15 `int HepMC::IO_HEPEVT::find_in_map (const std::map< HepMC::GenParticle *, int > & m, GenParticle * p) const` [protected]

find this particle in the particle map

Definition at line 340 of file IO_HEPEVT.cc.

References `p`.

Referenced by `write_event()`.

The documentation for this class was generated from the following files:

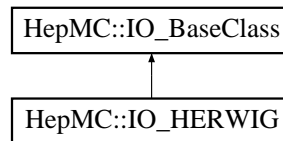
- `IO_HEPEVT.h`
- `IO_HEPEVT.cc`

8.26 HepMC::IO_HERWIG Class Reference

IO_HERWIG (p.176) is used to get Herwig information.

```
#include <IO_HERWIG.h>
```

Inheritance diagram for HepMC::IO_HERWIG::



Public Member Functions

- **IO_HERWIG** ()
- virtual **~IO_HERWIG** ()
- bool **fill_next_event** (GenEvent *)
get the next event
- void **print** (std::ostream &ostr=std::cout) const
write to ostr
- double **interfaces_to_version_number** () const
this information is dubious
- bool **print_inconsistency_errors** () const
default is true
- void **set_print_inconsistency_errors** (bool b=1)
decide whether or not to print inconsistency errors
- bool **no_gaps_in_barcodes** () const
ask how to deal with extra non-physical pseudo particles
- void **set_no_gaps_in_barcodes** (bool a)

Protected Member Functions

- bool **trust_both_mothers_and_daughters** () const
default is true
- bool **trust_mothers_before_daughters** () const
default is false
- void **set_trust_mothers_before_daughters** (bool b=1)
define mother daughter trust rules
- void **set_trust_both_mothers_and_daughters** (bool b=0)

define mother daughter trust rules

- **GenParticle * build_particle** (int index)
make a particle
- **void build_production_vertex** (int i, std::vector< **GenParticle** * > &hepevt_particle, **GenEvent** *evt)
make a production vertex
- **void build_end_vertex** (int i, std::vector< **GenParticle** * > &hepevt_particle, **GenEvent** *evt)
make a decay vertex
- **int find_in_map** (const std::map< **GenParticle** *, int > &m, **GenParticle** *p) const
find this particle in the map
- **void repair_hepevt** () const
make the HERWIG HEPEVT common block look like the standard
- **void remove_gaps_in_hepevt** () const
deal with artifacts of repairing HEPEVT
- **void zero_hepevt_entry** (int i) const
zero out a HEPEVT pseudo particle
- **int translate_herwig_to_pdg_id** (int i) const
translate particle ID

8.26.1 Detailed Description

IO_HERWIG (p. 176) is used to get Herwig information.

IO class for reading the HEPEVT common block from the Herwig monte carlo program.

Examples:

example_MyHerwig.cc, and **testHerwigCopies.cc**.

Definition at line 57 of file **IO_HERWIG.h**.

8.26.2 Constructor & Destructor Documentation

8.26.2.1 HepMC::IO_HERWIG::IO_HERWIG ()

Definition at line 12 of file **IO_HERWIG.cc**.

8.26.2.2 HepMC::IO_HERWIG::~~IO_HERWIG () [virtual]

Definition at line 83 of file **IO_HERWIG.cc**.

8.26.3 Member Function Documentation

8.26.3.1 `bool HepMC::IO_HERWIG::fill_next_event (GenEvent *) [virtual]`

get the next event

read one event from the Herwig HEPEVT common block and fill **GenEvent** (p.54) return T/F =success/failure

sufficient to do one or the other.

Implements **HepMC::IO_BaseClass** (p.152).

Definition at line 96 of file IO_HERWIG.cc.

References `HepMC::GenVertex::add_particle_in()`, `HepMC::GenVertex::add_particle_out()`, `HepMC::GenEvent::add_vertex()`, `build_end_vertex()`, `build_particle()`, `build_production_vertex()`, `HepMC::HEPEVT_Wrapper::event_number()`, `HepMC::HEPEVT_Wrapper::first_parent()`, `HepMC::HEPEVT_Wrapper::number_entries()`, `repair_hepevt()`, `HepMC::GenEvent::set_beam_particles()`, `HepMC::GenEvent::set_event_number()`, `HepMC::GenEvent::set_signal_process_vertex()`, and `HepMC::HEPEVT_Wrapper::status()`.

8.26.3.2 `void HepMC::IO_HERWIG::print (std::ostream & ostr = std::cout) const [virtual]`

write to ostr

Reimplemented from **HepMC::IO_BaseClass** (p.153).

Definition at line 85 of file IO_HERWIG.cc.

8.26.3.3 `double HepMC::IO_HERWIG::interfaces_to_version_number () const [inline]`

this information is dubious

Definition at line 66 of file IO_HERWIG.h.

8.26.3.4 `bool HepMC::IO_HERWIG::print_inconsistency_errors () const [inline]`

default is true

Definition at line 149 of file IO_HERWIG.h.

8.26.3.5 `void HepMC::IO_HERWIG::set_print_inconsistency_errors (bool b = 1) [inline]`

decide whether or not to print inconsistency errors

Definition at line 158 of file IO_HERWIG.h.

8.26.3.6 `bool HepMC::IO_HERWIG::no_gaps_in_barcodes () const [inline]`

ask how to deal with extra non-physical pseudo particles

Definition at line 75 of file IO_HERWIG.h.

8.26.3.7 `void HepMC::IO_HERWIG::set_no_gaps_in_barcodes (bool a)`
`[inline]`

The HERWIG HEPEVT common block has some EXTRA non-physical ENTRIES (such as CMS frame, HARD subprocess, and CONE). These are removed by **IO_HERWIG** (p. 176). Thus the **HepMC** (p. 19) event will APPEAR to have fewer particles in it that herwig did. There is a switch `m_no_gaps_in_barcodes`. For true - then the extra particles are removed from HEPEVT, with the result that the **HepMC** (p. 19) barcodes will be sequential, with no gaps. false - the barcodes will correspond directly to the HEPEVT index, but there will be gaps ... ie some barcodes will be unassigned. this switch requested by I Hinchliffe, October 31, 2002

Definition at line 88 of file IO_HERWIG.h.

8.26.3.8 `bool HepMC::IO_HERWIG::trust_both_mothers_and_daughters ()`
`const [inline, protected]`

default is true

Definition at line 143 of file IO_HERWIG.h.

8.26.3.9 `bool HepMC::IO_HERWIG::trust_mothers_before_daughters () const`
`[inline, protected]`

default is false

Definition at line 146 of file IO_HERWIG.h.

8.26.3.10 `void HepMC::IO_HERWIG::set_trust_mothers_before_daughters`
`(bool b = 1) [inline, protected]`

define mother daughter trust rules

Definition at line 155 of file IO_HERWIG.h.

8.26.3.11 `void HepMC::IO_HERWIG::set_trust_both_mothers_and_daughters`
`(bool b = 0) [inline, protected]`

define mother daughter trust rules

Definition at line 152 of file IO_HERWIG.h.

8.26.3.12 `GenParticle * HepMC::IO_HERWIG::build_particle (int index)`
`[protected]`

make a particle

Builds a particle object corresponding to index in HEPEVT

Definition at line 367 of file IO_HERWIG.cc.

References HepMC::HEPEVT_Wrapper::e(), HepMC::HEPEVT_Wrapper::id(), HepMC::HEPEVT_Wrapper::m(), p, HepMC::HEPEVT_Wrapper::px(), HepMC::HEPEVT_Wrapper::py(), HepMC::HEPEVT_Wrapper::pz(), and HepMC::HEPEVT_Wrapper::status().

Referenced by fill_next_event().

8.26.3.13 void HepMC::IO_HERWIG::build_production_vertex (int *i*,
std::vector< GenParticle * > & *hepevt_particle*, GenEvent * *evt*)
[protected]

make a production vertex

for particle in HEPEVT with index *i*, build a production vertex if appropriate, and add that vertex to the event

Definition at line 226 of file IO_HERWIG.cc.

References HepMC::GenVertex::add_particle_in(), HepMC::GenVertex::add_particle_out(), HepMC::GenEvent::add_vertex(), HepMC::HEPEVT_Wrapper::event_number(), HepMC::HEPEVT_Wrapper::first_parent(), HepMC::HEPEVT_Wrapper::last_parent(), HepMC::HEPEVT_Wrapper::number_parents(), p, HepMC::GenVertex::position(), HepMC::GenVertex::print(), HepMC::GenVertex::set_position(), HepMC::HEPEVT_Wrapper::t(), HepMC::HEPEVT_Wrapper::x(), HepMC::HEPEVT_Wrapper::y(), and HepMC::HEPEVT_Wrapper::z().

Referenced by build_end_vertex(), and fill_next_event().

8.26.3.14 void HepMC::IO_HERWIG::build_end_vertex (int *i*, std::vector<
GenParticle * > & *hepevt_particle*, GenEvent * *evt*) [protected]

make a decay vertex

for particle in HEPEVT with index *i*, build an end vertex if appropriate, and add that vertex to the event

Definition at line 299 of file IO_HERWIG.cc.

References HepMC::GenVertex::add_particle_in(), HepMC::GenVertex::add_particle_out(), HepMC::GenEvent::add_vertex(), build_production_vertex(), HepMC::HEPEVT_Wrapper::event_number(), HepMC::HEPEVT_Wrapper::first_child(), HepMC::HEPEVT_Wrapper::last_child(), HepMC::HEPEVT_Wrapper::number_children(), p, HepMC::GenVertex::position(), HepMC::GenVertex::set_position(), HepMC::HEPEVT_Wrapper::t(), HepMC::HEPEVT_Wrapper::x(), HepMC::HEPEVT_Wrapper::y(), and HepMC::HEPEVT_Wrapper::z().

Referenced by fill_next_event().

8.26.3.15 int HepMC::IO_HERWIG::find_in_map (const std::map< GenParticle
*, int > & *m*, GenParticle * *p*) const [protected]

find this particle in the map

Definition at line 382 of file IO_HERWIG.cc.

References p.

8.26.3.16 void HepMC::IO_HERWIG::repair_hepevt () const [protected]

make the HERWIG HEPEVT common block look like the standard

This routine takes the HEPEVT common block as used in HERWIG, and converts it into the HEPEVT common block in the standard format

This means it:

- removes the color structure, which herwig overloads into the mother/daughter fields
- zeros extra entries for hard subprocess, etc.

Special HERWIG status codes 101,102 colliding beam particles 103 beam-beam collision CMS vector 120 hard subprocess CMS vector 121,122 hard subprocess colliding partons 123-129 hard subprocess outgoing particles 141-149 (ID=94) mirror image of hard subprocess particles 100 (ID=0 cone)

Special HERWIG particle id's 91 clusters 94 jets 0 others with no pdg code

Definition at line 389 of file IO_HERWIG.cc.

References HepMC::HEPEVT_Wrapper::first_child(), HepMC::HEPEVT_Wrapper::first_parent(), HepMC::HEPEVT_Wrapper::id(), HepMC::HEPEVT_Wrapper::last_child(), HepMC::HEPEVT_Wrapper::last_parent(), HepMC::HEPEVT_Wrapper::number_entries(), remove_gaps_in_hepevt(), HepMC::HEPEVT_Wrapper::set_children(), HepMC::HEPEVT_Wrapper::set_id(), HepMC::HEPEVT_Wrapper::set_parents(), HepMC::HEPEVT_Wrapper::status(), translate_herwig_to_pdg_id(), and zero_hepevt_entry().

Referenced by fill_next_event().

8.26.3.17 void HepMC::IO_HERWIG::remove_gaps_in_hepevt () const [protected]

deal with artifacts of repairing HEPEVT

in this scenario, we do not allow there to be zero-ed entries in the HEPEVT common block, and so be reshuffle the common block, removing the zero-ed entries as we go and making sure we keep the mother/daughter relationships appropriate

Definition at line 673 of file IO_HERWIG.cc.

References HepMC::HEPEVT_Wrapper::e(), HepMC::HEPEVT_Wrapper::first_child(), HepMC::HEPEVT_Wrapper::first_parent(), HepMC::HEPEVT_Wrapper::id(), HepMC::HEPEVT_Wrapper::last_child(), HepMC::HEPEVT_Wrapper::last_parent(), HepMC::HEPEVT_Wrapper::m(), HepMC::HEPEVT_Wrapper::number_entries(), HepMC::HEPEVT_Wrapper::px(), HepMC::HEPEVT_Wrapper::py(), HepMC::HEPEVT_Wrapper::pz(), HepMC::HEPEVT_Wrapper::set_children(), HepMC::HEPEVT_Wrapper::set_id(), HepMC::HEPEVT_Wrapper::set_mass(), HepMC::HEPEVT_Wrapper::set_momentum(), HepMC::HEPEVT_Wrapper::set_number_entries(), HepMC::HEPEVT_Wrapper::set_parents(), HepMC::HEPEVT_Wrapper::set_position(), HepMC::HEPEVT_Wrapper::set_status(), HepMC::HEPEVT_Wrapper::status(), HepMC::HEPEVT_Wrapper::t(), HepMC::HEPEVT_Wrapper::x(), HepMC::HEPEVT_Wrapper::y(), and HepMC::HEPEVT_Wrapper::z().

Referenced by repair_hepevt().

8.26.3.18 `void HepMC::IO_HERWIG::zero_hepevt_entry (int i) const`
[protected]

zero out a HEPEVT pseudo particle

Definition at line 733 of file IO_HERWIG.cc.

References HepMC::HEPEVT_Wrapper::max_number_entries(), HepMC::HEPEVT_Wrapper::set_children(), HepMC::HEPEVT_Wrapper::set_id(), HepMC::HEPEVT_Wrapper::set_mass(), HepMC::HEPEVT_Wrapper::set_momentum(), HepMC::HEPEVT_Wrapper::set_parents(), HepMC::HEPEVT_Wrapper::set_position(), and HepMC::HEPEVT_Wrapper::set_status().

Referenced by repair_hepevt().

8.26.3.19 `int HepMC::IO_HERWIG::translate_herwig_to_pdg_id (int i) const`
[protected]

translate particle ID

This routine is copied from Lynn Garren's stdhep 5.01. see <http://cepa.fnal.gov/psm/stdhep/>

Definition at line 744 of file IO_HERWIG.cc.

Referenced by repair_hepevt().

The documentation for this class was generated from the following files:

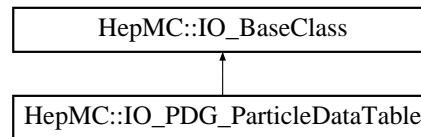
- IO_HERWIG.h
- IO_HERWIG.cc

8.27 HepMC::IO_PDG_ParticleDataTable Class Reference

an example **ParticleDataTable** (p. 213) IO method

```
#include <IO_PDG_ParticleDataTable.h>
```

Inheritance diagram for HepMC::IO_PDG_ParticleDataTable::



Public Member Functions

- **IO_PDG_ParticleDataTable** (const char *filename="PDG98_ParticleData-Table.txt")
constructor using filename
- virtual **~IO_PDG_ParticleDataTable** ()
- bool **fill_particle_data_table** (**ParticleDataTable** *)
read the input and fill the table
- void **add_quarks_to_table** (**ParticleDataTable** &)
add u, d, s, c, b, and t
- void **print** (std::ostream &ostr=std::cout) const
write to ostr
- int **rdstate** () const
check the IO state

Protected Member Functions

- bool **search_for_key_end** (std::istream &in, const char *key)
for internal use
- void **read_entry** (**ParticleDataTable** *)
read a line

8.27.1 Detailed Description

an example **ParticleDataTable** (p. 213) IO method

Example of reading from file PDG98_ParticleDataTable.txt

Definition at line 49 of file IO_PDG_ParticleDataTable.h.

8.27.2 Constructor & Destructor Documentation

8.27.2.1 `HepMC::IO_PDG_ParticleDataTable::IO_PDG_ParticleDataTable` (`const char * filename = "PDG98_ParticleDataTable.txt"`)

constructor using filename

Definition at line 17 of file `IO_PDG_ParticleDataTable.cc`.

8.27.2.2 `HepMC::IO_PDG_ParticleDataTable::~~IO_PDG_ParticleDataTable` () [virtual]

Definition at line 25 of file `IO_PDG_ParticleDataTable.cc`.

8.27.3 Member Function Documentation

8.27.3.1 `bool HepMC::IO_PDG_ParticleDataTable::fill_particle_data_table` (`ParticleDataTable *`) [virtual]

read the input and fill the table

Implements `HepMC::IO_BaseClass` (p.153).

Definition at line 29 of file `IO_PDG_ParticleDataTable.cc`.

References `read_entry()`, `search_for_key_end()`, and `HepMC::ParticleDataTable::set_description()`.

8.27.3.2 `void HepMC::IO_PDG_ParticleDataTable::add_quarks_to_table` (`ParticleDataTable &`)

add u, d, s, c, b, and t

since quarks aren't included in PDG table, this method adds them

Definition at line 170 of file `IO_PDG_ParticleDataTable.cc`.

References `HepMC::ParticleDataTable::erase()`, `HepMC::ParticleDataTable::find()`, `HepMC::ParticleDataTable::insert()`, and `HepMC::ParticleData::mass()`.

8.27.3.3 `void HepMC::IO_PDG_ParticleDataTable::print` (`std::ostream & ostr = std::cout`) `const` [inline, virtual]

write to ostr

Reimplemented from `HepMC::IO_BaseClass` (p.153).

Definition at line 85 of file `IO_PDG_ParticleDataTable.h`.

8.27.3.4 `int HepMC::IO_PDG_ParticleDataTable::rdstate` () `const` [inline]

check the IO state

Definition at line 63 of file `IO_PDG_ParticleDataTable.h`.

8.27.3.5 `bool HepMC::IO_PDG_ParticleDataTable::search_for_key_end` `(std::istream & in, const char * key)` [protected]

for internal use

(this method borrowed from `IO_Ascii` (p.142) class) reads characters from `in` until the string of characters matching `key` is found (success) or EOF is reached (failure). It stops immediately thereafter. Returns T/F for success/fail

Definition at line 208 of file `IO_PDG_ParticleDataTable.cc`.

Referenced by `fill_particle_data_table()`.

8.27.3.6 `void HepMC::IO_PDG_ParticleDataTable::read_entry` `(ParticleDataTable *)` [protected]

read a line

Definition at line 68 of file `IO_PDG_ParticleDataTable.cc`.

References `HepMC::lifetime_from_width()`, `HepMC::ParticleDataTable::find()`, `HepMC::ParticleDataTable::insert()`, `HepMC::ParticleData::set_clifetime()`, and `HepMC::ParticleData::set_mass()`.

Referenced by `fill_particle_data_table()`.

The documentation for this class was generated from the following files:

- `IO_PDG_ParticleDataTable.h`
- `IO_PDG_ParticleDataTable.cc`

8.28 HepMC::detail::is_arithmetic< T > Struct Template Reference

undefined and therefore non-arithmetic

```
#include <is_arithmetic.h>
```

Static Public Attributes

- static bool const **value** = false

8.28.1 Detailed Description

```
template<class T> struct HepMC::detail::is_arithmetic< T >
```

undefined and therefore non-arithmetic

Definition at line 22 of file is_arithmetic.h.

8.28.2 Member Data Documentation

8.28.2.1 `template<class T> bool const HepMC::detail::is_arithmetic< T >::value = false [static]`

Definition at line 24 of file is_arithmetic.h.

The documentation for this struct was generated from the following file:

- is_arithmetic.h

8.29 HepMC::detail::is_arithmetic< char > Struct Template Reference

character is arithmetic

```
#include <is_arithmetic.h>
```

Static Public Attributes

- static bool const **value** = true

8.29.1 Detailed Description

```
template<> struct HepMC::detail::is_arithmetic< char >
```

character is arithmetic

Definition at line 29 of file is_arithmetic.h.

8.29.2 Member Data Documentation

8.29.2.1 bool const HepMC::detail::is_arithmetic< char >::value = true [static]

Definition at line 30 of file is_arithmetic.h.

The documentation for this struct was generated from the following file:

- is_arithmetic.h

8.30 HepMC::detail::is_arithmetic< double > Struct Template Reference

double is arithmetic

```
#include <is_arithmetic.h>
```

Static Public Attributes

- static bool const **value** = true

8.30.1 Detailed Description

```
template<> struct HepMC::detail::is_arithmetic< double >
```

double is arithmetic

Definition at line 79 of file is_arithmetic.h.

8.30.2 Member Data Documentation

8.30.2.1 `bool const HepMC::detail::is_arithmetic< double >::value = true`
 [static]

Definition at line 80 of file is_arithmetic.h.

The documentation for this struct was generated from the following file:

- is_arithmetic.h

8.31 HepMC::detail::is_arithmetic< float > Struct Template Reference

float is arithmetic

```
#include <is_arithmetic.h>
```

Static Public Attributes

- static bool const **value** = true

8.31.1 Detailed Description

```
template<> struct HepMC::detail::is_arithmetic< float >
```

float is arithmetic

Definition at line 74 of file is_arithmetic.h.

8.31.2 Member Data Documentation

8.31.2.1 bool const HepMC::detail::is_arithmetic< float >::value = true [static]

Definition at line 75 of file is_arithmetic.h.

The documentation for this struct was generated from the following file:

- is_arithmetic.h

8.32 HepMC::detail::is_arithmetic< int > Struct Template Reference

int is arithmetic

```
#include <is_arithmetic.h>
```

Static Public Attributes

- static bool const **value** = true

8.32.1 Detailed Description

```
template<> struct HepMC::detail::is_arithmetic< int >
```

int is arithmetic

Definition at line 54 of file is_arithmetic.h.

8.32.2 Member Data Documentation

8.32.2.1 bool const HepMC::detail::is_arithmetic< int >::value = true [static]

Definition at line 55 of file is_arithmetic.h.

The documentation for this struct was generated from the following file:

- is_arithmetic.h

8.33 HepMC::detail::is_arithmetic< long > Struct Template Reference

long is arithmetic

```
#include <is_arithmetic.h>
```

Static Public Attributes

- static bool const **value** = true

8.33.1 Detailed Description

```
template<> struct HepMC::detail::is_arithmetic< long >
```

long is arithmetic

Definition at line 64 of file is_arithmetic.h.

8.33.2 Member Data Documentation

8.33.2.1 bool const HepMC::detail::is_arithmetic< long >::value = true [static]

Definition at line 65 of file is_arithmetic.h.

The documentation for this struct was generated from the following file:

- is_arithmetic.h

8.34 HepMC::detail::is_arithmetic< long double > Struct Template Reference

long double is arithmetic

```
#include <is_arithmetic.h>
```

Static Public Attributes

- static bool const **value** = true

8.34.1 Detailed Description

```
template<> struct HepMC::detail::is_arithmetic< long double >
```

long double is arithmetic

Definition at line 84 of file is_arithmetic.h.

8.34.2 Member Data Documentation

8.34.2.1 `bool const HepMC::detail::is_arithmetic< long double >::value = true`
 [static]

Definition at line 85 of file is_arithmetic.h.

The documentation for this struct was generated from the following file:

- is_arithmetic.h

8.35 HepMC::detail::is_arithmetic< short > Struct Template Reference

short is arithmetic

```
#include <is_arithmetic.h>
```

Static Public Attributes

- static bool const **value** = true

8.35.1 Detailed Description

```
template<> struct HepMC::detail::is_arithmetic< short >
```

short is arithmetic

Definition at line 44 of file is_arithmetic.h.

8.35.2 Member Data Documentation

8.35.2.1 bool const HepMC::detail::is_arithmetic< short >::value = true [static]

Definition at line 45 of file is_arithmetic.h.

The documentation for this struct was generated from the following file:

- is_arithmetic.h

8.36 HepMC::detail::is_arithmetic< signed char > Struct Template Reference

signed character is arithmetic

```
#include <is_arithmetic.h>
```

Static Public Attributes

- static bool const **value** = true

8.36.1 Detailed Description

```
template<> struct HepMC::detail::is_arithmetic< signed char >
```

signed character is arithmetic

Definition at line 39 of file is_arithmetic.h.

8.36.2 Member Data Documentation

8.36.2.1 `bool const HepMC::detail::is_arithmetic< signed char >::value = true`
 [static]

Definition at line 40 of file is_arithmetic.h.

The documentation for this struct was generated from the following file:

- is_arithmetic.h

8.37 HepMC::detail::is_arithmetic< unsigned char > Struct Template Reference

unsigned character is arithmetic

```
#include <is_arithmetic.h>
```

Static Public Attributes

- static bool const **value** = true

8.37.1 Detailed Description

```
template<> struct HepMC::detail::is_arithmetic< unsigned char >
```

unsigned character is arithmetic

Definition at line 34 of file is_arithmetic.h.

8.37.2 Member Data Documentation

8.37.2.1 `bool const HepMC::detail::is_arithmetic< unsigned char >::value = true`
 [static]

Definition at line 35 of file is_arithmetic.h.

The documentation for this struct was generated from the following file:

- is_arithmetic.h

8.38 HepMC::detail::is_arithmetic< unsigned int > Struct Template Reference

unsigned int is arithmetic

```
#include <is_arithmetic.h>
```

Static Public Attributes

- static bool const **value** = true

8.38.1 Detailed Description

```
template<> struct HepMC::detail::is_arithmetic< unsigned int >
```

unsigned int is arithmetic

Definition at line 59 of file is_arithmetic.h.

8.38.2 Member Data Documentation

8.38.2.1 `bool const HepMC::detail::is_arithmetic< unsigned int >::value = true`
 [static]

Definition at line 60 of file is_arithmetic.h.

The documentation for this struct was generated from the following file:

- is_arithmetic.h

8.39 HepMC::detail::is_arithmetic< unsigned long > Struct Template Reference

unsigned long is arithmetic

```
#include <is_arithmetic.h>
```

Static Public Attributes

- static bool const **value** = true

8.39.1 Detailed Description

template<> struct HepMC::detail::is_arithmetic< unsigned long >

unsigned long is arithmetic

Definition at line 69 of file is_arithmetic.h.

8.39.2 Member Data Documentation

8.39.2.1 bool const HepMC::detail::is_arithmetic< unsigned long >::value = true
[static]

Definition at line 70 of file is_arithmetic.h.

The documentation for this struct was generated from the following file:

- **is_arithmetic.h**

8.40 HepMC::detail::is_arithmetic< unsigned short > Struct Template Reference

unsigned short is arithmetic

```
#include <is_arithmetic.h>
```

Static Public Attributes

- static bool const **value** = true

8.40.1 Detailed Description

```
template<> struct HepMC::detail::is_arithmetic< unsigned short >
```

unsigned short is arithmetic

Definition at line 49 of file is_arithmetic.h.

8.40.2 Member Data Documentation

8.40.2.1 `bool const HepMC::detail::is_arithmetic< unsigned short >::value = true`
 [static]

Definition at line 50 of file is_arithmetic.h.

The documentation for this struct was generated from the following file:

- is_arithmetic.h

8.41 IsEventGood Class Reference

example class

Public Member Functions

- **bool operator()** (const **HepMC::GenEvent** *evt)
check this event for goodness

8.41.1 Detailed Description

example class

event selection predicate. returns true if the event contains a photon with $p_T > 50$ GeV

Examples:

example_EventSelection.cc.

Definition at line 20 of file example_EventSelection.cc.

8.41.2 Member Function Documentation

8.41.2.1 **bool IsEventGood::operator()** (const **HepMC::GenEvent** * *evt*) [inline]

check this event for goodness

Examples:

example_EventSelection.cc.

Definition at line 23 of file example_EventSelection.cc.

References `p`, `HepMC::GenEvent::particles_begin()`, and `HepMC::GenEvent::particles_end()`.

The documentation for this class was generated from the following file:

- **example_EventSelection.cc**

8.42 IsFinalState Class Reference

test class

```
#include <testHepMCIteration.h>
```

Public Member Functions

- **bool operator()** (const **HepMC::GenParticle** *p)
returns true if the GenParticle does not decay

8.42.1 Detailed Description

test class

this predicate returns true if the input has no decay vertex

Definition at line 26 of file testHepMCIteration.h.

8.42.2 Member Function Documentation

8.42.2.1 **bool IsFinalState::operator()** (const **HepMC::GenParticle** * p) [inline]

returns true if the GenParticle does not decay

Definition at line 29 of file testHepMCIteration.h.

References p.

The documentation for this class was generated from the following file:

- **testHepMCIteration.h**

8.43 IsGoodEvent Class Reference

```
#include <IsGoodEvent.h>
```

Public Member Functions

- **bool operator()** (const **HepMC::GenEvent** **evt*)

8.43.1 Detailed Description

event selection predicate. returns true if the event contains a photon with $p_T > 50$ GeV

Definition at line 12 of file IsGoodEvent.h.

8.43.2 Member Function Documentation

8.43.2.1 **bool IsGoodEvent::operator()** (const **HepMC::GenEvent** * *evt*) [inline]

Definition at line 14 of file IsGoodEvent.h.

References `p`, `HepMC::GenEvent::particles_begin()`, and `HepMC::GenEvent::particles_end()`.

The documentation for this class was generated from the following file:

- **IsGoodEvent.h**

8.44 IsGoodEventMyPythia Class Reference

example class

Public Member Functions

- **bool operator()** (const **HepMC::GenEvent** *evt)
returns true if event is "good"

8.44.1 Detailed Description

example class

event selection predicate. returns true if the event contains a photon with $p_T > 25$ GeV

Examples:

example_MyPythiaWithEventSelection.cc.

Definition at line 28 of file example_MyPythiaWithEventSelection.cc.

8.44.2 Member Function Documentation

- 8.44.2.1** **bool IsGoodEventMyPythia::operator()** (const **HepMC::GenEvent** * *evt*)
[inline]

returns true if event is "good"

Examples:

example_MyPythiaWithEventSelection.cc.

Definition at line 31 of file example_MyPythiaWithEventSelection.cc.

References `p`, `HepMC::GenEvent::particles_begin()`, and `HepMC::GenEvent::particles_end()`.

The documentation for this class was generated from the following file:

- **example_MyPythiaWithEventSelection.cc**

8.45 IsPhoton Class Reference

example class

Public Member Functions

- **bool operator()** (const **HepMC::GenParticle** *p)
returns true if the GenParticle is a photon with more than 10 GeV transverse momentum

8.45.1 Detailed Description

example class

this predicate returns true if the input particle is a photon in the central region ($\eta < 2.5$) with $p_T > 10$ GeV

Examples:

example_UsingIterators.cc.

Definition at line 20 of file example_UsingIterators.cc.

8.45.2 Member Function Documentation

8.45.2.1 bool IsPhoton::operator() (const HepMC::GenParticle * p) [inline]

returns true if the GenParticle is a photon with more than 10 GeV transverse momentum

Examples:

example_UsingIterators.cc.

Definition at line 23 of file example_UsingIterators.cc.

References p.

The documentation for this class was generated from the following file:

- **example_UsingIterators.cc**

8.46 IsStateFinal Class Reference

example class

Public Member Functions

- **bool operator()** (const **HepMC::GenParticle** *p)
returns true if the GenParticle does not decay

8.46.1 Detailed Description

example class

this predicate returns true if the input has no decay vertex

Examples:

example_UsingIterators.cc.

Definition at line 47 of file example_UsingIterators.cc.

8.46.2 Member Function Documentation

8.46.2.1 **bool IsStateFinal::operator()** (const **HepMC::GenParticle** * p) [inline]

returns true if the GenParticle does not decay

Examples:

example_UsingIterators.cc.

Definition at line 50 of file example_UsingIterators.cc.

References p.

The documentation for this class was generated from the following file:

- **example_UsingIterators.cc**

8.47 IsW_Boson Class Reference

example class

Public Member Functions

- **bool operator()** (const **HepMC::GenParticle** *p)
returns true if the GenParticle is a W

8.47.1 Detailed Description

example class

this predicate returns true if the input particle is a W⁺/W⁻

Examples:

example_UsingIterators.cc.

Definition at line 34 of file example_UsingIterators.cc.

8.47.2 Member Function Documentation

8.47.2.1 bool IsW_Boson::operator() (const **HepMC::GenParticle** * p) [inline]

returns true if the GenParticle is a W

Examples:

example_UsingIterators.cc.

Definition at line 37 of file example_UsingIterators.cc.

References p.

The documentation for this class was generated from the following file:

- **example_UsingIterators.cc**

8.48 HepMC::ParticleData Class Reference

an example **ParticleData** (p.206) class

```
#include <ParticleData.h>
```

Public Member Functions

- **ParticleData** (std::string name, int id, double charge, double mass=0, double cLifetime=-1, double spin=0)
constructor requiring name, ID, and charge
- **ParticleData** (const char *name, int id, double charge, double mass=0, double cLifetime=-1, double spin=0)
constructor requiring name, ID, and charge
- virtual **~ParticleData** ()
- bool **operator==** (const **ParticleData** &) const
equality
- bool **operator!=** (const **ParticleData** &) const
inequality
- void **print** (std::ostream &ostr=std::cout) const
write particle data information to ostr
- bool **is_lepton** () const
true if charged lepton /neutrino
- bool **is_charged_lepton** () const
true if a charged lepton
- bool **is_em** () const
true if an electron or photon
- bool **is_neutrino** () const
true if a neutrino
- bool **is_hadron** () const
true if a hadron
- bool **is_boson** () const
true if a gauge or higgs boson
- std::string **name** () const
description of the particle according to PDG, i.e. "Delta(1900) S_31"
- int **pdg_id** () const
PDG ID number.

- double **charge** () const
charge
- double **mass** () const
nominal mass
- double **width** () const
width as calculated from lifetime
- double **clifetime** () const
lifetime in mm
- double **spin** () const
J spin.
- void **set_charge** (double)
set charge
- void **set_mass** (double)
set nominal mass
- void **set_width** (double)
set width
- void **set_clifetime** (double)
set lifetime in mm
- void **set_spin** (double)
set J spin

Protected Member Functions

- int **model_independent_pdg_id** () const
omits susy/excited/technicolor digit from returned ID

Static Protected Member Functions

- static unsigned int **counter** ()
*num **ParticleData** (p. 206) objects in memory*

Friends

- std::ostream & **operator<<** (std::ostream &, const **ParticleData** &)
write to ostr

8.48.1 Detailed Description

an example **ParticleData** (p. 206) class

Particle Data common to all particles of a given PDG id

Examples:

`example_BuildEventFromScratch.cc.`

Definition at line 69 of file ParticleData.h.

8.48.2 Constructor & Destructor Documentation

8.48.2.1 HepMC::ParticleData::ParticleData (std::string *name*, int *id*, double *charge*, double *mass* = 0, double *cLifetime* = -1, double *spin* = 0)

constructor requiring name, ID, and charge

Units ID: defined by PDG group (particles are +ve, antipart are -ve) also consistent with the Pythia definitions charge: fraction of proton charge mass cLifetime: c*time Default mass=0 and cLifetime is -1 which means stable (width= 0.) These defaults exist because many very basic MC generators may produce only massless stable particles in the event record.

Definition at line 12 of file ParticleData.cc.

References `set_charge()`, and `set_spin()`.

8.48.2.2 HepMC::ParticleData::ParticleData (const char * *name*, int *id*, double *charge*, double *mass* = 0, double *cLifetime* = -1, double *spin* = 0)

constructor requiring name, ID, and charge

note, this constructor is redundant to the one above, i.e. one could use: `new HepMC::ParticleData (p. 206)(string("electron"),11,-1,0.000511,-1,5);` but we keep it because it is convenient.

Definition at line 30 of file ParticleData.cc.

References `set_charge()`, and `set_spin()`.

8.48.2.3 HepMC::ParticleData::~~ParticleData () [virtual]

Definition at line 44 of file ParticleData.cc.

8.48.3 Member Function Documentation

8.48.3.1 bool HepMC::ParticleData::operator==(const ParticleData &) const [inline]

equality

Definition at line 213 of file ParticleData.h.

References `m_2spin`, `m_3charge`, `m_clifetime`, `m_mass`, and `m_pdg_id`.

8.48.3.2 `bool HepMC::ParticleData::operator!= (const ParticleData &) const`
[inline]

inequality

Definition at line 222 of file ParticleData.h.

References `pdg_id()`.

8.48.3.3 `void HepMC::ParticleData::print (std::ostream & ostr = std::cout) const`

write particle data information to ostr

Definition at line 48 of file ParticleData.cc.

References `charge()`, `clifetime()`, `mass()`, `name()`, `pdg_id()`, and `spin()`.

8.48.3.4 `bool HepMC::ParticleData::is_lepton () const` [inline]

true if charged lepton /neutrino

true if a charged lepton or neutrino -> | 11,13,15,12,14,16,17,18 |

Definition at line 142 of file ParticleData.h.

