

HepPDT Reference Manual

3.04.01

Generated by Doxygen 1.4.7

Fri Dec 4 14:05:23 2009

Contents

1	HepPDT Directory Hierarchy	1
1.1	HepPDT Directories	1
2	HepPDT Namespace Index	3
2.1	HepPDT Namespace List	3
3	HepPDT Hierarchical Index	5
3.1	HepPDT Class Hierarchy	5
4	HepPDT Class Index	7
4.1	HepPDT Class List	7
5	HepPDT File Index	9
5.1	HepPDT File List	9
6	HepPDT Directory Documentation	11
6.1	/home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/examples/ Directory Reference	11
6.2	/home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/tests/HepPDT/ Directory Reference	12
6.3	/home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/examples/HepPDT/ Directory Reference	13
6.4	/home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/include/HepPDT/ Directory Reference	14
6.5	/home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/src/HepPDT/ Directory Reference	15
6.6	/home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/include/HepPID/ Directory Reference	16
6.7	/home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/src/HepPID/ Directory Reference	17
6.8	/home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/tests/HepPID/ Directory Reference	18
6.9	/home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/examples/HepPID/ Directory Reference	19
6.10	/home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/include/ Directory Reference	20
6.11	/home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/src/ Directory Reference	21
6.12	/home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/tests/ Directory Reference	22
7	HepPDT Namespace Documentation	23

7.1	HepPDT Namespace Reference	23
7.2	HepPDT::detail Namespace Reference	33
7.3	HepPID Namespace Reference	37
8	HepPDT Class Documentation	65
8.1	HepPDT::Constituent Class Reference	65
8.2	HepPDT::DefTable Class Reference	68
8.3	HepPDT::HeavyIonUnknownID Class Reference	71
8.4	HepPDT::Measurement Class Reference	72
8.5	HepPDT::ParticleData Class Reference	74
8.6	HepPDT::ParticleDataTable Class Reference	86
8.7	HepPDT::ParticleDataTableComparison Class Reference	92
8.8	HepPDT::ParticleID Class Reference	93
8.9	HepPID::ParticleNameMap Class Reference	106
8.10	HepPDT::ProcessUnknownID Class Reference	108
8.11	HepPDT::Quarks Struct Reference	110
8.12	HepPDT::ResonanceStructure Class Reference	112
8.13	HepPDT::SimpleProcessUnknownID Class Reference	116
8.14	HepPDT::SpinState Class Reference	117
8.15	HepPDT::TableBuilder Class Reference	120
8.16	HepPDT::TempAliasData Struct Reference	124
8.17	HepPDT::TempConstituent Struct Reference	126
8.18	HepPDT::TempDecayData Struct Reference	127
8.19	HepPDT::TempParticleData Struct Reference	128
8.20	HepPDT::TestNuclearFragment Class Reference	133
9	HepPDT File Documentation	135
9.1	addEvtGenParticles.cc File Reference	135
9.2	addHerwigParticles.cc File Reference	136
9.3	addIsajetParticles.cc File Reference	137
9.4	addParticleTable.cc File Reference	138
9.5	addPDGParticles.cc File Reference	139
9.6	addPythiaParticles.cc File Reference	140
9.7	addQQParticles.cc File Reference	141
9.8	calculateWidthFromLifetime.cc File Reference	142
9.9	Constituent.cc File Reference	143
9.10	Constituent.hh File Reference	144

9.11	convertTemporaryMap.cc File Reference	145
9.12	DefTable.cc File Reference	146
9.13	DefTable.hh File Reference	147
9.14	examListHerwig.cc File Reference	148
9.15	examListIsajet.cc File Reference	150
9.16	examListPythia.cc File Reference	152
9.17	examMyPDT.cc File Reference	154
9.18	getIsajetID.cc File Reference	155
9.19	getPDGpid.cc File Reference	156
9.20	getPythiaid.cc File Reference	157
9.21	hasMethods.cc File Reference	158
9.22	HeavyIonUnknownID.cc File Reference	159
9.23	HeavyIonUnknownID.hh File Reference	160
9.24	lifetime.cc File Reference	161
9.25	list_of_examples.cc File Reference	162
9.26	list_of_tests.cc File Reference	163
9.27	listEvtGenTranslation.cc File Reference	164
9.28	listHerwigTranslation.cc File Reference	165
9.29	listIsajetTranslation.cc File Reference	166
9.30	listParticleNames.cc File Reference	167
9.31	listPDGTranslation.cc File Reference	168
9.32	listPythiaTranslation.cc File Reference	169
9.33	listQQTranslation.cc File Reference	170
9.34	Measurement.hh File Reference	171
9.35	ParticleData.hh File Reference	172
9.36	ParticleDataTable.cc File Reference	173
9.37	ParticleDataTable.hh File Reference	174
9.38	ParticleDataTableComparison.hh File Reference	175
9.39	ParticleID.cc File Reference	176
9.40	ParticleID.hh File Reference	177
9.41	ParticleIDMethods.cc File Reference	178
9.42	ParticleIDMethods.hh File Reference	181
9.43	ParticleIDTranslations.hh File Reference	184
9.44	ParticleName.cc File Reference	186
9.45	ParticleName.hh File Reference	188
9.46	ProcessUnknownID.cc File Reference	189

9.47	ProcessUnknownID.hh File Reference	190
9.48	quarks.cc File Reference	191
9.49	ResonanceStructure.cc File Reference	192
9.50	ResonanceStructure.hh File Reference	193
9.51	SimpleProcessUnknownID.hh File Reference	194
9.52	spindtoi.cc File Reference	195
9.53	spinitod.cc File Reference	196
9.54	SpinState.hh File Reference	197
9.55	stringtodouble.cc File Reference	198
9.56	stringtodouble.hh File Reference	199
9.57	TableBuilder.hh File Reference	200
9.58	TempParticleData.cc File Reference	202
9.59	TempParticleData.hh File Reference	203
9.60	testHepPDT.cc File Reference	204
9.61	TestNuclearFragment.hh File Reference	206
9.62	testParticleIDMethods.cc File Reference	207
9.63	testPID.cc File Reference	208
9.64	translateEvtGen.cc File Reference	211
9.65	translateGeanttoPDT.cc File Reference	212
9.66	translateHerwig.cc File Reference	213
9.67	translateIsajet.cc File Reference	214
9.68	translatePDG.cc File Reference	215
9.69	translatePDTtoGeant.cc File Reference	216
9.70	translatePythia.cc File Reference	217
9.71	translateQQ.cc File Reference	218
9.72	Version.cc File Reference	220
9.73	Version.cc File Reference	221
9.74	Version.hh File Reference	222
9.75	Version.hh File Reference	223
9.76	write.cc File Reference	224
10	HepPDT Example Documentation	225
10.1	examListHerwig.cc	225
10.2	examListHerwigInterface.F	227
10.3	examListIsajet.cc	228
10.4	examListIsajetInterface.F	230
10.5	examListPythia.cc	232

10.6 examListPythiaInterface.F	234
10.7 examMyPDT.cc	235
10.8 listEvtGenNames.cc.in	237
10.9 listEvtGenTranslation.cc	238
10.10listHerwigTranslation.cc	239
10.11listIsajetTranslation.cc	240
10.12listParticleNames.cc	241
10.13listPDGNames.cc.in	242
10.14listPDGTranslation.cc	243
10.15listPythiaNames.cc.in	244
10.16listPythiaTranslation.cc	245
10.17listQQTranslation.cc	246
10.18testHepPDT.cc	247
10.19testPID.cc	251
10.20testReadEvtGen.cc.in	255
10.21testReadIsajet.cc.in	256
10.22testReadParticleTable.cc.in	257
10.23testReadQQ.cc.in	259

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	23
HepPDT::detail	33
HepPID	37

Chapter 3

HepPDT Hierarchical Index

3.1 HepPDT Class Hierarchy

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

HepPDT::Constituent	65
HepPDT::DefTable	68
HepPDT::Measurement	72
HepPDT::ParticleData	74
HepPDT::ParticleDataTable	86
HepPDT::ParticleDataTableComparison	92
HepPDT::ParticleID	93
HepPID::ParticleNameMap	106
HepPDT::ProcessUnknownID	108
HepPDT::HeavyIonUnknownID	71
HepPDT::SimpleProcessUnknownID	116
HepPDT::TestNuclearFragment	133
HepPDT::Quarks	110
HepPDT::ResonanceStructure	112
HepPDT::SpinState	117
HepPDT::TableBuilder	120
HepPDT::TempAliasData	124
HepPDT::TempConstituent	126
HepPDT::TempDecayData	127
HepPDT::TempParticleData	128

Chapter 4

HepPDT Class Index

4.1 HepPDT Class List

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

HepPDT::Constituent	65
HepPDT::DefTable	68
HepPDT::HeavyIonUnknownID	71
HepPDT::Measurement	72
HepPDT::ParticleData	74
HepPDT::ParticleDataTable	86
HepPDT::ParticleDataTableComparison	92
HepPDT::ParticleID	93
HepPID::ParticleNameMap	106
HepPDT::ProcessUnknownID	108
HepPDT::Quarks (Constituent (p. 65) quarks)	110
HepPDT::ResonanceStructure	112
HepPDT::SimpleProcessUnknownID	116
HepPDT::SpinState	117
HepPDT::TableBuilder	120
HepPDT::TempAliasData (Hold Alias information from EvtGen)	124
HepPDT::TempConstituent (Temporary constituent (e.g., quark) information)	126
HepPDT::TempDecayData (Temporary holder for decay data)	127
HepPDT::TempParticleData (Temporary holder for Particle Data information)	128
HepPDT::TestNuclearFragment	133

Chapter 5

HepPDT File Index

5.1 HepPDT File List

Here is a list of all files with brief descriptions:

addEvtGenParticles.cc	135
addHerwigParticles.cc	136
addIsajetParticles.cc	137
addParticleTable.cc	138
addPDGParticles.cc	139
addPythiaParticles.cc	140
addQQParticles.cc	141
calculateWidthFromLifetime.cc	142
Constituent.cc	143
Constituent.hh	144
convertTemporaryMap.cc	145
DefTable.cc	146
DefTable.hh	147
examListHerwig.cc	148
examListIsajet.cc	150
examListPythia.cc	152
examMyPDT.cc	154
getIsajetID.cc	155
getPDGpid.cc	156
getPythiaid.cc	157
hasMethods.cc	158
HeavyIonUnknownID.cc	159
HeavyIonUnknownID.hh	160
lifetime.cc	161
list_of_examples.cc	162
list_of_tests.cc	163
listEvtGenTranslation.cc	164
listHerwigTranslation.cc	165
listIsajetTranslation.cc	166
listParticleNames.cc	167
listPDGTranslation.cc	168
listPythiaTranslation.cc	169
listQQTranslation.cc	170

Measurement.hh	171
ParticleData.hh	172
ParticleDataTable.cc	173
ParticleDataTable.hh	174
ParticleDataTableComparison.hh	175
ParticleID.cc	176
ParticleID.hh	177
ParticleIDMethods.cc	178
ParticleIDMethods.hh	181
ParticleIDTranslations.hh	184
ParticleName.cc	186
ParticleName.hh	188
ProcessUnknownID.cc	189
ProcessUnknownID.hh	190
quarks.cc	191
ResonanceStructure.cc	192
ResonanceStructure.hh	193
SimpleProcessUnknownID.hh	194
spindtoi.cc	195
spinitod.cc	196
SpinState.hh	197
stringtodouble.cc	198
stringtodouble.hh	199
TableBuilder.hh	200
TempParticleData.cc	202
TempParticleData.hh	203
testHepPDT.cc	204
TestNuclearFragment.hh	206
testParticleIDMethods.cc	207
testPID.cc	208
translateEvtGen.cc	211
translateGeanttoPDT.cc	212
translateHerwig.cc	213
translateIsajet.cc	214
translatePDG.cc	215
translatePDTtoGeant.cc	216
translatePythia.cc	217
translateQQ.cc	218
HepPDT/Version.cc	220
HepPID/Version.cc	221
HepPDT/Version.hh	222
HepPID/Version.hh	223
write.cc	224

