

HepPDT Reference Manual
2.04.02

Generated by Doxygen 1.5.1-3

Thu Jun 21 15:55:54 2007

Contents

1	HepPDT Directory Hierarchy	1
1.1	HepPDT Directories	1
2	HepPDT Namespace Index	3
2.1	HepPDT Namespace List	3
3	HepPDT Class Index	5
3.1	HepPDT Class List	5
4	HepPDT File Index	7
4.1	HepPDT File List	7
5	HepPDT Directory Documentation	11
5.1	/home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/examples/ Directory Reference	11
5.2	/home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/tests/HepPDT/ Directory Reference	12
5.3	/home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/examples/HepPDT/ Directory Reference	13
5.4	/home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/include/HepPDT/ Directory Reference	14
5.5	/home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/src/HepPDT/ Directory Reference	15
5.6	/home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/include/HepPID/ Directory Reference	16
5.7	/home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/src/HepPID/ Directory Reference	17
5.8	/home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/tests/HepPID/ Directory Reference	18
5.9	/home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/examples/HepPID/ Directory Reference	19
5.10	/home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/include/ Directory Reference .	20
5.11	/home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/src/ Directory Reference . . .	21

5.12	/home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/tests/ Directory Reference . . .	22
6	HepPDT Namespace Documentation	23
6.1	HepPDT Namespace Reference	23
6.2	HepPDT::detail Namespace Reference	36
6.3	HepPID Namespace Reference	40
6.4	std Namespace Reference	64
7	HepPDT Class Documentation	65
7.1	HepPDT::CommonParticleData Class Reference	65
7.2	HepPDT::Constituent Class Reference	76
7.3	HepPDT::DecayChannel Class Reference	79
7.4	HepPDT::DecayData Class Reference	82
7.5	HepPDT::DefTable Class Reference	85
7.6	HepPDT::Measurement Class Reference	88
7.7	HepPDT::ParticleData Class Reference	91
7.8	HepPDT::ParticleDataTable Class Reference	100
7.9	HepPDT::ParticleDataTableComparison Class Reference	108
7.10	HepPDT::ParticleID Class Reference	109
7.11	HepPDT::QQAngularHelicity Struct Reference	118
7.12	HepPDT::QQChannel Class Reference	120
7.13	HepPDT::QQDecay Class Reference	127
7.14	HepPDT::QQhelicity Struct Reference	130
7.15	HepPDT::QQmatrix Struct Reference	131
7.16	HepPDT::Quarks Struct Reference	132
7.17	HepPDT::ResonanceStructure Class Reference	134
7.18	HepPDT::SpinState Class Reference	138
7.19	HepPDT::TableBuilder Class Reference	141
7.20	HepPDT::TempAliasData Struct Reference	146
7.21	HepPDT::TempConstituent Struct Reference	148
7.22	HepPDT::TempDecayData Struct Reference	149
7.23	HepPDT::TempParticleData Struct Reference	151
8	HepPDT File Documentation	157
8.1	addEvtGenParticles.cc File Reference	157
8.2	addHerwigParticles.cc File Reference	159
8.3	addIsajetParticles.cc File Reference	160
8.4	addParticleTable.cc File Reference	161

8.5	addPDGParticles.cc File Reference	162
8.6	addPythiaParticles.cc File Reference	163
8.7	addQQParticles.cc File Reference	164
8.8	calculateWidthFromLifetime.cc File Reference	165
8.9	CommonParticleData.hh File Reference	166
8.10	CommonParticleData.icc File Reference	167
8.11	Constituent.cc File Reference	168
8.12	Constituent.hh File Reference	169
8.13	convertTemporaryMap.cc File Reference	170
8.14	DecayChannel.cc File Reference	171
8.15	DecayChannel.hh File Reference	172
8.16	DecayChannel.icc File Reference	173
8.17	DecayData.cc File Reference	174
8.18	DecayData.hh File Reference	175
8.19	DecayData.icc File Reference	176
8.20	DefTable.cc File Reference	177
8.21	DefTable.hh File Reference	178
8.22	examListHerwig.cc File Reference	179
8.23	examListIsajet.cc File Reference	181
8.24	examListPythia.cc File Reference	183
8.25	examMyPDT.cc File Reference	185
8.26	getIsajetID.cc File Reference	187
8.27	getPDGpid.cc File Reference	188
8.28	getPythiaid.cc File Reference	189
8.29	hasMethods.cc File Reference	190
8.30	lifetime.cc File Reference	191
8.31	list_of_examples.cc File Reference	192
8.32	list_of_tests.cc File Reference	193
8.33	listEvtGenNames.cc.in File Reference	194
8.34	listEvtGenTranslation.cc File Reference	195
8.35	listHerwigTranslation.cc File Reference	196
8.36	listIsajetTranslation.cc File Reference	197
8.37	listParticleNames.cc File Reference	198
8.38	listPDGNames.cc.in File Reference	199
8.39	listPDGTranslation.cc File Reference	200
8.40	listPythiaNames.cc.in File Reference	201

8.41 listPythiaTranslation.cc File Reference	202
8.42 listQQTranslation.cc File Reference	203
8.43 Measurement.hh File Reference	204
8.44 Measurement.icc File Reference	205
8.45 ParticleData.hh File Reference	206
8.46 ParticleData.icc File Reference	207
8.47 ParticleDataTable.cc File Reference	208
8.48 ParticleDataTable.hh File Reference	209
8.49 ParticleDataTable.icc File Reference	210
8.50 ParticleDataTableComparison.hh File Reference	211
8.51 ParticleID.cc File Reference	212
8.52 ParticleID.hh File Reference	213
8.53 ParticleID.icc File Reference	214
8.54 ParticleIDMethods.cc File Reference	215
8.55 ParticleIDMethods.hh File Reference	217
8.56 ParticleIDTranslations.hh File Reference	219
8.57 ParticleName.cc File Reference	221
8.58 ParticleName.hh File Reference	222
8.59 QQChannel.cc File Reference	223
8.60 QQChannel.hh File Reference	224
8.61 QQDecay.cc File Reference	225
8.62 QQDecay.hh File Reference	226
8.63 quarks.cc File Reference	227
8.64 ResonanceStructure.cc File Reference	228
8.65 ResonanceStructure.hh File Reference	229
8.66 spindtoi.cc File Reference	230
8.67 spinitod.cc File Reference	231
8.68 SpinState.hh File Reference	232
8.69 SpinState.icc File Reference	233
8.70 stringtodouble.cc File Reference	234
8.71 stringtodouble.hh File Reference	235
8.72 TableBuilder.hh File Reference	236
8.73 TableBuilder.icc File Reference	238
8.74 TempParticleData.cc File Reference	239
8.75 TempParticleData.hh File Reference	240
8.76 testHepPDT.cc File Reference	241

8.77	testParticleIDMethods.cc File Reference	242
8.78	testPID.cc File Reference	243
8.79	testReadEvtGen.cc.in File Reference	244
8.80	testReadIsajet.cc.in File Reference	245
8.81	testReadParticleTable.cc.in File Reference	246
8.82	testReadQQ.cc.in File Reference	247
8.83	translateEvtGen.cc File Reference	248
8.84	translateGeanttoPDT.cc File Reference	249
8.85	translateHerwig.cc File Reference	250
8.86	translateIsajet.cc File Reference	251
8.87	translatePDG.cc File Reference	252
8.88	translatePDTtoGeant.cc File Reference	253
8.89	translatePythia.cc File Reference	254
8.90	translateQQ.cc File Reference	255
8.91	Version.cc File Reference	257
8.92	Version.cc File Reference	258
8.93	Version.hh File Reference	259
8.94	Version.hh File Reference	260
8.95	write.cc File Reference	261
9	HepPDT Example Documentation	263
9.1	examListHerwig.cc	263
9.2	examListHerwigInterface.F	265
9.3	examListIsajet.cc	267
9.4	examListIsajetInterface.F	269
9.5	examListPythia.cc	271
9.6	examListPythiaInterface.F	273
9.7	examMyPDT.cc	274
9.8	listEvtGenNames.cc.in	276
9.9	listEvtGenTranslation.cc	277
9.10	listHerwigTranslation.cc	278
9.11	listIsajetTranslation.cc	279
9.12	listParticleNames.cc	280
9.13	listPDGNames.cc.in	281
9.14	listPDGTranslation.cc	282
9.15	listPythiaNames.cc.in	283
9.16	listPythiaTranslation.cc	284

9.17 listQQTranslation.cc	285
9.18 testHepPDT.cc	286
9.19 testPID.cc	288
9.20 testReadEvtGen.cc.in	290
9.21 testReadIsajet.cc.in	291
9.22 testReadQQ.cc.in	292

Chapter 1

HepPDT Directory Hierarchy

1.1 HepPDT Directories

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

examples	11
HepPDT	13
HepPID	19
include	20
HepPDT	14
HepPID	16
src	21
HepPDT	15
HepPID	17
tests	22
HepPDT	12
HepPID	18

Chapter 2

HepPDT Namespace Index

2.1 HepPDT Namespace List

Here is a list of all namespaces with brief descriptions:

HepPDT (HepPDT (p. 23) is a Particle Data Table namespace)	23
HepPDT::detail (HepPDT::detail (p. 36) is for internal use)	36
HepPID (The HepPID (p. 40) namespace has independent particle ID translation methods)	40
std	64

Chapter 3

HepPDT Class Index

3.1 HepPDT Class List

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

HepPDT::CommonParticleData (The CommonParticleData (p.65) class holds basic particle data)	65
HepPDT::Constituent (The Constituent (p.76) class has information about constituent particles)	76
HepPDT::DecayChannel (The DecayChannel (p.79) class holds particle decay information)	79
HepPDT::DecayData (The DecayData (p.82) class holds particle decay information)	82
HepPDT::DefTable (The DefTable (p.85) class holds EvtGen definitions)	85
HepPDT::Measurement (The Measurement (p.88) class defines a value with its error)	88
HepPDT::ParticleData (The ParticleData (p.91) class holds data for a single particle in the table)	91
HepPDT::ParticleDataTable (The ParticleDataTable (p.100) class is the core of HepPDT (p.23))	100
HepPDT::ParticleDataTableComparison (The ParticleDataTableComparison (p.108) class provides a utility for sorting the PDT)	108
HepPDT::ParticleID (The ParticleID (p.109) has various utilities to extract information from the particle ID)	109
HepPDT::QQAngularHelicity (QQ decay angular helicity information)	118
HepPDT::QQChannel (QQChannel (p.120) is a temporary holder for QQ decay information)	120
HepPDT::QQDecay (QQDecay (p.127) is a temporary holder for QQ decay information)	127
HepPDT::QQHelicity (QQ decay helicity information)	130
HepPDT::QQmatrix (QQ decay matrix information)	131
HepPDT::Quarks (Constituent (p.76) quarks)	132
HepPDT::ResonanceStructure (The ResonanceStructure (p.134) class is holds mass and width information)	134
HepPDT::SpinState (The SpinState (p.138) class holds spin information)	138
HepPDT::TableBuilder (The TableBuilder (p.141) class is used to construct a ParticleDataTable (p.100))	141
HepPDT::TempAliasData (Hold Alias information from EvtGen)	146
HepPDT::TempConstituent (Constituent (p.76) (e.g., quark) information)	148

HepPDT::TempDecayData (Temporary holder for decay data)	149
HepPDT::TempParticleData (Temporary holder for Particle Data information) . .	151

Chapter 4

HepPDT File Index

4.1 HepPDT File List

Here is a list of all files with brief descriptions:

<code>addEvtGenParticles.cc</code>	157
<code>addHerwigParticles.cc</code>	159
<code>addIsajetParticles.cc</code>	160
<code>addParticleTable.cc</code>	161
<code>addPDGParticles.cc</code>	162
<code>addPythiaParticles.cc</code>	163
<code>addQQParticles.cc</code>	164
<code>calculateWidthFromLifetime.cc</code>	165
<code>CommonParticleData.hh</code>	166
<code>CommonParticleData.icc</code>	167
<code>Constituent.cc</code>	168
<code>Constituent.hh</code>	169
<code>convertTemporaryMap.cc</code>	170
<code>DecayChannel.cc</code>	171
<code>DecayChannel.hh</code>	172
<code>DecayChannel.icc</code>	173
<code>DecayData.cc</code>	174
<code>DecayData.hh</code>	175
<code>DecayData.icc</code>	176
<code>DefTable.cc</code>	177
<code>DefTable.hh</code>	178
<code>examListHerwig.cc</code>	179
<code>examListIsajet.cc</code>	181
<code>examListPythia.cc</code>	183
<code>examMyPDT.cc</code>	185
<code>getIsajetID.cc</code>	187
<code>getPDGpid.cc</code>	188
<code>getPythiaid.cc</code>	189
<code>hasMethods.cc</code>	190
<code>lifetime.cc</code>	191
<code>list_of_examples.cc</code>	192
<code>list_of_tests.cc</code>	193
<code>listEvtGenNames.cc.in</code>	194

listEvtGenTranslation.cc	195
listHerwigTranslation.cc	196
listIsajetTranslation.cc	197
listParticleNames.cc	198
listPDGNames.cc.in	199
listPDGTranslation.cc	200
listPythiaNames.cc.in	201
listPythiaTranslation.cc	202
listQQTranslation.cc	203
Measurement.hh	204
Measurement.icc	205
ParticleData.hh	206
ParticleData.icc	207
ParticleDataTable.cc	208
ParticleDataTable.hh	209
ParticleDataTable.icc	210
ParticleDataTableComparison.hh	211
ParticleID.cc	212
ParticleID.hh	213
ParticleID.icc	214
ParticleIDMethods.cc	215
ParticleIDMethods.hh	217
ParticleIDTranslations.hh	219
ParticleName.cc	221
ParticleName.hh	222
QQChannel.cc	223
QQChannel.hh	224
QQDecay.cc	225
QQDecay.hh	226
quarks.cc	227
ResonanceStructure.cc	228
ResonanceStructure.hh	229
spindtoi.cc	230
spinitod.cc	231
SpinState.hh	232
SpinState.icc	233
stringtodouble.cc	234
stringtodouble.hh	235
TableBuilder.hh	236
TableBuilder.icc	238
TempParticleData.cc	239
TempParticleData.hh	240
testHepPDT.cc	241
testParticleIDMethods.cc	242
testPID.cc	243
testReadEvtGen.cc.in	244
testReadIsajet.cc.in	245
testReadParticleTable.cc.in	246
testReadQQ.cc.in	247
translateEvtGen.cc	248
translateGeanttoPDT.cc	249
translateHerwig.cc	250
translateIsajet.cc	251
translatePDG.cc	252

<code>translatePDTtoGeant.cc</code>	253
<code>translatePythia.cc</code>	254
<code>translateQQ.cc</code>	255
<code>HepPDT/Version.cc</code>	257
<code>HepPID/Version.cc</code>	258
<code>HepPDT/Version.hh</code>	259
<code>HepPID/Version.hh</code>	260
<code>write.cc</code>	261

Chapter 5

HepPDT Directory Documentation

5.1 `/home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/examples/` Directory Reference

Directories

- directory `HepPDT`
- directory `HepPID`

Files

- file `list_of_examples.cc`

5.2 /home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/tests/HepPDT/ Directory Reference

Files

- file `listEvtGenNames.cc.in`
- file `listPDGNames.cc.in`
- file `listPythiaNames.cc.in`
- file `testHepPDT.cc`
- file `testPID.cc`
- file `testReadEvtGen.cc.in`
- file `testReadIsajet.cc.in`
- file `testReadParticleTable.cc.in`
- file `testReadQQ.cc.in`

5.3 /home/cepa01/garren/lcg/heppdt/HepPDT- 2.04.02/examples/HepPDT/ Directory Reference

Files

- file examMyPDT.cc

5.4 /home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/include/HepPDT/ Directory Reference

Files

- file **CommonParticleData.hh**
- file **CommonParticleData.icc**
- file **Constituent.hh**
- file **DecayChannel.hh**
- file **DecayChannel.icc**
- file **DecayData.hh**
- file **DecayData.icc**
- file **DefTable.hh**
- file **Measurement.hh**
- file **Measurement.icc**
- file **ParticleData.hh**
- file **ParticleData.icc**
- file **ParticleDataTable.hh**
- file **ParticleDataTable.icc**
- file **ParticleDataTableComparison.hh**
- file **ParticleID.hh**
- file **ParticleID.icc**
- file **QQChannel.hh**
- file **QQDecay.hh**
- file **ResonanceStructure.hh**
- file **SpinState.hh**
- file **SpinState.icc**
- file **stringtodouble.hh**
- file **TableBuilder.hh**
- file **TableBuilder.icc**
- file **TempParticleData.hh**
- file **HepPDT/Version.hh**

5.5 /home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/src/HepPDT/ Directory Reference

Files

- file `addEvtGenParticles.cc`
- file `addHerwigParticles.cc`
- file `addIsajetParticles.cc`
- file `addParticleTable.cc`
- file `addPDGParticles.cc`
- file `addPythiaParticles.cc`
- file `addQQParticles.cc`
- file `calculateWidthFromLifetime.cc`
- file `Constituent.cc`
- file `convertTemporaryMap.cc`
- file `DecayChannel.cc`
- file `DecayData.cc`
- file `DefTable.cc`
- file `getIsajetID.cc`
- file `getPDGpid.cc`
- file `getPythiaid.cc`
- file `hasMethods.cc`
- file `lifetime.cc`
- file `ParticleDataTable.cc`
- file `ParticleID.cc`
- file `QQChannel.cc`
- file `QQDecay.cc`
- file `quarks.cc`
- file `ResonanceStructure.cc`
- file `spindtoi.cc`
- file `spinitod.cc`
- file `stringtodouble.cc`
- file `TempParticleData.cc`
- file `HepPDT/Version.cc`
- file `write.cc`

5.6 /home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/include/HepPID/ Directory Reference

Files

- file **ParticleIDMethods.hh**
- file **ParticleIDTranslations.hh**
- file **ParticleName.hh**
- file **HepPID/Version.hh**

5.7 /home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/src/HepPID/ Directory Reference

Files

- file **ParticleIDMethods.cc**
- file **ParticleName.cc**
- file **translateEvtGen.cc**
- file **translateGeanttoPDT.cc**
- file **translateHerwig.cc**
- file **translateIsajet.cc**
- file **translatePDG.cc**
- file **translatePDTtoGeant.cc**
- file **translatePythia.cc**
- file **translateQQ.cc**
- file **HepPID/Version.cc**

5.8 /home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/tests/HepPID/ Directory Reference

Files

- file `listEvtGenTranslation.cc`
- file `listHerwigTranslation.cc`
- file `listIsajetTranslation.cc`
- file `listParticleNames.cc`
- file `listPDGTranslation.cc`
- file `listPythiaTranslation.cc`
- file `listQQTranslation.cc`
- file `testParticleIDMethods.cc`

5.9 /home/cepa01/garren/lcg/heppdt/HepPDT- 2.04.02/examples/HepPID/ Directory Reference

Files

- file examListHerwig.cc
- file examListIsajet.cc
- file examListPythia.cc

5.10 /home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/include/ Directory Reference

Directories

- directory **HepPDT**
- directory **HepPID**

5.11 /home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/src/ Directory Reference

Directories

- directory **HepPDT**
- directory **HepPID**

5.12 /home/cepa01/garren/lcg/heppdt/HepPDT-2.04.02/tests/ Directory Reference

Directories

- directory **HepPDT**
- directory **HepPID**

Files

- file `list_of_tests.cc`

Chapter 6

HepPDT Namespace Documentation

6.1 HepPDT Namespace Reference

HepPDT (p. 23) is a Particle Data Table namespace.

Classes

- class **CommonParticleData**
*The **CommonParticleData** (p. 65) class holds basic particle data.*
- class **Constituent**
*The **Constituent** (p. 76) class has information about constituent particles.*
- class **DecayChannel**
*The **DecayChannel** (p. 79) class holds particle decay information.*
- class **DecayData**
*The **DecayData** (p. 82) class holds particle decay information.*
- class **DefTable**
*The **DefTable** (p. 85) class holds EvtGen definitions.*
- class **Measurement**
*The **Measurement** (p. 88) class defines a value with its error.*
- class **ParticleData**
*The **ParticleData** (p. 91) class holds data for a single particle in the table.*
- class **ParticleDataTable**
*The **ParticleDataTable** (p. 100) class is the core of **HepPDT** (p. 23).*
- class **ParticleDataTableComparison**
*The **ParticleDataTableComparison** (p. 108) class provides a utility for sorting the PDT.*
- struct **Quarks**

constituent quarks

- class **ParticleID**
*The **ParticleID** (p. 109) has various utilities to extract information from the particle ID.*
- struct **QQhelicity**
QQ decay helicity information.
- struct **QQAngularHelicity**
QQ decay angular helicity information.
- struct **QQmatrix**
QQ decay matrix information.
- class **QQChannel**
QQChannel (p. 120) is a temporary holder for QQ decay information.
- class **QQDecay**
QQDecay (p. 127) is a temporary holder for QQ decay information.
- class **ResonanceStructure**
*The **ResonanceStructure** (p. 134) class is holds mass and width information.*
- class **SpinState**
*The **SpinState** (p. 138) class holds spin information.*
- class **TableBuilder**
*The **TableBuilder** (p. 141) class is used to construct a **ParticleDataTable** (p. 100).*
- struct **TempConstituent**
constituent (e.g., quark) information
- struct **TempDecayData**
temporary holder for decay data
- struct **TempAliasData**
Hold Alias information from EvtGen.
- struct **TempParticleData**
temporary holder for Particle Data information

Namespaces

- namespace **detail**
HepPDT::detail (p. 36) is for internal use.

Typedefs

- typedef std::vector< **TempDecayData** > **TDDlist**
useful typedef

Enumerations

- enum **location** {
 nj = 1, **nq3**, **nq2**, **nq1**,
 nl, **nr**, **n**, **n8**,
 n9, **n10** }

Functions

- void **swap** (**CommonParticleData** &first, **CommonParticleData** &second)
- void **swap** (**Constituent** &first, **Constituent** &second)
- void **swap** (**DecayChannel** &first, **DecayChannel** &second)
- void **swap** (**DecayData** &first, **DecayData** &second)
- void **swap** (**Measurement** &first, **Measurement** &second)
- double **NaN** ()
- void **swap** (**ParticleData** &first, **ParticleData** &second)
- bool **writePDGStream** (std::ostream &os, const **ParticleDataTable** &table)
- bool **writePythiaStream** (std::ostream &os, const **ParticleDataTable** &table)
- bool **writeHerwigStream** (std::ostream &os, const **ParticleDataTable** &table)
- bool **writeIsajetStream** (std::ostream &os, const **ParticleDataTable** &table)
- bool **writeQQStream** (std::ostream &os, const **ParticleDataTable** &table)
- bool **writeEvtGenStream** (std::ostream &os, const **ParticleDataTable** &table)
- double **spinitod** (int js)
convert from 2J+1 to the actual spin value
- int **spindtoi** (double spin)
convert an actual spin to 2J+1
- void **swap** (**ParticleID** &first, **ParticleID** &second)
- void **swap** (**QQChannel** &first, **QQChannel** &second)
- void **swap** (**QQDecay** &first, **QQDecay** &second)
- void **swap** (**ResonanceStructure** &first, **ResonanceStructure** &second)
- void **swap** (**SpinState** &first, **SpinState** &second)
- double **stringtodouble** (std::string &numb)
extract a double from a string
- bool **addPDGParticles** (std::istream &pdfile, **TableBuilder** &tb)
read PDG input and add particles to the table
- bool **addPythiaParticles** (std::istream &pdfile, **TableBuilder** &tb)
read Pythia input and add particles to the table
- bool **addHerwigParticles** (std::istream &pdfile, **TableBuilder** &tb)

- **bool addIsajetParticles** (std::istream &pdfile, **TableBuilder** &tb)
read Isajet particle input and add particles to the table
- **bool addIsajetDecay** (std::istream &pdfile, **TableBuilder** &tb)
read Isajet decay input and add decay information to the table
- **bool addQQParticles** (std::istream &pdfile, **TableBuilder** &tb)
read QQ input and add particles to the table
- **bool addEvtGenParticles** (std::istream &pdfile, **TableBuilder** &tb)
read EvtGen input and add particles to the table
- **bool addParticleTable** (std::istream &pdfile, **TableBuilder** &tb)
read particle.tbl (or something similar) and add particles to the table
- **double calculateWidthFromLifetime** (double)
- **void swap** (**TempParticleData** &first, **TempParticleData** &second)
- **void version** ()
print HepPDT (p. 23) version
- **void writeVersion** (std::ostream &os)
write HepPDT (p. 23) version to os
- **std::string versionName** ()
return HepPDT (p. 23) version
- **bool getEvtGenLineType** (std::string <ype, int &id, std::string &name, const std::string &pdline)
- **void parseEvtGenLine** (**TempParticleData** &tpd, const std::string &pdline)
- **void parseEvtGenAlias** (**TempAliasData** &tad, const std::string &pdline)
- **bool parseEvtGenDecayLine** (**TempParticleData** &tpd, const std::string &pdline)
- **bool parseEvtGenAliasDecayLine** (**TempAliasData** &tad, const std::string &pdline)
- **TempDecayData getEvtGenDecay** (const std::string &pdline)
- **void parseEvtGenConj** (std::string &cname, const std::string &pdline)
- **void parseEvtGenDefinition** (std::string &def, double &val, const std::string &pdline)
- **void findDecayModel** (**TempParticleData** &tpd, **TableBuilder** &tb)
- **void findAliasDecayModel** (**TempAliasData** &tad, **TableBuilder** &tb)
- **bool getQQLineType** (std::string <ype, int &id, std::string &name, const std::string &pdline)
- **bool parseQQDecayLine** (const std::string &pdline, **QQDecay** &qd)
- **void parseQQParticle** (**TempParticleData** &tpd, const std::string &pdline)
- **void buildQQDecay** (std::istream &thisline, std::string <ype, **QQDecay** &dcy, **QQChannel** &ch)
- **void buildTPD** (**TempParticleData** &tpd, **QQDecay** &qd)

6.1.1 Detailed Description

HepPDT (p. 23) is a Particle Data Table namespace.

The **HepPDT** (p. 23) classes are used to create a Particle Data Table.

6.1.2 Typedef Documentation

6.1.2.1 typedef std::vector<TempDecayData> HepPDT::TDDlist

useful typedef

Definition at line 45 of file TempParticleData.hh.

6.1.3 Enumeration Type Documentation

6.1.3.1 enum HepPDT::location

PID digits (base 10) are: n nr nl nq1 nq2 nq3 nj The location enum provides a convenient index into the PID.

Enumerator:

nj
nq3
nq2
nq1
nl
nr
n
n8
n9
n10

Definition at line 36 of file ParticleID.hh.

6.1.4 Function Documentation

6.1.4.1 bool HepPDT::addEvtGenParticles (std::istream & *pdfile*, TableBuilder & *tb*)

read EvtGen input and add particles to the table

Examples:

`listEvtGenNames.cc.in`, and `testReadEvtGen.cc.in`.

Definition at line 32 of file addEvtGenParticles.cc.

References HepPDT::TableBuilder::addAlias(), HepPDT::DefTable::addDefinition(), HepPDT::TableBuilder::addParticle(), HepPDT::TableBuilder::aliasData(), HepPDT::TableBuilder::aliasSize(), HepPDT::TableBuilder::definitions(), findDecayModel(), getEvtGenLineType(), HepPDT::TableBuilder::getParticleData(), HepPDT::TableBuilder::hasAlias(), HepPDT::TableBuilder::hasParticleData(), parseEvtGenAlias(), parseEvtGenAliasDecayLine(), parseEvtGenConj(), parseEvtGenDecayLine(), parseEvtGenDefinition(), parseEvtGenLine(), HepPDT::TableBuilder::size(), HepPDT::TempAliasData::tempChargeConj, and HepPID::translateEvtGentoPDT().

Referenced by main().

6.1.4.2 `bool HepPDT::addHerwigParticles (std::istream & pdfile, TableBuilder & tb)`

Definition at line 13 of file `addHerwigParticles.cc`.

6.1.4.3 `bool HepPDT::addIsajetDecay (std::istream & pdfile, TableBuilder & tb)`

read Isajet decay input and add decay information to the table

Examples:

`testReadIsajet.cc.in`.

Definition at line 32 of file `addIsajetParticles.cc`.

References `HepPDT::detail::getIsajetID()`, `HepPDT::TableBuilder::getParticleData()`, `HepPDT::detail::parseIsajetDecayLine()`, and `HepPID::translateIsajettoPDT()`.

Referenced by `main()`.

6.1.4.4 `bool HepPDT::addIsajetParticles (std::istream & pdfile, TableBuilder & tb)`

read Isajet particle input and add particles to the table

Examples:

`testReadIsajet.cc.in`.

Definition at line 14 of file `addIsajetParticles.cc`.

References `HepPDT::detail::getIsajetID()`, `HepPDT::TableBuilder::getParticleData()`, `HepPDT::detail::parseIsajetLine()`, `HepPDT::TableBuilder::size()`, and `HepPID::translateIsajettoPDT()`.

Referenced by `main()`.

6.1.4.5 `bool HepPDT::addParticleTable (std::istream & pdfile, TableBuilder & tb)`

read `particle.tbl` (or something similar) and add particles to the table

Definition at line 21 of file `addParticleTable.cc`.

References `HepPDT::TableBuilder::getParticleData()`, `HepPDT::detail::getParticleID()`, `HepPDT::ParticleID::isValid()`, `HepPDT::detail::parseParticleLine()`, and `HepPDT::TableBuilder::size()`.

Referenced by `main()`.

6.1.4.6 `bool HepPDT::addPDGParticles (std::istream & pdfile, TableBuilder & tb)`

read PDG input and add particles to the table

Examples:

`listPDGNames.cc.in`, and `testHepPDT.cc`.

Definition at line 22 of file addPDGParticles.cc.

References HepPDT::TableBuilder::getParticleData(), HepPDT::detail::getPDGnames(), HepPDT::detail::getPDGpid(), HepPDT::detail::parsePDGline(), HepPDT::TableBuilder::size(), and HepPDT::TempParticleData::tempSource.

Referenced by main().

6.1.4.7 bool HepPDT::addPythiaParticles (std::istream & *pfile*, TableBuilder & *tb*)

read Pythia input and add particles to the table

Examples:

`listPythiaNames.cc.in.`

Definition at line 20 of file addPythiaParticles.cc.

References HepPDT::TableBuilder::getAntiParticle(), HepPDT::TableBuilder::getParticleData(), HepPDT::detail::getPythiaid(), HepPDT::detail::parsePythiaDecayLine(), HepPDT::detail::parsePythiaLine(), HepPDT::TableBuilder::size(), HepPDT::TempParticleData::tempMass, HepPDT::TempParticleData::tempOriginalID, HepPDT::TempParticleData::tempSource, and HepPID::translatePythiatoPDT().

Referenced by main().

6.1.4.8 bool HepPDT::addQQParticles (std::istream & *pfile*, TableBuilder & *tb*)

read QQ input and add particles to the table

Examples:

`testReadQQ.cc.in.`

Definition at line 32 of file addQQParticles.cc.

References HepPDT::TableBuilder::addParticle(), buildTPD(), HepPDT::TableBuilder::getParticleData(), getQQLineType(), HepPDT::TableBuilder::hasParticleData(), parseQQDecayLine(), parseQQParticle(), HepPDT::TableBuilder::size(), HepPDT::TempParticleData::tempOriginalID, HepPDT::TempParticleData::tempParticleName, HepPDT::TempParticleData::tempSource, HepPID::translateQQbar(), and HepPID::translateQQtoPDT().

Referenced by main().

6.1.4.9 void HepPDT::buildQQDecay (std::istringstream & *thisline*, std::string & *ltype*, QQDecay & *dcy*, QQChannel & *ch*)

Definition at line 192 of file addQQParticles.cc.

References HepPDT::QQmatrix::a, HepPDT::QQChannel::addAngHelicity(), HepPDT::QQDecay::addChannel(), HepPDT::QQChannel::addHelicity(), HepPDT::QQmatrix::b, HepPDT::QQmatrix::c, HepPDT::QQChannel::clear(), HepPDT::QQDecay::makeStable(), HepPDT::QQChannel::parse(), HepPDT::QQChannel::setCPTag(), HepPDT::QQChannel::setMatrix(), and HepPDT::QQChannel::setSinPhi().

Referenced by parseQQDecayLine().

6.1.4.10 void HepPDT::buildTPD (TempParticleData & *tpd*, QQDecay & *qd*)

Definition at line 224 of file addQQParticles.cc.

References HepPDT::QQChannel::branchingFraction(), HepPDT::QQDecay::channel(), HepPDT::QQChannel::daughter(), HepPDT::QQChannel::matrixCode(), HepPDT::ParticleID::pid(), HepPDT::QQDecay::size(), HepPDT::QQChannel::sizeDtr(), HepPDT::TempParticleData::tempDecayList, and HepPDT::TempParticleData::tempID.

Referenced by addQQParticles().

6.1.4.11 double HepPDT::calculateWidthFromLifetime (double)

free function Given the lifetime, calculate the width.

Definition at line 13 of file calculateWidthFromLifetime.cc.

Referenced by parseEvtGenLine(), HepPDT::detail::parseParticleLine(), HepPDT::detail::parsePythiaLine(), and parseQQParticle().

6.1.4.12 void HepPDT::findAliasDecayModel (TempAliasData & *tad*, TableBuilder & *tb*)

Definition at line 154 of file addEvtGenParticles.cc.

References HepPDT::TableBuilder::definition(), HepPDT::TableBuilder::hasAlias(), HepPDT::TableBuilder::hasDefinition(), HepPDT::TableBuilder::hasParticleData(), stringtodouble(), and HepPDT::TempAliasData::tempAliasDecayList.

6.1.4.13 void HepPDT::findDecayModel (TempParticleData & *tpd*, TableBuilder & *tb*)

Definition at line 109 of file addEvtGenParticles.cc.

References HepPDT::TableBuilder::definition(), HepPDT::TableBuilder::hasAlias(), HepPDT::TableBuilder::hasDefinition(), HepPDT::TableBuilder::hasParticleData(), stringtodouble(), and HepPDT::TempParticleData::tempDecayList.

Referenced by addEvtGenParticles().

6.1.4.14 TempDecayData HepPDT::getEvtGenDecay (const std::string & *pdl*)

Definition at line 298 of file addEvtGenParticles.cc.

References HepPDT::TempDecayData::tempBranchingFraction, and HepPDT::TempDecayData::tempDaughterList.

Referenced by parseEvtGenAliasDecayLine(), and parseEvtGenDecayLine().

6.1.4.15 bool HepPDT::getEvtGenLineType (std::string & *ltype*, int & *id*, std::string & *name*, const std::string & *pdl*)

Definition at line 199 of file addEvtGenParticles.cc.

Referenced by addEvtGenParticles().

6.1.4.16 `bool HepPDT::getQQLineType (std::string & ltype, int & id, std::string & name, const std::string & pdl)`

Definition at line 85 of file `addQQParticles.cc`.

Referenced by `addQQParticles()`.

6.1.4.17 `double HepPDT::NaN () [inline]`

Definition at line 11 of file `Measurement.icc`.

6.1.4.18 `void HepPDT::parseEvtGenAlias (TempAliasData & tad, const std::string & pdl)`

Definition at line 341 of file `addEvtGenParticles.cc`.

References `HepPDT::TempAliasData::tempAlias`, and `HepPDT::TempAliasData::tempAliasedParticle`.

Referenced by `addEvtGenParticles()`.

6.1.4.19 `bool HepPDT::parseEvtGenAliasDecayLine (TempAliasData & tad, const std::string & pdl)`

Definition at line 318 of file `addEvtGenParticles.cc`.

References `getEvtGenDecay()`, and `HepPDT::TempAliasData::tempAliasDecayList`.

Referenced by `addEvtGenParticles()`.

6.1.4.20 `void HepPDT::parseEvtGenConj (std::string & cname, const std::string & pdl)`

Definition at line 358 of file `addEvtGenParticles.cc`.

Referenced by `addEvtGenParticles()`.

6.1.4.21 `bool HepPDT::parseEvtGenDecayLine (TempParticleData & tpd, const std::string & pdl)`

Definition at line 267 of file `addEvtGenParticles.cc`.

References `getEvtGenDecay()`, `HepPDT::ParticleID::pid()`, `HepPDT::TempParticleData::tempDecayList`, and `HepPDT::TempParticleData::tempPID`.

Referenced by `addEvtGenParticles()`.

6.1.4.22 `void HepPDT::parseEvtGenDefinition (std::string & def, double & val, const std::string & pdl)`

Definition at line 373 of file `addEvtGenParticles.cc`.

Referenced by `addEvtGenParticles()`.

6.1.4.23 void HepPDT::parseEvtGenLine (TempParticleData & *tpd*, const std::string & *pdl*)

Definition at line 231 of file addEvtGenParticles.cc.

References calculateWidthFromLifetime(), HepPDT::ParticleID::pid(), HepPDT::SpinState::setTotalSpin(), HepPDT::TempParticleData::tempCharge, HepPDT::TempParticleData::tempHighCutoff, HepPDT::TempParticleData::tempID, HepPDT::TempParticleData::tempMass, HepPDT::TempParticleData::tempOriginalID, HepPDT::TempParticleData::tempParticleName, HepPDT::TempParticleData::tempSource, HepPDT::TempParticleData::tempSpin, HepPDT::TempParticleData::tempWidth, and HepPDT::SpinState::totalSpin().

Referenced by addEvtGenParticles().

6.1.4.24 bool HepPDT::parseQQDecayLine (const std::string & *pdl*, QQDecay & *qd*)

Definition at line 167 of file addQQParticles.cc.

References buildQQDecay().

Referenced by addQQParticles().

6.1.4.25 void HepPDT::parseQQParticle (TempParticleData & *tpd*, const std::string & *pdl*)

Definition at line 126 of file addQQParticles.cc.

References calculateWidthFromLifetime(), HepPDT::ParticleID::pid(), HepPDT::SpinState::setTotalSpin(), HepPDT::TempParticleData::tempCharge, HepPDT::TempParticleData::tempHighCutoff, HepPDT::TempParticleData::tempID, HepPDT::TempParticleData::tempLowCutoff, HepPDT::TempParticleData::tempMass, HepPDT::TempParticleData::tempOriginalID, HepPDT::TempParticleData::tempParticleName, HepPDT::TempParticleData::tempSpin, HepPDT::TempParticleData::tempWidth, and HepPDT::SpinState::totalSpin().