References `pdg_id()`.

Referenced by `is_charged_lepton()`, and `is_neutrino()`.

8.48.3.5 `bool HepMC::ParticleData::is_charged_lepton () const` [inline]

true if a charged lepton

true if a charged lepton -> | 11,13,15 |

Definition at line 146 of file ParticleData.h.

References `is_lepton()`, and `pdg_id()`.

8.48.3.6 `bool HepMC::ParticleData::is_em () const` [inline]

true if an electron or photon

true if an electron or photon -> | 11, 22 |

Definition at line 154 of file ParticleData.h.

References `pdg_id()`.

8.48.3.7 `bool HepMC::ParticleData::is_neutrino () const` [inline]

true if a neutrino

true if a neutrino -> | 12,14,16 |

Definition at line 150 of file ParticleData.h.

References `is_lepton()`, and `pdg_id()`.

8.48.3.8 bool HepMC::ParticleData::is_hadron () const [inline]

true if a hadron

true if a hadron -> q,g,meson,baryon

Definition at line 158 of file ParticleData.h.

References `pdg_id()`.

8.48.3.9 bool HepMC::ParticleData::is_boson () const [inline]

true if a gauge or higgs boson

true if a gauge or higgs boson -> | 9, 21-39 |

Definition at line 163 of file ParticleData.h.

References `pdg_id()`.

8.48.3.10 std::string HepMC::ParticleData::name () const [inline]

description of the particle according to PDG, i.e. "Delta(1900) S_31"

Definition at line 173 of file ParticleData.h.

Referenced by `HepMC::operator<<()`, `print()`, `HepMC::IO_GenEvent::write_particle_data()`, `HepMC::IO_ExtendedAscii::write_particle_data()`, and `HepMC::IO_Ascii::write_particle_data()`.

8.48.3.11 int HepMC::ParticleData::pdg_id () const [inline]

PDG ID number.

Definition at line 174 of file ParticleData.h.

Referenced by `HepMC::ParticleDataTable::erase()`, `HepMC::ParticleDataTable::insert()`, `is_boson()`, `is_charged_lepton()`, `is_em()`, `is_hadron()`, `is_lepton()`, `is_neutrino()`, `operator!=()`, `HepMC::operator<<()`, `print()`, `HepMC::IO_GenEvent::write_particle_data()`, `HepMC::IO_ExtendedAscii::write_particle_data()`, and `HepMC::IO_Ascii::write_particle_data()`.

8.48.3.12 double HepMC::ParticleData::charge () const [inline]

charge

Definition at line 175 of file ParticleData.h.

Referenced by `HepMC::operator<<()`, `print()`, `HepMC::IO_GenEvent::write_particle_data()`, `HepMC::IO_ExtendedAscii::write_particle_data()`, and `HepMC::IO_Ascii::write_particle_data()`.

8.48.3.13 double HepMC::ParticleData::mass () const [inline]

nominal mass

Definition at line 178 of file ParticleData.h.

Referenced by HepMC::IO_PDG_ParticleDataTable::add_quarks_to_table(), HepMC::operator<<(), print(), HepMC::IO_GenEvent::write_particle_data(), HepMC::IO_ExtendedAscii::write_particle_data(), and HepMC::IO_Ascii::write_particle_data().

8.48.3.14 double HepMC::ParticleData::width () const

width as calculated from clifetime

Definition at line 71 of file ParticleData.cc.

References HepMC::HepMC_hbarc.

8.48.3.15 double HepMC::ParticleData::clifetime () const [inline]

lifetime in mm

Definition at line 179 of file ParticleData.h.

Referenced by HepMC::operator<<(), print(), HepMC::IO_GenEvent::write_particle_data(), HepMC::IO_ExtendedAscii::write_particle_data(), and HepMC::IO_Ascii::write_particle_data().

8.48.3.16 double HepMC::ParticleData::spin () const [inline]

J spin.

Definition at line 180 of file ParticleData.h.

Referenced by HepMC::operator<<(), print(), HepMC::IO_GenEvent::write_particle_data(), HepMC::IO_ExtendedAscii::write_particle_data(), and HepMC::IO_Ascii::write_particle_data().

8.48.3.17 void HepMC::ParticleData::set_charge (double) [inline]

set charge

Definition at line 181 of file ParticleData.h.

Referenced by ParticleData().

8.48.3.18 void HepMC::ParticleData::set_mass (double) [inline]

set nominal mass

Definition at line 190 of file ParticleData.h.

Referenced by HepMC::IO_PDG_ParticleDataTable::read_entry().

8.48.3.19 void HepMC::ParticleData::set_width (double) [inline]

set width

Definition at line 193 of file ParticleData.h.

References HepMC::HepMC_hbarc.

8.48.3.20 void HepMC::ParticleData::set_clifetime (double) [inline]

set lifetime in mm

Definition at line 202 of file ParticleData.h.

Referenced by HepMC::IO_PDG_ParticleDataTable::read_entry().

8.48.3.21 void HepMC::ParticleData::set_spin (double) [inline]

set J spin

Definition at line 205 of file ParticleData.h.

Referenced by ParticleData().

8.48.3.22 unsigned int HepMC::ParticleData::counter () [static, protected]

num **ParticleData** (p.206) objects in memory

Definition at line 86 of file ParticleData.cc.

8.48.3.23 int HepMC::ParticleData::model_independent_pdg_id_ () const [protected]

omits susy/excited/technicolor digit from returned ID

returns the particle id with the seventh digit removed for susy/excited/technicolor particles. Thus an excited electron (40000011) would be returned as 11 Useful only internally for sorting particles!

Definition at line 61 of file ParticleData.cc.

8.48.4 Friends And Related Function Documentation**8.48.4.1 std::ostream& operator<< (std::ostream & *ostr*, const ParticleData & *pdata*) [friend]**

write to ostr

Definition at line 94 of file ParticleData.cc.

The documentation for this class was generated from the following files:

- **ParticleData.h**
- **ParticleData.cc**

8.49 HepMC::ParticleDataTable Class Reference

an example **ParticleDataTable** (p. 213) class

```
#include <ParticleDataTable.h>
```

Public Types

- typedef std::map< int, **HepMC::ParticleData** * >::iterator **iterator**
*iterator for **ParticleData** (p. 206) map*
- typedef std::map< int, **HepMC::ParticleData** * >::const_iterator **const_iterator**
*const iterator for **ParticleData** (p. 206) map*

Public Member Functions

- **ParticleDataTable** (std::string description=std::string())
constructor with optional description
- **ParticleDataTable** (const char description)
constructor with description
- **ParticleDataTable** (const **ParticleDataTable** &)
copy constructor
- virtual ~**ParticleDataTable** ()
*Shallow: does not delete **ParticleData** (p. 206) entries.*
- **ParticleDataTable** & **operator=** (const **ParticleDataTable** &)
shallow: does not copy the entries, only makes new pointers
- void **make_antiparticles_from_particles** ()
make corresponding anti-particles for all particles in table
- int **merge_table** (const **ParticleDataTable** &)
merge two tables
- void **print** (std::ostream &ostr=std::cout) const
write the table to ostr
- void **delete_all** ()
*delete all **ParticleData** (p. 206) instances in this table*
- void **clear** ()
clears table without deleting
- **ParticleData** * **operator[]** (int id) const
*return pointer to requested **ParticleData** (p. 206)*

- **ParticleData * find** (int id) const
*return pointer to requested **ParticleData** (p. 206)*
- **int size** () const
size of table
- **bool empty** () const
true if the table is empty
- **bool insert** (**ParticleData** *)
true if successful
- **bool erase** (**ParticleData** *)
removes from table - does not delete
- **bool erase** (int id)
removes from table - does not delete
- **iterator begin** ()
begin iteration
- **iterator end** ()
end iteration
- **const_iterator begin** () const
begin const iteration
- **const_iterator end** () const
end const iteration
- **std::string description** () const
table description
- **void set_description** (std::string)
set table description
- **void set_description** (const char)
set table description

8.49.1 Detailed Description

an example **ParticleDataTable** (p. 213) class

Example container for **ParticleData** (p. 206) instances. Basically just an interface to STL map.

Examples:

example_BuildEventFromScratch.cc.

Definition at line 35 of file ParticleDataTable.h.

8.49.2 Member Typedef Documentation

8.49.2.1 `typedef std::map<int,HepMC::ParticleData*>::iterator HepMC::ParticleDataTable::iterator`

iterator for **ParticleData** (p. 206) map

Definition at line 75 of file ParticleDataTable.h.

8.49.2.2 `typedef std::map<int,HepMC::ParticleData*>::const_iterator HepMC::ParticleDataTable::const_iterator`

const iterator for **ParticleData** (p. 206) map

Definition at line 77 of file ParticleDataTable.h.

8.49.3 Constructor & Destructor Documentation

8.49.3.1 `HepMC::ParticleDataTable::ParticleDataTable (std::string description = std::string()) [inline]`

constructor with optional description

Definition at line 107 of file ParticleDataTable.h.

8.49.3.2 `HepMC::ParticleDataTable::ParticleDataTable (const char description) [inline]`

constructor with description

Definition at line 114 of file ParticleDataTable.h.

8.49.3.3 `HepMC::ParticleDataTable::ParticleDataTable (const ParticleDataTable &) [inline]`

copy constructor

Definition at line 122 of file ParticleDataTable.h.

8.49.3.4 `HepMC::ParticleDataTable::~~ParticleDataTable () [inline, virtual]`

Shallow: does not delete **ParticleData** (p. 206) entries.

Definition at line 126 of file ParticleDataTable.h.

8.49.4 Member Function Documentation

8.49.4.1 `ParticleDataTable & HepMC::ParticleDataTable::operator= (const ParticleDataTable &) [inline]`

shallow: does not copy the entries, only makes new pointers

Definition at line 128 of file ParticleDataTable.h.

References `m_data_table`, and `m_description`.

8.49.4.2 `void HepMC::ParticleDataTable::make_antiparticles_from_particles ()` [inline]

make corresponding anti-particles for all particles in table

make corresponding anti-particles for all particles in table

Definition at line 136 of file `ParticleDataTable.h`.

References `begin()`, `end()`, `insert()`, `merge_table()`, and `p`.

8.49.4.3 `int HepMC::ParticleDataTable::merge_table (const ParticleDataTable &)` [inline]

merge two tables

merges `pdt` into this table each entry from `pdt` is inserted only if this table does not already have an entry matching the `ParticleData`'s id returns the number of new entries inserted into this table.

Definition at line 251 of file `ParticleDataTable.h`.

References `begin()`, `end()`, `insert()`, and `p`.

Referenced by `make_antiparticles_from_particles()`.

8.49.4.4 `void HepMC::ParticleDataTable::print (std::ostream & ostr = std::cout)` `const` [inline]

write the table to `ostr`

prints a summary of all particle Data currently in memory

Examples:

`example_BuildEventFromScratch.cc`.

Definition at line 153 of file `ParticleDataTable.h`.

References `size()`.

Referenced by `main()`.

8.49.4.5 `void HepMC::ParticleDataTable::delete_all ()` [inline]

delete all **ParticleData** (p. 206) instances in this table

deletes all **ParticleData** (p. 206) instances in this table

Examples:

`example_BuildEventFromScratch.cc`.

Definition at line 242 of file `ParticleDataTable.h`.

References `clear()`.

Referenced by `main()`.

8.49.4.6 void HepMC::ParticleDataTable::clear () [inline]

clears table without deleting

Definition at line 249 of file ParticleDataTable.h.

Referenced by delete_all().

8.49.4.7 ParticleData * HepMC::ParticleDataTable::operator[] (int id) const [inline]

return pointer to requested **ParticleData** (p.206)

Definition at line 181 of file ParticleDataTable.h.

References find().

8.49.4.8 ParticleData * HepMC::ParticleDataTable::find (int id) const [inline]

return pointer to requested **ParticleData** (p.206)

finds a **ParticleData** (p.206) pointer corresponding to id IF it exists in the table. If not returns NULL

Definition at line 173 of file ParticleDataTable.h.

Referenced by HepMC::IO_PDG_ParticleDataTable::add_quarks_to_table(), operator[](), and HepMC::IO_PDG_ParticleDataTable::read_entry().

8.49.4.9 int HepMC::ParticleDataTable::size () const [inline]

size of table

Definition at line 185 of file ParticleDataTable.h.

Referenced by print().

8.49.4.10 bool HepMC::ParticleDataTable::empty () const [inline]

true if the table is empty

Definition at line 189 of file ParticleDataTable.h.

8.49.4.11 bool HepMC::ParticleDataTable::insert (ParticleData *) [inline]

true if successful

inserts pdata in the table IFF pdata's id has not already been used. It does NOT replace entries with the same id. True if successful. If you wish to overwrite another entry, first use **erase()** (p.218)

Examples:

example_BuildEventFromScratch.cc.

Definition at line 193 of file ParticleDataTable.h.

References HepMC::ParticleData::pdg_id().

Referenced by HepMC::IO_PDG_ParticleDataTable::add_quarks_to_table(), main(), make_antiparticles_from_particles(), merge_table(), HepMC::IO_PDG_ParticleDataTable::read_entry(), HepMC::IO_GenEvent::read_particle_data(), and HepMC::CommonIO::read_particle_data().

8.49.4.12 **bool HepMC::ParticleDataTable::erase (ParticleData *)** [inline]

removes from table - does not delete

removes from table does not delete returns True is an entry pdata existed in the table and was erased

Definition at line 201 of file ParticleDataTable.h.

References HepMC::ParticleData::pdg_id().

Referenced by HepMC::IO_PDG_ParticleDataTable::add_quarks_to_table().

8.49.4.13 **bool HepMC::ParticleDataTable::erase (int id)** [inline]

removes from table - does not delete

removes from table does not delete returns True is an entry pdata existed in the table and was erased

Definition at line 208 of file ParticleDataTable.h.

8.49.4.14 **ParticleDataTable::iterator HepMC::ParticleDataTable::begin ()** [inline]

begin iteration

Definition at line 214 of file ParticleDataTable.h.

Referenced by make_antiparticles_from_particles(), merge_table(), HepMC::IO_ExtendedAscii::write_particle_data_table(), and HepMC::IO_Ascii::write_particle_data_table().

8.49.4.15 **ParticleDataTable::iterator HepMC::ParticleDataTable::end ()** [inline]

end iteration

Definition at line 218 of file ParticleDataTable.h.

Referenced by make_antiparticles_from_particles(), merge_table(), HepMC::IO_ExtendedAscii::write_particle_data_table(), and HepMC::IO_Ascii::write_particle_data_table().

8.49.4.16 **ParticleDataTable::const_iterator HepMC::ParticleDataTable::begin ()** **const** [inline]

begin const iteration

Definition at line 222 of file ParticleDataTable.h.

8.49.4.17 ParticleDataTable::const_iterator HepMC::ParticleDataTable::end () const [inline]

end const iteration

Definition at line 226 of file ParticleDataTable.h.

8.49.4.18 std::string HepMC::ParticleDataTable::description () const [inline]

table description

Definition at line 230 of file ParticleDataTable.h.

8.49.4.19 void HepMC::ParticleDataTable::set_description (std::string) [inline]

set table description

Definition at line 234 of file ParticleDataTable.h.

Referenced by HepMC::IO_PDG_ParticleDataTable::fill_particle_data_table(), HepMC::IO_ExtendedAscii::fill_particle_data_table(), and HepMC::IO_Ascii::fill_particle_data_table().

8.49.4.20 void HepMC::ParticleDataTable::set_description (const char) [inline]

set table description

Definition at line 238 of file ParticleDataTable.h.

The documentation for this class was generated from the following file:

- **ParticleDataTable.h**

8.50 HepMC::PdfInfo Class Reference

The **PdfInfo** (p. 220) class stores PDF information.

```
#include <PdfInfo.h>
```

Public Member Functions

- **PdfInfo** ()
default constructor
- **PdfInfo** (int i1, int i2, double x1, double x2, double q, double p1, double p2)
all values must be provided
- **~PdfInfo** ()
- **PdfInfo** (**PdfInfo** const &orig)
copy constructor
- **PdfInfo** & **operator=** (**PdfInfo** const &rhs)
make a copy
- void **swap** (**PdfInfo** &other)
swap two PdfInfo (p. 220) objects
- bool **operator==** (const **PdfInfo** &) const
check for equality
- bool **operator!=** (const **PdfInfo** &) const
check for inequality
- int **id1** () const
flavour code of first parton
- int **id2** () const
flavour code of second parton
- double **x1** () const
fraction of beam momentum carried by first parton ("beam side")
- double **x2** () const
fraction of beam momentum carried by second parton ("target side")
- double **scalePDF** () const
Q-scale used in evaluation of PDF's (in GeV).
- double **pdf1** () const
PDF (id1, x1, Q).
- double **pdf2** () const
PDF (id2, x2, Q).

- void **set_id1** (const int &i)
set flavour code of first parton
- void **set_id2** (const int &i)
set flavour code of second parton
- void **set_x1** (const double &f)
set fraction of beam momentum carried by first parton ("beam side")
- void **set_x2** (const double &f)
set fraction of beam momentum carried by second parton ("target side")
- void **set_scalePDF** (const double &f)
set Q-scale used in evaluation of PDF's (in GeV)
- void **set_pdf1** (const double &f)
set PDF (id1, x1, Q)
- void **set_pdf2** (const double &f)
set PDF (id2, x2, Q)

8.50.1 Detailed Description

The **PdfInfo** (p. 220) class stores PDF information.

HepMC::PdfInfo (p. 220) stores additional PDF information for a **GenEvent** (p. 54). Creation and use of this information is optional.

Definition at line 30 of file PdfInfo.h.

8.50.2 Constructor & Destructor Documentation

8.50.2.1 HepMC::PdfInfo::PdfInfo () [inline]

default constructor

Definition at line 36 of file PdfInfo.h.

8.50.2.2 HepMC::PdfInfo::PdfInfo (int *i1*, int *i2*, double *x1*, double *x2*, double *q*, double *p1*, double *p2*) [inline]

all values must be provided

Definition at line 106 of file PdfInfo.h.

8.50.2.3 HepMC::PdfInfo::~~PdfInfo () [inline]

Definition at line 49 of file PdfInfo.h.

8.50.2.4 HepMC::PdfInfo::PdfInfo (PdfInfo const & *orig*) [inline]

copy constructor

Definition at line 116 of file PdfInfo.h.

8.50.3 Member Function Documentation**8.50.3.1 PdfInfo & HepMC::PdfInfo::operator= (PdfInfo const & *rhs*) [inline]**

make a copy

Definition at line 126 of file PdfInfo.h.

References swap().

8.50.3.2 void HepMC::PdfInfo::swap (PdfInfo & *other*) [inline]

swap two **PdfInfo** (p.220) objects

Definition at line 133 of file PdfInfo.h.

References m_id1, m_id2, m_pdf1, m_pdf2, m_scalePDF, m_x1, and m_x2.

Referenced by operator=().

8.50.3.3 bool HepMC::PdfInfo::operator== (const PdfInfo &) const [inline]

check for equality

equality requires that each member match

Definition at line 144 of file PdfInfo.h.

References id1(), id2(), pdf1(), pdf2(), scalePDF(), x1(), and x2().

8.50.3.4 bool HepMC::PdfInfo::operator!= (const PdfInfo &) const [inline]

check for inequality

any nonmatching member generates inequality

Definition at line 156 of file PdfInfo.h.

8.50.3.5 int HepMC::PdfInfo::id1 () const [inline]

flavour code of first parton

Definition at line 64 of file PdfInfo.h.

Referenced by operator==(), HepMC::IO_GenEvent::write_pdf_info(), and HepMC::IO_ExtendedAscii::write_pdf_info().

8.50.3.6 int HepMC::PdfInfo::id2 () const [inline]

flavour code of second parton

Definition at line 66 of file PdfInfo.h.

Referenced by operator==(), HepMC::IO_GenEvent::write_pdf_info(), and HepMC::IO_ExtendedAscii::write_pdf_info().

8.50.3.7 double HepMC::PdfInfo::x1 () const [inline]

fraction of beam momentum carried by first parton ("beam side")

Definition at line 68 of file PdfInfo.h.

Referenced by operator==(), HepMC::IO_GenEvent::write_pdf_info(), and HepMC::IO_ExtendedAscii::write_pdf_info().

8.50.3.8 double HepMC::PdfInfo::x2 () const [inline]

fraction of beam momentum carried by second parton ("target side")

Definition at line 70 of file PdfInfo.h.

Referenced by operator==(), HepMC::IO_GenEvent::write_pdf_info(), and HepMC::IO_ExtendedAscii::write_pdf_info().

8.50.3.9 double HepMC::PdfInfo::scalePDF () const [inline]

Q-scale used in evaluation of PDF's (in GeV).

Definition at line 72 of file PdfInfo.h.

Referenced by operator==(), HepMC::IO_GenEvent::write_pdf_info(), and HepMC::IO_ExtendedAscii::write_pdf_info().

8.50.3.10 double HepMC::PdfInfo::pdf1 () const [inline]

PDF (id1, x1, Q).

Definition at line 74 of file PdfInfo.h.

Referenced by operator==(), HepMC::IO_GenEvent::write_pdf_info(), and HepMC::IO_ExtendedAscii::write_pdf_info().

8.50.3.11 double HepMC::PdfInfo::pdf2 () const [inline]

PDF (id2, x2, Q).

Definition at line 76 of file PdfInfo.h.

Referenced by operator==(), HepMC::IO_GenEvent::write_pdf_info(), and HepMC::IO_ExtendedAscii::write_pdf_info().

8.50.3.12 void HepMC::PdfInfo::set_id1 (const int & i) [inline]

set flavour code of first parton

Definition at line 80 of file PdfInfo.h.

8.50.3.13 void HepMC::PdfInfo::set_id2 (const int & i) [inline]

set flavour code of second parton

Definition at line 82 of file PdfInfo.h.

8.50.3.14 void HepMC::PdfInfo::set_x1 (const double & f) [inline]

set fraction of beam momentum carried by first parton ("beam side")

Definition at line 84 of file PdfInfo.h.

8.50.3.15 void HepMC::PdfInfo::set_x2 (const double & f) [inline]

set fraction of beam momentum carried by second parton ("target side")

Definition at line 86 of file PdfInfo.h.

8.50.3.16 void HepMC::PdfInfo::set_scalePDF (const double & f) [inline]

set Q-scale used in evaluation of PDF's (in GeV)

Definition at line 88 of file PdfInfo.h.

8.50.3.17 void HepMC::PdfInfo::set_pdf1 (const double & f) [inline]

set PDF (id1, x1, Q)

Definition at line 90 of file PdfInfo.h.

8.50.3.18 void HepMC::PdfInfo::set_pdf2 (const double & f) [inline]

set PDF (id2, x2, Q)

Definition at line 92 of file PdfInfo.h.

The documentation for this class was generated from the following file:

- PdfInfo.h

8.51 HepMC::Polarization Class Reference

The **Polarization** (p. 225) class stores theta and phi for a **GenParticle** (p. 86).

```
#include <Polarization.h>
```

Public Member Functions

- **Polarization** (double theta=0, double phi=0)
default constructor
- **Polarization** (const **Polarization** &inpolar)
construct from another polarization object
- **Polarization** (const **ThreeVector** &vec3in)
*construct using the polar and azimuthal angles from a **ThreeVector** (p. 232)*
- virtual \sim **Polarization** ()
- void **swap** (**Polarization** &other)
swap
- **Polarization** & **operator=** (const **Polarization** &inpolar)
make a copy
- bool **operator==** (const **Polarization** &) const
equality requires that theta and phi are equal
- bool **operator!=** (const **Polarization** &) const
inequality results if either theta or phi differ
- void **print** (std::ostream &ostr=std::cout) const
print theta and phi
- double **theta** () const
returns polar angle in radians
- double **phi** () const
returns azimuthal angle in radians
- **ThreeVector** **normal3d** () const
unit 3 vector for easy manipulation
- double **set_theta** (double theta)
set polar angle in radians
- double **set_phi** (double phi)
set azimuthal angle in radians
- void **set_theta_phi** (double theta, double phi)
set both polar and azimuthal angles in radians

- **ThreeVector** `set_normal3d` (const **ThreeVector** &vec3in)
sets polarization according to direction of 3 vec

Friends

- `std::ostream & operator<<` (std::ostream &, const **Polarization** &)
print polarization information

8.51.1 Detailed Description

The **Polarization** (p. 225) class stores theta and phi for a **GenParticle** (p. 86).

HepMC::Polarization (p. 225) stores a particle's theta and phi in radians. Use of this information is optional. By default, the polarization is set to zero.

Definition at line 29 of file Polarization.h.

8.51.2 Constructor & Destructor Documentation

8.51.2.1 HepMC::Polarization::Polarization (double *theta* = 0, double *phi* = 0)

default constructor

Definition at line 11 of file Polarization.cc.

8.51.2.2 HepMC::Polarization::Polarization (const **Polarization** & *inpolar*)

construct from another polarization object

Definition at line 16 of file Polarization.cc.

8.51.2.3 HepMC::Polarization::Polarization (const **ThreeVector** & *vec3in*)

construct using the polar and azimuthal angles from a **ThreeVector** (p. 232)

Definition at line 21 of file Polarization.cc.

8.51.2.4 virtual HepMC::Polarization::~~Polarization () [inline, virtual]

Definition at line 41 of file Polarization.h.

8.51.3 Member Function Documentation

8.51.3.1 void HepMC::Polarization::swap (**Polarization** & *other*)

swap

Definition at line 26 of file Polarization.cc.

References `m_phi`, and `m_theta`.

Referenced by `operator=()`, and `HepMC::GenParticle::swap()`.

8.51.3.2 Polarization & HepMC::Polarization::operator= (const Polarization & *inpolar*)

make a copy

best practices implementation

Definition at line 32 of file `Polarization.cc`.

References `swap()`.

8.51.3.3 bool HepMC::Polarization::operator== (const Polarization &) const [inline]

equality requires that `theta` and `phi` are equal

Definition at line 93 of file `Polarization.h`.

References `phi()`, and `theta()`.

8.51.3.4 bool HepMC::Polarization::operator!= (const Polarization &) const [inline]

inequality results if either `theta` or `phi` differ

Definition at line 98 of file `Polarization.h`.

8.51.3.5 void HepMC::Polarization::print (std::ostream & *ostr* = std::cout) const

print `theta` and `phi`

Definition at line 39 of file `Polarization.cc`.

8.51.3.6 double HepMC::Polarization::theta () const [inline]

returns polar angle in radians

Definition at line 86 of file `Polarization.h`.

Referenced by `normal3d()`, `HepMC::operator<<()`, and `operator==()`.

8.51.3.7 double HepMC::Polarization::phi () const [inline]

returns azimuthal angle in radians

Definition at line 87 of file `Polarization.h`.

Referenced by `normal3d()`, `HepMC::operator<<()`, and `operator==()`.

8.51.3.8 ThreeVector HepMC::Polarization::normal3d () const

unit 3 vector for easy manipulation

Definition at line 47 of file Polarization.cc.

References `phi()`, `HepMC::ThreeVector::setPhi()`, `HepMC::ThreeVector::setTheta()`, and `theta()`.

8.51.3.9 double HepMC::Polarization::set_theta (double *theta*)

set polar angle in radians

Theta is restricted to be between 0 \rightarrow pi if an out of range value is given, it is translated to this range.

Definition at line 55 of file Polarization.cc.

Referenced by `set_normal3d()`, and `set_theta_phi()`.

8.51.3.10 double HepMC::Polarization::set_phi (double *phi*)

set azimuthal angle in radians

Phi is restricted to be between 0 \rightarrow 2pi if an out of range value is given, it is translated to this range.

Definition at line 61 of file Polarization.cc.

Referenced by `set_normal3d()`, and `set_theta_phi()`.

8.51.3.11 void HepMC::Polarization::set_theta_phi (double *theta*, double *phi*)

set both polar and azimuthal angles in radians

Definition at line 67 of file Polarization.cc.

References `set_phi()`, and `set_theta()`.

8.51.3.12 ThreeVector HepMC::Polarization::set_normal3d (const ThreeVector & *vec3in*)

sets polarization according to direction of 3 vec

Definition at line 72 of file Polarization.cc.

References `HepMC::ThreeVector::phi()`, `set_phi()`, `set_theta()`, and `HepMC::ThreeVector::theta()`.

8.51.4 Friends And Related Function Documentation

8.51.4.1 std::ostream& operator<< (std::ostream & *ostr*, const Polarization & *polar*) [friend]

print polarization information

Definition at line 107 of file Polarization.cc.

The documentation for this class was generated from the following files:

- **Polarization.h**
- **Polarization.cc**

8.52 HepMC::TempParticleMap Class Reference

TempParticleMap (p.229) is a temporary GenParticle* container used during input.

```
#include <TempParticleMap.h>
```

Public Types

- typedef std::map< **HepMC::GenParticle** *, int > **TempMap**
- typedef std::map< int, **HepMC::GenParticle** * > **TempOrderMap**
- typedef TempMap::iterator **TempMapIterator**
- typedef TempOrderMap::iterator **orderIterator**

Public Member Functions

- **TempParticleMap** ()
- ~**TempParticleMap** ()
- **TempMapIterator** **begin** ()
- **TempMapIterator** **end** ()
- **orderIterator** **order_begin** ()
- **orderIterator** **order_end** ()
- int **end_vertex** (**GenParticle** *)
- void **addEndParticle** (**GenParticle** *, int &)

8.52.1 Detailed Description

TempParticleMap (p.229) is a temporary GenParticle* container used during input.

Used by IO classes for recoverable particle ordering. Map GenParticle* against both outgoing vertex and particle order.

Definition at line 24 of file TempParticleMap.h.

8.52.2 Member Typedef Documentation

8.52.2.1 typedef std::map<HepMC::GenParticle*,int> HepMC::TempParticleMap::TempMap

Definition at line 26 of file TempParticleMap.h.

8.52.2.2 typedef std::map<int,HepMC::GenParticle*> HepMC::TempParticleMap::TempOrderMap

Definition at line 27 of file TempParticleMap.h.

8.52.2.3 typedef TempMap::iterator HepMC::TempParticleMap::TempMapIterator

Definition at line 28 of file TempParticleMap.h.

8.52.2.4 `typedef TempOrderMap::iterator HepMC::TempParticleMap::order-Iterator`

Definition at line 29 of file TempParticleMap.h.

8.52.3 Constructor & Destructor Documentation

8.52.3.1 `HepMC::TempParticleMap::TempParticleMap ()` [inline]

Definition at line 31 of file TempParticleMap.h.

8.52.3.2 `HepMC::TempParticleMap::~~TempParticleMap ()` [inline]

Definition at line 34 of file TempParticleMap.h.

8.52.4 Member Function Documentation

8.52.4.1 `TempMapIterator HepMC::TempParticleMap::begin ()` [inline]

Definition at line 36 of file TempParticleMap.h.

8.52.4.2 `TempMapIterator HepMC::TempParticleMap::end ()` [inline]

Definition at line 37 of file TempParticleMap.h.

Referenced by `end_vertex()`.

8.52.4.3 `orderIterator HepMC::TempParticleMap::order_begin ()` [inline]

Definition at line 38 of file TempParticleMap.h.

Referenced by `HepMC::CommonIO::read_io_ascii()`, `HepMC::CommonIO::read_io_extendedascii()`, and `HepMC::CommonIO::read_io_genevent()`.

8.52.4.4 `orderIterator HepMC::TempParticleMap::order_end ()` [inline]

Definition at line 39 of file TempParticleMap.h.

Referenced by `HepMC::CommonIO::read_io_ascii()`, `HepMC::CommonIO::read_io_extendedascii()`, and `HepMC::CommonIO::read_io_genevent()`.

8.52.4.5 `int HepMC::TempParticleMap::end_vertex (GenParticle *)` [inline]

Definition at line 50 of file TempParticleMap.h.

References `end()`, and `p`.

Referenced by `HepMC::CommonIO::read_io_ascii()`, `HepMC::CommonIO::read_io_extendedascii()`, and `HepMC::CommonIO::read_io_genevent()`.

8.52.4.6 void HepMC::TempParticleMap::addEndParticle (GenParticle *, int &) [inline]

Definition at line 58 of file TempParticleMap.h.

References [p](#).

Referenced by [HepMC::IO_GenEvent::read_particle\(\)](#), and [HepMC::CommonIO::read_particle\(\)](#).

The documentation for this class was generated from the following file:

- **TempParticleMap.h**

8.53 HepMC::ThreeVector Class Reference

ThreeVector (p. 232) is a simple representation of a position or displacement 3 vector.

```
#include <SimpleVector.h>
```

Public Member Functions

- **ThreeVector** (double xin, double yin=0, double zin=0)
construct using x, y, and z (only x is required)
- **ThreeVector** ()
- template<class T> **ThreeVector** (const T &v, typename detail::disable_if<detail::is_arithmetic< T >::value, void >::type * =0)
- **ThreeVector** (const **ThreeVector** &v)
copy constructor
- void **swap** (**ThreeVector** &other)
swap
- double **x** () const
return x
- double **y** () const
return y
- double **z** () const
return z
- void **setX** (double x)
set x
- void **setY** (double y)
set y
- void **setZ** (double z)
set z
- void **set** (double x, double y, double z)
set x, y, and z
- double **phi** () const
The azimuth angle.
- double **theta** () const
The polar angle.
- double **r** () const
The magnitude.

- double **mag** () const
The magnitude (r in spherical coordinate system).
- void **setPhi** (double)
Set phi keeping mag and theta constant (BaBar).
- void **setTheta** (double)
Set theta keeping mag and phi constant (BaBar).
- double **perp2** () const
The transverse component squared (ρ^2 in cylindrical coordinate system).
- double **perp** () const
The transverse component (ρ in cylindrical coordinate system).
- **ThreeVector** & **operator=** (const **ThreeVector** &)
make a copy
- bool **operator==** (const **ThreeVector** &) const
equality
- bool **operator!=** (const **ThreeVector** &) const
inequality

8.53.1 Detailed Description

ThreeVector (p. 232) is a simple representation of a position or displacement 3 vector.

For compatibility with existing code, the basic expected geometrical access methods are provided. Also, there is a templated constructor that will take another vector (HepLorentzVector, GenVector, ...) which must have the following methods: **x()** (p. 234), **y()** (p. 234), **z()** (p. 234).

Examples:

VectorConversion.h.

Definition at line 132 of file SimpleVector.h.

8.53.2 Constructor & Destructor Documentation

8.53.2.1 HepMC::ThreeVector::ThreeVector (double *xin*, double *yin* = 0, double *zin* = 0) [inline]

construct using x, y, and z (only x is required)

Definition at line 137 of file SimpleVector.h.

8.53.2.2 HepMC::ThreeVector::ThreeVector () [inline]

Definition at line 140 of file SimpleVector.h.

8.53.2.3 `template<class T> HepMC::ThreeVector::ThreeVector (const T & v,
typename detail::disable_if< detail::is_arithmetic< T >::value, void
>::type * = 0) [inline]`

templated constructor this is used ONLY if T is not arithmetic

Definition at line 146 of file SimpleVector.h.

8.53.2.4 `HepMC::ThreeVector::ThreeVector (const ThreeVector & v) [inline]`

copy constructor

Definition at line 151 of file SimpleVector.h.

8.53.3 Member Function Documentation

8.53.3.1 `void HepMC::ThreeVector::swap (ThreeVector & other)`

swap

8.53.3.2 `double HepMC::ThreeVector::x () const [inline]`

return x

Definition at line 156 of file SimpleVector.h.

Referenced by main().

8.53.3.3 `double HepMC::ThreeVector::y () const [inline]`

return y

Definition at line 157 of file SimpleVector.h.

Referenced by main().

8.53.3.4 `double HepMC::ThreeVector::z () const [inline]`

return z

Definition at line 158 of file SimpleVector.h.

Referenced by main().

8.53.3.5 `void HepMC::ThreeVector::setX (double x) [inline]`

set x

Definition at line 160 of file SimpleVector.h.

Referenced by main().

8.53.3.6 void HepMC::ThreeVector::setY (double *y*) [inline]

set *y*

Definition at line 161 of file SimpleVector.h.

Referenced by main().

8.53.3.7 void HepMC::ThreeVector::setZ (double *z*) [inline]

set *z*

Definition at line 162 of file SimpleVector.h.

Referenced by main().

8.53.3.8 void HepMC::ThreeVector::set (double *x*, double *y*, double *z*)

set *x*, *y*, and *z*

Referenced by main().

8.53.3.9 double HepMC::ThreeVector::phi () const

The azimuth angle.