Chapter 6

HepPDT Directory Documentation

6.1 /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/examples/ Directory Reference

Directories

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

Files

- file `list_of_examples.cc`

6.2 /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/tests/HepPDT/ Directory Reference

Files

- file `testHepPDT.cc`
- file `TestNuclearFragment.hh`
- file `testPID.cc`

6.3 /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/examples/HepPDT/ Directory Reference

Files

- file examMyPDT.cc

6.4 /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/include/HepPDT/ Directory Reference

Files

- file Constituent.hh
- file DefTable.hh
- file HeavyIonUnknownID.hh
- file Measurement.hh
- file ParticleData.hh
- file ParticleDataTable.hh
- file ParticleDataTableComparison.hh
- file ParticleID.hh
- file ProcessUnknownID.hh
- file ResonanceStructure.hh
- file SimpleProcessUnknownID.hh
- file SpinState.hh
- file stringtodouble.hh
- file TableBuilder.hh
- file TempParticleData.hh
- file HepPDT/Version.hh

6.5 /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/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 DefTable.cc
- file getIsajetID.cc
- file getPDGpid.cc
- file getPythiaid.cc
- file hasMethods.cc
- file HeavyIonUnknownID.cc
- file lifetime.cc
- file ParticleDataTable.cc
- file ParticleID.cc
- file ProcessUnknownID.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

6.6 /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/include/HepPID/ Directory Reference

Files

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

6.7 /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/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

6.8 /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/tests/Hep-PID/ 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

6.9 /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/examples/HepPID/ Directory Reference

Files

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

6.10 /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/include/ Directory Reference

Directories

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

6.11 /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/src/ Directory Reference

Directories

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

6.12 /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/tests/ Directory Reference

Directories

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

Files

- file **list_of_tests.cc**

Chapter 7

HepPDT Namespace Documentation

7.1 HepPDT Namespace Reference

Classes

- class **Constituent**
- class **DefTable**
- class **HeavyIonUnknownID**
- class **Measurement**
- class **ParticleData**
- class **ParticleDataTable**
- class **ParticleDataTableComparison**
- struct **Quarks**
constituent quarks
- class **ParticleID**
- class **ProcessUnknownID**
- class **ResonanceStructure**
- class **SimpleProcessUnknownID**
- class **SpinState**
- class **TableBuilder**
- struct **TempConstituent**
Temporary 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
- class **TestNuclearFragment**

Namespaces

- namespace **detail**

Typedefs

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

Enumerations

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

The location enum provides a convenient index into the PID.

Functions

- void **swap** (**Constituent** &first, **Constituent** &second)
- void **swap** (**Measurement** &first, **Measurement** &second)
- 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** (**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 &, **TableBuilder** &)
read PDG input and add particles to the table
- bool **addPythiaParticles** (std::istream &, **TableBuilder** &)
read Pythia input and add particles to the table

- **bool addHerwigParticles** (std::istream &, TableBuilder &)
- **bool addIsajetParticles** (std::istream &, TableBuilder &)
read Isajet particle input and add particles to the table
- **bool addIsajetDecay** (std::istream &, TableBuilder &)
read Isajet decay input and add decay information to the table
- **bool addQQParticles** (std::istream &, TableBuilder &)
read QQ input and add particles to the table
- **bool addEvtGenParticles** (std::istream &, TableBuilder &)
read EvtGen input and add particles to the table
- **bool addParticleTable** (std::istream &, TableBuilder &, bool validate=false)
validate=true => verify that the ParticleID (p. 93) is valid
- **double calculateWidthFromLifetime** (double)
Given the lifetime, calculate the width.
- **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** (const std::string &pdline)
- **void parseEvtGenConj** (std::string &cname, const std::string &pdline)
- **void parseEvtGenDefinition** (std::string &def, double &val, const std::string &pdline)
- **bool getQQLineType** (std::string <ype, int &id, std::string &name, const std::string &pdline)
- **bool parseQQDecayLine** (const std::string &pdline)
- **void parseQQParticle** (TempParticleData &tpd, const std::string &pdline)

7.1.1 Typedef Documentation

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

useful typedef

Definition at line 45 of file TempParticleData.hh.

7.1.2 Enumeration Type Documentation

7.1.2.1 enum HepPDT::location

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.

7.1.3 Function Documentation

7.1.3.1 void HepPDT::swap (Constituent & *first*, Constituent & *second*) [inline]

Definition at line 77 of file Constituent.hh.

References HepPDT::Constituent::swap().

Referenced by HepPDT::TempParticleData::swap(), HepPDT::ResonanceStructure::swap(), HepPDT::ParticleID::swap(), and HepPDT::Constituent::swap().

7.1.3.2 void HepPDT::swap (Measurement & *first*, Measurement & *second*) [inline]

Definition at line 45 of file Measurement.hh.

References HepPDT::Measurement::swap().

7.1.3.3 void HepPDT::swap (ParticleData & *first*, ParticleData & *second*) [inline]

Definition at line 184 of file ParticleData.hh.

References HepPDT::ParticleData::swap().

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

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

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

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

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

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

7.1.3.10 `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 HepPDT::TempParticleData::processPID(), and testValid().

7.1.3.11 `int HepPDT::spindtoi (double spin)`

convert an actual spin to $2J+1$

Definition at line 13 of file spindtoi.cc.

7.1.3.12 `void HepPDT::swap (ParticleID & first, ParticleID & second)` `[inline]`

Definition at line 166 of file ParticleID.hh.

References HepPDT::ParticleID::swap().

7.1.3.13 `void HepPDT::swap (ResonanceStructure & first, ResonanceStructure & second)`
`[inline]`

Definition at line 83 of file ResonanceStructure.hh.

References HepPDT::ResonanceStructure::swap().

7.1.3.14 `void HepPDT::swap (SpinState & first, SpinState & second)` `[inline]`

Definition at line 69 of file SpinState.hh.

References HepPDT::SpinState::swap().

7.1.3.15 `double HepPDT::stringtodouble (std::string & numb)`

extract a double from a string

Definition at line 15 of file stringtodouble.cc.

7.1.3.16 `bool HepPDT::addPDGParticles (std::istream &, TableBuilder &)`

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 `duplicateFragmentTest()`, `pdFfragmentTest()`, and `pdSimpleTest()`.

7.1.3.17 `bool HepPDT::addPythiaParticles (std::istream &, TableBuilder &)`

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()`.

7.1.3.18 `bool HepPDT::addHerwigParticles (std::istream &, TableBuilder &)`

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

7.1.3.19 `bool HepPDT::addIsajetParticles (std::istream &, TableBuilder &)`

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()`.

7.1.3.20 `bool HepPDT::addIsajetDecay (std::istream &, TableBuilder &)`

read Isajet decay input and add decay information to the table

7.1.3.21 bool HepPDT::addQQParticles (std::istream &, TableBuilder &)

read QQ input and add particles to the table

Examples:

testReadQQ.cc.in.

Definition at line 27 of file addQQParticles.cc.

References HepPDT::TableBuilder::addParticle(), 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().

7.1.3.22 bool HepPDT::addEvtGenParticles (std::istream &, TableBuilder &)

read EvtGen input and add particles to the table

Examples:

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

Definition at line 29 of file addEvtGenParticles.cc.

References HepPDT::TableBuilder::addAlias(), HepPDT::DefTable::addDefinition(), HepPDT::TableBuilder::addParticle(), HepPDT::TableBuilder::aliasData(), HepPDT::TableBuilder::aliasSize(), HepPDT::TableBuilder::definitions(), 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().

7.1.3.23 bool HepPDT::addParticleTable (std::istream &, TableBuilder &, bool *validate* = false)

validate=true => verify that the **ParticleID** (p.93) is valid

Examples:

testReadParticleTable.cc.in.

Definition at line 21 of file addParticleTable.cc.

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

7.1.3.24 double HepPDT::calculateWidthFromLifetime (double)

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().

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

Definition at line 106 of file TempParticleData.hh.

References HepPDT::TempParticleData::swap().

7.1.3.26 void HepPDT::version ()

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

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

References versionName().

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

7.1.3.27 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().

7.1.3.28 std::string HepPDT::versionName ()

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

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

Referenced by version(), HepPDT::ParticleDataTable::writeParticleStatus(), and writeVersion().

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

Definition at line 104 of file addEvtGenParticles.cc.

Referenced by addEvtGenParticles().

7.1.3.30 void HepPDT::parseEvtGenLine (TempParticleData & *tpd*, const std::string & *pdline*)

Definition at line 136 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().

7.1.3.31 void HepPDT::parseEvtGenAlias (TempAliasData & *tad*, const std::string & *pdline*)

Definition at line 222 of file addEvtGenParticles.cc.

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

Referenced by addEvtGenParticles().

7.1.3.32 bool HepPDT::parseEvtGenDecayLine (TempParticleData & *tpd*, const std::string & *pdline*)

Definition at line 172 of file addEvtGenParticles.cc.

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

Referenced by addEvtGenParticles().

7.1.3.33 bool HepPDT::parseEvtGenAliasDecayLine (const std::string & *pdline*)

Definition at line 201 of file addEvtGenParticles.cc.

Referenced by addEvtGenParticles().

7.1.3.34 void HepPDT::parseEvtGenConj (std::string & *cname*, const std::string & *pdline*)

Definition at line 239 of file addEvtGenParticles.cc.

Referenced by addEvtGenParticles().

7.1.3.35 void HepPDT::parseEvtGenDefinition (std::string & *def*, double & *val*, const std::string & *pdline*)

Definition at line 254 of file addEvtGenParticles.cc.

Referenced by addEvtGenParticles().

7.1.3.36 bool HepPDT::getQQLineType (std::string & *ltype*, int & *id*, std::string & *name*, const std::string & *pdline*)

Definition at line 76 of file addQQParticles.cc.

Referenced by addQQParticles().

7.1.3.37 bool HepPDT::parseQQDecayLine (const std::string & *pdline*)

Definition at line 158 of file addQQParticles.cc.

Referenced by addQQParticles().

7.1.3.38 void HepPDT::parseQQParticle (TempParticleData & *tpd*, const std::string & *pdline*)

Definition at line 117 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()`.

7.2 HepPDT::detail Namespace Reference

Functions

- **void getPDGpid** (std::vector< int > &, std::string &)
for internal use
- **void getPDGnames** (std::vector< std::string > &, std::string &)
for internal use
- **void parsePDGline** (TempParticleData &, std::string &)
for internal use
- **bool CheckPDGEntry** (TempParticleData &, const std::string &, double, double)
for internal use
- **bool getPythiaid** (int &, const std::string &)
for internal use
- **void parsePythiaLine** (TempParticleData &, int &, std::string &, const std::string &)
for internal use
- **void parsePythiaDecayLine** (TempParticleData &, const std::string &)
for internal use
- **TempDecayData getPythiaDecay** (const std::string &)
for internal use
- **bool getIsajetID** (int &, const std::string &)
for internal use
- **void parseIsajetLine** (TempParticleData &, const std::string &)
for internal use
- **void parseIsajetDecayLine** (TempParticleData &, const std::string &, TableBuilder &)
for internal use
- **bool getParticleID** (int &id, const std::string &)
for internal use
- **void parseParticleLine** (TempParticleData &, const std::string &)
for internal use

7.2.1 Function Documentation

7.2.1.1 void HepPDT::detail::getPDGpid (std::vector< int > &, std::string &)

for internal use

Definition at line 20 of file getPDGpid.cc.

Referenced by HepPDT::addPDGParticles().

7.2.1.2 void HepPDT::detail::getPDGNames (std::vector< std::string > &, std::string &)

for internal use

Definition at line 40 of file getPDGpid.cc.

Referenced by HepPDT::addPDGParticles().

7.2.1.3 void HepPDT::detail::parsePDGline (TempParticleData &, std::string &)

for internal use

Definition at line 51 of file addPDGParticles.cc.

References CheckPDGEntry().

Referenced by HepPDT::addPDGParticles().

7.2.1.4 bool HepPDT::detail::CheckPDGEntry (TempParticleData &, 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().

7.2.1.5 bool HepPDT::detail::getPythiaid (int &, const std::string &)

for internal use

Definition at line 20 of file getPythiaid.cc.