Referenced by addQQParticles().

6.1.4.26 int HepPDT::spindtoi (double *spin*)

convert an actual spin to $2J+1$

Definition at line 13 of file spindtoi.cc.

6.1.4.27 double HepPDT::spinitod (int *js*)

convert from $2J+1$ to the actual spin value

Examples:

`testPID.cc.`

Definition at line 13 of file spinitod.cc.

Referenced by main(), and HepPDT::TempParticleData::processPID().

6.1.4.28 double HepPDT::stringtodouble (std::string & *numb*)

extract a double from a string

Definition at line 15 of file stringtodouble.cc.

Referenced by findAliasDecayModel(), and findDecayModel().

6.1.4.29 void HepPDT::swap (TempParticleData & *first*, TempParticleData & *second*) [inline]

Definition at line 106 of file TempParticleData.hh.

References HepPDT::TempParticleData::swap().

Referenced by HepPDT::TempParticleData::swap(), HepPDT::SpinState::swap(), HepPDT::ResonanceStructure::swap(), HepPDT::QQDecay::swap(), HepPDT::QQChannel::swap(), HepPDT::ParticleID::swap(), HepPDT::ParticleData::swap(), HepPDT::Measurement::swap(), HepPDT::DecayData::swap(), HepPDT::DecayChannel::swap(), HepPDT::Constituent::swap(), and HepPDT::CommonParticleData::swap().

6.1.4.30 void HepPDT::swap (SpinState & *first*, SpinState & *second*) [inline]

Definition at line 69 of file SpinState.hh.

References HepPDT::SpinState::swap().

6.1.4.31 void HepPDT::swap (ResonanceStructure & *first*, ResonanceStructure & *second*) [inline]

Definition at line 83 of file ResonanceStructure.hh.

References HepPDT::ResonanceStructure::swap().

6.1.4.32 void HepPDT::swap (QQDecay & *first*, QQDecay & *second*) [inline]

Definition at line 71 of file QQDecay.hh.

References HepPDT::QQDecay::swap().

6.1.4.33 void HepPDT::swap (QQChannel & *first*, QQChannel & *second*) [inline]

Definition at line 141 of file QQChannel.hh.

References HepPDT::QQChannel::swap().

6.1.4.34 void HepPDT::swap (ParticleID & *first*, ParticleID & *second*) [inline]

Definition at line 159 of file ParticleID.hh.

References HepPDT::ParticleID::swap().

6.1.4.35 `void HepPDT::swap (ParticleData & first, ParticleData & second)`
[inline]

Definition at line 181 of file ParticleData.hh.

References HepPDT::ParticleData::swap().

6.1.4.36 `void HepPDT::swap (Measurement & first, Measurement & second)`
[inline]

Definition at line 45 of file Measurement.hh.

References HepPDT::Measurement::swap().

6.1.4.37 `void HepPDT::swap (DecayData & first, DecayData & second)` [inline]

Definition at line 66 of file DecayData.hh.

References HepPDT::DecayData::swap().

6.1.4.38 `void HepPDT::swap (DecayChannel & first, DecayChannel & second)`
[inline]

Definition at line 91 of file DecayChannel.hh.

References HepPDT::DecayChannel::swap().

6.1.4.39 `void HepPDT::swap (Constituent & first, Constituent & second)` [inline]

Definition at line 76 of file Constituent.hh.

References HepPDT::Constituent::swap().

6.1.4.40 `void HepPDT::swap (CommonParticleData & first, CommonParticleData & second)` [inline]

Definition at line 166 of file CommonParticleData.hh.

References HepPDT::CommonParticleData::swap().

6.1.4.41 `void HepPDT::version ()`

print **HepPDT** (p. 23) version

Definition at line 20 of file HepPDT/Version.cc.

References versionName().

Referenced by HepPDT::ParticleDataTable::ParticleDataTable().

6.1.4.42 `std::string HepPDT::versionName ()`

return **HepPDT** (p. 23) version

Definition at line 15 of file HepPDT/Version.cc.

Referenced by `version()`, and `writeVersion()`.

6.1.4.43 `bool HepPDT::writeEvtGenStream (std::ostream & os, const ParticleDataTable & table)`

6.1.4.44 `bool HepPDT::writeHerwigStream (std::ostream & os, const ParticleDataTable & table)`

6.1.4.45 `bool HepPDT::writeIsajetStream (std::ostream & os, const ParticleDataTable & table)`

6.1.4.46 `bool HepPDT::writePDGStream (std::ostream & os, const ParticleDataTable & table)`

6.1.4.47 `bool HepPDT::writePythiaStream (std::ostream & os, const ParticleDataTable & table)`

6.1.4.48 `bool HepPDT::writeQQStream (std::ostream & os, const ParticleDataTable & table)`

6.1.4.49 `void HepPDT::writeVersion (std::ostream & os)`

write **HepPDT** (p. 23) version to os

Definition at line 26 of file HepPDT/Version.cc.

References `versionName()`.

Referenced by `HepPDT::ParticleDataTable::writeParticleData()`, and `HepPDT::ParticleDataTable::writeParticleInfo()`.

6.2 HepPDT::detail Namespace Reference

HepPDT::detail (p. 36) is for internal use.

Functions

- void **getPDGpid** (std::vector< int > &idlist, std::string &pdlne)
for internal use
- void **getPDGnames** (std::vector< std::string > &name1st, std::string &pdlne)
for internal use
- void **parsePDGline** (TempParticleData &tpd, std::string &pdlne)
for internal use
- bool **CheckPDGEntry** (TempParticleData &tpd, const std::string &, double, double)
for internal use
- bool **getPythiaid** (int &id, const std::string &pdlne)
for internal use
- void **parsePythiaLine** (TempParticleData &tpd, int &anti, std::string &aname, const std::string &pdlne)
for internal use
- void **parsePythiaDecayLine** (TempParticleData &tpd, const std::string &pdlne)
for internal use
- TempDecayData **getPythiaDecay** (const std::string &pdlne)
for internal use
- bool **getIsajetID** (int &id, const std::string &pdlne)
for internal use
- void **parseIsajetLine** (TempParticleData &tpd, const std::string &pdlne)
for internal use
- void **parseIsajetDecayLine** (TempParticleData &tpd, const std::string &pdlne, Table-Builder &tb)
for internal use
- bool **getParticleID** (int &id, const std::string &pdlne)
for internal use
- void **parseParticleLine** (TempParticleData &tpd, const std::string &pdlne)
for internal use

6.2.1 Detailed Description

HepPDT::detail (p.36) is for internal use.

This namespace encapsulates free functions used when parsing various input streams.

6.2.2 Function Documentation

6.2.2.1 `bool HepPDT::detail::CheckPDGEntry (TempParticleData & tpd, const std::string &, double, double)`

for internal use

Definition at line 67 of file addPDGParticles.cc.

References `HepPDT::TempParticleData::tempMass`, and `HepPDT::TempParticleData::tempWidth`.

Referenced by `parsePDGline()`.

6.2.2.2 `bool HepPDT::detail::getIsajetID (int & id, const std::string & pdline)`

for internal use

Definition at line 17 of file getIsajetID.cc.

Referenced by `HepPDT::addIsajetDecay()`, and `HepPDT::addIsajetParticles()`.

6.2.2.3 `bool HepPDT::detail::getParticleID (int & id, const std::string & pdline)`

for internal use

Definition at line 42 of file addParticleTable.cc.

Referenced by `HepPDT::addParticleTable()`.

6.2.2.4 `void HepPDT::detail::getPDGnames (std::vector< std::string > & namelst, std::string & pdline)`

for internal use

Definition at line 40 of file getPDGpid.cc.

Referenced by `HepPDT::addPDGParticles()`.

6.2.2.5 `void HepPDT::detail::getPDGpid (std::vector< int > & idlist, std::string & pdline)`

for internal use

Definition at line 20 of file getPDGpid.cc.

Referenced by `HepPDT::addPDGParticles()`.

6.2.2.6 TempDecayData HepPDT::detail::getPythiaDecay (const std::string & *pdl*)

for internal use

Definition at line 106 of file addPythiaParticles.cc.

References HepPDT::TempDecayData::tempBranchingFraction, HepPDT::TempDecayData::tempDaughterList, and HepPDT::TempDecayData::tempDecayName.

Referenced by parsePythiaDecayLine().

6.2.2.7 bool HepPDT::detail::getPythiaid (int & *id*, const std::string & *pdl*)

for internal use

Definition at line 20 of file getPythiaid.cc.

Referenced by HepPDT::addPythiaParticles().

6.2.2.8 void HepPDT::detail::parseIsajetDecayLine (TempParticleData & *tpd*, const std::string & *pdl*, TableBuilder & *tb*)

for internal use

Definition at line 83 of file addIsajetParticles.cc.

References HepPDT::TableBuilder::getParticleData(), HepPDT::ParticleID::pid(), HepPDT::TempDecayData::tempBranchingFraction, HepPDT::TempDecayData::tempDaughterList, HepPDT::TempParticleData::tempDecayList, HepPDT::TempDecayData::tempDecayName, HepPDT::TempParticleData::tempID, HepPDT::TempParticleData::tempParticleName, and HepPID::translateIsajettoPDT().

Referenced by HepPDT::addIsajetDecay().

6.2.2.9 void HepPDT::detail::parseIsajetLine (TempParticleData & *tpd*, const std::string & *pdl*)

for internal use

Definition at line 53 of file addIsajetParticles.cc.

References HepPDT::ParticleID::pid(), HepPDT::TempParticleData::tempCharge, HepPDT::TempParticleData::tempID, HepPDT::TempParticleData::tempMass, HepPDT::TempParticleData::tempParticleName, and HepPDT::TempParticleData::tempSpin.

Referenced by HepPDT::addIsajetParticles().

6.2.2.10 void HepPDT::detail::parseParticleLine (TempParticleData & *tpd*, const std::string & *pdl*)

for internal use

Definition at line 62 of file addParticleTable.cc.

References HepPDT::calculateWidthFromLifetime(), HepPID::particleName(), HepPDT::ParticleID::pid(), HepPDT::TempParticleData::tempCharge, HepPDT::TempParticleData::tempID, HepPDT::TempParticleData::tempMass, HepPDT::TempParticleData::temp

OriginalID, HepPDT::TempParticleData::tempParticleName, HepPDT::TempParticleData::tempSource, and HepPDT::TempParticleData::tempWidth.

Referenced by HepPDT::addParticleTable().

6.2.2.11 void HepPDT::detail::parsePDGline (TempParticleData & *tpd*, std::string & *pdl*)

for internal use

Definition at line 51 of file addPDGParticles.cc.

References CheckPDGEntry().

Referenced by HepPDT::addPDGParticles().

6.2.2.12 void HepPDT::detail::parsePythiaDecayLine (TempParticleData & *tpd*, const std::string & *pdl*)

for internal use

Definition at line 98 of file addPythiaParticles.cc.

References getPythiaDecay(), HepPDT::ParticleID::pid(), HepPDT::TempParticleData::tempDecayList, and HepPDT::TempParticleData::tempID.

Referenced by HepPDT::addPythiaParticles().

6.2.2.13 void HepPDT::detail::parsePythiaLine (TempParticleData & *tpd*, int & *anti*, std::string & *aname*, const std::string & *pdl*)

for internal use

Definition at line 56 of file addPythiaParticles.cc.

References HepPDT::calculateWidthFromLifetime(), HepPDT::ParticleID::pid(), HepPDT::TempParticleData::tempCharge, HepPDT::TempParticleData::tempColorCharge, HepPDT::TempParticleData::tempHighCutoff, HepPDT::TempParticleData::tempID, HepPDT::TempParticleData::tempMass, HepPDT::TempParticleData::tempOriginalID, HepPDT::TempParticleData::tempParticleName, HepPDT::TempParticleData::tempSource, and HepPDT::TempParticleData::tempWidth.

Referenced by HepPDT::addPythiaParticles().

6.3 HepPID Namespace Reference

The **HepPID** (p. 40) namespace has independent particle ID translation methods.

Typedefs

- typedef std::map< int, std::string > **ParticleNameMap**
define ParticleNameMap
- typedef std::map< int, std::string > **ParticleNameMap**
- typedef std::map< int, int > **EvtGenPDTMap**
- typedef std::map< int, int > **PDTEvtGenMap**
- typedef std::map< int, int > **HerwigPDTMap**
- typedef std::map< int, int > **PDTHerwigMap**
- typedef std::map< int, int > **IsajetPDTMap**
- typedef std::map< int, int > **PDTIsajetMap**
- typedef std::map< int, int > **PDGtoPDTMap**
- typedef std::map< int, int > **PDTtoPDGMap**
- typedef std::map< int, int > **PythiaPDTMap**
- typedef std::map< int, int > **PDTPythiaMap**
- typedef std::map< int, int > **QQPDTMap**
- typedef std::map< int, int > **PDTQQMap**
- typedef std::map< int, int > **QQbarMap**
- typedef std::map< int, int > **InverseQQbarMap**

Enumerations

- enum **location** {
 nj = 1, **nq3**, **nq2**, **nq1**,
 nl, **nr**, **n**, **n8**,
 n9, **n10** }

Functions

- unsigned short **digit** (**location** loc, const int &pid)
return the digit at a named location in the PID
- int **A** (const int &pid)
- int **Z** (const int &pid)
- int **lambda** (const int &pid)
- int **abspid** (const int &pid)
absolute value of particle ID
- int **fundamentalID** (const int &pid)
return the first two digits if this is a "fundamental" particle
- bool **hasFundamentalAnti** (const int &pid)
if this is a fundamental particle, does it have a valid antiparticle?

- int **extraBits** (const int &pid)
- bool **isValid** (const int &pid)
is this a valid ID?
- bool **isMeson** (const int &pid)
is this a valid meson ID?
- bool **isBaryon** (const int &pid)
is this a valid baryon ID?
- bool **isDiQuark** (const int &pid)
is this a valid diquark ID?
- bool **isHadron** (const int &pid)
is this a valid hadron ID?
- bool **isLepton** (const int &pid)
is this a valid lepton ID?
- bool **isNucleus** (const int &pid)
is this a valid ion ID?
- bool **isPentaquark** (const int &pid)
is this a valid pentaquark ID?
- bool **isSUSY** (const int &pid)
is this a valid SUSY ID?
- bool **isRhadron** (const int &pid)
is this a valid R-hadron ID?
- bool **hasUp** (const int &pid)
does this particle contain an up quark?
- bool **hasDown** (const int &pid)
does this particle contain a down quark?
- bool **hasStrange** (const int &pid)
does this particle contain a strange quark?
- bool **hasCharm** (const int &pid)
does this particle contain a charm quark?
- bool **hasBottom** (const int &pid)
does this particle contain a bottom quark?
- bool **hasTop** (const int &pid)
does this particle contain a top quark?

- int **jSpin** (const int &pid)
jSpin returns $2J+1$, where J is the total spin
- int **sSpin** (const int &pid)
sSpin returns $2S+1$, where S is the spin
- int **lSpin** (const int &pid)
lSpin returns $2L+1$, where L is the orbital angular momentum
- int **threeCharge** (const int &pid)
return 3 times the charge (3 x quark charge is an int)
- int **translateHerwigtoPDT** (const int herwigID)
translate Herwig to PDG standard
- int **translatePDTtoHerwig** (const int pid)
translate PDG standard to Herwig
- void **writeHerwigTranslation** (std::ostream &os)
output the translation list
- int **translateIsajettoPDT** (const int isajetID)
translate Isajet to PDG standard
- int **translatePDTtoIsajet** (const int pid)
translate PDG standard to Isajet
- void **writeIsajetTranslation** (std::ostream &os)
output the translation list
- int **translatePythiatoPDT** (const int pythiaID)
translate Pythia to PDG standard
- int **translatePDTtoPythia** (const int pid)
translate PDG standard to Pythia
- void **writePythiaTranslation** (std::ostream &os)
output the translation list
- int **translateEvtGentoPDT** (const int evtGenID)
translate EvtGen to PDG standard
- int **translatePDTtoEvtGen** (const int pid)
translate PDG standard to EvtGen
- void **writeEvtGenTranslation** (std::ostream &os)
output the translation list
- int **translatePDGtabletoPDT** (const int pdgID)
translate PDG table to PDG standard

- int **translatePDTtoPDGtable** (const int pid)
translate PDG standard to PDG table
- void **writePDGTranslation** (std::ostream &os)
output the translation list
- int **translateQQtoPDT** (const int qqID)
translate QQ to PDG standard
- int **translatePDTtoQQ** (const int pid)
translate PDG standard to QQ
- int **translateQQbar** (const int id)
QQ helper function.
- int **translateInverseQQbar** (const int id)
QQ helper function.
- void **writeQQTranslation** (std::ostream &os)
output the translation list
- int **translateGeanttoPDT** (const int geantID)
translate Geant3 to PDG standard
- int **translatePDTtoGeant** (const int pid)
translate PDG standard to Geant3
- std::string **particleName** (const int &)
*get a known **HepPID** (p. 40) Particle name*
- void **listParticleNames** (std::ostream &os)
list all known names
- bool **validParticleName** (const int &)
verify that this number has a valid name
- **ParticleNameMap** const & **getParticleNameMap** ()
access the ParticleNameMap for other purposes
- void **version** ()
*print **HepPID** (p. 40) version*
- void **writeVersion** (std::ostream &os)
*write **HepPID** (p. 40) version to os*
- std::string **versionName** ()
*return **HepPID** (p. 40) version*
- **ParticleNameMap** const & **ParticleNameInit** ()

- void **writeParticleNameLine** (int i, std::ostream &os)
- **EvtGenPDTMap** const & **getEvtGenPDTMap** ()
- **PDTEvtGenMap** const & **getPDTEvtGenMap** ()
- **EvtGenPDTMap** const & **EvtGenPDTMapInit** ()
- **PDTEvtGenMap** const & **PDTEvtGenMapInit** ()
- **EvtGenPDTMap** const & **getEvtGenPDTMap** ()
- **PDTEvtGenMap** const & **getPDTEvtGenMap** ()
- void **writeEvtGenTranslationLine** (int i, std::ostream &os)
- **HerwigPDTMap** const & **getHerwigPDTMap** ()
- **PDTHerwigMap** const & **getPDTHerwigMap** ()
- **HerwigPDTMap** const & **HerwigPDTMapInit** ()
- **PDTHerwigMap** const & **PDTHerwigMapInit** ()
- **HerwigPDTMap** const & **getHerwigPDTMap** ()
- **PDTHerwigMap** const & **getPDTHerwigMap** ()
- void **writeHerwigTranslationLine** (int i, std::ostream &os)
- **IsajetPDTMap** const & **getIsajetPDTMap** ()
- **PDTIsajetMap** const & **getPDTIsajetMap** ()
- **IsajetPDTMap** const & **IsajetPDTMapInit** ()
- **PDTIsajetMap** const & **PDTIsajetMapInit** ()
- int **convIsajettoPDT** (const int id)
- int **convPDTtoIsajet** (const int id)
- **IsajetPDTMap** const & **getIsajetPDTMap** ()
- **PDTIsajetMap** const & **getPDTIsajetMap** ()
- void **writeIsajetTranslationLine** (int i, std::ostream &os)
- **PDGtoPDTMap** const & **getPDGtoPDTMap** ()
- **PDTtoPDGMap** const & **getPDTtoPDGMap** ()
- **PDGtoPDTMap** const & **PDGtoPDTMapInit** ()
- **PDTtoPDGMap** const & **PDTtoPDGMapInit** ()
- **PDGtoPDTMap** const & **getPDGtoPDTMap** ()
- **PDTtoPDGMap** const & **getPDTtoPDGMap** ()
- void **writePDGTranslationLine** (int i, std::ostream &os)
- **PythiaPDTMap** const & **getPythiaPDTMap** ()
- **PDTPythiaMap** const & **getPDTPythiaMap** ()
- **PythiaPDTMap** const & **PythiaPDTMapInit** ()
- **PDTPythiaMap** const & **PDTPythiaMapInit** ()
- **PythiaPDTMap** const & **getPythiaPDTMap** ()
- **PDTPythiaMap** const & **getPDTPythiaMap** ()
- void **writePythiaTranslationLine** (int i, std::ostream &os)
- **QQPDTMap** const & **getQQPDTMap** ()
- **PDTQQMap** const & **getPDTQQMap** ()
- **QQbarMap** const & **getQQbarMap** ()
- **InverseQQbarMap** const & **getInverseQQbarMap** ()
- **QQPDTMap** const & **QQPDTMapInit** ()
- **QQbarMap** const & **QQbarMapInit** ()
- **PDTQQMap** const & **PDTQQMapInit** ()
- **InverseQQbarMap** const & **InverseQQbarMapInit** ()
- **QQPDTMap** const & **getQQPDTMap** ()
- **PDTQQMap** const & **getPDTQQMap** ()
- **QQbarMap** const & **getQQbarMap** ()
- **InverseQQbarMap** const & **getInverseQQbarMap** ()

6.3.1 Detailed Description

The **HepPID** (p. 40) namespace has independent particle ID translation methods.

The **HepPID** (p. 40) namespace contains a set of independent particle ID translation methods

6.3.2 Typedef Documentation

6.3.2.1 `typedef std::map< int, int > HepPID::EvtGenPDTMap`

Definition at line 33 of file `translateEvtGen.cc`.

6.3.2.2 `typedef std::map< int, int > HepPID::HerwigPDTMap`

Definition at line 33 of file `translateHerwig.cc`.

6.3.2.3 `typedef std::map< int, int > HepPID::InverseQQbarMap`

Definition at line 40 of file `translateQQ.cc`.

6.3.2.4 `typedef std::map< int, int > HepPID::IsajetPDTMap`

Definition at line 36 of file `translateIsajet.cc`.

6.3.2.5 `typedef std::map< int, std::string > HepPID::ParticleNameMap`

Definition at line 32 of file `ParticleName.cc`.

6.3.2.6 `typedef std::map< int, std::string > HepPID::ParticleNameMap`

```
define ParticleNameMap
```

Definition at line 29 of file `ParticleName.hh`.

6.3.2.7 `typedef std::map< int, int > HepPID::PDGtoPDTMap`

Definition at line 33 of file `translatePDG.cc`.

6.3.2.8 `typedef std::map< int, int > HepPID::PDTEvtGenMap`

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

6.3.2.9 `typedef std::map< int, int > HepPID::PDTHerwigMap`

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

6.3.2.10 `typedef std::map< int, int > HepPID::PDTIsajetMap`

Definition at line 37 of file `translateIsajet.cc`.

6.3.2.11 typedef std::map< int, int > HepPID::PDTPythiaMap

Definition at line 34 of file translatePythia.cc.

6.3.2.12 typedef std::map< int, int > HepPID::PDTQQMap

Definition at line 38 of file translateQQ.cc.

6.3.2.13 typedef std::map< int, int > HepPID::PDTtoPDGMap

Definition at line 34 of file translatePDG.cc.

6.3.2.14 typedef std::map< int, int > HepPID::PythiaPDTMap

Definition at line 33 of file translatePythia.cc.

6.3.2.15 typedef std::map< int, int > HepPID::QQbarMap

Definition at line 39 of file translateQQ.cc.

6.3.2.16 typedef std::map< int, int > HepPID::QQPDTMap

Definition at line 37 of file translateQQ.cc.

6.3.3 Enumeration Type Documentation**6.3.3.1 enum HepPID::location**

PID digits (base 10) are: n nr nl nq1 nq2 nq3 nj The location enum provides a convenient index into the PID.

Enumerator:

nj
nq3
nq2
nq1
nl
nr
n
n8
n9
n10

Definition at line 24 of file ParticleIDMethods.hh.

6.3.4 Function Documentation

6.3.4.1 int HepPID::A (const int & pid)

if this is a nucleus (ion), get A Ion numbers are +/- 10LZZZAAAI.

Definition at line 59 of file ParticleIDMethods.cc.

References `abspid()`, and `isNucleus()`.

Referenced by `main()`.

6.3.4.2 int HepPID::abspid (const int & pid)

absolute value of particle ID

Definition at line 15 of file ParticleIDMethods.cc.

Referenced by `A()`, `convIsajettoPDT()`, `convPDTtoIsajet()`, `digit()`, `extraBits()`, `fundamentalID()`, `isBaryon()`, `isDiQuark()`, `isMeson()`, `isNucleus()`, `jSpin()`, `lambda()`, `threeCharge()`, and `Z()`.

6.3.4.3 int HepPID::@71::convIsajettoPDT (const int id) [static]

Definition at line 689 of file translateIsajet.cc.

References `abspid()`, `digit()`, `nj`, `nl`, `nq1`, `nq2`, and `nq3`.

Referenced by `translateIsajettoPDT()`.

6.3.4.4 int HepPID::@71::convPDTtoIsajet (const int id) [static]

Definition at line 790 of file translateIsajet.cc.

References `abspid()`, `digit()`, `fundamentalID()`, `nj`, `nl`, `nq1`, `nq2`, and `nq3`.

Referenced by `translatePDTtoIsajet()`.

6.3.4.5 unsigned short HepPID::digit (location loc, const int & pid)

return the digit at a named location in the PID

Definition at line 27 of file ParticleIDMethods.cc.

References `abspid()`.

Referenced by `convIsajettoPDT()`, `convPDTtoIsajet()`, `fundamentalID()`, `hasBottom()`, `hasCharm()`, `hasDown()`, `hasStrange()`, `hasTop()`, `hasUp()`, `isBaryon()`, `isDiQuark()`, `isMeson()`, `isNucleus()`, `isPentaquark()`, `isRhadron()`, `isSUSY()`, `lambda()`, `lSpin()`, `main()`, `sSpin()`, and `threeCharge()`.

6.3.4.6 EvtGenPDTMap const& HepPID::@68::EvtGenPDTMapInit () [static]

Definition at line 41 of file translateEvtGen.cc.

Referenced by `getEvtGenPDTMap()`.

6.3.4.7 int HepPID::extraBits (const int & pid)

returns everything beyond the 7th digit (e.g. outside the standard numbering scheme)

Definition at line 21 of file ParticleIDMethods.cc.

References `abspid()`.

Referenced by `fundamentalID()`, `hasBottom()`, `hasCharm()`, `hasDown()`, `hasStrange()`, `hasTop()`, `hasUp()`, `isBaryon()`, `isDiQuark()`, `isHadron()`, `isLepton()`, `isMeson()`, `isPentaquark()`, `isRhadron()`, `isSUSY()`, `isValid()`, `jSpin()`, `main()`, and `threeCharge()`.

6.3.4.8 int HepPID::fundamentalID (const int & pid)

return the first two digits if this is a "fundamental" particle

Definition at line 37 of file ParticleIDMethods.cc.

References `abspid()`, `digit()`, `extraBits()`, `nq1`, and `nq2`.

Referenced by `convPDTtoIsajet()`, `hasBottom()`, `hasCharm()`, `hasDown()`, `hasFundamentalAnti()`, `hasStrange()`, `hasTop()`, `hasUp()`, `isBaryon()`, `isDiQuark()`, `isLepton()`, `isMeson()`, `isSUSY()`, `isValid()`, `jSpin()`, `main()`, and `threeCharge()`.

6.3.4.9 EvtGenPDTMap const& HepPID::@68::getEvtGenPDTMap () [static]

Definition at line 608 of file translateEvtGen.cc.

References `EvtGenPDTMapInit()`.

6.3.4.10 EvtGenPDTMap const& HepPID::@68::getEvtGenPDTMap () [static]

Referenced by `PDTEvtGenMapInit()`, `translateEvtGentoPDT()`, and `writeEvtGenTranslationLine()`.

6.3.4.11 HerwigPDTMap const& HepPID::@70::getHerwigPDTMap () [static]

Definition at line 500 of file translateHerwig.cc.

References `HerwigPDTMapInit()`.

6.3.4.12 HerwigPDTMap const& HepPID::@70::getHerwigPDTMap () [static]

Referenced by `PDTHerwigMapInit()`, `translateHerwigtoPDT()`, and `writeHerwigTranslationLine()`.

6.3.4.13 InverseQQbarMap const& HepPID::@75::getInverseQQbarMap () [static]

Definition at line 560 of file translateQQ.cc.

References `InverseQQbarMapInit()`.

6.3.4.14 InverseQQbarMap `const& HepPID::@75::getInverseQQbarMap ()` [static]

Referenced by `translateInverseQQbar()`.

6.3.4.15 IsajetPDTMap `const& HepPID::@71::getIsajetPDTMap ()` [static]

Definition at line 891 of file `translateIsajet.cc`.

References `IsajetPDTMapInit()`.

6.3.4.16 IsajetPDTMap `const& HepPID::@71::getIsajetPDTMap ()` [static]

Referenced by `PDTIsajetMapInit()`, `translateIsajettoPDT()`, and `writeIsajetTranslationLine()`.

6.3.4.17 ParticleNameMap `const & HepPID::getParticleNameMap ()`

access the `ParticleNameMap` for other purposes

Definition at line 1589 of file `ParticleName.cc`.

References `ParticleNameInit()`.

Referenced by `particleName()`, and `validParticleName()`.

6.3.4.18 PDGtoPDTMap `const& HepPID::@72::getPDGtoPDTMap ()` [static]

Definition at line 369 of file `translatePDG.cc`.

References `PDGtoPDTMapInit()`.

6.3.4.19 PDGtoPDTMap `const& HepPID::@72::getPDGtoPDTMap ()` [static]

Referenced by `PDTtoPDGMapInit()`, `translatePDGtabletoPDT()`, and `writePDGTranslationLine()`.

6.3.4.20 PDTEvtGenMap `const& HepPID::@68::getPDTEvtGenMap ()` [static]

Definition at line 617 of file `translateEvtGen.cc`.

References `PDTEvtGenMapInit()`.

6.3.4.21 PDTEvtGenMap `const& HepPID::@68::getPDTEvtGenMap ()` [static]

Referenced by `translatePDTtoEvtGen()`.

6.3.4.22 PDTHerwigMap `const& HepPID::@70::getPDTHerwigMap ()` [static]

Definition at line 509 of file `translateHerwig.cc`.

References `PDTHerwigMapInit()`.

6.3.4.23 PDTHerwigMap const& HepPID::@70::getPDTHerwigMap () [static]

Referenced by translatePDTtoHerwig().

6.3.4.24 PDTIsajetMap const& HepPID::@71::getPDTIsajetMap () [static]

Definition at line 900 of file translateIsajet.cc.

References PDTIsajetMapInit().

6.3.4.25 PDTIsajetMap const& HepPID::@71::getPDTIsajetMap () [static]

Referenced by translatePDTtoIsajet().

6.3.4.26 PDTPythiaMap const& HepPID::@74::getPDTPythiaMap () [static]

Definition at line 625 of file translatePythia.cc.

References PDTPythiaMapInit().

6.3.4.27 PDTPythiaMap const& HepPID::@74::getPDTPythiaMap () [static]

Referenced by translatePDTtoPythia().

6.3.4.28 PDTQQMap const& HepPID::@75::getPDTQQMap () [static]

Definition at line 543 of file translateQQ.cc.

References PDTQQMapInit().

6.3.4.29 PDTQQMap const& HepPID::@75::getPDTQQMap () [static]

Referenced by translatePDTtoQQ().

6.3.4.30 PDTtoPDGMap const& HepPID::@72::getPDTtoPDGMap () [static]

Definition at line 378 of file translatePDG.cc.

References PDTtoPDGMapInit().

6.3.4.31 PDTtoPDGMap const& HepPID::@72::getPDTtoPDGMap () [static]

Referenced by translatePDTtoPDGtable().

6.3.4.32 PythiaPDTMap const& HepPID::@74::getPythiaPDTMap () [static]

Definition at line 616 of file translatePythia.cc.

References PythiaPDTMapInit().

6.3.4.33 `PythiaPDTMap const& HepPID::@74::getPythiaPDTMap () [static]`

Referenced by `PDTPythiaMapInit()`, `translatePythiatoPDT()`, and `writePythiaTranslationLine()`.

6.3.4.34 `QQbarMap const& HepPID::@75::getQQbarMap () [static]`

Definition at line 551 of file `translateQQ.cc`.

References `QQbarMapInit()`.

6.3.4.35 `QQbarMap const& HepPID::@75::getQQbarMap () [static]`

Referenced by `InverseQQbarMapInit()`, and `translateQQbar()`.

6.3.4.36 `QQPDTMap const& HepPID::@75::getQQPDTMap () [static]`

Definition at line 534 of file `translateQQ.cc`.

References `QQPDTMapInit()`.

6.3.4.37 `QQPDTMap const& HepPID::@75::getQQPDTMap () [static]`

Referenced by `PDTQQMapInit()`, and `translateQQtoPDT()`.

6.3.4.38 `bool HepPID::hasBottom (const int & pid)`

does this particle contain a bottom quark?

Definition at line 297 of file `ParticleIDMethods.cc`.

References `digit()`, `extraBits()`, `fundamentalID()`, `nq1`, `nq2`, and `nq3`.

6.3.4.39 `bool HepPID::hasCharm (const int & pid)`

does this particle contain a charm quark?

Definition at line 289 of file `ParticleIDMethods.cc`.

References `digit()`, `extraBits()`, `fundamentalID()`, `nq1`, `nq2`, and `nq3`.

6.3.4.40 `bool HepPID::hasDown (const int & pid)`

does this particle contain a down quark?

Definition at line 273 of file `ParticleIDMethods.cc`.

References `digit()`, `extraBits()`, `fundamentalID()`, `nq1`, `nq2`, and `nq3`.

6.3.4.41 `bool HepPID::hasFundamentalAnti (const int & pid)`

if this is a fundamental particle, does it have a valid antiparticle?

Definition at line 112 of file ParticleIDMethods.cc.

References `fundamentalID()`, and `validParticleName()`.

Referenced by `isValid()`.

6.3.4.42 `bool HepPID::hasStrange (const int & pid)`

does this particle contain a strange quark?

Definition at line 281 of file ParticleIDMethods.cc.

References `digit()`, `extraBits()`, `fundamentalID()`, `nq1`, `nq2`, and `nq3`.

6.3.4.43 `bool HepPID::hasTop (const int & pid)`

does this particle contain a top quark?

Definition at line 305 of file ParticleIDMethods.cc.

References `digit()`, `extraBits()`, `fundamentalID()`, `nq1`, `nq2`, and `nq3`.

6.3.4.44 `bool HepPID::hasUp (const int & pid)`

does this particle contain an up quark?

Definition at line 265 of file ParticleIDMethods.cc.

References `digit()`, `extraBits()`, `fundamentalID()`, `nq1`, `nq2`, and `nq3`.

6.3.4.45 `HerwigPDTMap const& HepPID::@70::HerwigPDTMapInit () [static]`

Definition at line 41 of file `translateHerwig.cc`.

Referenced by `getHerwigPDTMap()`.

6.3.4.46 `InverseQQbarMap const& HepPID::@75::InverseQQbarMapInit () [static]`

Definition at line 520 of file `translateQQ.cc`.

References `getQQbarMap()`.

Referenced by `getInverseQQbarMap()`.

6.3.4.47 `IsajetPDTMap const& HepPID::@71::IsajetPDTMapInit () [static]`

Definition at line 44 of file `translateIsajet.cc`.

Referenced by `getIsajetPDTMap()`.

6.3.4.48 `bool HepPID::isBaryon (const int & pid)`

is this a valid baryon ID?

Definition at line 148 of file ParticleIDMethods.cc.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `nj`, `nq1`, `nq2`, and `nq3`.
Referenced by `isHadron()`, `isValid()`, `main()`, and `threeCharge()`.

6.3.4.49 `bool HepPID::isDiQuark (const int & pid)`

is this a valid diquark ID?

Definition at line 160 of file `ParticleIDMethods.cc`.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `nj`, `nq1`, `nq2`, and `nq3`.
Referenced by `isValid()`, `main()`, and `threeCharge()`.

6.3.4.50 `bool HepPID::isHadron (const int & pid)`

is this a valid hadron ID?

Definition at line 179 of file `ParticleIDMethods.cc`.

References `extraBits()`, `isBaryon()`, `isMeson()`, and `isPentaquark()`.
Referenced by `main()`.

6.3.4.51 `bool HepPID::isLepton (const int & pid)`

is this a valid lepton ID?

Definition at line 188 of file `ParticleIDMethods.cc`.

References `extraBits()`, and `fundamentalID()`.
Referenced by `main()`.

6.3.4.52 `bool HepPID::isMeson (const int & pid)`

is this a valid meson ID?

Definition at line 124 of file `ParticleIDMethods.cc`.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `nj`, `nq1`, `nq2`, and `nq3`.
Referenced by `isHadron()`, `isValid()`, `lSpin()`, `main()`, `sSpin()`, and `threeCharge()`.

6.3.4.53 `bool HepPID::isNucleus (const int & pid)`

is this a valid ion ID?

Definition at line 202 of file `ParticleIDMethods.cc`.

References `abspid()`, `digit()`, `n10`, and `n9`.
Referenced by `A()`, `isValid()`, `lambda()`, `main()`, and `Z()`.

6.3.4.54 `bool HepPID::isPentaquark (const int & pid)`

is this a valid pentaquark ID?

Definition at line 216 of file `ParticleIDMethods.cc`.

References `digit()`, `extraBits()`, `n`, `nj`, `nl`, `nq1`, `nq2`, `nq3`, and `nr`.

Referenced by `isHadron()`, and `isValid()`.

6.3.4.55 `bool HepPID::isRhadron (const int & pid)`

is this a valid R-hadron ID?

Definition at line 248 of file `ParticleIDMethods.cc`.

References `digit()`, `extraBits()`, `isSUSY()`, `n`, `nj`, `nq2`, `nq3`, and `nr`.

Referenced by `isValid()`.

6.3.4.56 `bool HepPID::isSUSY (const int & pid)`

is this a valid SUSY ID?

Definition at line 236 of file `ParticleIDMethods.cc`.

References `digit()`, `extraBits()`, `fundamentalID()`, `n`, and `nr`.

Referenced by `isRhadron()`, and `isValid()`.

6.3.4.57 `bool HepPID::isValid (const int & pid)`

is this a valid ID?