Referenced by main(), and HepMC::Polarization::set_normal3d().

8.53.3.10 double HepMC::ThreeVector::theta () const

The polar angle.

Referenced by main(), and HepMC::Polarization::set_normal3d().

8.53.3.11 double HepMC::ThreeVector::r () const

The magnitude.

Referenced by main().

8.53.3.12 double HepMC::ThreeVector::mag () const

The magnitude (*r* in spherical coordinate system).

Referenced by main().

8.53.3.13 void HepMC::ThreeVector::setPhi (double)

Set *phi* keeping *mag* and *theta* constant (BaBar).

Referenced by main(), and HepMC::Polarization::normal3d().

8.53.3.14 void HepMC::ThreeVector::setTheta (double)

Set theta keeping mag and phi constant (BaBar).

Referenced by `main()`, and `HepMC::Polarization::normal3d()`.

8.53.3.15 double HepMC::ThreeVector::perp2 () const

The transverse component squared (ρ^2 in cylindrical coordinate system).

Referenced by `main()`.

8.53.3.16 double HepMC::ThreeVector::perp () const

The transverse component (ρ in cylindrical coordinate system).

Referenced by `main()`.

8.53.3.17 ThreeVector& HepMC::ThreeVector::operator= (const ThreeVector &)

make a copy

8.53.3.18 bool HepMC::ThreeVector::operator== (const ThreeVector &) const

equality

8.53.3.19 bool HepMC::ThreeVector::operator!= (const ThreeVector &) const

inequality

The documentation for this class was generated from the following file:

- **SimpleVector.h**

8.54 HepMC::WeightContainer Class Reference

Container for the Weights associated with an event or vertex.

```
#include <WeightContainer.h>
```

Public Types

- `typedef std::vector< double >::iterator iterator`
iterator for the weight container
- `typedef std::vector< double >::const_iterator const_iterator`
const iterator for the weight container

Public Member Functions

- **WeightContainer** (unsigned int `n=0`, const double &value=0.)
default constructor
- **WeightContainer** (const std::vector< double > &weights)
construct from a vector of weights
- **WeightContainer** (const **WeightContainer** &in)
copy
- virtual **~WeightContainer** ()
- void **swap** (**WeightContainer** &other)
swap
- **WeightContainer** & **operator=** (const **WeightContainer** &)
copy
- **WeightContainer** & **operator=** (const std::vector< double > &in)
copy
- void **print** (std::ostream &ostr=std::cout) const
print weights
- int **size** () const
size of weight container
- bool **empty** () const
return true if weight container is empty
- void **push_back** (const double &)
push onto weight container
- void **pop_back** ()
pop from weight container

- **void clear ()**
clear the weight container
- **double & operator[] (unsigned int n)**
access the weight container
- **const double & operator[] (unsigned int n) const**
access the weight container
- **double & front ()**
returns the first element
- **const double & front () const**
returns the first element
- **double & back ()**
returns the last element
- **const double & back () const**
returns the last element
- **iterator begin ()**
beginning of the weight container
- **iterator end ()**
end of the weight container
- **const_iterator begin () const**
beginning of the weight container
- **const_iterator end () const**
end of the weight container

8.54.1 Detailed Description

Container for the Weights associated with an event or vertex.

Basically just an interface to STL vector.

Definition at line 24 of file WeightContainer.h.

8.54.2 Member Typedef Documentation

8.54.2.1 `typedef std::vector<double>::iterator HepMC::WeightContainer::iterator`

iterator for the weight container

Definition at line 71 of file WeightContainer.h.

8.54.2.2 `typedef std::vector<double>::const_iterator HepMC::WeightContainer::const_iterator`

const iterator for the weight container

Definition at line 73 of file WeightContainer.h.

8.54.3 Constructor & Destructor Documentation

8.54.3.1 `HepMC::WeightContainer::WeightContainer (unsigned int n = 0, const double & value = 0.) [inline]`

default constructor

Definition at line 91 of file WeightContainer.h.

8.54.3.2 `HepMC::WeightContainer::WeightContainer (const std::vector< double > & weights) [inline]`

construct from a vector of weights

Definition at line 96 of file WeightContainer.h.

8.54.3.3 `HepMC::WeightContainer::WeightContainer (const WeightContainer & in) [inline]`

copy

Definition at line 100 of file WeightContainer.h.

8.54.3.4 `HepMC::WeightContainer::~~WeightContainer () [inline, virtual]`

Definition at line 104 of file WeightContainer.h.

8.54.4 Member Function Documentation

8.54.4.1 `void HepMC::WeightContainer::swap (WeightContainer & other) [inline]`

swap

Definition at line 106 of file WeightContainer.h.

References `m_weights`.

Referenced by `operator=()`, `HepMC::GenVertex::swap()`, and `HepMC::GenEvent::swap()`.

8.54.4.2 `WeightContainer & HepMC::WeightContainer::operator= (const WeightContainer &) [inline]`

copy

best practices implementation

Definition at line 110 of file WeightContainer.h.

References swap().

8.54.4.3 **WeightContainer & HepMC::WeightContainer::operator= (const std::vector< double > & in) [inline]**

copy

best practices implementation

Definition at line 118 of file WeightContainer.h.

References swap().

8.54.4.4 **void HepMC::WeightContainer::print (std::ostream & ostr = std::cout) const [inline]**

print weights

Definition at line 125 of file WeightContainer.h.

References begin(), and end().

8.54.4.5 **int HepMC::WeightContainer::size () const [inline]**

size of weight container

Definition at line 134 of file WeightContainer.h.

Referenced by HepMC::compareWeights(), HepMC::GenVertex::print(), HepMC::GenEvent::print(), HepMC::IO_GenEvent::write_event(), HepMC::IO_ExtendedAscii::write_event(), HepMC::IO_AsciiParticles::write_event(), and HepMC::IO_Ascii::write_event().

8.54.4.6 **bool HepMC::WeightContainer::empty () const [inline]**

return true if weight container is empty

Definition at line 136 of file WeightContainer.h.

8.54.4.7 **void HepMC::WeightContainer::push_back (const double &) [inline]**

push onto weight container

Definition at line 138 of file WeightContainer.h.

8.54.4.8 **void HepMC::WeightContainer::pop_back () [inline]**

pop from weight container

Definition at line 141 of file WeightContainer.h.

8.54.4.9 void HepMC::WeightContainer::clear () [inline]

clear the weight container

Definition at line 143 of file WeightContainer.h.

8.54.4.10 double & HepMC::WeightContainer::operator[] (unsigned int *n*) [inline]

access the weight container

Definition at line 145 of file WeightContainer.h.

8.54.4.11 const double & HepMC::WeightContainer::operator[] (unsigned int *n*) const [inline]

access the weight container

Definition at line 148 of file WeightContainer.h.

8.54.4.12 double & HepMC::WeightContainer::front () [inline]

returns the first element

Definition at line 151 of file WeightContainer.h.

8.54.4.13 const double & HepMC::WeightContainer::front () const [inline]

returns the first element

Definition at line 153 of file WeightContainer.h.

8.54.4.14 double & HepMC::WeightContainer::back () [inline]

returns the last element

Definition at line 156 of file WeightContainer.h.

8.54.4.15 const double & HepMC::WeightContainer::back () const [inline]

returns the last element

Definition at line 158 of file WeightContainer.h.

8.54.4.16 WeightContainer::iterator HepMC::WeightContainer::begin () [inline]

beginning of the weight container

Definition at line 161 of file WeightContainer.h.

Referenced by `print()`, `HepMC::IO_GenEvent::write_event()`, `HepMC::IO_ExtendedAscii::write_event()`, `HepMC::IO_AsciiParticles::write_event()`, and `HepMC::IO_Ascii::write_event()`.

8.54.4.17 WeightContainer::iterator HepMC::WeightContainer::end () [inline]

end of the weight container

Definition at line 164 of file WeightContainer.h.

Referenced by `print()`, `HepMC::GenVertex::print()`, `HepMC::GenEvent::print()`, `HepMC::IO_GenEvent::write_event()`, `HepMC::IO_ExtendedAscii::write_event()`, `HepMC::IO_AsciiParticles::write_event()`, and `HepMC::IO_Ascii::write_event()`.

8.54.4.18 WeightContainer::const_iterator HepMC::WeightContainer::begin () const [inline]

beginning of the weight container

Definition at line 167 of file WeightContainer.h.

8.54.4.19 WeightContainer::const_iterator HepMC::WeightContainer::end () const [inline]

end of the weight container

Definition at line 170 of file WeightContainer.h.

The documentation for this class was generated from the following file:

- **WeightContainer.h**

Chapter 9

HepMC File Documentation

9.1 CommonIO.cc File Reference

```
#include "HepMC/CommonIO.h"  
#include "HepMC/GenEvent.h"  
#include "HepMC/HeavyIon.h"  
#include "HepMC/PdfInfo.h"  
#include "HepMC/TempParticleMap.h"  
#include "HepMC/ParticleDataTable.h"
```

Namespaces

- namespace **HepMC**

9.2 CommonIO.h File Reference

```
#include <fstream>
#include <string>
#include "HepMC/GenEvent.h"
#include "HepMC/TempParticleMap.h"
#include "HepMC/ParticleDataTable.h"
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::CommonIO**

Enumerations

- enum **HepMC::known_io** {
 HepMC::gen = 1, **HepMC::ascii**, **HepMC::extascii**, **HepMC::ascii_pdt**,
 HepMC::extascii_pdt }

The known_io enum is used to track which type of input is being read.

9.3 CompareGenEvent.cc File Reference

```
#include <iostream>
#include "HepMC/CompareGenEvent.h"
#include "HepMC/GenEvent.h"
```

Namespaces

- namespace **HepMC**

Functions

- bool **HepMC::compareGenEvent** (GenEvent *, GenEvent *)
- bool **HepMC::compareSignalProcessVertex** (GenEvent *, GenEvent *)
- bool **HepMC::compareBeamParticles** (GenEvent *, GenEvent *)
- bool **HepMC::compareWeights** (GenEvent *, GenEvent *)
- bool **HepMC::compareParticles** (GenEvent *, GenEvent *)
- bool **HepMC::compareVertices** (GenEvent *, GenEvent *)
- bool **HepMC::compareVertex** (GenVertex *v1, GenVertex *v2)

9.4 CompareGenEvent.h File Reference

```
#include <iostream>
#include "HepMC/GenEvent.h"
```

Namespaces

- namespace **HepMC**

Functions

- bool **HepMC::compareGenEvent** (GenEvent *, GenEvent *)
- bool **HepMC::compareSignalProcessVertex** (GenEvent *, GenEvent *)
- bool **HepMC::compareBeamParticles** (GenEvent *, GenEvent *)
- bool **HepMC::compareWeights** (GenEvent *, GenEvent *)
- bool **HepMC::compareVertices** (GenEvent *, GenEvent *)
- bool **HepMC::compareParticles** (GenEvent *, GenEvent *)
- bool **HepMC::compareVertex** (GenVertex *v1, GenVertex *v2)

9.5 enable_if.h File Reference

Namespaces

- namespace **HepMC**
- namespace **HepMC::detail**

Classes

- struct **HepMC::detail::enable_if**<, >
internal - used to decide if a class is arithmetic
- struct **HepMC::detail::enable_if**< true, T >
internal - use if class T is arithmetic
- struct **HepMC::detail::disable_if**<, >
internal - used by SimpleVector to decide if a class is arithmetic
- struct **HepMC::detail::disable_if**< false, T >
internal - used by SimpleVector to decide if a class is arithmetic

9.6 example_BuildEventFromScratch.cc File Reference

```
#include <iostream>
#include "VectorConversion.h"
#include "HepMC/GenEvent.h"
#include "HepMC/ParticleDataTable.h"
#include "CLHEP/Vector/LorentzVector.h"
```

Functions

- `int main ()`

9.6.1 Function Documentation

9.6.1.1 `int main ()`

Examples:

`example_BuildEventFromScratch.cc`, `example_EventSelection.cc`, `example_MyHerwig.cc`, `example_MyPythia.cc`, `example_MyPythiaOnlyToHepMC.cc`, `example_MyPythiaRead.cc`, `example_MyPythiaWithEventSelection.cc`, `example_PythiaParticle.cc`, `example_UsingIterators.cc`, `testHerwigCopies.cc`, and `testPythiaCopies.cc`.

Definition at line 26 of file `example_BuildEventFromScratch.cc`.

References `HepMC::GenVertex::add_particle_in()`, `HepMC::GenVertex::add_particle_out()`, `HepMC::GenEvent::add_vertex()`, `HepMC::clifetime_from_width()`, `convertTo()`, `HepMC::ParticleDataTable::delete_all()`, `HepMC::ParticleDataTable::insert()`, `p`, `HepMC::GenEvent::particles_begin()`, `HepMC::GenEvent::particles_end()`, `HepMC::GenEvent::print()`, `HepMC::ParticleDataTable::print()`, and `HepMC::GenEvent::set_signal_process_vertex()`.

9.7 example_EventSelection.cc File Reference

```
#include "HepMC/IO_GenEvent.h"
```

```
#include "HepMC/GenEvent.h"
```

Classes

- class **IsEventGood**
example class

Functions

- int **main** ()

9.7.1 Function Documentation

9.7.1.1 int main ()

Definition at line 37 of file example_EventSelection.cc.

References `HepMC::GenEvent::event_number()`, and `HepMC::IO_BaseClass::read_next_event()`.

9.8 example_MyHerwig.cc File Reference

```
#include <iostream>
#include "HepMC/HerwigWrapper.h"
#include "HepMC/IO_HERWIG.h"
#include "HepMC/GenEvent.h"
#include "HepMC/HEPEVT_Wrapper.h"
```

Functions

- `int main ()`

9.8.1 Function Documentation

9.8.1.1 `int main ()`

To Compile: go to the **HepMC** (p.19) directory and type: `gmake examples/example_MyHerwig.exe`

In this example the precision and number of entries for the HEPEVT fortran common block are explicitly defined to correspond to those used in the Herwig version of the HEPEVT common block. If you get funny output from HEPEVT in your own code, probably you have set these values incorrectly!

Definition at line 23 of file `example_MyHerwig.cc`.

References `hwbgen`, `hwbmch`, `hwcdec`, `hwcfor`, `hwdhad`, `hwdhob`, `hwdhvy`, `hwefin`, `hweini`, `hwepro`, `hwevnt`, `hwigin`, `hwmevt`, `hwproc`, `hwufne`, `hwuinc`, `hwuine`, `HepMC::GenEvent::print()`, `HepMC::HEPEVT_Wrapper::print_hepevt()`, `HepMC::IO_BaseClass::read_next_event()`, `HepMC::GenEvent::set_event_number()`, `HepMC::HEPEVT_Wrapper::set_max_number_entries()`, `HepMC::GenEvent::set_signal_process_id()`, and `HepMC::HEPEVT_Wrapper::set_sizeof_real()`.

9.9 example_MyPythia.cc File Reference

```
#include <iostream>
#include "HepMC/PythiaWrapper.h"
#include "HepMC/IO_HEPEVT.h"
#include "HepMC/IO_GenEvent.h"
#include "HepMC/GenEvent.h"
#include "PythiaHelper.h"
```

Functions

- `int main ()`

9.9.1 Function Documentation

9.9.1.1 `int main ()`

To Compile: go to the **HepMC** (p.19) directory and type: `gmake examples/example_MyPythia.exe`

In this example the precision and number of entries for the HEPEVT fortran common block are explicitly defined to correspond to those used in the Pythia version of the HEPEVT common block.

If you get funny output from HEPEVT in your own code, probably you have set these values incorrectly!

Definition at line 28 of file `example_MyPythia.cc`.

References `initPythia()`, `pypars`, `HepMC::IO_BaseClass::read_next_event()`, `HepMC::GenEvent::set_event_number()`, `HepMC::HEPEVT_Wrapper::set_max_number_entries()`, `HepMC::GenEvent::set_mpi()`, `HepMC::GenEvent::set_signal_process_id()`, and `HepMC::HEPEVT_Wrapper::set_sizeof_real()`.

9.10 example_MyPythiaOnlyToHepMC.cc File Reference

```
#include <iostream>
#include "HepMC/PythiaWrapper.h"
#include "HepMC/IO_HEPEVT.h"
#include "HepMC/GenEvent.h"
#include "PythiaHelper.h"
```

Functions

- `int main ()`

9.10.1 Function Documentation

9.10.1.1 `int main ()`

Definition at line 23 of file `example_MyPythiaOnlyToHepMC.cc`.

References `initPythia()`, `pypars`, `HepMC::IO_BaseClass::read_next_event()`, `HepMC::HEPEVT_Wrapper::set_max_number_entries()`, `HepMC::GenEvent::set_mpi()`, and `HepMC::HEPEVT_Wrapper::set_sizeof_real()`.

9.11 example_MyPythiaRead.cc File Reference

```
#include <iostream>
#include "HepMC/PythiaWrapper.h"
#include "HepMC/IO_HEPEVT.h"
#include "HepMC/IO_GenEvent.h"
#include "HepMC/GenEvent.h"
#include "PythiaHelper.h"
```

Functions

- `int main ()`

9.11.1 Function Documentation

9.11.1.1 `int main ()`

Definition at line 28 of file `example_MyPythiaRead.cc`.

References `HepMC::GenEvent::event_number()`, `initPythia()`, `HepMC::IO_BaseClass::read_next_event()`, `HepMC::GenEvent::set_event_number()`, `HepMC::HEPEVT_Wrapper::set_max_number_entries()`, `HepMC::GenEvent::set_signal_process_id()`, and `HepMC::HEPEVT_Wrapper::set_sizeof_real()`.

9.12 example_MyPythiaWithEventSelection.cc File Reference

```
#include <iostream>
#include "HepMC/PythiaWrapper.h"
#include "HepMC/IO_HEPEVT.h"
#include "HepMC/GenEvent.h"
#include "PythiaHelper.h"
```

Classes

- class **IsGoodEventMyPythia**
example class

Functions

- int **main** ()

9.12.1 Function Documentation

9.12.1.1 int main ()

Definition at line 45 of file example_MyPythiaWithEventSelection.cc.

References `initPythia()`, `pypars`, `HepMC::IO_BaseClass::read_next_event()`, `HepMC::HEPEVT_Wrapper::set_max_number_entries()`, `HepMC::GenEvent::set_mpi()`, and `HepMC::HEPEVT_Wrapper::set_sizeof_real()`.

9.13 example_PythiaParticle.cc File Reference

```
#include <iostream>
#include "HepMC/PythiaWrapper.h"
#include "HepMC/IO_HEPEVT.h"
#include "HepMC/IO_AsciiParticles.h"
#include "HepMC/GenEvent.h"
#include "PythiaHelper.h"
```

Functions

- `int main ()`

9.13.1 Function Documentation

9.13.1.1 `int main ()`

Definition at line 29 of file `example_PythiaParticle.cc`.

References `initPythia()`, `HepMC::IO_BaseClass::read_next_event()`, `HepMC::GenEvent::set_event_number()`, `HepMC::HEPEVT_Wrapper::set_max_number_entries()`, `HepMC::GenEvent::set_signal_process_id()`, and `HepMC::HEPEVT_Wrapper::set_sizeof_real()`.

9.14 example_ReadMyPythia.cc File Reference

```
#include <iostream>
#include <fstream>
#include "HepMC/IO_GenEvent.h"
#include "HepMC/GenEvent.h"
```

Functions

- `int main ()`

9.14.1 Function Documentation

9.14.1.1 `int main ()`

Definition at line 15 of file `example_ReadMyPythia.cc`.

References `HepMC::GenEvent::event_number()`, and `HepMC::IO_BaseClass::read_next_event()`.

9.15 example_UsingIterators.cc File Reference

```
#include "HepMC/IO_GenEvent.h"
#include "HepMC/GenEvent.h"
#include <math.h>
#include <algorithm>
#include <list>
```

Classes

- class **IsPhoton**
example class
- class **IsW_Boson**
example class
- class **IsStateFinal**
example class

Functions

- int **main** ()

9.15.1 Function Documentation

9.15.1.1 int main ()

Definition at line 56 of file example_UsingIterators.cc.

References `HepMC::copy_if()`, `HepMC::descendants`, `p`, `HepMC::parents`, `HepMC::GenEvent::particles_begin()`, `HepMC::GenEvent::particles_end()`, `HepMC::IO_GenEvent::rdstate()`, `HepMC::IO_BaseClass::read_next_event()`, `v`, `HepMC::GenEvent::vertices_begin()`, and `HepMC::GenEvent::vertices_end()`.

9.16 Flow.cc File Reference

```
#include "HepMC/Flow.h"
#include "HepMC/GenParticle.h"
#include "HepMC/GenVertex.h"
#include "HepMC/SearchVector.h"
```

Namespaces

- namespace **HepMC**

Functions

- `std::ostream & HepMC::operator<< (std::ostream &ostr, const Flow &f)`
for printing

9.17 Flow.h File Reference

```
#include <iostream>
#include <map>
#include <vector>
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::Flow**
The flow object.

9.18 GenEvent.cc File Reference

```
#include "HepMC/GenEvent.h"
```

```
#include "HepMC/Version.h"
```

Namespaces

- namespace **HepMC**

9.19 GenEvent.h File Reference

```
#include "HepMC/GenVertex.h"
#include "HepMC/GenParticle.h"
#include "HepMC/WeightContainer.h"
#include "HepMC/HeavyIon.h"
#include "HepMC/PdfInfo.h"
#include <map>
#include <vector>
#include <algorithm>
#include <iostream>
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::GenEvent**
*The **GenEvent** (p. 54) class is the core of **HepMC** (p. 19).*
- class **HepMC::GenEvent::vertex_const_iterator**
const vertex iterator
- class **HepMC::GenEvent::vertex_iterator**
non-const vertex iterator
- class **HepMC::GenEvent::particle_const_iterator**
const particle iterator
- class **HepMC::GenEvent::particle_iterator**
non-const particle iterator

Functions

- template<class InputIterator, class OutputIterator, class Predicate> void **HepMC::copy_if**(InputIterator first, InputIterator last, OutputIterator out, Predicate pred)
define the type of iterator to use

9.20 GenParticle.cc File Reference

```
#include "HepMC/GenEvent.h"
#include "HepMC/GenVertex.h"
#include "HepMC/GenParticle.h"
#include <iomanip>
```

Namespaces

- namespace **HepMC**

Functions

- `std::ostream & HepMC::operator<< (std::ostream &ostr, const GenParticle &part)`
print particle

9.21 GenParticle.h File Reference

```
#include "HepMC/Flow.h"
#include "HepMC/Polarization.h"
#include "HepMC/SimpleVector.h"
#include <iostream>
#include <stdint.h>
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::GenParticle**

*The **GenParticle** (p. 86) class contains information about generated particles.*

Defines

- `#define hepmc_uint64_t uint64_t`

9.21.1 Define Documentation

9.21.1.1 `#define hepmc_uint64_t uint64_t`

Definition at line 37 of file GenParticle.h.

9.22 GenVertex.cc File Reference

```
#include "HepMC/GenParticle.h"
#include "HepMC/GenVertex.h"
#include "HepMC/GenEvent.h"
#include "HepMC/SearchVector.h"
#include <iomanip>
```

Namespaces

- namespace **HepMC**

Functions

- `std::ostream & HepMC::operator<< (std::ostream &ostr, const GenVertex &vtx)`
print vertex information

9.23 GenVertex.h File Reference

```
#include "HepMC/WeightContainer.h"
#include "HepMC/SimpleVector.h"
#include <iostream>
#include <iterator>
#include <vector>
#include <set>
#include <algorithm>
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::GenVertex**
GenVertex (p. 95) contains information about decay vertices.
- class **HepMC::GenVertex::edge_iterator**
edge iterator
- class **HepMC::GenVertex::vertex_iterator**
vertex iterator
- class **HepMC::GenVertex::particle_iterator**
particle iterator

Enumerations

- enum **HepMC::IteratorRange** {
 HepMC::parents, **HepMC::children**, **HepMC::family**, **HepMC::ancestors**,
 HepMC::descendants, **HepMC::relatives** }
type of iteration

9.24 HeavyIon.h File Reference

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::HeavyIon**

*The **HeavyIon** (p. 121) class stores information about heavy ions.*

9.25 HEPEVT_Wrapper.cc File Reference

```
#include "HepMC/HEPEVT_Wrapper.h"
```

Namespaces

- namespace **HepMC**

9.26 HEPEVT_Wrapper.h File Reference

```
#include <ctype.h>
#include <iostream>
#include <cstdio>
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::HEPEVT_Wrapper**
Generic Wrapper for the fortran HEPEVT common block.

Defines

- `#define HEPEVT_EntriesAllocation 10000`
- `#define hepevt hepevt_`

Variables

- `const unsigned int hepevt_bytes_allocation`
- `struct {`
 `char data [hepevt_bytes_allocation]`
 `} hepevt_`

9.26.1 Define Documentation

9.26.1.1 `#define hepevt hepevt_`

Definition at line 84 of file HEPEVT_Wrapper.h.

Referenced by `HepMC::HEPEVT_Wrapper::byte_num_to_double()`, `HepMC::HEPEVT_Wrapper::byte_num_to_int()`, and `HepMC::HEPEVT_Wrapper::write_byte_num()`.

9.26.1.2 `#define HEPEVT_EntriesAllocation 10000`

Definition at line 4 of file HEPEVT_Wrapper.h.

9.26.2 Variable Documentation

9.26.2.1 `char data[hepevt_bytes_allocation]`

Definition at line 81 of file HEPEVT_Wrapper.h.

9.26.2.2 struct { ... } hepevt_

9.26.2.3 const unsigned int hepevt_bytes_allocation

Initial value:

```
sizeof(long int) * ( 2 + 6 * HEPEVT_EntriesAllocation )  
+ sizeof(double) * ( 9 * HEPEVT_EntriesAllocation )
```

Definition at line 66 of file HEPEVT_Wrapper.h.

Referenced by HepMC::HEPEVT_Wrapper::byte_num_to_double(), HepMC::HEPEVT_Wrapper::byte_num_to_int(), and HepMC::HEPEVT_Wrapper::write_byte_num().

9.27 HepMC_CLHEP20.h File Reference

Typedefs

- `typedef CLHEP::HepLorentzVector HepLorentzVector`
- `typedef HepGeom::Normal3D< double > HepNormal3D`
- `typedef HepGeom::Point3D< double > HepPoint3D`

9.27.1 Typedef Documentation

9.27.1.1 `typedef CLHEP::HepLorentzVector HepLorentzVector`

Examples:

`example_BuildEventFromScratch.cc`, and `VectorConversion.h`.

Definition at line 13 of file `HepMC_CLHEP20.h`.

9.27.1.2 `typedef HepGeom::Normal3D<double> HepNormal3D`

Definition at line 16 of file `HepMC_CLHEP20.h`.

9.27.1.3 `typedef HepGeom::Point3D<double> HepPoint3D`

Definition at line 17 of file `HepMC_CLHEP20.h`.

9.28 HerwigWrapper.h File Reference

```
#include "HepMC/HerwigWrapper6_4.h"
```

9.29 HerwigWrapper6_4.h File Reference

```
#include <ctype.h>
```

Defines

- `#define hwproc hwproc_`
- `#define hwbmch hwbmch_`
- `#define hwevnt hwevnt_`
- `#define hwpram hwpram_`
- `#define hwigin hwigin_`
- `#define hwigup hwigup_`
- `#define hwuinc hwuinc_`
- `#define hwusta hwusta_`
- `#define hweini hweini_`
- `#define hwuine hwuine_`
- `#define hwepro hwepro_`
- `#define hwupro hwupro_`
- `#define hwbgen hwbgen_`
- `#define hwdhob hwdhob_`
- `#define hwcfor hwcfor_`
- `#define hwcdec hwcdec_`
- `#define hwdhad hwdhad_`
- `#define hwdhvy hwdhvy_`
- `#define hwmevt hwmevt_`
- `#define hwufne hwufne_`
- `#define hwefin hwefin_`
- `#define hwudpr hwudpr_`
- `#define hwuepr hwuepr_`
- `#define hwupup hwupup_`
- `#define hwegup hwegup_`
- `#define hwudat hwudat_`

Variables

- struct {
 double **EBEAM1**
 double **EBEAM2**
 double **PBEAM1**
 double **PBEAM2**
 int **IPROC**
 int **MAXEV**
 } **hwproc_**
- struct {
 char **PART1** [8]
 char **PART2** [8]
 } **hwbmch_**
- const int **herwig_hepevt_size** = 4000

- struct {
 - double **AVWGT**
 - double **EVWGT**
 - double **GAMWT**
 - double **TLOUT**
 - double **WBGST**
 - double **WGTMAX**
 - double **WGTSUM**
 - double **WSQSUM**
 - int **IDHW** [herwig_hepevt_size]
 - int **IERROR**
 - int **ISTAT**
 - int **LWEVT**
 - int **MAXER**
 - int **MAXPR**
 - int **NOWGT**
 - int **NRN** [2]
 - int **NUMER**
 - int **NUMERU**
 - int **NWGTS**
 - int **GENSOF**
 } **hwevnt_**
- struct {
 - double **AFCH** [2][16]
 - double **ALPHEM**
 - double **BILIM**
 - double **BETAF**
 - double **BTCLM**
 - double **CAFAC**
 - double **CFFAC**
 - double **CLMAX**
 - double **CLPOW**
 - double **CLSMR** [2]
 - double **CSPEED**
 - double **ENSO**
 - double **ETAMIX**
 - double **F0MIX**
 - double **F1MIX**
 - double **F2MIX**
 - double **GAMH**
 - double **GAMW**
 - double **GAMZ**
 - double **GAMZP**
 - double **GEV2NB**
 - double **H1MIX**
 - double **PDIQK**
 - double **PGSMX**
 - double **PGSPL** [4]
 - double **PHIMIX**
 - double **PIFAC**
 - double **PRSO**
 - double **PSPLT** [2]
 - double **PTRMS**

```
double PXRMS
double QCDL3
double QCDL5
double QCDLAM
double QDIQK
double QFCH [16]
double QG
double QSPAC
double QV
double SCABI
double SWEIN
double TMTOP
double VFCH [2][16]
double VCKM [3][3]
double VGCUT
double VQCUT
double VPCUT
double ZBINM
double EFFMIN
double OMHMIX
double ET2MIX
double PH3MIX
double GCUTME
int IOPREM
int IPRINT
int ISPAC
int LRSUD
int LWSUD
int MODPDF [2]
int NBTRY
int NCOLO
int NCTRY
int NDTRY
int NETRY
int NFLAV
int NGSPL
int NSTRU
int NSTRY
int NZBIN
int IOP4JT [2]
int NPRFMT
int AZSOFT
int AZSPIN
int CLDIR [2]
int HARDME
int NOSPAC
int PRNDEC
int PRVTX
int SOFTME
int ZPRIME
int PRNDEF
int PRNTEX
int PRNWEB
} hwpram_
```

9.29.1 Define Documentation

9.29.1.1 `#define hwbgen hwbgen_`

Examples:

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 87 of file `HerwigWrapper6_4.h`.

Referenced by `main()`.

9.29.1.2 `#define hwbmch hwbmch_`

Examples:

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 32 of file `HerwigWrapper6_4.h`.

Referenced by `main()`.

9.29.1.3 `#define hwcdec hwcdec_`

Examples:

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 90 of file `HerwigWrapper6_4.h`.

Referenced by `main()`.

9.29.1.4 `#define hwcfor hwcfor_`

Examples:

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 89 of file `HerwigWrapper6_4.h`.

Referenced by `main()`.

9.29.1.5 `#define hwdhad hwdhad_`

Examples:

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 91 of file `HerwigWrapper6_4.h`.

Referenced by `main()`.

9.29.1.6 `#define hwdhob hwdhob_`**Examples:**

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 88 of file `HerwigWrapper6_4.h`.

Referenced by `main()`.

9.29.1.7 `#define hwdhvy hwdhvy_`**Examples:**

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 92 of file `HerwigWrapper6_4.h`.

Referenced by `main()`.

9.29.1.8 `#define hwefn hwefn_`**Examples:**

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 95 of file `HerwigWrapper6_4.h`.

Referenced by `main()`.

9.29.1.9 `#define hwegup hwegup_`

Definition at line 100 of file `HerwigWrapper6_4.h`.

9.29.1.10 `#define hweini hweini_`**Examples:**

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 83 of file `HerwigWrapper6_4.h`.

Referenced by `main()`.

9.29.1.11 `#define hwepro hwepro_`**Examples:**

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 85 of file `HerwigWrapper6_4.h`.

Referenced by `main()`.

9.29.1.12 `#define hwevnt hwevnt_`**Examples:**

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 46 of file `HerwigWrapper6_4.h`.

Referenced by `main()`.

9.29.1.13 `#define hwigin hwigin_`**Examples:**

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 79 of file `HerwigWrapper6_4.h`.

Referenced by `main()`.

9.29.1.14 `#define hwigup hwigup_`

Definition at line 80 of file `HerwigWrapper6_4.h`.

9.29.1.15 `#define hwmevt hwmevt_`**Examples:**

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 93 of file `HerwigWrapper6_4.h`.

Referenced by `main()`.

9.29.1.16 `#define hwpram hwpram_`

Definition at line 74 of file `HerwigWrapper6_4.h`.

9.29.1.17 `#define hwproc hwproc_`**Examples:**

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 23 of file `HerwigWrapper6_4.h`.

Referenced by `main()`.

9.29.1.18 `#define hwudat hwudat_`**9.29.1.19** `#define hwudpr hwudpr_`

Definition at line 97 of file `HerwigWrapper6_4.h`.

9.29.1.20 `#define hwuepr hwuepr_`

Definition at line 98 of file HerwigWrapper6_4.h.

9.29.1.21 `#define hwufne hwufne_`**Examples:**

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 94 of file HerwigWrapper6_4.h.

Referenced by `main()`.

9.29.1.22 `#define hwuinc hwuinc_`**Examples:**

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 81 of file HerwigWrapper6_4.h.

Referenced by `main()`.

9.29.1.23 `#define hwuine hwuine_`**Examples:**

`example_MyHerwig.cc`, and `testHerwigCopies.cc`.

Definition at line 84 of file HerwigWrapper6_4.h.

Referenced by `main()`.

9.29.1.24 `#define hwupro hwupro_`

Definition at line 86 of file HerwigWrapper6_4.h.

9.29.1.25 `#define hwupup hwupup_`

Definition at line 99 of file HerwigWrapper6_4.h.

9.29.1.26 `#define hwusta hwusta_`

Definition at line 82 of file HerwigWrapper6_4.h.

9.29.2 Variable Documentation**9.29.2.1** `double AFCH[2][16]`

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.2 double ALPHEM

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.3 double AVWGT

Definition at line 40 of file HerwigWrapper6_4.h.

9.29.2.4 int AZSOFT

Definition at line 70 of file HerwigWrapper6_4.h.

9.29.2.5 int AZSPIN

Definition at line 70 of file HerwigWrapper6_4.h.

9.29.2.6 double B1LIM

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.7 double BETAF

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.8 double BTCLM

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.9 double CAFAC

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.10 double CFFAC

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.11 int CLDIR[2]

Definition at line 70 of file HerwigWrapper6_4.h.

9.29.2.12 double CLMAX

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.13 double CLPOW

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.14 double CLSMR[2]

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.15 double CSPEED

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.16 double EBEAM1

Definition at line 19 of file HerwigWrapper6_4.h.

9.29.2.17 double EBEAM2

Definition at line 19 of file HerwigWrapper6_4.h.

9.29.2.18 double EFFMIN

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.19 double ENSOF

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.20 double ET2MIX

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.21 double ETAMIX

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.22 double EVWGT

Definition at line 40 of file HerwigWrapper6_4.h.

9.29.2.23 double F0MIX

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.24 double F1MIX

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.25 double F2MIX

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.26 double GAMH

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.27 double GAMW

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.28 double GAMWT

Definition at line 40 of file HerwigWrapper6_4.h.

9.29.2.29 double GAMZ

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.30 double GAMZP

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.31 double GCUTME

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.32 int GENSOFF

Definition at line 43 of file HerwigWrapper6_4.h.

9.29.2.33 double GEV2NB

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.34 double H1MIX

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.35 int HARDME

Definition at line 70 of file HerwigWrapper6_4.h.

9.29.2.36 const int herwig_hepevt_size = 4000

Definition at line 37 of file HerwigWrapper6_4.h.

9.29.2.37 struct { ... } hwbmch_

9.29.2.38 struct { ... } hwevnt_

9.29.2.39 struct { ... } hwpram_

9.29.2.40 struct { ... } hwproc_

9.29.2.41 int IDHW[herwig_hepevt_size]

Definition at line 41 of file HerwigWrapper6_4.h.