Referenced by HepPDT::addPythiaParticles().

7.2.1.6 void HepPDT::detail::parsePythiaLine (TempParticleData &, int &, std::string &, const std::string &)

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().

7.2.1.7 void HepPDT::detail::parsePythiaDecayLine (TempParticleData &, const std::string &)

for internal use

Definition at line 98 of file addPythiaParticles.cc.

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

Referenced by HepPDT::addPythiaParticles().

7.2.1.8 TempDecayData HepPDT::detail::getPythiaDecay (const std::string &)

for internal use

7.2.1.9 bool HepPDT::detail::getIsajetID (int &, const std::string &)

for internal use

Definition at line 17 of file getIsajetID.cc.

Referenced by HepPDT::addIsajetParticles().

7.2.1.10 void HepPDT::detail::parseIsajetLine (TempParticleData &, const std::string &)

for internal use

Definition at line 33 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().

7.2.1.11 void HepPDT::detail::parseIsajetDecayLine (TempParticleData &, const std::string &, TableBuilder &)

for internal use

7.2.1.12 bool HepPDT::detail::getParticleID (int & id, const std::string &)

for internal use

Definition at line 50 of file addParticleTable.cc.

Referenced by HepPDT::addParticleTable().

7.2.1.13 void HepPDT::detail::parseParticleLine (TempParticleData &, const std::string &)

for internal use

Definition at line 70 of file addParticleTable.cc.

References HepPDT::calculateWidthFromLifetime(), HepPDT::ParticleID::isQBall(), HepPDT::ParticleID::pid(), HepPDT::TempParticleData::tempCharge, HepPDT::TempParticleData::tempID, HepPDT::TempParticleData::tempMass, HepPDT::TempParticleData::tempOriginalID, HepPDT::Temp-

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

Referenced by HepPDT::addParticleTable().

7.3 HepPID Namespace Reference

Classes

- class **ParticleNameMap**

Typedefs

- typedef std::map< int, std::string > **ParteleIdMap**
- typedef std::map< std::string, int > **ParticleLookupMap**
- 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 }

The location enum provides a convenient index into the PID.

Functions

- unsigned short **digit** (location loc, const int &pid)
 return the digit at a named location in the PID
- int **A** (const int &pid)
 Ion numbers are +/- 10LZZZAAAI.
- int **Z** (const int &pid)
 Ion numbers are +/- 10LZZZAAAI.
- int **lambda** (const int &pid)
 Ion numbers are +/- 10LZZZAAAI.

- **int abspid (const int &pid)**
absolute value of particle ID
- **int fundamentalID (const int &pid)**
extract fundamental ID (1-100) 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)**
(e.g. outside the standard numbering scheme)
- **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 isDyon (const int &pid)**
is this a valid Dyon (magnetic monopole) ID?
- **bool isQBall (const int &pid)**
Ad-hoc numbering for such particles is 100xxxx0, where xxxx is the charge in tenths.
- **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)**
If this is a Q -ball, return 30 times the charge.
- **double charge (const int &pid)**
return the actual charge
- **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.37) Particle name

- **int** `particleName` (const std::string &)
lookup a known ID
- **void** `listParticleNames` (std::ostream &os)
list all known names
- **bool** `validParticleName` (const int &)
verify that this number has a valid name
- **bool** `validParticleName` (const std::string &)
verify that this string has a valid id
- **ParticleNameMap** const & `getParticleNameMap` ()
access the ParticleNameMap (p. 106) for other purposes
- **void** `version` ()
print HepPID (p. 37) version
- **void** `writeVersion` (std::ostream &os)
write HepPID (p. 37) version to os
- **std::string** `versionName` ()
return HepPID (p. 37) version
- **bool** `findQ` (const int &pid, const int &q)
- **ParticleNameMap** const & `ParticleNameInit` ()
- **void** `writeParticleNameLine` (int i, std::ostream &os)
- **std::string** `dyonName` (const int &pid)
- **std::string** `qballName` (const int &pid)
- **int** `checkForSpecialParticle` (const std::string &s)
- **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 ()

7.3.1 Typedef Documentation

7.3.1.1 typedef std::map< int, std::string > HepPID::ParticleIdMap

Definition at line 35 of file ParticleName.cc.

7.3.1.2 typedef std::map< std::string, int > HepPID::ParticleLookupMap

Definition at line 36 of file ParticleName.cc.

7.3.1.3 typedef std::map< int, int > HepPID::EvtGenPDTMap

Definition at line 33 of file translateEvtGen.cc.

7.3.1.4 typedef std::map< int, int > HepPID::PDTEvtGenMap

Definition at line 34 of file translateEvtGen.cc.

7.3.1.5 typedef std::map< int, int > HepPID::HerwigPDTMap

Definition at line 33 of file translateHerwig.cc.

7.3.1.6 typedef std::map< int, int > HepPID::PDTHerwigMap

Definition at line 34 of file translateHerwig.cc.

7.3.1.7 typedef std::map< int, int > HepPID::IsajetPDTMap

Definition at line 36 of file translateIsajet.cc.

7.3.1.8 typedef std::map< int, int > HepPID::PDTIsajetMap

Definition at line 37 of file translateIsajet.cc.

7.3.1.9 typedef std::map< int, int > HepPID::PDGtoPDTMap

Definition at line 33 of file translatePDG.cc.

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

Definition at line 34 of file translatePDG.cc.

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

Definition at line 33 of file translatePythia.cc.

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

Definition at line 34 of file translatePythia.cc.

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

Definition at line 37 of file translateQQ.cc.

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

Definition at line 38 of file translateQQ.cc.

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

Definition at line 39 of file translateQQ.cc.

7.3.1.16 typedef std::map< int, int > HepPID::InverseQQbarMap

Definition at line 40 of file translateQQ.cc.

7.3.2 Enumeration Type Documentation

7.3.2.1 enum HepPID::location

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.

7.3.3 Function Documentation

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

return the digit at a named location in the PID

Definition at line 55 of file ParticleIDMethods.cc.

References `abspid()`.

Referenced by `convIsajettoPDT()`, `convPDTtoIsajet()`, `dyonName()`, `findQ()`, `fundamentalID()`, `isBaryon()`, `isDiQuark()`, `isDyon()`, `isMeson()`, `isNucleus()`, `isPentaquark()`, `isQBall()`, `isRhadron()`, `isSUSY()`, `lambda()`, `lSpin()`, `main()`, `qballName()`, `sSpin()`, and `threeCharge()`.

7.3.3.2 int HepPID::A (const int & *pid*)

Ion numbers are +/- 10LZZZAAAI.

Definition at line 87 of file ParticleIDMethods.cc.

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

Referenced by `main()`.

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

Ion numbers are +/- 10LZZZAAAI.

Definition at line 78 of file ParticleIDMethods.cc.

References abspid(), and isNucleus().

Referenced by main().

7.3.3.4 int HepPID::lambda (const int & pid)

Ion numbers are +/- 10LZZZAAAI.

Definition at line 97 of file ParticleIDMethods.cc.

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

Referenced by main().

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

absolute value of particle ID

Definition at line 43 of file ParticleIDMethods.cc.

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

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

extract fundamental ID (1-100) if this is a "fundamental" particle

Definition at line 65 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().

7.3.3.7 bool HepPID::hasFundamentalAnti (const int & pid)

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

Definition at line 142 of file ParticleIDMethods.cc.

References fundamentalID(), and validParticleName().

Referenced by isValid().

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

(e.g. outside the standard numbering scheme)

Definition at line 49 of file ParticleIDMethods.cc.

References `abspid()`.

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

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

is this a valid ID?

Examples:

`examListPythia.cc`.

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

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

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

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

is this a valid meson ID?

Definition at line 154 of file `ParticleIDMethods.cc`.

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

Referenced by `isHadron()`, `isValid()`, `lSpin()`, `main()`, `sSpin()`, and `threeCharge()`.

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

is this a valid baryon ID?

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

References `abspid()`, `digit()`, `extraBits()`, `fundamentalID()`, `isPentaquark()`, `isRhadron()`, `nj`, `nq1`, `nq2`, and `nq3`.

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

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

is this a valid diquark ID?

Definition at line 193 of file `ParticleIDMethods.cc`.

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

Referenced by `isValid()`, `main()`, and `threeCharge()`.

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

is this a valid hadron ID?

Definition at line 212 of file ParticleIDMethods.cc.

References extraBits(), isBaryon(), isMeson(), isPentaquark(), and isRhadron().

Referenced by main().

7.3.3.14 bool HepPID::isLepton (const int & pid)

is this a valid lepton ID?

Definition at line 222 of file ParticleIDMethods.cc.

References extraBits(), and fundamentalID().

Referenced by main().

7.3.3.15 bool HepPID::isNucleus (const int & pid)

is this a valid ion ID?

Definition at line 236 of file ParticleIDMethods.cc.

References abspid(), digit(), n10, and n9.

Referenced by A(), isValid(), lambda(), main(), and Z().

7.3.3.16 bool HepPID::isPentaquark (const int & pid)

is this a valid pentaquark ID?

Definition at line 250 of file ParticleIDMethods.cc.

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

Referenced by findQ(), isBaryon(), isHadron(), isValid(), and main().

7.3.3.17 bool HepPID::isSUSY (const int & pid)

is this a valid SUSY ID?

Definition at line 270 of file ParticleIDMethods.cc.

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

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

7.3.3.18 bool HepPID::isRhadron (const int & pid)

is this a valid R-hadron ID?

Definition at line 282 of file ParticleIDMethods.cc.

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

Referenced by findQ(), isBaryon(), isHadron(), isMeson(), isValid(), main(), and threeCharge().

7.3.3.19 bool HepPID::isDyon (const int & pid)

is this a valid Dyon (magnetic monopole) ID?

Magnetic monopoles and Dyons are assumed to have one unit of Dirac monopole charge and a variable integer number xyz units of electric charge.

Codes 411xyz0 are then used when the magnetic and electrical charge sign agree and 412xyz0 when they disagree, with the overall sign of the particle set by the magnetic charge.

For now no spin information is provided.

Definition at line 300 of file ParticleIDMethods.cc.

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

Referenced by findQ(), isValid(), main(), particleName(), threeCharge(), and validParticleName().

7.3.3.20 bool HepPID::isQBall (const int & pid)

Ad-hoc numbering for such particles is 100xxxx0, where xxxx is the charge in tenths.

Definition at line 325 of file ParticleIDMethods.cc.

References abspid(), digit(), extraBits(), n, nj, and nr.

Referenced by charge(), isValid(), main(), particleName(), threeCharge(), and validParticleName().

7.3.3.21 bool HepPID::hasUp (const int & pid)

does this particle contain an up quark?

Definition at line 338 of file ParticleIDMethods.cc.

References extraBits(), findQ(), and fundamentalID().

Referenced by HepPDT::ParticleID::hasUp(), and main().

7.3.3.22 bool HepPID::hasDown (const int & pid)

does this particle contain a down quark?

Definition at line 345 of file ParticleIDMethods.cc.

References extraBits(), findQ(), and fundamentalID().

Referenced by HepPDT::ParticleID::hasDown(), and main().

7.3.3.23 bool HepPID::hasStrange (const int & pid)

does this particle contain a strange quark?

Definition at line 352 of file ParticleIDMethods.cc.

References extraBits(), findQ(), and fundamentalID().

Referenced by HepPDT::ParticleID::hasStrange(), and main().

7.3.3.24 bool HepPID::hasCharm (const int & pid)

does this particle contain a charm quark?

Definition at line 359 of file ParticleIDMethods.cc.

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

Referenced by `HepPDT::ParticleID::hasCharm()`, and `main()`.

7.3.3.25 bool HepPID::hasBottom (const int & pid)

does this particle contain a bottom quark?

Definition at line 366 of file ParticleIDMethods.cc.

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

Referenced by `HepPDT::ParticleID::hasBottom()`, and `main()`.

7.3.3.26 bool HepPID::hasTop (const int & pid)

does this particle contain a top quark?

Definition at line 373 of file ParticleIDMethods.cc.