Examples:

`examListPythia.cc`.

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

References `extraBits()`, `fundamentalID()`, `hasFundamentalAnti()`, `isBaryon()`, `isDiQuark()`, `isMeson()`, `isNucleus()`, `isPentaquark()`, `isRhadron()`, and `isSUSY()`.

Referenced by `main()`, `translateEvtGentoPDT()`, `translateHerwigtoPDT()`, `translatePDGtabletoPDT()`, `translatePDTtoEvtGen()`, `translatePDTtoHerwig()`, `translatePDTtoPDGtable()`, `translatePDTtoPythia()`, and `translatePythiatoPDT()`.

6.3.4.58 `int HepPID::jSpin (const int & pid)`

`jSpin` returns $2J+1$, where J is the total spin

Definition at line 316 of file `ParticleIDMethods.cc`.

References `abspid()`, `extraBits()`, and `fundamentalID()`.

Referenced by `main()`.

6.3.4.59 `int HepPID::lambda (const int & pid)`

if this is a nucleus (ion), get `nLambda` Ion numbers are +/- 10LZZZAAAI.

Definition at line 69 of file `ParticleIDMethods.cc`.

References `abspid()`, `digit()`, `isNucleus()`, and `n8`.

Referenced by `main()`.

6.3.4.60 `void HepPID::listParticleNames (std::ostream & os)`

list all known names

Examples:

`listParticleNames.cc`.

Definition at line 1618 of file `ParticleName.cc`.

References `n`, `writeParticleNameLine()`, and `writeVersion()`.

Referenced by `main()`.

6.3.4.61 `int HepPID::lSpin (const int & pid)`

`lSpin` returns $2L+1$, where L is the orbital angular momentum

Definition at line 357 of file `ParticleIDMethods.cc`.

References `digit()`, `isMeson()`, `n`, `nj`, and `nl`.

Referenced by `main()`.

6.3.4.62 `std::string HepPID::particleName (const int &)`

get a known **HepPID** (p.40) Particle name

Examples:

`examListHerwig.cc`, `examListIsajet.cc`, and `examListPythia.cc`.

Definition at line 1605 of file `ParticleName.cc`.

References `getParticleNameMap()`.

Referenced by `main()`, `HepPDT::detail::parseParticleLine()`, `HepPDT::ParticleID::PDName()`, `writeEvtGenTranslationLine()`, `writeHerwigTranslationLine()`, `writeIsajetTranslationLine()`, `writeParticleNameLine()`, `writePDGTranslationLine()`, `writePythiaTranslationLine()`, and `writeQQTranslation()`.

6.3.4.63 `ParticleNameMap const& HepPID::@67::ParticleNameInit () [static]`

Definition at line 36 of file `ParticleName.cc`.

Referenced by `getParticleNameMap()`.

6.3.4.64 `PDGtoPDTMap const& HepPID::@72::PDGtoPDTMapInit () [static]`

Definition at line 41 of file `translatePDG.cc`.

Referenced by `getPDGtoPDTMap()`.

6.3.4.65 PDTEvtGenMap const& HepPID::@68::PDTEvtGenMapInit () [static]

Definition at line 594 of file translateEvtGen.cc.

References `getEvtGenPDTMap()`.

Referenced by `getPDTEvtGenMap()`.

6.3.4.66 PDTHerwigMap const& HepPID::@70::PDTHerwigMapInit () [static]

Definition at line 486 of file translateHerwig.cc.

References `getHerwigPDTMap()`.

Referenced by `getPDTHerwigMap()`.

6.3.4.67 PDTIsajetMap const& HepPID::@71::PDTIsajetMapInit () [static]

Definition at line 677 of file translateIsajet.cc.

References `getIsajetPDTMap()`.

Referenced by `getPDTIsajetMap()`.

6.3.4.68 PDTPythiaMap const& HepPID::@74::PDTPythiaMapInit () [static]

Definition at line 602 of file translatePythia.cc.

References `getPythiaPDTMap()`.

Referenced by `getPDTPythiaMap()`.

6.3.4.69 PDTQQMap const& HepPID::@75::PDTQQMapInit () [static]

Definition at line 509 of file translateQQ.cc.

References `getQQPDTMap()`.

Referenced by `getPDTQQMap()`.

6.3.4.70 PDTtoPDGMap const& HepPID::@72::PDTtoPDGMapInit () [static]

Definition at line 355 of file translatePDG.cc.

References `getPDGtoPDTMap()`.

Referenced by `getPDTtoPDGMap()`.

6.3.4.71 PythiaPDTMap const& HepPID::@74::PythiaPDTMapInit () [static]

Definition at line 41 of file translatePythia.cc.

Referenced by `getPythiaPDTMap()`.

6.3.4.72 `QQbarMap const& HepPID::@75::QQbarMapInit () [static]`

Definition at line 455 of file translateQQ.cc.

Referenced by `getQQbarMap()`.

6.3.4.73 `QQPDTMap const& HepPID::@75::QQPDTMapInit () [static]`

Definition at line 49 of file translateQQ.cc.

Referenced by `getQQPDTMap()`.

6.3.4.74 `int HepPID::sSpin (const int & pid)`

`sSpin` returns $2S+1$, where S is the spin

Definition at line 332 of file ParticleIDMethods.cc.

References `digit()`, `isMeson()`, `n`, `nj`, and `nl`.

Referenced by `main()`.

6.3.4.75 `int HepPID::threeCharge (const int & pid)`

return 3 times the charge (3 x quark charge is an int)

Definition at line 406 of file ParticleIDMethods.cc.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `isBaryon()`, `isDiQuark()`, `isMeson()`, `nj`, `nq1`, `nq2`, and `nq3`.

Referenced by `main()`.

6.3.4.76 `int HepPID::translateEvtGenToPDT (const int evtGenID)`

translate EvtGen to PDG standard

Definition at line 625 of file translateEvtGen.cc.

References `getEvtGenPDTMap()`, and `isValid()`.

Referenced by `HepPDT::addEvtGenParticles()`.

6.3.4.77 `int HepPID::translateGeanttoPDT (const int geantID)`

translate Geant3 to PDG standard

Definition at line 20 of file translateGeanttoPDT.cc.

References `IDMAX`.

6.3.4.78 `int HepPID::translateHerwigtoPDT (const int herwigID)`

translate Herwig to PDG standard

Examples:**examListHerwig.cc.**

Definition at line 517 of file translateHerwig.cc.

References getHerwigPDTMap(), and isValid().

Referenced by main().

6.3.4.79 int HepPID::translateInverseQQbar (const int *id*)

QQ helper function.

Definition at line 579 of file translateQQ.cc.

References getInverseQQbarMap().

Referenced by writeQQTranslation().

6.3.4.80 int HepPID::translateIsajettoPDT (const int *isajetID*)

translate Isajet to PDG standard

Examples:**examListIsajet.cc.**

Definition at line 908 of file translateIsajet.cc.

References convIsajettoPDT(), and getIsajetPDTMap().

Referenced by HepPDT::addIsajetDecay(), HepPDT::addIsajetParticles(), main(), and HepPDT::detail::parseIsajetDecayLine().

6.3.4.81 int HepPID::translatePDGtabletoPDT (const int *pdgID*)

translate PDG table to PDG standard

Definition at line 386 of file translatePDG.cc.

References getPDGtoPDTMap(), and isValid().

6.3.4.82 int HepPID::translatePDTtoEvtGen (const int *pid*)

translate PDG standard to EvtGen

Definition at line 638 of file translateEvtGen.cc.

References getPDTEvtGenMap(), and isValid().

Referenced by writeEvtGenTranslationLine().

6.3.4.83 int HepPID::translatePDTtoGeant (const int *pid*)

translate PDG standard to Geant3

Definition at line 22 of file translatePDTtoGeant.cc.

References IDMAX.

6.3.4.84 `int HepPID::translatePDTtoHerwig (const int pid)`

translate PDG standard to Herwig

Definition at line 530 of file `translateHerwig.cc`.

References `getPDTHerwigMap()`, and `isValid()`.

Referenced by `writeHerwigTranslationLine()`.

6.3.4.85 `int HepPID::translatePDTtoIsajet (const int pid)`

translate PDG standard to Isajet

Definition at line 919 of file `translateIsajet.cc`.

References `convPDTtoIsajet()`, and `getPDTIsajetMap()`.

Referenced by `writeIsajetTranslationLine()`.

6.3.4.86 `int HepPID::translatePDTtoPDGtable (const int pid)`

translate PDG standard to PDG table

Definition at line 399 of file `translatePDG.cc`.

References `getPDTtoPDGMap()`, and `isValid()`.

Referenced by `writePDGTranslationLine()`.

6.3.4.87 `int HepPID::translatePDTtoPythia (const int pid)`

translate PDG standard to Pythia

Definition at line 646 of file `translatePythia.cc`.

References `getPDTPythiaMap()`, and `isValid()`.

Referenced by `writePythiaTranslationLine()`.

6.3.4.88 `int HepPID::translatePDTtoQQ (const int pid)`

translate PDG standard to QQ

Definition at line 601 of file `translateQQ.cc`.

References `getPDTQQMap()`.

Referenced by `writeQQTranslation()`.

6.3.4.89 `int HepPID::translatePythiatoPDT (const int pythiaID)`

translate Pythia to PDG standard

Examples:

`examListPythia.cc`.

Definition at line 633 of file translatePythia.cc.

References getPythiaPDTMap(), and isValid().

Referenced by HepPDT::addPythiaParticles(), and main().

6.3.4.90 int HepPID::translateQQbar (const int *id*)

QQ helper function.

Definition at line 568 of file translateQQ.cc.

References getQQbarMap().

Referenced by HepPDT::addQQParticles(), and writeQQTranslation().

6.3.4.91 int HepPID::translateQQtoPDT (const int *qqID*)

translate QQ to PDG standard

Definition at line 590 of file translateQQ.cc.

References getQQPDTMap().

Referenced by HepPDT::addQQParticles(), and writeQQTranslation().

6.3.4.92 bool HepPID::validParticleName (const int &)

verify that this number has a valid name

Definition at line 1595 of file ParticleName.cc.

References getParticleNameMap().

Referenced by hasFundamentalAnti(), and writeParticleNameLine().

6.3.4.93 void HepPID::version ()

print **HepPID** (p. 40) version

Definition at line 19 of file HepPID/Version.cc.

References versionName().

6.3.4.94 std::string HepPID::versionName ()

return **HepPID** (p. 40) version

Definition at line 14 of file HepPID/Version.cc.

Referenced by version(), and writeVersion().

6.3.4.95 void HepPID::writeEvtGenTranslation (std::ostream & *os*)

output the translation list

Examples:

listEvtGenTranslation.cc.

Definition at line 675 of file translateEvtGen.cc.

References writeEvtGenTranslationLine(), and writeVersion().

Referenced by main().

6.3.4.96 void HepPID::writeEvtGenTranslationLine (int *i*, std::ostream & *os*)

Definition at line 651 of file translateEvtGen.cc.

References getEvtGenPDTMap(), particleName(), and translatePDTtoEvtGen().

Referenced by writeEvtGenTranslation().

6.3.4.97 void HepPID::writeHerwigTranslation (std::ostream & *os*)

output the translation list

Examples:

listHerwigTranslation.cc.

Definition at line 567 of file translateHerwig.cc.

References n, writeHerwigTranslationLine(), and writeVersion().

Referenced by main().

6.3.4.98 void HepPID::writeHerwigTranslationLine (int *i*, std::ostream & *os*)

Definition at line 543 of file translateHerwig.cc.

References getHerwigPDTMap(), particleName(), and translatePDTtoHerwig().

Referenced by writeHerwigTranslation().

6.3.4.99 void HepPID::writeIsajetTranslation (std::ostream & *os*)

output the translation list

Examples:

listIsajetTranslation.cc.

Definition at line 954 of file translateIsajet.cc.

References writeIsajetTranslationLine(), and writeVersion().

Referenced by main().

6.3.4.100 void HepPID::writeIsajetTranslationLine (int *i*, std::ostream & *os*)

Definition at line 930 of file translateIsajet.cc.

References getIsajetPDTMap(), particleName(), and translatePDTtoIsajet().

Referenced by writeIsajetTranslation().

6.3.4.101 void HepPID::@67::writeParticleNameLine (int *i*, std::ostream & *os*)
[static]

Definition at line 1572 of file ParticleName.cc.

References `particleName()`, and `validParticleName()`.

Referenced by `listParticleNames()`.

6.3.4.102 void HepPID::writePDGTranslation (std::ostream & *os*)

output the translation list

Examples:

`listPDGTranslation.cc`.

Definition at line 436 of file `translatePDG.cc`.

References `writePDGTranslationLine()`, and `writeVersion()`.

Referenced by `main()`.

6.3.4.103 void HepPID::writePDGTranslationLine (int *i*, std::ostream & *os*)

Definition at line 412 of file `translatePDG.cc`.

References `getPDGtoPDTMap()`, `particleName()`, and `translatePDTtoPDGtable()`.

Referenced by `writePDGTranslation()`.

6.3.4.104 void HepPID::writePythiaTranslation (std::ostream & *os*)

output the translation list

Examples:

`listPythiaTranslation.cc`.

Definition at line 683 of file `translatePythia.cc`.

References `n`, `writePythiaTranslationLine()`, and `writeVersion()`.

Referenced by `main()`.

6.3.4.105 void HepPID::writePythiaTranslationLine (int *i*, std::ostream & *os*)

Definition at line 659 of file `translatePythia.cc`.

References `getPythiaPDTMap()`, `particleName()`, and `translatePDTtoPythia()`.

Referenced by `writePythiaTranslation()`.

6.3.4.106 void HepPID::writeQQTranslation (std::ostream & *os*)

output the translation list

Examples:

listQQTranslation.cc.

Definition at line 612 of file translateQQ.cc.

References `particleName()`, `translateInverseQQbar()`, `translatePDTtoQQ()`, `translateQQbar()`, `translateQQtoPDT()`, and `writeVersion()`.

Referenced by `main()`.

6.3.4.107 void HepPID::writeVersion (std::ostream & os)

write **HepPID** (p. 40) version to os

Examples:

examListHerwig.cc, **examListIsajet.cc**, and **examListPythia.cc**.

Definition at line 25 of file HepPID/Version.cc.

References `versionName()`.

Referenced by `listParticleNames()`, `main()`, `writeEvtGenTranslation()`, `writeHerwigTranslation()`, `writeIsajetTranslation()`, `writePDGTranslation()`, `writePythiaTranslation()`, and `writeQQTranslation()`.

6.3.4.108 int HepPID::Z (const int & pid)

if this is a nucleus (ion), get Z Ion numbers are +/- 10LZZZAAAI.

Definition at line 50 of file ParticleIDMethods.cc.

References `abspid()`, and `isNucleus()`.

Referenced by `main()`.

6.4 std Namespace Reference

Chapter 7

HepPDT Class Documentation

7.1 HepPDT::CommonParticleData Class Reference

The **CommonParticleData** (p. 65) class holds basic particle data.

```
#include <CommonParticleData.hh>
```

Public Member Functions

- **CommonParticleData** (const **TempParticleData** &tpd)
*create **CommonParticleData** (p. 65) from the temporary information*
- **~CommonParticleData** ()
- void **swap** (**CommonParticleData** &rhs)
- **CommonParticleData** (const **CommonParticleData** &orig)
- **CommonParticleData** & **operator=** (const **CommonParticleData** &rhs)
- const std::string & **name** () const
return particle name as defined by user input
- const std::string **PDName** () const
return PDG particle name
- const std::string & **source** () const
the name of the input source
- **ParticleID** **ID** () const
*return the **ParticleID** (p. 109)*
- int **pid** () const
return the integer ID
- int **originalID** () const
return untranslated integer ID
- double **charge** () const

return charge

- double **color** () const
color information
- **SpinState spin** () const
spin information
- **Measurement mass** () const
mass
- **Measurement totalWidth** () const
return the total width
- double **lowerCutoff** () const
lower cutoff of allowed width values
- double **upperCutoff** () const
upper cutoff of allowed width values
- **Measurement lifetime** () const
calculate the lifetime
- int **numConstituents** () const
number of constituent particles (e.g., quarks)
- **Constituent constituent** (unsigned int i) const
constituent information
- **ParticleID constituentParticle** (unsigned int i) const
***ParticleID** (p. 109) for a constituent particle.*
- **ResonanceStructure** const **resonance** () const
resonance (width) information
- void **write** (std::ostream &os) const
output information about this particle
- void **writeTranslation** (std::ostream &os) const
output the translation information for this particle
- bool **isMeson** () const
is this a valid meson?
- bool **isBaryon** () const
is this a valid baryon?
- bool **isDiQuark** () const
is this a valid diquark?

- **bool isHadron () const**
is this a valid hadron?
- **bool isLepton () const**
is this a valid lepton?
- **bool isNucleus () const**
is this a valid ion?
- **bool hasUp () const**
does this particle contain an up quark?
- **bool hasDown () const**
does this particle contain a down quark?
- **bool hasStrange () const**
does this particle contain a strange quark?
- **bool hasCharm () const**
does this particle contain a charm quark?
- **bool hasBottom () const**
does this particle contain a bottom quark?
- **bool hasTop () const**
does this particle contain a top quark?
- **bool operator< (const CommonParticleData &other) const**
compare masses
- **bool operator== (const CommonParticleData &other) const**
use PID and ignore everything else
- **void setCharge (double chg)**
change the charge
- **void setColor (double col)**
change color information
- **void setSpin (const SpinState &spin)**
change spin information
- **void addConstituent (Constituent c)**
add a constituent particle
- **void setMass (Measurement const &mass)**
change the mass
- **void setTotalWidth (Measurement const &width)**
change the total width

- void **setTotalWidthFromLifetime** (**Measurement** const <)
change the total width using a lifetime
- void **setLowerCutoff** (double cut)
change the total width lower cutoff
- void **setUpperCutoff** (double cut)
change the total width upper cutoff

7.1.1 Detailed Description

The **CommonParticleData** (p. 65) class holds basic particle data.

Author:

Lynn Garren

This class holds the basic particle data - name, ID, spin, etc. Methods are provided to get various information about the particle.

Definition at line 31 of file CommonParticleData.hh.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 **HepPDT::CommonParticleData::CommonParticleData** (const **TempParticleData** & *tpd*) [inline]

create **CommonParticleData** (p. 65) from the temporary information

Definition at line 12 of file CommonParticleData.icc.

7.1.2.2 **HepPDT::CommonParticleData::~~CommonParticleData** () [inline]

Definition at line 26 of file CommonParticleData.icc.

7.1.2.3 **HepPDT::CommonParticleData::CommonParticleData** (const **CommonParticleData** & *orig*) [inline]

Definition at line 42 of file CommonParticleData.icc.

7.1.3 Member Function Documentation

7.1.3.1 **void HepPDT::CommonParticleData::swap** (**CommonParticleData** & *rhs*) [inline]

Definition at line 29 of file CommonParticleData.icc.

References `itsCharge`, `itsColorCharge`, `itsID`, `itsOriginalID`, `itsParticleName`, `itsQuarks`, `itsResonance`, `itsSource`, `itsSpin`, `HepPDT::SpinState::swap()`, `HepPDT::ParticleID::swap()`, and `HepPDT::swap()`.

Referenced by `operator=()`, and `HepPDT::swap()`.

7.1.3.2 `CommonParticleData & HepPDT::CommonParticleData::operator= (const CommonParticleData & rhs)` [inline]

Definition at line 54 of file `CommonParticleData.icc`.

References `swap()`.

7.1.3.3 `const std::string& HepPDT::CommonParticleData::name () const` [inline]

return particle name as defined by user input

Definition at line 50 of file `CommonParticleData.hh`.

7.1.3.4 `const std::string HepPDT::CommonParticleData::PDName () const` [inline]

return PDG particle name

Definition at line 52 of file `CommonParticleData.hh`.

References `HepPDT::ParticleID::PDName()`.

Referenced by `writeTranslation()`.

7.1.3.5 `const std::string& HepPDT::CommonParticleData::source () const` [inline]

the name of the input source

Definition at line 54 of file `CommonParticleData.hh`.

7.1.3.6 `ParticleID HepPDT::CommonParticleData::ID () const` [inline]

return the **ParticleID** (p. 109)

Definition at line 56 of file `CommonParticleData.hh`.

7.1.3.7 `int HepPDT::CommonParticleData::pid () const` [inline]

return the integer ID

Definition at line 58 of file `CommonParticleData.hh`.

References `HepPDT::ParticleID::pid()`.

Referenced by `writeTranslation()`.

7.1.3.8 `int HepPDT::CommonParticleData::originalID () const` [inline]

return untranslated integer ID

Definition at line 60 of file `CommonParticleData.hh`.

7.1.3.9 double HepPDT::CommonParticleData::charge () const [inline]

return charge

Definition at line 62 of file CommonParticleData.hh.

7.1.3.10 double HepPDT::CommonParticleData::color () const [inline]

color information

Definition at line 64 of file CommonParticleData.hh.

7.1.3.11 SpinState HepPDT::CommonParticleData::spin () const [inline]

spin information

Definition at line 66 of file CommonParticleData.hh.

Referenced by setSpin().

7.1.3.12 Measurement HepPDT::CommonParticleData::mass () const [inline]

mass

Definition at line 68 of file CommonParticleData.hh.

References HepPDT::ResonanceStructure::mass().

Referenced by operator<(), setMass(), and write().

7.1.3.13 Measurement HepPDT::CommonParticleData::totalWidth () const [inline]

return the total width

Definition at line 70 of file CommonParticleData.hh.

References HepPDT::ResonanceStructure::totalWidth().

7.1.3.14 double HepPDT::CommonParticleData::lowerCutoff () const [inline]

lower cutoff of allowed width values

Definition at line 72 of file CommonParticleData.hh.

References HepPDT::ResonanceStructure::lowerCutoff().

Referenced by write().

7.1.3.15 double HepPDT::CommonParticleData::upperCutoff () const [inline]

upper cutoff of allowed width values

Definition at line 74 of file CommonParticleData.hh.

References HepPDT::ResonanceStructure::upperCutoff().

Referenced by write().

7.1.3.16 Measurement HepPDT::CommonParticleData::lifetime () const [inline]

calculate the lifetime

Definition at line 76 of file CommonParticleData.hh.

References HepPDT::ResonanceStructure::lifetime().

Referenced by write().

7.1.3.17 int HepPDT::CommonParticleData::numConstituents () const [inline]

number of constituent particles (e.g., quarks)

Definition at line 78 of file CommonParticleData.hh.

7.1.3.18 Constituent HepPDT::CommonParticleData::constituent (unsigned int *i*) const [inline]

constituent information

return this constituent if it exists

Definition at line 91 of file CommonParticleData.icc.

7.1.3.19 ParticleID HepPDT::CommonParticleData::constituentParticle (unsigned int *i*) const [inline]

ParticleID (p. 109) for a constituent particle.

Definition at line 102 of file CommonParticleData.icc.

7.1.3.20 ResonanceStructure const HepPDT::CommonParticleData::resonance () const [inline]

resonance (width) information

Definition at line 84 of file CommonParticleData.hh.

7.1.3.21 void HepPDT::CommonParticleData::write (std::ostream & *os*) const

output information about this particle

Definition at line 18 of file write.cc.

References HepPDT::ParticleID::isValid(), lifetime(), lowerCutoff(), HepPDT::ResonanceStructure::mass(), mass(), HepPDT::SpinState::orbAngMom(), HepPDT::ParticleID::pid(), HepPDT::Measurement::sigma(), HepPDT::SpinState::spin(), HepPDT::SpinState::totalSpin(), HepPDT::ResonanceStructure::totalWidth(), upperCutoff(), and HepPDT::Measurement::value().

7.1.3.22 void HepPDT::CommonParticleData::writeTranslation (std::ostream & *os*) const

output the translation information for this particle

Definition at line 68 of file write.cc.

References `PDName()`, and `pid()`.

7.1.3.23 `bool HepPDT::CommonParticleData::isMeson () const [inline]`

is this a valid meson?

Definition at line 94 of file `CommonParticleData.hh`.

References `HepPDT::ParticleID::isMeson()`.

7.1.3.24 `bool HepPDT::CommonParticleData::isBaryon () const [inline]`

is this a valid baryon?

Definition at line 96 of file `CommonParticleData.hh`.

References `HepPDT::ParticleID::isBaryon()`.

7.1.3.25 `bool HepPDT::CommonParticleData::isDiQuark () const [inline]`

is this a valid diquark?

Definition at line 98 of file `CommonParticleData.hh`.

References `HepPDT::ParticleID::isDiQuark()`.

7.1.3.26 `bool HepPDT::CommonParticleData::isHadron () const [inline]`

is this a valid hadron?

Definition at line 100 of file `CommonParticleData.hh`.

References `HepPDT::ParticleID::isHadron()`.

7.1.3.27 `bool HepPDT::CommonParticleData::isLepton () const [inline]`

is this a valid lepton?

Definition at line 102 of file `CommonParticleData.hh`.

References `HepPDT::ParticleID::isLepton()`.

7.1.3.28 `bool HepPDT::CommonParticleData::isNucleus () const [inline]`

is this a valid ion?

Definition at line 104 of file `CommonParticleData.hh`.

References `HepPDT::ParticleID::isNucleus()`.

7.1.3.29 `bool HepPDT::CommonParticleData::hasUp () const`

does this particle contain an up quark?

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

7.1.3.30 bool HepPDT::CommonParticleData::hasDown () const

does this particle contain a down quark?

Definition at line 26 of file hasMethods.cc.

7.1.3.31 bool HepPDT::CommonParticleData::hasStrange () const

does this particle contain a strange quark?

Definition at line 36 of file hasMethods.cc.

7.1.3.32 bool HepPDT::CommonParticleData::hasCharm () const

does this particle contain a charm quark?

Definition at line 46 of file hasMethods.cc.

7.1.3.33 bool HepPDT::CommonParticleData::hasBottom () const

does this particle contain a bottom quark?

Definition at line 56 of file hasMethods.cc.

7.1.3.34 bool HepPDT::CommonParticleData::hasTop () const

does this particle contain a top quark?

Definition at line 66 of file hasMethods.cc.

7.1.3.35 bool HepPDT::CommonParticleData::operator< (const CommonParticleData & other) const [inline]

compare masses

Definition at line 61 of file CommonParticleData.icc.

References mass().

7.1.3.36 bool HepPDT::CommonParticleData::operator== (const CommonParticleData & other) const [inline]

use PID and ignore everything else

Definition at line 66 of file CommonParticleData.icc.

References itsID.

7.1.3.37 void HepPDT::CommonParticleData::setCharge (double chg) [inline]

change the charge

Definition at line 126 of file CommonParticleData.hh.

7.1.3.38 void HepPDT::CommonParticleData::setColor (double *col*) [inline]

change color information

Definition at line 128 of file CommonParticleData.hh.

7.1.3.39 void HepPDT::CommonParticleData::setSpin (const SpinState & *spin*) [inline]

change spin information

Definition at line 130 of file CommonParticleData.hh.

References spin().

7.1.3.40 void HepPDT::CommonParticleData::addConstituent (Constituent *c*) [inline]

add a constituent particle

Definition at line 132 of file CommonParticleData.hh.

7.1.3.41 void HepPDT::CommonParticleData::setMass (Measurement const & *mass*) [inline]

change the mass

Definition at line 134 of file CommonParticleData.hh.

References mass(), and HepPDT::ResonanceStructure::setMass().

7.1.3.42 void HepPDT::CommonParticleData::setTotalWidth (Measurement const & *width*) [inline]

change the total width

Definition at line 136 of file CommonParticleData.hh.

References HepPDT::ResonanceStructure::setTotalWidth().

7.1.3.43 void HepPDT::CommonParticleData::setTotalWidthFromLifetime (Measurement const & *lt*) [inline]

change the total width using a lifetime

Definition at line 138 of file CommonParticleData.hh.

References HepPDT::ResonanceStructure::setTotalWidthFromLifetime().

7.1.3.44 void HepPDT::CommonParticleData::setLowerCutoff (double *cut*) [inline]

change the total width lower cutoff

Definition at line 140 of file CommonParticleData.hh.

References HepPDT::ResonanceStructure::setLowerCutoff().

7.1.3.45 void HepPDT::CommonParticleData::setUpperCutoff (double *cut*)
[inline]

change the total width upper cutoff

Definition at line 142 of file CommonParticleData.hh.

References HepPDT::ResonanceStructure::setUpperCutoff().

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

- **CommonParticleData.hh**
- **CommonParticleData.icc**
- **hasMethods.cc**
- **write.cc**

7.2 HepPDT::Constituent Class Reference

The **Constituent** (p. 76) class has information about constituent particles.

```
#include <Constituent.hh>
```

Public Member Functions

- **Constituent** (**ParticleID** p=**ParticleID**(0), int m=-1)
- **Constituent** (**Constituent** const &orig)
- **Constituent** & **operator=** (**Constituent** const &rhs)
- void **swap** (**Constituent** &other)
- int **multiplicity** () const
how many of this constituent are there?
- **ParticleID** **pid** () const
ParticleID (p. 109) of this constituent.
- bool **isUp** () const
is this an up quark?
- bool **isDown** () const
is this a down quark?
- bool **isStrange** () const
is this a strange quark?
- bool **isCharm** () const
is this a charm quark?
- bool **isBottom** () const
is this a bottom quark?
- bool **isTop** () const
is this a top quark?

7.2.1 Detailed Description

The **Constituent** (p. 76) class has information about constituent particles.

Author:

Lynn Garren

Holds a particle constituent (e.g. **ParticleID** (p. 109) of the quark type)

Definition at line 25 of file Constituent.hh.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 HepPDT::Constituent::Constituent (ParticleID *p* = ParticleID(0), int *m* = -1) [inline]

Definition at line 30 of file Constituent.hh.

7.2.2.2 HepPDT::Constituent::Constituent (Constituent const & *orig*) [inline]

Definition at line 35 of file Constituent.hh.

7.2.3 Member Function Documentation

7.2.3.1 Constituent& HepPDT::Constituent::operator= (Constituent const & *rhs*) [inline]

Definition at line 37 of file Constituent.hh.

References swap().

7.2.3.2 void HepPDT::Constituent::swap (Constituent & *other*) [inline]

Definition at line 42 of file Constituent.hh.

References itsMultiplicity, itsPid, and HepPDT::swap().

Referenced by operator=(), and HepPDT::swap().

7.2.3.3 int HepPDT::Constituent::multiplicity () const [inline]

how many of this constituent are there?

Definition at line 49 of file Constituent.hh.

7.2.3.4 ParticleID HepPDT::Constituent::pid () const [inline]

ParticleID (p. 109) of this constituent.

Definition at line 51 of file Constituent.hh.

7.2.3.5 bool HepPDT::Constituent::isUp () const

is this an up quark?

Definition at line 12 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

7.2.3.6 bool HepPDT::Constituent::isDown () const

is this a down quark?

Definition at line 18 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

7.2.3.7 bool HepPDT::Constituent::isStrange () const

is this a strange quark?

Definition at line 24 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

7.2.3.8 bool HepPDT::Constituent::isCharm () const

is this a charm quark?

Definition at line 30 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

7.2.3.9 bool HepPDT::Constituent::isBottom () const

is this a bottom quark?

Definition at line 36 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

7.2.3.10 bool HepPDT::Constituent::isTop () const

is this a top quark?

Definition at line 42 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

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

- **Constituent.hh**
- **Constituent.cc**

7.3 HepPDT::DecayChannel Class Reference

The **DecayChannel** (p. 79) class holds particle decay information.

```
#include <DecayChannel.hh>
```

Public Types

- typedef std::vector< **ParticleData** * > **ProductMap**

Public Member Functions

- **DecayChannel** (const std::string &name="unknown", const **Measurement** &branchFrac=**Measurement**(), const std::vector< **ParticleData** * > &decaylist=**ProductMap**(), const std::vector< double > &dmparam=std::vector< double >())
defaults 'cause we need vector<DecayChannel<> >
- virtual ~**DecayChannel** ()
- **DecayChannel** (const **DecayChannel** &orig)
- **DecayChannel** & **operator=** (const **DecayChannel** &rhs)
- void **swap** (**DecayChannel** &other)
- std::string **decayName** () const
get the name of this decay channel
- **Measurement** **branchingFraction** () const
return the branching fraction
- int **size** () const
how many decay products in this decay channel?
- **ParticleData** * **decayProduct** (int i) const
return this decay product
- void **write** (std::ostream &os) const
output decay information
- void **setBranchingFraction** (**Measurement** const &rate)
change the branching fraction
- void **appendDecayChannelProduct** (**ParticleData** *const &dcp)
append a new decay product

7.3.1 Detailed Description

The **DecayChannel** (p. 79) class holds particle decay information.

Author:

Lynn Garren

This class holds information about a single decay channel.

Definition at line 28 of file DecayChannel.hh.

7.3.2 Member Typedef Documentation

7.3.2.1 `typedef std::vector< ParticleData* > HepPDT::DecayChannel::ProductMap`

Definition at line 34 of file DecayChannel.hh.

7.3.3 Constructor & Destructor Documentation

7.3.3.1 `HepPDT::DecayChannel::DecayChannel (const std::string & name = "unknown", const Measurement & branchFrac = Measurement(), const std::vector< ParticleData * > & decaylist = ProductMap(), const std::vector< double > & dmparam = std::vector<double>()) [inline]`

defaults 'cause we need vector<DecayChannel<> >

Definition at line 39 of file DecayChannel.hh.

7.3.3.2 `virtual HepPDT::DecayChannel::~~DecayChannel () [inline, virtual]`

Definition at line 49 of file DecayChannel.hh.

7.3.3.3 `HepPDT::DecayChannel::DecayChannel (const DecayChannel & orig) [inline]`

Definition at line 11 of file DecayChannel.icc.

7.3.4 Member Function Documentation

7.3.4.1 `DecayChannel & HepPDT::DecayChannel::operator= (const DecayChannel & rhs) [inline]`

Definition at line 18 of file DecayChannel.icc.

References itsBranchingFraction, itsDecayChannelProducts, itsDecayModelParameters, and itsDecayName.

7.3.4.2 `void HepPDT::DecayChannel::swap (DecayChannel & other) [inline]`

Definition at line 30 of file DecayChannel.icc.

References itsBranchingFraction, itsDecayChannelProducts, itsDecayModelParameters, itsDecayName, and HepPDT::swap().

Referenced by HepPDT::swap().

7.3.4.3 `std::string HepPDT::DecayChannel::decayName () const [inline]`

get the name of this decay channel

Definition at line 61 of file DecayChannel.hh.

7.3.4.4 `Measurement HepPDT::DecayChannel::branchingFraction () const [inline]`

return the branching fraction

Definition at line 63 of file DecayChannel.hh.

7.3.4.5 `int HepPDT::DecayChannel::size () const [inline]`

how many decay products in this decay channel?

Definition at line 65 of file DecayChannel.hh.

7.3.4.6 `ParticleData* HepPDT::DecayChannel::decayProduct (int i) const [inline]`

return this decay product

Definition at line 67 of file DecayChannel.hh.

7.3.4.7 `void HepPDT::DecayChannel::write (std::ostream & os) const`

output decay information

Definition at line 21 of file DecayChannel.cc.

7.3.4.8 `void HepPDT::DecayChannel::setBranchingFraction (Measurement const & rate) [inline]`

change the branching fraction

Definition at line 75 of file DecayChannel.hh.

7.3.4.9 `void HepPDT::DecayChannel::appendDecayChannelProduct (ParticleData *const & dcp)`

append a new decay product

Definition at line 16 of file DecayChannel.cc.

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

- DecayChannel.hh
- DecayChannel.icc
- DecayChannel.cc

7.4 HepPDT::DecayData Class Reference

The **DecayData** (p. 82) class holds particle decay information.

```
#include <DecayData.hh>
```

Public Member Functions

- **DecayData** ()
default constructor
- **DecayData** (std::vector< **DecayChannel** > const &dv)
construct from a vector of decay channels
- **~DecayData** ()
- **DecayData** (**DecayData** const &orig)
- **DecayData** & **operator=** (**DecayData** const &rhs)
- void **swap** (**DecayData** &other)
- bool **isStable** () const
is this a stable particle?
- int **size** () const
how many decay channels does this particle have?
- **DecayChannel** **channel** (int i) const
*return this **DecayChannel** (p. 79)*
- void **write** (std::ostream &os) const
output decay information
- void **appendMode** (**DecayChannel**)
append a new decay channel

7.4.1 Detailed Description

The **DecayData** (p. 82) class holds particle decay information.

Author:

Lynn Garren

Use this class to access particle decay information.

Definition at line 24 of file DecayData.hh.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 HepPDT::DecayData::DecayData () [inline]

default constructor

Definition at line 9 of file DecayData.icc.

7.4.2.2 HepPDT::DecayData::DecayData (std::vector< DecayChannel > const & *dv*) [inline]

construct from a vector of decay channels

Definition at line 13 of file DecayData.icc.

7.4.2.3 HepPDT::DecayData::~DecayData () [inline]

Definition at line 17 of file DecayData.icc.

7.4.2.4 HepPDT::DecayData::DecayData (DecayData const & *orig*) [inline]

Definition at line 20 of file DecayData.icc.

7.4.3 Member Function Documentation**7.4.3.1 DecayData & HepPDT::DecayData::operator= (DecayData const & *rhs*) [inline]**

Definition at line 24 of file DecayData.icc.

References itsDecayList.

7.4.3.2 void HepPDT::DecayData::swap (DecayData & *other*) [inline]

Definition at line 33 of file DecayData.icc.

References itsDecayList, and HepPDT::swap().

Referenced by HepPDT::swap().

7.4.3.3 bool HepPDT::DecayData::isStable () const

is this a stable particle?

Definition at line 21 of file DecayData.cc.

7.4.3.4 int HepPDT::DecayData::size () const [inline]

how many decay channels does this particle have?

Definition at line 47 of file DecayData.hh.

7.4.3.5 DecayChannel HepPDT::DecayData::channel (int *i*) const [inline]

return this DecayChannel (p.79)

Definition at line 49 of file DecayData.hh.

7.4.3.6 void HepPDT::DecayData::write (std::ostream & *os*) const

output decay information

Definition at line 12 of file DecayData.cc.

7.4.3.7 void HepPDT::DecayData::appendMode (DecayChannel)

append a new decay channel

Definition at line 27 of file DecayData.cc.