9.29.2.42 int IERROR

Definition at line 41 of file HerwigWrapper6_4.h.

9.29.2.43 int IOP4JT[2]

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.44 int IOPREM

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.45 int IPRINT

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.46 int IPROC

Definition at line 20 of file HerwigWrapper6_4.h.

9.29.2.47 int ISPAC

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.48 int ISTAT

Definition at line 41 of file HerwigWrapper6_4.h.

9.29.2.49 int LRSUD

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.50 int LWEVT

Definition at line 41 of file HerwigWrapper6_4.h.

9.29.2.51 int LWSUD

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.52 int MAXER

Definition at line 41 of file HerwigWrapper6_4.h.

9.29.2.53 int MAXEV

Definition at line 20 of file HerwigWrapper6_4.h.

9.29.2.54 int MAXPR

Definition at line 41 of file HerwigWrapper6_4.h.

9.29.2.55 int MODPDF[2]

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.56 int NBTRY

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.57 int NCOLO

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.58 int NCTRY

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.59 int NDTRY

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.60 int NETRY

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.61 int NFLAV

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.62 int NGSPL

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.63 int NOSPAC

Definition at line 70 of file HerwigWrapper6_4.h.

9.29.2.64 int NOWGT

Definition at line 42 of file HerwigWrapper6_4.h.

9.29.2.65 int NPRFMT

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.66 int NRN[2]

Definition at line 42 of file HerwigWrapper6_4.h.

9.29.2.67 int NSTRU

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.68 int NSTRY

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.69 int NUMER

Definition at line 42 of file HerwigWrapper6_4.h.

9.29.2.70 int NUMERU

Definition at line 42 of file HerwigWrapper6_4.h.

9.29.2.71 int NWGTS

Definition at line 42 of file HerwigWrapper6_4.h.

9.29.2.72 int NZBIN

Definition at line 68 of file HerwigWrapper6_4.h.

9.29.2.73 double OMHMIX

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.74 char PART1[8]

Definition at line 29 of file HerwigWrapper6_4.h.

9.29.2.75 char PART2[8]

Definition at line 29 of file HerwigWrapper6_4.h.

9.29.2.76 double PBEAM1

Definition at line 19 of file HerwigWrapper6_4.h.

9.29.2.77 double PBEAM2

Definition at line 19 of file HerwigWrapper6_4.h.

9.29.2.78 double PDIQK

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.79 double PGSMX

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.80 double PGSPL[4]

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.81 double PH3MIX

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.82 double PHIMIX

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.83 double PIFAC

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.84 int PRNDEC

Definition at line 70 of file HerwigWrapper6_4.h.

9.29.2.85 int PRNDEF

Definition at line 70 of file HerwigWrapper6_4.h.

9.29.2.86 int PRNTEX

Definition at line 70 of file HerwigWrapper6_4.h.

9.29.2.87 int PRNWEB

Definition at line 70 of file HerwigWrapper6_4.h.

9.29.2.88 double PRSOF

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.89 int PRVTX

Definition at line 70 of file HerwigWrapper6_4.h.

9.29.2.90 double PSPLT[2]

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.91 double PTRMS

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.92 double PXRMS

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.93 double QC DL3

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.94 double QC DL5

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.95 double QC DLAM

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.96 double QDIQK

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.97 double QFCH[16]

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.98 double QG

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.99 double QSPAC

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.100 double QV

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.101 double SCABI

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.102 int SOFTME

Definition at line 70 of file HerwigWrapper6_4.h.

9.29.2.103 double SWEIN

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.104 double TLOUT

Definition at line 40 of file HerwigWrapper6_4.h.

9.29.2.105 double TMTOP

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.106 double VCKM[3][3]

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.107 double VFCH[2][16]

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.108 double VGCUT

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.109 double VPCUT

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.110 double VQCUT

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.111 double WBIGST

Definition at line 40 of file HerwigWrapper6_4.h.

9.29.2.112 double WGTMAX

Definition at line 40 of file HerwigWrapper6_4.h.

9.29.2.113 double WGTSUM

Definition at line 40 of file HerwigWrapper6_4.h.

9.29.2.114 double WSQSUM

Definition at line 40 of file HerwigWrapper6_4.h.

9.29.2.115 double ZBINM

Definition at line 62 of file HerwigWrapper6_4.h.

9.29.2.116 int ZPRIME

Definition at line 70 of file HerwigWrapper6_4.h.

9.30 initPythia.cc File Reference

```
#include "HepMC/PythiaWrapper.h"
```

Functions

- void initPythia ()

9.30.1 Function Documentation

9.30.1.1 void initPythia ()

Examples:

`example_MyPythia.cc`, `example_MyPythiaOnlyToHepMC.cc`, `example_MyPythiaRead.cc`, `example_MyPythiaWithEventSelection.cc`, `example_PythiaParticle.cc`, and `testPythiaCopies.cc`.

Definition at line 11 of file `initPythia.cc`.

References `pydat2`, `pydatr`, `pypars`, and `pysubs`.

Referenced by `main()`.

9.31 IO_Ascii.cc File Reference

```
#include "HepMC/IO_Ascii.h"  
#include "HepMC/GenEvent.h"  
#include "HepMC/ParticleDataTable.h"  
#include "HepMC/CommonIO.h"  
#include "HepMC/Version.h"
```

Namespaces

- namespace **HepMC**

9.32 `IO_Ascii.h` File Reference

```
#include <fstream>
#include <string>
#include <map>
#include <vector>
#include "HepMC/IO_BaseClass.h"
#include "HepMC/TempParticleMap.h"
#include "HepMC/CommonIO.h"
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::IO_Ascii**
IO_Ascii (p. 142) is used to read or write from an ascii file.

9.33 IO_AsciiParticles.cc File Reference

```
#include "HepMC/IO_AsciiParticles.h"  
#include "HepMC/GenEvent.h"  
#include "HepMC/ParticleDataTable.h"  
#include "HepMC/Version.h"
```

Namespaces

- namespace **HepMC**

9.34 IO_AsciiParticles.h File Reference

```
#include <fstream>
#include <string>
#include <map>
#include <vector>
#include "HepMC/IO_BaseClass.h"
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::IO_AsciiParticles**
event input/output in ascii format for eye and machine reading

9.35 IO_BaseClass.h File Reference

```
#include <iostream>
#include "HepMC/ParticleDataTable.h"
#include "HepMC/GenEvent.h"
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::IO_BaseClass**
*all input/output classes inherit from **IO_BaseClass** (p. 151)*

9.36 IO_ExtendedAscii.cc File Reference

```
#include "HepMC/IO_ExtendedAscii.h"  
#include "HepMC/GenEvent.h"  
#include "HepMC/ParticleDataTable.h"  
#include "HepMC/HeavyIon.h"  
#include "HepMC/PdfInfo.h"  
#include "HepMC/CommonIO.h"  
#include "HepMC/Version.h"
```

Namespaces

- namespace **HepMC**

9.37 IO_ExtendedAscii.h File Reference

```
#include <fstream>
#include <string>
#include <map>
#include <vector>
#include "HepMC/IO_BaseClass.h"
#include "HepMC/TempParticleMap.h"
#include "HepMC/CommonIO.h"
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::IO_ExtendedAscii**
IO_ExtendedAscii (p. 155) also deals with *HeavyIon* (p. 121) and *PdfInfo* (p. 220).

9.38 IO_GenEvent.cc File Reference

```
#include "HepMC/IO_GenEvent.h"  
#include "HepMC/GenEvent.h"  
#include "HepMC/ParticleDataTable.h"  
#include "HepMC/HeavyIon.h"  
#include "HepMC/PdfInfo.h"  
#include "HepMC/CommonIO.h"  
#include "HepMC/Version.h"
```

Namespaces

- namespace **HepMC**

9.39 IO_GenEvent.h File Reference

```
#include <fstream>
#include <string>
#include <map>
#include <vector>
#include "HepMC/IO_BaseClass.h"
#include "HepMC/TempParticleMap.h"
#include "HepMC/CommonIO.h"
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::IO_GenEvent**
IO_GenEvent (p. 162) also deals with *HeavyIon* (p. 121) and *PdfInfo* (p. 220).

9.40 IO_HEPEVT.cc File Reference

```
#include "HepMC/IO_HEPEVT.h"  
#include "HepMC/GenEvent.h"  
#include <cstdio>
```

Namespaces

- namespace **HepMC**

9.41 IO_HEPEVT.h File Reference

```
#include <map>
#include <vector>
#include "HepMC/IO_BaseClass.h"
#include "HepMC/HEPEVT_Wrapper.h"
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::IO_HEPEVT**
HEPEVT IO class.

9.42 IO_HERWIG.cc File Reference

```
#include "HepMC/IO_HERWIG.h"  
#include "HepMC/GenEvent.h"  
#include <cstdio>
```

Namespaces

- namespace **HepMC**

9.43 IO_HERWIG.h File Reference

```
#include <set>
#include <vector>
#include "HepMC/IO_BaseClass.h"
#include "HepMC/HEPEVT_Wrapper.h"
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::IO_HERWIG**
IO_HERWIG (p. 176) is used to get Herwig information.

9.44 IO_PDG_ParticleDataTable.cc File Reference

```
#include <ctype.h>
#include <string>
#include <vector>
#include <cstdlib>
#include "HepMC/IO_PDG_ParticleDataTable.h"
```

Namespaces

- namespace **HepMC**

9.45 IO_PDG_ParticleDataTable.h File Reference

```
#include "HepMC/IO_BaseClass.h"  
#include <fstream>
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::IO_PDG_ParticleDataTable**
*an example **ParticleDataTable** (p. 213) IO method*

9.46 is _arithmetic.h File Reference

Namespaces

- namespace **HepMC**
- namespace **detail**
- namespace **HepMC::detail**

Classes

- struct **HepMC::detail::is _arithmetic**< T >
undefined and therefore non-arithmetic
- struct **HepMC::detail::is _arithmetic**< char >
character is arithmetic
- struct **HepMC::detail::is _arithmetic**< unsigned char >
unsigned character is arithmetic
- struct **HepMC::detail::is _arithmetic**< signed char >
signed character is arithmetic
- struct **HepMC::detail::is _arithmetic**< short >
short is arithmetic
- struct **HepMC::detail::is _arithmetic**< unsigned short >
unsigned short is arithmetic
- struct **HepMC::detail::is _arithmetic**< int >
int is arithmetic
- struct **HepMC::detail::is _arithmetic**< unsigned int >
unsigned int is arithmetic
- struct **HepMC::detail::is _arithmetic**< long >
long is arithmetic
- struct **HepMC::detail::is _arithmetic**< unsigned long >
unsigned long is arithmetic
- struct **HepMC::detail::is _arithmetic**< float >
float is arithmetic
- struct **HepMC::detail::is _arithmetic**< double >
double is arithmetic
- struct **HepMC::detail::is _arithmetic**< long double >
long double is arithmetic

9.47 IsGoodEvent.h File Reference

Classes

- class **IsGoodEvent**

9.48 list_of_examples.cc File Reference

9.49 ParticleData.cc File Reference

```
#include "HepMC/ParticleData.h"
#include <cstdio>
```

Namespaces

- namespace **HepMC**

Functions

- `std::ostream & HepMC::operator<< (std::ostream &ostr, const ParticleData &pdata)`
write to ostr
- `double HepMC::lifetime_from_width (double width)`
set lifetime from width

9.50 ParticleData.h File Reference

```
#include <iostream>
#include <string>
#include <cmath>
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::ParticleData**
*an example **ParticleData** (p. 206) class*

Functions

- double **HepMC::lifetime_from_width** (double width)
set lifetime from width

Variables

- static const double **HepMC::HepMC_hbarc**
 *$\hbar * c \rightarrow$ calculated with units of [mm*GeV]*

9.51 ParticleDataTable.h File Reference

```
#include <iostream>
#include <map>
#include <cstdio>
#include "HepMC/ParticleData.h"
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::ParticleDataTable**
*an example **ParticleDataTable** (p. 213) class*

9.52 PdfInfo.h File Reference

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::PdfInfo**
*The **PdfInfo** (p. 220) class stores PDF information.*

9.53 Polarization.cc File Reference

```
#include "HepMC/Polarization.h"
```

Namespaces

- namespace **HepMC**

Functions

- `std::ostream & HepMC::operator<< (std::ostream &ostr, const Polarization &polar)`
print polarization information

9.54 Polarization.h File Reference

```
#include "HepMC/SimpleVector.h"
#include <iostream>
#include <cmath>
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::Polarization**

*The **Polarization** (p. 225) class stores θ and ϕ for a **GenParticle** (p. 86).*

Variables

- static const double **HepMC::HepMC_pi** = 3.14159265358979323846

9.55 PythiaHelper.h File Reference

Functions

- `void initPythia ()`

9.55.1 Function Documentation

9.55.1.1 `void initPythia ()`

Definition at line 11 of file `initPythia.cc`.

References `pydat2`, `pydatr`, `pypars`, and `pysubs`.

Referenced by `main()`.

9.56 PythiaWrapper.h File Reference

```
#include "HepMC/PythiaWrapper6_2.h"
```

9.57 PythiaWrapper5_720.h File Reference

```
#include <ctype.h>
```

Defines

- `#define initpydata initpydata_`
- `#define lujets lujets_`
- `#define ludat1 ludat1_`
- `#define ludat2 ludat2_`
- `#define ludat3 ludat3_`
- `#define ludatr ludatr_`
- `#define pysubs pysubs_`
- `#define pypars pypars_`
- `#define pyint1 pyint1_`
- `#define pyint2 pyint2_`
- `#define pyint5 pyint5_`
- `#define luhepc luhepc_`
- `#define pyinit pyinit_`
- `#define lulist lulist_`
- `#define pystat pystat_`
- `#define pyevnt pyevnt_`
- `#define ludata ludata_`
- `#define pydata pydata_`

Variables

- struct {
 int **n**
 int **k** [5][**pyjets_**maxn]
 float **p** [5][**pyjets_**maxn]
 float **v** [5][**pyjets_**maxn]
} **lujets_**
- struct {
 int **mstu** [200]
 float **paru** [200]
 int **mstj** [200]
 float **parj** [200]
} **ludat1_**
- struct {
 int **kchg** [3][500]
 float **pmas** [4][500]
 float **parf** [2000]
 float **vckm** [4][4]
} **ludat2_**

- struct {
 int **mdcy** [3][500]
 int **mdme** [2][2000]
 float **brat** [2000]
 int **kfdp** [5][2000]
 } **ludat3_**

- struct {
 int **mrlu** [6]
 float **rrlu** [100]
 } **ludatr_**

- struct {
 int **msel**
 int **msub** [200]
 int **kfin** [81][2]
 float **ckin** [200]
 } **pysubs_**

- struct {
 int **mstp** [200]
 float **parp** [200]
 int **msti** [200]
 float **pari** [200]
 } **pypars_**

- struct {
 int **mint** [400]
 float **vint** [400]
 } **pyint1_**

- struct {
 int **iset** [200]
 int **kfpr** [2][200]
 float **coef** [20][200]
 int **icol** [2][4][40]
 } **pyint2_**

- struct {
 int **ngen** [3][201][3]
 float **xsec** [3][201]
 } **pyint5_**

9.57.1 Define Documentation

9.57.1.1 #define initpydata initpydata_

Definition at line 26 of file PythiaWrapper5_720.h.

9.57.1.2 #define ludat1 ludat1_

Definition at line 47 of file PythiaWrapper5_720.h.

9.57.1.3 #define ludat2 ludat2_

Definition at line 53 of file PythiaWrapper5_720.h.

9.57.1.4 #define ludat3 ludat3_

Definition at line 60 of file PythiaWrapper5_720.h.

9.57.1.5 #define ludata ludata_

Definition at line 109 of file PythiaWrapper5_720.h.

9.57.1.6 #define ludatr ludatr_

Definition at line 66 of file PythiaWrapper5_720.h.

9.57.1.7 #define luhepc luhepc_

Definition at line 104 of file PythiaWrapper5_720.h.

9.57.1.8 #define lujets lujets_

Definition at line 39 of file PythiaWrapper5_720.h.

9.57.1.9 #define lulist lulist_

Definition at line 106 of file PythiaWrapper5_720.h.

9.57.1.10 #define pydata pydata_**9.57.1.11 #define pyevnt pyevnt_**

Definition at line 108 of file PythiaWrapper5_720.h.

9.57.1.12 #define pyinit pyinit_

Definition at line 105 of file PythiaWrapper5_720.h.

9.57.1.13 #define pyint1 pyint1_

Definition at line 86 of file PythiaWrapper5_720.h.

9.57.1.14 `#define pyint2 pyint2_`

Definition at line 93 of file PythiaWrapper5_720.h.

9.57.1.15 `#define pyint5 pyint5_`

Definition at line 99 of file PythiaWrapper5_720.h.

9.57.1.16 `#define pypars pypars_`

Examples:

`example_MyPythia.cc`, `example_MyPythiaOnlyToHepMC.cc`, `example_MyPythiaWithEventSelection.cc`, and `testPythiaCopies.cc`.

Definition at line 80 of file PythiaWrapper5_720.h.

Referenced by `initPythia()`, and `main()`.

9.57.1.17 `#define pystat pystat_`

Definition at line 107 of file PythiaWrapper5_720.h.

9.57.1.18 `#define pysubs pysubs_`

Definition at line 72 of file PythiaWrapper5_720.h.

Referenced by `initPythia()`.

9.57.2 Variable Documentation

9.57.2.1 `float brat[2000]`

Definition at line 57 of file PythiaWrapper5_720.h.

9.57.2.2 `float ckin[200]`

Definition at line 70 of file PythiaWrapper5_720.h.

9.57.2.3 `float coef[20][200]`

Definition at line 90 of file PythiaWrapper5_720.h.

9.57.2.4 `int icol[2][4][40]`

Definition at line 91 of file PythiaWrapper5_720.h.

9.57.2.5 `int iset[200]`

Definition at line 89 of file PythiaWrapper5_720.h.

9.57.2.6 int k[5][pyjets_maxn]

Definition at line 36 of file PythiaWrapper5_720.h.

9.57.2.7 int kchg[3][500]

Definition at line 50 of file PythiaWrapper5_720.h.

9.57.2.8 int kfdp[5][2000]

Definition at line 58 of file PythiaWrapper5_720.h.

9.57.2.9 int kfin[81][2]

Definition at line 69 of file PythiaWrapper5_720.h.

9.57.2.10 int kfpr[2][200]

Definition at line 89 of file PythiaWrapper5_720.h.

9.57.2.11 struct { ... } ludat1_**9.57.2.12 struct { ... } ludat2_****9.57.2.13 struct { ... } ludat3_****9.57.2.14 struct { ... } ludatr_****9.57.2.15 struct { ... } lujets_****9.57.2.16 int mdcy[3][500]**

Definition at line 56 of file PythiaWrapper5_720.h.

9.57.2.17 int mdme[2][2000]

Definition at line 56 of file PythiaWrapper5_720.h.

9.57.2.18 int mint[400]

Definition at line 83 of file PythiaWrapper5_720.h.

9.57.2.19 int mrlu[6]

Definition at line 63 of file PythiaWrapper5_720.h.

9.57.2.20 int msel

Definition at line 69 of file PythiaWrapper5_720.h.

9.57.2.21 int msti[200]

Definition at line 77 of file PythiaWrapper5_720.h.

9.57.2.22 int mstj[200]

Definition at line 44 of file PythiaWrapper5_720.h.

9.57.2.23 int mstp[200]

Definition at line 75 of file PythiaWrapper5_720.h.

9.57.2.24 int mstu[200]

Definition at line 42 of file PythiaWrapper5_720.h.

9.57.2.25 int msub[200]

Definition at line 69 of file PythiaWrapper5_720.h.

9.57.2.26 int n

Definition at line 36 of file PythiaWrapper5_720.h.

9.57.2.27 int ngen[3][201][3]

Definition at line 96 of file PythiaWrapper5_720.h.

9.57.2.28 float p[5][pyjets_maxn]**Examples:**

example_BuildEventFromScratch.cc, example_EventSelection.cc, example_My-PythiaWithEventSelection.cc, and example_UsingIterators.cc.

Definition at line 37 of file PythiaWrapper5_720.h.

Referenced by HepMC::TempParticleMap::addEndParticle(), HepMC::already_in_vector(), HepMC::IO_HERWIG::build_end_vertex(), HepMC::IO_HEPEVT::build_end_vertex(), HepMC::IO_HERWIG::build_particle(), HepMC::IO_HEPEVT::build_particle(), HepMC::IO_HERWIG::build_production_vertex(), HepMC::IO_HEPEVT::build_production_vertex(), HepMC::Flow::connected_partners(), HepMC::Flow::dangling_connected_partners(), HepMC::GenVertex::edge_iterator::edge_iterator(), HepMC::TempParticleMap::end_vertex(),

HepMC::IO_HERWIG::find_in_map(), HepMC::IO_HEPEVT::find_in_map(), HepMC::GenEvent::GenEvent(), IsPhoton(), IsWBoson(), main(), HepMC::ParticleDataTable::make_antiparticles_from_particles(), HepMC::ParticleDataTable::merge_table(), HepMC::not_in_vector(), IsFinalState::operator(), IsGoodEvent::operator(), IsStateFinal::operator(), IsW_Boson::operator(), IsPhoton::operator(), IsGoodEventMyPythia::operator(), IsEventGood::operator(), HepMC::GenVertex::edge_iterator::operator=(), HepMC::CommonIO::read_io_ascii(), HepMC::CommonIO::read_io_extendedascii(), HepMC::CommonIO::read_io_genevent(), HepMC::IO_GenEvent::read_particle(), HepMC::CommonIO::read_particle(), HepMC::GenEvent::remove_barcode(), HepMC::GenEvent::set_barcode(), HepMC::GenEvent::set_pdf_info(), HepMC::GenEvent::valid_beam_particles(), HepMC::IO_GenEvent::write_beam_particles(), HepMC::IO_ExtendedAscii::write_beam_particles(), HepMC::IO_HEPEVT::write_event(), HepMC::IO_GenEvent::write_particle(), HepMC::IO_ExtendedAscii::write_particle(), and HepMC::IO_Ascii::write_particle().

9.57.2.29 float parf[2000]

Definition at line 51 of file PythiaWrapper5_720.h.

9.57.2.30 float pari[200]

Definition at line 78 of file PythiaWrapper5_720.h.

9.57.2.31 float parj[200]

Definition at line 45 of file PythiaWrapper5_720.h.

9.57.2.32 float parp[200]

Definition at line 76 of file PythiaWrapper5_720.h.

9.57.2.33 float paru[200]

Definition at line 43 of file PythiaWrapper5_720.h.

9.57.2.34 float pmas[4][500]

Definition at line 51 of file PythiaWrapper5_720.h.

9.57.2.35 struct { ... } pyint1_

9.57.2.36 struct { ... } pyint2_

9.57.2.37 struct { ... } pyint5_

9.57.2.38 struct { ... } pypars_

9.57.2.39 struct { ... } pysubs_

9.57.2.40 float rrlu[100]

Definition at line 64 of file PythiaWrapper5_720.h.

9.57.2.41 float v[5][pyjets_maxn]

Examples:

`example_UsingIterators.cc`, and `VectorConversion.h`.

Definition at line 37 of file PythiaWrapper5_720.h.

Referenced by `HepMC::compareVertices()`, `convertTo()`, `HepMC::GenEvent::GenEvent()`, `main()`, `HepMC::CommonIO::read_io_ascii()`, `HepMC::CommonIO::read_io_extendedascii()`, `HepMC::CommonIO::read_io_genevent()`, `HepMC::CommonIO::read_vertex()`, `HepMC::GenEvent::remove_barcode()`, `HepMC::GenEvent::set_barcode()`, `HepMC::IO_HEPEVT::write_event()`, `HepMC::IO_GenEvent::write_event()`, `HepMC::IO_ExtendedAscii::write_event()`, `HepMC::IO_Ascii::write_event()`, `HepMC::IO_GenEvent::write_vertex()`, `HepMC::IO_ExtendedAscii::write_vertex()`, and `HepMC::IO_Ascii::write_vertex()`.

9.57.2.42 float vckm[4][4]

Definition at line 51 of file PythiaWrapper5_720.h.

9.57.2.43 float vint[400]

Definition at line 84 of file PythiaWrapper5_720.h.

9.57.2.44 float xsec[3][201]

Definition at line 97 of file PythiaWrapper5_720.h.

9.58 PythiaWrapper6_152.h File Reference

```
#include <ctype.h>
```

```
#include <cstring>
```

Defines

- `#define initpydata initpydata_`
- `#define pyjets pyjets_`
- `#define pydat1 pydat1_`
- `#define pydat2 pydat2_`
- `#define pydat3 pydat3_`
- `#define pydatr pydatr_`
- `#define pysubs pysubs_`
- `#define pypars pypars_`
- `#define pyint1 pyint1_`
- `#define pyint2 pyint2_`
- `#define pyint5 pyint5_`
- `#define pyhepc pyhepc_`
- `#define pyinit pyinit_`
- `#define pylist pylist_`
- `#define pystat pystat_`
- `#define pyevnt pyevnt_`
- `#define pydata pydata_`

Variables

- struct {
 int **n**
 int **npad**
 int **k** [5][**pyjets_**maxn]
 double **p** [5][**pyjets_**maxn]
 double **v** [5][**pyjets_**maxn]
 } **pyjets_**
- struct {
 int **mstu** [200]
 double **paru** [200]
 int **mstj** [200]
 double **parj** [200]
 } **pydat1_**
- struct {
 int **kchg** [4][500]
 double **pmas** [4][500]
 double **parf** [2000]
 double **vckm** [4][4]
 } **pydat2_**

- struct {
 int **mdcy** [3][500]
 int **mdme** [2][4000]
 double **brat** [4000]
 int **kfdp** [5][4000]
 } **pydat3_**
- struct {
 int **mrpy** [6]
 double **rrpy** [100]
 } **pydatr_**
- struct {
 int **msel**
 int **mselpd**
 int **msub** [500]
 int **kfn** [81][2]
 double **ckin** [200]
 } **pysubs_**
- struct {
 int **mstp** [200]
 double **parp** [200]
 int **msti** [200]
 double **pari** [200]
 } **pypars_**
- struct {
 int **mint** [400]
 double **vint** [400]
 } **pyint1_**
- struct {
 int **iset** [500]
 int **kfpr** [2][500]
 double **coef** [20][500]
 int **icol** [2][4][40]
 } **pyint2_**
- struct {
 int **ngenpd**
 int **ngen** [3][501]
 double **xsec** [3][501]
 } **pyint5_**

9.58.1 Define Documentation

9.58.1.1 #define initpydata initpydata_

Definition at line 27 of file PythiaWrapper6_152.h.

9.58.1.2 `#define pydat1 pydat1_`

Definition at line 48 of file PythiaWrapper6_152.h.

9.58.1.3 `#define pydat2 pydat2_`

Definition at line 54 of file PythiaWrapper6_152.h.

Referenced by `initPythia()`.

9.58.1.4 `#define pydat3 pydat3_`

Definition at line 61 of file PythiaWrapper6_152.h.

9.58.1.5 `#define pydata pydata_`**9.58.1.6** `#define pydatr pydatr_`

Definition at line 67 of file PythiaWrapper6_152.h.

Referenced by `initPythia()`.

9.58.1.7 `#define pyevnt pyevnt_`

Definition at line 109 of file PythiaWrapper6_152.h.

9.58.1.8 `#define pyhepc pyhepc_`

Definition at line 105 of file PythiaWrapper6_152.h.

9.58.1.9 `#define pyinit pyinit_`

Definition at line 106 of file PythiaWrapper6_152.h.

9.58.1.10 `#define pyint1 pyint1_`

Definition at line 87 of file PythiaWrapper6_152.h.

9.58.1.11 `#define pyint2 pyint2_`

Definition at line 94 of file PythiaWrapper6_152.h.

9.58.1.12 `#define pyint5 pyint5_`

Definition at line 100 of file PythiaWrapper6_152.h.

9.58.1.13 `#define pyjets pyjets_`

Definition at line 40 of file PythiaWrapper6_152.h.

9.58.1.14 `#define pylist pylist_`

Definition at line 107 of file PythiaWrapper6_152.h.

9.58.1.15 `#define pypars pypars_`

Definition at line 81 of file PythiaWrapper6_152.h.

9.58.1.16 `#define pystat pystat_`

Definition at line 108 of file PythiaWrapper6_152.h.

9.58.1.17 `#define pysubs pysubs_`

Definition at line 73 of file PythiaWrapper6_152.h.

9.58.2 Variable Documentation**9.58.2.1** `double brat[4000]`

Definition at line 58 of file PythiaWrapper6_152.h.

9.58.2.2 `double ckin[200]`

Definition at line 71 of file PythiaWrapper6_152.h.

9.58.2.3 `double coef[20][500]`

Definition at line 91 of file PythiaWrapper6_152.h.

9.58.2.4 `int icol[2][4][40]`

Definition at line 92 of file PythiaWrapper6_152.h.

9.58.2.5 `int iset[500]`

Definition at line 90 of file PythiaWrapper6_152.h.

9.58.2.6 `int k[5][pyjets_maxn]`

Definition at line 37 of file PythiaWrapper6_152.h.

9.58.2.7 `int kchg[4][500]`

Definition at line 51 of file PythiaWrapper6_152.h.

9.58.2.8 int kfdp[5][4000]

Definition at line 59 of file PythiaWrapper6_152.h.

9.58.2.9 int kfin[81][2]

Definition at line 70 of file PythiaWrapper6_152.h.

9.58.2.10 int kfpr[2][500]

Definition at line 90 of file PythiaWrapper6_152.h.

9.58.2.11 int mdcy[3][500]

Definition at line 57 of file PythiaWrapper6_152.h.

9.58.2.12 int mdme[2][4000]

Definition at line 57 of file PythiaWrapper6_152.h.

9.58.2.13 int mint[400]

Definition at line 84 of file PythiaWrapper6_152.h.

9.58.2.14 int mrpy[6]

Definition at line 64 of file PythiaWrapper6_152.h.

9.58.2.15 int msel

Definition at line 70 of file PythiaWrapper6_152.h.

9.58.2.16 int mselpd

Definition at line 70 of file PythiaWrapper6_152.h.

9.58.2.17 int msti[200]

Definition at line 78 of file PythiaWrapper6_152.h.

9.58.2.18 int mstj[200]

Definition at line 45 of file PythiaWrapper6_152.h.

9.58.2.19 int mstp[200]

Definition at line 76 of file PythiaWrapper6_152.h.

9.58.2.20 int mstu[200]

Definition at line 43 of file PythiaWrapper6_152.h.

9.58.2.21 int msub[500]

Definition at line 70 of file PythiaWrapper6_152.h.

9.58.2.22 int n

Definition at line 37 of file PythiaWrapper6_152.h.

9.58.2.23 int ngen[3][501]

Definition at line 97 of file PythiaWrapper6_152.h.

9.58.2.24 int ngenpd

Definition at line 97 of file PythiaWrapper6_152.h.

9.58.2.25 int npad

Definition at line 37 of file PythiaWrapper6_152.h.

9.58.2.26 double p[5][pyjets_maxn]

Definition at line 38 of file PythiaWrapper6_152.h.

9.58.2.27 double parf[2000]

Definition at line 52 of file PythiaWrapper6_152.h.

9.58.2.28 double pari[200]

Definition at line 79 of file PythiaWrapper6_152.h.

9.58.2.29 double parj[200]

Definition at line 46 of file PythiaWrapper6_152.h.

9.58.2.30 double parp[200]

Definition at line 77 of file PythiaWrapper6_152.h.

9.58.2.31 double paru[200]

Definition at line 44 of file PythiaWrapper6_152.h.

9.58.2.32 `double pmas[4][500]`

Definition at line 52 of file PythiaWrapper6_152.h.

9.58.2.33 `struct { ... } pydat1_`

9.58.2.34 `struct { ... } pydat2_`

9.58.2.35 `struct { ... } pydat3_`

9.58.2.36 `struct { ... } pydatr_`

9.58.2.37 `struct { ... } pyint1_`

9.58.2.38 `struct { ... } pyint2_`

9.58.2.39 `struct { ... } pyint5_`

9.58.2.40 `struct { ... } pyjets_`

9.58.2.41 `struct { ... } pypars_`

9.58.2.42 `struct { ... } pysubs_`

9.58.2.43 `double rrp[100]`

Definition at line 65 of file PythiaWrapper6_152.h.

9.58.2.44 `double v[5][pyjets_maxn]`

Definition at line 38 of file PythiaWrapper6_152.h.

9.58.2.45 `double vckm[4][4]`

Definition at line 52 of file PythiaWrapper6_152.h.

9.58.2.46 `double vint[400]`

Definition at line 85 of file PythiaWrapper6_152.h.

9.58.2.47 `double xsec[3][501]`

Definition at line 98 of file PythiaWrapper6_152.h.

9.59 PythiaWrapper6_152_WIN32.h File Reference

9.60 PythiaWrapper6_2.h File Reference