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

Referenced by `HepPDT::ParticleID::hasTop()`, and `main()`.

7.3.3.27 int HepPID::jSpin (const int & pid)

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

Definition at line 383 of file ParticleIDMethods.cc.

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

Referenced by `main()`.

7.3.3.28 int HepPID::sSpin (const int & pid)

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

Definition at line 399 of file ParticleIDMethods.cc.

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

Referenced by `main()`.

7.3.3.29 int HepPID::lSpin (const int & pid)

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

Definition at line 424 of file ParticleIDMethods.cc.

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

Referenced by `main()`.

7.3.3.30 int HepPID::threeCharge (const int & *pid*)

If this is a Q-ball, return 30 times the charge.

Definition at line 473 of file ParticleIDMethods.cc.

References `abspid()`, `charge()`, `digit()`, `extraBits()`, `fundamentalID()`, `isBaryon()`, `isDiQuark()`, `isDyon()`, `isMeson()`, `isQBall()`, `isRhadron()`, `nj`, `nl`, `nq1`, `nq2`, `nq3`, and `nr`.

Referenced by `charge()`, and `main()`.

7.3.3.31 double HepPID::charge (const int & *pid*)

return the actual charge

Examples:

examMyPDT.cc.

Definition at line 549 of file ParticleIDMethods.cc.

References `isQBall()`, and `threeCharge()`.

Referenced by `main()`, and `threeCharge()`.

7.3.3.32 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()`.

7.3.3.33 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()`.

7.3.3.34 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()`.

7.3.3.35 `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::addIsajetParticles()`, and `main()`.

7.3.3.36 `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()`.

7.3.3.37 `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()`.

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

translate Pythia to PDG standard

Examples:

`examListPythia.cc`.

Definition at line 664 of file translatePythia.cc.

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

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

7.3.3.39 int HepPID::translatePDTtoPythia (const int *pid*)

translate PDG standard to Pythia

Definition at line 677 of file translatePythia.cc.

References getPDTPythiaMap(), and isValid().

Referenced by writePythiaTranslationLine().

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

output the translation list

Examples:

listPythiaTranslation.cc.

Definition at line 714 of file translatePythia.cc.

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

Referenced by main().

7.3.3.41 int HepPID::translateEvtGenToPDT (const int *evtGenID*)

translate EvtGen to PDG standard

Definition at line 606 of file translateEvtGen.cc.

References getEvtGenPDTMap(), and isValid().

Referenced by HepPDT::addEvtGenParticles().

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

translate PDG standard to EvtGen

Definition at line 619 of file translateEvtGen.cc.

References getPDTEvtGenMap(), and isValid().

Referenced by writeEvtGenTranslationLine().

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

output the translation list

Examples:

listEvtGenTranslation.cc.

Definition at line 656 of file translateEvtGen.cc.

References writeEvtGenTranslationLine(), and writeVersion().

Referenced by main().

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

translate PDG table to PDG standard

Definition at line 386 of file translatePDG.cc.

References getPDGtoPDTMap(), and isValid().

7.3.3.45 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().

7.3.3.46 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().

7.3.3.47 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().

7.3.3.48 int HepPID::translatePDTtoQQ (const int *pid*)

translate PDG standard to QQ

Definition at line 601 of file translateQQ.cc.

References getPDTQQMap().

Referenced by writeQQTranslation().

7.3.3.49 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().

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

QQ helper function.

Definition at line 579 of file translateQQ.cc.

References getInverseQQbarMap().

Referenced by writeQQTranslation().

7.3.3.51 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().

7.3.3.52 int HepPID::translateGeanttoPDT (const int *geantID*)

translate Geant3 to PDG standard

Definition at line 20 of file translateGeanttoPDT.cc.

References IDMAX.

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

translate PDG standard to Geant3

Definition at line 22 of file translatePDTtoGeant.cc.

References abspid(), and IDMAX.

7.3.3.54 std::string HepPID::particleName (const int &)

get a known **HepPID** (p. 37) Particle name

Examples:

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

Definition at line 1799 of file ParticleName.cc.

References `dyonName()`, `HepPID::ParticleNameMap::end()`, `HepPID::ParticleNameMap::find()`, `getParticleNameMap()`, `isDyon()`, `isQBall()`, and `qballName()`.

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

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

lookup a known ID

Definition at line 1813 of file `ParticleName.cc`.

References `checkForSpecialParticle()`, `HepPID::ParticleNameMap::endLookupMap()`, `HepPID::ParticleNameMap::findString()`, and `getParticleNameMap()`.

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

list all known names

Examples:

`listParticleNames.cc`.

Definition at line 1825 of file `ParticleName.cc`.

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

Referenced by `main()`.

7.3.3.57 `bool HepPID::validParticleName (const int &)`

verify that this number has a valid name

Definition at line 1776 of file `ParticleName.cc`.

References `HepPID::ParticleNameMap::end()`, `HepPID::ParticleNameMap::find()`, `getParticleNameMap()`, `isDyon()`, and `isQBall()`.

Referenced by `hasFundamentalAnti()`, and `writeParticleNameLine()`.

7.3.3.58 `bool HepPID::validParticleName (const std::string &)`

verify that this string has a valid id

Definition at line 1790 of file `ParticleName.cc`.

References `HepPID::ParticleNameMap::endLookupMap()`, `HepPID::ParticleNameMap::findString()`, and `getParticleNameMap()`.

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

access the **ParticleNameMap** (p. 106) for other purposes

Definition at line 1770 of file `ParticleName.cc`.

References `ParticleNameInit()`.

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

7.3.3.60 `void HepPID::version ()`

print **HepPID** (p. 37) version

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

References `versionName()`.

7.3.3.61 `void HepPID::writeVersion (std::ostream & os)`

write **HepPID** (p. 37) 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()`.

7.3.3.62 `std::string HepPID::versionName ()`

return **HepPID** (p. 37) version

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

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

7.3.3.63 `bool HepPID::@0::findQ (const int & pid, const int & q) [static]`

Definition at line 17 of file `ParticleIDMethods.cc`.

References `digit()`, `isDyon()`, `isPentaquark()`, `isRhadron()`, `nl`, `nq1`, `nq2`, `nq3`, and `nr`.

Referenced by `hasBottom()`, `hasCharm()`, `hasDown()`, `hasStrange()`, `hasTop()`, and `hasUp()`.

7.3.3.64 `ParticleNameMap const& HepPID::@1::ParticleNameInit () [static]`

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

Referenced by `getParticleNameMap()`.

7.3.3.65 `void HepPID::@1::writeParticleNameLine (int i, std::ostream & os) [static]`

Definition at line 1689 of file `ParticleName.cc`.

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

Referenced by `listParticleNames()`.

7.3.3.66 `std::string HepPID::@1::dyonName (const int & pid)` [static]

Definition at line 1706 of file ParticleName.cc.

References digit(), nl, nq1, nq2, and nq3.

Referenced by particleName().

7.3.3.67 `std::string HepPID::@1::qballName (const int & pid)` [static]

Definition at line 1726 of file ParticleName.cc.

References abspid(), digit(), and nq3.

Referenced by particleName().

7.3.3.68 `int HepPID::@1::checkForSpecialParticle (const std::string & s)` [static]

Definition at line 1738 of file ParticleName.cc.

Referenced by particleName().

7.3.3.69 `EvtGenPDTMap const& HepPID::@2::getEvtGenPDTMap ()` [static]

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

7.3.3.70 `PDTEvtGenMap const& HepPID::@2::getPDTEvtGenMap ()` [static]

Referenced by translatePDTtoEvtGen().

7.3.3.71 `EvtGenPDTMap const& HepPID::@2::EvtGenPDTMapInit ()` [static]

Definition at line 41 of file translateEvtGen.cc.

Referenced by getEvtGenPDTMap().

7.3.3.72 `PDTEvtGenMap const& HepPID::@2::PDTEvtGenMapInit ()` [static]

Definition at line 575 of file translateEvtGen.cc.

References getEvtGenPDTMap().

Referenced by getPDTEvtGenMap().

7.3.3.73 `EvtGenPDTMap const& HepPID::@2::getEvtGenPDTMap ()` [static]

Definition at line 589 of file translateEvtGen.cc.

References EvtGenPDTMapInit().

7.3.3.74 PDTEvtGenMap const& HepPID::@2::getPDTEvtGenMap () [static]

Definition at line 598 of file translateEvtGen.cc.

References PDTEvtGenMapInit().

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

Definition at line 632 of file translateEvtGen.cc.

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

Referenced by writeEvtGenTranslation().

7.3.3.76 HerwigPDTMap const& HepPID::@3::getHerwigPDTMap () [static]

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

7.3.3.77 PDTHerwigMap const& HepPID::@3::getPDTHerwigMap () [static]

Referenced by translatePDTtoHerwig().

7.3.3.78 HerwigPDTMap const& HepPID::@3::HerwigPDTMapInit () [static]

Definition at line 41 of file translateHerwig.cc.

Referenced by getHerwigPDTMap().

7.3.3.79 PDTHerwigMap const& HepPID::@3::PDTHerwigMapInit () [static]

Definition at line 486 of file translateHerwig.cc.

References getHerwigPDTMap().

Referenced by getPDTHerwigMap().

7.3.3.80 HerwigPDTMap const& HepPID::@3::getHerwigPDTMap () [static]

Definition at line 500 of file translateHerwig.cc.

References HerwigPDTMapInit().

7.3.3.81 PDTHerwigMap const& HepPID::@3::getPDTHerwigMap () [static]

Definition at line 509 of file translateHerwig.cc.

References PDTHerwigMapInit().

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

Definition at line 543 of file translateHerwig.cc.

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

Referenced by writeHerwigTranslation().

7.3.3.83 IsajetPDTMap const& HepPID::@4::getIsajetPDTMap () [static]

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

7.3.3.84 PDTIsajetMap const& HepPID::@4::getPDTIsajetMap () [static]

Referenced by translatePDTtoIsajet().

7.3.3.85 IsajetPDTMap const& HepPID::@4::IsajetPDTMapInit () [static]

Definition at line 44 of file translateIsajet.cc.

Referenced by getIsajetPDTMap().

7.3.3.86 PDTIsajetMap const& HepPID::@4::PDTIsajetMapInit () [static]

Definition at line 677 of file translateIsajet.cc.

References getIsajetPDTMap().

Referenced by getPDTIsajetMap().

7.3.3.87 int HepPID::@4::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().

7.3.3.88 int HepPID::@4::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().

7.3.3.89 IsajetPDTMap const& HepPID::@4::getIsajetPDTMap () [static]

Definition at line 891 of file translateIsajet.cc.

References IsajetPDTMapInit().

7.3.3.90 PDTIsajetMap const& HepPID::@4::getPDTIsajetMap () [static]

Definition at line 900 of file translateIsajet.cc.

References PDTIsajetMapInit().

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

Definition at line 930 of file translateIsajet.cc.

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

Referenced by writeIsajetTranslation().

7.3.3.92 PDGtoPDTMap const& HepPID::@5::getPDGtoPDTMap () [static]

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

7.3.3.93 PDTtoPDGMap const& HepPID::@5::getPDTtoPDGMap () [static]

Referenced by translatePDTtoPDGtable().

7.3.3.94 PDGtoPDTMap const& HepPID::@5::PDGtoPDTMapInit () [static]

Definition at line 41 of file translatePDG.cc.

Referenced by getPDGtoPDTMap().

7.3.3.95 PDTtoPDGMap const& HepPID::@5::PDTtoPDGMapInit () [static]

Definition at line 355 of file translatePDG.cc.

References getPDGtoPDTMap().

Referenced by getPDTtoPDGMap().

7.3.3.96 PDGtoPDTMap const& HepPID::@5::getPDGtoPDTMap () [static]

Definition at line 369 of file translatePDG.cc.

References PDGtoPDTMapInit().

7.3.3.97 PDTtoPDGMap const& HepPID::@5::getPDTtoPDGMap () [static]

Definition at line 378 of file translatePDG.cc.

References PDTtoPDGMapInit().

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

Definition at line 412 of file translatePDG.cc.

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

Referenced by writePDGTranslation().