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

- **DecayData.hh**
- **DecayData.icc**
- **DecayData.cc**

7.5 HepPDT::DefTable Class Reference

The **DefTable** (p. 85) class holds EvtGen definitions.

```
#include <DefTable.hh>
```

Public Types

- typedef TempDefMap::const_iterator **const_iterator**
- typedef TempDefMap::iterator **iterator**

Public Member Functions

- **DefTable** ()
- **~DefTable** ()
- void **addDefinition** (std::string const &def, double val)
add a definition to the map
- bool **hasDefinition** (std::string const &def) const
is this definition already defined?
- int **size** () const
get the size of the definition map
- double **definition** (std::string const &def)
return the definition of this parameter
- void **writeDefinitions** () const
use for diagnostics
- **iterator begin** ()
begin iterating over the definition map
- **const_iterator begin** () const
begin iterating over the definition map
- **iterator end** ()
end iterating over the definition map
- **const_iterator end** () const
end iterating over the definition map

7.5.1 Detailed Description

The **DefTable** (p. 85) class holds EvtGen definitions.

Author:

Lynn Garren

This is temporary information storage used when reading EvtGen input.

Definition at line 23 of file DefTable.hh.

7.5.2 Member Typedef Documentation

7.5.2.1 `typedef TempDefMap::const_iterator HepPDT::DefTable::const_iterator`

Definition at line 27 of file DefTable.hh.

7.5.2.2 `typedef TempDefMap::iterator HepPDT::DefTable::iterator`

Definition at line 28 of file DefTable.hh.

7.5.3 Constructor & Destructor Documentation

7.5.3.1 `HepPDT::DefTable::DefTable ()`

Definition at line 15 of file DefTable.cc.

7.5.3.2 `HepPDT::DefTable::~~DefTable () [inline]`

Definition at line 31 of file DefTable.hh.

7.5.4 Member Function Documentation

7.5.4.1 `void HepPDT::DefTable::addDefinition (std::string const & def, double val) [inline]`

add a definition to the map

Definition at line 36 of file DefTable.hh.

Referenced by `HepPDT::addEvtGenParticles()`.

7.5.4.2 `bool HepPDT::DefTable::hasDefinition (std::string const & def) const`

is this definition already defined?

Definition at line 31 of file DefTable.cc.

Referenced by `HepPDT::TableBuilder::hasDefinition()`.

7.5.4.3 `int HepPDT::DefTable::size () const [inline]`

get the size of the definition map

Definition at line 46 of file DefTable.hh.

7.5.4.4 double HepPDT::DefTable::definition (std::string const & def)

return the definition of this parameter

Definition at line 18 of file DefTable.cc.

Referenced by HepPDT::TableBuilder::definition().

7.5.4.5 void HepPDT::DefTable::writeDefinitions () const

use for diagnostics

Definition at line 41 of file DefTable.cc.

References begin(), and end().

7.5.4.6 iterator HepPDT::DefTable::begin () [inline]

begin iterating over the definition map

Definition at line 53 of file DefTable.hh.

Referenced by writeDefinitions().

7.5.4.7 const_iterator HepPDT::DefTable::begin () const [inline]

begin iterating over the definition map

Definition at line 55 of file DefTable.hh.

7.5.4.8 iterator HepPDT::DefTable::end () [inline]

end iterating over the definition map

Definition at line 58 of file DefTable.hh.

Referenced by writeDefinitions().

7.5.4.9 const_iterator HepPDT::DefTable::end () const [inline]

end iterating over the definition map

Definition at line 60 of file DefTable.hh.

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

- DefTable.hh
- DefTable.cc

7.6 HepPDT::Measurement Class Reference

The **Measurement** (p. 88) class defines a value with its error.

```
#include <Measurement.hh>
```

Public Member Functions

- **Measurement** ()
- **Measurement** (double value, double sigma)
- **Measurement** (const **Measurement** &m)
- void **swap** (**Measurement** &other)
- **Measurement** & **operator=** (**Measurement** const &rhs)
- bool **operator<** (**Measurement** const &other) const
compare the value, ignore the error
- bool **operator==** (**Measurement** const &other) const
compare the value, ignore the error
- double **value** () const
- double **sigma** () const
- **operator double** () const

7.6.1 Detailed Description

The **Measurement** (p. 88) class defines a value with its error.

Author:

Mark Fischler

Contain a value with associated errors. Provide simple comparisons.

Examples:

examMyPDT.cc.

Definition at line 20 of file Measurement.hh.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 HepPDT::Measurement::Measurement () [inline]

Definition at line 33 of file Measurement.icc.

7.6.2.2 HepPDT::Measurement::Measurement (double value, double sigma) [inline]

Definition at line 30 of file Measurement.icc.

7.6.2.3 HepPDT::Measurement::Measurement (const Measurement & *m*) [inline]

Definition at line 35 of file Measurement.icc.

7.6.3 Member Function Documentation

7.6.3.1 void HepPDT::Measurement::swap (Measurement & *other*) [inline]

Definition at line 38 of file Measurement.icc.

References sig, HepPDT::swap(), and val.

Referenced by operator=(), HepPDT::ResonanceStructure::swap(), and HepPDT::swap().

7.6.3.2 Measurement & HepPDT::Measurement::operator= (Measurement const & *rhs*) [inline]

Definition at line 44 of file Measurement.icc.

References swap().

7.6.3.3 bool HepPDT::Measurement::operator< (Measurement const & *other*) const [inline]

compare the value, ignore the error

Definition at line 51 of file Measurement.icc.

References val.

7.6.3.4 bool HepPDT::Measurement::operator== (Measurement const & *other*) const [inline]

compare the value, ignore the error

Definition at line 56 of file Measurement.icc.

References val.

7.6.3.5 double HepPDT::Measurement::value () const [inline]

Definition at line 61 of file Measurement.icc.

Referenced by HepPDT::ResonanceStructure::lifetime(), operator double(), HepPDT::ResonanceStructure::setTotalWidthFromLifetime(), and HepPDT::CommonParticleData::write().

7.6.3.6 double HepPDT::Measurement::sigma () const [inline]

Definition at line 62 of file Measurement.icc.

Referenced by HepPDT::ResonanceStructure::lifetime(), HepPDT::ResonanceStructure::setTotalWidthFromLifetime(), and HepPDT::CommonParticleData::write().

7.6.3.7 HepPDT::Measurement::operator double () const [inline]

Definition at line 64 of file Measurement.icc.

References value().

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

- **Measurement.hh**
- **Measurement.icc**

7.7 HepPDT::ParticleData Class Reference

The **ParticleData** (p. 91) class holds data for a single particle in the table.

```
#include <ParticleData.hh>
```

Public Types

- typedef **CommonParticleData** **CPD**
- typedef std::list< **CPD** > **CPDlist**
- typedef std::list< **DecayData** > **DDlist**
- typedef std::list< **CPD** >::const_iterator **CPDID**
- typedef std::list< **DecayData** >::const_iterator **DDID**

Public Member Functions

- **ParticleData** (const **CPDID** &cpdid, const **DDID** &ddid)
*create from a **CommonParticleData** (p. 65) and a **DecayData** (p. 82)*
- **~ParticleData** ()
- void **swap** (**ParticleData** &other)
- **ParticleData** (const **ParticleData** &orig)
- **ParticleData** & **operator=** (const **ParticleData** &rhs)
- const std::string & **name** () const
return the name of this particle
- **ParticleID** **ID** () const
*return the **ParticleID** (p. 109) of this particle*
- int **pid** () const
return the ID of this particle
- double **charge** () const
return the charge of this particle
- double **color** () const
return the color information for this particle
- **SpinState** **spin** () const
return the spin information for this particle
- **Measurement** **mass** () const
return the mass of this particle
- **Measurement** **totalWidth** () const
return the total width of this particle
- **Measurement** **lifetime** () const
return the calculated lifetime of this particle

- double **lowerCutoff** () const
lower cutoff of allowed width values
- double **upperCutoff** () const
upper cutoff of allowed width values
- int **numConstituents** () const
number of constituent particles (e.g., quarks)
- **Constituent constituent** (unsigned int i) const
constituent information
- **ParticleID constituentParticle** (unsigned int i) const
***ParticleID** (p. 109) for a constituent particle.*
- **ResonanceStructure** const **resonance** () const
resonance (width) information
- bool **isMeson** () const
is this a valid meson?
- bool **isBaryon** () const
is this a valid baryon?
- bool **isDiQuark** () const
is this a valid diquark?
- bool **isHadron** () const
is this a valid hadron?
- bool **isLepton** () const
is this a valid lepton?
- bool **isNucleus** () const
is this a valid ion?
- bool **hasUp** () const
does this particle contain an up quark?
- bool **hasDown** () const
does this particle contain a down quark?
- bool **hasStrange** () const
does this particle contain a strange quark?
- bool **hasCharm** () const
does this particle contain a charm quark?
- bool **hasBottom** () const
does this particle contain a bottom quark?

- **bool hasTop** () const
does this particle contain a top quark?
- **int numDecayChannels** () const
how many decay channels does this particle have?
- **bool isStable** () const
is this a stable particle?
- **DecayChannel channel** (int i) const
*return this *DecayChannel* (p. 79)*
- **void setDecayData** (DDID dd)
*change the *DecayData* (p. 82) reference*
- **DDID getDecayData** () const
*get the *DecayData* (p. 82)*
- **CPDID getCommonParticleData** () const
*get the *CommonParticleData* (p. 65)*
- **void write** (std::ostream &os) const
output all information about a particle INCLUDING its decays
- **void writeParticleInfo** (std::ostream &os) const
output all information about a particle EXCEPT its decays
- **void writeParticleTranslation** (std::ostream &os) const
output the translation information for this particle

7.7.1 Detailed Description

The **ParticleData** (p. 91) class holds data for a single particle in the table.

Author:

Lynn Garren

All the particle information is accessed via this class.

Examples:

examMyPDT.cc, and **testHepPDT.cc**.

Definition at line 35 of file ParticleData.hh.

7.7.2 Member Typedef Documentation

7.7.2.1 typedef CommonParticleData HepPDT::ParticleData::CPD

Definition at line 40 of file ParticleData.hh.

7.7.2.2 typedef std::list<CPD> HepPDT::ParticleData::CPDlist

Definition at line 42 of file ParticleData.hh.

7.7.2.3 typedef std::list<DecayData> HepPDT::ParticleData::DDlist

Definition at line 43 of file ParticleData.hh.

7.7.2.4 typedef std::list<CPD>::const_iterator HepPDT::ParticleData::CPDID

Definition at line 45 of file ParticleData.hh.

7.7.2.5 typedef std::list<DecayData>::const_iterator HepPDT::ParticleData::DDID

Definition at line 46 of file ParticleData.hh.

7.7.3 Constructor & Destructor Documentation

7.7.3.1 HepPDT::ParticleData::ParticleData (const CPDID & *cpdid*, const DDID & *ddid*) [inline]

create from a **CommonParticleData** (p. 65) and a **DecayData** (p. 82)

Definition at line 12 of file ParticleData.icc.

7.7.3.2 HepPDT::ParticleData::~~ParticleData () [inline]

Definition at line 17 of file ParticleData.icc.

7.7.3.3 HepPDT::ParticleData::ParticleData (const ParticleData & *orig*) [inline]

Definition at line 26 of file ParticleData.icc.

7.7.4 Member Function Documentation

7.7.4.1 void HepPDT::ParticleData::swap (ParticleData & *other*) [inline]

Definition at line 20 of file ParticleData.icc.

References itsCPDID, itsDDID, and HepPDT::swap().

Referenced by operator=(), and HepPDT::swap().

7.7.4.2 ParticleData & HepPDT::ParticleData::operator= (const ParticleData & rhs) [inline]

Definition at line 31 of file ParticleData.icc.

References swap().

7.7.4.3 const std::string& HepPDT::ParticleData::name () const [inline]

return the name of this particle

Examples:

examMyPDT.cc.

Definition at line 63 of file ParticleData.hh.

Referenced by main().

7.7.4.4 ParticleID HepPDT::ParticleData::ID () const [inline]

return the **ParticleID** (p. 109) of this particle

Definition at line 65 of file ParticleData.hh.

7.7.4.5 int HepPDT::ParticleData::pid () const [inline]

return the ID of this particle

Definition at line 67 of file ParticleData.hh.

7.7.4.6 double HepPDT::ParticleData::charge () const [inline]

return the charge of this particle

Definition at line 69 of file ParticleData.hh.

7.7.4.7 double HepPDT::ParticleData::color () const [inline]

return the color information for this particle

Definition at line 71 of file ParticleData.hh.

7.7.4.8 SpinState HepPDT::ParticleData::spin () const [inline]

return the spin information for this particle

Definition at line 73 of file ParticleData.hh.

7.7.4.9 Measurement HepPDT::ParticleData::mass () const [inline]

return the mass of this particle

Definition at line 75 of file ParticleData.hh.

7.7.4.10 Measurement HepPDT::ParticleData::totalWidth () const [inline]

return the total width of this particle

Examples:

`testHepPDT.cc.`

Definition at line 77 of file ParticleData.hh.

Referenced by `main()`.

7.7.4.11 Measurement HepPDT::ParticleData::lifetime () const [inline]

return the calculated lifetime of this particle

Definition at line 79 of file ParticleData.hh.

7.7.4.12 double HepPDT::ParticleData::lowerCutoff () const [inline]

lower cutoff of allowed width values

Examples:

`testHepPDT.cc.`

Definition at line 81 of file ParticleData.hh.

Referenced by `main()`.

7.7.4.13 double HepPDT::ParticleData::upperCutoff () const [inline]

upper cutoff of allowed width values

Examples:

`testHepPDT.cc.`

Definition at line 83 of file ParticleData.hh.

Referenced by `main()`.

7.7.4.14 int HepPDT::ParticleData::numConstituents () const [inline]

number of constituent particles (e.g., quarks)

Definition at line 85 of file ParticleData.hh.

7.7.4.15 Constituent HepPDT::ParticleData::constituent (unsigned int *i*) const [inline]

constituent information

Definition at line 87 of file ParticleData.hh.

7.7.4.16 ParticleID HepPDT::ParticleData::constituentParticle (unsigned int *i*) const [inline]

ParticleID (p. 109) for a constituent particle.

Definition at line 90 of file ParticleData.hh.

7.7.4.17 ResonanceStructure const HepPDT::ParticleData::resonance () const [inline]

resonance (width) information

Definition at line 93 of file ParticleData.hh.

7.7.4.18 bool HepPDT::ParticleData::isMeson () const [inline]

is this a valid meson?

Definition at line 98 of file ParticleData.hh.

7.7.4.19 bool HepPDT::ParticleData::isBaryon () const [inline]

is this a valid baryon?

Definition at line 100 of file ParticleData.hh.

7.7.4.20 bool HepPDT::ParticleData::isDiQuark () const [inline]

is this a valid diquark?

Definition at line 102 of file ParticleData.hh.

7.7.4.21 bool HepPDT::ParticleData::isHadron () const [inline]

is this a valid hadron?

Definition at line 104 of file ParticleData.hh.

7.7.4.22 bool HepPDT::ParticleData::isLepton () const [inline]

is this a valid lepton?

Definition at line 106 of file ParticleData.hh.

7.7.4.23 bool HepPDT::ParticleData::isNucleus () const [inline]

is this a valid ion?

Definition at line 108 of file ParticleData.hh.

7.7.4.24 bool HepPDT::ParticleData::hasUp () const [inline]

does this particle contain an up quark?

Definition at line 110 of file ParticleData.hh.

7.7.4.25 `bool HepPDT::ParticleData::hasDown () const [inline]`

does this particle contain a down quark?

Definition at line 112 of file ParticleData.hh.

7.7.4.26 `bool HepPDT::ParticleData::hasStrange () const [inline]`

does this particle contain a strange quark?

Definition at line 114 of file ParticleData.hh.

7.7.4.27 `bool HepPDT::ParticleData::hasCharm () const [inline]`

does this particle contain a charm quark?

Definition at line 116 of file ParticleData.hh.

7.7.4.28 `bool HepPDT::ParticleData::hasBottom () const [inline]`

does this particle contain a bottom quark?

Definition at line 118 of file ParticleData.hh.

7.7.4.29 `bool HepPDT::ParticleData::hasTop () const [inline]`

does this particle contain a top quark?

Definition at line 120 of file ParticleData.hh.

7.7.4.30 `int HepPDT::ParticleData::numDecayChannels () const [inline]`

how many decay channels does this particle have?

Definition at line 137 of file ParticleData.hh.

7.7.4.31 `bool HepPDT::ParticleData::isStable () const [inline]`

is this a stable particle?

Definition at line 139 of file ParticleData.hh.

7.7.4.32 `DecayChannel HepPDT::ParticleData::channel (int i) const [inline]`

return this `DecayChannel` (p.79)

Definition at line 141 of file ParticleData.hh.

7.7.4.33 void HepPDT::ParticleData::setDecayData (DDID *dd*) [inline]

change the **DecayData** (p. 82) reference

Definition at line 147 of file ParticleData.hh.

7.7.4.34 DDID HepPDT::ParticleData::getDecayData () const [inline]

get the **DecayData** (p. 82)

Definition at line 152 of file ParticleData.hh.

7.7.4.35 CPDID HepPDT::ParticleData::getCommonParticleData () const [inline]

get the **CommonParticleData** (p. 65)

Definition at line 154 of file ParticleData.hh.

7.7.4.36 void HepPDT::ParticleData::write (std::ostream & *os*) const [inline]

output all information about a particle INCLUDING its decays

Examples:

testHepPDT.cc.

Definition at line 38 of file ParticleData.icc.

Referenced by main().

7.7.4.37 void HepPDT::ParticleData::writeParticleInfo (std::ostream & *os*) const [inline]

output all information about a particle EXCEPT its decays

Definition at line 44 of file ParticleData.icc.

7.7.4.38 void HepPDT::ParticleData::writeParticleTranslation (std::ostream & *os*) const [inline]

output the translation information for this particle

Definition at line 49 of file ParticleData.icc.

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

- **ParticleData.hh**
- **ParticleData.icc**

7.8 HepPDT::ParticleDataTable Class Reference

The `ParticleDataTable` (p.100) class is the core of `HepPDT` (p.23).

```
#include <ParticleDataTable.hh>
```

Public Types

- typedef `CommonParticleData` `CPD`
- typedef `std::list< CommonParticleData >` `CPDlist`
- typedef `std::list< DecayData >` `DDlist`
- typedef `CPDlist::const_iterator` `CPDID`
- typedef `DDlist::const_iterator` `DDID`
- typedef `std::map< ParticleID, TempParticleData >` `TempMap`
- typedef `std::map< ParticleID, ParticleData, ParticleDataTableComparison >` `PDTMap`
- typedef `std::map< std::string, ParticleID >` `PDTNameMap`
- typedef `PDTMap::const_iterator` `const_iterator`
- typedef `PDTNameMap::const_iterator` `const_iteratorByName`

Public Member Functions

- `ParticleDataTable` (`std::string name=" "`)
construct a particle data table with an identifying name
- `~ParticleDataTable` ()
- `int size` () `const`
size of the particle data table
- `const_iterator begin` () `const`
begin iterating over the particle data table
- `const_iterator end` () `const`
end iterating over the particle data table
- `int sizeNameMap` () `const`
size of the map of particle names
- `const_iteratorByName beginNameMap` () `const`
begin iterating over the map of particle names
- `const_iteratorByName endNameMap` () `const`
end iterating over the map of particle names
- `std::string tableName` () `const`
return the name of this particle data table
- `ParticleData const * particle` (`ParticleID`) `const`
access particle information via `ParticleID` (p.109)

- **ParticleData * particle (ParticleID)**
*access particle information via **ParticleID** (p. 109)*
- **ParticleData const * particle (std::string) const**
access particle information via a particle name
- **ParticleData * particle (std::string)**
access particle information via a particle name
- **ParticleData * operator[] (ParticleID)**
*access particle information via **ParticleID** (p. 109)*
- **ParticleData const * operator[] (ParticleID) const**
*access particle information via **ParticleID** (p. 109)*
- **ParticleData * operator[] (std::string)**
access particle information via a particle name
- **ParticleData const * operator[] (std::string) const**
access particle information via a particle name
- **void writeParticleData (std::ostream &ostr)**
output all information in the PDT
- **void writeParticleInfo (std::ostream &ostr)**
output all information about a particle EXCEPT its decays
- **void writeParticleTranslation (std::ostream &ostr)**
output a list of original IDs and their translations
- **void convertTemporaryMap (TempMap &tempPDT, std::ostream &err)**
*used by the **TableBuilder** (p. 141) destructor to fill the PDT*
- **DDID addDecayData (DecayData const d)**
add decay data information

7.8.1 Detailed Description

The **ParticleDataTable** (p. 100) class is the core of **HepPDT** (p. 23).

Author:

Lynn Garren, Walter Brown

This is the table of particle data information. This table is created once at the beginning of a job and referenced as needed. We expect that the table will be saved as part of the standard job output. Methods exist to fill the table from a variety of input formats. The user may fill the table from as many different input streams as desired. See the examples.

Examples:

`examMyPDT.cc`, `listEvtGenNames.cc.in`, `listPDGNames.cc.in`, `listPythiaNames.cc.in`, `testHepPDT.cc`, `testReadEvtGen.cc.in`, `testReadIsajet.cc.in`, and `testReadQQ.cc.in`.

Definition at line 45 of file `ParticleDataTable.hh`.

7.8.2 Member Typedef Documentation**7.8.2.1 typedef CommonParticleData HepPDT::ParticleDataTable::CPD**

Definition at line 48 of file `ParticleDataTable.hh`.

7.8.2.2 typedef std::list<CommonParticleData> HepPDT::ParticleDataTable::CPDlist

Definition at line 50 of file `ParticleDataTable.hh`.

7.8.2.3 typedef std::list<DecayData> HepPDT::ParticleDataTable::DDlist

Definition at line 51 of file `ParticleDataTable.hh`.

7.8.2.4 typedef CPDlist::const_iterator HepPDT::ParticleDataTable::CPDID

Definition at line 53 of file `ParticleDataTable.hh`.

7.8.2.5 typedef DDlist::const_iterator HepPDT::ParticleDataTable::DDID

Definition at line 54 of file `ParticleDataTable.hh`.

7.8.2.6 typedef std::map<ParticleID,TempParticleData> HepPDT::ParticleDataTable::TempMap

Definition at line 56 of file `ParticleDataTable.hh`.

7.8.2.7 typedef std::map<ParticleID,ParticleData,ParticleDataTableComparison> HepPDT::ParticleDataTable::PDTMap

Definition at line 57 of file `ParticleDataTable.hh`.

7.8.2.8 typedef std::map<std::string,ParticleID> HepPDT::ParticleDataTable::PDTNameMap

Definition at line 58 of file `ParticleDataTable.hh`.

7.8.2.9 typedef PDTMap::const_iterator HepPDT::ParticleDataTable::const_iterator

Definition at line 60 of file ParticleDataTable.hh.

7.8.2.10 typedef PDTNameMap::const_iterator HepPDT::ParticleDataTable::const_iteratorByName

Definition at line 61 of file ParticleDataTable.hh.

7.8.3 Constructor & Destructor Documentation

7.8.3.1 HepPDT::ParticleDataTable::ParticleDataTable (std::string *name* = " ") [inline]

construct a particle data table with an identifying name

Definition at line 11 of file ParticleDataTable.icc.

References HepPDT::version().

7.8.3.2 HepPDT::ParticleDataTable::~~ParticleDataTable () [inline]

Definition at line 17 of file ParticleDataTable.icc.

7.8.4 Member Function Documentation

7.8.4.1 int HepPDT::ParticleDataTable::size () const [inline]

size of the particle data table

Definition at line 72 of file ParticleDataTable.hh.

Referenced by writeParticleData(), and writeParticleInfo().

7.8.4.2 const_iterator HepPDT::ParticleDataTable::begin () const [inline]

begin iterating over the particle data table

Definition at line 74 of file ParticleDataTable.hh.

Referenced by writeParticleData(), writeParticleInfo(), and writeParticleTranslation().

7.8.4.3 const_iterator HepPDT::ParticleDataTable::end () const [inline]

end iterating over the particle data table

Definition at line 76 of file ParticleDataTable.hh.

Referenced by writeParticleData(), writeParticleInfo(), and writeParticleTranslation().

7.8.4.4 `int HepPDT::ParticleDataTable::sizeNameMap () const [inline]`

size of the map of particle names

Definition at line 78 of file ParticleDataTable.hh.

7.8.4.5 `const_iteratorByName HepPDT::ParticleDataTable::beginNameMap () const [inline]`

begin iterating over the map of particle names

Definition at line 80 of file ParticleDataTable.hh.

7.8.4.6 `const_iteratorByName HepPDT::ParticleDataTable::endNameMap () const [inline]`

end iterating over the map of particle names

Definition at line 82 of file ParticleDataTable.hh.

7.8.4.7 `std::string HepPDT::ParticleDataTable::tableName () const [inline]`

return the name of this particle data table

Definition at line 84 of file ParticleDataTable.hh.

Referenced by writeParticleData(), and writeParticleInfo().

7.8.4.8 `ParticleData const * HepPDT::ParticleDataTable::particle (ParticleID) const [inline]`

access particle information via **ParticleID** (p. 109)

Examples:

`examMyPDT.cc`, and `testHepPDT.cc`.

Definition at line 32 of file ParticleDataTable.icc.

Referenced by convertTemporaryMap(), main(), operator[](), and particle().

7.8.4.9 `ParticleData * HepPDT::ParticleDataTable::particle (ParticleID) [inline]`

access particle information via **ParticleID** (p. 109)

Definition at line 20 of file ParticleDataTable.icc.

7.8.4.10 `ParticleData const * HepPDT::ParticleDataTable::particle (std::string) const [inline]`

access particle information via a particle name

Definition at line 66 of file ParticleDataTable.icc.

References particle().

7.8.4.11 ParticleData * HepPDT::ParticleDataTable::particle (std::string)
[inline]

access particle information via a particle name

Definition at line 55 of file ParticleDataTable.icc.

References particle().

7.8.4.12 ParticleData * HepPDT::ParticleDataTable::operator[] (ParticleID)
[inline]

access particle information via **ParticleID** (p. 109)

Definition at line 45 of file ParticleDataTable.icc.

References particle().

7.8.4.13 ParticleData const * HepPDT::ParticleDataTable::operator[] (ParticleID)
const [inline]

access particle information via **ParticleID** (p. 109)

Definition at line 50 of file ParticleDataTable.icc.

References particle().

7.8.4.14 ParticleData * HepPDT::ParticleDataTable::operator[] (std::string)
[inline]

access particle information via a particle name

Definition at line 77 of file ParticleDataTable.icc.

References particle().

7.8.4.15 ParticleData const * HepPDT::ParticleDataTable::operator[] (std::string)
const [inline]

access particle information via a particle name

Definition at line 82 of file ParticleDataTable.icc.

References particle().

7.8.4.16 void HepPDT::ParticleDataTable::writeParticleData (std::ostream & outstr)

output all information in the PDT

Examples:

examMyPDT.cc, testHepPDT.cc, testReadEvtGen.cc.in, testReadIsajet.cc.in, and testReadQQ.cc.in.

Definition at line 18 of file ParticleDataTable.cc.

References `begin()`, `end()`, `size()`, `tableName()`, and `HepPDT::writeVersion()`.

Referenced by `main()`.

7.8.4.17 `void HepPDT::ParticleDataTable::writeParticleInfo (std::ostream & ostr)`

output all information about a particle EXCEPT its decays

Examples:

`testHepPDT.cc`, and `testReadIsajet.cc.in`.

Definition at line 44 of file `ParticleDataTable.cc`.

References `begin()`, `end()`, `size()`, `tableName()`, and `HepPDT::writeVersion()`.

Referenced by `main()`.

7.8.4.18 `void HepPDT::ParticleDataTable::writeParticleTranslation (std::ostream & ostr) [inline]`

output a list of original IDs and their translations

Examples:

`listEvtGenNames.cc.in`, `listPDGNames.cc.in`, `listPythiaNames.cc.in`, and `test-ReadQQ.cc.in`.

Definition at line 87 of file `ParticleDataTable.icc`.

References `begin()`, and `end()`.

Referenced by `main()`.

7.8.4.19 `void HepPDT::ParticleDataTable::convertTemporaryMap (TempMap & tempPDT, std::ostream & err)`

used by the `TableBuilder` (p. 141) destructor to fill the PDT

Definition at line 19 of file `convertTemporaryMap.cc`.

References `addDecayData()`, and `particle()`.

Referenced by `HepPDT::TableBuilder::~~TableBuilder()`.

7.8.4.20 `ParticleDataTable::DDID HepPDT::ParticleDataTable::addDecayData (DecayData const d)`

add decay data information

Definition at line 70 of file `ParticleDataTable.cc`.

Referenced by `convertTemporaryMap()`.

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

- ParticleDataTable.hh
- ParticleDataTable.icc
- convertTemporaryMap.cc
- ParticleDataTable.cc

7.9 HepPDT::ParticleDataTableComparison Class Reference

The `ParticleDataTableComparison` (p.108) class provides a utility for sorting the PDT.

```
#include <ParticleDataTableComparison.hh>
```

Public Member Functions

- `ParticleDataTableComparison ()`
stateless class
- `bool operator() (const ParticleID &, const ParticleID &) const`

7.9.1 Detailed Description

The `ParticleDataTableComparison` (p.108) class provides a utility for sorting the PDT.

Author:

Lynn Garren

Sort the PDT by absolute value of the particle ID. If the absolute values are equal, the positive number comes first.

Definition at line 23 of file `ParticleDataTableComparison.hh`.

7.9.2 Constructor & Destructor Documentation

7.9.2.1 HepPDT::ParticleDataTableComparison::ParticleDataTableComparison () [inline]

stateless class

Definition at line 27 of file `ParticleDataTableComparison.hh`.

7.9.3 Member Function Documentation

7.9.3.1 bool HepPDT::ParticleDataTableComparison::operator() (const ParticleID &, const ParticleID &) const [inline]

Comparison method sorts by absolute value. If the absolute values are the same, the positive entry comes first.

Definition at line 36 of file `ParticleDataTableComparison.hh`.

References `HepPDT::ParticleID::abspid()`, and `HepPDT::ParticleID::pid()`.

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

- `ParticleDataTableComparison.hh`

7.10 HepPDT::ParticleID Class Reference

The **ParticleID** (p.109) has various utilities to extract information from the particle ID.

```
#include <ParticleID.hh>
```

Public Member Functions

- **ParticleID** (int pid=0)
create from an integer ID
- **ParticleID** (const **ParticleID** &orig)
- **ParticleID** & **operator=** (const **ParticleID** &)
- void **swap** (**ParticleID** &other)
- bool **operator<** (**ParticleID** const &other) const
- bool **operator==** (**ParticleID** const &other) const
- int **pid** () const
get the integer ID
- int **abspid** () const
get the absolute value
- bool **isValid** () const
is this a valid ID?
- bool **isMeson** () const
is this a valid meson ID?
- bool **isBaryon** () const
is this a valid baryon ID?
- bool **isDiQuark** () const
is this a valid diquark ID?
- bool **isHadron** () const
is this a valid hadron ID?
- bool **isLepton** () const
is this a valid lepton ID?
- bool **isNucleus** () const
is this a valid ion ID?
- bool **isPentaquark** () const
is this a valid pentaquark ID?
- bool **isSUSY** () const
is this a valid SUSY ID?
- bool **isRhadron** () const

is this a valid R-hadron ID?

- bool **hasUp** () const
does this particle contain an up quark?
- bool **hasDown** () const
does this particle contain a down quark?
- bool **hasStrange** () const
does this particle contain a strange quark?
- bool **hasCharm** () const
does this particle contain a charm quark?
- bool **hasBottom** () const
does this particle contain a bottom quark?
- bool **hasTop** () const
does this particle contain a top quark?
- int **jSpin** () const
jSpin returns $2J+1$, where J is the total spin
- int **sSpin** () const
sSpin returns $2S+1$, where S is the spin
- int **lSpin** () const
lSpin returns $2L+1$, where L is the orbital angular momentum
- int **fundamentalID** () const
return the first two digits if this is a "fundamental" particle
- int **extraBits** () const
- **Quarks quarks** () const
returns a list of 3 constituent quarks
- int **threeCharge** () const
- int **A** () const
if this is a nucleus (ion), get A
- int **Z** () const
if this is a nucleus (ion), get Z
- int **lambda** () const
if this is a nucleus (ion), get nLambda
- unsigned short **digit (location)** const
return the digit at a named location in the PID
- const std::string **PDName** () const
standard particle name

7.10.1 Detailed Description

The **ParticleID** (p. 109) has various utilities to extract information from the particle ID.

Author:

Lynn Garren

In the standard numbering scheme, the PID digits (base 10) are: +/- n nr nl nq1 nq2 nq3 nj It is expected that any 7 digit number used as a PID will adhere to the Monte Carlo numbering scheme documented by the PDG. Note that particles not already explicitly defined can be expressed within this numbering scheme.

Examples:

`examMyPDT.cc`, `testHepPDT.cc`, and `testPID.cc`.

Definition at line 64 of file `ParticleID.hh`.

7.10.2 Constructor & Destructor Documentation

7.10.2.1 HepPDT::ParticleID::ParticleID (int *pid* = 0) [inline]

create from an integer ID

Definition at line 10 of file `ParticleID.icc`.

7.10.2.2 HepPDT::ParticleID::ParticleID (const ParticleID & *orig*) [inline]

Definition at line 14 of file `ParticleID.icc`.

7.10.3 Member Function Documentation

7.10.3.1 ParticleID & HepPDT::ParticleID::operator= (const ParticleID &) [inline]

Definition at line 18 of file `ParticleID.icc`.

References `swap()`.

7.10.3.2 void HepPDT::ParticleID::swap (ParticleID & *other*) [inline]

Definition at line 25 of file `ParticleID.icc`.

References `itsPID`, and `HepPDT::swap()`.

Referenced by `operator=()`, `HepPDT::swap()`, and `HepPDT::CommonParticleData::swap()`.

7.10.3.3 bool HepPDT::ParticleID::operator< (ParticleID const & *other*) const [inline]

Definition at line 30 of file `ParticleID.icc`.

References `itsPID`.

7.10.3.4 `bool HepPDT::ParticleID::operator==(ParticleID const & other) const` [inline]

Definition at line 35 of file ParticleID.icc.

References itsPID.

7.10.3.5 `int HepPDT::ParticleID::pid () const` [inline]

get the integer ID

Definition at line 85 of file ParticleID.hh.

Referenced by `HepPDT::TempParticleData::antiparticle()`, `HepPDT::buildTPD()`, `HepPDT::TableBuilder::getAntiParticle()`, `isMeson()`, `HepPDT::ParticleDataTableComparison::operator()()`, `HepPDT::parseEvtGenDecayLine()`, `HepPDT::parseEvtGenLine()`, `HepPDT::detail::parseIsajetDecayLine()`, `HepPDT::detail::parseIsajetLine()`, `HepPDT::detail::parseParticleLine()`, `HepPDT::detail::parsePythiaDecayLine()`, `HepPDT::detail::parsePythiaLine()`, `HepPDT::parseQQParticle()`, `HepPDT::CommonParticleData::pid()`, `HepPDT::TempParticleData::processPID()`, `threeCharge()`, and `HepPDT::CommonParticleData::write()`.

7.10.3.6 `int HepPDT::ParticleID::abspid () const` [inline]

get the absolute value

return a value greater than or equal to zero

Definition at line 40 of file ParticleID.icc.

Referenced by `A()`, `digit()`, `extraBits()`, `fundamentalID()`, `hasBottom()`, `hasCharm()`, `hasDown()`, `hasStrange()`, `hasTop()`, `hasUp()`, `isBaryon()`, `isDiQuark()`, `isMeson()`, `isNucleus()`, `jSpin()`, `lambda()`, `lSpin()`, `HepPDT::ParticleDataTableComparison::operator()()`, `quarks()`, `sSpin()`, `threeCharge()`, and `Z()`.

7.10.3.7 `bool HepPDT::ParticleID::isValid () const`

is this a valid ID?

Definition at line 100 of file ParticleID.cc.

References `extraBits()`, `fundamentalID()`, `isBaryon()`, `isDiQuark()`, `isMeson()`, `isNucleus()`, `isPentaquark()`, `isRhadron()`, and `isSUSY()`.

Referenced by `HepPDT::addParticleTable()`, and `HepPDT::CommonParticleData::write()`.

7.10.3.8 `bool HepPDT::ParticleID::isMeson () const`

is this a valid meson ID?

Definition at line 219 of file ParticleID.cc.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `HepPDT::nj`, `HepPDT::mq1`, `HepPDT::mq2`, `HepPDT::mq3`, and `pid()`.

Referenced by `isHadron()`, `HepPDT::CommonParticleData::isMeson()`, `isValid()`, `lSpin()`, and `sSpin()`.

7.10.3.9 bool HepPDT::ParticleID::isBaryon () const

is this a valid baryon ID?

Definition at line 279 of file ParticleID.cc.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `HepPDT::nj`, `HepPDT::nq1`, `HepPDT::nq2`, and `HepPDT::nq3`.

Referenced by `HepPDT::CommonParticleData::isBaryon()`, `isHadron()`, and `isValid()`.

7.10.3.10 bool HepPDT::ParticleID::isDiQuark () const

is this a valid diquark ID?

Definition at line 261 of file ParticleID.cc.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `HepPDT::nj`, `HepPDT::nq1`, `HepPDT::nq2`, and `HepPDT::nq3`.

Referenced by `HepPDT::CommonParticleData::isDiQuark()`, and `isValid()`.

7.10.3.11 bool HepPDT::ParticleID::isHadron () const

is this a valid hadron ID?

Definition at line 251 of file ParticleID.cc.

References `extraBits()`, `isBaryon()`, `isMeson()`, and `isPentaquark()`.

Referenced by `HepPDT::CommonParticleData::isHadron()`.

7.10.3.12 bool HepPDT::ParticleID::isLepton () const

is this a valid lepton ID?

Definition at line 243 of file ParticleID.cc.

References `extraBits()`, and `fundamentalID()`.

Referenced by `HepPDT::CommonParticleData::isLepton()`.