```
#include <ctype.h>
```

```
#include <cstring>
```

Defines

- `#define initpydata initpydata_`
- `#define pyjets pyjets_`
- `#define pydat1 pydat1_`
- `#define pydat2 pydat2_`
- `#define pydat3 pydat3_`
- `#define pydatr pydatr_`
- `#define pysubs pysubs_`
- `#define pypars pypars_`
- `#define pyint1 pyint1_`
- `#define pyint2 pyint2_`
- `#define pyint5 pyint5_`
- `#define pyhepc pyhepc_`
- `#define pyinit pyinit_`
- `#define pylist pylist_`
- `#define pystat pystat_`
- `#define pyevnt pyevnt_`
- `#define upinit upinit_`
- `#define upevnt upevnt_`
- `#define pydata pydata_`

Functions

- `void initpydata (void)`

Variables

- `const int pyjets_maxn = 4000`
- `struct {
 int n
 int npad
 int k [5][pyjets_maxn]
 double p [5][pyjets_maxn]
 double v [5][pyjets_maxn]
} pyjets_`
- `struct {
 int mstu [200]
 double paru [200]
 int mstj [200]
 double parj [200]
} pydat1_`

- struct {
 int **kchg** [4][500]
 double **pmas** [4][500]
 double **parf** [2000]
 double **vckm** [4][4]
 } **pydat2_**
- struct {
 int **mdcy** [3][500]
 int **mdme** [2][8000]
 double **brat** [8000]
 int **kfdp** [5][8000]
 } **pydat3_**
- struct {
 int **mrpy** [6]
 double **rrpy** [100]
 } **pydatr_**
- struct {
 int **msel**
 int **mselpd**
 int **msub** [500]
 int **kfin** [81][2]
 double **ckin** [200]
 } **pysubs_**
- struct {
 int **mstp** [200]
 double **parp** [200]
 int **msti** [200]
 double **pari** [200]
 } **pypars_**
- struct {
 int **mint** [400]
 double **vint** [400]
 } **pyint1_**
- struct {
 int **iset** [500]
 int **kfpr** [2][500]
 double **coef** [20][500]
 int **icol** [2][4][40]
 } **pyint2_**
- struct {
 int **ngenpd**
 int **ngen** [3][501]
 double **xsec** [3][501]
 } **pyint5_**

9.60.1 Define Documentation

9.60.1.1 `#define initpydata initpydata_`

Definition at line 30 of file PythiaWrapper6_2.h.

9.60.1.2 `#define pydat1 pydat1_`

Definition at line 52 of file PythiaWrapper6_2.h.

9.60.1.3 `#define pydat2 pydat2_`

Definition at line 60 of file PythiaWrapper6_2.h.

9.60.1.4 `#define pydat3 pydat3_`

Definition at line 69 of file PythiaWrapper6_2.h.

9.60.1.5 `#define pydata pydata_`

9.60.1.6 `#define pydatr pydatr_`

Definition at line 77 of file PythiaWrapper6_2.h.

9.60.1.7 `#define pyevnt pyevnt_`

Definition at line 129 of file PythiaWrapper6_2.h.

9.60.1.8 `#define pyhepc pyhepc_`

Definition at line 125 of file PythiaWrapper6_2.h.

9.60.1.9 `#define pyinit pyinit_`

Definition at line 126 of file PythiaWrapper6_2.h.

9.60.1.10 `#define pyint1 pyint1_`

Definition at line 103 of file PythiaWrapper6_2.h.

9.60.1.11 `#define pyint2 pyint2_`

Definition at line 112 of file PythiaWrapper6_2.h.

9.60.1.12 `#define pyint5 pyint5_`

Definition at line 120 of file PythiaWrapper6_2.h.

9.60.1.13 `#define pyjets pyjets_`

Definition at line 42 of file PythiaWrapper6_2.h.

9.60.1.14 `#define pylist pylist_`

Definition at line 127 of file PythiaWrapper6_2.h.

9.60.1.15 `#define pypars pypars_`

Definition at line 95 of file PythiaWrapper6_2.h.

9.60.1.16 `#define pystat pystat_`

Definition at line 128 of file PythiaWrapper6_2.h.

9.60.1.17 `#define pysubs pysubs_`

Definition at line 85 of file PythiaWrapper6_2.h.

9.60.1.18 `#define upevnt upevnt_`

Definition at line 131 of file PythiaWrapper6_2.h.

9.60.1.19 `#define upinit upinit_`

Definition at line 130 of file PythiaWrapper6_2.h.

9.60.2 Function Documentation**9.60.2.1** `void initpydata (void)`**9.60.3** Variable Documentation**9.60.3.1** `double brat[8000]`

Definition at line 65 of file PythiaWrapper6_2.h.

9.60.3.2 `double ckin[200]`

Definition at line 82 of file PythiaWrapper6_2.h.

9.60.3.3 `double coef[20][500]`

Definition at line 108 of file PythiaWrapper6_2.h.

9.60.3.4 int icol[2][4][40]

Definition at line 109 of file PythiaWrapper6_2.h.

9.60.3.5 int iset[500]

Definition at line 107 of file PythiaWrapper6_2.h.

9.60.3.6 int k[5][pyjets_maxn]

Definition at line 38 of file PythiaWrapper6_2.h.

9.60.3.7 int kchg[4][500]

Definition at line 56 of file PythiaWrapper6_2.h.

9.60.3.8 int kfdp[5][8000]

Definition at line 66 of file PythiaWrapper6_2.h.

9.60.3.9 int kfn[81][2]

Definition at line 81 of file PythiaWrapper6_2.h.

9.60.3.10 int kfpr[2][500]

Definition at line 107 of file PythiaWrapper6_2.h.

9.60.3.11 int mdcy[3][500]

Definition at line 64 of file PythiaWrapper6_2.h.

9.60.3.12 int mdme[2][8000]

Definition at line 64 of file PythiaWrapper6_2.h.

9.60.3.13 int mint[400]

Definition at line 99 of file PythiaWrapper6_2.h.

9.60.3.14 int mrpy[6]

Definition at line 73 of file PythiaWrapper6_2.h.

9.60.3.15 int msel

Definition at line 81 of file PythiaWrapper6_2.h.

9.60.3.16 int mselpd

Definition at line 81 of file PythiaWrapper6_2.h.

9.60.3.17 int msti[200]

Definition at line 91 of file PythiaWrapper6_2.h.

9.60.3.18 int mstj[200]

Definition at line 48 of file PythiaWrapper6_2.h.

9.60.3.19 int mstp[200]

Definition at line 89 of file PythiaWrapper6_2.h.

9.60.3.20 int mstu[200]

Definition at line 46 of file PythiaWrapper6_2.h.

9.60.3.21 int msub[500]

Definition at line 81 of file PythiaWrapper6_2.h.

9.60.3.22 int n

Definition at line 38 of file PythiaWrapper6_2.h.

9.60.3.23 int ngen[3][501]

Definition at line 116 of file PythiaWrapper6_2.h.

9.60.3.24 int ngenpd

Definition at line 116 of file PythiaWrapper6_2.h.

9.60.3.25 int npad

Definition at line 38 of file PythiaWrapper6_2.h.

9.60.3.26 double p[5][pyjets_maxn]

Definition at line 39 of file PythiaWrapper6_2.h.

9.60.3.27 double parf[2000]

Definition at line 57 of file PythiaWrapper6_2.h.

9.60.3.28 double pari[200]

Definition at line 92 of file PythiaWrapper6_2.h.

9.60.3.29 double parj[200]

Definition at line 49 of file PythiaWrapper6_2.h.

9.60.3.30 double parp[200]

Definition at line 90 of file PythiaWrapper6_2.h.

9.60.3.31 double paru[200]

Definition at line 47 of file PythiaWrapper6_2.h.

9.60.3.32 double pmas[4][500]

Definition at line 57 of file PythiaWrapper6_2.h.

9.60.3.33 struct { ... } pydat1_**9.60.3.34 struct { ... } pydat2_****9.60.3.35 struct { ... } pydat3_****9.60.3.36 struct { ... } pydatr_****9.60.3.37 struct { ... } pyint1_****9.60.3.38 struct { ... } pyint2_****9.60.3.39 struct { ... } pyint5_****9.60.3.40 struct { ... } pyjets_****9.60.3.41 const int pyjets_maxn = 4000**

Definition at line 35 of file PythiaWrapper6_2.h.

9.60.3.42 struct { ... } pypars_**9.60.3.43 struct { ... } pysubs_****9.60.3.44 double rrp[100]**

Definition at line 74 of file PythiaWrapper6_2.h.

9.60.3.45 double v[5][pyjets_maxn]

Definition at line 39 of file PythiaWrapper6_2.h.

9.60.3.46 double vckm[4][4]

Definition at line 57 of file PythiaWrapper6_2.h.

9.60.3.47 double vint[400]

Definition at line 100 of file PythiaWrapper6_2.h.

9.60.3.48 double xsec[3][501]

Definition at line 117 of file PythiaWrapper6_2.h.

9.61 PythiaWrapper6_2_WIN32.h File Reference

9.62 SearchVector.cc File Reference

```
#include "HepMC/SearchVector.h"
```

Namespaces

- namespace **HepMC**

Functions

- bool **HepMC::not_in_vector** (std::vector< GenParticle * > ***v**, GenParticle ***p**)
- std::vector< **HepMC::GenParticle** * >::iterator **HepMC::already_in_vector** (std::vector< GenParticle * > ***v**, GenParticle ***p**)
*returns true if **GenParticle** (p. 86) is in the vector*

9.63 SearchVector.h File Reference

```
#include "HepMC/GenVertex.h"
#include "HepMC/GenParticle.h"
```

Namespaces

- namespace **HepMC**

Functions

- **bool HepMC::not_in_vector** (std::vector< **HepMC::GenParticle** * > *, **GenParticle** *)
returns true if it cannot find GenParticle in the vector*
- **std::vector< HepMC::GenParticle * >::iterator HepMC::already_in_vector** (std::vector< **HepMC::GenParticle** * > *, **GenParticle** *)

9.64 SimpleVector.h File Reference

```
#include "HepMC/enable_if.h"
#include "HepMC/is_arithmetic.h"
#include "HepMC/SimpleVector.icc"
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::FourVector**
FourVector (p. 46) is a simple representation of a physics 4 vector.
- class **HepMC::ThreeVector**
ThreeVector (p. 232) is a simple representation of a position or displacement 3 vector.

9.65 TempParticleMap.h File Reference

`#include <map>`

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::TempParticleMap**
TempParticleMap (p. 229) is a temporary *GenParticle** container used during input.

9.66 testHepMCIteration.h File Reference

Classes

- class **IsFinalState**
test class

Functions

- bool **IsPhoton** (const **HepMC::GenParticle** *p)
returns true if the GenParticle particle is a photon with $pT > 10$ GeV
- bool **IsWBoson** (const **HepMC::GenParticle** *p)
returns true if the GenParticle is a W^+/W^-

9.66.1 Function Documentation

9.66.1.1 bool IsPhoton (const HepMC::GenParticle * p)

returns true if the GenParticle particle is a photon with $pT > 10$ GeV

Definition at line 10 of file testHepMCIteration.h.

References p.

9.66.1.2 bool IsWBoson (const HepMC::GenParticle * p)

returns true if the GenParticle is a W^+/W^-

Definition at line 17 of file testHepMCIteration.h.

References p.

9.67 testHerwigCopies.cc File Reference

```
#include <fstream>
#include <iostream>
#include "HepMC/HerwigWrapper.h"
#include "HepMC/IO_HERWIG.h"
#include "HepMC/GenEvent.h"
#include "HepMC/CompareGenEvent.h"
#include "HepMC/HEPEVT_Wrapper.h"
```

Functions

- `int main ()`

9.67.1 Function Documentation

9.67.1.1 `int main ()`

Definition at line 16 of file testHerwigCopies.cc.

References `HepMC::compareGenEvent()`, `HepMC::GenEvent::event_number()`, `hwbgen`, `hwbmch`, `hwcdec`, `hwcfor`, `hwdhad`, `hwdhob`, `hwdhvy`, `hwefin`, `hweini`, `hwepro`, `hwevnt`, `hwigin`, `hwmevt`, `hwproc`, `hwufne`, `hwiinc`, `hwiine`, `HepMC::GenEvent::print()`, `HepMC::IO_BaseClass::read_next_event()`, `HepMC::GenEvent::set_event_number()`, `HepMC::HEPEVT_Wrapper::set_max_number_entries()`, `HepMC::GenEvent::set_signal_process_id()`, and `HepMC::HEPEVT_Wrapper::set_sizeof_real()`.

9.68 testPrintBug.cc File Reference

```
#include <fstream>
#include "HepMC/GenEvent.h"
#include "HepMC/SimpleVector.h"
```

Functions

- `int main (int argc, char *argv[])`

9.68.1 Function Documentation

9.68.1.1 `int main (int argc, char * argv[])`

Definition at line 10 of file testPrintBug.cc.

References `HepMC::GenVertex::add_particle_in()`, `HepMC::GenVertex::add_particle_out()`, `HepMC::GenEvent::add_vertex()`, and `HepMC::GenEvent::print()`.

9.69 testPythiaCopies.cc File Reference

```
#include <fstream>
#include <iostream>
#include "HepMC/PythiaWrapper.h"
#include "HepMC/IO_HEPEVT.h"
#include "HepMC/GenEvent.h"
#include "HepMC/CompareGenEvent.h"
#include "PythiaHelper.h"
```

Functions

- `int main ()`

9.69.1 Function Documentation

9.69.1.1 `int main ()`

Definition at line 16 of file testPythiaCopies.cc.

References `HepMC::compareGenEvent()`, `HepMC::GenEvent::event_number()`, `initPythia()`, `HepMC::GenEvent::print()`, `pypars`, `HepMC::IO_BaseClass::read_next_event()`, `HepMC::HEPEVT_Wrapper::set_max_number_entries()`, `HepMC::GenEvent::set_mpi()`, and `HepMC::HEPEVT_Wrapper::set_sizeof_real()`.

9.70 testSimpleVector.cc File Reference

```
#include <iostream>
#include "HepMC/SimpleVector.h"
```

Functions

- `int main ()`

9.70.1 Function Documentation

9.70.1.1 `int main ()`

Definition at line 8 of file testSimpleVector.cc.

References `HepMC::FourVector::e()`, `HepMC::FourVector::eta()`, `HepMC::FourVector::m()`, `HepMC::FourVector::m2()`, `HepMC::FourVector::mag()`, `HepMC::ThreeVector::mag()`, `HepMC::FourVector::perp()`, `HepMC::ThreeVector::perp()`, `HepMC::FourVector::perp2()`, `HepMC::ThreeVector::perp2()`, `HepMC::FourVector::phi()`, `HepMC::ThreeVector::phi()`, `HepMC::FourVector::pseudoRapidity()`, `HepMC::FourVector::px()`, `HepMC::FourVector::py()`, `HepMC::FourVector::pz()`, `HepMC::ThreeVector::r()`, `HepMC::FourVector::rho()`, `HepMC::FourVector::set()`, `HepMC::ThreeVector::set()`, `HepMC::FourVector::setE()`, `HepMC::ThreeVector::setPhi()`, `HepMC::FourVector::setPx()`, `HepMC::FourVector::setPy()`, `HepMC::FourVector::setPz()`, `HepMC::FourVector::setT()`, `HepMC::ThreeVector::setTheta()`, `HepMC::FourVector::setX()`, `HepMC::ThreeVector::setX()`, `HepMC::FourVector::setY()`, `HepMC::ThreeVector::setY()`, `HepMC::FourVector::setZ()`, `HepMC::ThreeVector::setZ()`, `HepMC::FourVector::t()`, `HepMC::FourVector::theta()`, `HepMC::ThreeVector::theta()`, `HepMC::FourVector::x()`, `HepMC::ThreeVector::x()`, `HepMC::FourVector::y()`, `HepMC::ThreeVector::y()`, `HepMC::FourVector::z()`, and `HepMC::ThreeVector::z()`.

9.71 VectorConversion.h File Reference

```
#include "HepMC/SimpleVector.h"
#include "CLHEP/Vector/LorentzVector.h"
```

Namespaces

- namespace **CLHEP**

Functions

- **CLHEP::Hep3Vector convertTo** (const **HepMC::ThreeVector** &*v*)
*Convert from **HepMC::ThreeVector** (p. 232) to **CLHEP::Hep3Vector**.*
- **CLHEP::HepLorentzVector convertTo** (const **HepMC::FourVector** &*v*)
*Convert from **HepMC::FourVector** (p. 46) to **CLHEP::HepLorentzVector** (p. 270).*

9.71.1 Function Documentation

9.71.1.1 **CLHEP::HepLorentzVector convertTo** (const **HepMC::FourVector** & *v*)

Convert from **HepMC::FourVector** (p. 46) to **CLHEP::HepLorentzVector** (p. 270).

Definition at line 25 of file VectorConversion.h.

References [v](#).

9.71.1.2 **CLHEP::Hep3Vector convertTo** (const **HepMC::ThreeVector** & *v*)

Convert from **HepMC::ThreeVector** (p. 232) to **CLHEP::Hep3Vector**.

Examples:

`example_BuildEventFromScratch.cc`, and `VectorConversion.h`.

Definition at line 21 of file VectorConversion.h.

References [v](#).

Referenced by `main()`.

9.72 Version.h File Reference

```
#include <string>
#include <iostream>
```

Namespaces

- namespace **HepMC**

Functions

- void **HepMC::version** ()
*print **HepMC** (p. 19) version*
- void **HepMC::writeVersion** (std::ostream &os)
*write **HepMC** (p. 19) version to os*
- std::string **HepMC::versionName** ()
*return **HepMC** (p. 19) version*

9.73 WeightContainer.h File Reference

```
#include <iostream>
```

```
#include <vector>
```

Namespaces

- namespace **HepMC**

Classes

- class **HepMC::WeightContainer**
Container for the Weights associated with an event or vertex.

Chapter 10

HepMC Example Documentation

10.1 example_BuildEventFromScratch.cc

```
1
2 // Matt.Dobbs@Cern.CH, Feb 2000
3 // Example of building an event and a particle data table from scratch
4 // This is meant to be of use for persons implementing HepMC inside a MC
5 // event generator
6 // To Compile: go to the HepMC directory and type:
7 // gmake examples/example_BuildEventFromScratch.exe
8 //
9 //
10
11 #include <iostream>
12
13 #include "VectorConversion.h"
14 #include "HepMC/GenEvent.h"
15 #include "HepMC/ParticleDataTable.h"
16 #include "CLHEP/Vector/LorentzVector.h"
17
18 // in this example we use the HepMC namespace, so that we do not have to
19 // precede all HepMC classes with HepMC::
20
21 // This example also shows how to use the CLHEP Lorentz vector with HepMC2
22
23 using namespace HepMC;
24 using namespace CLHEP;
25
26 int main() {
27     //
28     // In this example we will place the following event into HepMC "by hand"
29     //
30     //      name status pdg_id parent Px      Py      Pz      Energy      Mass
31     //  1  !p+!      3    2212    0,0    0.000    0.000 7000.000 7000.000    0.938
32     //  2  !p+!      3    2212    0,0    0.000    0.000-7000.000 7000.000    0.938
33     //=====
34     //  3  !d!        3        1    1,1    0.750   -1.569   32.191   32.238    0.000
35     //  4  !u~!       3       -2    2,2   -3.047  -19.000  -54.629   57.920    0.000
36     //  5  !W-!       3      -24    1,2    1.517   -20.68   -20.605   85.925    80.799
37     //  6  !gamma!    1        22    1,2   -3.813    0.113   -1.833    4.233    0.000
38     //  7  !d!        1         1    5,5   -2.445   28.816    6.082   29.552    0.010
39     //  8  !u~!       1        -2    5,5    3.962  -49.498  -26.687   56.373    0.006
40
41     // first we construct a ParticleDataTable with all the particles we need
42     ParticleDataTable pdt("my particle data table");
43     // create a particle data entry for the proton and add it to pdt at the
44     // same time
45     pdt.insert( new ParticleData( "p+", 2212,  +1, 0.938,  -1, .5 ) );
```

```

46  pdt.insert( new ParticleData( "d",  1, -2./3., 0,      -1, .5 ) );
47  pdt.insert( new ParticleData( "u~", -2, -1./3., 0,      -1, .5 ) );
48  pdt.insert( new ParticleData( "W-", -24,    -1, 80.396,
49                      clifetime_from_width(2.06), 1 ) );
50  pdt.insert( new ParticleData( "gamma", 22,  0, 0,      -1, 1 ) );
51
52  // print out the GenParticle Data to the screen
53  pdt.print();
54
55  // now we build the graph, which will look like
56  //
57  //      p7
58  //      /
59  // p1  \v1--p3      p5---v4
60  //      \_v3_/      \
61  //      /      \      p8
62  // v2--p4      \
63  // /      p6
64  // p2
65
66  // First create the event container, with Signal Process 20, event number 1
67  //
68  // Note that the HepLorentzVectors will be automatically converted to
69  // HepMC::FourVector within GenParticle and GenVertex
70  GenEvent* evt = new GenEvent( 20, 1 );
71  //
72  // create vertex 1 and vertex 2, together with their inparticles
73  GenVertex* v1 = new GenVertex();
74  evt->add_vertex( v1 );
75  v1->add_particle_in( new GenParticle( HepLorentzVector(0,0,7000,7000),
76                      2212, 3 ) );
77  GenVertex* v2 = new GenVertex();
78  evt->add_vertex( v2 );
79  v2->add_particle_in( new GenParticle( HepLorentzVector(0,0,-7000,7000),
80                      2212, 3 ) );
81  //
82  // create the outgoing particles of v1 and v2
83  GenParticle* p3 =
84      new GenParticle( HepLorentzVector(.750,-1.569,32.191,32.238), 1, 3 );
85  v1->add_particle_out( p3 );
86  GenParticle* p4 =
87      new GenParticle( HepLorentzVector(-3.047,-19.,-54.629,57.920), -2, 3 );
88  v2->add_particle_out( p4 );
89  //
90  // create v3
91  GenVertex* v3 = new GenVertex();
92  evt->add_vertex( v3 );
93  v3->add_particle_in( p3 );
94  v3->add_particle_in( p4 );
95  v3->add_particle_out(
96      new GenParticle( HepLorentzVector(-3.813,0.113,-1.833,4.233 ), 22, 1 )
97      );
98  GenParticle* p5 =
99      new GenParticle( HepLorentzVector(1.517,-20.68,-20.605,85.925), -24,3);
100  v3->add_particle_out( p5 );
101  //
102  // create v4
103  GenVertex* v4 = new GenVertex(HepLorentzVector(0.12,-0.3,0.05,0.004));
104  evt->add_vertex( v4 );
105  v4->add_particle_in( p5 );
106  v4->add_particle_out(
107      new GenParticle( HepLorentzVector(-2.445,28.816,6.082,29.552), 1,1 )
108      );
109  v4->add_particle_out(
110      new GenParticle( HepLorentzVector(3.962,-49.498,-26.687,56.373), -2,1 )
111      );
112  //

```

```
113     // tell the event which vertex is the signal process vertex
114     evt->set_signal_process_vertex( v3 );
115     // the event is complete, we now print it out to the screen
116     evt->print();
117
118     // example conversion back to Lorentz vector
119     // add all outgoing momenta
120     std::cout << std::endl;
121     std::cout << " Add output momenta " << std::endl;
122     HepLorentzVector sum;
123     for ( GenEvent::particle_const_iterator p = evt->particles_begin();
124           p != evt->particles_end(); ++p ){
125         if( (*p)->status() == 1 ) {
126             sum += convertTo( (*p)->momentum() );
127             (*p)->print();
128         }
129     }
130     std::cout << "Vector Sum: " << sum << std::endl;
131
132     // now clean-up by deleting all objects from memory
133     //
134     // deleting the event deletes all contained vertices, and all particles
135     // contained in those vertices
136     delete evt;
137
138     // delete all particle data objects in the particle data table pdt
139     pdt.delete_all();
140
141     return 0;
142 }
```

10.2 example_EventSelection.cc

```

1
2 // Matt.Dobbs@Cern.CH, Feb 2000
3 // Example of applying an event selection to the events written to file
4 // using example_MyPythia.cxx
5 // Events containing a photon of pT > 25 GeV pass the selection and are
6 // written to "example_EventSelection.dat"
7 // To Compile: go to the HepMC directory and type:
8 // gmake examples/example_EventSelection.exe
9 //
10 //
11
12 #include "HepMC/IO_GenEvent.h"
13 #include "HepMC/GenEvent.h"
14
15
16
17 class IsEventGood {
18 public:
19     bool operator()( const HepMC::GenEvent* evt ) {
20         for ( HepMC::GenEvent::particle_const_iterator p
21              = evt->particles_begin(); p != evt->particles_end(); ++p ){
22             if ( (*p)->pdg_id() == 22 && (*p)->momentum().perp() > 25. ) {
23                 //std::cout << "Event " << evt->event_number()
24                 //      << " is a good event." << std::endl;
25                 //(*p)->print();
26                 return 1;
27             }
28         }
29         return 0;
30     }
31 };
32
33 int main() {
34     // declare an input strategy to read the data produced with the
35     // example_MyPythia
36     { // begin scope of ascii_in and ascii_out
37         HepMC::IO_GenEvent ascii_in("example_MyPythia.dat",std::ios::in);
38         // declare another IO_GenEvent for writing out the good events
39         HepMC::IO_GenEvent ascii_out("example_EventSelection.dat",std::ios::out);
40         // declare an instance of the event selection predicate
41         IsEventGood is_good_event;
42         //.....EVENT LOOP
43         int icount=0;
44         int num_good_events=0;
45         HepMC::GenEvent* evt = ascii_in.read_next_event();
46         while ( evt ) {
47             icount++;
48             if ( icount%50==1 ) std::cout << "Processing Event Number " << icount
49                                     << " its # " << evt->event_number()
50                                     << std::endl;
51
52             if ( is_good_event(evt) ) {
53                 ascii_out << evt;
54                 ++num_good_events;
55             }
56             delete evt;
57             ascii_in >> evt;
58         }
59         //.....PRINT RESULT
60         std::cout << num_good_events << " out of " << icount
61                 << " processed events passed the cuts. Finished." << std::endl;
62     } // end scope of ascii_in and ascii_out
63     return 0;
64 }
65
66
67
68
69
70
71

```

72
73
74
75
76
77

10.3 example_MyHerwig.cc

```

1
2 // Matt.Dobbs@Cern.CH, October 2002
3 // example of generating events with Herwig using HepMC/HerwigWrapper.h
4 // Events are read into the HepMC event record from the FORTRAN HEPEVT
5 // common block using the IO_HERWIG strategy.
6
7 #include <iostream>
8 #include "HepMC/HerwigWrapper.h"
9 #include "HepMC/IO_HERWIG.h"
10 #include "HepMC/GenEvent.h"
11 #include "HepMC/HEPEVT_Wrapper.h"
12
13 int main() {
14     //
15     //.....HEPEVT
16     // Herwig 6.4 uses HEPEVT with 4000 entries and 8-byte floating point
17     // numbers. We need to explicitly pass this information to the
18     // HEPEVT_Wrapper.
19     //
20     HepMC::HEPEVT_Wrapper::set_max_number_entries(4000);
21     HepMC::HEPEVT_Wrapper::set_sizeof_real(8);
22     //
23     //.....INITIALIZATIONS
24
25     hwproc.PBEAM1 = 7000.; // energy of beam1
26     hwproc.PBEAM2 = 7000.; // energy of beam2
27     // 1610 = gg->H-> WW, 1706 = qq->ttbar, 2510 = ttH -> ttWW
28     hwproc.IPROC = 1706; // qq -> ttbar production
29     hwproc.MAXEV = 100; // number of events
30     // tell it what the beam particles are:
31     for ( unsigned int i = 0; i < 8; ++i ) {
32         hwbmch.PART1[i] = (i < 1) ? 'P' : ' ';
33         hwbmch.PART2[i] = (i < 1) ? 'P' : ' ';
34     }
35     hwigin(); // INITIALISE OTHER COMMON BLOCKS
36     hwevnt.MAXPR = 1; // number of events to print
37     hwiinc(); // compute parameter-dependent constants
38     hweini(); // initialise elementary process
39
40     //.....HepMC INITIALIZATIONS
41     //
42     // Instantiate an IO strategy for reading from HEPEVT.
43     HepMC::IO_HERWIG hepevtio;
44     //
45     //.....EVENT LOOP
46     for ( int i = 1; i <= hwproc.MAXEV; i++ ) {
47         if ( i%50==1 ) std::cout << "Processing Event Number "
48             << i << std::endl;
49
50         // initialise event
51         hwiue();
52         // generate hard subprocess
53         hwepro();
54         // generate parton cascades
55         hwbgen();
56         // do heavy object decays
57         hwdhob();
58         // do cluster formation
59         hwcfor();
60         // do cluster decays
61         hwcdec();
62         // do unstable particle decays
63         hwdhad();
64         // do heavy flavour hadron decays
65         hwdhvy();
66         // add soft underlying event if needed

```

```
76         hwmevt();
77         // finish event
78         hwufne();
79         HepMC::GenEvent* evt = hepevtio.read_next_event();
80         // add some information to the event
81         evt->set_event_number(i);
82         evt->set_signal_process_id(20);
83         if (i<=hwevnt.MAXPR) {
84             std::cout << "\n\n This is the FIXED version of HEPEVT as "
85                 << "coded in IO_HERWIG " << std::endl;
86             HepMC::HEPEVT_Wrapper::print_hepevt();
87             evt->print();
88         }
89
90         // we also need to delete the created event from memory
91         delete evt;
92     }
93     //.....TERMINATION
94     hwefin();
95
96     return 0;
97 }
```

10.4 example_MyPythia.cc

```

1
2 // Matt.Dobbs@Cern.CH, December 1999
3 // November 2000, updated to use Pythia 6.1
4 // example of generating events with Pythia
5 // using HepMC/PythiaWrapper.h
6 // Events are read into the HepMC event record from the FORTRAN HEPEVT
7 // common block using the IO_HEPEVT strategy and then output to file in
8 // ascii format using the IO_GenEvent strategy.
20
21 #include <iostream>
22 #include "HepMC/PythiaWrapper.h"
23 #include "HepMC/IO_HEPEVT.h"
24 #include "HepMC/IO_GenEvent.h"
25 #include "HepMC/GenEvent.h"
26 #include "PythiaHelper.h"
27
28 int main() {
29     //
30     //.....HEPEVT
31     // Pythia 6.1 uses HEPEVT with 4000 entries and 8-byte floating point
32     // numbers. We need to explicitly pass this information to the
33     // HEPEVT_Wrapper.
34     //
35     HepMC::HEPEVT_Wrapper::set_max_number_entries(4000);
36     HepMC::HEPEVT_Wrapper::set_sizeof_real(8);
37     //
38     //.....PYTHIA INITIALIZATIONS
39     initPythia();
40
41     //.....HepMC INITIALIZATIONS
42     //
43     // Instantiate an IO strategy for reading from HEPEVT.
44     HepMC::IO_HEPEVT hepevtio;
45     //
46     { // begin scope of ascii_io
47         // Instantiate an IO strategy to write the data to file
48         HepMC::IO_GenEvent ascii_io("example_MyPythia.dat",std::ios::out);
49         //
50         //.....EVENT LOOP
51         for ( int i = 1; i <= 100; i++ ) {
52             if ( i%50==1 ) std::cout << "Processing Event Number "
53                             << i << std::endl;
54             call_pyevnt(); // generate one event with Pythia
55             // pythia pyhepc routine converts common PYJETS in common HEPEVT
56             call_pyhepc( 1 );
57             HepMC::GenEvent* evt = hepevtio.read_next_event();
58             // add some information to the event
59             evt->set_event_number(i);
60             evt->set_signal_process_id(20);
61             // set number of multi parton interactions
62             evt->set_mpi( pypars.msti[31-1] );
63             // write the event out to the ascii files
64             ascii_io << evt;
65             // we also need to delete the created event from memory
66             delete evt;
67         }
68         //.....TERMINATION
69         // write out some information from Pythia to the screen
70         call_pystat( 1 );
71     } // end scope of ascii_io
72
73     return 0;
74 }
75
76

```


10.5 example_MyPythiaOnlyToHepMC.cc

```

1
2 // Matt.Dobbs@Cern.CH, December 1999
3 // November 2000, updated to use Pythia 6.1
4 // example of generating events with Pythia
5 // using HepMC/PythiaWrapper.h
6 // Events are read into the HepMC event record from the FORTRAN HEPEVT
7 // common block using the IO_HEPEVT strategy -- nothing is done with them.
8 // This program is just used to find the total time required to transfer
9 // from HEPEVT into the HepMC event record.
11 // To Compile: go to the HepMC directory and type:
12 // gmake examples/example_MyPythiaOnlyTo HepMC.exe
13 //
14 // See comments in examples/example_MyPythia.cxx regarding the HEPEVT wrapper.
15 //
16
17 #include <iostream>
18 #include "HepMC/PythiaWrapper.h"
19 #include "HepMC/IO_HEPEVT.h"
20 #include "HepMC/GenEvent.h"
21 #include "PythiaHelper.h"
22
23 int main() {
24     //
25     //.....HEPEVT
26     // Pythia 6.1 uses HEPEVT with 4000 entries and 8-byte floating point
27     // numbers. We need to explicitly pass this information to the
28     // HEPEVT_Wrapper.
29     //
30     HepMC::HEPEVT_Wrapper::set_max_number_entries(4000);
31     HepMC::HEPEVT_Wrapper::set_sizeof_real(8);
32     //
33     //.....PYTHIA INITIALIZATIONS
34     initPythia();
35     //
36     //.....HepMC INITIALIZATIONS
37     //
38     // Instantiate an IO strategy for reading from HEPEVT.
39     HepMC::IO_HEPEVT hepevtio;
40     //
41     //.....EVENT LOOP
42     for ( int i = 1; i <= 100; i++ ) {
43         if ( i%50==1 ) std::cout << "Processing Event Number "
44             << i << std::endl;
45         call_pyevnt(); // generate one event with Pythia
46         // pythia pyhepc routine convert common PYJETS in common HEPEVT
47         call_pyhepc( 1 );
48         HepMC::GenEvent* evt = hepevtio.read_next_event();
49         // set number of multi parton interactions
50         evt->set_mpi( pypars.msti[31-1] );
51         //
52         //.....USER WOULD PROCESS EVENT HERE
53         //
54         // we also need to delete the created event from memory
55         delete evt;
56     }
57     //.....TERMINATION
58     // write out some information from Pythia to the screen
59     call_pystat( 1 );
60
61     return 0;
62 }
63
64
65

```

10.6 example_MyPythiaRead.cc

```

1
2 // garren@fnal.gov, January 2007
3 // This example is an extension of example_MyPythia.cc
4 //
5 // generate events with Pythia, write a file, and read the resulting output
6 // Notice that we use scope to explicitly close the output files.
7 // The two output files should be the same size, but because particles are
8 // saved as sets within a vertex, they will be written in arbitrary order.
9 // To Compile: go to the HepMC directory and type:
10 // gmake examples/example_MyPythiaRead.exe
11 //
12 //
13 // In this example the precision and number of entries for the HEPEVT
14 // fortran common block are explicitly defined to correspond to those
15 // used in the Pythia version of the HEPEVT common block.
16 //
17 // If you get funny output from HEPEVT in your own code, probably you have
18 // set these values incorrectly!
19 //
20
21 #include <iostream>
22 #include "HepMC/PythiaWrapper.h"
23 #include "HepMC/IO_HEPEVT.h"
24 #include "HepMC/IO_GenEvent.h"
25 #include "HepMC/GenEvent.h"
26 #include "PythiaHelper.h"
27
28 int main() {
29     //
30     //.....HEPEVT
31     // Pythia 6.3 uses HEPEVT with 4000 entries and 8-byte floating point
32     // numbers. We need to explicitly pass this information to the
33     // HEPEVT_Wrapper.
34     //
35     HepMC::HEPEVT_Wrapper::set_max_number_entries(4000);
36     HepMC::HEPEVT_Wrapper::set_sizeof_real(8);
37     //
38     //.....PYTHIA INITIALIZATIONS
39     initPythia();
40
41     //.....HepMC INITIALIZATIONS
42     //
43     // Instantiate an IO strategy for reading from HEPEVT.
44     HepMC::IO_HEPEVT hepevtio;
45     //
46     //.....define the output scope
47     {
48         // Instantial an IO strategy to write the data to file - it uses the
49         // same ParticleDataTable
50         HepMC::IO_GenEvent ascii_io("example_MyPythiaRead.dat",std::ios::out);
51         //
52         //.....EVENT LOOP
53         for ( int i = 1; i <= 100; i++ ) {
54             if ( i%50==1 ) std::cout << "Processing Event Number "
55                 << i << std::endl;
56             call_pyevnt(); // generate one event with Pythia
57             // pythia pyhepc routine converts common PYJETS in common HEPEVT
58             call_pyhepc( 1 );
59             HepMC::GenEvent* evt = hepevtio.read_next_event();
60             // add some information to the event
61             evt->set_event_number(i);
62             evt->set_signal_process_id(20);
63             // write the event out to the ascii file
64             ascii_io << evt;
65             // we also need to delete the created event from memory
66             delete evt;

```

```
67     }
68     //.....TERMINATION
69     // write out some information from Pythia to the screen
70     call_pystat( 1 );
71 } // ascii_io destructor is called here
72 //
73 //.....define an input scope
74 {
75     // now read the file we wrote
76     HepMC::IO_GenEvent ascii_in("example_MyPythiaRead.dat",std::ios::in);
77     HepMC::IO_GenEvent ascii_io2("example_MyPythiaRead2.dat",std::ios::out);
78     int icount=0;
79     HepMC::GenEvent* evt = ascii_in.read_next_event();
80     while ( evt ) {
81         icount++;
82         if ( icount%50==1 ) std::cout << "Processing Event Number " << icount
83                                 << " its # " << evt->event_number()
84                                 << std::endl;
85         // write the event out to the ascii file
86         ascii_io2 << evt;
87         delete evt;
88         ascii_in >> evt;
89     }
90     //.....PRINT RESULT
91     std::cout << icount << " events found. Finished." << std::endl;
92 } // ascii_io2 and ascii_in destructors are called here
93
94 return 0;
95 }
96
97
98
```

10.7 example_MyPythiaWithEventSelection.cc

```

1
2 // Matt.Dobbs@Cern.CH, December 1999
3 // November 2000, updated to use Pythia 6.1
4 // example of generating events with Pythia
5 // using HepMC/PythiaWrapper.h
6 // Events are read into the HepMC event record from the FORTRAN HEPEVT
7 // common block using the IO_HEPEVT strategy and then a very simple event
8 // selection is performed.
9 // To Compile: go to the HepMC directory and type:
10 // gmake examples/example_MyPythiaWithEventSelection.exe
11 //
12 // See comments in examples/example_MyPythia.cxx regarding the HEPEVT wrapper.
13 //
14
15
16 #include <iostream>
17 #include "HepMC/PythiaWrapper.h"
18 #include "HepMC/IO_HEPEVT.h"
19 #include "HepMC/GenEvent.h"
20 #include "PythiaHelper.h"
21
22
23
24
25 class IsGoodEventMyPythia {
26 public:
27     bool operator()( const HepMC::GenEvent* evt ) {
28         for ( HepMC::GenEvent::particle_const_iterator p
29               = evt->particles_begin(); p != evt->particles_end(); ++p ){
30             if ( (*p)->pdg_id() == 22 && (*p)->momentum().perp() > 25. ) {
31                 //std::cout << "Event " << evt->event_number()
32                 //      << " is a good event." << std::endl;
33                 //(*p)->print();
34                 return 1;
35             }
36         }
37         return 0;
38     }
39 };
40
41 int main() {
42     //
43     //.....HEPEVT
44     // Pythia 6.1 uses HEPEVT with 4000 entries and 8-byte floating point
45     // numbers. We need to explicitly pass this information to the
46     // HEPEVT_Wrapper.
47     //
48     HepMC::HEPEVT_Wrapper::set_max_number_entries(4000);
49     HepMC::HEPEVT_Wrapper::set_sizeof_real(8);
50     //
51     //.....PYTHIA INITIALIZATIONS
52     initPythia();
53     //
54     //.....HepMC INITIALIZATIONS
55     // Instantiate an IO strategy for reading from HEPEVT.
56     HepMC::IO_HEPEVT hepevtio;
57     // declare an instance of the event selection predicate
58     IsGoodEventMyPythia is_good_event;
59     //.....EVENT LOOP
60     int icount=0;
61     int num_good_events=0;
62     for ( int i = 1; i <= 100; i++ ) {
63         icount++;
64         if ( i%50==1 ) std::cout << "Processing Event Number "
65                                << i << std::endl;
66         call_pyevnt(); // generate one event with Pythia
67         // pythia pyhepc routine convert common PYJETS in common HEPEVT