7.3.3.99 PythiaPDTMap const& HepPID::@6::getPythiaPDTMap () [static]

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

7.3.3.100 PDTPythiaMap const& HepPID::@6::getPDTPythiaMap () [static]

Referenced by translatePDTtoPythia().

7.3.3.101 PythiaPDTMap const& HepPID::@6::PythiaPDTMapInit () [static]

Definition at line 41 of file translatePythia.cc.

Referenced by getPythiaPDTMap().

7.3.3.102 PDTPythiaMap const& HepPID::@6::PDTPythiaMapInit () [static]

Definition at line 633 of file translatePythia.cc.

References getPythiaPDTMap().

Referenced by getPDTPythiaMap().

7.3.3.103 PythiaPDTMap const& HepPID::@6::getPythiaPDTMap () [static]

Definition at line 647 of file translatePythia.cc.

References PythiaPDTMapInit().

7.3.3.104 PDTPythiaMap const& HepPID::@6::getPDTPythiaMap () [static]

Definition at line 656 of file translatePythia.cc.

References PDTPythiaMapInit().

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

Definition at line 690 of file translatePythia.cc.

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

Referenced by writePythiaTranslation().

7.3.3.106 QQPDTMap const& HepPID::@7::getQQPDTMap () [static]

Referenced by PDTQQMapInit(), and translateQQtoPDT().

7.3.3.107 PDTQQMap const& HepPID::@7::getPDTQQMap () [static]

Referenced by translatePDTtoQQ().

7.3.3.108 QQbarMap const& HepPID::@7::getQQbarMap () [static]

Referenced by InverseQQbarMapInit(), and translateQQbar().

7.3.3.109 InverseQQbarMap const& HepPID::@7::getInverseQQbarMap () [static]

Referenced by translateInverseQQbar().

7.3.3.110 QQPDTMap const& HepPID::@7::QQPDTMapInit () [static]

Definition at line 49 of file translateQQ.cc.

Referenced by getQQPDTMap().

7.3.3.111 QQbarMap const& HepPID::@7::QQbarMapInit () [static]

Definition at line 455 of file translateQQ.cc.

Referenced by getQQbarMap().

7.3.3.112 PDTQQMap const& HepPID::@7::PDTQQMapInit () [static]

Definition at line 509 of file translateQQ.cc.

References getQQPDTMap().

Referenced by getPDTQQMap().

7.3.3.113 InverseQQbarMap const& HepPID::@7::InverseQQbarMapInit () [static]

Definition at line 520 of file translateQQ.cc.

References getQQbarMap().

Referenced by getInverseQQbarMap().

7.3.3.114 QQPDTMap const& HepPID::@7::getQQPDTMap () [static]

Definition at line 534 of file translateQQ.cc.

References QQPDTMapInit().

7.3.3.115 PDTQQMap const& HepPID::@7::getPDTQQMap () [static]

Definition at line 543 of file translateQQ.cc.

References PDTQQMapInit().

7.3.3.116 QQbarMap const& HepPID::@7::getQQbarMap () [static]

Definition at line 551 of file translateQQ.cc.

References QQbarMapInit().

7.3.3.117 InverseQQbarMap const& HepPID::@7::getInverseQQbarMap () [static]

Definition at line 560 of file translateQQ.cc.

References InverseQQbarMapInit().

Chapter 8

HepPDT Class Documentation

8.1 HepPDT::Constituent Class Reference

```
#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. 93) 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?

8.1.1 Detailed Description

Author:

Lynn Garren

Definition at line 26 of file Constituent.hh.

8.1.2 Constructor & Destructor Documentation

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

Definition at line 31 of file Constituent.hh.

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

Definition at line 36 of file Constituent.hh.

8.1.3 Member Function Documentation

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

is this a bottom quark?

Definition at line 36 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

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

is this a charm quark?

Definition at line 30 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

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

is this a down quark?

Definition at line 18 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

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

is this a strange quark?

Definition at line 24 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

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

is this a top quark?

Definition at line 42 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

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

is this an up quark?

Definition at line 12 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

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

how many of this constituent are there?

Definition at line 50 of file Constituent.hh.

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

Definition at line 38 of file Constituent.hh.

References swap().

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

ParticleID (p. 93) of this constituent.

Definition at line 52 of file Constituent.hh.

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

Definition at line 43 of file Constituent.hh.

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

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

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

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

8.2 HepPDT::DefTable Class Reference

```
#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

8.2.1 Detailed Description

Author:

Lynn Garren

Definition at line 23 of file DefTable.hh.

8.2.2 Member Typedef Documentation

8.2.2.1 typedef TempDefMap::const_iterator HepPDT::DefTable::const_iterator

Definition at line 27 of file DefTable.hh.

8.2.2.2 typedef TempDefMap::iterator HepPDT::DefTable::iterator

Definition at line 28 of file DefTable.hh.

8.2.3 Constructor & Destructor Documentation

8.2.3.1 HepPDT::DefTable::DefTable ()

Definition at line 15 of file DefTable.cc.

8.2.3.2 HepPDT::DefTable::~~DefTable () [inline]

Definition at line 31 of file DefTable.hh.

8.2.4 Member Function Documentation

8.2.4.1 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().

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

begin iterating over the definition map

Definition at line 55 of file DefTable.hh.

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

begin iterating over the definition map

Definition at line 53 of file DefTable.hh.

Referenced by writeDefinitions().

8.2.4.4 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().

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

end iterating over the definition map

Definition at line 60 of file DefTable.hh.

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

end iterating over the definition map

Definition at line 58 of file DefTable.hh.

Referenced by writeDefinitions().

8.2.4.7 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().

8.2.4.8 int HepPDT::DefTable::size () const [inline]

get the size of the definition map

Definition at line 46 of file DefTable.hh.

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

use for diagnostics

Definition at line 41 of file DefTable.cc.

References begin(), and end().

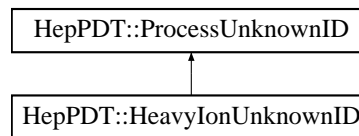
The documentation for this class was generated from the following files:

- DefTable.hh
- DefTable.cc

8.3 HepPDT::HeavyIonUnknownID Class Reference

```
#include <HeavyIonUnknownID.hh>
```

Inheritance diagram for HepPDT::HeavyIonUnknownID::



Public Member Functions

- **HeavyIonUnknownID ()**
- **virtual ParticleData * processUnknownID (ParticleID, const ParticleDataTable &pdt)**

8.3.1 Detailed Description

Author:

Lynn Garren

Examples:

testHepPDT.cc.

Definition at line 25 of file HeavyIonUnknownID.hh.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 HepPDT::HeavyIonUnknownID::HeavyIonUnknownID () [inline]

Definition at line 27 of file HeavyIonUnknownID.hh.

8.3.3 Member Function Documentation

8.3.3.1 ParticleData * HepPDT::HeavyIonUnknownID::processUnknownID (ParticleID, const ParticleDataTable &pdt) [virtual]

Implements **HepPDT::ProcessUnknownID p.** (classHepPDT₁₁*ProcessUnknownID*??)

Definition at line 13 of file HeavyIonUnknownID.cc.

References HepPDT::ParticleID::isNucleus(), HepPDT::ParticleData::mass(), and HepPDT::ParticleDataTable::particle().