7.10.3.13 bool HepPDT::ParticleID::isNucleus () const

is this a valid ion ID?

Definition at line 179 of file ParticleID.cc.

References `A()`, `abspid()`, `digit()`, `HepPDT::n10`, `HepPDT::n9`, and `Z()`.

Referenced by `HepPDT::CommonParticleData::isNucleus()`, `isValid()`, and `lambda()`.

7.10.3.14 bool HepPDT::ParticleID::isPentaquark () const

is this a valid pentaquark ID?

Definition at line 125 of file ParticleID.cc.

References `digit()`, `extraBits()`, `HepPDT::n`, `HepPDT::nj`, `HepPDT::nl`, `HepPDT::nq1`, `HepPDT::nq2`, `HepPDT::nq3`, and `HepPDT::mr`.

Referenced by `isHadron()`, and `isValid()`.

7.10.3.15 bool HepPDT::ParticleID::isSUSY () const

is this a valid SUSY ID?

Definition at line 162 of file `ParticleID.cc`.

References `digit()`, `extraBits()`, `fundamentalID()`, `HepPDT::n`, and `HepPDT::nr`.

Referenced by `isRhadron()`, and `isValid()`.

7.10.3.16 bool HepPDT::ParticleID::isRhadron () const

is this a valid R-hadron ID?

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

References `digit()`, `extraBits()`, `isSUSY()`, `HepPDT::n`, `HepPDT::nj`, `HepPDT::nq2`, `HepPDT::nq3`, and `HepPDT::nr`.

Referenced by `isValid()`.

7.10.3.17 bool HepPDT::ParticleID::hasUp () const

does this particle contain an up quark?

Definition at line 45 of file `ParticleID.cc`.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `HepPDT::nq1`, `HepPDT::nq2`, and `HepPDT::nq3`.

7.10.3.18 bool HepPDT::ParticleID::hasDown () const

does this particle contain a down quark?

Definition at line 54 of file `ParticleID.cc`.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `HepPDT::nq1`, `HepPDT::nq2`, and `HepPDT::nq3`.

7.10.3.19 bool HepPDT::ParticleID::hasStrange () const

does this particle contain a strange quark?

Definition at line 63 of file `ParticleID.cc`.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `HepPDT::nq1`, `HepPDT::nq2`, and `HepPDT::nq3`.

7.10.3.20 bool HepPDT::ParticleID::hasCharm () const

does this particle contain a charm quark?

Definition at line 72 of file `ParticleID.cc`.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `HepPDT::nq1`, `HepPDT::nq2`, and `HepPDT::nq3`.

7.10.3.21 `bool HepPDT::ParticleID::hasBottom () const`

does this particle contain a bottom quark?

Definition at line 81 of file `ParticleID.cc`.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `HepPDT::nq1`, `HepPDT::nq2`, and `HepPDT::nq3`.

7.10.3.22 `bool HepPDT::ParticleID::hasTop () const`

does this particle contain a top quark?

Definition at line 90 of file `ParticleID.cc`.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `HepPDT::nq1`, `HepPDT::nq2`, and `HepPDT::nq3`.

7.10.3.23 `int HepPDT::ParticleID::jSpin () const`

`jSpin` returns $2J+1$, where J is the total spin

Definition at line 290 of file `ParticleID.cc`.

References `abspid()`, `extraBits()`, and `fundamentalID()`.

Referenced by `HepPDT::TempParticleData::processPID()`.

7.10.3.24 `int HepPDT::ParticleID::sSpin () const`

`sSpin` returns $2S+1$, where S is the spin

Definition at line 357 of file `ParticleID.cc`.

References `abspid()`, `isMeson()`, and `HepPDT::nl`.

Referenced by `HepPDT::TempParticleData::processPID()`.

7.10.3.25 `int HepPDT::ParticleID::lSpin () const`

`lSpin` returns $2L+1$, where L is the orbital angular momentum

Definition at line 307 of file `ParticleID.cc`.

References `abspid()`, `isMeson()`, and `HepPDT::nl`.

Referenced by `HepPDT::TempParticleData::processPID()`.

7.10.3.26 `int HepPDT::ParticleID::fundamentalID () const`

return the first two digits if this is a "fundamental" particle

Definition at line 33 of file `ParticleID.cc`.

References `abspid()`, `digit()`, `HepPDT::n10`, `HepPDT::n9`, `HepPDT::nq1`, and `HepPDT::nq2`.

Referenced by `hasBottom()`, `hasCharm()`, `hasDown()`, `hasStrange()`, `hasTop()`, `hasUp()`, `isBaryon()`, `HepPDT::Constituent::isBottom()`, `HepPDT::Constituent::isCharm()`, `isDiQuark()`, `HepPDT::Constituent::isDown()`, `isLepton()`, `isMeson()`, `HepPDT::Constituent::isStrange()`, `isSUSY()`, `HepPDT::Constituent::isTop()`, `HepPDT::Constituent::isUp()`, `isValid()`, `jSpin()`, `quarks()`, and `threeCharge()`.

7.10.3.27 `int HepPDT::ParticleID::extraBits () const`

returns everything beyond the 7th digit (e.g. outside the standard numbering scheme)

Definition at line 25 of file `ParticleID.cc`.

References `abspid()`.

Referenced by `hasBottom()`, `hasCharm()`, `hasDown()`, `hasStrange()`, `hasTop()`, `hasUp()`, `isBaryon()`, `isDiQuark()`, `isHadron()`, `isLepton()`, `isMeson()`, `isPentaquark()`, `isRhadron()`, `isSUSY()`, `isValid()`, `jSpin()`, `quarks()`, and `threeCharge()`.

7.10.3.28 `Quarks HepPDT::ParticleID::quarks () const`

returns a list of 3 constituent quarks

Definition at line 14 of file `quarks.cc`.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `HepPDT::nq1`, `HepPDT::Quarks::nq1`, `HepPDT::nq2`, `HepPDT::Quarks::nq2`, `HepPDT::nq3`, and `HepPDT::Quarks::nq3`.

Referenced by `HepPDT::TempParticleData::processPID()`.

7.10.3.29 `int HepPDT::ParticleID::threeCharge () const`

this is mostly for use by functions like `addPDGParticles` that have to figure out the charge from the PID

Definition at line 382 of file `ParticleID.cc`.

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `HepPDT::nj`, `HepPDT::nq1`, `HepPDT::nq2`, `HepPDT::nq3`, and `pid()`.

Referenced by `HepPDT::TempParticleData::processPID()`.

7.10.3.30 `int HepPDT::ParticleID::A () const`

if this is a nucleus (ion), get A

Definition at line 192 of file `ParticleID.cc`.

References `abspid()`, `digit()`, `HepPDT::n10`, and `HepPDT::n9`.

Referenced by `isNucleus()`.

7.10.3.31 `int HepPDT::ParticleID::Z () const`

if this is a nucleus (ion), get Z

Definition at line 201 of file `ParticleID.cc`.

References `abspid()`, `digit()`, `HepPDT::n10`, and `HepPDT::n9`.

Referenced by `isNucleus()`.

7.10.3.32 `int HepPDT::ParticleID::lambda () const`

if this is a nucleus (ion), get `nLambda`

Definition at line 210 of file `ParticleID.cc`.

References `abspid()`, `digit()`, `isNucleus()`, and `HepPDT::n8`.

7.10.3.33 `unsigned short HepPDT::ParticleID::digit (location) const`

return the digit at a named location in the PID

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

References `abspid()`.

Referenced by `A()`, `fundamentalID()`, `hasBottom()`, `hasCharm()`, `hasDown()`, `hasStrange()`, `hasTop()`, `hasUp()`, `isBaryon()`, `isDiQuark()`, `isMeson()`, `isNucleus()`, `isPentaquark()`, `isRhadron()`, `isSUSY()`, `lambda()`, `quarks()`, `threeCharge()`, and `Z()`.

7.10.3.34 `const std::string HepPDT::ParticleID::PDName () const [inline]`

standard particle name

Definition at line 150 of file `ParticleID.hh`.

References `HepPID::particleName()`.

Referenced by `HepPDT::CommonParticleData::PDName()`.

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

- `ParticleID.hh`
- `ParticleID.icc`
- `ParticleID.cc`
- `quarks.cc`

7.11 HepPDT::QQAngularHelicity Struct Reference

QQ decay angular helicity information.

```
#include <QQChannel.hh>
```

Public Attributes

- int **hel**
- double **a0**
- double **a1**
- double **a2**
- double **a3**
- double **a4**
- double **a5**
- double **a6**

7.11.1 Detailed Description

QQ decay angular helicity information.

Definition at line 29 of file QQChannel.hh.

7.11.2 Member Data Documentation

7.11.2.1 int HepPDT::QQAngularHelicity::hel

Definition at line 30 of file QQChannel.hh.

Referenced by HepPDT::QQChannel::addAngHelicity().

7.11.2.2 double HepPDT::QQAngularHelicity::a0

Definition at line 31 of file QQChannel.hh.

Referenced by HepPDT::QQChannel::addAngHelicity().

7.11.2.3 double HepPDT::QQAngularHelicity::a1

Definition at line 32 of file QQChannel.hh.

Referenced by HepPDT::QQChannel::addAngHelicity().

7.11.2.4 double HepPDT::QQAngularHelicity::a2

Definition at line 33 of file QQChannel.hh.

Referenced by HepPDT::QQChannel::addAngHelicity().

7.11.2.5 double HepPDT::QQAngularHelicity::a3

Definition at line 34 of file QQChannel.hh.

Referenced by HepPDT::QQChannel::addAngHelicity().

7.11.2.6 double HepPDT::QQAngularHelicity::a4

Definition at line 35 of file QQChannel.hh.

Referenced by HepPDT::QQChannel::addAngHelicity().

7.11.2.7 double HepPDT::QQAngularHelicity::a5

Definition at line 36 of file QQChannel.hh.

Referenced by HepPDT::QQChannel::addAngHelicity().

7.11.2.8 double HepPDT::QQAngularHelicity::a6

Definition at line 37 of file QQChannel.hh.

Referenced by HepPDT::QQChannel::addAngHelicity().

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

- **QQChannel.hh**

7.12 HepPDT::QQChannel Class Reference

QQChannel (p. 120) is a temporary holder for QQ decay information.

```
#include <QQChannel.hh>
```

Public Types

- typedef std::vector< std::string > **SringVect**
- typedef std::vector< **QQhelicity** > **HelVect**
- typedef std::vector< **QQAngularHelicity** > **AngHelVect**

Public Member Functions

- **QQChannel** ()
- **~QQChannel** ()
- void **swap** (**QQChannel** &rhs)
- **QQChannel** (const **QQChannel** &orig)
- **QQChannel** & **operator=** (const **QQChannel** &rhs)
- void **addDaughter** (std::string nm)
add a decay particle to the list
- void **addHelicity** (**QQhelicity** hel)
add helicity information
- void **addAngHelicity** (**QQAngularHelicity** hel)
add angular helicity information
- void **setMatrixCode** (std::string &m)
change the decay matrix code
- void **setBranchingFraction** (double bf)
change the branching fraction
- void **setSinPhi** (double s)
change sin(phi)
- void **setMatrix** (**QQmatrix** m)
change the decay matrix information
- void **setCPTag** ()
change the CPT tag
- void **clear** ()
clear this channel
- void **addHelicity** (std::istream &thisline)
add helicity information
- void **addAngHelicity** (std::istream &thisline)

add angular helicity information

- void **parse** (std::istream &thisline)
parse a QQ decay channel line
- bool **cpt** () const
get the CPT tag
- std::string **matrixCode** () const
get the matrix code
- double **branchingFraction** () const
get the branching fraction
- double **sinPhi** () const
get sin(phi)
- std::string **daughter** (int i) const
get the name of this daughter
- **QQhelicity** **helicity** (int i) const
get helicity information
- **QQAngularHelicity** **angHelicity** (int i) const
get angular helicity information
- int **sizeDtr** () const
how many decay particles in this channel?
- int **sizeHel** () const
how many helicity entries in this channel?
- int **sizeAngHel** () const
how many angular helicity entries in this channel?
- void **write** (std::ostream &os) const
use for diagnostics

7.12.1 Detailed Description

QQChannel (p. 120) is a temporary holder for QQ decay information.

Author:

Lynn Garren

Temporarily keep QQ decay information in this class and convert it when we finish inputting QQ information.

Definition at line 56 of file QQChannel.hh.

7.12.2 Member Typedef Documentation

7.12.2.1 `typedef std::vector<std::string> HepPDT::QQChannel::SringVect`

Definition at line 60 of file QQChannel.hh.

7.12.2.2 `typedef std::vector<QQHelicity> HepPDT::QQChannel::HelVect`

Definition at line 61 of file QQChannel.hh.

7.12.2.3 `typedef std::vector<QQAngularHelicity> HepPDT::QQChannel::AngHelVect`

Definition at line 62 of file QQChannel.hh.

7.12.3 Constructor & Destructor Documentation

7.12.3.1 `HepPDT::QQChannel::QQChannel ()`

Definition at line 16 of file QQChannel.cc.

7.12.3.2 `HepPDT::QQChannel::~~QQChannel ()`

Definition at line 27 of file QQChannel.cc.

7.12.3.3 `HepPDT::QQChannel::QQChannel (const QQChannel & orig)`

Definition at line 42 of file QQChannel.cc.

7.12.4 Member Function Documentation

7.12.4.1 `void HepPDT::QQChannel::swap (QQChannel & rhs)`

Definition at line 30 of file QQChannel.cc.

References `itsAngularHelicity`, `itsBranchingFraction`, `itsCPTag`, `itsDaughters`, `itsHelicity`, `itsMatrix`, `itsMatrixCode`, `itsSinPhi`, and `HepPDT::swap()`.

Referenced by `operator=()`, and `HepPDT::swap()`.

7.12.4.2 `QQChannel & HepPDT::QQChannel::operator= (const QQChannel & rhs)`

Definition at line 53 of file QQChannel.cc.

References `swap()`.

7.12.4.3 void HepPDT::QQChannel::addDaughter (std::string *nm*) [inline]

add a decay particle to the list

Definition at line 77 of file QQChannel.hh.

7.12.4.4 void HepPDT::QQChannel::addHelicity (QQhelicity *hel*) [inline]

add helicity information

Definition at line 79 of file QQChannel.hh.

Referenced by HepPDT::buildQQDecay().

7.12.4.5 void HepPDT::QQChannel::addAngHelicity (QQAngularHelicity *hel*) [inline]

add angular helicity information

Definition at line 81 of file QQChannel.hh.

Referenced by HepPDT::buildQQDecay().

7.12.4.6 void HepPDT::QQChannel::setMatrixCode (std::string & *m*) [inline]

change the decay matrix code

Definition at line 83 of file QQChannel.hh.

7.12.4.7 void HepPDT::QQChannel::setBranchingFraction (double *bf*) [inline]

change the branching fraction

Definition at line 85 of file QQChannel.hh.

7.12.4.8 void HepPDT::QQChannel::setSinPhi (double *s*) [inline]

change sin(phi)

Definition at line 87 of file QQChannel.hh.

Referenced by HepPDT::buildQQDecay().

7.12.4.9 void HepPDT::QQChannel::setMatrix (QQmatrix *m*) [inline]

change the decay matrix information

Definition at line 89 of file QQChannel.hh.

Referenced by HepPDT::buildQQDecay().

7.12.4.10 void HepPDT::QQChannel::setCPTag () [inline]

change the CPT tag

Definition at line 91 of file QQChannel.hh.

Referenced by HepPDT::buildQQDecay().

7.12.4.11 void HepPDT::QQChannel::clear ()

clear this channel

Definition at line 60 of file QQChannel.cc.

References HepPDT::QQmatrix::a, HepPDT::QQmatrix::b, and HepPDT::QQmatrix::c.

Referenced by HepPDT::buildQQDecay().

7.12.4.12 void HepPDT::QQChannel::addHelicity (std::istream & *thisline*)

add helicity information

Definition at line 86 of file QQChannel.cc.

References HepPDT::QQhelicity::hel, and HepPDT::QQhelicity::prob.

7.12.4.13 void HepPDT::QQChannel::addAngHelicity (std::istream & *thisline*)

add angular helicity information

Definition at line 99 of file QQChannel.cc.

References HepPDT::QQAngularHelicity::a0, HepPDT::QQAngularHelicity::a1, HepPDT::QQAngularHelicity::a2, HepPDT::QQAngularHelicity::a3, HepPDT::QQAngularHelicity::a4, HepPDT::QQAngularHelicity::a5, HepPDT::QQAngularHelicity::a6, and HepPDT::QQAngularHelicity::hel.

7.12.4.14 void HepPDT::QQChannel::parse (std::istream & *thisline*)

parse a QQ decay channel line

Definition at line 107 of file QQChannel.cc.

Referenced by HepPDT::buildQQDecay().

7.12.4.15 bool HepPDT::QQChannel::cpt () const [inline]

get the CPT tag

Definition at line 104 of file QQChannel.hh.

7.12.4.16 std::string HepPDT::QQChannel::matrixCode () const [inline]

get the matrix code

Definition at line 106 of file QQChannel.hh.

Referenced by HepPDT::buildTPD().

7.12.4.17 double HepPDT::QQChannel::branchingFraction () const [inline]

get the branching fraction

Definition at line 108 of file QQChannel.hh.

Referenced by HepPDT::buildTPD().

7.12.4.18 double HepPDT::QQChannel::sinPhi () const [inline]

get sin(phi)

Definition at line 110 of file QQChannel.hh.

7.12.4.19 std::string HepPDT::QQChannel::daughter (int *i*) const [inline]

get the name of this daughter

Definition at line 112 of file QQChannel.hh.

Referenced by HepPDT::buildTPD().

7.12.4.20 QQhelicity HepPDT::QQChannel::helicity (int *i*) const [inline]

get helicity information

Definition at line 114 of file QQChannel.hh.

7.12.4.21 QQAngularHelicity HepPDT::QQChannel::angHelicity (int *i*) const [inline]

get angular helicity information

Definition at line 116 of file QQChannel.hh.

7.12.4.22 int HepPDT::QQChannel::sizeDtr () const [inline]

how many decay particles in this channel?

Definition at line 118 of file QQChannel.hh.

Referenced by HepPDT::buildTPD().

7.12.4.23 int HepPDT::QQChannel::sizeHel () const [inline]

how many helicity entries in this channel?

Definition at line 120 of file QQChannel.hh.

7.12.4.24 int HepPDT::QQChannel::sizeAngHel () const [inline]

how many angular helicity entries in this channel?

Definition at line 122 of file QQChannel.hh.

7.12.4.25 void HepPDT::QQChannel::write (std::ostream & *os*) const

use for diagnostics

Definition at line 74 of file QQChannel.cc.

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

- **QQChannel.hh**
- **QQChannel.cc**

7.13 HepPDT::QQDecay Class Reference

QQDecay (p. 127) is a temporary holder for QQ decay information.

```
#include <QQDecay.hh>
```

Public Types

- typedef std::vector< **QQChannel** > **ChannelList**

Public Member Functions

- **QQDecay** ()
- **~QQDecay** ()
- void **swap** (**QQDecay** &rhs)
- **QQDecay** (const **QQDecay** &orig)
- **QQDecay** & **operator=** (const **QQDecay** &rhs)
- void **addChannel** (**QQChannel** c)
add a decay channel
- void **makeStable** ()
declare this particle stable
- bool **isStable** () const
does this particle decay?
- **QQChannel** **channel** (int i) const
return a QQ decay channel
- void **write** (std::ostream &os) const
write decay information for diagnostics
- int **size** () const
how many decay channels does this particle have?

7.13.1 Detailed Description

QQDecay (p. 127) is a temporary holder for QQ decay information.

Author:

Lynn Garren

Temporarily keep QQ decay information in this class and convert it when we finish inputing QQ information.

Definition at line 29 of file QQDecay.hh.

7.13.2 Member Typedef Documentation

7.13.2.1 `typedef std::vector<QQChannel> HepPDT::QQDecay::ChannelList`

Definition at line 33 of file QQDecay.hh.

7.13.3 Constructor & Destructor Documentation

7.13.3.1 `HepPDT::QQDecay::QQDecay ()`

Definition at line 17 of file QQDecay.cc.

7.13.3.2 `HepPDT::QQDecay::~~QQDecay ()`

Definition at line 22 of file QQDecay.cc.

7.13.3.3 `HepPDT::QQDecay::QQDecay (const QQDecay & orig)`

Definition at line 31 of file QQDecay.cc.

7.13.4 Member Function Documentation

7.13.4.1 `void HepPDT::QQDecay::swap (QQDecay & rhs)`

Definition at line 25 of file QQDecay.cc.

References itsChannels, itsStable, and HepPDT::swap().

Referenced by operator=(), and HepPDT::swap().

7.13.4.2 `QQDecay & HepPDT::QQDecay::operator= (const QQDecay & rhs)`

Definition at line 36 of file QQDecay.cc.

References swap().

7.13.4.3 `void HepPDT::QQDecay::addChannel (QQChannel c) [inline]`

add a decay channel

Definition at line 48 of file QQDecay.hh.

Referenced by HepPDT::buildQQDecay().

7.13.4.4 `void HepPDT::QQDecay::makeStable () [inline]`

declare this particle stable

Definition at line 50 of file QQDecay.hh.

Referenced by HepPDT::buildQQDecay().

7.13.4.5 `bool HepPDT::QQDecay::isStable () const [inline]`

does this particle decay?

Definition at line 55 of file QQDecay.hh.

7.13.4.6 `QQChannel HepPDT::QQDecay::channel (int i) const [inline]`

return a QQ decay channel

Definition at line 57 of file QQDecay.hh.

Referenced by HepPDT::buildTPD().

7.13.4.7 `void HepPDT::QQDecay::write (std::ostream & os) const`

write decay information for diagnostics

Definition at line 43 of file QQDecay.cc.

7.13.4.8 `int HepPDT::QQDecay::size () const [inline]`

how many decay channels does this particle have?

Definition at line 62 of file QQDecay.hh.

Referenced by HepPDT::buildTPD().

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

- **QQDecay.hh**
- **QQDecay.cc**

7.14 HepPDT::QQhelicity Struct Reference

QQ decay helicity information.

```
#include <QQChannel.hh>
```

Public Attributes

- double **prob**
normally HELICITY has 3 values, but the documentation allows for more
- std::vector< int > **hel**

7.14.1 Detailed Description

QQ decay helicity information.

Definition at line 22 of file QQChannel.hh.

7.14.2 Member Data Documentation

7.14.2.1 double HepPDT::QQhelicity::prob

normally HELICITY has 3 values, but the documentation allows for more

Definition at line 24 of file QQChannel.hh.

Referenced by HepPDT::QQChannel::addHelicity().

7.14.2.2 std::vector<int> HepPDT::QQhelicity::hel

Definition at line 25 of file QQChannel.hh.

Referenced by HepPDT::QQChannel::addHelicity().

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

- QQChannel.hh

7.15 HepPDT::QQmatrix Struct Reference

QQ decay matrix information.

```
#include <QQChannel.hh>
```

Public Attributes

- double **a**
- double **b**
- double **c**

7.15.1 Detailed Description

QQ decay matrix information.

Definition at line 41 of file QQChannel.hh.

7.15.2 Member Data Documentation

7.15.2.1 double HepPDT::QQmatrix::a

Definition at line 42 of file QQChannel.hh.

Referenced by HepPDT::buildQQDecay(), and HepPDT::QQChannel::clear().

7.15.2.2 double HepPDT::QQmatrix::b

Definition at line 43 of file QQChannel.hh.

Referenced by HepPDT::buildQQDecay(), and HepPDT::QQChannel::clear().

7.15.2.3 double HepPDT::QQmatrix::c

Definition at line 44 of file QQChannel.hh.

Referenced by HepPDT::buildQQDecay(), and HepPDT::QQChannel::clear().

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

- **QQChannel.hh**

7.16 HepPDT::Quarks Struct Reference

constituent quarks

```
#include <ParticleID.hh>
```

Public Member Functions

- **Quarks** ()
- **Quarks** (short q1, short q2, short q3)

Public Attributes

- short **nq1**
- short **nq2**
- short **nq3**

7.16.1 Detailed Description

constituent quarks

Examples:

```
testPID.cc.
```

Definition at line 39 of file ParticleID.hh.

7.16.2 Constructor & Destructor Documentation

7.16.2.1 HepPDT::Quarks::Quarks () [inline]

Definition at line 42 of file ParticleID.hh.

7.16.2.2 HepPDT::Quarks::Quarks (short q1, short q2, short q3) [inline]

Definition at line 43 of file ParticleID.hh.

7.16.3 Member Data Documentation

7.16.3.1 short HepPDT::Quarks::nq1

Examples:

```
testPID.cc.
```

Definition at line 46 of file ParticleID.hh.

Referenced by `main()`, `HepPDT::TempParticleData::processPID()`, and `HepPDT::ParticleID::quarks()`.

7.16.3.2 short HepPDT::Quarks::nq2

Examples:

`testPID.cc.`

Definition at line 47 of file ParticleID.hh.

Referenced by `main()`, `HepPDT::TempParticleData::processPID()`, and `HepPDT::ParticleID::quarks()`.

7.16.3.3 short HepPDT::Quarks::nq3

Examples:

`testPID.cc.`

Definition at line 48 of file ParticleID.hh.

Referenced by `main()`, `HepPDT::TempParticleData::processPID()`, and `HepPDT::ParticleID::quarks()`.

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

- `ParticleID.hh`

7.17 HepPDT::ResonanceStructure Class Reference

The **ResonanceStructure** (p.134) class is holds mass and width information.

```
#include <ResonanceStructure.hh>
```

Public Member Functions

- **ResonanceStructure** (**Measurement** mass=**Measurement**(), **Measurement** width=**Measurement**(), double min=0., double max=0.)
construct from mass and total width
- virtual **~ResonanceStructure** ()
- **ResonanceStructure** (const **ResonanceStructure** &orig)
- **ResonanceStructure** & **operator=** (const **ResonanceStructure** &rhs)
- void **swap** (**ResonanceStructure** &other)
- **Measurement** const & **mass** () const
get the mass
- **Measurement** const & **totalWidth** () const
get the total width
- **Measurement** **lifetime** () const
calculate lifetime from total width
- double **lowerCutoff** () const
lower cutoff of allowed width values
- double **upperCutoff** () const
upper cutoff of allowed width values
- void **setMass** (**Measurement** const &mass)
change the mass
- void **setTotalWidth** (**Measurement** const &width)
change the total width
- void **setTotalWidthFromLifetime** (**Measurement** const <)
change the total width using a lifetime
- void **setLowerCutoff** (double cut)
change the lower cutoff of allowed width values
- void **setUpperCutoff** (double cut)
change the upper cutoff of allowed width values

7.17.1 Detailed Description

The **ResonanceStructure** (p.134) class holds mass and width information.

Author:

Lynn Garren

ResonanceStructure (p.134) contains the minimum information for a Breit-Wigner distribution about a given mass.

Definition at line 27 of file ResonanceStructure.hh.

7.17.2 Constructor & Destructor Documentation

7.17.2.1 HepPDT::ResonanceStructure::ResonanceStructure (Measurement *mass* = Measurement(), Measurement *width* = Measurement(), double *min* = 0., double *max* = 0.)

construct from mass and total width

Definition at line 13 of file ResonanceStructure.cc.

7.17.2.2 HepPDT::ResonanceStructure::~~ResonanceStructure () [virtual]

Definition at line 44 of file ResonanceStructure.cc.

7.17.2.3 HepPDT::ResonanceStructure::ResonanceStructure (const ResonanceStructure & *orig*)

Definition at line 21 of file ResonanceStructure.cc.

7.17.3 Member Function Documentation

7.17.3.1 ResonanceStructure & HepPDT::ResonanceStructure::operator= (const ResonanceStructure & *rhs*)

Definition at line 28 of file ResonanceStructure.cc.

References swap().

7.17.3.2 void HepPDT::ResonanceStructure::swap (ResonanceStructure & *other*)

Definition at line 35 of file ResonanceStructure.cc.

References itsLowerCutoff, itsMass, itsTotalWidth, itsUpperCutoff, HepPDT::swap(), and HepPDT::Measurement::swap().

Referenced by operator=(), and HepPDT::swap().

7.17.3.3 Measurement const& HepPDT::ResonanceStructure::mass () const
[inline]

get the mass

Definition at line 48 of file ResonanceStructure.hh.

Referenced by HepPDT::CommonParticleData::mass(), setMass(), and HepPDT::CommonParticleData::write().

7.17.3.4 Measurement const& HepPDT::ResonanceStructure::totalWidth () const
[inline]

get the total width

Definition at line 50 of file ResonanceStructure.hh.

Referenced by HepPDT::CommonParticleData::totalWidth(), and HepPDT::CommonParticleData::write().

7.17.3.5 Measurement HepPDT::ResonanceStructure::lifetime () const

calculate lifetime from total width

Definition at line 13 of file lifetime.cc.

References HepPDT::Measurement::sigma(), and HepPDT::Measurement::value().

Referenced by HepPDT::CommonParticleData::lifetime().

7.17.3.6 double HepPDT::ResonanceStructure::lowerCutoff () const [inline]

lower cutoff of allowed width values

Definition at line 54 of file ResonanceStructure.hh.

Referenced by HepPDT::CommonParticleData::lowerCutoff().

7.17.3.7 double HepPDT::ResonanceStructure::upperCutoff () const [inline]

upper cutoff of allowed width values

Definition at line 56 of file ResonanceStructure.hh.

Referenced by HepPDT::CommonParticleData::upperCutoff().

7.17.3.8 void HepPDT::ResonanceStructure::setMass (Measurement const & mass) [inline]

change the mass

Definition at line 61 of file ResonanceStructure.hh.

References mass().

Referenced by HepPDT::CommonParticleData::setMass().

7.17.3.9 void HepPDT::ResonanceStructure::setTotalWidth (Measurement const & *width*) [inline]

change the total width

Definition at line 63 of file ResonanceStructure.hh.

Referenced by HepPDT::CommonParticleData::setTotalWidth().

7.17.3.10 void HepPDT::ResonanceStructure::setTotalWidthFromLifetime (Measurement const & *lt*)

change the total width using a lifetime

Definition at line 47 of file ResonanceStructure.cc.

References HepPDT::Measurement::sigma(), and HepPDT::Measurement::value().

Referenced by HepPDT::CommonParticleData::setTotalWidthFromLifetime().

7.17.3.11 void HepPDT::ResonanceStructure::setLowerCutoff (double *cut*) [inline]

change the lower cutoff of allowed width values

Definition at line 67 of file ResonanceStructure.hh.

Referenced by HepPDT::CommonParticleData::setLowerCutoff().

7.17.3.12 void HepPDT::ResonanceStructure::setUpperCutoff (double *cut*) [inline]

change the upper cutoff of allowed width values

Definition at line 69 of file ResonanceStructure.hh.

Referenced by HepPDT::CommonParticleData::setUpperCutoff().

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

- **ResonanceStructure.hh**
- **lifetime.cc**
- **ResonanceStructure.cc**

7.18 HepPDT::SpinState Class Reference

The **SpinState** (p. 138) class holds spin information.

```
#include <SpinState.hh>
```

Public Member Functions

- **SpinState** (double ts=0., double spin=0., double oam=0.)
(default) constructor
- **SpinState** (const **SpinState** &orig)
- **SpinState** & **operator=** (const **SpinState** &rhs)
- void **swap** (**SpinState** &other)
- bool **operator==** (const **SpinState** &rhs) const
all three spins must match
- double **totalSpin** () const
return the total spin
- double **spin** () const
return the spin
- double **orbAngMom** () const
return the orbital angular momentum
- void **setTotalSpin** (double spin)
change the total spin
- void **setSpin** (double spin)
change the spin
- void **setOrbAngMom** (double ang)
change the orbital angular momentum

7.18.1 Detailed Description

The **SpinState** (p. 138) class holds spin information.

Author:

Lynn Garren

SpinState (p. 138) contains total spin, spin, and orbital angular momentum.

Examples:

`examMyPDT.cc.`

Definition at line 20 of file SpinState.hh.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 HepPDT::SpinState::SpinState (double *ts* = 0., double *spin* = 0., double *oam* = 0.) [inline]

(default) constructor

Definition at line 11 of file SpinState.icc.

7.18.2.2 HepPDT::SpinState::SpinState (const SpinState & *orig*) [inline]

Definition at line 17 of file SpinState.icc.

7.18.3 Member Function Documentation

7.18.3.1 SpinState & HepPDT::SpinState::operator= (const SpinState & *rhs*) [inline]

Definition at line 23 of file SpinState.icc.

References swap().

7.18.3.2 void HepPDT::SpinState::swap (SpinState & *other*) [inline]

Definition at line 30 of file SpinState.icc.

References itsOrbAngMom, itsSpin, itsTotalSpin, and HepPDT::swap().

Referenced by operator=(), HepPDT::swap(), and HepPDT::CommonParticleData::swap().

7.18.3.3 bool HepPDT::SpinState::operator==(const SpinState & *rhs*) const [inline]

all three spins must match

Definition at line 36 of file SpinState.icc.

References itsOrbAngMom, itsSpin, and itsTotalSpin.

7.18.3.4 double HepPDT::SpinState::totalSpin () const [inline]

return the total spin

Definition at line 43 of file SpinState.hh.

Referenced by HepPDT::parseEvtGenLine(), HepPDT::parseQQParticle(), and HepPDT::CommonParticleData::write().

7.18.3.5 double HepPDT::SpinState::spin () const [inline]

return the spin

Definition at line 45 of file SpinState.hh.

Referenced by HepPDT::CommonParticleData::write().

7.18.3.6 `double HepPDT::SpinState::orbAngMom () const` [inline]

return the orbital angular momentum

Definition at line 47 of file SpinState.hh.

Referenced by HepPDT::CommonParticleData::write().

7.18.3.7 `void HepPDT::SpinState::setTotalSpin (double spin)` [inline]

change the total spin

Definition at line 52 of file SpinState.hh.

Referenced by HepPDT::parseEvtGenLine(), HepPDT::parseQQParticle(), and HepPDT::TempParticleData::processPID().

7.18.3.8 `void HepPDT::SpinState::setSpin (double spin)` [inline]

change the spin

Definition at line 54 of file SpinState.hh.

Referenced by HepPDT::TempParticleData::processPID().

7.18.3.9 `void HepPDT::SpinState::setOrbAngMom (double ang)` [inline]

change the orbital angular momentum

Definition at line 56 of file SpinState.hh.

Referenced by HepPDT::TempParticleData::processPID().

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

- **SpinState.hh**
- **SpinState.icc**

7.19 HepPDT::TableBuilder Class Reference

The `TableBuilder` (p.141) class is used to construct a `ParticleDataTable` (p.100).

```
#include <TableBuilder.hh>
```

Public Member Functions

- **TableBuilder** (`ParticleDataTable` &table, `std::ostream` &str=`std::cerr`)
create `TableBuilder` (p. 141) from a `ParticleDataTable` (p. 100)
- **~TableBuilder** ()
call the `ParticleDataTable` (p. 100) conversion method upon destruction
- **TempParticleData** & **getParticleData** (`ParticleID` pid)
create a `TempParticleData` (p. 151) from a `ParticleID` (p. 109)
- **TempParticleData** & **getParticleData** (`std::string` const &name)
create a `TempParticleData` (p. 151) from a particle name
- **TempParticleData** & **getAntiParticle** (`ParticleID` pid, `const std::string` &aname)
create an antiparticle `TempParticleData` (p. 151) from a `ParticleID` (p. 109)
- **void addParticle** (`TempParticleData` const &pd)
add a `TempParticleData` (p. 151) to the map
- **void removeParticle** (`ParticleID` pid)
remove a `TempParticleData` (p. 151) from the map
- **void addAlias** (`TempAliasData` const &ad)
add alias information to the alias map
- **bool hasParticleData** (`std::string` const &name)
check to see if this particle is already defined
- **bool hasAlias** (`std::string` const &alias)
check to see if this alias is already defined
- **bool hasDefinition** (`std::string` const &def)
check to see if this particle name is already defined
- **int size** () const
get size of particle data map
- **int aliasSize** () const
get size of alias map
- **DefTable** & **definitions** ()
get the list of definitions (for `EvtGen`)

- double **definition** (`std::string const &def`)
return a parameter definition (for `EvtGen`)
- **TempAliasData & aliasData** (`std::string const &alias`)
find an entry in the alias map

7.19.1 Detailed Description

The **TableBuilder** (p.141) class is used to construct a **ParticleDataTable** (p.100).

Author:

Marc Paterno, Walter Brown, Lynn Garren

Define this class and use the add methods to define a **ParticleDataTable** (p.100). The destructor fills **ParticleDataTable** (p.100) from the information in **TableBuilder** (p.141). See the examples for user code.

Examples:

`examMyPDT.cc`, `listEvtGenNames.cc.in`, `listPDGNames.cc.in`, `listPythiaNames.cc.in`, `testHepPDT.cc`, `testReadEvtGen.cc.in`, `testReadIsajet.cc.in`, and `testReadQQ.cc.in`.

Definition at line 43 of file `TableBuilder.hh`.

7.19.2 Constructor & Destructor Documentation

7.19.2.1 `HepPDT::TableBuilder::TableBuilder (ParticleDataTable & table, std::ostream & str = std::cerr) [inline, explicit]`

create **TableBuilder** (p.141) from a **ParticleDataTable** (p.100)

Definition at line 50 of file `TableBuilder.hh`.

7.19.2.2 `HepPDT::TableBuilder::~~TableBuilder () [inline]`

call the **ParticleDataTable** (p.100) conversion method upon destruction

Definition at line 54 of file `TableBuilder.hh`.

References `HepPDT::ParticleDataTable::convertTemporaryMap()`.

7.19.3 Member Function Documentation

7.19.3.1 `TempParticleData & HepPDT::TableBuilder::getParticleData (ParticleID pid) [inline]`

create a **TempParticleData** (p.151) from a **ParticleID** (p.109)

Examples:

`examMyPDT.cc`.

Definition at line 21 of file TableBuilder.icc.

Referenced by `addData()`, `HepPDT::addEvtGenParticles()`, `HepPDT::addIsajetDecay()`, `HepPDT::addIsajetParticles()`, `addParticle()`, `HepPDT::addParticleTable()`, `HepPDT::addPDGParticles()`, `HepPDT::addPythiaParticles()`, `HepPDT::addQQParticles()`, `getAntiParticle()`, `getParticleData()`, and `HepPDT::detail::parseIsajetDecayLine()`.