```

```
72     call_pyhepc( 1 );
73     HepMC::GenEvent* evt = hepevtio.read_next_event();
74     // set number of multi parton interactions
75     evt->set_mpi( pypars.msti[31-1] );
76     // do event selection
77     if ( is_good_event(evt) ) ++num_good_events;
78     // we also need to delete the created event from memory
79     delete evt;
80 }
81 //.....TERMINATION
82 // write out some information from Pythia to the screen
83 call_pystat( 1 );
84 //.....PRINT RESULTS
85 std::cout << num_good_events << " out of " << icount
86           << " processed events passed the cuts. Finished." << std::endl;
87 return 0;
88 }
89
90
91
```

10.8 example_PythiaParticle.cc

```

1
2 // garren@fnal.gov, July 2006
3 // example of generating events with Pythia
4 // using HepMC/PythiaWrapper.h
5 // Events are read into the HepMC event record from the FORTRAN HEPEVT
6 // common block using the IO_HEPEVT strategy and then output to file in
7 // ascii format using the IO_AsciiParticles strategy.
8 //
9 // This is identical to example_MyPythia.cc except that it uses IO_AsciiParticles.
10 // To Compile: go to the examples directory and type:
11 // gmake example_PythiaParticle.exe
12 //
13 // In this example the precision and number of entries for the HEPEVT
14 // fortran common block are explicitly defined to correspond to those
15 // used in the Pythia version of the HEPEVT common block.
16 //
17 // If you get funny output from HEPEVT in your own code, probably you have
18 // set these values incorrectly!
19 //
20 //
21
22 #include <iostream>
23 #include "HepMC/PythiaWrapper.h"
24 #include "HepMC/IO_HEPEVT.h"
25 #include "HepMC/IO_AsciiParticles.h"
26 #include "HepMC/GenEvent.h"
27 #include "PythiaHelper.h"
28
29 int main() {
30     //
31     //.....HEPEVT
32     // Pythia 6.1 uses HEPEVT with 4000 entries and 8-byte floating point
33     // numbers. We need to explicitly pass this information to the
34     // HEPEVT_Wrapper.
35     //
36     HepMC::HEPEVT_Wrapper::set_max_number_entries(4000);
37     HepMC::HEPEVT_Wrapper::set_sizeof_real(8);
38     //
39     //.....PYTHIA INITIALIZATIONS
40     initPythia();
41
42     //.....HepMC INITIALIZATIONS
43     //
44     // Instantiate an IO strategy for reading from HEPEVT.
45     HepMC::IO_HEPEVT hepevtio;
46     //
47     { // begin scope of ascii_io
48         // Instantiate an IO strategy to write the data to file
49         HepMC::IO_AsciiParticles ascii_io("example_PythiaParticle.dat",std::ios::out);
50         //
51         //.....EVENT LOOP
52         for ( int i = 1; i <= 100; i++ ) {
53             if ( i%50==1 ) std::cout << "Processing Event Number "
54                 << i << std::endl;
55             call_pyevnt(); // generate one event with Pythia
56             // pythia pyhepc routine converts common PYJETS in common HEPEVT
57             call_pyhepc( 1 );
58             HepMC::GenEvent* evt = hepevtio.read_next_event();
59             // add some information to the event
60             evt->set_event_number(i);
61             evt->set_signal_process_id(20);
62             // write the event out to the ascii file
63             ascii_io << evt;
64             // we also need to delete the created event from memory
65             delete evt;
66         }

```

```
67      //.....TERMINATION
68      // write out some information from Pythia to the screen
69      call_pystat( 1 );
70  } // end scope of ascii_io
71
72  return 0;
73 }
74
75
76
```


10.9 example_UsingIterators.cc

```

1
2 // Matt.Dobbs@Cern.CH, Feb 2000
3 // This example shows how to use the particle and vertex iterators
5 // To Compile: go to the HepMC directory and type:
6 // gmake examples/example_UsingIterators.exe
7 //
8
9 #include "HepMC/IO_GenEvent.h"
10 #include "HepMC/GenEvent.h"
11 #include <math.h>
12 #include <algorithm>
13 #include <list>
14
15
16
17 class IsPhoton {
18 public:
19     bool operator()( const HepMC::GenParticle* p ) {
20         if ( p->pdg_id() == 22
21             && p->momentum().perp() > 10. ) return 1;
22         return 0;
23     }
24 };
25
26
27 class IsW_Boson {
28 public:
29     bool operator()( const HepMC::GenParticle* p ) {
30         if ( abs(p->pdg_id()) == 24 ) return 1;
31         return 0;
32     }
33 };
34
35
36 class IsStateFinal {
37 public:
38     bool operator()( const HepMC::GenParticle* p ) {
39         if ( !p->end_vertex() && p->status()==1 ) return 1;
40         return 0;
41     }
42 };
43
44
45 int main() {
46     { // begin scope of ascii_in
47         // an event has been prepared in advance for this example, read it
48         // into memory using the IO_GenEvent input strategy
49         HepMC::IO_GenEvent ascii_in("example_UsingIterators.txt",std::ios::in);
50         if ( ascii_in.rdstate() == std::ios::failbit ) {
51             std::cerr << "ERROR input file example_UsingIterators.txt is needed "
52                 << "and does not exist. "
53                 << "\n Look for it in HepMC/examples, Exit." << std::endl;
54             return 1;
55         }
56     }
57
58     HepMC::GenEvent* evt = ascii_in.read_next_event();
59
60     // if you wish to have a look at the event, then use evt->print();
61
62     // use GenEvent::vertex_iterator to fill a list of all
63     // vertices in the event
64     std::list<HepMC::GenVertex*> allvertices;
65     for ( HepMC::GenEvent::vertex_iterator v = evt->vertices_begin();
66           v != evt->vertices_end(); ++v ) {
67         allvertices.push_back(*v);
68     }
69 }

```

```

80 // we could do the same thing with the STL algorithm copy
81 std::list<HepMC::GenVertex*> allvertices2;
82 copy( evt->vertices_begin(), evt->vertices_end(),
83       back_inserter(allvertices2) );
84
85 // fill a list of all final state particles in the event, by requiring
86 // that each particle satisfyies the IsStateFinal predicate
87 IsStateFinal isfinal;
88 std::list<HepMC::GenParticle*> finalstateparticles;
89 for ( HepMC::GenEvent::particle_iterator p = evt->particles_begin();
90       p != evt->particles_end(); ++p ) {
91     if ( isfinal(*p) ) finalstateparticles.push_back(*p);
92 }
93
94 // an STL-like algorithm called HepMC::copy_if is provided in the
95 // GenEvent.h header to do this sort of operation more easily,
96 // you could get the identical results as above by using:
97 std::list<HepMC::GenParticle*> finalstateparticles2;
98 HepMC::copy_if( evt->particles_begin(), evt->particles_end(),
99                back_inserter(finalstateparticles2), IsStateFinal() );
100
101 // lets print all photons in the event that satisfy the IsPhoton criteria
102 IsPhoton isphoton;
103 for ( HepMC::GenEvent::particle_iterator p = evt->particles_begin();
104       p != evt->particles_end(); ++p ) {
105     if ( isphoton(*p) ) (*p)->print();
106 }
107
108 // the GenVertex::particle_iterator and GenVertex::vertex_iterator
109 // are slightly different from the GenEvent:: versions, in that
110 // the iterator starts at the given vertex, and walks through the attached
111 // vertex returning particles/vertices.
112 // Thus only particles/vertices which are in the same graph as the given
113 // vertex will be returned. A range is specified with these iterators,
114 // the choices are:
115 //   parents, children, family, ancestors, descendants, relatives
116 // here are some examples.
117
118 // use GenEvent::particle_iterator to find all W's in the event,
119 // then
120 // (1) for each W user the GenVertex::particle_iterator with a range of
121 //     parents to return and print the immediate mothers of these W's.
122 // (2) for each W user the GenVertex::particle_iterator with a range of
123 //     descendants to return and print all descendants of these W's.
124 IsW_Boson isw;
125 for ( HepMC::GenEvent::particle_iterator p = evt->particles_begin();
126       p != evt->particles_end(); ++p ) {
127     if ( isw(*p) ) {
128         std::cout << "A W boson has been found: " << std::endl;
129         (*p)->print();
130         // return all parents
131         // we do this by pointing to the production vertex of the W
132         // particle and asking for all particle parents of that vertex
133         std::cout << "\t Its parents are: " << std::endl;
134         if ( (*p)->production_vertex() ) {
135             for ( HepMC::GenVertex::particle_iterator mother
136                   = (*p)->production_vertex()->
137                     particles_begin(HepMC::parents);
138                   mother != (*p)->production_vertex()->
139                     particles_end(HepMC::parents);
140                   ++mother ) {
141             std::cout << "\t";
142             (*mother)->print();
143         }
144     }
145     // return all descendants
146     // we do this by pointing to the end vertex of the W

```

```
147         // particle and asking for all particle descendants of that vertex
148         std::cout << "\t\t Its descendants are: " << std::endl;
149         if ( (*p)->end_vertex() ) {
150             for ( HepMC::GenVertex::particle_iterator des
151                  =(*p)->end_vertex()->
152                      particles_begin(HepMC::descendants);
153                   des != (*p)->end_vertex()->
154                       particles_end(HepMC::descendants);
155                   ++des ) {
156                 std::cout << "\t\t";
157                 (*des)->print();
158             }
159         }
160     }
161 }
162 // cleanup
163 delete evt;
164 // in analogy to the above, similar use can be made of the
165 // HepMC::GenVertex::vertex_iterator, which also accepts a range.
166 } // end scope of ascii_in
167
168 return 0;
169 }
```

10.10 testHerwigCopies.cc

```

1
2 // testHerwigCopies.cc
3 //
4 // garren@fnal.gov, January 2008
5 // Multiple events in memory at the same time
6
7
8 #include <fstream>
9 #include <iostream>
10 #include "HepMC/HerwigWrapper.h"
11 #include "HepMC/IO_HERWIG.h"
12 #include "HepMC/GenEvent.h"
13 #include "HepMC/CompareGenEvent.h"
14 #include "HepMC/HEPEVT_Wrapper.h"
15
16 int main() {
17     //
18     //.....HEPEVT
19     // Herwig 6.4 uses HEPEVT with 4000 entries and 8-byte floating point
20     // numbers. We need to explicitly pass this information to the
21     // HEPEVT_Wrapper.
22     //
23     HepMC::HEPEVT_Wrapper::set_max_number_entries(4000);
24     HepMC::HEPEVT_Wrapper::set_sizeof_real(8);
25     //
26     //.....INITIALIZATIONS
27
28     hwproc.PBEAM1 = 7000.; // energy of beam1
29     hwproc.PBEAM2 = 7000.; // energy of beam2
30     // 1610 = gg->H-> WW, 1706 = qq->ttbar, 2510 = ttH -> ttWW
31     hwproc.IPROC = 1706; // qq -> ttbar production
32     hwproc.MAXEV = 50; // number of events
33     // tell it what the beam particles are:
34     for ( unsigned int i = 0; i < 8; ++i ) {
35         hwbmch.PART1[i] = (i < 1) ? 'P' : ' ';
36         hwbmch.PART2[i] = (i < 1) ? 'P' : ' ';
37     }
38     hwigin(); // INITIALISE OTHER COMMON BLOCKS
39     hwevnt.MAXPR = 0; // number of events to print
40     hwuinc(); // compute parameter-dependent constants
41     hweini(); // initialise elementary process
42
43     //.....HepMC INITIALIZATIONS
44     //
45     // Instantiate an IO strategy for reading from HEPEVT.
46     HepMC::IO_HERWIG hepevtio;
47     //
48     // open some output files
49     std::ofstream out1( "testHerwigOriginals.dat" );
50     std::ofstream out2( "testHerwigCopies1.dat" );
51     std::ofstream out3( "testHerwigCopies2.dat" );
52     //
53     //.....EVENT LOOP
54     for ( int i = 1; i <= hwproc.MAXEV; i++ ) {
55         if ( i%50==1 ) std::cout << "Processing Event Number "
56             << i << std::endl;
57
58         // initialise event
59         hwuine();
60         // generate hard subprocess
61         hwepro();
62         // generate parton cascades
63         hwbgen();
64         // do heavy object decays
65         hwdhob();
66         // do cluster formation
67         hwcfor();

```

```
67         // do cluster decays
68         hwcdec();
69         // do unstable particle decays
70         hwdhad();
71         // do heavy flavour hadron decays
72         hwdhvy();
73         // add soft underlying event if needed
74         hwmevt();
75         // finish event
76         hwufne();
77         HepMC::GenEvent* evt = hepevtio.read_next_event();
78         // add some information to the event
79         evt->set_event_number(i);
80         evt->set_signal_process_id(20);
81         //
82         //.....make some copies
83         evt->print(out1);
84         HepMC::GenEvent ec = (*evt);
85         ec.print(out2);
86         HepMC::GenEvent* evt4 = new HepMC::GenEvent(*evt);
87         evt4->print(out3);
88         if( !compareGenEvent(evt,evt4) ) {
89             std::cerr << "testHerwigCopies: GenEvent comparison fails at event "
90                 << evt->event_number() << std::endl;
91             return -1;
92         }
93
94         // we also need to delete the created event from memory
95         delete evt;
96         delete evt4;
97     }
98     //.....TERMINATION
99     hwefin();
100     std::cout << "testHerwigCopies: event comparison is successful" << std::endl;
101
102     return 0;
103 }
```

10.11 testPythiaCopies.cc

```

1
2 // testPythiaCopies.cc
3 //
4 // garren@fnal.gov, January 2008
5 // Multiple events in memory at the same time
6
7
8 #include <fstream>
9 #include <iostream>
10 #include "HepMC/PythiaWrapper.h"
11 #include "HepMC/IO_HEPEVT.h"
12 #include "HepMC/GenEvent.h"
13 #include "HepMC/CompareGenEvent.h"
14 #include "PythiaHelper.h"
15
16 int main() {
17     //
18     //.....HEPEVT
19     // Pythia 6.1 uses HEPEVT with 4000 entries and 8-byte floating point
20     // numbers. We need to explicitly pass this information to the
21     // HEPEVT_Wrapper.
22     //
23     HepMC::HEPEVT_Wrapper::set_max_number_entries(4000);
24     HepMC::HEPEVT_Wrapper::set_sizeof_real(8);
25     //
26     //.....PYTHIA INITIALIZATIONS
27     initPythia();
28     //
29     //.....HepMC INITIALIZATIONS
30     //
31     // Instantiate an IO strategy for reading from HEPEVT.
32     HepMC::IO_HEPEVT hepevtio;
33     //
34     // open some output files
35     std::ofstream out1( "testPythiaOriginals.dat" );
36     std::ofstream out2( "testPythiaCopies1.dat" );
37     std::ofstream out3( "testPythiaCopies2.dat" );
38     //
39     //.....EVENT LOOP
40     for ( int i = 1; i <= 50; i++ ) {
41         if ( i%50==1 ) std::cout << "Processing Event Number "
42             << i << std::endl;
43         call_pyevnt(); // generate one event with Pythia
44         // pythia pyhepc routine convert common PYJETS in common HEPEVT
45         call_pyhepc( 1 );
46         HepMC::GenEvent* evt = hepevtio.read_next_event();
47         // set number of multi parton interactions
48         evt->set_mpi( pypars.msti[31-1] );
49         //
50         //.....make some copies
51         evt->print(out1);
52         HepMC::GenEvent ec = (*evt);
53         ec.print(out2);
54         HepMC::GenEvent* evt4 = new HepMC::GenEvent(*evt);
55         evt4->print(out3);
56         if( !compareGenEvent(evt,evt4) ) {
57             std::cerr << "testPythiaCopies: GenEvent comparison fails at event "
58                 << evt->event_number() << std::endl;
59             return -1;
60         }
61         //
62         // now delete the created events from memory
63         delete evt;
64         delete evt4;
65     }
66     //.....TERMINATION

```

```
67     // write out some information from Pythia to the screen
68     call_pystat( 1 );
69     std::cout << "testPythiaCopies: event comparison is successful" << std::endl;
70
71     return 0;
72 }
73
74
75
```

10.12 VectorConversion.h

```
1 #ifndef VECTOR_CONVERSION_H
2 #define VECTOR_CONVERSION_H
3
4 // garren@fnal.gov, January 2007
5 //
6 // This example converts from ThreeVector and FourVector to
7 // CLHEP::Hep3Vector and CLHEP::HepLorentzVector
8 // Similar (or perhaps templated) conversion methods could be added to
9 // any vector class.
10 //
11
12
13 #include "HepMC/SimpleVector.h"
14 #include "CLHEP/Vector/LorentzVector.h"
15
16
17 CLHEP::Hep3Vector convertTo( const HepMC::ThreeVector& v )
18 { return CLHEP::Hep3Vector( v.x(), v.y(), v.z() ); }
19
20
21 CLHEP::HepLorentzVector convertTo( const HepMC::FourVector& v )
22 { return CLHEP::HepLorentzVector( v.x(), v.y(), v.z(), v.t() ); }
23
24
25 #endif // VECTOR_CONVERSION_H
```


Index

- /home/cepa01/garren/lcg/hepmc/HepMC-2.03.06/HepMC/ Directory Reference, 13
- /home/cepa01/garren/lcg/hepmc/HepMC-2.03.06/examples/ Directory Reference, 11
- /home/cepa01/garren/lcg/hepmc/HepMC-2.03.06/fio/ Directory Reference, 12
- /home/cepa01/garren/lcg/hepmc/HepMC-2.03.06/src/ Directory Reference, 14
- /home/cepa01/garren/lcg/hepmc/HepMC-2.03.06/test/ Directory Reference, 15
- ~CommonIO
 - HepMC::CommonIO, 30
- ~Flow
 - HepMC::Flow, 41
- ~GenEvent
 - HepMC::GenEvent, 59
- ~GenParticle
 - HepMC::GenParticle, 89
- ~GenVertex
 - HepMC::GenVertex, 99
- ~HeavyIon
 - HepMC::HeavyIon, 123
- ~IO_Ascii
 - HepMC::IO_Ascii, 143
- ~IO_AsciiParticles
 - HepMC::IO_AsciiParticles, 148
- ~IO_BaseClass
 - HepMC::IO_BaseClass, 152
- ~IO_ExtendedAscii
 - HepMC::IO_ExtendedAscii, 156
- ~IO_GenEvent
 - HepMC::IO_GenEvent, 165
- ~IO_HEPEVT
 - HepMC::IO_HEPEVT, 171
- ~IO_HERWIG
 - HepMC::IO_HERWIG, 177
- ~IO_PDG_ParticleDataTable
 - HepMC::IO_PDG_ParticleDataTable, 184
- ~ParticleData
 - HepMC::ParticleData, 208
- ~ParticleDataTable
 - HepMC::ParticleDataTable, 215
- ~PdfInfo
 - HepMC::PdfInfo, 221
- ~Polarization
 - HepMC::Polarization, 226
- ~TempParticleMap
 - HepMC::TempParticleMap, 230
- ~WeightContainer
 - HepMC::WeightContainer, 239
- ~edge_iterator
 - HepMC::GenVertex::edge_iterator, 110
- ~particle_const_iterator
 - HepMC::GenEvent::particle_const_iterator, 73
- ~particle_iterator
 - HepMC::GenEvent::particle_iterator, 76
 - HepMC::GenVertex::particle_iterator, 114
- ~vertex_const_iterator
 - HepMC::GenEvent::vertex_const_iterator, 80
- ~vertex_iterator
 - HepMC::GenEvent::vertex_iterator, 83
 - HepMC::GenVertex::vertex_iterator, 118
- add_particle_in
 - HepMC::GenVertex, 100
- add_particle_out
 - HepMC::GenVertex, 101
- add_quarks_to_table
 - HepMC::IO_PDG_ParticleDataTable, 184
- add_vertex
 - HepMC::GenEvent, 67
- addEndParticle
 - HepMC::TempParticleMap, 230
- advance_to_first_
 - HepMC::GenVertex::particle_iterator, 115
- AFCH
 - HerwigWrapper6_4.h, 278

- alphaQCD
 - HepMC::GenEvent, 61
- alphaQED
 - HepMC::GenEvent, 62
- ALPHEM
 - HerwigWrapper6_4.h, 278
- already_in_vector
 - HepMC, 22
- ancestors
 - HepMC, 22
- ascii
 - HepMC, 22
- ascii_pdt
 - HepMC, 22
- AVWGT
 - HerwigWrapper6_4.h, 279
- AZSOFT
 - HerwigWrapper6_4.h, 279
- AZSPIN
 - HerwigWrapper6_4.h, 279
- B1LIM
 - HerwigWrapper6_4.h, 279
- back
 - HepMC::WeightContainer, 241
- barcode
 - HepMC::GenParticle, 92
 - HepMC::GenVertex, 103
- barcode_to_particle
 - HepMC::GenEvent, 60
- barcode_to_vertex
 - HepMC::GenEvent, 60
- beam_particles
 - HepMC::GenEvent, 62
- begin
 - HepMC::Flow, 44
 - HepMC::ParticleDataTable, 218
 - HepMC::TempParticleMap, 230
 - HepMC::WeightContainer, 241, 242
- BETAF
 - HerwigWrapper6_4.h, 279
- brat
 - PythiaWrapper5_720.h, 319
 - PythiaWrapper6_152.h, 327
 - PythiaWrapper6_2.h, 335
- BTCLM
 - HerwigWrapper6_4.h, 279
- build_end_vertex
 - HepMC::IO_HEPEVT, 174
 - HepMC::IO_HERWIG, 180
- build_particle
 - HepMC::IO_HEPEVT, 173
 - HepMC::IO_HERWIG, 179
- build_production_vertex
 - HepMC::IO_HEPEVT, 174
 - HepMC::IO_HERWIG, 180
- byte_num_to_double
 - HepMC::HEPEVT_Wrapper, 140
- byte_num_to_int
 - HepMC::HEPEVT_Wrapper, 140
- CAFAC
 - HerwigWrapper6_4.h, 279
- CFFAC
 - HerwigWrapper6_4.h, 279
- change_parent_event
 - HepMC::GenVertex, 105
- charge
 - HepMC::ParticleData, 210
- check_hepevt_consistency
 - HepMC::HEPEVT_Wrapper, 132
- check_momentum_conservation
 - HepMC::GenVertex, 100
- children
 - HepMC, 22
- ckin
 - PythiaWrapper5_720.h, 319
 - PythiaWrapper6_152.h, 327
 - PythiaWrapper6_2.h, 335
- CLDIR
 - HerwigWrapper6_4.h, 279
- clear
 - HepMC::Flow, 43
 - HepMC::GenEvent, 67
 - HepMC::IO_Ascii, 145
 - HepMC::IO_AsciiParticles, 149
 - HepMC::IO_ExtendedAscii, 158
 - HepMC::IO_GenEvent, 166
 - HepMC::ParticleDataTable, 216
 - HepMC::WeightContainer, 240
- CLHEP, 17
- clifetime
 - HepMC::ParticleData, 211
- clifetime_from_width
 - HepMC, 23
- CLMAX
 - HerwigWrapper6_4.h, 279
- CLPOW
 - HerwigWrapper6_4.h, 279
- CLSMR
 - HerwigWrapper6_4.h, 279
- coef
 - PythiaWrapper5_720.h, 319
 - PythiaWrapper6_152.h, 327
 - PythiaWrapper6_2.h, 335
- CommonIO
 - HepMC::CommonIO, 30
- CommonIO.cc, 243

- CommonIO.h, 244
- compareBeamParticles
 - HepMC, 23
- compareGenEvent
 - HepMC, 23
- CompareGenEvent.cc, 245
- CompareGenEvent.h, 246
- compareParticles
 - HepMC, 23
- compareSignalProcessVertex
 - HepMC, 23
- compareVertex
 - HepMC, 24
- compareVertices
 - HepMC, 24
- compareWeights
 - HepMC, 24
- connected_partners
 - HepMC::Flow, 42, 44
- const_iterator
 - HepMC::Flow, 41
 - HepMC::ParticleDataTable, 215
 - HepMC::WeightContainer, 238
- convertTo
 - VectorConversion.h, 350
- copy_if
 - HepMC, 24
- copy_recursive_iterator_
 - HepMC::GenVertex::vertex_iterator, 119
- copy_with_own_set
 - HepMC::GenVertex::vertex_iterator, 119
- counter
 - HepMC::ParticleData, 212
- CSPEED
 - HerwigWrapper6_4.h, 280
- dangling_connected_partners
 - HepMC::Flow, 42, 44
- data
 - HEPEVT_Wrapper.h, 268
- delete_adopted_particles
 - HepMC::GenVertex, 106
- delete_all
 - HepMC::ParticleDataTable, 216
- delete_all_vertices
 - HepMC::GenEvent, 70
- descendants
 - HepMC, 22
- description
 - HepMC::ParticleDataTable, 219
- detail, 18
- e
 - HepMC::FourVector, 49
 - HepMC::HEPEVT_Wrapper, 136
- EBEAM1
 - HerwigWrapper6_4.h, 280
- EBEAM2
 - HerwigWrapper6_4.h, 280
- eccentricity
 - HepMC::HeavyIon, 126
- edge_iterator
 - HepMC::GenVertex, 107
 - HepMC::GenVertex::edge_iterator, 110
- edges_begin
 - HepMC::GenVertex, 105
- edges_end
 - HepMC::GenVertex, 105
- edges_size
 - HepMC::GenVertex, 105
- EFFMIN
 - HerwigWrapper6_4.h, 280
- empty
 - HepMC::Flow, 43
 - HepMC::ParticleDataTable, 217
 - HepMC::WeightContainer, 240
- enable_if.h, 247
- end
 - HepMC::Flow, 44
 - HepMC::ParticleDataTable, 218
 - HepMC::TempParticleMap, 230
 - HepMC::WeightContainer, 241, 242
- end_vertex
 - HepMC::GenParticle, 91
 - HepMC::TempParticleMap, 230
- ENSOFF
 - HerwigWrapper6_4.h, 280
- erase
 - HepMC::Flow, 43
 - HepMC::ParticleDataTable, 218
- ET2MIX
 - HerwigWrapper6_4.h, 280
- eta
 - HepMC::FourVector, 51
- ETAMIX
 - HerwigWrapper6_4.h, 280
- event_number
 - HepMC::GenEvent, 61
 - HepMC::HEPEVT_Wrapper, 133
- event_plane_angle
 - HepMC::HeavyIon, 126
- event_scale
 - HepMC::GenEvent, 61
- EVWGT
 - HerwigWrapper6_4.h, 280
- example_BuildEventFromScratch.cc, 248

- example_BuildEventFromScratch.cc
 - main, 248
- example_EventSelection.cc, 249
- example_EventSelection.cc
 - main, 249
- example_MyHerwig.cc, 250
- example_MyHerwig.cc
 - main, 250
- example_MyPythia.cc, 251
- example_MyPythia.cc
 - main, 251
- example_MyPythiaOnlyToHepMC.cc, 252
- example_MyPythiaOnlyToHepMC.cc
 - main, 252
- example_MyPythiaRead.cc, 253
- example_MyPythiaRead.cc
 - main, 253
- example_MyPythiaWithEventSelection.cc, 254
- example_MyPythiaWithEventSelection.cc
 - main, 254
- example_PythiaParticle.cc, 255
- example_PythiaParticle.cc
 - main, 255
- example_ReadMyPythia.cc, 256
- example_ReadMyPythia.cc
 - main, 256
- example_UsingIterators.cc, 257
- example_UsingIterators.cc
 - main, 257
- extascii
 - HepMC, 22
- extascii_pdt
 - HepMC, 22
- F0MIX
 - HerwigWrapper6_4.h, 280
- F1MIX
 - HerwigWrapper6_4.h, 280
- F2MIX
 - HerwigWrapper6_4.h, 280
- family
 - HepMC, 22
- fill_next_event
 - HepMC::IO_Ascii, 144
 - HepMC::IO_AsciiParticles, 148
 - HepMC::IO_BaseClass, 152
 - HepMC::IO_ExtendedAscii, 157
 - HepMC::IO_GenEvent, 165
 - HepMC::IO_HEPEVT, 171
 - HepMC::IO_HERWIG, 178
- fill_particle_data_table
 - HepMC::IO_Ascii, 144
 - HepMC::IO_AsciiParticles, 149
 - HepMC::IO_BaseClass, 152
 - HepMC::IO_ExtendedAscii, 157
 - HepMC::IO_GenEvent, 165
 - HepMC::IO_PDG_ParticleDataTable, 184
- find
 - HepMC::ParticleDataTable, 217
- find_end_key
 - HepMC::CommonIO, 32
- find_file_type
 - HepMC::CommonIO, 32
- find_in_map
 - HepMC::IO_HEPEVT, 174
 - HepMC::IO_HERWIG, 180
- first_child
 - HepMC::HEPEVT_Wrapper, 134
- first_parent
 - HepMC::HEPEVT_Wrapper, 133
- Flow
 - HepMC::Flow, 41
- flow
 - HepMC::GenParticle, 90
- Flow.cc, 258
- Flow.h, 259
- follow_edge_
 - HepMC::GenVertex::vertex_iterator, 119
- FourVector
 - HepMC::FourVector, 48, 49
- front
 - HepMC::WeightContainer, 241
- GAMH
 - HerwigWrapper6_4.h, 280
- GAMW
 - HerwigWrapper6_4.h, 281
- GAMWT
 - HerwigWrapper6_4.h, 281
- GAMZ
 - HerwigWrapper6_4.h, 281
- GAMZP
 - HerwigWrapper6_4.h, 281
- GCUTME
 - HerwigWrapper6_4.h, 281
- gen
 - HepMC, 22
- generated_mass
 - HepMC::GenParticle, 91
- generatedMass
 - HepMC::GenParticle, 92
- GenEvent
 - HepMC::GenEvent, 58, 59
 - HepMC::GenParticle, 94
 - HepMC::GenVertex, 107

- GenEvent.cc, 260
- GenEvent.h, 261
- GenParticle
 - HepMC::GenEvent, 70
 - HepMC::GenParticle, 88
- GenParticle.cc, 262
- GenParticle.h, 263
- GenParticle.h
 - hepmc_uint64_t, 263
- GENSOF
 - HerwigWrapper6_4.h, 281
- GenVertex
 - HepMC::GenEvent, 70
 - HepMC::GenParticle, 94
 - HepMC::GenVertex, 99
- GenVertex.cc, 264
- GenVertex.h, 265
- GEV2NB
 - HerwigWrapper6_4.h, 281
- H1MIX
 - HerwigWrapper6_4.h, 281
- HARDME
 - HerwigWrapper6_4.h, 281
- heavy_ion
 - HepMC::GenEvent, 63
- HeavyIon
 - HepMC::HeavyIon, 123
- HeavyIon.h, 266
- hepevt
 - HEPEVT_Wrapper.h, 268
- hepevt_
 - HEPEVT_Wrapper.h, 268
- hepevt_bytes_allocation
 - HEPEVT_Wrapper.h, 269
- HEPEVT_EntriesAllocation
 - HEPEVT_Wrapper.h, 268
- HEPEVT_Wrapper.cc, 267
- HEPEVT_Wrapper.h, 268
 - data, 268
 - hepevt, 268
 - hepevt_, 268
 - hepevt_bytes_allocation, 269
 - HEPEVT_EntriesAllocation, 268
- HepLorentzVector
 - HepMC_CLHEP20.h, 270
- HepMC, 19
 - ancestors, 22
 - ascii, 22
 - ascii_pdt, 22
 - children, 22
 - descendants, 22
 - extascii, 22
 - extascii_pdt, 22
 - family, 22
 - gen, 22
 - parents, 22
 - relatives, 22
- HepMC
 - already_in_vector, 22
 - clifetime_from_width, 23
 - compareBeamParticles, 23
 - compareGenEvent, 23
 - compareParticles, 23
 - compareSignalProcessVertex, 23
 - compareVertex, 24
 - compareVertices, 24
 - compareWeights, 24
 - copy_if, 24
 - HepMC_hbarc, 26
 - HepMC_pi, 26
 - IteratorRange, 22
 - known_io, 22
 - not_in_vector, 24
 - operator<<, 25
 - version, 26
 - versionName, 26
 - writeVersion, 26
- HepMC::CommonIO, 29
- HepMC::CommonIO
 - ~CommonIO, 30
 - CommonIO, 30
 - find_end_key, 32
 - find_file_type, 32
 - IO_Ascii_End, 30
 - IO_Ascii_Key, 30
 - IO_ExtendedAscii_End, 30
 - IO_ExtendedAscii_Key, 30
 - IO_GenEvent_End, 30
 - IO_GenEvent_Key, 30
 - io_type, 31
 - read_heavy_ion, 33
 - read_io_ascii, 32
 - read_io_extendedascii, 33
 - read_io_genevent, 33
 - read_io_particle_data_table, 33
 - read_particle, 34
 - read_particle_data, 34
 - read_pdf_info, 34
 - read_vertex, 34
 - write_IO_Ascii_End, 31
 - write_IO_Ascii_Key, 31
 - write_IO_Ascii_PDT_End, 32
 - write_IO_Ascii_PDT_Key, 31
 - write_IO_ExtendedAscii_End, 31
 - write_IO_ExtendedAscii_Key, 31
 - write_IO_ExtendedAscii_PDT_End, 32

- write_IO_ExtendedAscii_PDT_Key, 32
- write_IO_GenEvent_End, 31
- write_IO_GenEvent_Key, 31
- HepMC::detail, 27
- HepMC::detail::disable_if, 35
- HepMC::detail::disable_if< false, T >, 36
- HepMC::detail::disable_if< false, T > type, 36
- HepMC::detail::enable_if, 37
- HepMC::detail::enable_if< true, T >, 38
- HepMC::detail::enable_if< true, T > type, 38
- HepMC::detail::is_arithmetic, 186
- HepMC::detail::is_arithmetic value, 186
- HepMC::detail::is_arithmetic< char >, 187
- HepMC::detail::is_arithmetic< char > value, 187
- HepMC::detail::is_arithmetic< double >, 188
- HepMC::detail::is_arithmetic< double > value, 188
- HepMC::detail::is_arithmetic< float >, 189
- HepMC::detail::is_arithmetic< float > value, 189
- HepMC::detail::is_arithmetic< int >, 190
- HepMC::detail::is_arithmetic< int > value, 190
- HepMC::detail::is_arithmetic< long >, 191
- HepMC::detail::is_arithmetic< long > value, 191
- HepMC::detail::is_arithmetic< long double >, 192
- HepMC::detail::is_arithmetic< long double > value, 192
- HepMC::detail::is_arithmetic< short >, 193
- HepMC::detail::is_arithmetic< short > value, 193
- HepMC::detail::is_arithmetic< signed char >, 194
- HepMC::detail::is_arithmetic< signed char > value, 194
- HepMC::detail::is_arithmetic< unsigned char >, 195
- HepMC::detail::is_arithmetic< unsigned char > value, 195
- HepMC::detail::is_arithmetic< unsigned int >, 196
- HepMC::detail::is_arithmetic< unsigned int > value, 196
- HepMC::detail::is_arithmetic< unsigned long >, 197
- HepMC::detail::is_arithmetic< unsigned long > value, 197
- HepMC::detail::is_arithmetic< unsigned short >, 198
- HepMC::detail::is_arithmetic< unsigned short > value, 198
- HepMC::Flow, 39
- HepMC::Flow
 - ~Flow, 41
 - begin, 44
 - clear, 43
 - connected_partners, 42, 44
 - const_iterator, 41
 - dangling_connected_partners, 42, 44
 - empty, 43
 - end, 44
 - erase, 43
 - Flow, 41
 - icode, 43
 - iterator, 41
 - operator!=, 42
 - operator<<, 45
 - operator=, 41
 - operator==, 42
 - particle_owner, 43
 - print, 42
 - set_icode, 43
 - set_unique_icode, 43
 - size, 43
 - swap, 41
- HepMC::FourVector, 46
- HepMC::FourVector
 - e, 49
 - eta, 51
 - FourVector, 48, 49
 - m, 50
 - m2, 50
 - mag, 50
 - operator!=, 51
 - operator=, 51
 - operator==, 51
 - perp, 50
 - perp2, 50
 - phi, 51
 - pseudoRapidity, 51
 - px, 49
 - py, 49
 - pz, 49
 - rho, 51