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

- **HeavyIonUnknownID.hh**
- **HeavyIonUnknownID.cc**

8.4 HepPDT::Measurement Class Reference

```
#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**

8.4.1 Detailed Description

Author:

Mark Fischler

Examples:

examMyPDT.cc.

Definition at line 20 of file Measurement.hh.

8.4.2 Constructor & Destructor Documentation

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

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

8.4.2.3 HepPDT::Measurement::Measurement (const Measurement & m) [inline]

8.4.3 Member Function Documentation

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

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

compare the value, ignore the error

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

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

compare the value, ignore the error

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

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

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

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

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

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

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

- **Measurement.hh**

8.5 HepPDT::ParticleData Class Reference

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

Public Member Functions

- **ParticleData (const TempParticleData &tpd)**
create ParticleData (p. 74) from the temporary information
- **~ParticleData ()**
- **void swap (ParticleData &rhs)**
- **ParticleData (const ParticleData &orig)**
- **ParticleData & operator= (const ParticleData &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. 93)
- **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. 93) for a constituent particle.
- **ResonanceStructure const resonance () const**
resonance (width) information
- **void write (std::ostream &os) const**
output information about this particle
- **void writeParticleInfo (std::ostream &os) const**
This method is redundant with write() (p. 85), but retained for backwards compatibility.
- **void writeParticleTranslation (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 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 isDyon () const**
is this a valid Dyon (magnetic monopole) ID?
- **bool isQBall () const**
Ad-hoc numbering for such particles is 100xxxx0, where xxxx is the charge in tenths.
- **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 isStable () const**
is this particle allowed to decay?
- **bool operator< (const ParticleData &other) const**
compare masses
- **bool operator== (const ParticleData &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

8.5.1 Detailed Description

Author:

Lynn Garren

Examples:

examMyPDT.cc, testHepPDT.cc, and testReadParticleTable.cc.in.

Definition at line 32 of file ParticleData.hh.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 HepPDT::ParticleData::ParticleData (const TempParticleData & tpd)

create **ParticleData** (p. 74) from the temporary information

8.5.2.2 HepPDT::ParticleData::~~ParticleData ()

8.5.2.3 HepPDT::ParticleData::ParticleData (const ParticleData & orig)

8.5.3 Member Function Documentation

8.5.3.1 void HepPDT::ParticleData::addConstituent (Constituent c) [inline]

add a constituent particle

Definition at line 150 of file ParticleData.hh.

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

return charge

Definition at line 63 of file ParticleData.hh.

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

color information

Definition at line 65 of file ParticleData.hh.

8.5.3.4 Constituent HepPDT::ParticleData::constituent (unsigned int *i*) const

constituent information

8.5.3.5 ParticleID HepPDT::ParticleData::constituentParticle (unsigned int *i*) const

ParticleID (p. 93) for a constituent particle.

8.5.3.6 bool HepPDT::ParticleData::hasBottom () const

does this particle contain a bottom quark?

Examples:

testHepPDT.cc.

Definition at line 56 of file hasMethods.cc.

Referenced by testPDMethods().

8.5.3.7 bool HepPDT::ParticleData::hasCharm () const

does this particle contain a charm quark?

Examples:

testHepPDT.cc.

Definition at line 46 of file hasMethods.cc.

Referenced by testPDMethods().

8.5.3.8 bool HepPDT::ParticleData::hasDown () const

does this particle contain a down quark?

Examples:

testHepPDT.cc.

Definition at line 26 of file hasMethods.cc.

Referenced by testPDMethods().

8.5.3.9 bool HepPDT::ParticleData::hasStrange () const

does this particle contain a strange quark?

Examples:

testHepPDT.cc.

Definition at line 36 of file hasMethods.cc.

Referenced by testPDMethods().

8.5.3.10 bool HepPDT::ParticleData::hasTop () const

does this particle contain a top quark?

Examples:

testHepPDT.cc.

Definition at line 66 of file hasMethods.cc.

Referenced by testPDMethods().

8.5.3.11 bool HepPDT::ParticleData::hasUp () const

does this particle contain an up quark?

Examples:

testHepPDT.cc.

Definition at line 16 of file hasMethods.cc.

Referenced by testPDMethods().

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

return the **ParticleID** (p. 93)

Definition at line 57 of file ParticleData.hh.

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

is this a valid baryon?

Definition at line 100 of file ParticleData.hh.

References HepPDT::ParticleID::isBaryon().

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

is this a valid diquark?

Definition at line 102 of file ParticleData.hh.

References HepPDT::ParticleID::isDiQuark().

8.5.3.15 bool HepPDT::ParticleData::isDyon () const [inline]

is this a valid Dyon (magnetic monopole) ID?

Definition at line 116 of file ParticleData.hh.

References HepPDT::ParticleID::isDyon().

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

is this a valid hadron?

Definition at line 104 of file ParticleData.hh.

References HepPDT::ParticleID::isHadron().

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

is this a valid lepton?

Definition at line 106 of file ParticleData.hh.

References HepPDT::ParticleID::isLepton().

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

is this a valid meson?

Definition at line 98 of file ParticleData.hh.

References HepPDT::ParticleID::isMeson().

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

is this a valid ion?

Definition at line 108 of file ParticleData.hh.

References HepPDT::ParticleID::isNucleus().

8.5.3.20 bool HepPDT::ParticleData::isPentaquark () const [inline]

is this a valid pentaquark ID?

Definition at line 110 of file ParticleData.hh.

References HepPDT::ParticleID::isPentaquark().

8.5.3.21 bool HepPDT::ParticleData::isQBall () const [inline]

Ad-hoc numbering for such particles is 100xxxx0, where xxxx is the charge in tenths.

Definition at line 119 of file ParticleData.hh.

References HepPDT::ParticleID::isQBall().

8.5.3.22 bool HepPDT::ParticleData::isRhadron () const [inline]

is this a valid R-hadron ID?

Definition at line 114 of file ParticleData.hh.

References HepPDT::ParticleID::isRhadron().

8.5.3.23 bool HepPDT::ParticleData::isStable () const

is this particle allowed to decay?

8.5.3.24 bool HepPDT::ParticleData::isSUSY () const [inline]

is this a valid SUSY ID?

Definition at line 112 of file ParticleData.hh.

References HepPDT::ParticleID::isSUSY().

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

calculate the lifetime

Definition at line 77 of file ParticleData.hh.

References HepPDT::ResonanceStructure::lifetime().

Referenced by write().

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

lower cutoff of allowed width values

Examples:

testHepPDT.cc.

Definition at line 73 of file ParticleData.hh.

References HepPDT::ResonanceStructure::lowerCutoff().

Referenced by pdtSimpleTest(), and write().

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

mass

Definition at line 69 of file ParticleData.hh.

References HepPDT::ResonanceStructure::mass().

Referenced by HepPDT::TestNuclearFragment::processUnknownID(), HepPDT::HeavyIonUnknownID::processUnknownID(), setMass(), and write().

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

return particle name as defined by user input

Examples:

examMyPDT.cc, and testHepPDT.cc.

Definition at line 51 of file ParticleData.hh.

Referenced by main(), and testPDMethods().

8.5.3.29 `int HepPDT::ParticleData::numConstituents () const` `[inline]`

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

Definition at line 79 of file ParticleData.hh.

8.5.3.30 `bool HepPDT::ParticleData::operator< (const ParticleData & other) const`

compare masses

8.5.3.31 `ParticleData& HepPDT::ParticleData::operator= (const ParticleData & rhs)`**8.5.3.32** `bool HepPDT::ParticleData::operator== (const ParticleData & other) const`

use PID and ignore everything else

8.5.3.33 `int HepPDT::ParticleData::originalID () const` `[inline]`

return untranslated integer ID

Definition at line 61 of file ParticleData.hh.

8.5.3.34 `const std::string HepPDT::ParticleData::PDTname () const` `[inline]`

return PDG particle name

Definition at line 53 of file ParticleData.hh.

References HepPDT::ParticleID::PDTname().

Referenced by writeParticleTranslation().

8.5.3.35 `int HepPDT::ParticleData::pid () const` `[inline]`

return the integer ID

Examples:

testHepPDT.cc.

Definition at line 59 of file ParticleData.hh.

References HepPDT::ParticleID::pid().

Referenced by testPDMethods(), and writeParticleTranslation().

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

resonance (width) information

Definition at line 85 of file ParticleData.hh.

8.5.3.37 void HepPDT::ParticleData::setCharge (double *chg*) [inline]

change the charge

Definition at line 144 of file ParticleData.hh.

8.5.3.38 void HepPDT::ParticleData::setColor (double *col*) [inline]

change color information

Definition at line 146 of file ParticleData.hh.

8.5.3.39 void HepPDT::ParticleData::setLowerCutoff (double *cut*) [inline]

change the total width lower cutoff

Definition at line 158 of file ParticleData.hh.

References HepPDT::ResonanceStructure::setLowerCutoff().

8.5.3.40 void HepPDT::ParticleData::setMass (Measurement const & *mass*) [inline]

change the mass

Definition at line 152 of file ParticleData.hh.

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

8.5.3.41 void HepPDT::ParticleData::setSpin (const SpinState & *spin*) [inline]

change spin information

Definition at line 148 of file ParticleData.hh.

References spin().

8.5.3.42 void HepPDT::ParticleData::setTotalWidth (Measurement const & *width*) [inline]

change the total width

Definition at line 154 of file ParticleData.hh.

References HepPDT::ResonanceStructure::setTotalWidth().

8.5.3.43 void HepPDT::ParticleData::setTotalWidthFromLifetime (Measurement const & *lt*)
[inline]

change the total width using a lifetime

Definition at line 156 of file ParticleData.hh.

References HepPDT::ResonanceStructure::setTotalWidthFromLifetime().

8.5.3.44 void HepPDT::ParticleData::setUpperCutoff (double *cut*) [inline]

change the total width upper cutoff

Definition at line 160 of file ParticleData.hh.

References HepPDT::ResonanceStructure::setUpperCutoff().

8.5.3.45 const std::string& HepPDT::ParticleData::source () const [inline]

the name of the input source

Definition at line 55 of file ParticleData.hh.

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

spin information

Definition at line 67 of file ParticleData.hh.

Referenced by setSpin().

8.5.3.47 void HepPDT::ParticleData::swap (ParticleData & *rhs*)

Referenced by HepPDT::swap().

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

return the total width

Examples:

testHepPDT.cc.

Definition at line 71 of file ParticleData.hh.

References HepPDT::ResonanceStructure::totalWidth().

Referenced by pdtSimpleTest().

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

upper cutoff of allowed width values

Examples:

testHepPDT.cc.

Definition at line 75 of file ParticleData.hh.

References HepPDT::ResonanceStructure::upperCutoff().

Referenced by pdtSimpleTest(), and write().

8.5.3.50 void HepPDT::ParticleData::write (std::ostream & os) const

output information about this particle

Examples:

testHepPDT.cc, and testReadParticleTable.cc.in.

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().

Referenced by duplicateFragmentTest(), pdtFragmentTest(), pdtSimpleTest(), and writeParticleInfo().

8.5.3.51 void HepPDT::ParticleData::writeParticleInfo (std::ostream & os) const

This method is redundant with **write()** (p. 85), but retained for backwards compatibility.

Definition at line 68 of file write.cc.

References write().

8.5.3.52 void HepPDT::ParticleData::writeParticleTranslation (std::ostream & os) const

output the translation information for this particle

Definition at line 73 of file write.cc.

References PDName(), and pid().

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

- **ParticleData.hh**
- **hasMethods.cc**
- **write.cc**

8.6 HepPDT::ParticleDataTable Class Reference

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

Public Types

- typedef ParticleData CPD
- 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="", ProcessUnknownID *==new SimpleProcessUnknownID)
- Require a method to deal with unknown PID's.*
- ~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. 93)*
- ParticleData * particle (ParticleID)
- access particle information via ParticleID (p. 93)*
- 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. 93)
- **ParticleData const * operator[] (ParticleID) const**
access particle information via ParticleID (p. 93)
- **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 &outstr)**
output all information in the PDT
- **void writeParticleInfo (std::ostream &outstr)**
output all information about a particle EXCEPT its decays
- **void writeParticleTranslation (std::ostream &outstr)**
output a list of original IDs and their translations
- **void writeParticleStatus (std::ostream &)**
primarily useful for testing
- **void convertTemporaryMap (TempMap &tempPDT, std::ostream &err)**
used by the TableBuilder (p. 120) destructor to fill the PDT

8.6.1 Detailed Description

Author:

Lynn Garren, Walter Brown

Examples:

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

Definition at line 44 of file ParticleDataTable.hh.

8.6.2 Member Typedef Documentation

8.6.2.1 typedef PDTMap::const_iterator HepPDT::ParticleDataTable::const_iterator

Definition at line 53 of file ParticleDataTable.hh.

8.6.2.2 `typedef PDTNameMap::const_iterator HepPDT::ParticleDataTable::const_iteratorByName`

Definition at line 54 of file ParticleDataTable.hh.

8.6.2.3 `typedef ParticleData HepPDT::ParticleDataTable::CPD`

Definition at line 47 of file ParticleDataTable.hh.

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

Definition at line 50 of file ParticleDataTable.hh.

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

Definition at line 51 of file ParticleDataTable.hh.

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

Definition at line 49 of file ParticleDataTable.hh.

8.6.3 Constructor & Destructor Documentation

8.6.3.1 `HepPDT::ParticleDataTable::ParticleDataTable (std::string name = " ", ProcessUnknownID * = new SimpleProcessUnknownID)`

Require a method to deal with unknown PID's.

Definition at line 18 of file ParticleDataTable.cc.

References `HepPDT::version()`.

8.6.3.2 `HepPDT::ParticleDataTable::~~ParticleDataTable ()`

8.6.4 Member Function Documentation

8.6.4.1 `const_iterator HepPDT::ParticleDataTable::begin () const [inline]`

begin iterating over the particle data table

Definition at line 68 of file ParticleDataTable.hh.

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

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

begin iterating over the map of particle names

Definition at line 74 of file ParticleDataTable.hh.

8.6.4.3 void HepPDT::ParticleDataTable::convertTemporaryMap (TempMap & *tempPDT*, std::ostream & *err*)

used by the **TableBuilder** (p. 120) destructor to fill the PDT

Definition at line 17 of file convertTemporaryMap.cc.

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

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

end iterating over the particle data table

Definition at line 70 of file ParticleDataTable.hh.

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

8.6.4.5 const_iteratorByName HepPDT::ParticleDataTable::endNameMap () const [inline]

end iterating over the map of particle names

Definition at line 76 of file ParticleDataTable.hh.

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

access particle information via a particle name

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

access particle information via a particle name

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

access particle information via **ParticleID** (p. 93)

8.6.4.9 ParticleData* HepPDT::ParticleDataTable::operator[] (ParticleID)

access particle information via **ParticleID** (p. 93)

8.6.4.10 ParticleData * HepPDT::ParticleDataTable::particle (std::string)

access particle information via a particle name

Definition at line 177 of file ParticleDataTable.cc.

References particle().

8.6.4.11 ParticleData const * HepPDT::ParticleDataTable::particle (std::string) const

access particle information via a particle name

Definition at line 188 of file ParticleDataTable.cc.

References particle().

8.6.4.12 ParticleData * HepPDT::ParticleDataTable::particle (ParticleID)

access particle information via **ParticleID** (p. 93)

Definition at line 141 of file ParticleDataTable.cc.

8.6.4.13 ParticleData const * HepPDT::ParticleDataTable::particle (ParticleID) const

access particle information via **ParticleID** (p. 93)

Examples:

examMyPDT.cc, testHepPDT.cc, and testReadParticleTable.cc.in.

Definition at line 153 of file ParticleDataTable.cc.

Referenced by duplicateFragmentTest(), main(), particle(), pdtFragmentTest(), pdtSimpleTest(), HepPDT::TestNuclearFragment::processUnknownID(), HepPDT::HeavyIonUnknownID::processUnknownID(), and testPDMMethods().

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

size of the particle data table

Definition at line 66 of file ParticleDataTable.hh.

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

8.6.4.15 int HepPDT::ParticleDataTable::sizeNameMap () const [inline]

size of the map of particle names

Definition at line 72 of file ParticleDataTable.hh.

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

return the name of this particle data table

Definition at line 78 of file ParticleDataTable.hh.

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

8.6.4.17 void HepPDT::ParticleDataTable::writeParticleData (std::ostream & ostr)

output all information in the PDT

Examples:

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

Definition at line 25 of file ParticleDataTable.cc.

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

Referenced by `duplicateFragmentTest()`, `main()`, and `pdtSimpleTest()`.

8.6.4.18 void HepPDT::ParticleDataTable::writeParticleInfo (std::ostream & *oustr*)

output all information about a particle EXCEPT its decays

Examples:

testHepPDT.cc, and testReadIsajet.cc.in.

Definition at line 51 of file ParticleDataTable.cc.

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

Referenced by `pdtSimpleTest()`.

8.6.4.19 void HepPDT::ParticleDataTable::writeParticleStatus (std::ostream &)

primarily useful for testing

This routine writes the particle name, ID, charge, mass, width, lifetime, and the results of `isStable()`.

Examples:

testReadParticleTable.cc.in.

Definition at line 88 of file ParticleDataTable.cc.

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

Referenced by `duplicateFragmentTest()`.

8.6.4.20 void HepPDT::ParticleDataTable::writeParticleTranslation (std::ostream & *oustr*)

output a list of original IDs and their translations

Examples:

listEvtGenNames.cc.in, listPDGNames.cc.in, listPythiaNames.cc.in, and testReadQQ.cc.in.

Definition at line 77 of file ParticleDataTable.cc.

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

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

- **ParticleDataTable.hh**
- **convertTemporaryMap.cc**
- **ParticleDataTable.cc**

8.7 HepPDT::ParticleDataTableComparison Class Reference

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

Public Member Functions

- **ParticleDataTableComparison ()**
stateless class
- **bool operator() (const ParticleID &, const ParticleID &) const**
If the absolute values are the same, the positive entry comes first.

8.7.1 Detailed Description

Author:

Lynn Garren

Definition at line 23 of file ParticleDataTableComparison.hh.

8.7.2 Constructor & Destructor Documentation

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

stateless class

Definition at line 27 of file ParticleDataTableComparison.hh.

8.7.3 Member Function Documentation

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

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**

8.8 HepPDT::ParticleID Class Reference