7.19.3.2 TempParticleData & HepPDT::TableBuilder::getParticleData (std::string const & name) [inline]

create a **TempParticleData** (p. 151) from a particle name

Definition at line 35 of file TableBuilder.icc.

References `getParticleData()`.

7.19.3.3 TempParticleData & HepPDT::TableBuilder::getAntiParticle (ParticleID pid, const std::string & aname) [inline]

create an antiparticle **TempParticleData** (p. 151) from a **ParticleID** (p. 109)

Definition at line 47 of file TableBuilder.icc.

References `HepPDT::TempParticleData::antiparticle()`, `getParticleData()`, `HepPDT::ParticleID::pid()`, and `HepPDT::TempParticleData::tempParticleName`.

Referenced by `HepPDT::addPythiaParticles()`.

7.19.3.4 void HepPDT::TableBuilder::addParticle (TempParticleData const & pd) [inline]

add a **TempParticleData** (p. 151) to the map

Examples:

`examMyPDT.cc`.

Definition at line 83 of file TableBuilder.icc.

References `getParticleData()`, `hasParticleData()`, `HepPDT::TempParticleData::tempID`, and `HepPDT::TempParticleData::tempParticleName`.

Referenced by `addData()`, `HepPDT::addEvtGenParticles()`, and `HepPDT::addQQParticles()`.

7.19.3.5 void HepPDT::TableBuilder::removeParticle (ParticleID pid) [inline]

remove a **TempParticleData** (p. 151) from the map

Examples:

`examMyPDT.cc`.

Definition at line 69 of file TableBuilder.hh.

Referenced by `main()`.

7.19.3.6 void HepPDT::TableBuilder::addAlias (TempAliasData const & *ad*)
[inline]

add alias information to the alias map

Definition at line 91 of file TableBuilder.icc.

References HepPDT::TempAliasData::tempAlias.

Referenced by HepPDT::addEvtGenParticles().

7.19.3.7 bool HepPDT::TableBuilder::hasParticleData (std::string const & *name*)
[inline]

check to see if this particle is already defined

Definition at line 71 of file TableBuilder.icc.

Referenced by HepPDT::addEvtGenParticles(), addParticle(), HepPDT::addQQParticles(), HepPDT::findAliasDecayModel(), and HepPDT::findDecayModel().

7.19.3.8 bool HepPDT::TableBuilder::hasAlias (std::string const & *alias*) [inline]

check to see if this alias is already defined

Definition at line 77 of file TableBuilder.icc.

Referenced by HepPDT::addEvtGenParticles(), HepPDT::findAliasDecayModel(), and HepPDT::findDecayModel().

7.19.3.9 bool HepPDT::TableBuilder::hasDefinition (std::string const & *def*)
[inline]

check to see if this particle name is already defined

Definition at line 81 of file TableBuilder.hh.

References HepPDT::DefTable::hasDefinition().

Referenced by HepPDT::findAliasDecayModel(), and HepPDT::findDecayModel().

7.19.3.10 int HepPDT::TableBuilder::size () const [inline]

get size of particle data map

Definition at line 86 of file TableBuilder.hh.

Referenced by HepPDT::addEvtGenParticles(), HepPDT::addIsajetParticles(), HepPDT::addParticleTable(), HepPDT::addPDGParticles(), HepPDT::addPythiaParticles(), and HepPDT::addQQParticles().

7.19.3.11 int HepPDT::TableBuilder::aliasSize () const [inline]

get size of alias map

Definition at line 88 of file TableBuilder.hh.

Referenced by HepPDT::addEvtGenParticles().

7.19.3.12 DefTable& HepPDT::TableBuilder::definitions () [inline]

get the list of definitions (for EvtGen)

Definition at line 90 of file TableBuilder.hh.

Referenced by HepPDT::addEvtGenParticles().

7.19.3.13 double HepPDT::TableBuilder::definition (std::string const & *def*) [inline]

return a parameter definition (for EvtGen)

Definition at line 92 of file TableBuilder.hh.

References HepPDT::DefTable::definition().

Referenced by HepPDT::findAliasDecayModel(), and HepPDT::findDecayModel().

7.19.3.14 TempAliasData & HepPDT::TableBuilder::aliasData (std::string const & *alias*) [inline]

find an entry in the alias map

Definition at line 97 of file TableBuilder.icc.

Referenced by HepPDT::addEvtGenParticles().

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

- **TableBuilder.hh**
- **TableBuilder.icc**

7.20 HepPDT::TempAliasData Struct Reference

Hold Alias information from EvtGen.

```
#include <TempParticleData.hh>
```

Public Member Functions

- **TempAliasData** ()
used in a map<>

Public Attributes

- `std::string tempAlias`
the alias
- `std::string tempAliasedParticle`
the "real" particle
- `std::string tempChargeConj`
set if there is a charge conjugate alias
- **TDDlist tempAliasDecayList**
decay list for the alias

7.20.1 Detailed Description

Hold Alias information from EvtGen.

Definition at line 48 of file TempParticleData.hh.

7.20.2 Constructor & Destructor Documentation

7.20.2.1 HepPDT::TempAliasData::TempAliasData ()

used in a map<>

Definition at line 176 of file TempParticleData.cc.

7.20.3 Member Data Documentation

7.20.3.1 `std::string HepPDT::TempAliasData::tempAlias`

the alias

Definition at line 54 of file TempParticleData.hh.

Referenced by `HepPDT::TableBuilder::addAlias()`, and `HepPDT::parseEvtGenAlias()`.

7.20.3.2 `std::string` HepPDT::TempAliasData::tempAliasedParticle

the "real" particle

Definition at line 55 of file TempParticleData.hh.

Referenced by HepPDT::parseEvtGenAlias().

7.20.3.3 `std::string` HepPDT::TempAliasData::tempChargeConj

set if there is a charge conjugate alias

Definition at line 56 of file TempParticleData.hh.

Referenced by HepPDT::addEvtGenParticles().

7.20.3.4 `TDDlist` HepPDT::TempAliasData::tempAliasDecayList

decay list for the alias

Definition at line 57 of file TempParticleData.hh.

Referenced by HepPDT::findAliasDecayModel(), and HepPDT::parseEvtGenAliasDecayLine().

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

- `TempParticleData.hh`
- `TempParticleData.cc`

7.21 HepPDT::TempConstituent Struct Reference

constituent (e.g., quark) information

```
#include <TempParticleData.hh>
```

Public Member Functions

- **TempConstituent** (**ParticleID** p=**ParticleID**(0), int m=-1)

Public Attributes

- **ParticleID** tempConstituentPID
- int tempMultiplicity

7.21.1 Detailed Description

constituent (e.g., quark) information

Definition at line 25 of file TempParticleData.hh.

7.21.2 Constructor & Destructor Documentation

7.21.2.1 HepPDT::TempConstituent::TempConstituent (**ParticleID** p = **ParticleID**(0), int m = -1) [*inline*]

Definition at line 26 of file TempParticleData.hh.

7.21.3 Member Data Documentation

7.21.3.1 ParticleID HepPDT::TempConstituent::tempConstituentPID

Definition at line 28 of file TempParticleData.hh.

Referenced by HepPDT::TempParticleData::processPID().

7.21.3.2 int HepPDT::TempConstituent::tempMultiplicity

Definition at line 29 of file TempParticleData.hh.

Referenced by HepPDT::TempParticleData::processPID().

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

- **TempParticleData.hh**

7.22 HepPDT::TempDecayData Struct Reference

temporary holder for decay data

```
#include <TempParticleData.hh>
```

Public Attributes

- `std::string tempDecayName`
- `double tempBranchingFraction`
- `std::vector< std::string > tempDaughterList`
list of decay particles
- `std::vector< double > tempDecayParameters`
other decay parameters

7.22.1 Detailed Description

temporary holder for decay data

Definition at line 33 of file TempParticleData.hh.

7.22.2 Member Data Documentation

7.22.2.1 `std::string HepPDT::TempDecayData::tempDecayName`

Type of decay. Use string to hold int if necessary.

Definition at line 36 of file TempParticleData.hh.

Referenced by `HepPDT::detail::getPythiaDecay()`, and `HepPDT::detail::parseIsajetDecayLine()`.

7.22.2.2 `double HepPDT::TempDecayData::tempBranchingFraction`

Definition at line 37 of file TempParticleData.hh.

Referenced by `HepPDT::getEvtGenDecay()`, `HepPDT::detail::getPythiaDecay()`, and `HepPDT::detail::parseIsajetDecayLine()`.

7.22.2.3 `std::vector<std::string> HepPDT::TempDecayData::tempDaughterList`

list of decay particles

Definition at line 39 of file TempParticleData.hh.

Referenced by `HepPDT::getEvtGenDecay()`, `HepPDT::detail::getPythiaDecay()`, and `HepPDT::detail::parseIsajetDecayLine()`.

7.22.2.4 `std::vector<double> HepPDT::TempDecayData::tempDecayParameters`

other decay parameters

Definition at line 41 of file TempParticleData.hh.

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

- **TempParticleData.hh**

7.23 HepPDT::TempParticleData Struct Reference

temporary holder for Particle Data information

```
#include <TempParticleData.hh>
```

Public Member Functions

- **TempParticleData** ()
- **TempParticleData** (int id)
 - construct a basic **TempParticleData** (p. 151) from just the particle ID*
- **TempParticleData** (ParticleID pid)
 - construct a basic **TempParticleData** (p. 151) from just the **ParticleID** (p. 109)*
- **TempParticleData** (int id, std::string const &name, std::string const &source, int oid, double charge, **SpinState** const &Spin, **Measurement** const &mass, **Measurement** const &wid)
 - given all the information, construct a **TempParticleData** (p. 151)*
- **TempParticleData** (**TempParticleData** const &orig)
- **TempParticleData** & operator= (**TempParticleData** const &rhs)
- void swap (**TempParticleData** &other)
- **TempParticleData** antiparticle (std::string const &name)
 - given a particle definition, create an antiparticle*
- bool processPID ()
 - mutator - get spin state and constituent list from PID*

Public Attributes

- ParticleID tempID
- std::string tempParticleName
- std::string tempSource
- int tempOriginalID
- double tempCharge
- double tempColorCharge
- **SpinState** tempSpin
- **Measurement** tempMass
- **Measurement** tempWidth
- double tempLowCutoff
- double tempHighCutoff
- std::vector< **TempConstituent** > tempQuarks
- **TDDlist** tempDecayList

7.23.1 Detailed Description

temporary holder for Particle Data information

Examples:

```
examMyPDT.cc.
```

Definition at line 61 of file TempParticleData.hh.

7.23.2 Constructor & Destructor Documentation

7.23.2.1 HepPDT::TempParticleData::TempParticleData ()

used in a map<>

Definition at line 15 of file TempParticleData.cc.

7.23.2.2 HepPDT::TempParticleData::TempParticleData (int *id*) [explicit]

construct a basic **TempParticleData** (p. 151) from just the particle ID

Definition at line 32 of file TempParticleData.cc.

References processPID().

7.23.2.3 HepPDT::TempParticleData::TempParticleData (ParticleID *pid*) [explicit]

construct a basic **TempParticleData** (p. 151) from just the **ParticleID** (p. 109)

Definition at line 50 of file TempParticleData.cc.

References processPID().

7.23.2.4 HepPDT::TempParticleData::TempParticleData (int *id*, std::string const & *name*, std::string const & *source*, int *oid*, double *charge*, SpinState const & *Spin*, Measurement const & *mass*, Measurement const & *wid*)

given all the information, construct a **TempParticleData** (p. 151)

Definition at line 68 of file TempParticleData.cc.

7.23.2.5 HepPDT::TempParticleData::TempParticleData (TempParticleData const & *orig*)

Definition at line 106 of file TempParticleData.cc.

7.23.3 Member Function Documentation

7.23.3.1 TempParticleData & HepPDT::TempParticleData::operator= (TempParticleData const & *rhs*)

Definition at line 122 of file TempParticleData.cc.

References swap().

7.23.3.2 void HepPDT::TempParticleData::swap (TempParticleData & other)

Definition at line 129 of file TempParticleData.cc.

References HepPDT::swap(), tempCharge, tempColorCharge, tempDecayList, tempHighCutoff, tempID, tempLowCutoff, tempMass, tempOriginalID, tempParticleName, tempQuarks, tempSource, tempSpin, and tempWidth.

Referenced by operator=(), and HepPDT::swap().

7.23.3.3 TempParticleData HepPDT::TempParticleData::antiparticle (std::string const & name)

given a particle definition, create an antiparticle

Definition at line 87 of file TempParticleData.cc.

References HepPDT::ParticleID::pid(), processPID(), tempCharge, tempColorCharge, tempDecayList, tempHighCutoff, tempID, tempLowCutoff, tempMass, tempOriginalID, tempParticleName, tempSource, and tempWidth.

Referenced by HepPDT::TableBuilder::getAntiParticle().

7.23.3.4 bool HepPDT::TempParticleData::processPID ()

– mutator - get spin state and constituent list from PID

Definition at line 146 of file TempParticleData.cc.

References HepPDT::ParticleID::jSpin(), HepPDT::ParticleID::lSpin(), HepPDT::Quarks::nq1, HepPDT::Quarks::nq2, HepPDT::Quarks::nq3, HepPDT::ParticleID::pid(), HepPDT::ParticleID::quarks(), HepPDT::SpinState::setOrbAngMom(), HepPDT::SpinState::setSpin(), HepPDT::SpinState::setTotalSpin(), HepPDT::spinitod(), HepPDT::ParticleID::sSpin(), tempCharge, HepPDT::TempConstituent::tempConstituentPID, tempID, HepPDT::TempConstituent::tempMultiplicity, tempQuarks, tempSpin, and HepPDT::ParticleID::threeCharge().

Referenced by antiparticle(), and TempParticleData().

7.23.4 Member Data Documentation

7.23.4.1 ParticleID HepPDT::TempParticleData::tempID

Definition at line 89 of file TempParticleData.hh.

Referenced by HepPDT::TableBuilder::addParticle(), antiparticle(), HepPDT::buildTPD(), HepPDT::parseEvtGenDecayLine(), HepPDT::parseEvtGenLine(), HepPDT::detail::parseIsajetDecayLine(), HepPDT::detail::parseIsajetLine(), HepPDT::detail::parseParticleLine(), HepPDT::detail::parsePythiaDecayLine(), HepPDT::detail::parsePythiaLine(), HepPDT::parseQQParticle(), processPID(), and swap().

7.23.4.2 `std::string HepPDT::TempParticleData::tempParticleName`

Examples:

`examMyPDT.cc.`

Definition at line 90 of file `TempParticleData.hh`.

Referenced by `addData()`, `HepPDT::TableBuilder::addParticle()`, `HepPDT::addQQParticles()`, `antiparticle()`, `HepPDT::TableBuilder::getAntiParticle()`, `HepPDT::parseEvtGenLine()`, `HepPDT::detail::parseIsajetDecayLine()`, `HepPDT::detail::parseIsajetLine()`, `HepPDT::detail::parseParticleLine()`, `HepPDT::detail::parsePythiaLine()`, `HepPDT::parseQQParticle()`, and `swap()`.

7.23.4.3 `std::string HepPDT::TempParticleData::tempSource`

Definition at line 91 of file `TempParticleData.hh`.

Referenced by `HepPDT::addPDGParticles()`, `HepPDT::addPythiaParticles()`, `HepPDT::addQQParticles()`, `antiparticle()`, `HepPDT::parseEvtGenLine()`, `HepPDT::detail::parseParticleLine()`, `HepPDT::detail::parsePythiaLine()`, and `swap()`.

7.23.4.4 `int HepPDT::TempParticleData::tempOriginalID`

Definition at line 92 of file `TempParticleData.hh`.

Referenced by `HepPDT::addPythiaParticles()`, `HepPDT::addQQParticles()`, `antiparticle()`, `HepPDT::parseEvtGenLine()`, `HepPDT::detail::parseParticleLine()`, `HepPDT::detail::parsePythiaLine()`, `HepPDT::parseQQParticle()`, and `swap()`.

7.23.4.5 `double HepPDT::TempParticleData::tempCharge`

Examples:

`examMyPDT.cc.`

Definition at line 93 of file `TempParticleData.hh`.

Referenced by `addData()`, `antiparticle()`, `HepPDT::parseEvtGenLine()`, `HepPDT::detail::parseIsajetLine()`, `HepPDT::detail::parseParticleLine()`, `HepPDT::detail::parsePythiaLine()`, `HepPDT::parseQQParticle()`, `processPID()`, and `swap()`.

7.23.4.6 `double HepPDT::TempParticleData::tempColorCharge`

Definition at line 94 of file `TempParticleData.hh`.

Referenced by `antiparticle()`, `HepPDT::detail::parsePythiaLine()`, and `swap()`.

7.23.4.7 `SpinState HepPDT::TempParticleData::tempSpin`

Examples:

`examMyPDT.cc.`

Definition at line 95 of file `TempParticleData.hh`.

Referenced by `addData()`, `HepPDT::parseEvtGenLine()`, `HepPDT::detail::parseIsajetLine()`, `HepPDT::parseQQParticle()`, `processPID()`, and `swap()`.

7.23.4.8 Measurement HepPDT::TempParticleData::tempMass

Examples:

`examMyPDT.cc`.

Definition at line 96 of file `TempParticleData.hh`.

Referenced by `addData()`, `HepPDT::addPythiaParticles()`, `antiparticle()`, `HepPDT::detail::CheckPDGEntry()`, `HepPDT::parseEvtGenLine()`, `HepPDT::detail::parseIsajetLine()`, `HepPDT::detail::parseParticleLine()`, `HepPDT::detail::parsePythiaLine()`, `HepPDT::parseQQParticle()`, and `swap()`.

7.23.4.9 Measurement HepPDT::TempParticleData::tempWidth

Examples:

`examMyPDT.cc`.

Definition at line 97 of file `TempParticleData.hh`.

Referenced by `addData()`, `antiparticle()`, `HepPDT::detail::CheckPDGEntry()`, `HepPDT::parseEvtGenLine()`, `HepPDT::detail::parseParticleLine()`, `HepPDT::detail::parsePythiaLine()`, `HepPDT::parseQQParticle()`, and `swap()`.

7.23.4.10 double HepPDT::TempParticleData::tempLowCutoff

Definition at line 98 of file `TempParticleData.hh`.

Referenced by `antiparticle()`, `HepPDT::parseQQParticle()`, and `swap()`.

7.23.4.11 double HepPDT::TempParticleData::tempHighCutoff

Definition at line 99 of file `TempParticleData.hh`.

Referenced by `antiparticle()`, `HepPDT::parseEvtGenLine()`, `HepPDT::detail::parsePythiaLine()`, `HepPDT::parseQQParticle()`, and `swap()`.

7.23.4.12 std::vector<TempConstituent> HepPDT::TempParticleData::tempQuarks

Definition at line 100 of file `TempParticleData.hh`.

Referenced by `processPID()`, and `swap()`.

7.23.4.13 TDDlist HepPDT::TempParticleData::tempDecayList

Definition at line 101 of file `TempParticleData.hh`.

Referenced by `antiparticle()`, `HepPDT::buildTPD()`, `HepPDT::findDecayModel()`, `HepPDT::parseEvtGenDecayLine()`, `HepPDT::detail::parseIsajetDecayLine()`, `HepPDT::detail::parsePythiaDecayLine()`, and `swap()`.

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

- **TempParticleData.hh**
- **TempParticleData.cc**

Chapter 8

HepPDT File Documentation

8.1 addEvtGenParticles.cc File Reference

```
#include <string>
#include <algorithm>
#include <iostream>
#include <sstream>
#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/TempParticleData.hh"
```

Namespaces

- namespace **HepPDT**

Functions

- bool **HepPDT::getEvtGenLineType** (std::string <ype, int &id, std::string &name, const std::string &pdline)
- void **HepPDT::parseEvtGenLine** (TempParticleData &tpd, const std::string &pdline)
- void **HepPDT::parseEvtGenAlias** (TempAliasData &tad, const std::string &pdline)
- bool **HepPDT::parseEvtGenDecayLine** (TempParticleData &tpd, const std::string &pdline)
- bool **HepPDT::parseEvtGenAliasDecayLine** (TempAliasData &tad, const std::string &pdline)
- TempDecayData **HepPDT::getEvtGenDecay** (const std::string &pdline)
- void **HepPDT::parseEvtGenConj** (std::string &cname, const std::string &pdline)
- void **HepPDT::parseEvtGenDefinition** (std::string &def, double &val, const std::string &pdline)
- void **HepPDT::findDecayModel** (TempParticleData &tpd, TableBuilder &tb)
- void **HepPDT::findAliasDecayModel** (TempAliasData &tad, TableBuilder &tb)
- bool **HepPDT::addEvtGenParticles** (std::istream &pdfile, TableBuilder &tb)

read EvtGen input and add particles to the table

8.2 addHerwigParticles.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/TableBuilder.hh"
```

Namespaces

- namespace **HepPDT**

Functions

- bool **HepPDT::addHerwigParticles** (std::istream &pdfile, TableBuilder &tb)

8.3 addIsajetParticles.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/TempParticleData.hh"  
#include "HepPDT/TableBuilder.hh"
```

Namespaces

- namespace **HepPDT**
- namespace **HepPDT::detail**

Functions

- bool **HepPDT::addIsajetParticles** (std::istream &pdfile, TableBuilder &tb)
read Isajet particle input and add particles to the table
- bool **HepPDT::addIsajetDecay** (std::istream &pdfile, TableBuilder &tb)
read Isajet decay input and add decay information to the table
- void **HepPDT::detail::parseIsajetLine** (TempParticleData &tpd, const std::string &pdline)
for internal use
- void **HepPDT::detail::parseIsajetDecayLine** (TempParticleData &tpd, const std::string &pdline, TableBuilder &tb)
for internal use

8.4 addParticleTable.cc File Reference

```
#include <iostream>
#include <string>
#include <sstream>
#include "HepPDT/defs.h"
#include "HepPDT/TempParticleData.hh"
#include "HepPDT/TableBuilder.hh"
#include "HepPID/ParticleName.hh"
```

Namespaces

- namespace **HepPDT**
- namespace **HepPDT::detail**

Functions

- bool **HepPDT::addParticleTable** (std::istream &pfile, TableBuilder &tb)
read particle.tbl (or something similar) and add particles to the table
- bool **HepPDT::detail::getParticleID** (int &id, const std::string &pdlne)
for internal use
- void **HepPDT::detail::parseParticleLine** (TempParticleData &tpd, const std::string &pdlne)
for internal use

8.5 addPDGParticles.cc File Reference

```
#include <iostream>
#include <string>
#include <vector>
#include <cmath>
#include <sstream>
#include "HepPDT/defs.h"
#include "HepPDT/TempParticleData.hh"
#include "HepPDT/TableBuilder.hh"
```

Namespaces

- namespace **HepPDT**
- namespace **HepPDT::detail**

Functions

- bool **HepPDT::addPDGParticles** (std::istream &pdfile, TableBuilder &tb)
read PDG input and add particles to the table
- void **HepPDT::detail::parsePDGline** (TempParticleData &tpd, std::string &pdline)
for internal use
- bool **HepPDT::detail::CheckPDGEntry** (TempParticleData &tpd, const std::string &, double, double)
for internal use

8.6 addPythiaParticles.cc File Reference

```
#include <iostream>
#include <string>
#include <sstream>
#include "HepPDT/defs.h"
#include "HepPDT/TempParticleData.hh"
#include "HepPDT/TableBuilder.hh"
```

Namespaces

- namespace **HepPDT**
- namespace **HepPDT::detail**

Functions

- bool **HepPDT::addPythiaParticles** (std::istream &pfile, TableBuilder &tb)
read Pythia input and add particles to the table
- void **HepPDT::detail::parsePythiaLine** (TempParticleData &tpd, int &anti, std::string &aname, const std::string &pdline)
for internal use
- void **HepPDT::detail::parsePythiaDecayLine** (TempParticleData &tpd, const std::string &pdline)
for internal use
- TempDecayData **HepPDT::detail::getPythiaDecay** (const std::string &pdline)
for internal use

8.7 addQQParticles.cc File Reference

```
#include <sstream>
#include <string>
#include <iostream>
#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/TempParticleData.hh"
#include "HepPID/ParticleIDTranslations.hh"
#include "HepPDT/qqDecay.hh"
#include "HepPDT/qqChannel.hh"
```

Namespaces

- namespace **HepPDT**

Functions

- bool **HepPDT::getQQLineType** (std::string <ype, int &id, std::string &name, const std::string &pdline)
- bool **HepPDT::parseQQDecayLine** (const std::string &pdline, QQDecay &qd)
- void **HepPDT::parseQQParticle** (TempParticleData &tpd, const std::string &pdline)
- void **HepPDT::buildQQDecay** (std::istringstream &thisline, std::string <ype, QQDecay &dcy, QQChannel &ch)
- void **HepPDT::buildTPD** (TempParticleData &tpd, QQDecay &qd)
- bool **HepPDT::addQQParticles** (std::istream &pdfile, TableBuilder &tb)
read QQ input and add particles to the table

8.8 calculateWidthFromLifetime.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/TempParticleData.hh"
```

Namespaces

- namespace **HepPDT**

Functions

- double **HepPDT::calculateWidthFromLifetime** (double)

8.9 CommonParticleData.hh File Reference

```
#include <string>
#include <vector>
#include "HepPDT/ParticleID.hh"
#include "HepPDT/SpinState.hh"
#include "HepPDT/Constituent.hh"
#include "HepPDT/ResonanceStructure.hh"
#include "HepPDT/TempParticleData.hh"
#include "HepPDT/CommonParticleData.icc"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::CommonParticleData**
*The **CommonParticleData** (p. 65) class holds basic particle data.*

Functions

- void **HepPDT::swap** (CommonParticleData &first, CommonParticleData &second)

8.10 CommonParticleData.icc File Reference

```
#include <algorithm>
```

Namespaces

- namespace **HepPDT**

8.11 Constituent.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/Constituent.hh"
```

Namespaces

- namespace **HepPDT**

8.12 Constituent.hh File Reference

```
#include <algorithm>
#include "HepPDT/ParticleID.hh"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::Constituent**
*The **Constituent** (p. 76) class has information about constituent particles.*

Functions

- void **HepPDT::swap** (Constituent &first, Constituent &second)

8.13 convertTemporaryMap.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/ParticleID.hh"  
#include "HepPDT/DecayData.hh"  
#include "HepPDT/DecayChannel.hh"  
#include "HepPDT/ParticleDataTable.hh"  
#include "HepPDT/ParticleData.hh"  
#include "HepPDT/TempParticleData.hh"  
#include "HepPDT/Version.hh"
```

Namespaces

- namespace **HepPDT**

8.14 DecayChannel.cc File Reference

```
#include <algorithm>
#include "HepPDT/defs.h"
#include "HepPDT/DecayChannel.hh"
#include "HepPDT/ParticleData.hh"
```

Namespaces

- namespace **HepPDT**

8.15 DecayChannel.hh File Reference

```
#include <vector>
#include <string>
#include "HepPDT/Measurement.hh"
#include "HepPDT/DecayChannel.icc"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::DecayChannel**
*The **DecayChannel** (p. 79) class holds particle decay information.*

Functions

- void **HepPDT::swap** (DecayChannel &first, DecayChannel &second)

8.16 DecayChannel.icc File Reference

```
#include <algorithm>
```

Namespaces

- namespace **HepPDT**

8.17 DecayData.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/DecayData.hh"
```

Namespaces

- namespace **HepPDT**

8.18 DecayData.hh File Reference

```
#include <vector>
#include "HepPDT/DecayChannel.hh"
#include "HepPDT/DecayData.icc"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::DecayData**
*The **DecayData** (p. 82) class holds particle decay information.*

Functions

- void **HepPDT::swap** (DecayData &first, DecayData &second)

8.19 DecayData.icc File Reference

Namespaces

- namespace **HepPDT**

8.20 DefTable.cc File Reference

```
#include <iostream>
#include "HepPDT/defs.h"
#include "HepPDT/DefTable.hh"
```

Namespaces

- namespace **HepPDT**

8.21 DefTable.hh File Reference

```
#include <string>
```

```
#include <map>
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::DefTable**
*The **DefTable** (p. 85) class holds EvtGen definitions.*

8.22 examListHerwig.cc File Reference

```
#include <fstream>
#include <iostream>
#include "HepPID/ParticleIDTranslations.hh"
#include "HepPID/ParticleName.hh"
#include "HepPID/Version.hh"
```

Functions

- void `list_herwig_init_` (int *nevt)
- void `list_herwig_end_` ()
- void `get_list_size_` (int *)
- void `get_herwig_name_` (int *ihwg, int *id, char *name)
- int `main` ()

8.22.1 Function Documentation

8.22.1.1 void `get_herwig_name_` (int * *ihwg*, int * *id*, char * *name*)

Examples:

`examListHerwig.cc`.

Referenced by `main()`.

8.22.1.2 void `get_list_size_` (int *)

Examples:

`examListHerwig.cc`.

Referenced by `main()`.

8.22.1.3 void `list_herwig_end_` ()

Examples:

`examListHerwig.cc`.

Referenced by `main()`.

8.22.1.4 void `list_herwig_init_` (int * *nevt*)

Author:

Lynn Garren

list Herwig particle ID translations Get ID list directly from Herwig

When mixing C++ and Fortran, the main program must be C++

Examples:

examListHerwig.cc.

Referenced by main().

8.22.1.5 int main ()

Definition at line 29 of file examListHerwig.cc.

References `get_herwig_name_()`, `get_list_size_()`, `list_herwig_end_()`, `list_herwig_init_()`, `HepPID::particleName()`, `HepPID::translateHerwigtoPDT()`, and `HepPID::writeVersion()`.

8.23 examListIsajet.cc File Reference

```
#include <fstream>
#include <iostream>
#include "HepPID/ParticleIDTranslations.hh"
#include "HepPID/ParticleName.hh"
#include "HepPID/Version.hh"
```

Functions

- void `list_isajet_init_()`
- void `flavor_(int *, int *, int *, int *, int *, int *)`
- void `get_label_(int *id, char *name)`
- int `main()`

8.23.1 Function Documentation

8.23.1.1 void `flavor_(int *, int *, int *, int *, int *, int *)`

Examples:

`examListIsajet.cc`.

Referenced by `main()`.

8.23.1.2 void `get_label_(int * id, char * name)`

Examples:

`examListIsajet.cc`.

Referenced by `main()`.

8.23.1.3 void `list_isajet_init_()`

Author:

Lynn Garren

List the isajet particle ID translations Get ID list directly from isajet

When mixing C++ and Fortran, the main program must be C++

Examples:

`examListIsajet.cc`.

Referenced by `main()`.

8.23.1.4 int main ()

Definition at line 27 of file examListIsajet.cc.

References `flavor_()`, `get_label_()`, `list_isajet_init_()`, `HepPID::particleName()`, `HepPID::translateIsajettoPDT()`, and `HepPID::writeVersion()`.

8.24 examListPythia.cc File Reference

```
#include <fstream>
#include <string>
#include "HepPID/ParticleIDTranslations.hh"
#include "HepPID/ParticleIDMethods.hh"
#include "HepPID/ParticleName.hh"
#include "HepPID/Version.hh"
```

Functions

- void `list_pythia_` ()
- void `getkf_` (int *, int *)
- void `getpyname_` (int *, int *, char *name)
- void `writeLine` (int &i, int &kf, int &id, std::string &name, std::string &pn, std::ofstream &os)
- int `main` ()

8.24.1 Function Documentation

8.24.1.1 void `getkf_` (int *, int *)

Examples:

`examListPythia.cc.`

Referenced by `main()`.

8.24.1.2 void `getpyname_` (int *, int *, char * *name*)

Examples:

`examListPythia.cc.`

Referenced by `main()`.

8.24.1.3 void `list_pythia_` ()

Author:

Lynn Garren

list Pythia particle ID translations Get ID list directly from Pythia

When mixing C++ and Fortran, the main program must be C++

Examples:

`examListPythia.cc.`

Referenced by `main()`.

8.24.1.4 int main ()

Definition at line 31 of file examListPythia.cc.

References getkf_(), getpyname_(), HepPID::isValid(), list_pythia_(), HepPID::particleName(), HepPID::translatePythiatoPDT(), writeLine(), and HepPID::writeVersion().

8.24.1.5 void writeLine (int & i, int & kf, int & id, std::string & name, std::string & pn, std::ofstream & os)**Examples:**

examListPythia.cc.

Definition at line 78 of file examListPythia.cc.

Referenced by main().

8.25 examMyPDT.cc File Reference

```
#include "HepPDT/defs.h"
#include <fstream>
#include <string>
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
#include "HepPDT/TempParticleData.hh"
```

Functions

- void **addData** (**HepPDT::TableBuilder** &tb, std::string const &name, int const id, double const mass, double const charge, double const width, double const tspin)
- int **main** ()

8.25.1 Function Documentation

8.25.1.1 void **addData** (**HepPDT::TableBuilder** & *tb*, std::string const & *name*, int const *id*, double const *mass*, double const *charge*, double const *width*, double const *tspin*)

Author:

Lynn Garren

create a custom PDT using our own definitions and write out the resulting PDT

Examples:

examMyPDT.cc.

Definition at line 67 of file examMyPDT.cc.

References **HepPDT::TableBuilder::addParticle()**, **HepPDT::TableBuilder::getParticleData()**, **HepPDT::TempParticleData::tempCharge**, **HepPDT::TempParticleData::tempMass**, **HepPDT::TempParticleData::tempParticleName**, **HepPDT::TempParticleData::tempSpin**, and **HepPDT::TempParticleData::tempWidth**.

Referenced by **main()**.

8.25.1.2 int main ()

Examples:

examListHerwig.cc, **examListIsajet.cc**, **examListPythia.cc**, **examMyPDT.cc**, **listEvtGenNames.cc.in**, **listEvtGenTranslation.cc**, **listHerwigTranslation.cc**, **listIsajetTranslation.cc**, **listParticleNames.cc**, **listPDGNames.cc.in**, **listPDGTranslation.cc**, **listPythiaNames.cc.in**, **listPythiaTranslation.cc**, **listQQTranslation.cc**, **testHepPDT.cc**, **testPID.cc**, **testReadEvtGen.cc.in**, **testReadIsajet.cc.in**, and **testReadQQ.cc.in**.

Definition at line 23 of file examMyPDT.cc.

References `addData()`, `HepPDT::ParticleData::name()`, `HepPDT::ParticleDataTable::particle()`, `HepPDT::TableBuilder::removeParticle()`, and `HepPDT::ParticleDataTable::writeParticleData()`.

8.26 getIsajetID.cc File Reference

```
#include <iostream>
#include <string>
#include <sstream>
```

Namespaces

- namespace **HepPDT**
- namespace **HepPDT::detail**

Functions

- bool **HepPDT::detail::getIsajetID** (int &id, const std::string &pdlne)
for internal use

8.27 getPDGpid.cc File Reference

```
#include <string>
#include <vector>
#include <sstream>
#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
```

Namespaces

- namespace **HepPDT**
- namespace **HepPDT::detail**

Functions

- void **HepPDT::detail::getPDGpid** (std::vector< int > &idlist, std::string &pdlines)
for internal use
- void **HepPDT::detail::getPDGnames** (std::vector< std::string > &names, std::string &pdlines)
for internal use

8.28 getPythiaid.cc File Reference

```
#include <iostream>
#include <string>
#include <sstream>
#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
```

Namespaces

- namespace **HepPDT**
- namespace **HepPDT::detail**

Functions

- bool **HepPDT::detail::getPythiaid** (int &id, const std::string &pdline)
for internal use

8.29 hasMethods.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/CommonParticleData.hh"
```

Namespaces

- namespace **HepPDT**

8.30 lifetime.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/ResonanceStructure.hh"
```

Namespaces

- namespace **HepPDT**

8.31 list_of_examples.cc File Reference

8.32 list_of_tests.cc File Reference

8.33 listEvtGenNames.cc.in File Reference

```
#include <fstream>
#include <iostream>
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

Functions

- `int main ()`

8.33.1 Function Documentation

8.33.1.1 `int main ()`

Definition at line 15 of file listEvtGenNames.cc.in.

References `HepPDT::addEvtGenParticles()`, and `HepPDT::ParticleDataTable::writeParticle-Translation()`.

8.34 listEvtGenTranslation.cc File Reference

```
#include <fstream>
#include <iostream>
#include "HepPID/ParticleIDTranslations.hh"
```

Functions

- `int main ()`

8.34.1 Function Documentation

8.34.1.1 `int main ()`

Definition at line 14 of file listEvtGenTranslation.cc.

References `HepPID::writeEvtGenTranslation()`.

8.35 listHerwigTranslation.cc File Reference

```
#include <fstream>
#include <iostream>
#include "HepPID/ParticleIDTranslations.hh"
```

Functions

- `int main ()`

8.35.1 Function Documentation

8.35.1.1 `int main ()`

Definition at line 14 of file listHerwigTranslation.cc.

References `HepPID::writeHerwigTranslation()`.

8.36 listIsajetTranslation.cc File Reference

```
#include <fstream>
#include <iostream>
#include "HepPID/ParticleIDTranslations.hh"
```

Functions

- `int main ()`

8.36.1 Function Documentation

8.36.1.1 `int main ()`

Definition at line 14 of file listIsajetTranslation.cc.

References `HepPID::writeIsajetTranslation()`.

8.37 listParticleNames.cc File Reference

```
#include <fstream>
#include <iostream>
#include "HepPID/ParticleName.hh"
```

Functions

- `int main ()`

8.37.1 Function Documentation

8.37.1.1 `int main ()`

Definition at line 16 of file listParticleNames.cc.

References HepPID::listParticleNames().