- set, 51
- setE, 53
- setPx, 52
- setPy, 52
- setPz, 52
- setT, 52
- setX, 52
- setY, 52
- setZ, 52
- swap, 49
- t, 50
- theta, 51
- x, 49
- y, 50
- z, 50
- HepMC::GenEvent, 54
- HepMC::GenEvent
 - ~GenEvent, 59
 - add_vertex, 67
 - alphaQCD, 61
 - alphaQED, 62
 - barcode_to_particle, 60
 - barcode_to_vertex, 60
 - beam_particles, 62
 - clear, 67
 - delete_all_vertices, 70
 - event_number, 61
 - event_scale, 61
 - GenEvent, 58, 59
 - GenParticle, 70
 - GenVertex, 70
 - heavy_ion, 63
 - mpi, 61
 - operator=, 59
 - particle_const_iterator, 71
 - particle_iterator, 71
 - particles_begin, 68, 69
 - particles_empty, 66
 - particles_end, 69
 - particles_size, 66
 - pdf_info, 63
 - print, 59
 - print_version, 60
 - random_states, 63
 - remove_barcode, 69, 70
 - remove_vertex, 67
 - set_alphaQCD, 64
 - set_alphaQED, 65
 - set_barcode, 69
 - set_beam_particles, 65
 - set_event_number, 64
 - set_event_scale, 64
 - set_heavy_ion, 66
 - set_mpi, 64
 - set_pdf_info, 66
 - set_random_states, 66
 - set_signal_process_id, 64
 - set_signal_process_vertex, 65
 - signal_process_id, 61
 - signal_process_vertex, 62
 - swap, 59
 - valid_beam_particles, 62
 - vertex_const_iterator, 70
 - vertex_iterator, 70
 - vertices_begin, 67, 68
 - vertices_empty, 67
 - vertices_end, 68
 - vertices_size, 66
 - weights, 62, 63
- HepMC::GenEvent::particle_const_iterator, 72
- HepMC::GenEvent::particle_const_iterator
 - ~particle_const_iterator, 73
 - m_map_iterator, 74
 - operator *, 73
 - operator!=, 74
 - operator++, 73, 74
 - operator=, 73
 - operator==, 74
 - particle_const_iterator, 73
- HepMC::GenEvent::particle_iterator, 75
- HepMC::GenEvent::particle_iterator
 - ~particle_iterator, 76
 - m_map_iterator, 77
 - operator *, 76
 - operator particle_const_iterator, 76
 - operator!=, 77
 - operator++, 77
 - operator=, 76
 - operator==, 77
 - particle_iterator, 76
- HepMC::GenEvent::vertex_const_iterator, 79
- HepMC::GenEvent::vertex_const_iterator
 - ~vertex_const_iterator, 80
 - m_map_iterator, 81
 - operator *, 80
 - operator!=, 81
 - operator++, 80, 81
 - operator=, 80
 - operator==, 81
 - vertex_const_iterator, 80
- HepMC::GenEvent::vertex_iterator, 82
- HepMC::GenEvent::vertex_iterator
 - ~vertex_iterator, 83
 - m_map_iterator, 84
 - operator *, 84
 - operator vertex_const_iterator, 83

- operator!=, 84
- operator++, 84
- operator=, 83
- operator==, 84
- vertex_iterator, 83
- HepMC::GenParticle, 86
- HepMC::GenParticle
 - ~GenParticle, 89
 - barcode, 92
 - end_vertex, 91
 - flow, 90
 - generated_mass, 91
 - generatedMass, 92
 - GenEvent, 94
 - GenParticle, 88
 - GenVertex, 94
 - momentum, 90
 - operator HepMC::FourVector, 90
 - operator!=, 89
 - operator<<, 94
 - operator=, 89
 - operator==, 89
 - parent_event, 91
 - pdg_id, 90
 - polarization, 91
 - print, 90
 - production_vertex, 91
 - set_barcode_, 94
 - set_end_vertex_, 94
 - set_flow, 93
 - set_generated_mass, 93
 - set_momentum, 92
 - set_pdg_id, 92
 - set_polarization, 93
 - set_production_vertex_, 93
 - set_status, 92
 - setGeneratedMass, 93
 - status, 90
 - suggest_barcode, 92
 - swap, 89
- HepMC::GenVertex, 95
- HepMC::GenVertex
 - ~GenVertex, 99
 - add_particle_in, 100
 - add_particle_out, 101
 - barcode, 103
 - change_parent_event_, 105
 - check_momentum_conservation, 100
 - delete_adopted_particles, 106
 - edge_iterator, 107
 - edges_begin, 105
 - edges_end, 105
 - edges_size, 105
 - GenEvent, 107
 - GenVertex, 99
 - id, 102
 - operator HepMC::FourVector, 101
 - operator HepMC::ThreeVector, 102
 - operator!=, 100
 - operator<<, 107
 - operator=, 99
 - operator==, 100
 - parent_event, 102
 - particle_iterator, 107
 - particles_begin, 106
 - particles_end, 106
 - particles_in_const_begin, 104
 - particles_in_const_end, 104
 - particles_in_const_iterator, 98
 - particles_in_size, 104
 - particles_out_const_begin, 104
 - particles_out_const_end, 104
 - particles_out_const_iterator, 98
 - particles_out_size, 104
 - point3d, 102
 - position, 102
 - print, 100
 - remove_particle, 101
 - remove_particle_in, 107
 - remove_particle_out, 107
 - set_barcode_, 105
 - set_id, 103
 - set_parent_event_, 105
 - set_position, 102
 - suggest_barcode, 103
 - swap, 99
 - vertex_iterator, 107
 - vertices_begin, 106
 - vertices_end, 106
 - weights, 103
- HepMC::GenVertex::edge_iterator, 109
- HepMC::GenVertex::edge_iterator
 - ~edge_iterator, 110
 - edge_iterator, 110
 - is_child, 111
 - is_parent, 111
 - operator *, 110
 - operator!=, 111
 - operator++, 110, 111
 - operator=, 110
 - operator==, 111
 - vertex_root, 111
- HepMC::GenVertex::particle_iterator, 113
- HepMC::GenVertex::particle_iterator
 - ~particle_iterator, 114
 - advance_to_first_, 115
 - operator *, 114
 - operator!=, 115

- operator++, 114, 115
- operator=, 114
- operator==, 115
- particle_iterator, 114
- HepMC::GenVertex::vertex_iterator, 116
- HepMC::GenVertex::vertex_iterator
 - ~vertex_iterator, 118
 - copy_recursive_iterator_, 119
 - copy_with_own_set, 119
 - follow_edge_, 119
 - operator*, 118
 - operator!=, 118
 - operator++, 118
 - operator=, 118
 - operator==, 118
 - range, 119
 - vertex_iterator, 117
 - vertex_root, 119
- HepMC::HeavyIon, 121
- HepMC::HeavyIon
 - ~HeavyIon, 123
 - eccentricity, 126
 - event_plane_angle, 126
 - HeavyIon, 123
 - impact_parameter, 126
 - N_Nwounded_collisions, 125
 - Ncoll, 125
 - Ncoll_hard, 124
 - Npart_proj, 124
 - Npart_targ, 125
 - Nwounded_N_collisions, 125
 - Nwounded_Nwounded_collisions, 126
 - operator!=, 124
 - operator=, 124
 - operator==, 124
 - set_eccentricity, 128
 - set_event_plane_angle, 128
 - set_impact_parameter, 127
 - set_N_Nwounded_collisions, 127
 - set_Ncoll, 127
 - set_Ncoll_hard, 126
 - set_Npart_proj, 126
 - set_Npart_targ, 127
 - set_Nwounded_N_collisions, 127
 - set_Nwounded_Nwounded_collisions, 127
 - set_sigma_inel_NN, 128
 - set_spectator_neutrons, 127
 - set_spectator_protons, 127
 - sigma_inel_NN, 126
 - spectator_neutrons, 125
 - spectator_protons, 125
 - swap, 124
- HepMC::HEPEVT_Wrapper
 - byte_num_to_double, 140
 - byte_num_to_int, 140
 - check_hepevt_consistency, 132
 - e, 136
 - event_number, 133
 - first_child, 134
 - first_parent, 133
 - id, 133
 - is_double_precision, 132
 - last_child, 134
 - last_parent, 134
 - m, 136
 - max_number_entries, 139
 - number_children, 135
 - number_entries, 133
 - number_parents, 134
 - print_hepevt, 132
 - print_hepevt_particle, 132
 - print_legend, 141
 - px, 135
 - py, 135
 - pz, 135
 - set_children, 138
 - set_event_number, 137
 - set_id, 137
 - set_mass, 138
 - set_max_number_entries, 140
 - set_momentum, 138
 - set_number_entries, 137
 - set_parents, 138
 - set_position, 139
 - set_sizeof_int, 139
 - set_sizeof_real, 140
 - set_status, 137
 - sizeof_int, 139
 - sizeof_real, 139
 - status, 133
 - t, 137
 - write_byte_num, 141
 - x, 136
 - y, 136
 - z, 136
 - zero_everything, 132
- HepMC::IO_Ascii, 142
- HepMC::IO_Ascii
 - ~IO_Ascii, 143
 - clear, 145
 - fill_next_event, 144
 - fill_particle_data_table, 144
 - IO_Ascii, 143
 - output, 146
 - print, 145
 - rdstate, 144

- write_comment, 144
- write_end_listing, 145
- write_event, 143
- write_particle, 145
- write_particle_data, 145
- write_particle_data_table, 144
- write_vertex, 145
- HepMC::IO_AsciiParticles, 147
- HepMC::IO_AsciiParticles
 - ~IO_AsciiParticles, 148
 - clear, 149
 - fill_next_event, 148
 - fill_particle_data_table, 149
 - IO_AsciiParticles, 148
 - print, 149
 - rdstate, 149
 - setPrecision, 149
 - write_comment, 149
 - write_end_listing, 149
 - write_event, 148
 - write_particle_data_table, 148
- HepMC::IO_BaseClass, 151
- HepMC::IO_BaseClass
 - ~IO_BaseClass, 152
 - fill_next_event, 152
 - fill_particle_data_table, 152
 - operator<<, 154
 - operator>>, 153, 154
 - print, 153
 - read_next_event, 153
 - read_particle_data_table, 153
 - write_event, 152
 - write_particle_data_table, 152
- HepMC::IO_ExtendedAscii, 155
- HepMC::IO_ExtendedAscii
 - ~IO_ExtendedAscii, 156
 - clear, 158
 - fill_next_event, 157
 - fill_particle_data_table, 157
 - IO_ExtendedAscii, 156
 - output, 160
 - print, 158
 - rdstate, 158
 - write_beam_particles, 158
 - write_comment, 158
 - write_end_listing, 160
 - write_event, 157
 - write_heavy_ion, 159
 - write_particle, 159
 - write_particle_data, 159
 - write_particle_data_table, 157
 - write_pdf_info, 159
 - write_vertex, 158
- HepMC::IO_GenEvent, 162
- HepMC::IO_GenEvent
 - ~IO_GenEvent, 165
 - clear, 166
 - fill_next_event, 165
 - fill_particle_data_table, 165
 - IO_GenEvent, 164
 - output, 168, 169
 - print, 166
 - rdstate, 166
 - read_heavy_ion, 168
 - read_particle, 168
 - read_particle_data, 168
 - read_pdf_info, 168
 - read_vertex, 167
 - write_beam_particles, 166
 - write_comment, 166
 - write_end_listing, 168
 - write_event, 165
 - write_heavy_ion, 167
 - write_particle, 167
 - write_particle_data, 167
 - write_particle_data_table, 165
 - write_pdf_info, 167
 - write_vertex, 166
- HepMC::IO_HEPEVT, 170
- HepMC::IO_HEPEVT
 - ~IO_HEPEVT, 171
 - build_end_vertex, 174
 - build_particle, 173
 - build_production_vertex, 174
 - fill_next_event, 171
 - find_in_map, 174
 - IO_HEPEVT, 171
 - print, 172
 - print_inconsistency_errors, 172
 - set_print_inconsistency_errors, 173
 - set_trust_beam_particles, 173
 - set_trust_both_mothers_and_daughters, 173
 - set_trust_mothers_before_daughters, 173
 - trust_beam_particles, 172
 - trust_both_mothers_and_daughters, 172
 - trust_mothers_before_daughters, 172
 - write_event, 172
- HepMC::IO_HERWIG, 176
- HepMC::IO_HERWIG
 - ~IO_HERWIG, 177
 - build_end_vertex, 180
 - build_particle, 179
 - build_production_vertex, 180
 - fill_next_event, 178
 - find_in_map, 180

- interfaces_to_version_number, 178
- IO_HERWIG, 177
- no_gaps_in_barcodes, 178
- print, 178
- print_inconsistency_errors, 178
- remove_gaps_in_hepevt, 181
- repair_hepevt, 180
- set_no_gaps_in_barcodes, 179
- set_print_inconsistency_errors, 178
- set_trust_both_mothers_and_daughters, 179
- set_trust_mothers_before_daughters, 179
- translate_herwig_to_pdg_id, 182
- trust_both_mothers_and_daughters, 179
- trust_mothers_before_daughters, 179
- zero_hepevt_entry, 181
- HepMC::IO_PDG_ParticleDataTable, 183
- HepMC::IO_PDG_ParticleDataTable
 - ~IO_PDG_ParticleDataTable, 184
 - add_quarks_to_table, 184
 - fill_particle_data_table, 184
 - IO_PDG_ParticleDataTable, 184
 - print, 184
 - rdstate, 184
 - read_entry, 185
 - search_for_key_end, 184
- HepMC::ParticleData, 206
- HepMC::ParticleData
 - ~ParticleData, 208
 - charge, 210
 - clifetime, 211
 - counter, 212
 - is_boson, 210
 - is_charged_lepton, 209
 - is_em, 209
 - is_hadron, 209
 - is_lepton, 209
 - is_neutrino, 209
 - mass, 210
 - model_independent_pdg_id_, 212
 - name, 210
 - operator!=, 208
 - operator<=, 212
 - operator==, 208
 - ParticleData, 208
 - pdg_id, 210
 - print, 209
 - set_charge, 211
 - set_clifetime, 211
 - set_mass, 211
 - set_spin, 212
 - set_width, 211
 - spin, 211
 - width, 211
- HepMC::ParticleDataTable, 213
- HepMC::ParticleDataTable
 - ~ParticleDataTable, 215
 - begin, 218
 - clear, 216
 - const_iterator, 215
 - delete_all, 216
 - description, 219
 - empty, 217
 - end, 218
 - erase, 218
 - find, 217
 - insert, 217
 - iterator, 215
 - make_antiparticles_from_particles, 216
 - merge_table, 216
 - operator=, 215
 - operator[], 217
 - ParticleDataTable, 215
 - print, 216
 - set_description, 219
 - size, 217
- HepMC::PdfInfo, 220
- HepMC::PdfInfo
 - ~PdfInfo, 221
 - id1, 222
 - id2, 222
 - operator!=, 222
 - operator=, 222
 - operator==, 222
 - pdf1, 223
 - pdf2, 223
 - PdfInfo, 221
 - scalePDF, 223
 - set_id1, 223
 - set_id2, 223
 - set_pdf1, 224
 - set_pdf2, 224
 - set_scalePDF, 224
 - set_x1, 224
 - set_x2, 224
 - swap, 222
 - x1, 223
 - x2, 223
- HepMC::Polarization, 225
- HepMC::Polarization
 - ~Polarization, 226
 - normal3d, 227
 - operator!=, 227
 - operator<=, 228
 - operator=, 227

- operator==, 227
- phi, 227
- Polarization, 226
- print, 227
- set_normal3d, 228
- set_phi, 228
- set_theta, 228
- set_theta_phi, 228
- swap, 226
- theta, 227
- HepMC::TempParticleMap, 229
- HepMC::TempParticleMap
 - ~TempParticleMap, 230
 - addEndParticle, 230
 - begin, 230
 - end, 230
 - end_vertex, 230
 - order_begin, 230
 - order_end, 230
 - orderIterator, 229
 - TempMap, 229
 - TempMapIterator, 229
 - TempOrderMap, 229
 - TempParticleMap, 230
- HepMC::ThreeVector, 232
- HepMC::ThreeVector
 - mag, 235
 - operator!=, 236
 - operator=, 236
 - operator==, 236
 - perp, 236
 - perp2, 236
 - phi, 235
 - r, 235
 - set, 235
 - setPhi, 235
 - setTheta, 235
 - setX, 234
 - setY, 234
 - setZ, 235
 - swap, 234
 - theta, 235
 - ThreeVector, 233, 234
 - x, 234
 - y, 234
 - z, 234
- HepMC::WeightContainer, 237
- HepMC::WeightContainer
 - ~WeightContainer, 239
 - back, 241
 - begin, 241, 242
 - clear, 240
 - const_iterator, 238
 - empty, 240
 - end, 241, 242
 - front, 241
 - iterator, 238
 - operator=, 239, 240
 - operator[], 241
 - pop_back, 240
 - print, 240
 - push_back, 240
 - size, 240
 - swap, 239
 - WeightContainer, 239
- HepMC_CLHEP20.h, 270
- HepMC_CLHEP20.h
 - HepLorentzVector, 270
 - HepNormal3D, 270
 - HepPoint3D, 270
- HepMC_hbarc
 - HepMC, 26
- HepMC_pi
 - HepMC, 26
- hepmc_uint64_t
 - GenParticle.h, 263
- HepNormal3D
 - HepMC_CLHEP20.h, 270
- HepPoint3D
 - HepMC_CLHEP20.h, 270
- herwig_hepevt_size
 - HerwigWrapper6_4.h, 281
- HerwigWrapper.h, 271
- HerwigWrapper6_4.h, 272
- HerwigWrapper6_4.h
 - AFCH, 278
 - ALPHEM, 278
 - AVWGT, 279
 - AZSOFT, 279
 - AZSPIN, 279
 - B1LIM, 279
 - BETAF, 279
 - BTCLM, 279
 - CAFAC, 279
 - CFFAC, 279
 - CLDIR, 279
 - CLMAX, 279
 - CLPOW, 279
 - CLSMR, 279
 - CSPEED, 280
 - EBEAM1, 280
 - EBEAM2, 280
 - EFFMIN, 280
 - ENSOF, 280
 - ET2MIX, 280
 - ETAMIX, 280
 - EVWGT, 280
 - F0MIX, 280

F1MIX, 280
F2MIX, 280
GAMH, 280
GAMW, 281
GAMWT, 281
GAMZ, 281
GAMZP, 281
GCUTME, 281
GENSOF, 281
GEV2NB, 281
H1MIX, 281
HARDME, 281
herwig_hepevt_size, 281
hwbgen, 275
hwbmch, 275
hwbmch_, 281
hwcdec, 275
hwcfor, 275
hwdhad, 275
hwdhob, 275
hwdhvy, 276
hwefin, 276
hwegup, 276
hweini, 276
hwepro, 276
hwevnt, 276
hwevnt_, 282
hwigin, 277
hwigup, 277
hwmevt, 277
hwpram, 277
hwpram_, 282
hwproc, 277
hwproc_, 282
hwudat, 277
hwudpr, 277
hwuepr, 277
hwufne, 278
hwuinc, 278
hwuine, 278
hwupro, 278
hwupup, 278
hwusta, 278
IDHW, 282
IERROR, 282
IOP4JT, 282
IOPREM, 282
IPRINT, 282
IPROC, 282
ISPAC, 282
ISTAT, 282
LRSUD, 282
LWEVT, 282
LWSUD, 282
MAXER, 283
MAXEV, 283
MAXPR, 283
MODPDF, 283
NBTRY, 283
NCOLO, 283
NCTRY, 283
NDTRY, 283
NETRY, 283
NFLAV, 283
NGSPL, 283
NOSPAC, 283
NOWGT, 284
NPRFMT, 284
NRN, 284
NSTRU, 284
NSTRY, 284
NUMER, 284
NUMERU, 284
NWGTS, 284
NZBIN, 284
OMHMIX, 284
PART1, 284
PART2, 284
PBEAM1, 285
PBEAM2, 285
PDIQK, 285
PGSMX, 285
PGSPL, 285
PH3MIX, 285
PHIMIX, 285
PIFAC, 285
PRNDEC, 285
PRNDEF, 285
PRNTEX, 285
PRNWEB, 285
PRSOFT, 286
PRVTX, 286
PSPLT, 286
PTRMS, 286
PXRMS, 286
QCDL3, 286
QCDL5, 286
QCDLAM, 286
QDIQK, 286
QFCH, 286
QG, 286
QSPAC, 286
QV, 287
SCABI, 287
SOFTME, 287
SWEIN, 287
TLOUT, 287
TMTOP, 287

- VCKM, 287
- VFCH, 287
- VGCUT, 287
- VPCUT, 287
- VQCUT, 287
- WBGST, 287
- WGTMAX, 288
- WGTSUM, 288
- WSQSUM, 288
- ZBINM, 288
- ZPRIME, 288
- hwbgen
 - HerwigWrapper6_4.h, 275
- hwbmch
 - HerwigWrapper6_4.h, 275
- hwbmch_
 - HerwigWrapper6_4.h, 281
- hwcdec
 - HerwigWrapper6_4.h, 275
- hwcfor
 - HerwigWrapper6_4.h, 275
- hwdhad
 - HerwigWrapper6_4.h, 275
- hwdhob
 - HerwigWrapper6_4.h, 275
- hwdhvy
 - HerwigWrapper6_4.h, 276
- hwefin
 - HerwigWrapper6_4.h, 276
- hwegup
 - HerwigWrapper6_4.h, 276
- hweini
 - HerwigWrapper6_4.h, 276
- hwepro
 - HerwigWrapper6_4.h, 276
- hwevnt
 - HerwigWrapper6_4.h, 276
- hwevnt_
 - HerwigWrapper6_4.h, 282
- hwigin
 - HerwigWrapper6_4.h, 277
- hwigup
 - HerwigWrapper6_4.h, 277
- hwmevt
 - HerwigWrapper6_4.h, 277
- hwpram
 - HerwigWrapper6_4.h, 277
- hwpram_
 - HerwigWrapper6_4.h, 282
- hwproc
 - HerwigWrapper6_4.h, 277
- hwproc_
 - HerwigWrapper6_4.h, 282
- hwudat
 - HerwigWrapper6_4.h, 277
- hwudpr
 - HerwigWrapper6_4.h, 277
- hwuepr
 - HerwigWrapper6_4.h, 277
- hwufne
 - HerwigWrapper6_4.h, 278
- hwuinc
 - HerwigWrapper6_4.h, 278
- hwuine
 - HerwigWrapper6_4.h, 278
- hwupro
 - HerwigWrapper6_4.h, 278
- hwupup
 - HerwigWrapper6_4.h, 278
- hwusta
 - HerwigWrapper6_4.h, 278
- icode
 - HepMC::Flow, 43
- icol
 - PythiaWrapper5_720.h, 319
 - PythiaWrapper6_152.h, 327
 - PythiaWrapper6_2.h, 335
- id
 - HepMC::GenVertex, 102
 - HepMC::HEPEVT_Wrapper, 133
- id1
 - HepMC::PdfInfo, 222
- id2
 - HepMC::PdfInfo, 222
- IDHW
 - HerwigWrapper6_4.h, 282
- IERROR
 - HerwigWrapper6_4.h, 282
- impact_parameter
 - HepMC::HeavyIon, 126
- initpydata
 - PythiaWrapper5_720.h, 317
 - PythiaWrapper6_152.h, 325
 - PythiaWrapper6_2.h, 334, 335
- initPythia
 - initPythia.cc, 289
 - PythiaHelper.h, 314
- initPythia.cc, 289
- initPythia.cc
 - initPythia, 289
- insert
 - HepMC::ParticleDataTable, 217
- interfaces_to_version_number
 - HepMC::IO_HERWIG, 178
- IO_Ascii
 - HepMC::IO_Ascii, 143
- IO_Ascii.cc, 290

- IO_Ascii.h, 291
- IO_Ascii_End
 - HepMC::CommonIO, 30
- IO_Ascii_Key
 - HepMC::CommonIO, 30
- IO_AsciiParticles
 - HepMC::IO_AsciiParticles, 148
- IO_AsciiParticles.cc, 292
- IO_AsciiParticles.h, 293
- IO_BaseClass.h, 294
- IO_ExtendedAscii
 - HepMC::IO_ExtendedAscii, 156
- IO_ExtendedAscii.cc, 295
- IO_ExtendedAscii.h, 296
- IO_ExtendedAscii_End
 - HepMC::CommonIO, 30
- IO_ExtendedAscii_Key
 - HepMC::CommonIO, 30
- IO_GenEvent
 - HepMC::IO_GenEvent, 164
- IO_GenEvent.cc, 297
- IO_GenEvent.h, 298
- IO_GenEvent_End
 - HepMC::CommonIO, 30
- IO_GenEvent_Key
 - HepMC::CommonIO, 30
- IO_HEPEVT
 - HepMC::IO_HEPEVT, 171
- IO_HEPEVT.cc, 299
- IO_HEPEVT.h, 300
- IO_HERWIG
 - HepMC::IO_HERWIG, 177
- IO_HERWIG.cc, 301
- IO_HERWIG.h, 302
- IO_PDG_ParticleDataTable
 - HepMC::IO_PDG_ParticleDataTable, 184
- IO_PDG_ParticleDataTable.cc, 303
- IO_PDG_ParticleDataTable.h, 304
- io_type
 - HepMC::CommonIO, 31
- IOP4JT
 - HerwigWrapper6_4.h, 282
- IOPREM
 - HerwigWrapper6_4.h, 282
- IPRINT
 - HerwigWrapper6_4.h, 282
- IPROC
 - HerwigWrapper6_4.h, 282
- is_arithmetic.h, 305
- is_boson
 - HepMC::ParticleData, 210
- is_charged_lepton
 - HepMC::ParticleData, 209
- is_child
 - HepMC::GenVertex::edge_iterator, 111
- is_double_precision
 - HepMC::HEPEVT_Wrapper, 132
- is_em
 - HepMC::ParticleData, 209
- is_hadron
 - HepMC::ParticleData, 209
- is_lepton
 - HepMC::ParticleData, 209
- is_neutrino
 - HepMC::ParticleData, 209
- is_parent
 - HepMC::GenVertex::edge_iterator, 111
- iset
 - PythiaWrapper5_720.h, 319
 - PythiaWrapper6_152.h, 327
 - PythiaWrapper6_2.h, 336
- IsEventGood, 199
- IsEventGood
 - operator(), 199
- IsFinalState, 200
- IsFinalState
 - operator(), 200
- IsGoodEvent, 201
- IsGoodEvent
 - operator(), 201
- IsGoodEvent.h, 306
- IsGoodEventMyPythia, 202
- IsGoodEventMyPythia
 - operator(), 202
- ISPAC
 - HerwigWrapper6_4.h, 282
- IsPhoton, 203
 - testHepMCIteration.h, 345
- IsPhoton
 - operator(), 203
- IsStateFinal, 204
- IsStateFinal
 - operator(), 204
- ISTAT
 - HerwigWrapper6_4.h, 282
- IsW_Boson, 205
- IsW_Boson
 - operator(), 205
- IsWBoson
 - testHepMCIteration.h, 345
- iterator
 - HepMC::Flow, 41
 - HepMC::ParticleDataTable, 215
 - HepMC::WeightContainer, 238
- IteratorRange
 - HepMC, 22

- k
 - PythiaWrapper5_720.h, 319
 - PythiaWrapper6_152.h, 327
 - PythiaWrapper6_2.h, 336
- kchg
 - PythiaWrapper5_720.h, 320
 - PythiaWrapper6_152.h, 327
 - PythiaWrapper6_2.h, 336
- kfdp
 - PythiaWrapper5_720.h, 320
 - PythiaWrapper6_152.h, 327
 - PythiaWrapper6_2.h, 336
- kfin
 - PythiaWrapper5_720.h, 320
 - PythiaWrapper6_152.h, 328
 - PythiaWrapper6_2.h, 336
- kfpr
 - PythiaWrapper5_720.h, 320
 - PythiaWrapper6_152.h, 328
 - PythiaWrapper6_2.h, 336
- known_io
 - HepMC, 22
- last_child
 - HepMC::HEPEVT_Wrapper, 134
- last_parent
 - HepMC::HEPEVT_Wrapper, 134
- list_of_examples.cc, 307
- LRSUD
 - HerwigWrapper6_4.h, 282
- ludat1
 - PythiaWrapper5_720.h, 317
- ludat1_
 - PythiaWrapper5_720.h, 320
- ludat2
 - PythiaWrapper5_720.h, 318
- ludat2_
 - PythiaWrapper5_720.h, 320
- ludat3
 - PythiaWrapper5_720.h, 318
- ludat3_
 - PythiaWrapper5_720.h, 320
- ludata
 - PythiaWrapper5_720.h, 318
- ludatr
 - PythiaWrapper5_720.h, 318
- ludatr_
 - PythiaWrapper5_720.h, 320
- luhepc
 - PythiaWrapper5_720.h, 318
- lujets
 - PythiaWrapper5_720.h, 318
- lujets_
 - PythiaWrapper5_720.h, 320
- lulist
 - PythiaWrapper5_720.h, 318
- LWEVT
 - HerwigWrapper6_4.h, 282
- LWSUD
 - HerwigWrapper6_4.h, 282
- m
 - HepMC::FourVector, 50
 - HepMC::HEPEVT_Wrapper, 136
- m2
 - HepMC::FourVector, 50
- m_map_iterator
 - HepMC::GenEvent::particle_const_iterator, 74
 - HepMC::GenEvent::particle_iterator, 77
 - HepMC::GenEvent::vertex_const_iterator, 81
 - HepMC::GenEvent::vertex_iterator, 84
- mag
 - HepMC::FourVector, 50
 - HepMC::ThreeVector, 235
- main
 - example_BuildEventFromScratch.cc, 248
 - example_EventSelection.cc, 249
 - example_MyHerwig.cc, 250
 - example_MyPythia.cc, 251
 - example_MyPythiaOnlyToHepMC.cc, 252
 - example_MyPythiaRead.cc, 253
 - example_MyPythiaWithEventSelection.cc, 254
 - example_PythiaParticle.cc, 255
 - example_ReadMyPythia.cc, 256
 - example_UsingIterators.cc, 257
 - testHerwigCopies.cc, 346
 - testPrintBug.cc, 347
 - testPythiaCopies.cc, 348
 - testSimpleVector.cc, 349
- make_antiparticles_from_particles
 - HepMC::ParticleDataTable, 216
- mass
 - HepMC::ParticleData, 210
- max_number_entries
 - HepMC::HEPEVT_Wrapper, 139
- MAXER
 - HerwigWrapper6_4.h, 283
- MAXEV
 - HerwigWrapper6_4.h, 283
- MAXPR
 - HerwigWrapper6_4.h, 283
- mdcy

- PythiaWrapper5_720.h, 320
- PythiaWrapper6_152.h, 328
- PythiaWrapper6_2.h, 336
- mdme
 - PythiaWrapper5_720.h, 320
 - PythiaWrapper6_152.h, 328
 - PythiaWrapper6_2.h, 336
- merge_table
 - HepMC::ParticleDataTable, 216
- mint
 - PythiaWrapper5_720.h, 320
 - PythiaWrapper6_152.h, 328
 - PythiaWrapper6_2.h, 336
- model_independent_pdg_id_
 - HepMC::ParticleData, 212
- MODPDF
 - HerwigWrapper6_4.h, 283
- momentum
 - HepMC::GenParticle, 90
- mpi
 - HepMC::GenEvent, 61
- mrlu
 - PythiaWrapper5_720.h, 320
- mrpy
 - PythiaWrapper6_152.h, 328
 - PythiaWrapper6_2.h, 336
- msel
 - PythiaWrapper5_720.h, 320
 - PythiaWrapper6_152.h, 328
 - PythiaWrapper6_2.h, 336
- mselpd
 - PythiaWrapper6_152.h, 328
 - PythiaWrapper6_2.h, 336
- msti
 - PythiaWrapper5_720.h, 321
 - PythiaWrapper6_152.h, 328
 - PythiaWrapper6_2.h, 337
- mstj
 - PythiaWrapper5_720.h, 321
 - PythiaWrapper6_152.h, 328
 - PythiaWrapper6_2.h, 337
- mstp
 - PythiaWrapper5_720.h, 321
 - PythiaWrapper6_152.h, 328
 - PythiaWrapper6_2.h, 337
- mstu
 - PythiaWrapper5_720.h, 321
 - PythiaWrapper6_152.h, 328
 - PythiaWrapper6_2.h, 337
- msub
 - PythiaWrapper5_720.h, 321
 - PythiaWrapper6_152.h, 329
 - PythiaWrapper6_2.h, 337
- n
 - PythiaWrapper5_720.h, 321
 - PythiaWrapper6_152.h, 329
 - PythiaWrapper6_2.h, 337
- N_Nwounded_collisions
 - HepMC::HeavyIon, 125
- name
 - HepMC::ParticleData, 210
- NBTRY
 - HerwigWrapper6_4.h, 283
- Ncoll
 - HepMC::HeavyIon, 125
- Ncoll_hard
 - HepMC::HeavyIon, 124
- NCOLO
 - HerwigWrapper6_4.h, 283
- NCTRY
 - HerwigWrapper6_4.h, 283
- NDTRY
 - HerwigWrapper6_4.h, 283
- NETRY
 - HerwigWrapper6_4.h, 283
- NFLAV
 - HerwigWrapper6_4.h, 283
- ngen
 - PythiaWrapper5_720.h, 321
 - PythiaWrapper6_152.h, 329
 - PythiaWrapper6_2.h, 337
- ngenpd
 - PythiaWrapper6_152.h, 329
 - PythiaWrapper6_2.h, 337
- NGSPL
 - HerwigWrapper6_4.h, 283
- no_gaps_in_barcodes
 - HepMC::IO_HERWIG, 178
- normal3d
 - HepMC::Polarization, 227
- NOSPAC
 - HerwigWrapper6_4.h, 283
- not_in_vector
 - HepMC, 24
- NOWGT
 - HerwigWrapper6_4.h, 284
- npad
 - PythiaWrapper6_152.h, 329
 - PythiaWrapper6_2.h, 337
- Npart_proj
 - HepMC::HeavyIon, 124
- Npart_targ
 - HepMC::HeavyIon, 125
- NPRFMT
 - HerwigWrapper6_4.h, 284
- NRN
 - HerwigWrapper6_4.h, 284