```
#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 isDyon () const**
is this a valid Dyon (magnetic monopole) ID?
- **bool isQBall () const**
Ad-hoc numbering for such particles is 100xxxx0, where xxxx is the charge in tenths.
- **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**
(e.g. outside the standard numbering scheme)
- **Quarks quarks () const**
returns a list of 3 constituent quarks
- **int threeCharge () const**
figure out the charge from the PID
- **double charge () const**
get the actual charge, which might be fractional
- **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 PDTname () const**
standard particle name

8.8.1 Detailed Description

Author:

Lynn Garren

Examples:

examMyPDT.cc.

Definition at line 64 of file ParticleID.hh.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 HepPDT::ParticleID::ParticleID (int *pid* = 0)

create from an integer ID

Definition at line 17 of file ParticleID.cc.

8.8.2.2 HepPDT::ParticleID::ParticleID (const ParticleID & *orig*)

Definition at line 21 of file ParticleID.cc.

8.8.3 Member Function Documentation

8.8.3.1 int HepPDT::ParticleID::A () const

if this is a nucleus (ion), get A

Examples:

testPID.cc.

Definition at line 247 of file ParticleID.cc.

References `abspid()`, `digit()`, `HepPDT::n10`, and `HepPDT::n9`.

Referenced by `isNucleus()`, and `testNucleus()`.

8.8.3.2 int HepPDT::ParticleID::abspid () const

get the absolute value

return a value greater than or equal to zero

Examples:

testPID.cc.

Definition at line 47 of file ParticleID.cc.

Referenced by A(), digit(), extraBits(), fundamentalID(), isBaryon(), isDiQuark(), isMeson(), isNucleus(), isQBall(), jSpin(), lambda(), lSpin(), HepPDT::ParticleDataTableComparison::operator>(), quarks(), s-Spin(), testUnknown(), threeCharge(), and Z().

8.8.3.3 double HepPDT::ParticleID::charge () const

get the actual charge, which might be fractional

Examples:

testPID.cc.

Definition at line 501 of file ParticleID.cc.

References isQBall(), and threeCharge().

Referenced by testValid(), and threeCharge().

8.8.3.4 unsigned short HepPDT::ParticleID::digit (location) const

return the digit at a named location in the PID

Examples:

testPID.cc.

Definition at line 54 of file ParticleID.cc.

References abspid().

Referenced by A(), fundamentalID(), isBaryon(), isDiQuark(), isDyon(), isMeson(), isNucleus(), isPentaquark(), isQBall(), isRhadron(), isSUSY(), lambda(), quarks(), testDiQuark(), testHadron(), testLepton(), testNucleus(), testUnknown(), threeCharge(), and Z().

8.8.3.5 int HepPDT::ParticleID::extraBits () const

(e.g. outside the standard numbering scheme)

Examples:

testPID.cc.

Definition at line 63 of file ParticleID.cc.

References `abspid()`.

Referenced by `isBaryon()`, `isDiQuark()`, `isDyon()`, `isHadron()`, `isLepton()`, `isMeson()`, `isPentaquark()`, `isQBall()`, `isRhadron()`, `isSUSY()`, `isValid()`, `jSpin()`, `quarks()`, `testDiQuark()`, `testHadron()`, `testLepton()`, `testNucleus()`, `testUnknown()`, and `threeCharge()`.

8.8.3.6 `int HepPDT::ParticleID::fundamentalID () const`

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

Examples:

`testPID.cc.`

Definition at line 71 of file ParticleID.cc.

References `abspid()`, `digit()`, `HepPDT::n10`, `HepPDT::n9`, `HepPDT::nq1`, and `HepPDT::nq2`.

Referenced by `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()`, `testDiQuark()`, `testHadron()`, `testLepton()`, `testNucleus()`, `testUnknown()`, and `threeCharge()`.

8.8.3.7 `bool HepPDT::ParticleID::hasBottom () const`

does this particle contain a bottom quark?

Examples:

`testPID.cc.`

Definition at line 103 of file ParticleID.cc.

References `HepPID::hasBottom()`.

Referenced by `testHasMethods()`.

8.8.3.8 `bool HepPDT::ParticleID::hasCharm () const`

does this particle contain a charm quark?

Examples:

`testPID.cc.`

Definition at line 98 of file ParticleID.cc.

References `HepPID::hasCharm()`.

Referenced by `testHasMethods()`.

8.8.3.9 bool HepPDT::ParticleID::hasDown () const

does this particle contain a down quark?

Examples:

testPID.cc.

Definition at line 88 of file ParticleID.cc.

References HepPID::hasDown().

Referenced by testHasMethods().

8.8.3.10 bool HepPDT::ParticleID::hasStrange () const

does this particle contain a strange quark?

Examples:

testPID.cc.

Definition at line 93 of file ParticleID.cc.

References HepPID::hasStrange().

Referenced by testHasMethods().

8.8.3.11 bool HepPDT::ParticleID::hasTop () const

does this particle contain a top quark?

Examples:

testPID.cc.

Definition at line 108 of file ParticleID.cc.

References HepPID::hasTop().

Referenced by testHasMethods().

8.8.3.12 bool HepPDT::ParticleID::hasUp () const

does this particle contain an up quark?

Examples:

testPID.cc.

Definition at line 83 of file ParticleID.cc.

References HepPID::hasUp().

Referenced by testHasMethods().

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

is this a valid baryon ID?

Examples:

testPID.cc.

Definition at line 334 of file ParticleID.cc.

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

Referenced by `HepPDT::ParticleData::isBaryon()`, `isHadron()`, `isValid()`, `testHadron()`, and `threeCharge()`.

8.8.3.14 bool HepPDT::ParticleID::isDiQuark () const

is this a valid diquark ID?

Examples:

testPID.cc.

Definition at line 316 of file ParticleID.cc.

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

Referenced by `HepPDT::ParticleData::isDiQuark()`, `isValid()`, `testDiQuark()`, and `testUnknown()`.

8.8.3.15 bool HepPDT::ParticleID::isDyon () const

is this a valid Dyon (magnetic monopole) ID?

Magnetic monopoles and Dyons are assumed to have one unit of Dirac monopole charge and a variable integer number xyz units of electric charge.

Codes 411xyz0 are then used when the magnetic and electrical charge sign agree and 412xyz0 when they disagree, with the overall sign of the particle set by the magnetic charge.

For now no spin information is provided.

Examples:

testPID.cc.

Definition at line 179 of file ParticleID.cc.

References `digit()`, `extraBits()`, `HepPDT::n`, `HepPDT::nj`, `HepPDT::nl`, `HepPDT::nq3`, and `HepPDT::nr`.

Referenced by `HepPDT::ParticleData::isDyon()`, `isValid()`, `quarks()`, `testUnknown()`, and `threeCharge()`.

8.8.3.16 bool HepPDT::ParticleID::isHadron () const

is this a valid hadron ID?

Examples:

testPID.cc.

Definition at line 306 of file ParticleID.cc.

References extraBits(), isBaryon(), isMeson(), and isPentaquark().

Referenced by HepPDT::ParticleData::isHadron(), testHadron(), and testUnknown().

8.8.3.17 bool HepPDT::ParticleID::isLepton () const

is this a valid lepton ID?

Examples:

testPID.cc.

Definition at line 298 of file ParticleID.cc.

References extraBits(), and fundamentalID().

Referenced by HepPDT::ParticleData::isLepton(), testLepton(), and testUnknown().

8.8.3.18 bool HepPDT::ParticleID::isMeson () const

is this a valid meson ID?

Examples:

testPID.cc.

Definition at line 274 of file ParticleID.cc.

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

Referenced by isHadron(), HepPDT::ParticleData::isMeson(), isValid(), lSpin(), sSpin(), and testHadron().

8.8.3.19 bool HepPDT::ParticleID::isNucleus () const

is this a valid ion ID?

Examples:

testPID.cc.

Definition at line 234 of file ParticleID.cc.

References A(), abspid(), digit(), HepPDT::n10, HepPDT::n9, and Z().

Referenced by HepPDT::ParticleData::isNucleus(), isValid(), lambda(), HepPDT::TestNuclear-Fragment::processUnknownID(), HepPDT::HeavyIonUnknownID::processUnknownID(), testNucleus(), testUnknown(), and threeCharge().

8.8.3.20 bool HepPDT::ParticleID::isPentaquark () const

is this a valid pentaquark ID?

Definition at line 142 of file ParticleID.cc.

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

Referenced by isHadron(), HepPDT::ParticleData::isPentaquark(), and isValid().

8.8.3.21 bool HepPDT::ParticleID::isQBall () const

Ad-hoc numbering for such particles is 100xxxx0, where xxxx is the charge in tenths.

Definition at line 202 of file ParticleID.cc.

References abspid(), digit(), extraBits(), HepPDT::n, HepPDT::nj, and HepPDT::nr.

Referenced by charge(), HepPDT::ParticleData::isQBall(), isValid(), HepPDT::detail::parseParticleLine(), and threeCharge().

8.8.3.22 bool HepPDT::ParticleID::isRhadron () const

is this a valid R-hadron ID?

Examples:

testPID.cc.

Definition at line 162 of file ParticleID.cc.

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

Referenced by HepPDT::ParticleData::isRhadron(), isValid(), testHadron(), and threeCharge().

8.8.3.23 bool HepPDT::ParticleID::isSUSY () const

is this a valid SUSY ID?

Examples:

testPID.cc.

Definition at line 217 of file ParticleID.cc.

References digit(), extraBits(), fundamentalID(), HepPDT::n, and HepPDT::nr.

Referenced by isRhadron(), HepPDT::ParticleData::isSUSY(), isValid(), and testUnknown().

8.8.3.24 bool HepPDT::ParticleID::isValid () const

is this a valid ID?

Examples:

testPID.cc.

Definition at line 114 of file ParticleID.cc.

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

Referenced by `testValid()`, and `HepPDT::ParticleData::write()`.

8.8.3.25 `int HepPDT::ParticleID::jSpin () const`

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

Examples:

`testPID.cc.`

Definition at line 345 of file ParticleID.cc.

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

Referenced by `HepPDT::TempParticleData::processPID()`, `testDiQuark()`, `testHadron()`, `testLepton()`, `testNucleus()`, `testUnknown()`, and `testValid()`.

8.8.3.26 `int HepPDT::ParticleID::lambda () const`

if this is a nucleus (ion), get `nLambda`

Examples:

`testPID.cc.`

Definition at line 265 of file ParticleID.cc.

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

Referenced by `testNucleus()`.

8.8.3.27 `int HepPDT::ParticleID::lSpin () const`

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

Examples:

`testPID.cc.`

Definition at line 362 of file ParticleID.cc.