8.38 listPDGNames.cc.in File Reference

```
#include <fstream>
#include <iostream>
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

Functions

- `int main ()`

8.38.1 Function Documentation

8.38.1.1 `int main ()`

Definition at line 15 of file listPDGNames.cc.in.

References `HepPDT::addPDGParticles()`, and `HepPDT::ParticleDataTable::writeParticle-Translation()`.

8.39 listPDGTranslation.cc File Reference

```
#include <fstream>
#include <iostream>
#include "HepPID/ParticleIDTranslations.hh"
```

Functions

- `int main ()`

8.39.1 Function Documentation

8.39.1.1 `int main ()`

Definition at line 14 of file listPDGTranslation.cc.

References `HepPID::writePDGTranslation()`.

8.40 listPythiaNames.cc.in File Reference

```
#include <fstream>
#include <iostream>
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

Functions

- `int main ()`

8.40.1 Function Documentation

8.40.1.1 `int main ()`

Definition at line 15 of file listPythiaNames.cc.in.

References `HepPDT::addPythiaParticles()`, and `HepPDT::ParticleDataTable::writeParticle-Translation()`.

8.41 listPythiaTranslation.cc File Reference

```
#include <fstream>
#include <iostream>
#include "HepPID/ParticleIDTranslations.hh"
```

Functions

- `int main ()`

8.41.1 Function Documentation

8.41.1.1 `int main ()`

Definition at line 14 of file listPythiaTranslation.cc.

References HepPID::writePythiaTranslation().

8.42 listQQTranslation.cc File Reference

```
#include <fstream>
#include <iostream>
#include "HepPID/ParticleIDTranslations.hh"
```

Functions

- `int main ()`

8.42.1 Function Documentation

8.42.1.1 `int main ()`

Definition at line 14 of file listQQTranslation.cc.

References `HepPID::writeQQTranslation()`.

8.43 Measurement.hh File Reference

```
#include "HepPDT/Measurement.icc"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::Measurement**
*The **Measurement** (p. 88) class defines a value with its error.*

Functions

- void **HepPDT::swap** (Measurement &first, Measurement &second)

8.44 Measurement.icc File Reference

```
#include <algorithm>
```

Namespaces

- namespace **HepPDT**

Functions

- double **HepPDT::NaN** ()

8.45 ParticleData.hh File Reference

```
#include <algorithm>
#include <list>
#include <string>
#include "HepPDT/ParticleID.hh"
#include "HepPDT/CommonParticleData.hh"
#include "HepPDT/Measurement.hh"
#include "HepPDT/SpinState.hh"
#include "HepPDT/ResonanceStructure.hh"
#include "HepPDT/Constituent.hh"
#include "HepPDT/TempParticleData.hh"
#include "HepPDT/DecayData.hh"
#include "HepPDT/DecayChannel.hh"
#include "HepPDT/ParticleData.icc"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::ParticleData**

*The **ParticleData** (p. 91) class holds data for a single particle in the table.*

Functions

- void **HepPDT::swap** (ParticleData &first, ParticleData &second)

8.46 ParticleData.icc File Reference

```
#include <algorithm>
```

Namespaces

- namespace **HepPDT**

8.47 ParticleDataTable.cc File Reference

```
#include <iostream>
#include <string>
#include <map>
#include <list>
#include "HepPDT/defs.h"
#include "HepPDT/ParticleDataTable.hh"
```

Namespaces

- namespace **HepPDT**

8.48 ParticleDataTable.hh File Reference

```
#include <iostream>
#include <string>
#include <map>
#include <list>
#include "HepPDT/ParticleID.hh"
#include "HepPDT/DecayData.hh"
#include "HepPDT/DecayChannel.hh"
#include "HepPDT/ParticleData.hh"
#include "HepPDT/ParticleDataTableComparison.hh"
#include "HepPDT/Version.hh"
#include "HepPDT/ParticleDataTable.icc"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::ParticleDataTable**
*The **ParticleDataTable** (p. 100) class is the core of **HepPDT** (p. 23).*

Functions

- bool **HepPDT::writePDGStream** (std::ostream &os, const ParticleDataTable &table)
- bool **HepPDT::writePythiaStream** (std::ostream &os, const ParticleDataTable &table)
- bool **HepPDT::writeHerwigStream** (std::ostream &os, const ParticleDataTable &table)
- bool **HepPDT::writeIsajetStream** (std::ostream &os, const ParticleDataTable &table)
- bool **HepPDT::writeQQStream** (std::ostream &os, const ParticleDataTable &table)
- bool **HepPDT::writeEvtGenStream** (std::ostream &os, const ParticleDataTable &table)

8.49 ParticleDataTable.icc File Reference

Namespaces

- namespace **HepPDT**

8.50 ParticleDataTableComparison.hh File Reference

```
#include "HepPDT/ParticleID.hh"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::ParticleDataTableComparison**

*The **ParticleDataTableComparison** (p. 108) class provides a utility for sorting the PDT.*

8.51 ParticleID.cc File Reference

```
#include <stdlib.h>
#include <cmath>
#include "HepPDT/defs.h"
#include "HepPDT/ParticleID.hh"
```

Namespaces

- namespace **HepPDT**

8.52 ParticleID.hh File Reference

```
#include <string>
#include <algorithm>
#include "HepPID/ParticleName.hh"
#include "HepPID/ParticleIDTranslations.hh"
#include "HepPDT/ParticleID.icc"
```

Namespaces

- namespace **HepPDT**

Classes

- struct **HepPDT::Quarks**
constituent quarks
- class **HepPDT::ParticleID**
*The **ParticleID** (p. 109) has various utilities to extract information from the particle ID.*

Enumerations

- enum **HepPDT::location** {
 HepPDT::nj = 1, **HepPDT::nq3**, **HepPDT::nq2**, **HepPDT::nq1**,
 HepPDT::nl, **HepPDT::nr**, **HepPDT::n**, **HepPDT::n8**,
 HepPDT::n9, **HepPDT::n10** }

Functions

- double **HepPDT::spinitod** (int js)
convert from $2J+1$ to the actual spin value
- int **HepPDT::spindtoi** (double spin)
convert an actual spin to $2J+1$
- void **HepPDT::swap** (ParticleID &first, ParticleID &second)

8.53 ParticleID.icc File Reference

Namespaces

- namespace **HepPDT**

8.54 ParticleIDMethods.cc File Reference

```
#include <cmath>
#include "HepPID/ParticleIDMethods.hh"
#include "HepPID/ParticleName.hh"
```

Namespaces

- namespace **HepPID**

Functions

- int **HepPID::abspid** (const int &pid)
absolute value of particle ID
- int **HepPID::extraBits** (const int &pid)
- unsigned short **HepPID::digit** (location loc, const int &pid)
return the digit at a named location in the PID
- int **HepPID::fundamentalID** (const int &pid)
return the first two digits if this is a "fundamental" particle
- int **HepPID::Z** (const int &pid)
- int **HepPID::A** (const int &pid)
- int **HepPID::lambda** (const int &pid)
- bool **HepPID::isValid** (const int &pid)
is this a valid ID?
- bool **HepPID::hasFundamentalAnti** (const int &pid)
if this is a fundamental particle, does it have a valid antiparticle?
- bool **HepPID::isMeson** (const int &pid)
is this a valid meson ID?
- bool **HepPID::isBaryon** (const int &pid)
is this a valid baryon ID?
- bool **HepPID::isDiQuark** (const int &pid)
is this a valid diquark ID?
- bool **HepPID::isHadron** (const int &pid)
is this a valid hadron ID?
- bool **HepPID::isLepton** (const int &pid)
is this a valid lepton ID?
- bool **HepPID::isNucleus** (const int &pid)
is this a valid ion ID?

- bool **HepPID::isPentaquark** (const int &pid)
is this a valid pentaquark ID?
- bool **HepPID::isSUSY** (const int &pid)
is this a valid SUSY ID?
- bool **HepPID::isRhadron** (const int &pid)
is this a valid R-hadron ID?
- bool **HepPID::hasUp** (const int &pid)
does this particle contain an up quark?
- bool **HepPID::hasDown** (const int &pid)
does this particle contain a down quark?
- bool **HepPID::hasStrange** (const int &pid)
does this particle contain a strange quark?
- bool **HepPID::hasCharm** (const int &pid)
does this particle contain a charm quark?
- bool **HepPID::hasBottom** (const int &pid)
does this particle contain a bottom quark?
- bool **HepPID::hasTop** (const int &pid)
does this particle contain a top quark?
- int **HepPID::jSpin** (const int &pid)
jSpin returns $2J+1$, where J is the total spin
- int **HepPID::sSpin** (const int &pid)
sSpin returns $2S+1$, where S is the spin
- int **HepPID::lSpin** (const int &pid)
lSpin returns $2L+1$, where L is the orbital angular momentum
- int **HepPID::threeCharge** (const int &pid)
return 3 times the charge (3 x quark charge is an int)

8.55 ParticleIDMethods.hh File Reference

Namespaces

- namespace **HepPID**

Enumerations

- enum **HepPID::location** {
HepPID::nj = 1, **HepPID::nq3**, **HepPID::nq2**, **HepPID::nq1**,
HepPID::nl, **HepPID::nr**, **HepPID::n**, **HepPID::n8**,
HepPID::n9, **HepPID::n10** }

Functions

- unsigned short **HepPID::digit** (location loc, const int &pid)
return the digit at a named location in the PID
- int **HepPID::A** (const int &pid)
- int **HepPID::Z** (const int &pid)
- int **HepPID::lambda** (const int &pid)
- int **HepPID::abspid** (const int &pid)
absolute value of particle ID
- int **HepPID::fundamentalID** (const int &pid)
return the first two digits if this is a "fundamental" particle
- bool **HepPID::hasFundamentalAnti** (const int &pid)
if this is a fundamental particle, does it have a valid antiparticle?
- int **HepPID::extraBits** (const int &pid)
- bool **HepPID::isValid** (const int &pid)
is this a valid ID?
- bool **HepPID::isMeson** (const int &pid)
is this a valid meson ID?
- bool **HepPID::isBaryon** (const int &pid)
is this a valid baryon ID?
- bool **HepPID::isDiQuark** (const int &pid)
is this a valid diquark ID?
- bool **HepPID::isHadron** (const int &pid)
is this a valid hadron ID?
- bool **HepPID::isLepton** (const int &pid)
is this a valid lepton ID?

- bool **HepPID::isNucleus** (const int &pid)
is this a valid ion ID?
- bool **HepPID::isPentaquark** (const int &pid)
is this a valid pentaquark ID?
- bool **HepPID::isSUSY** (const int &pid)
is this a valid SUSY ID?
- bool **HepPID::isRhadron** (const int &pid)
is this a valid R-hadron ID?
- bool **HepPID::hasUp** (const int &pid)
does this particle contain an up quark?
- bool **HepPID::hasDown** (const int &pid)
does this particle contain a down quark?
- bool **HepPID::hasStrange** (const int &pid)
does this particle contain a strange quark?
- bool **HepPID::hasCharm** (const int &pid)
does this particle contain a charm quark?
- bool **HepPID::hasBottom** (const int &pid)
does this particle contain a bottom quark?
- bool **HepPID::hasTop** (const int &pid)
does this particle contain a top quark?
- int **HepPID::jSpin** (const int &pid)
jSpin returns $2J+1$, where J is the total spin
- int **HepPID::sSpin** (const int &pid)
sSpin returns $2S+1$, where S is the spin
- int **HepPID::lSpin** (const int &pid)
lSpin returns $2L+1$, where L is the orbital angular momentum
- int **HepPID::threeCharge** (const int &pid)
return 3 times the charge (3 x quark charge is an int)

8.56 ParticleIDTranslations.hh File Reference

```
#include <iostream>
```

Namespaces

- namespace **HepPID**

Functions

- int **HepPID::translateHerwigtoPDT** (const int herwigID)
translate Herwig to PDG standard
- int **HepPID::translatePDTtoHerwig** (const int pid)
translate PDG standard to Herwig
- void **HepPID::writeHerwigTranslation** (std::ostream &os)
output the translation list
- int **HepPID::translateIsajettoPDT** (const int isajetID)
translate Isajet to PDG standard
- int **HepPID::translatePDTtoIsajet** (const int pid)
translate PDG standard to Isajet
- void **HepPID::writeIsajetTranslation** (std::ostream &os)
output the translation list
- int **HepPID::translatePythiatoPDT** (const int pythiaID)
translate Pythia to PDG standard
- int **HepPID::translatePDTtoPythia** (const int pid)
translate PDG standard to Pythia
- void **HepPID::writePythiaTranslation** (std::ostream &os)
output the translation list
- int **HepPID::translateEvtGentoPDT** (const int evtGenID)
translate EvtGen to PDG standard
- int **HepPID::translatePDTtoEvtGen** (const int pid)
translate PDG standard to EvtGen
- void **HepPID::writeEvtGenTranslation** (std::ostream &os)
output the translation list
- int **HepPID::translatePDGtabletoPDT** (const int pdgID)
translate PDG table to PDG standard

- int **HepPID::translatePDTtoPDGtable** (const int pid)
translate PDG standard to PDG table
- void **HepPID::writePDGTranslation** (std::ostream &os)
output the translation list
- int **HepPID::translateQQtoPDT** (const int qqID)
translate QQ to PDG standard
- int **HepPID::translatePDTtoQQ** (const int pid)
translate PDG standard to QQ
- int **HepPID::translateQQbar** (const int id)
QQ helper function.
- int **HepPID::translateInverseQQbar** (const int id)
QQ helper function.
- void **HepPID::writeQQTranslation** (std::ostream &os)
output the translation list
- int **HepPID::translateGeanttoPDT** (const int geantID)
translate Geant3 to PDG standard
- int **HepPID::translatePDTtoGeant** (const int pid)
translate PDG standard to Geant3

8.57 ParticleName.cc File Reference

```
#include <string>
#include <map>
#include <iostream>
#include <iomanip>
#include <utility>
#include "HepPID/ParticleName.hh"
#include "HepPID/Version.hh"
```

Namespaces

- namespace **HepPID**

Typedefs

- typedef std::map< int, std::string > **HepPID::ParticleNameMap**

Functions

- ParticleNameMap const & **HepPID::ParticleNameInit** ()
- void **HepPID::writeParticleNameLine** (int i, std::ostream &os)
- ParticleNameMap const & **HepPID::getParticleNameMap** ()
access the ParticleNameMap for other purposes
- bool **HepPID::validParticleName** (const int &)
verify that this number has a valid name
- std::string **HepPID::particleName** (const int &)
*get a known **HepPID** (p. 40) Particle name*
- void **HepPID::listParticleNames** (std::ostream &os)
list all known names

8.58 ParticleName.hh File Reference

```
#include <string>
#include <map>
#include <iostream>
```

Namespaces

- namespace **HepPID**

Typedefs

- typedef std::map< int, std::string > **HepPID::ParticleNameMap**
define ParticleNameMap

Functions

- std::string **HepPID::particleName** (const int &)
get a known HepPID (p. 40) Particle name
- void **HepPID::listParticleNames** (std::ostream &os)
list all known names
- bool **HepPID::validParticleName** (const int &)
verify that this number has a valid name
- ParticleNameMap const & **HepPID::getParticleNameMap** ()
access the ParticleNameMap for other purposes

8.59 QQChannel.cc File Reference

```
#include <sstream>
#include <algorithm>
#include "HepPDT/defs.h"
#include "HepPDT/QQChannel.hh"
```

Namespaces

- namespace **HepPDT**

8.60 QQChannel.hh File Reference

```
#include <string>
#include <vector>
#include <sstream>
```

Namespaces

- namespace **HepPDT**

Classes

- struct **HepPDT::QQhelicity**
QQ decay helicity information.
- struct **HepPDT::QQAngularHelicity**
QQ decay angular helicity information.
- struct **HepPDT::QQmatrix**
QQ decay matrix information.
- class **HepPDT::QQChannel**
QQChannel (p. 120) is a temporary holder for QQ decay information.

Functions

- void **HepPDT::swap** (QQChannel &first, QQChannel &second)

8.61 QQDecay.cc File Reference

```
#include <sstream>
#include <algorithm>
#include "HepPDT/defs.h"
#include "HepPDT/qqDecay.hh"
#include "HepPDT/qqChannel.hh"
```

Namespaces

- namespace **HepPDT**

8.62 QQDecay.hh File Reference

```
#include <vector>
#include <sstream>
#include "HepPDT/defs.h"
#include "HepPDT/QQChannel.hh"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::QQDecay**
QQDecay (p. 127) is a temporary holder for QQ decay information.

Functions

- void **HepPDT::swap** (QQDecay &first, QQDecay &second)

8.63 quarks.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/ParticleID.hh"
```

Namespaces

- namespace **HepPDT**

8.64 ResonanceStructure.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/ResonanceStructure.hh"
```

Namespaces

- namespace **HepPDT**

8.65 ResonanceStructure.hh File Reference

```
#include <algorithm>
#include "HepPDT/Measurement.hh"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::ResonanceStructure**
*The **ResonanceStructure** (p. 134) class holds mass and width information.*

Functions

- void **HepPDT::swap** (ResonanceStructure &first, ResonanceStructure &second)

8.66 spindtoi.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/ParticleID.hh"
```

Namespaces

- namespace **HepPDT**

Functions

- int **HepPDT::spindtoi** (double spin)
convert an actual spin to $2J+1$

8.67 spinitod.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/ParticleID.hh"
```

Namespaces

- namespace **HepPDT**

Functions

- double **HepPDT::spinitod** (int js)
convert from $2J+1$ to the actual spin value

8.68 SpinState.hh File Reference

```
#include "HepPDT/SpinState.icc"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::SpinState**
*The **SpinState** (p. 138) class holds spin information.*

Functions

- void **HepPDT::swap** (SpinState &first, SpinState &second)

8.69 SpinState.icc File Reference

```
#include <algorithm>
```

Namespaces

- namespace **HepPDT**

8.70 stringtodouble.cc File Reference

```
#include <sstream>
#include "HepPDT/defs.h"
#include "HepPDT/stringtodouble.hh"
```

Namespaces

- namespace **HepPDT**

Functions

- double **HepPDT::stringtodouble** (std::string &numb)
extract a double from a string

8.71 stringtodouble.hh File Reference

```
#include <string>
```

Namespaces

- namespace **HepPDT**

Functions

- double **HepPDT::stringtodouble** (std::string &numb)
extract a double from a string

8.72 TableBuilder.hh File Reference

```
#include <iostream>
#include <string>
#include <map>
#include "HepPDT/ParticleDataTable.hh"
#include "HepPDT/TempParticleData.hh"
#include "HepPDT/DefTable.hh"
#include "HepPDT/stringtodouble.hh"
#include "HepPDT/TableBuilder.icc"
```

Namespaces

- namespace **HepPDT**
- namespace **HepPDT::detail**

Classes

- class **HepPDT::TableBuilder**
*The **TableBuilder** (p. 141) class is used to construct a **ParticleDataTable** (p. 100).*

Functions

- bool **HepPDT::addPDGParticles** (std::istream &pdfile, TableBuilder &tb)
read PDG input and add particles to the table
- bool **HepPDT::addPythiaParticles** (std::istream &pdfile, TableBuilder &tb)
read Pythia input and add particles to the table
- bool **HepPDT::addHerwigParticles** (std::istream &pdfile, TableBuilder &tb)
- bool **HepPDT::addIsajetParticles** (std::istream &pdfile, TableBuilder &tb)
read Isajet particle input and add particles to the table
- bool **HepPDT::addIsajetDecay** (std::istream &pdfile, TableBuilder &tb)
read Isajet decay input and add decay information to the table
- bool **HepPDT::addQQParticles** (std::istream &pdfile, TableBuilder &tb)
read QQ input and add particles to the table
- bool **HepPDT::addEvtGenParticles** (std::istream &pdfile, TableBuilder &tb)
read EvtGen input and add particles to the table
- bool **HepPDT::addParticleTable** (std::istream &pdfile, TableBuilder &tb)
read particle.tbl (or something similar) and add particles to the table

- void **HepPDT::detail::getPDGpid** (std::vector< int > &idlist, std::string &pdline)
for internal use
- void **HepPDT::detail::getPDGnames** (std::vector< std::string > &namelst, std::string &pdline)
for internal use
- void **HepPDT::detail::parsePDGline** (TempParticleData &tpd, std::string &pdline)
for internal use
- bool **HepPDT::detail::CheckPDGEntry** (TempParticleData &tpd, const std::string &, double, double)
for internal use
- bool **HepPDT::detail::getPythiaid** (int &id, const std::string &pdline)
for internal use
- void **HepPDT::detail::parsePythiaLine** (TempParticleData &tpd, int &anti, std::string &aname, const std::string &pdline)
for internal use
- void **HepPDT::detail::parsePythiaDecayLine** (TempParticleData &tpd, const std::string &pdline)
for internal use
- TempDecayData **HepPDT::detail::getPythiaDecay** (const std::string &pdline)
for internal use
- bool **HepPDT::detail::getIsajetID** (int &id, const std::string &pdline)
for internal use
- void **HepPDT::detail::parseIsajetLine** (TempParticleData &tpd, const std::string &pdline)
for internal use
- void **HepPDT::detail::parseIsajetDecayLine** (TempParticleData &tpd, const std::string &pdline, TableBuilder &tb)
for internal use
- bool **HepPDT::detail::getParticleID** (int &id, const std::string &pdline)
for internal use
- void **HepPDT::detail::parseParticleLine** (TempParticleData &tpd, const std::string &pdline)
for internal use

8.73 TableBuilder.icc File Reference

```
#include <sstream>
```

Namespaces

- namespace **HepPDT**

8.74 TempParticleData.cc File Reference

```
#include <algorithm>
#include "HepPDT/defs.h"
#include "HepPDT/TempParticleData.hh"
```

Namespaces

- namespace **HepPDT**

8.75 TempParticleData.hh File Reference

```
#include <string>
#include <vector>
#include "HepPDT/SpinState.hh"
#include "HepPDT/ParticleID.hh"
#include "HepPDT/Measurement.hh"
```

Namespaces

- namespace **HepPDT**

Classes

- struct **HepPDT::TempConstituent**
constituent (e.g., quark) information
- struct **HepPDT::TempDecayData**
temporary holder for decay data
- struct **HepPDT::TempAliasData**
Hold Alias information from EvtGen.
- struct **HepPDT::TempParticleData**
temporary holder for Particle Data information

Typedefs

- typedef `std::vector< TempDecayData >` **HepPDT::TDDlist**
useful typedef

Functions

- double **HepPDT::calculateWidthFromLifetime** (double)
- void **HepPDT::swap** (TempParticleData &first, TempParticleData &second)

8.76 testHepPDT.cc File Reference

```
#include <fstream>
#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

Functions

- `int main ()`

8.76.1 Function Documentation

8.76.1.1 `int main ()`

Definition at line 22 of file testHepPDT.cc.

References `HepPDT::addPDGParticles()`, `HepPDT::ParticleData::lowerCutoff()`, `HepPDT::ParticleDataTable::particle()`, `HepPDT::ParticleData::totalWidth()`, `HepPDT::ParticleData::upperCutoff()`, `HepPDT::ParticleData::write()`, `HepPDT::ParticleDataTable::writeParticleData()`, and `HepPDT::ParticleDataTable::writeParticleInfo()`.

8.77 testParticleIDMethods.cc File Reference

```
#include <iostream>
#include <iomanip>
#include "HepPID/ParticleIDMethods.hh"
```

Functions

- `int main ()`

8.77.1 Function Documentation

8.77.1.1 `int main ()`

Definition at line 19 of file testParticleIDMethods.cc.

References `HepPID::A()`, `HepPID::digit()`, `HepPID::extraBits()`, `HepPID::fundamentalID()`, `HepPID::isBaryon()`, `HepPID::isDiQuark()`, `HepPID::isHadron()`, `HepPID::isLepton()`, `HepPID::isMeson()`, `HepPID::isNucleus()`, `HepPID::isValid()`, `HepPID::jSpin()`, `HepPID::lambda()`, `HepPID::lSpin()`, `HepPID::n`, `HepPID::n10`, `HepPID::nj`, `HepPID::nl`, `HepPID::nq1`, `HepPID::nq2`, `HepPID::nq3`, `HepPID::nr`, `HepPID::sSpin()`, `HepPID::threeCharge()`, and `HepPID::Z()`.

8.78 testPID.cc File Reference

```
#include <iostream>
#include <iomanip>
#include "HepPDT/defs.h"
#include "HepPDT/ParticleID.hh"
```

Functions

- `int main ()`

8.78.1 Function Documentation

8.78.1.1 `int main ()`

Definition at line 20 of file testPID.cc.

References `HepPDT::n`, `HepPDT::n10`, `HepPDT::nj`, `HepPDT::nl`, `HepPDT::Quarks::nq1`, `HepPDT::nq1`, `HepPDT::Quarks::nq2`, `HepPDT::nq2`, `HepPDT::Quarks::nq3`, `HepPDT::nq3`, `HepPDT::nr`, `HepPID::nr`, and `HepPDT::spinitod()`.

8.79 testReadEvtGen.cc.in File Reference

```
#include <fstream>
#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

Functions

- `int main ()`

8.79.1 Function Documentation

8.79.1.1 `int main ()`

Definition at line 15 of file testReadEvtGen.cc.in.

References `HepPDT::addEvtGenParticles()`, and `HepPDT::ParticleDataTable::writeParticleData()`.

8.80 testReadIsajet.cc.in File Reference

```
#include <fstream>
#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

Functions

- `int main ()`

8.80.1 Function Documentation

8.80.1.1 `int main ()`

Definition at line 22 of file testReadIsajet.cc.in.

References `HepPDT::addIsajetDecay()`, `HepPDT::addIsajetParticles()`, `HepPDT::ParticleDataTable::writeParticleData()`, and `HepPDT::ParticleDataTable::writeParticleInfo()`.

8.81 testReadParticleTable.cc.in File Reference

```
#include <fstream>
#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

Functions

- `int main ()`

8.81.1 Function Documentation

8.81.1.1 `int main ()`

Definition at line 15 of file testReadParticleTable.cc.in.

References `HepPDT::addParticleTable()`, and `HepPDT::ParticleDataTable::writeParticleData()`.

8.82 testReadQQ.cc.in File Reference

```
#include <fstream>
#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

Functions

- `int main ()`

8.82.1 Function Documentation

8.82.1.1 `int main ()`

Definition at line 15 of file testReadQQ.cc.in.

References `HepPDT::addQQParticles()`, `HepPDT::ParticleDataTable::writeParticleData()`, and `HepPDT::ParticleDataTable::writeParticleTranslation()`.

8.83 translateEvtGen.cc File Reference

```
#include <map>
#include <utility>
#include "HepPID/Version.hh"
#include "HepPID/ParticleIDTranslations.hh"
#include "HepPID/ParticleIDMethods.hh"
#include "HepPID/ParticleName.hh"
```

Namespaces

- namespace **HepPID**

Typedefs

- typedef std::map< int, int > **HepPID::EvtGenPDTMap**
- typedef std::map< int, int > **HepPID::PDTEvtGenMap**

Functions

- EvtGenPDTMap const & **HepPID::getEvtGenPDTMap** ()
- PDTEvtGenMap const & **HepPID::getPDTEvtGenMap** ()
- EvtGenPDTMap const & **HepPID::EvtGenPDTMapInit** ()
- PDTEvtGenMap const & **HepPID::PDTEvtGenMapInit** ()
- EvtGenPDTMap const & **HepPID::getEvtGenPDTMap** ()
- PDTEvtGenMap const & **HepPID::getPDTEvtGenMap** ()
- int **HepPID::translateEvtGenToPDT** (const int evtGenID)
translate EvtGen to PDG standard
- int **HepPID::translatePDTtoEvtGen** (const int pid)
translate PDG standard to EvtGen
- void **HepPID::writeEvtGenTranslationLine** (int i, std::ostream &os)
- void **HepPID::writeEvtGenTranslation** (std::ostream &os)
output the translation list

8.84 translateGeanttoPDT.cc File Reference

```
#include <iostream>
#include "HepPID/Version.hh"
#include "HepPID/ParticleIDTranslations.hh"
#include "HepPID/ParticleIDMethods.hh"
```

Namespaces

- namespace **HepPID**

Defines

- `#define IDMAX 49`

Functions

- int **HepPID::translateGeanttoPDT** (const int geantID)
translate Geant3 to PDG standard

8.84.1 Define Documentation

8.84.1.1 `#define IDMAX 49`

Definition at line 16 of file translateGeanttoPDT.cc.

Referenced by HepPID::translateGeanttoPDT(), and HepPID::translatePDTtoGeant().

8.85 translateHerwig.cc File Reference

```
#include <map>
#include <utility>
#include "HepPID/Version.hh"
#include "HepPID/ParticleIDTranslations.hh"
#include "HepPID/ParticleIDMethods.hh"
#include "HepPID/ParticleName.hh"
```

Namespaces

- namespace **HepPID**

Typedefs

- typedef std::map< int, int > **HepPID::HerwigPDTMap**
- typedef std::map< int, int > **HepPID::PDTHerwigMap**

Functions

- HerwigPDTMap const & **HepPID::getHerwigPDTMap** ()
- PDTHerwigMap const & **HepPID::getPDTHerwigMap** ()
- HerwigPDTMap const & **HepPID::HerwigPDTMapInit** ()
- PDTHerwigMap const & **HepPID::PDTHerwigMapInit** ()
- HerwigPDTMap const & **HepPID::getHerwigPDTMap** ()
- PDTHerwigMap const & **HepPID::getPDTHerwigMap** ()
- int **HepPID::translateHerwigtoPDT** (const int herwigID)
translate Herwig to PDG standard
- int **HepPID::translatePDTtoHerwig** (const int pid)
translate PDG standard to Herwig
- void **HepPID::writeHerwigTranslationLine** (int i, std::ostream &os)
- void **HepPID::writeHerwigTranslation** (std::ostream &os)
output the translation list

8.86 translateIsajet.cc File Reference

```
#include <map>
#include <utility>
#include "HepPID/Version.hh"
#include "HepPID/ParticleIDTranslations.hh"
#include "HepPID/ParticleIDMethods.hh"
#include "HepPID/ParticleName.hh"
```

Namespaces

- namespace **HepPID**

Typedefs

- typedef std::map< int, int > **HepPID::IsajetPDTMap**
- typedef std::map< int, int > **HepPID::PDTIsajetMap**

Functions

- IsajetPDTMap const & **HepPID::getIsajetPDTMap** ()
- PDTIsajetMap const & **HepPID::getPDTIsajetMap** ()
- IsajetPDTMap const & **HepPID::IsajetPDTMapInit** ()
- PDTIsajetMap const & **HepPID::PDTIsajetMapInit** ()
- int **HepPID::convIsajettoPDT** (const int id)
- int **HepPID::convPDTtoIsajet** (const int id)
- IsajetPDTMap const & **HepPID::getIsajetPDTMap** ()
- PDTIsajetMap const & **HepPID::getPDTIsajetMap** ()
- int **HepPID::translateIsajettoPDT** (const int isajetID)
translate Isajet to PDG standard
- int **HepPID::translatePDTtoIsajet** (const int pid)
translate PDG standard to Isajet
- void **HepPID::writeIsajetTranslationLine** (int i, std::ostream &os)
- void **HepPID::writeIsajetTranslation** (std::ostream &os)
output the translation list

8.87 translatePDG.cc File Reference

```
#include <map>
#include <utility>
#include "HepPID/Version.hh"
#include "HepPID/ParticleIDTranslations.hh"
#include "HepPID/ParticleIDMethods.hh"
#include "HepPID/ParticleName.hh"
```

Namespaces

- namespace **HepPID**

Typedefs

- typedef std::map< int, int > **HepPID::PDGtoPDTMap**
- typedef std::map< int, int > **HepPID::PDTtoPDGMap**

Functions

- PDGtoPDTMap const & **HepPID::getPDGtoPDTMap** ()
- PDTtoPDGMap const & **HepPID::getPDTtoPDGMap** ()
- PDGtoPDTMap const & **HepPID::PDGtoPDTMapInit** ()
- PDTtoPDGMap const & **HepPID::PDTtoPDGMapInit** ()
- PDGtoPDTMap const & **HepPID::getPDGtoPDTMap** ()
- PDTtoPDGMap const & **HepPID::getPDTtoPDGMap** ()
- int **HepPID::translatePDGtabletoPDT** (const int pdgID)
translate PDG table to PDG standard
- int **HepPID::translatePDTtoPDGtable** (const int pid)
translate PDG standard to PDG table
- void **HepPID::writePDGTranslationLine** (int i, std::ostream &os)
- void **HepPID::writePDGTranslation** (std::ostream &os)
output the translation list

8.88 translatePDTtoGeant.cc File Reference

```
#include <iostream>
#include "HepPID/Version.hh"
#include "HepPID/ParticleIDTranslations.hh"
#include "HepPID/ParticleIDMethods.hh"
```

Namespaces

- namespace **HepPID**

Defines

- `#define IDMAX 49`

Functions

- int **HepPID::translatePDTtoGeant** (const int pid)
translate PDG standard to Geant3

8.88.1 Define Documentation

8.88.1.1 `#define IDMAX 49`

Definition at line 18 of file translatePDTtoGeant.cc.