- NSTRU
 - HerwigWrapper6_4.h, 284
- NSTRY
 - HerwigWrapper6_4.h, 284
- number_children
 - HepMC::HEPEVT_Wrapper, 135
- number_entries
 - HepMC::HEPEVT_Wrapper, 133
- number_parents
 - HepMC::HEPEVT_Wrapper, 134
- NUMER
 - HerwigWrapper6_4.h, 284
- NUMERU
 - HerwigWrapper6_4.h, 284
- NWGTS
 - HerwigWrapper6_4.h, 284
- Nwounded_N_collisions
 - HepMC::HeavyIon, 125
- Nwounded_Nwounded_collisions
 - HepMC::HeavyIon, 126
- NZBIN
 - HerwigWrapper6_4.h, 284
- OMHMX
 - HerwigWrapper6_4.h, 284
- operator *
 - HepMC::GenEvent::particle_const_iterator, 73
 - HepMC::GenEvent::particle_iterator, 76
 - HepMC::GenEvent::vertex_const_iterator, 80
 - HepMC::GenEvent::vertex_iterator, 84
 - HepMC::GenVertex::edge_iterator, 110
 - HepMC::GenVertex::particle_iterator, 114
 - HepMC::GenVertex::vertex_iterator, 118
- operator HepMC::FourVector
 - HepMC::GenParticle, 90
 - HepMC::GenVertex, 101
- operator HepMC::ThreeVector
 - HepMC::GenVertex, 102
- operator particle_const_iterator
 - HepMC::GenEvent::particle_iterator, 76
- operator vertex_const_iterator
 - HepMC::GenEvent::vertex_iterator, 83
- operator!=
 - HepMC::Flow, 42
 - HepMC::FourVector, 51
 - HepMC::GenEvent::particle_const_iterator, 74
 - HepMC::GenEvent::particle_iterator, 77
 - HepMC::GenEvent::vertex_const_iterator, 81
 - HepMC::GenEvent::vertex_iterator, 84
 - HepMC::GenParticle, 89
 - HepMC::GenVertex, 100
 - HepMC::GenVertex::edge_iterator, 111
 - HepMC::GenVertex::particle_iterator, 115
 - HepMC::GenVertex::vertex_iterator, 118
 - HepMC::HeavyIon, 124
 - HepMC::ParticleData, 208
 - HepMC::PdfInfo, 222
 - HepMC::Polarization, 227
 - HepMC::ThreeVector, 236
- operator()
 - IsEventGood, 199
 - IsFinalState, 200
 - IsGoodEvent, 201
 - IsGoodEventMyPythia, 202
 - IsPhoton, 203
 - IsStateFinal, 204
 - IsW_Boson, 205
- operator++
 - HepMC::GenEvent::particle_const_iterator, 73, 74
 - HepMC::GenEvent::particle_iterator, 77
 - HepMC::GenEvent::vertex_const_iterator, 80, 81
 - HepMC::GenEvent::vertex_iterator, 84
 - HepMC::GenVertex::edge_iterator, 110, 111
 - HepMC::GenVertex::particle_iterator, 114, 115
 - HepMC::GenVertex::vertex_iterator, 118
- operator<<
 - HepMC, 25
 - HepMC::Flow, 45
 - HepMC::GenParticle, 94
 - HepMC::GenVertex, 107
 - HepMC::IO_BaseClass, 154
 - HepMC::ParticleData, 212
 - HepMC::Polarization, 228
- operator=
 - HepMC::Flow, 41
 - HepMC::FourVector, 51
 - HepMC::GenEvent, 59
 - HepMC::GenEvent::particle_const_iterator, 73

- HepMC::GenEvent::particle_iterator, 76
- HepMC::GenEvent::vertex_const_iterator, 80
- HepMC::GenEvent::vertex_iterator, 83
- HepMC::GenParticle, 89
- HepMC::GenVertex, 99
- HepMC::GenVertex::edge_iterator, 110
- HepMC::GenVertex::particle_iterator, 114
- HepMC::GenVertex::vertex_iterator, 118
- HepMC::HeavyIon, 124
- HepMC::ParticleDataTable, 215
- HepMC::PdfInfo, 222
- HepMC::Polarization, 227
- HepMC::ThreeVector, 236
- HepMC::WeightContainer, 239, 240
- operator==
 - HepMC::Flow, 42
 - HepMC::FourVector, 51
 - HepMC::GenEvent::particle_const_iterator, 74
 - HepMC::GenEvent::particle_iterator, 77
 - HepMC::GenEvent::vertex_const_iterator, 81
 - HepMC::GenEvent::vertex_iterator, 84
 - HepMC::GenParticle, 89
 - HepMC::GenVertex, 100
 - HepMC::GenVertex::edge_iterator, 111
 - HepMC::GenVertex::particle_iterator, 115
 - HepMC::GenVertex::vertex_iterator, 118
 - HepMC::HeavyIon, 124
 - HepMC::ParticleData, 208
 - HepMC::PdfInfo, 222
 - HepMC::Polarization, 227
 - HepMC::ThreeVector, 236
- operator>>
 - HepMC::IO_BaseClass, 153, 154
- operator[]
 - HepMC::ParticleDataTable, 217
 - HepMC::WeightContainer, 241
- order_begin
 - HepMC::TempParticleMap, 230
- order_end
 - HepMC::TempParticleMap, 230
- orderIterator
 - HepMC::TempParticleMap, 229
- output
 - HepMC::IO_Ascii, 146
 - HepMC::IO_ExtendedAscii, 160
 - HepMC::IO_GenEvent, 168, 169
- p
 - PythiaWrapper5_720.h, 321
 - PythiaWrapper6_152.h, 329
 - PythiaWrapper6_2.h, 337
- parent_event
 - HepMC::GenParticle, 91
 - HepMC::GenVertex, 102
- parents
 - HepMC, 22
- parf
 - PythiaWrapper5_720.h, 322
 - PythiaWrapper6_152.h, 329
 - PythiaWrapper6_2.h, 337
- pari
 - PythiaWrapper5_720.h, 322
 - PythiaWrapper6_152.h, 329
 - PythiaWrapper6_2.h, 337
- parj
 - PythiaWrapper5_720.h, 322
 - PythiaWrapper6_152.h, 329
 - PythiaWrapper6_2.h, 338
- parp
 - PythiaWrapper5_720.h, 322
 - PythiaWrapper6_152.h, 329
 - PythiaWrapper6_2.h, 338
- PART1
 - HerwigWrapper6_4.h, 284
- PART2
 - HerwigWrapper6_4.h, 284
- particle_const_iterator
 - HepMC::GenEvent, 71
 - HepMC::GenEvent::particle_const_iterator, 73
- particle_iterator
 - HepMC::GenEvent, 71
 - HepMC::GenEvent::particle_iterator, 76
 - HepMC::GenVertex, 107
 - HepMC::GenVertex::particle_iterator, 114
- particle_owner
 - HepMC::Flow, 43
- ParticleData
 - HepMC::ParticleData, 208
- ParticleData.cc, 308
- ParticleData.h, 309
- ParticleDataTable
 - HepMC::ParticleDataTable, 215
- ParticleDataTable.h, 310
- particles_begin
 - HepMC::GenEvent, 68, 69
 - HepMC::GenVertex, 106

- particles_empty
 - HepMC::GenEvent, 66
- particles_end
 - HepMC::GenEvent, 69
 - HepMC::GenVertex, 106
- particles_in_const_begin
 - HepMC::GenVertex, 104
- particles_in_const_end
 - HepMC::GenVertex, 104
- particles_in_const_iterator
 - HepMC::GenVertex, 98
- particles_in_size
 - HepMC::GenVertex, 104
- particles_out_const_begin
 - HepMC::GenVertex, 104
- particles_out_const_end
 - HepMC::GenVertex, 104
- particles_out_const_iterator
 - HepMC::GenVertex, 98
- particles_out_size
 - HepMC::GenVertex, 104
- particles_size
 - HepMC::GenEvent, 66
- paru
 - PythiaWrapper5_720.h, 322
 - PythiaWrapper6_152.h, 329
 - PythiaWrapper6_2.h, 338
- PBEAM1
 - HerwigWrapper6_4.h, 285
- PBEAM2
 - HerwigWrapper6_4.h, 285
- pdf1
 - HepMC::PdfInfo, 223
- pdf2
 - HepMC::PdfInfo, 223
- pdf_info
 - HepMC::GenEvent, 63
- PdfInfo
 - HepMC::PdfInfo, 221
- PdfInfo.h, 311
- pdg_id
 - HepMC::GenParticle, 90
 - HepMC::ParticleData, 210
- PDIQK
 - HerwigWrapper6_4.h, 285
- perp
 - HepMC::FourVector, 50
 - HepMC::ThreeVector, 236
- perp2
 - HepMC::FourVector, 50
 - HepMC::ThreeVector, 236
- PGSMX
 - HerwigWrapper6_4.h, 285
- PGSPL
 - HerwigWrapper6_4.h, 285
- PH3MIX
 - HerwigWrapper6_4.h, 285
- phi
 - HepMC::FourVector, 51
 - HepMC::Polarization, 227
 - HepMC::ThreeVector, 235
- PHIMIX
 - HerwigWrapper6_4.h, 285
- PIFAC
 - HerwigWrapper6_4.h, 285
- pmas
 - PythiaWrapper5_720.h, 322
 - PythiaWrapper6_152.h, 329
 - PythiaWrapper6_2.h, 338
- point3d
 - HepMC::GenVertex, 102
- Polarization
 - HepMC::Polarization, 226
- polarization
 - HepMC::GenParticle, 91
- Polarization.cc, 312
- Polarization.h, 313
- pop_back
 - HepMC::WeightContainer, 240
- position
 - HepMC::GenVertex, 102
- print
 - HepMC::Flow, 42
 - HepMC::GenEvent, 59
 - HepMC::GenParticle, 90
 - HepMC::GenVertex, 100
 - HepMC::IO_Ascii, 145
 - HepMC::IO_AsciiParticles, 149
 - HepMC::IO_BaseClass, 153
 - HepMC::IO_ExtendedAscii, 158
 - HepMC::IO_GenEvent, 166
 - HepMC::IO_HEPEVT, 172
 - HepMC::IO_HERWIG, 178
 - HepMC::IO_PDG_ParticleDataTable, 184
 - HepMC::ParticleData, 209
 - HepMC::ParticleDataTable, 216
 - HepMC::Polarization, 227
 - HepMC::WeightContainer, 240
- print_hepevt
 - HepMC::HEPEVT_Wrapper, 132
- print_hepevt_particle
 - HepMC::HEPEVT_Wrapper, 132
- print_inconsistency_errors
 - HepMC::IO_HEPEVT, 172
 - HepMC::IO_HERWIG, 178
- print_legend
 - HepMC::HEPEVT_Wrapper, 141

- print_version
 - HepMC::GenEvent, 60
- PRNDEC
 - HerwigWrapper6_4.h, 285
- PRNDEF
 - HerwigWrapper6_4.h, 285
- PRNTEX
 - HerwigWrapper6_4.h, 285
- PRNWEB
 - HerwigWrapper6_4.h, 285
- production_vertex
 - HepMC::GenParticle, 91
- PRSOF
 - HerwigWrapper6_4.h, 286
- PRVTX
 - HerwigWrapper6_4.h, 286
- pseudoRapidity
 - HepMC::FourVector, 51
- PSPLT
 - HerwigWrapper6_4.h, 286
- PTRMS
 - HerwigWrapper6_4.h, 286
- push_back
 - HepMC::WeightContainer, 240
- px
 - HepMC::FourVector, 49
 - HepMC::HEPEVT_Wrapper, 135
- PXRMS
 - HerwigWrapper6_4.h, 286
- py
 - HepMC::FourVector, 49
 - HepMC::HEPEVT_Wrapper, 135
- pydat1
 - PythiaWrapper6_152.h, 325
 - PythiaWrapper6_2.h, 334
- pydat1_
 - PythiaWrapper6_152.h, 330
 - PythiaWrapper6_2.h, 338
- pydat2
 - PythiaWrapper6_152.h, 326
 - PythiaWrapper6_2.h, 334
- pydat2_
 - PythiaWrapper6_152.h, 330
 - PythiaWrapper6_2.h, 338
- pydat3
 - PythiaWrapper6_152.h, 326
 - PythiaWrapper6_2.h, 334
- pydat3_
 - PythiaWrapper6_152.h, 330
 - PythiaWrapper6_2.h, 338
- pydata
 - PythiaWrapper5_720.h, 318
 - PythiaWrapper6_152.h, 326
 - PythiaWrapper6_2.h, 334
- pydatr
 - PythiaWrapper6_152.h, 326
 - PythiaWrapper6_2.h, 334
- pydatr_
 - PythiaWrapper6_152.h, 330
 - PythiaWrapper6_2.h, 338
- pyevnt
 - PythiaWrapper5_720.h, 318
 - PythiaWrapper6_152.h, 326
 - PythiaWrapper6_2.h, 334
- pyhepc
 - PythiaWrapper6_152.h, 326
 - PythiaWrapper6_2.h, 334
- pyinit
 - PythiaWrapper5_720.h, 318
 - PythiaWrapper6_152.h, 326
 - PythiaWrapper6_2.h, 334
- pyint1
 - PythiaWrapper5_720.h, 318
 - PythiaWrapper6_152.h, 326
 - PythiaWrapper6_2.h, 334
- pyint1_
 - PythiaWrapper5_720.h, 322
 - PythiaWrapper6_152.h, 330
 - PythiaWrapper6_2.h, 338
- pyint2
 - PythiaWrapper5_720.h, 318
 - PythiaWrapper6_152.h, 326
 - PythiaWrapper6_2.h, 334
- pyint2_
 - PythiaWrapper5_720.h, 323
 - PythiaWrapper6_152.h, 330
 - PythiaWrapper6_2.h, 338
- pyint5
 - PythiaWrapper5_720.h, 319
 - PythiaWrapper6_152.h, 326
 - PythiaWrapper6_2.h, 334
- pyint5_
 - PythiaWrapper5_720.h, 323
 - PythiaWrapper6_152.h, 330
 - PythiaWrapper6_2.h, 338
- pyjets
 - PythiaWrapper6_152.h, 326
 - PythiaWrapper6_2.h, 334
- pyjets_
 - PythiaWrapper6_152.h, 330
 - PythiaWrapper6_2.h, 338
- pyjets_maxn
 - PythiaWrapper6_2.h, 338
- pylist
 - PythiaWrapper6_152.h, 326
 - PythiaWrapper6_2.h, 335
- pypars
 - PythiaWrapper5_720.h, 319

- PythiaWrapper6_152.h, 327
- PythiaWrapper6_2.h, 335
- pypars_
 - PythiaWrapper5_720.h, 323
 - PythiaWrapper6_152.h, 330
 - PythiaWrapper6_2.h, 338
- pystat
 - PythiaWrapper5_720.h, 319
 - PythiaWrapper6_152.h, 327
 - PythiaWrapper6_2.h, 335
- pysubs
 - PythiaWrapper5_720.h, 319
 - PythiaWrapper6_152.h, 327
 - PythiaWrapper6_2.h, 335
- pysubs_
 - PythiaWrapper5_720.h, 323
 - PythiaWrapper6_152.h, 330
 - PythiaWrapper6_2.h, 338
- PythiaHelper.h, 314
- PythiaHelper.h
 - initPythia, 314
- PythiaWrapper.h, 315
- PythiaWrapper5_720.h, 316
- PythiaWrapper5_720.h
 - brat, 319
 - ckin, 319
 - coef, 319
 - icol, 319
 - initpydata, 317
 - iset, 319
 - k, 319
 - kchg, 320
 - kfdp, 320
 - kfin, 320
 - kfpr, 320
 - ludat1, 317
 - ludat1_, 320
 - ludat2, 318
 - ludat2_, 320
 - ludat3, 318
 - ludat3_, 320
 - ludata, 318
 - ludatr, 318
 - ludatr_, 320
 - luhepc, 318
 - lujets, 318
 - lujets_, 320
 - lulist, 318
 - mdcy, 320
 - mdme, 320
 - mint, 320
 - mrlu, 320
 - msel, 320
 - msti, 321
 - mstj, 321
 - mstp, 321
 - mstu, 321
 - msub, 321
 - n, 321
 - ngen, 321
 - p, 321
 - parf, 322
 - pari, 322
 - parj, 322
 - parp, 322
 - paru, 322
 - pmas, 322
 - pydata, 318
 - pyevnt, 318
 - pyinit, 318
 - pyint1, 318
 - pyint1_, 322
 - pyint2, 318
 - pyint2_, 323
 - pyint5, 319
 - pyint5_, 323
 - pypars, 319
 - pypars_, 323
 - pystat, 319
 - pysubs, 319
 - pysubs_, 323
 - rllu, 323
 - v, 323
 - vckm, 323
 - vint, 323
 - xsec, 323
- PythiaWrapper6_152.h, 324
- PythiaWrapper6_152.h
 - brat, 327
 - ckin, 327
 - coef, 327
 - icol, 327
 - initpydata, 325
 - iset, 327
 - k, 327
 - kchg, 327
 - kfdp, 327
 - kfin, 328
 - kfpr, 328
 - mdcy, 328
 - mdme, 328
 - mint, 328
 - mrpy, 328
 - msel, 328
 - mselpd, 328
 - msti, 328
 - mstj, 328
 - mstp, 328

- mstu, 328
- msub, 329
- n, 329
- ngen, 329
- ngenpd, 329
- npad, 329
- p, 329
- parf, 329
- pari, 329
- parj, 329
- parp, 329
- paru, 329
- pmas, 329
- pydat1, 325
- pydat1_, 330
- pydat2, 326
- pydat2_, 330
- pydat3, 326
- pydat3_, 330
- pydata, 326
- pydatr, 326
- pydatr_, 330
- pyevnt, 326
- pyhepc, 326
- pyinit, 326
- pyint1, 326
- pyint1_, 330
- pyint2, 326
- pyint2_, 330
- pyint5, 326
- pyint5_, 330
- pyjets, 326
- pyjets_, 330
- pylist, 326
- pypars, 327
- pypars_, 330
- pystat, 327
- pysubs, 327
- pysubs_, 330
- rrpy, 330
- v, 330
- vckm, 330
- vint, 330
- xsec, 330
- PythiaWrapper6_152_WIN32.h, 331
- PythiaWrapper6_2.h, 332
- PythiaWrapper6_2.h
 - brat, 335
 - ckin, 335
 - coef, 335
 - icol, 335
 - initpydata, 334, 335
 - iset, 336
 - k, 336
 - kchg, 336
 - kfdp, 336
 - kfin, 336
 - kfpr, 336
 - mdcy, 336
 - mdme, 336
 - mint, 336
 - mrpy, 336
 - msel, 336
 - mselpd, 336
 - msti, 337
 - mstj, 337
 - mstp, 337
 - mstu, 337
 - msub, 337
 - n, 337
 - ngen, 337
 - ngenpd, 337
 - npad, 337
 - p, 337
 - parf, 337
 - pari, 337
 - parj, 338
 - parp, 338
 - paru, 338
 - pmas, 338
 - pydat1, 334
 - pydat1_, 338
 - pydat2, 334
 - pydat2_, 338
 - pydat3, 334
 - pydat3_, 338
 - pydata, 334
 - pydatr, 334
 - pydatr_, 338
 - pyevnt, 334
 - pyhepc, 334
 - pyinit, 334
 - pyint1, 334
 - pyint1_, 338
 - pyint2, 334
 - pyint2_, 338
 - pyint5, 334
 - pyint5_, 338
 - pyjets, 334
 - pyjets_, 338
 - pyjets_maxn, 338
 - pylist, 335
 - pypars, 335
 - pypars_, 338
 - pystat, 335
 - pysubs, 335
 - pysubs_, 338
 - rrpy, 338

- upegvt, 335
- upinit, 335
- v, 338
- vckm, 339
- vint, 339
- xsec, 339
- PythiaWrapper6_2_WIN32.h, 340
- pz
 - HepMC::FourVector, 49
 - HepMC::HEPEVT_Wrapper, 135
- QCDL3
 - HerwigWrapper6_4.h, 286
- QCDL5
 - HerwigWrapper6_4.h, 286
- QCDLAM
 - HerwigWrapper6_4.h, 286
- QDIQK
 - HerwigWrapper6_4.h, 286
- QFCH
 - HerwigWrapper6_4.h, 286
- QG
 - HerwigWrapper6_4.h, 286
- QSPAC
 - HerwigWrapper6_4.h, 286
- QV
 - HerwigWrapper6_4.h, 287
- r
 - HepMC::ThreeVector, 235
- random_states
 - HepMC::GenEvent, 63
- range
 - HepMC::GenVertex::vertex_iterator, 119
- rdstate
 - HepMC::IO_Ascii, 144
 - HepMC::IO_AsciiParticles, 149
 - HepMC::IO_ExtendedAscii, 158
 - HepMC::IO_GenEvent, 166
 - HepMC::IO_PDG_ParticleDataTable, 184
- read_entry
 - HepMC::IO_PDG_ParticleDataTable, 185
- read_heavy_ion
 - HepMC::CommonIO, 33
 - HepMC::IO_GenEvent, 168
- read_io_ascii
 - HepMC::CommonIO, 32
- read_io_extendedascii
 - HepMC::CommonIO, 33
- read_io_genevent
 - HepMC::CommonIO, 33
- read_io_particle_data_table
 - HepMC::CommonIO, 33
- read_next_event
 - HepMC::IO_BaseClass, 153
- read_particle
 - HepMC::CommonIO, 34
 - HepMC::IO_GenEvent, 168
- read_particle_data
 - HepMC::CommonIO, 34
 - HepMC::IO_GenEvent, 168
- read_particle_data_table
 - HepMC::IO_BaseClass, 153
- read_pdf_info
 - HepMC::CommonIO, 34
 - HepMC::IO_GenEvent, 168
- read_vertex
 - HepMC::CommonIO, 34
 - HepMC::IO_GenEvent, 167
- relatives
 - HepMC, 22
- remove_barcode
 - HepMC::GenEvent, 69, 70
- remove_gaps_in_hepevt
 - HepMC::IO_HERWIG, 181
- remove_particle
 - HepMC::GenVertex, 101
- remove_particle_in
 - HepMC::GenVertex, 107
- remove_particle_out
 - HepMC::GenVertex, 107
- remove_vertex
 - HepMC::GenEvent, 67
- repair_hepevt
 - HepMC::IO_HERWIG, 180
- rho
 - HepMC::FourVector, 51
- rrlu
 - PythiaWrapper5_720.h, 323
- rrpy
 - PythiaWrapper6_152.h, 330
 - PythiaWrapper6_2.h, 338
- SCABI
 - HerwigWrapper6_4.h, 287
- scalePDF
 - HepMC::PdfInfo, 223
- search_for_key_end
 - HepMC::IO_PDG_ParticleDataTable, 184
- SearchVector.cc, 341
- SearchVector.h, 342
- set
 - HepMC::FourVector, 51
 - HepMC::ThreeVector, 235

- set_alphaQCD
 - HepMC::GenEvent, 64
- set_alphaQED
 - HepMC::GenEvent, 65
- set_barcode
 - HepMC::GenEvent, 69
- set_barcode_
 - HepMC::GenParticle, 94
 - HepMC::GenVertex, 105
- set_beam_particles
 - HepMC::GenEvent, 65
- set_charge
 - HepMC::ParticleData, 211
- set_children
 - HepMC::HEPEVT_Wrapper, 138
- set_clifetime
 - HepMC::ParticleData, 211
- set_description
 - HepMC::ParticleDataTable, 219
- set_eccentricity
 - HepMC::HeavyIon, 128
- set_end_vertex_
 - HepMC::GenParticle, 94
- set_event_number
 - HepMC::GenEvent, 64
 - HepMC::HEPEVT_Wrapper, 137
- set_event_plane_angle
 - HepMC::HeavyIon, 128
- set_event_scale
 - HepMC::GenEvent, 64
- set_flow
 - HepMC::GenParticle, 93
- set_generated_mass
 - HepMC::GenParticle, 93
- set_heavy_ion
 - HepMC::GenEvent, 66
- set_icode
 - HepMC::Flow, 43
- set_id
 - HepMC::GenVertex, 103
 - HepMC::HEPEVT_Wrapper, 137
- set_id1
 - HepMC::PdfInfo, 223
- set_id2
 - HepMC::PdfInfo, 223
- set_impact_parameter
 - HepMC::HeavyIon, 127
- set_mass
 - HepMC::HEPEVT_Wrapper, 138
 - HepMC::ParticleData, 211
- set_max_number_entries
 - HepMC::HEPEVT_Wrapper, 140
- set_momentum
 - HepMC::GenParticle, 92
- HepMC::HEPEVT_Wrapper, 138
 - set_mpi
 - HepMC::GenEvent, 64
 - set_N_Nwounded_collisions
 - HepMC::HeavyIon, 127
 - set_Ncoll
 - HepMC::HeavyIon, 127
 - set_Ncoll_hard
 - HepMC::HeavyIon, 126
 - set_no_gaps_in_barcodes
 - HepMC::IO_HERWIG, 179
 - set_normal3d
 - HepMC::Polarization, 228
 - set_Npart_proj
 - HepMC::HeavyIon, 126
 - set_Npart_targ
 - HepMC::HeavyIon, 127
 - set_number_entries
 - HepMC::HEPEVT_Wrapper, 137
 - set_Nwounded_N_collisions
 - HepMC::HeavyIon, 127
 - set_Nwounded_Nwounded_collisions
 - HepMC::HeavyIon, 127
 - set_parent_event_
 - HepMC::GenVertex, 105
 - set_parents
 - HepMC::HEPEVT_Wrapper, 138
 - set_pdf1
 - HepMC::PdfInfo, 224
 - set_pdf2
 - HepMC::PdfInfo, 224
 - set_pdf_info
 - HepMC::GenEvent, 66
 - set_pdg_id
 - HepMC::GenParticle, 92
 - set_phi
 - HepMC::Polarization, 228
 - set_polarization
 - HepMC::GenParticle, 93
 - set_position
 - HepMC::GenVertex, 102
 - HepMC::HEPEVT_Wrapper, 139
 - set_print_inconsistency_errors
 - HepMC::IO_HEPEVT, 173
 - HepMC::IO_HERWIG, 178
 - set_production_vertex_
 - HepMC::GenParticle, 93
 - set_random_states
 - HepMC::GenEvent, 66
 - set_scalePDF
 - HepMC::PdfInfo, 224
 - set_sigma_inel_NN
 - HepMC::HeavyIon, 128
 - set_signal_process_id

HepMC::GenEvent, 64
 set_signal_process_vertex
 HepMC::GenEvent, 65
 set_sizeof_int
 HepMC::HEPEVT_Wrapper, 139
 set_sizeof_real
 HepMC::HEPEVT_Wrapper, 140
 set_spectator_neutrons
 HepMC::HeavyIon, 127
 set_spectator_protons
 HepMC::HeavyIon, 127
 set_spin
 HepMC::ParticleData, 212
 set_status
 HepMC::GenParticle, 92
 HepMC::HEPEVT_Wrapper, 137
 set_theta
 HepMC::Polarization, 228
 set_theta_phi
 HepMC::Polarization, 228
 set_trust_beam_particles
 HepMC::IO_HEPEVT, 173
 set_trust_both_mothers_and_daughters
 HepMC::IO_HEPEVT, 173
 HepMC::IO_HERWIG, 179
 set_trust_mothers_before_daughters
 HepMC::IO_HEPEVT, 173
 HepMC::IO_HERWIG, 179
 set_unique_icode
 HepMC::Flow, 43
 set_width
 HepMC::ParticleData, 211
 set_x1
 HepMC::PdfInfo, 224
 set_x2
 HepMC::PdfInfo, 224
 setE
 HepMC::FourVector, 53
 setGeneratedMass
 HepMC::GenParticle, 93
 setPhi
 HepMC::ThreeVector, 235
 setPrecision
 HepMC::IO_AsciiParticles, 149
 setPx
 HepMC::FourVector, 52
 setPy
 HepMC::FourVector, 52
 setPz
 HepMC::FourVector, 52
 setT
 HepMC::FourVector, 52
 setTheta
 HepMC::ThreeVector, 235

setX
 HepMC::FourVector, 52
 HepMC::ThreeVector, 234
 setY
 HepMC::FourVector, 52
 HepMC::ThreeVector, 234
 setZ
 HepMC::FourVector, 52
 HepMC::ThreeVector, 235
 sigma_inel_NN
 HepMC::HeavyIon, 126
 signal_process_id
 HepMC::GenEvent, 61
 signal_process_vertex
 HepMC::GenEvent, 62
 SimpleVector.h, 343
 size
 HepMC::Flow, 43
 HepMC::ParticleDataTable, 217
 HepMC::WeightContainer, 240
 sizeof_int
 HepMC::HEPEVT_Wrapper, 139
 sizeof_real
 HepMC::HEPEVT_Wrapper, 139
 SOFTME
 HerwigWrapper6_4.h, 287
 spectator_neutrons
 HepMC::HeavyIon, 125
 spectator_protons
 HepMC::HeavyIon, 125
 spin
 HepMC::ParticleData, 211
 status
 HepMC::GenParticle, 90
 HepMC::HEPEVT_Wrapper, 133
 suggest_barcode
 HepMC::GenParticle, 92
 HepMC::GenVertex, 103
 swap
 HepMC::Flow, 41
 HepMC::FourVector, 49
 HepMC::GenEvent, 59
 HepMC::GenParticle, 89
 HepMC::GenVertex, 99
 HepMC::HeavyIon, 124
 HepMC::PdfInfo, 222
 HepMC::Polarization, 226
 HepMC::ThreeVector, 234
 HepMC::WeightContainer, 239
 SWEIN
 HerwigWrapper6_4.h, 287
 t
 HepMC::FourVector, 50

- HepMC::HEPEVT_Wrapper, 137
- TempMap
 - HepMC::TempParticleMap, 229
- TempMapIterator
 - HepMC::TempParticleMap, 229
- TempOrderMap
 - HepMC::TempParticleMap, 229
- TempParticleMap
 - HepMC::TempParticleMap, 230
- TempParticleMap.h, 344
- testHepMCIteration.h, 345
- testHepMCIteration.h
 - IsPhoton, 345
 - IsWBoson, 345
- testHerwigCopies.cc, 346
- testHerwigCopies.cc
 - main, 346
- testPrintBug.cc, 347
- testPrintBug.cc
 - main, 347
- testPythiaCopies.cc, 348
- testPythiaCopies.cc
 - main, 348
- testSimpleVector.cc, 349
- testSimpleVector.cc
 - main, 349
- theta
 - HepMC::FourVector, 51
 - HepMC::Polarization, 227
 - HepMC::ThreeVector, 235
- ThreeVector
 - HepMC::ThreeVector, 233, 234
- TLOUT
 - HerwigWrapper6_4.h, 287
- TMTOP
 - HerwigWrapper6_4.h, 287
- translate_herwig_to_pdg_id
 - HepMC::IO_HERWIG, 182
- trust_beam_particles
 - HepMC::IO_HEPEVT, 172
- trust_both_mothers_and_daughters
 - HepMC::IO_HEPEVT, 172
 - HepMC::IO_HERWIG, 179
- trust_mothers_before_daughters
 - HepMC::IO_HEPEVT, 172
 - HepMC::IO_HERWIG, 179
- type
 - HepMC::detail::disable_if< false, T >, 36
 - HepMC::detail::enable_if< true, T >, 38
- upevnt
 - PythiaWrapper6_2.h, 335
- upinit
 - PythiaWrapper6_2.h, 335
- v
 - PythiaWrapper5_720.h, 323
 - PythiaWrapper6_152.h, 330
 - PythiaWrapper6_2.h, 338
- valid_beam_particles
 - HepMC::GenEvent, 62
- value
 - HepMC::detail::is_arithmetic, 186
 - HepMC::detail::is_arithmetic< char >, 187
 - HepMC::detail::is_arithmetic< double >, 188
 - HepMC::detail::is_arithmetic< float >, 189
 - HepMC::detail::is_arithmetic< int >, 190
 - HepMC::detail::is_arithmetic< long >, 191
 - HepMC::detail::is_arithmetic< long double >, 192
 - HepMC::detail::is_arithmetic< short >, 193
 - HepMC::detail::is_arithmetic< signed char >, 194
 - HepMC::detail::is_arithmetic< unsigned char >, 195
 - HepMC::detail::is_arithmetic< unsigned int >, 196
 - HepMC::detail::is_arithmetic< unsigned long >, 197
 - HepMC::detail::is_arithmetic< unsigned short >, 198
- VCKM
 - HerwigWrapper6_4.h, 287
- vckm
 - PythiaWrapper5_720.h, 323
 - PythiaWrapper6_152.h, 330
 - PythiaWrapper6_2.h, 339
- VectorConversion.h, 350
- VectorConversion.h
 - convertTo, 350
- version
 - HepMC, 26
- Version.h, 351
- versionName
 - HepMC, 26
- vertex_const_iterator
 - HepMC::GenEvent, 70
 - HepMC::GenEvent::vertex_const_iterator, 80
- vertex_iterator

- HepMC::GenEvent, 70
- HepMC::GenEvent::vertex_iterator, 83
- HepMC::GenVertex, 107
- HepMC::GenVertex::vertex_iterator, 117
- vertex_root
 - HepMC::GenVertex::edge_iterator, 111
 - HepMC::GenVertex::vertex_iterator, 119
- vertices_begin
 - HepMC::GenEvent, 67, 68
 - HepMC::GenVertex, 106
- vertices_empty
 - HepMC::GenEvent, 67
- vertices_end
 - HepMC::GenEvent, 68
 - HepMC::GenVertex, 106
- vertices_size
 - HepMC::GenEvent, 66
- VFCH
 - HerwigWrapper6_4.h, 287
- VGCUT
 - HerwigWrapper6_4.h, 287
- vint
 - PythiaWrapper5_720.h, 323
 - PythiaWrapper6_152.h, 330
 - PythiaWrapper6_2.h, 339
- VPCUT
 - HerwigWrapper6_4.h, 287
- VQCUT
 - HerwigWrapper6_4.h, 287
- WBGST
 - HerwigWrapper6_4.h, 287
- WeightContainer
 - HepMC::WeightContainer, 239
- WeightContainer.h, 352
- weights
 - HepMC::GenEvent, 62, 63
 - HepMC::GenVertex, 103
- WGTMAX
 - HerwigWrapper6_4.h, 288
- WGTSUM
 - HerwigWrapper6_4.h, 288
- width
 - HepMC::ParticleData, 211
- write_beam_particles
 - HepMC::IO_ExtendedAscii, 158
 - HepMC::IO_GenEvent, 166
- write_byte_num
 - HepMC::HEPEVT_Wrapper, 141
- write_comment
 - HepMC::IO_Ascii, 144
 - HepMC::IO_AsciiParticles, 149
 - HepMC::IO_ExtendedAscii, 158
 - HepMC::IO_GenEvent, 166
- write_end_listing
 - HepMC::IO_Ascii, 145
 - HepMC::IO_AsciiParticles, 149
 - HepMC::IO_ExtendedAscii, 160
 - HepMC::IO_GenEvent, 168
- write_event
 - HepMC::IO_Ascii, 143
 - HepMC::IO_AsciiParticles, 148
 - HepMC::IO_BaseClass, 152
 - HepMC::IO_ExtendedAscii, 157
 - HepMC::IO_GenEvent, 165
 - HepMC::IO_HEPEVT, 172
- write_heavy_ion
 - HepMC::IO_ExtendedAscii, 159
 - HepMC::IO_GenEvent, 167
- write_IO_Ascii_End
 - HepMC::CommonIO, 31
- write_IO_Ascii_Key
 - HepMC::CommonIO, 31
- write_IO_Ascii_PDT_End
 - HepMC::CommonIO, 32
- write_IO_Ascii_PDT_Key
 - HepMC::CommonIO, 31
- write_IO_ExtendedAscii_End
 - HepMC::CommonIO, 31
- write_IO_ExtendedAscii_Key
 - HepMC::CommonIO, 31
- write_IO_ExtendedAscii_PDT_End
 - HepMC::CommonIO, 32
- write_IO_ExtendedAscii_PDT_Key
 - HepMC::CommonIO, 32
- write_IO_GenEvent_End
 - HepMC::CommonIO, 31
- write_IO_GenEvent_Key
 - HepMC::CommonIO, 31
- write_particle
 - HepMC::IO_Ascii, 145
 - HepMC::IO_ExtendedAscii, 159
 - HepMC::IO_GenEvent, 167
- write_particle_data
 - HepMC::IO_Ascii, 145
 - HepMC::IO_ExtendedAscii, 159
 - HepMC::IO_GenEvent, 167
- write_particle_data_table
 - HepMC::IO_Ascii, 144
 - HepMC::IO_AsciiParticles, 148
 - HepMC::IO_BaseClass, 152
 - HepMC::IO_ExtendedAscii, 157
 - HepMC::IO_GenEvent, 165
- write_pdf_info
 - HepMC::IO_ExtendedAscii, 159
 - HepMC::IO_GenEvent, 167

write_vertex
 HepMC::IO_Ascii, 145
 HepMC::IO_ExtendedAscii, 158
 HepMC::IO_GenEvent, 166
writeVersion
 HepMC, 26
WSQSUM
 HerwigWrapper6_4.h, 288

x
 HepMC::FourVector, 49
 HepMC::HEPEVT_Wrapper, 136
 HepMC::ThreeVector, 234
x1
 HepMC::PdfInfo, 223
x2
 HepMC::PdfInfo, 223
xsec
 PythiaWrapper5_720.h, 323
 PythiaWrapper6_152.h, 330
 PythiaWrapper6_2.h, 339

y
 HepMC::FourVector, 50
 HepMC::HEPEVT_Wrapper, 136
 HepMC::ThreeVector, 234

z
 HepMC::FourVector, 50
 HepMC::HEPEVT_Wrapper, 136
 HepMC::ThreeVector, 234
ZBINM
 HerwigWrapper6_4.h, 288
zero_everything
 HepMC::HEPEVT_Wrapper, 132
zero_hepevt_entry
 HepMC::IO_HERWIG, 181
ZPRIME
 HerwigWrapper6_4.h, 288