8.89 translatePythia.cc File Reference

```
#include <map>
#include <utility>
#include "HepPID/Version.hh"
#include "HepPID/ParticleIDTranslations.hh"
#include "HepPID/ParticleIDMethods.hh"
#include "HepPID/ParticleName.hh"
```

Namespaces

- namespace **HepPID**

Typedefs

- typedef std::map< int, int > **HepPID::PythiaPDTMap**
- typedef std::map< int, int > **HepPID::PDTPythiaMap**

Functions

- **PythiaPDTMap** const & **HepPID::getPythiaPDTMap** ()
- **PDTPythiaMap** const & **HepPID::getPDTPythiaMap** ()
- **PythiaPDTMap** const & **HepPID::PythiaPDTMapInit** ()
- **PDTPythiaMap** const & **HepPID::PDTPythiaMapInit** ()
- **PythiaPDTMap** const & **HepPID::getPythiaPDTMap** ()
- **PDTPythiaMap** const & **HepPID::getPDTPythiaMap** ()
- int **HepPID::translatePythiatoPDT** (const int pythiaID)
translate Pythia to PDG standard
- int **HepPID::translatePDTtoPythia** (const int pid)
translate PDG standard to Pythia
- void **HepPID::writePythiaTranslationLine** (int i, std::ostream &os)
- void **HepPID::writePythiaTranslation** (std::ostream &os)
output the translation list

8.90 translateQQ.cc File Reference

```
#include <map>
#include <utility>
#include "HepPID/Version.hh"
#include "HepPID/ParticleIDTranslations.hh"
#include "HepPID/ParticleIDMethods.hh"
#include "HepPID/ParticleName.hh"
```

Namespaces

- namespace **HepPID**

Typedefs

- typedef std::map< int, int > **HepPID::QQPDTMap**
- typedef std::map< int, int > **HepPID::PDTQQMap**
- typedef std::map< int, int > **HepPID::QQbarMap**
- typedef std::map< int, int > **HepPID::InverseQQbarMap**

Functions

- QQPDTMap const & **HepPID::getQQPDTMap** ()
- PDTQQMap const & **HepPID::getPDTQQMap** ()
- QQbarMap const & **HepPID::getQQbarMap** ()
- InverseQQbarMap const & **HepPID::getInverseQQbarMap** ()
- QQPDTMap const & **HepPID::QQPDTMapInit** ()
- QQbarMap const & **HepPID::QQbarMapInit** ()
- PDTQQMap const & **HepPID::PDTQQMapInit** ()
- InverseQQbarMap const & **HepPID::InverseQQbarMapInit** ()
- QQPDTMap const & **HepPID::getQQPDTMap** ()
- PDTQQMap const & **HepPID::getPDTQQMap** ()
- QQbarMap const & **HepPID::getQQbarMap** ()
- InverseQQbarMap const & **HepPID::getInverseQQbarMap** ()
- int **HepPID::translateQQbar** (const int id)
QQ helper function.
- int **HepPID::translateInverseQQbar** (const int id)
QQ helper function.
- int **HepPID::translateQQtoPDT** (const int qqID)
translate QQ to PDG standard
- int **HepPID::translatePDTtoQQ** (const int pid)
translate PDG standard to QQ
- void **HepPID::writeQQTranslation** (std::ostream &os)

output the translation list

8.91 Version.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/Version.hh"
```

Namespaces

- namespace **HepPDT**

Functions

- `std::string HepPDT::versionName ()`
return HepPDT (p. 23) version
- `void HepPDT::version ()`
print HepPDT (p. 23) version
- `void HepPDT::writeVersion (std::ostream &os)`
write HepPDT (p. 23) version to os

8.92 Version.cc File Reference

```
#include "HepPID/Version.hh"
```

Namespaces

- namespace **HepPID**

Functions

- `std::string HepPID::versionName ()`
return HepPID (p. 40) version
- `void HepPID::version ()`
print HepPID (p. 40) version
- `void HepPID::writeVersion (std::ostream &os)`
write HepPID (p. 40) version to os

8.93 Version.hh File Reference

```
#include <string>
#include <iostream>
```

Namespaces

- namespace **HepPDT**

Functions

- void **HepPDT::version** ()
print HepPDT (p. 23) version
- void **HepPDT::writeVersion** (std::ostream &os)
write HepPDT (p. 23) version to os
- std::string **HepPDT::versionName** ()
return HepPDT (p. 23) version

8.94 Version.hh File Reference

```
#include <string>
#include <iostream>
```

Namespaces

- namespace **HepPID**

Functions

- void **HepPID::version** ()
print HepPID (p. 40) version
- void **HepPID::writeVersion** (std::ostream &os)
write HepPID (p. 40) version to os
- std::string **HepPID::versionName** ()
return HepPID (p. 40) version

8.95 write.cc File Reference

```
#include <iomanip>
#include <sstream>
#include "HepPDT/defs.h"
#include "HepPDT/CommonParticleData.hh"
```

Namespaces

- namespace **HepPDT**
- namespace **std**

Chapter 9

HepPDT Example Documentation

9.1 examListHerwig.cc

list Herwig particle ID translations

```
1 // -----
2 // examListHerwig.cc
3 //
4 // -----
13
14 #include <fstream>
15 #include <iostream>
16
17 #include "HepPID/ParticleIDTranslations.hh"
18 #include "HepPID/ParticleName.hh"
19 #include "HepPID/Version.hh"
20
21 extern "C" {
22 // these functions are defined in examListHerwigInterface.F
23     void list_herwig_init_ ( int * nevt );
24     void list_herwig_end_ ( );
25     void get_list_size_ ( int * );
26     void get_herwig_name_( int * ihwg, int * id, char *name );
27 }
28
29 int main()
30 {
31     int nevt=20;
32     int i, j, iend, isize;
33     int hid, id;
34     char cname[10];
35     std::string hname;
36     std::string pn;
37     static char outfile[] = "examListHerwig.out";
38     std::string title = "HepPID listing of Herwig translations";
39
40     // initialize herwig
41     list_herwig_init_ ( & nevt );
42
43     // open the output stream
44     std::ofstream os( outfile );
45     if( !os ) {
46         std::cout << "error opening output file" << std::endl;
47         exit(1);
48     }
49     HepPID::writeVersion(os);
```

```
50
51  get_list_size_( & isize );
52  os << "          " << title << std::endl;
53  os << " number of Herwig particles: " << isize << std::endl;
54
55  for( i=1, iend=isize+1; i<iend; ++i ) {
56      // get info from herwig
57      for( j=0; j<10; ++j) { cname[j] = '\0'; }
58      get_herwig_name_( & i, & hid, cname );
59      hname = std::string( cname );
60      id = HepPID::translateHerwigtoPDT( hid );
61      pn = HepPID::particleName( id );
62      os << "Herwig: ";
63      os.width(7);
64      os << i ;
65      os.width(12);
66      os << hid << " " << hname;
67      os << " HepPID: " ;
68      os.width(12);
69      os << id << " " << pn << std::endl;
70  }
71
72  list_herwig_end_();
73
74  return 0;
75 }
76
```

9.2 examListHerwigInterface.F

interface to some Herwig Fortran routines

```

1
2     subroutine list_herwig_init(nevt)
3 c
4 c initialization for the herwig C++ listing
5 c
6 #include "herwig65.inc"
7     integer lnhwrt,lnhrd,lnhout,lnhdcy
8     common/heplun/lnhwrt,lnhrd,lnhout,lnhdcy
9
10    external hwudat
11    integer n
12    integer istr,nevt
13 C
14 C initialize HEP logical units
15     lnhwrt=0
16     lnhrd=0
17     lnhdcy=0
18     lnhout=22
19     lhwout=lnhout
20 C open(unit=lnhout,file='examHerwigToStdHep.lpt',status='new')
21 C
22 c call hptrlsth
23 C
24     return
25     end
26
27     subroutine get_list_size( isize )
28 c return the maximum size of herwig's particle list
29 #include "herwig65.inc"
30     integer isize
31     isize = NRES
32     return
33     end
34
35     subroutine get_herwig_name( ihwg, id, name )
36 c ihwg is the index into herwig's short list
37 #include "herwig65.inc"
38     integer id, ihwg
39     character*8 name
40     id = 0
41     call HWUIDT(2,id,ihwg,name)
42     return
43     end
44
45     subroutine list_herwig_end
46     integer lnhwrt,lnhrd,lnhout,lnhdcy
47     common/heplun/lnhwrt,lnhrd,lnhout,lnhdcy
48 C---terminate elementary process
49 c call hwefin
50 C close(unit=lnhout)
51     return
52     end
53
54 C-----
55     subroutine hwabeg
56 C... user's routine for initialization
57     end
58     subroutine hwaend
59 C... user's routine for terminal calculations, histogram output, etc
60     end
61     subroutine hwanal
62 C... user's routine to analyse data from event

```

63 end

64 C-----

9.3 examListIsajet.cc

List the Isajet particle ID translations

```

1 // -----
2 // examListIsajet.cc
3 //
4 // -----
5
13
14 #include <fstream>
15 #include <iostream>
16
17 #include "HepPID/ParticleIDTranslations.hh"
18 #include "HepPID/ParticleName.hh"
19 #include "HepPID/Version.hh"
20
21 extern "C" {
22     void list_isajet_init_ ( );
23     void flavor_( int *, int *, int *, int *, int *, int * );
24     void get_label_( int * id, char *name );
25 }
26
27 int main()
28 {
29     static char outfile[] = "examListIsajet.out";
30     std::string title = "HepPID listing of Isajet translations";
31
32     // initialize isajet
33     list_isajet_init_ ( );
34
35     // open the output stream
36     std::ofstream os( outfile );
37     if( !os ) {
38         std::cout << "error opening output file" << std::endl;
39         exit(1);
40     }
41     HepPID::writeVersion(os);
42
43     os << "      " << title << std::endl;
44
45     int i, j;
46     int id, aid, fl1, fl2, fl3, js, indx;
47     int pid;
48     char cname[10];
49     char acname[10];
50     std::string hname;
51     std::string pn;
52     for( i=1; i<100005; ++i ) {
53         // make sure names are empty
54         for( j=0; j<10; ++j) { cname[j] = '\0'; }
55         for( j=0; j<10; ++j) { acname[j] = '\0'; }
56         // get info from isajet
57         id = i;
58         aid = 0;
59         flavor_( &id, &fl1, &fl2, &fl3, &js, &indx );
60         // we need both a valid index and a valid label
61         // check the label only if there is a valid translation
62         if ( indx > 0 ) {
63             get_label_( &id, cname );
64             aid = -id;
65             get_label_( &aid, acname );
66         } else {
67             id = aid = 0;
68         }
69
70         // print particle

```

```
71     if( id != 0 ) {
72         pid = HepPID::translateIsajettoPDT( id );
73         hname = std::string( cname );
74         if ( pid != 0 ) {
75             pn = HepPID::particleName( pid );
76             os << "Isajet: ";
77             os.width(10);
78             os << id << " " << hname;
79             os << " HepPID: " ;
80             os.width(12);
81             os << pid << " " << pn << std::endl;
82         } else if ( strcmp( cname, "ERR", 3 ) != 0 ) {
83             os << "Isajet: ";
84             os.width(10);
85             os << id << " with name \"" << hname;
86             os << "\" has no HepPID translation " << std::endl;
87         }
88     }
89     // print antiparticle
90     if( aid != 0 ) {
91         hname = std::string( acname );
92         pid = HepPID::translateIsajettoPDT( aid );
93         if ( pid != 0 ) {
94             pn = HepPID::particleName( pid );
95             os << "Isajet: ";
96             os.width(10);
97             os << aid << " " << hname;
98             os << " HepPID: " ;
99             os.width(12);
100            os << pid << " " << pn << std::endl;
101        } else if ( strcmp( acname, "ERR", 3 ) != 0 ) {
102            os << "Isajet: ";
103            os.width(10);
104            os << aid << " with name \"" << hname;
105            os << "\" has no HepPID translation " << std::endl;
106        }
107    }
108 }
109
110 return 0;
111 }
```

9.4 examListIsajetInterface.F

interface to some Isajet Fortran routines

```

1      subroutine list_isajet_init
2 C
3 C      JTDKY = +/- unit number for decay table file.
4 C          If it is negative, decay table is not printed.
5 C      JTEVT = +/- unit number for output event file.
6 C          If it is negative, only stable particles are written on it.
7 C      JTCOM =      unit number for command file.
8 C      JTLIS =      unit number for listing.
9 C
10     IMPLICIT NONE
11
12 C
13     integer istr,nevt,itotal
14     INTEGER JTDKY,JTEVT,JTCOM,JTLIS,IFL,ILOOP,IPRT,LOK,ILOOP2
15     INTEGER INDEC,INDEC2
16     CHARACTER*132 ISADEC
17     LOGICAL OK,DONE
18     SAVE ILOOP,JTDKY,JTEVT,JTCOM,JTLIS
19
20     EXTERNAL ALDATA
21 C
22 C          Initialize ISAJET
23 C
24     JTDKY=-1
25     JTEVT=23
26     JTCOM=21
27     JTLIS=22
28     nevt=1000
29     itotal=0
30 C point to standard decay tables
31     CALL GETENV( 'ISAJET_DIR', ISADEC )
32     IF ( ISADEC .EQ. ' ' ) THEN
33         ISADEC = 'isadecay.dat'
34     ELSE
35         INDEC = INDEX ( ISADEC , ' ' )
36         ISADEC(INDEC:INDEC+13) = '/isadecay.dat'
37         INDEC2 = INDEX ( ISADEC , ' ' ) - 1
38 c      print *,'looking for ',ISADEC(:INDEC2)
39     ENDIF
40     OPEN(UNIT=1,FILE=ISADEC,STATUS='OLD')
41     OPEN(UNIT=JTLIS,FILE='examListIsajet.lpt',STATUS='NEW')
42     CALL ISAINI(JTDKY,JTEVT,JTCOM,JTLIS)
43     IPRT=1
44 C
45 C print list of defined particles and their translations
46 C (you have to call ISAINI first)
47 C use bogus masses for 4th generation quarks so we can see their definitions
48 C
49     CALL PRTLST(JTLIS,200.,220.)
50 C
51
52     return
53     END
54
55     subroutine get_label( id, lb)
56
57     integer id
58     character*8 LB,LABEL
59
60     lb = '      '
61
62     if ( id.ne.0) then

```

```
63         lb = LABEL(id)
64     endif
65
66     return
67 end
```

9.5 examListPythia.cc

list Pythia particle ID translations

```

1 // -----
2 // examListPythia.cc
3 //
4 // -----
13
14 #include <fstream>
15 #include <string>
16
17 #include "HepPID/ParticleIDTranslations.hh"
18 #include "HepPID/ParticleIDMethods.hh"
19 #include "HepPID/ParticleName.hh"
20 #include "HepPID/Version.hh"
21
22 extern "C" {
23     void list_pythia_ ( );
24     void getkf_( int *, int * );
25     void getpyname_( int * , int * , char *name );
26 }
27
28 void writeLine( int & i, int & kf, int & id,
29               std::string & name, std::string & pn, std::ofstream & os );
30
31 int main()
32 {
33     int kf,akf,pid,apid,iok;
34     char cname[17],caname[17];
35     std::string name, aname, pn, apn;
36     const char outfile[] = "examListPythia.out";
37     std::string title = "HepPID listing of Pythia translations";
38     // open the output file
39     std::ofstream os( outfile );
40     if( !os ) {
41         std::cerr << "cannot open " << outfile << std::endl;
42         exit(-1);
43     }
44     HepPID::writeVersion(os);
45     // get Pythia listing using the fortran pyplot function
46     // write the output of pyplot to examListPythia.lpt
47     list_pythia_();
48
49     os << "      " << title << std::endl;
50
51     for(int i=1; i<501; ++i) {
52         getkf_( &i, &kf);
53         if( kf != 0 ) {
54             getpyname_(&kf, &iok, cname);
55             if( iok == 1 ) {
56                 cname[16]='\0';
57                 name = std::string( cname );
58                 pid = HepPID::translatePythiatoPDT( kf );
59                 pn = HepPID::particleName( pid );
60                 writeLine( i, kf, pid, name, pn, os );
61                 if( HepPID::isValid( -pid ) ) {
62                     akf=-kf;
63                     getpyname_(&akf, &iok, caname);
64                     if( iok == 1 ) {
65                         apid = HepPID::translatePythiatoPDT( akf );
66                         apn = HepPID::particleName( apid );
67                         caname[16]='\0';
68                         aname = std::string( caname );
69                         writeLine( i, akf, apid, aname, apn, os );
70                     }

```

```
71         }
72     }
73 }
74 }
75     return 0;
76 }
77
78 void writeLine( int & i, int & kf, int & id,
79               std::string & name, std::string & pn, std::ofstream & os )
80 {
81     os << "Pythia: ";
82     os.width(7);
83     os << i ;
84     os.width(12);
85     os << kf << " " << name;
86     os << " HepPID: " ;
87     os.width(12);
88     os << id << " " << pn << std::endl;
89 }
```

9.6 examListPythiaInterface.F

interface to some Pythia Fortran routines

```
1      subroutine list_pythia
2 C
3 C      list jetset particle definitions
4 C
5 #include "pydat1.inc"
6
7      INTEGER LNHOUT
8 C
9      lnhout=22
10     MSTU(11)=LNHOUT
11     OPEN(UNIT=LNHOUT,FILE='examListPythia.lpt',STATUS='NEW')
12 C
13 C...
14     call pylist(11)
15 C...enable pylist(12) if you need a new pythia decay table
16 C     call pylist(12)
17 C
18 C
19 c...close output file
20     CLOSE(UNIT=LNHOUT)
21     return
22     END
23
24     subroutine getkf(kc,kf)
25 #include "pydat2.inc"
26     integer kf,kc
27     kf = KCHG(KC,4)
28     return
29     END
30
31     subroutine getpname(kf,idef,chap)
32 #include "pydat2.inc"
33     integer kf
34     character*16 chap
35     integer idef
36     idef = 1
37     CALL PYNAM(KF,CHAP)
38     if(CHAP.EQ.' ') idef=0
39     return
40     END
41
42
```

9.7 examMyPDT.cc

create a custom PDT using our own definitions and write out the resulting PDT

```

1 // -----
2 // examMyPDT.cc
3 //
9
10 #include "HepPDT/defs.h"
11 #include <fstream>
12
13 #include <string>
14
15 #include "HepPDT/TableBuilder.hh"
16 #include "HepPDT/ParticleDataTable.hh"
17 #include "HepPDT/TempParticleData.hh"
18
19 void addData( HepPDT::TableBuilder& tb, std::string const & name, int const id,
20              double const mass, double const charge, double const width,
21              double const tspin );
22
23 int main()
24 {
25     const char outfile[] = "examMyPDT.out";
26     // construct empty PDT
27     HepPDT::ParticleDataTable datacol;
28     {
29         // Construct table builder
30         HepPDT::TableBuilder tb(datacol);
31         // create my own particles here
32         addData( tb, "p+", 2212, 0.938, +1.0, -1, .5 );
33         addData( tb, "d", 1, 0., -2./3, -1, .5 );
34         addData( tb, "u-", -2, 0., -1./3, -1, .5 );
35         addData( tb, "W-", -24, 80.396, -1.0, 2.06, 1.0 );
36         addData( tb, "gamma", 22, 0., 0., -1, 1.0 );
37         addData( tb, "badgamma", 122, 0., 0., -1, 1.0 );
38         tb.removeParticle( 122 );
39     } // the tb destructor fills datacol
40     std::ofstream wpdfile( outfile );
41     if( !wpdfile ) {
42         std::cerr << "cannot open " << outfile << std::endl;
43         exit(-1);
44     }
45     datacol.writeParticleData(wpdfile);
46     // access a particle
47     // you get a null pointer if you request an undefined particle
48     HepPDT::ParticleData * pd = datacol.particle( HepPDT::ParticleID(22) );
49     if( pd ) {
50         std::cout << "particle " << pd->name() << " is defined" << std::endl;
51     } else {
52         std::cout << "ERROR: particle is not in particle data table" << std::endl;
53     }
54     pd = datacol[ HepPDT::ParticleID(-24) ];
55     // we expect this next line to produce an error
56     std::cout << "the error is expected" << std::endl;
57     if( datacol[ HepPDT::ParticleID(111) ] ) {
58         std::cout << "particle " << datacol[ HepPDT::ParticleID(111) ]->name() << " is defined" << std::endl;
59     } else {
60         std::cout << "ERROR: particle " << HepPDT::ParticleID(111).pid()
61             << " is not in particle data table" << std::endl;
62     }
63
64     return 0;
65 }
66
67 void addData( HepPDT::TableBuilder& tb, std::string const & name, int const id,

```

```
68         double const mass, double const charge, double const width,  
69         double const tspin )  
70 {  
71     HepPDT::TempParticleData& tpd = tb.getParticleData( HepPDT::ParticleID( id ) );  
72     tpd.tempParticleName = name;  
73     tpd.tempCharge = charge;  
74     tpd.tempMass = HepPDT::Measurement( mass, 0. );  
75     tpd.tempSpin = HepPDT::SpinState( tspin, 0., 0. );  
76     tpd.tempWidth = HepPDT::Measurement( width, 0. );  
77     tb.addParticle( tpd );  
78 }
```

9.8 listEvtGenNames.cc.in

The **HepPDT** (p. 23) tests are also useful examples

read EvtGen table and write out translation from EvtGen to **HepPDT** (p. 23)

```

1 // -----
2 // listEvtGenNames.cc
3 // Author: Lynn Garren
4 //
5 // read EvtGen table and write out translation from EvtGen to HepPDT
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11
12 #include "HepPDT/TableBuilder.hh"
13 #include "HepPDT/ParticleDataTable.hh"
14
15 int main()
16 {
17     const char infile1[] = "@top_srcdir@/data/pdt.table";
18     const char infile2[] = "@top_srcdir@/data/DECAY.DEC";
19     const char outfile[] = "listEvtGenNames.out";
20     // open input files
21     std::ifstream pdfile1( infile1 );
22     if( !pdfile1 ) {
23         std::cerr << "cannot open " << infile1 << std::endl;
24         exit(-1);
25     }
26     std::ifstream pdfile2( infile2 );
27     if( !pdfile2 ) {
28         std::cerr << "cannot open " << infile2 << std::endl;
29         exit(-1);
30     }
31     // construct PDT
32     HepPDT::ParticleDataTable datacol( "EvtGen Table" );
33     {
34         // Construct table builder
35         HepPDT::TableBuilder tb(datacol);
36         // read the input - put as many here as you want
37         if( !addEvtGenParticles( pdfile1, tb ) ) { std::cout << "error reading EvtGen pdt file " << std::endl; }
38         if( !addEvtGenParticles( pdfile2, tb ) ) { std::cout << "error reading EvtGen decay file " << std::endl; }
39     } // the tb destructor fills datacol
40     // open output file
41     std::ofstream wpdfile( outfile );
42     if( !wpdfile ) {
43         std::cerr << "cannot open " << outfile << std::endl;
44         exit(-1);
45     }
46     // write a translation list
47     datacol.writeParticleTranslation( wpdfile );
48
49     return 0;
50 }

```

9.9 listEvtGenTranslation.cc

The **HepPDT** (p. 23) tests are also useful examples

write the **HepPID** (p. 40) EvtGen translations

```
1 // -----
2 // listEvtGenTranslation.cc
3 // Author: Lynn Garren
4 //
5 // Usage: listEvtGenTranslation
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11
12 #include "HepPID/ParticleIDTranslations.hh"
13
14 int main()
15 {
16     const char outfile[] = "listEvtGenTranslation.out";
17     // open the output file
18     std::ofstream wpdfile( outfile );
19     if( !wpdfile ) {
20         std::cerr << "cannot open " << outfile << std::endl;
21         exit(-1);
22     }
23     // write the particle names
24     HepPID::writeEvtGenTranslation( wpdfile );
25 }
```

9.10 listHerwigTranslation.cc

The **HepPDT** (p. 23) tests are also useful examples

write the **HepPID** (p. 40) Herwig translations

```
1 // -----
2 // listHerwigTranslation.cc
3 // Author: Lynn Garren
4 //
5 // Usage: listHerwigTranslation
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11
12 #include "HepPID/ParticleIDTranslations.hh"
13
14 int main()
15 {
16     const char outfile[] = "listHerwigTranslation.out";
17     // open the output file
18     std::ofstream wpdfile( outfile );
19     if( !wpdfile ) {
20         std::cerr << "cannot open " << outfile << std::endl;
21         exit(-1);
22     }
23     // write the particle names
24     HepPID::writeHerwigTranslation( wpdfile );
25 }
```

9.11 listIsajetTranslation.cc

The **HepPDT** (p. 23) tests are also useful examples

write the **HepPID** (p. 40) Isajet translations

```
1 // -----
2 // listIsajetTranslation.cc
3 // Author: Lynn Garren
4 //
5 // Usage: listIsajetTranslation
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11
12 #include "HepPID/ParticleIDTranslations.hh"
13
14 int main()
15 {
16     const char outfile[] = "listIsajetTranslation.out";
17     // open the output file
18     std::ofstream wpdfile( outfile );
19     if( !wpdfile ) {
20         std::cerr << "cannot open " << outfile << std::endl;
21         exit(-1);
22     }
23     // write the particle names
24     HepPID::writeIsajetTranslation( wpdfile );
25 }
```

9.12 listParticleNames.cc

The **HepPDT** (p. 23) tests are also useful examples

list all known **HepPID** (p. 40) particle names

```
1 // -----
2 // listParticleNames.cc
3 // Author: Lynn Garren
4 //
5 // list all known HepPID particle names
6 //
7 // Usage: listParticleNames
8 //
9 // -----
10
11 #include <fstream>
12 #include <iostream>
13
14 #include "HepPID/ParticleName.hh"
15
16 int main()
17 {
18     const char outfile[] = "listParticleNames.out";
19     // open the output file
20     std::ofstream wpdfile( outfile );
21     if( !wpdfile ) {
22         std::cerr << "cannot open " << outfile << std::endl;
23         exit(-1);
24     }
25     // write the particle names
26     HepPID::listParticleNames( wpdfile );
27 }
```

9.13 listPDGNames.cc.in

The **HepPDT** (p. 23) tests are also useful examples

read PDG table and write out translation to **HepPDT** (p. 23)

```
1 // -----
2 // listPDGNames.cc
3 // Author: Lynn Garren
4 //
5 // read PDG table and write out translation to HepPDT
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11
12 #include "HepPDT/TableBuilder.hh"
13 #include "HepPDT/ParticleDataTable.hh"
14
15 int main()
16 {
17     const char infile[] = "@top_srcdir@/data/mass_width_2006.mc";
18     const char outfile[] = "listPDGNames.out";
19     // open input file
20     std::ifstream pdfile( infile );
21     if( !pdfile ) {
22         std::cerr << "cannot open " << infile << std::endl;
23         exit(-1);
24     }
25     // construct empty PDT
26     HepPDT::ParticleDataTable datacol( "PDG Table" );
27     {
28         // Construct table builder
29         HepPDT::TableBuilder tb(datacol);
30         // read the input - put as many here as you want
31         if( !addPDGParticles( pdfile, tb ) )
32             { std::cout << "error reading PDG file " << std::endl; }
33     } // the tb destructor fills datacol
34     // open output file
35     std::ofstream wpdfile( outfile );
36     if( !wpdfile ) {
37         std::cerr << "cannot open " << outfile << std::endl;
38         exit(-1);
39     }
40     // write a translation list
41     datacol.writeParticleTranslation( wpdfile );
42
43     return 0;
44 }
```

9.14 listPDGTranslation.cc

The **HepPDT** (p. 23) tests are also useful examples

write the **HepPID** (p. 40) PDG translations

```
1 // -----
2 // listPDGTranslation.cc
3 // Author: Lynn Garren
4 //
5 // Usage: listPDGTranslation
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11
12 #include "HepPID/ParticleIDTranslations.hh"
13
14 int main()
15 {
16     const char outfile[] = "listPDGTranslation.out";
17     // open the output file
18     std::ofstream wpdfile( outfile );
19     if( !wpdfile ) {
20         std::cerr << "cannot open " << outfile << std::endl;
21         exit(-1);
22     }
23     // write the particle names
24     HepPID::writePDGTranslation( wpdfile );
25 }
```

9.15 listPythiaNames.cc.in

The **HepPDT** (p. 23) tests are also useful examples

read Pythia table and write out translation from Pythia to **HepPDT** (p. 23)

```
1 // -----
2 // listPythiaNames.cc
3 // Author: Lynn Garren
4 //
5 // read Pythia table and write out translation from pythia to HepPDT
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11
12 #include "HepPDT/TableBuilder.hh"
13 #include "HepPDT/ParticleDataTable.hh"
14
15 int main()
16 {
17     const char infile[] = "@srcdir@/listPythia.tbl";
18     const char outfile[] = "listPythiaNames.out";
19     // open input file
20     std::ifstream pdfile( infile );
21     if( !pdfile ) {
22         std::cerr << "cannot open " << infile << std::endl;
23         exit(-1);
24     }
25     // construct empty PDT
26     HepPDT::ParticleDataTable datacol( "Pythia Table" );
27     {
28         // Construct table builder
29         HepPDT::TableBuilder tb(datacol);
30         // read the input - put as many here as you want
31         if( !addPythiaParticles( pdfile, tb ) )
32             { std::cout << "error reading pythia file " << std::endl; }
33     } // the tb destructor fills datacol
34     // open output file
35     std::ofstream wpdfile( outfile );
36     if( !wpdfile ) {
37         std::cerr << "cannot open " << outfile << std::endl;
38         exit(-1);
39     }
40     // write a translation list
41     datacol.writeParticleTranslation( wpdfile );
42
43     return 0;
44 }
```

9.16 listPythiaTranslation.cc

The **HepPDT** (p. 23) tests are also useful examples

write the **HepPID** (p. 40) Pythia translations

```
1 // -----
2 // listPythiaTranslation.cc
3 // Author: Lynn Garren
4 //
5 // Usage: listPythiaTranslation
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11
12 #include "HepPID/ParticleIDTranslations.hh"
13
14 int main()
15 {
16     const char outfile[] = "listPythiaTranslation.out";
17     // open the output file
18     std::ofstream wpdfile( outfile );
19     if( !wpdfile ) {
20         std::cerr << "cannot open " << outfile << std::endl;
21         exit(-1);
22     }
23     // write the particle names
24     HepPID::writePythiaTranslation( wpdfile );
25 }
```

9.17 listQQTranslation.cc

The **HepPDT** (p. 23) tests are also useful examples

write the **HepPID** (p. 40) QQ translations

```
1 // -----
2 // listQQTranslation.cc
3 // Author: Lynn Garren
4 //
5 // Usage: listQQTranslation
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11
12 #include "HepPID/ParticleIDTranslations.hh"
13
14 int main()
15 {
16     const char outfile[] = "listQQTranslation.out";
17     // open the output file
18     std::ofstream wpdfile( outfile );
19     if( !wpdfile ) {
20         std::cerr << "cannot open " << outfile << std::endl;
21         exit(-1);
22     }
23     // write the particle names
24     HepPID::writeQQTranslation( wpdfile );
25 }
```

9.18 testHepPDT.cc

The **HepPDT** (p. 23) tests are also useful examples

Test by reading the PDG table. Get filename and location of PDG table from input stream.

```

1 // -----
2 // testHepPDT.cc
3 // Author: Lynn Garren
4 //
5 // test by reading the PDG table
6 // get filename and location of PDG table from input stream
7 //
8 // Usage: testHepPDT
9 //
10 // -----
11
12 #include <fstream>
13
14 #include "HepPDT/defs.h"
15 #include "HepPDT/TableBuilder.hh"
16 #include "HepPDT/ParticleDataTable.hh"
17
18 using std::cout;
19 using std::cerr;
20 using std::endl;
21
22 int main()
23 {
24     char pdgfile[300] = "";
25     const char outfile[] = "testHepPDT.out";
26     std::cin >> pdgfile;
27     // open input file
28     std::ifstream pdfile( pdgfile );
29     if( !pdfile ) {
30         cerr << "cannot open " << pdgfile << endl;
31         exit(-1);
32     }
33     // construct empty PDT
34     HepPDT::ParticleDataTable datacol( "PDG Table" );
35     {
36         // Construct table builder
37         HepPDT::TableBuilder tb(datacol);
38         // read the input - put as many here as you want
39         if( !HepPDT::addPDGParticles( pdfile, tb ) ) { cout << "error reading PDG file " << endl; }
40     } // the tb destructor fills datacol
41     std::ofstream wpdfile( outfile );
42     if( !wpdfile ) {
43         cerr << "cannot open " << outfile << endl;
44         exit(-1);
45     }
46     datacol.writeParticleData(wpdfile);
47     wpdfile << std::endl;
48
49     // output some pion information
50     HepPDT::ParticleData * pd;
51     pd=datacol.particle(HepPDT::ParticleID(111));
52     // test the ResonanceStructure cutoff methods here
53     if(pd) {
54         pd->write(wpdfile);
55         wpdfile << "Resonance info for 111 "
56             << pd->totalWidth().value() << " "
57             << pd->totalWidth().sigma() << " "
58             << pd->lowerCutoff() << " "
59             << pd->upperCutoff() << std::endl;
60     }
61     // -111 is an illegal particle, no info will be written

```

```
62     pd=dataacol.particle(HepPDT::ParticleID(-111));
63     if(pd) pd->write(wpdfile);
64     pd=dataacol.particle(HepPDT::ParticleID(211));
65     if(pd) pd->write(wpdfile);
66     // string lookup
67     pd=dataacol.particle(std::string("pi0"));
68     if(pd) pd->write(wpdfile);
69     // particle info
70     dataacol.writeParticleInfo(wpdfile);
71
72     return 0;
73 }
```

9.19 testPID.cc

The **HepPDT** (p. 23) tests are also useful examples

test ParticleID methods

```

1 // -----
2 // TestPID.cc
3 // Author: Lynn Garren
4 //
5 // test ParticleID
6 //
7 // -----
8
9 #include <iostream>
10 #include <iomanip>
11
12 #include "HepPDT/defs.h"
13 #include "HepPDT/ParticleID.hh"
14
15 using std::cout;
16 using std::cerr;
17 using std::endl;
18 using std::setw;
19
20 int main()
21 {
22     int id[16] = { 5, 25, 15, 213, -3214, 10213, 9050225, -200543, 129050225,
23                  2000025, 3101, 3301, -2212, 1000020040, -1000060120, 555 };
24     int it;
25     int nr, nx;
26     int chg, sid, extra;
27     int js, ls;
28     for( it=0; it < 16; it++ ) {
29         HepPDT::ParticleID pid( id[it] );
30         nx = pid.digit(HepPDT::n);
31         nr = pid.digit(HepPDT::nr);
32         extra = pid.extraBits();
33         cout << endl;
34         cout << setw(18) << id[it] << " : " << nx << " " << nr
35              << " " << pid.digit(HepPDT::nl)
36              << " " << pid.digit(HepPDT::nq1) << " "
37              << pid.digit(HepPDT::nq2) << " " << pid.digit(HepPDT::nq3)
38              << " " << pid.digit(HepPDT::nj)
39              << " extra bits " << extra << endl;
40         js = pid.jSpin();
41         HepPDT::Quarks cqks = pid.quarks( );
42         ls = pid.lSpin();
43         sid = pid.fundamentalID();
44         chg = pid.threeCharge();
45         if( !pid.isValid() ) {
46             cout << "**** Invalid PID: " << pid.pid() << " ****" << endl;
47         } else {
48             if( pid.isHadron() ) {
49                 if( pid.isMeson() ) {
50                     cout << "meson " << setw(10) << id[it] << " : " << nx
51                          << " " << nr << " " << ls << " "
52                          << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
53                          << " " << js << " " << sid << " " << chg
54                          << " extra bits " << extra << endl;
55                 } else if( pid.isBaryon() ) {
56                     cout << "baryon " << setw(10) << id[it] << " : " << nx
57                          << " " << nr << " " << ls << " "
58                          << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
59                          << " " << js << " " << sid << " " << chg
60                          << " extra bits " << extra << endl;
61                 } else {

```

```

62         cout << "**** undefined hadron: " << pid.pid()
63             << " ****" << endl;
64     }
65 }
66 if( pid.isLepton() ) {
67     cout << "lepton " << setw(10) << id[it] << ": " << nx
68         << " " << nr << " " << ls << " "
69         << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
70         << " " << js << " " << sid << " " << chg
71         << " extra bits " << extra << endl;
72 }
73 if( pid.isDiQuark() ) {
74     cout << "diquark " << setw(10) << id[it] << ": " << nx
75         << " " << nr << " " << ls << " "
76         << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
77         << " " << js << " " << sid << " " << chg
78         << " extra bits " << extra << endl;
79 }
80 if( pid.isNucleus() ) {
81     cout << "ion      " << setw(11) << id[it]
82         << ": " << pid.digit(HepPDT::n10)
83         << " " << setw(3) << pid.A()
84         << " " << setw(3) << pid.Z()
85         << " " << setw(3) << pid.lambda()
86         << " " << js << " " << sid << " " << chg
87         << " extra bits " << extra << endl;
88 }
89 if( pid.isHadron() || pid.isLepton() || pid.isDiQuark() || pid.isNucleus() ) {
90 } else {
91     cout << "unknown " << setw(10) << id[it] << ": " << nx
92         << " " << nr << " " << ls << " "
93         << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
94         << " " << js << " " << sid << " " << chg
95         << " extra bits " << extra << endl;
96 }
97 }
98 if( pid.isValid() ) {
99     js = pid.jSpin();
100    chg = pid.threeCharge();
101    cout << "total spin: " << js << " " << HepPDT::spinitod(js) ;
102    cout << " orbital angular momentum: " << pid.lSpin() ;
103    cout << " spin: " << pid.sSpin() ;
104    cout << " charge: " << chg << " " << double(chg)/3.0 << endl;
105    HepPDT::Quarks qlist = pid.quarks( );
106    cout << "quarks: " << qlist.nq1 << " " << qlist.nq2 << " " << qlist.nq3 << endl;
107 }
108 }
109 }

```

9.20 testReadEvtGen.cc.in

The **HepPDT** (p. 23) tests are also useful examples

read EvtGen table and write it out

```

1 // $Id: testReadEvtGen.cc.in,v 1.1 2007/05/22 22:12:23 garren Exp $
2 // -----
3 // testReadEvtGen.cc
4 //
5 // read EvtGen table and write it out
6 //
7 // -----
8
9 #include <fstream>
10
11 #include "HepPDT/defs.h"
12 #include "HepPDT/TableBuilder.hh"
13 #include "HepPDT/ParticleDataTable.hh"
14
15 int main()
16 {
17     const char infile1[] = "@top_srcdir@/data/pdt.table";
18     const char infile2[] = "@top_srcdir@/data/DECAY.DEC";
19     const char outfile[] = "testReadEvtGen.out";
20     // open input files
21     std::ifstream pdfile1( infile1 );
22     if( !pdfile1 ) {
23         std::cerr << "cannot open " << infile1 << std::endl;
24         exit(-1);
25     }
26     // construct empty PDT
27     std::ifstream pdfile2( infile2 );
28     if( !pdfile2 ) {
29         std::cerr << "cannot open " << infile2 << std::endl;
30         exit(-1);
31     }
32     HepPDT::ParticleDataTable datacol( "EvtGen Table" );
33     {
34         // Construct table builder
35         HepPDT::TableBuilder tb(datacol);
36         // read the input - put as many here as you want
37         if( !addEvtGenParticles( pdfile1, tb ) ) { std::cout << "error reading EvtGen pdt file " << std::endl; }
38         if( !addEvtGenParticles( pdfile2, tb ) ) { std::cout << "error reading EvtGen decay file " << std::endl; }
39     } // the tb destructor fills datacol
40     std::ofstream wfile( outfile );
41     if( !wfile ) {
42         std::cerr << "cannot open " << outfile << std::endl;
43         exit(-1);
44     }
45     datacol.writeParticleData(wfile);
46
47     return 0;
48 }

```

9.21 testReadIsajet.cc.in

The **HepPDT** (p. 23) tests are also useful examples

read the isajet particle and decay tables for testing purposes Note that isaparticles.dat was created with PRTLST(...)

```

1 // -----
2 // readIsajet.cc
3 // Author: Lynn Garren
4 //
5 // read the isajet particle and decay tables for testing purposes
6 // note that isaparticles.dat was created with PRTLST(...)
7 //
8 // Usage: readIsajet
9 //
10 // -----
11
12 #include <fstream>
13
14 #include "HepPDT/defs.h"
15 #include "HepPDT/TableBuilder.hh"
16 #include "HepPDT/ParticleDataTable.hh"
17
18 using std::cout;
19 using std::cerr;
20 using std::endl;
21
22 int main()
23 {
24     char pdgfile[300] = "@top_srcdir/examples/data/isaparticles.dat";
25     char decaydat[300] = "@top_srcdir/examples/data/isadecay.dat";
26     const char outfile[] = "testReadIsajet.out";
27     // construct empty PDT
28     HepPDT::ParticleDataTable dataacol( "Isajet Table" );
29     {
30         // open input files
31         std::ifstream pdfile( pdgfile );
32         if( !pdfile ) {
33             cerr << "cannot open " << pdgfile << endl;
34             exit(-1);
35         }
36         std::ifstream dcyfile( decaydat );
37         if( !dcyfile ) {
38             cerr << "cannot open " << pdgfile << endl;
39             exit(-1);
40         }
41         // Construct table builder
42         HepPDT::TableBuilder tb(dataacol);
43         // read the input - put as many here as you want
44         if( !HepPDT::addIsajetParticles( pdfile, tb ) ) { cout << "error reading " << pdgfile << endl; }
45         if( !HepPDT::addIsajetDecay( dcyfile, tb ) ) { cout << "error reading " << decaydat << endl; }
46     } // the tb destructor fills dataacol
47     std::ofstream wpdfile( outfile );
48     if( !wpdfile ) {
49         cerr << "cannot open " << outfile << endl;
50         exit(-1);
51     }
52     dataacol.writeParticleData(wpdfile);
53     wpdfile << std::endl;
54
55     // particle info
56     dataacol.writeParticleInfo(wpdfile);
57
58     return 0;
59 }

```

9.22 testReadQQ.cc.in

The **HepPDT** (p. 23) tests are also useful examples

read QQ table and write it out

```

1 // $Id: testReadQQ.cc.in,v 1.1 2007/05/22 22:12:23 garren Exp $
2 // -----
3 // testReadQQ.cc
4 //
5 // read QQ table and write it out
6 //
7 // -----
8
9 #include <fstream>
10
11 #include "HepPDT/defs.h"
12 #include "HepPDT/TableBuilder.hh"
13 #include "HepPDT/ParticleDataTable.hh"
14
15 int main()
16 {
17     const char infile[] = "@srcdir@/listQQ.dec";
18     const char outfile[] = "testReadQQ.out";
19     // open input file
20     std::ifstream pdfile( infile );
21     if( !pdfile ) {
22         std::cerr << "cannot open " << infile << std::endl;
23         exit(-1);
24     }
25     // construct empty PDT
26     HepPDT::ParticleDataTable datacol( "QQ Table" );
27     {
28         // Construct table builder
29         HepPDT::TableBuilder tb(datacol);
30         // read the input - put as many here as you want
31         if( !addQQParticles( pdfile, tb ) )
32             { std::cout << "error reading QQ table file " << std::endl; }
33     } // the tb destructor fills the PDT
34     std::ofstream wpdfile( outfile );
35     if( !wpdfile ) {
36         std::cerr << "cannot open " << outfile << std::endl;
37         exit(-1);
38     }
39     // write a translation list
40     datacol.writeParticleTranslation( wpdfile );
41     // write the particle and decay info
42     datacol.writeParticleData( wpdfile );
43
44     return 0;
45 }

```