References `abspid()`, `isMeson()`, and `HepPDT::nl`.

Referenced by `HepPDT::TempParticleData::processPID()`, `testDiQuark()`, `testHadron()`, `testLepton()`, `testUnknown()`, and `testValid()`.

8.8.3.28 `bool HepPDT::ParticleID::operator< (ParticleID const & other) const`

Definition at line 37 of file ParticleID.cc.

References `itsPID`.

8.8.3.29 ParticleID & HepPDT::ParticleID::operator= (const ParticleID &)

Definition at line 25 of file ParticleID.cc.

References swap().

8.8.3.30 bool HepPDT::ParticleID::operator== (ParticleID const & other) const

Definition at line 42 of file ParticleID.cc.

References itsPID.

8.8.3.31 const std::string HepPDT::ParticleID::PDTname () const [inline]

standard particle name

Examples:

testPID.cc.

Definition at line 157 of file ParticleID.hh.

References HepPID::particleName().

Referenced by HepPDT::ParticleData::PDTname(), and testHasMethods().

8.8.3.32 int HepPDT::ParticleID::pid () const [inline]

get the integer ID

Examples:

testPID.cc.

Definition at line 85 of file ParticleID.hh.

Referenced by HepPDT::TempParticleData::antiparticle(), isMeson(), HepPDT::ParticleDataTableComparison::operator(), HepPDT::parseEvtGenDecayLine(), HepPDT::parseEvtGenLine(), HepPDT::detail::parseIsajetLine(), HepPDT::detail::parseParticleLine(), HepPDT::detail::parsePythiaDecayLine(), HepPDT::detail::parsePythiaLine(), HepPDT::parseQQParticle(), HepPDT::ParticleData::pid(), HepPDT::TempParticleData::processPID(), testHadron(), testHasMethods(), threeCharge(), and HepPDT::ParticleData::write().

8.8.3.33 Quarks HepPDT::ParticleID::quarks () const

returns a list of 3 constituent quarks

Examples:

testPID.cc.

Definition at line 14 of file quarks.cc.

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

Referenced by `HepPDT::TempParticleData::processPID()`, `testDiQuark()`, `testHadron()`, `testLepton()`, `testUnknown()`, and `testValid()`.

8.8.3.34 `int HepPDT::ParticleID::sSpin () const`

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

Examples:

`testPID.cc`.

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

References `abspid()`, `isMeson()`, and `HepPDT::nl`.

Referenced by `HepPDT::TempParticleData::processPID()`, and `testValid()`.

8.8.3.35 `void HepPDT::ParticleID::swap (ParticleID & other)`

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

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

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

8.8.3.36 `int HepPDT::ParticleID::threeCharge () const`

figure out the charge from the PID

Examples:

`testPID.cc`.

Definition at line 437 of file `ParticleID.cc`.

References `abspid()`, `charge()`, `digit()`, `extraBits()`, `fundamentalID()`, `isBaryon()`, `isDyon()`, `isNucleus()`, `isQBall()`, `isRhadron()`, `HepPDT::nj`, `HepPDT::nl`, `HepPDT::nq1`, `HepPDT::nq2`, `HepPDT::nq3`, `pid()`, and `Z()`.

Referenced by `charge()`, `HepPDT::TempParticleData::processPID()`, `testDiQuark()`, `testHadron()`, `testLepton()`, `testNucleus()`, `testUnknown()`, and `testValid()`.

8.8.3.37 `int HepPDT::ParticleID::Z () const`

if this is a nucleus (ion), get Z

Examples:

`testPID.cc`.

Definition at line 256 of file `ParticleID.cc`.

References `abspid()`, `digit()`, `HepPDT::n10`, and `HepPDT::n9`.

Referenced by `isNucleus()`, `testNucleus()`, and `threeCharge()`.

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

- **ParticleID.hh**
- **ParticleID.cc**
- **quarks.cc**

8.9 HepPID::ParticleNameMap Class Reference

Public Types

- typedef ParticleIdMap::const_iterator **idIterator**
- typedef ParticleLookupMap::const_iterator **nameIterator**

Public Member Functions

- **ParticleNameMap** (ParticleIdMap m1, ParticleLookupMap m2)
- **~ParticleNameMap** ()
- **ParticleIdMap** nameMap () const
- **ParticleLookupMap** lookupMap () const
- **idIterator** begin () const
- **idIterator** end () const
- **idIterator** find (const int &id) const
- **nameIterator** beginLookupMap () const
- **nameIterator** endLookupMap () const
- **nameIterator** findString (const std::string &s) const

8.9.1 Detailed Description

Author:

Lynn Garren

Definition at line 44 of file ParticleName.cc.

8.9.2 Member Typedef Documentation

8.9.2.1 typedef ParticleIdMap::const_iterator HepPID::ParticleNameMap::idIterator

Definition at line 48 of file ParticleName.cc.

8.9.2.2 typedef ParticleLookupMap::const_iterator HepPID::ParticleNameMap::nameIterator

Definition at line 49 of file ParticleName.cc.

8.9.3 Constructor & Destructor Documentation

8.9.3.1 HepPID::ParticleNameMap::ParticleNameMap (ParticleIdMap *m1*, ParticleLookupMap *m2*) [inline]

Definition at line 51 of file ParticleName.cc.

8.9.3.2 HepPID::ParticleNameMap::~~ParticleNameMap () [inline]

Definition at line 53 of file ParticleName.cc.

8.9.4 Member Function Documentation

8.9.4.1 idIterator HepPID::ParticleNameMap::begin () const [inline]

Definition at line 57 of file ParticleName.cc.

8.9.4.2 nameIterator HepPID::ParticleNameMap::beginLookupMap () const [inline]

Definition at line 60 of file ParticleName.cc.

8.9.4.3 idIterator HepPID::ParticleNameMap::end () const [inline]

Definition at line 58 of file ParticleName.cc.

Referenced by HepPID::particleName(), and HepPID::validParticleName().

8.9.4.4 nameIterator HepPID::ParticleNameMap::endLookupMap () const [inline]

Definition at line 61 of file ParticleName.cc.

Referenced by HepPID::particleName(), and HepPID::validParticleName().

8.9.4.5 idIterator HepPID::ParticleNameMap::find (const int & id) const [inline]

Definition at line 59 of file ParticleName.cc.

Referenced by HepPID::particleName(), and HepPID::validParticleName().

8.9.4.6 nameIterator HepPID::ParticleNameMap::findString (const std::string & s) const [inline]

Definition at line 62 of file ParticleName.cc.

Referenced by HepPID::particleName(), and HepPID::validParticleName().

8.9.4.7 ParticleLookupMap HepPID::ParticleNameMap::lookupMap () const [inline]

Definition at line 56 of file ParticleName.cc.

8.9.4.8 ParticleIdMap HepPID::ParticleNameMap::nameMap () const [inline]

Definition at line 55 of file ParticleName.cc.

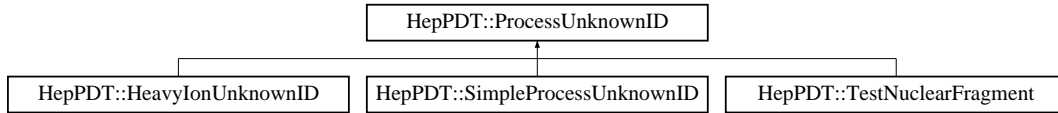
The documentation for this class was generated from the following file:

- ParticleName.cc

8.10 HepPDT::ProcessUnknownID Class Reference

```
#include <ProcessUnknownID.hh>
```

Inheritance diagram for HepPDT::ProcessUnknownID::



Public Member Functions

- **ParticleData * callProcessUnknownID (ParticleID, const ParticleDataTable &)**
safety wrapper to avoid secondary calls to processUnknownID
- **virtual ~ProcessUnknownID ()**
cleanup by ParticleDataTable (p. 86)

Protected Member Functions

- **ProcessUnknownID ()**

8.10.1 Detailed Description

Author:

Lynn Garren

Definition at line 28 of file ProcessUnknownID.hh.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 virtual HepPDT::ProcessUnknownID::~~ProcessUnknownID () [inline, virtual]

cleanup by **ParticleDataTable** (p. 86)

Definition at line 36 of file ProcessUnknownID.hh.

8.10.2.2 HepPDT::ProcessUnknownID::ProcessUnknownID () [inline, protected]

Definition at line 39 of file ProcessUnknownID.hh.

8.10.3 Member Function Documentation

8.10.3.1 ParticleData * HepPDT::ProcessUnknownID::callProcessUnknownID (ParticleID, const ParticleDataTable &)

safety wrapper to avoid secondary calls to processUnknownID

Definition at line 13 of file ProcessUnknownID.cc.

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

- **ProcessUnknownID.hh**
- **ProcessUnknownID.cc**

8.11 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**

8.11.1 Detailed Description

constituent quarks

Examples:

testPID.cc.

Definition at line 39 of file ParticleID.hh.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 HepPDT::Quarks::Quarks () [inline]

Definition at line 42 of file ParticleID.hh.

8.11.2.2 HepPDT::Quarks::Quarks (short q1, short q2, short q3) [inline]

Definition at line 43 of file ParticleID.hh.

8.11.3 Member Data Documentation

8.11.3.1 short HepPDT::Quarks::nq1

Examples:

testPID.cc.

Definition at line 46 of file ParticleID.hh.

Referenced by HepPDT::TempParticleData::processPID(), HepPDT::ParticleID::quarks(), testDiQuark(), testHadron(), testLepton(), testUnknown(), and testValid().

8.11.3.2 short HepPDT::Quarks::nq2

Examples:

testPID.cc.

Definition at line 47 of file ParticleID.hh.

Referenced by HepPDT::TempParticleData::processPID(), HepPDT::ParticleID::quarks(), testDiQuark(), testHadron(), testLepton(), testUnknown(), and testValid().

8.11.3.3 short HepPDT::Quarks::nq3

Examples:

testPID.cc.

Definition at line 48 of file ParticleID.hh.

Referenced by HepPDT::TempParticleData::processPID(), HepPDT::ParticleID::quarks(), testDiQuark(), testHadron(), testLepton(), testUnknown(), and testValid().

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

- **ParticleID.hh**

8.12 HepPDT::ResonanceStructure Class Reference

```
#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

8.12.1 Detailed Description

Author:

Lynn Garren

Definition at line 27 of file ResonanceStructure.hh.

8.12.2 Constructor & Destructor Documentation

8.12.2.1 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.

8.12.2.2 HepPDT::ResonanceStructure::~~ResonanceStructure () [virtual]

Definition at line 44 of file ResonanceStructure.cc.

8.12.2.3 HepPDT::ResonanceStructure::ResonanceStructure (const ResonanceStructure & *orig*)

Definition at line 21 of file ResonanceStructure.cc.

8.12.3 Member Function Documentation

8.12.3.1 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::ParticleData::lifetime().

8.12.3.2 double HepPDT::ResonanceStructure::lowerCutoff () const [inline]

lower cutoff of allowed width values

Definition at line 54 of file ResonanceStructure.hh.

Referenced by HepPDT::ParticleData::lowerCutoff().

8.12.3.3 Measurement const& HepPDT::ResonanceStructure::mass () const [inline]

get the mass

Definition at line 48 of file ResonanceStructure.hh.

Referenced by HepPDT::ParticleData::mass(), setMass(), and HepPDT::ParticleData::write().

8.12.3.4 ResonanceStructure & HepPDT::ResonanceStructure::operator= (const ResonanceStructure & *rhs*)

Definition at line 28 of file ResonanceStructure.cc.

References swap().

8.12.3.5 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::ParticleData::setLowerCutoff().

8.12.3.6 void HepPDT::ResonanceStructure::setMass (Measurement const & *mass*) [inline]

change the mass

Definition at line 61 of file ResonanceStructure.hh.

References mass().

Referenced by HepPDT::ParticleData::setMass().

8.12.3.7 void HepPDT::ResonanceStructure::setTotalWidth (Measurement const & *width*) [inline]

change the total width

Definition at line 63 of file ResonanceStructure.hh.

Referenced by HepPDT::ParticleData::setTotalWidth().

8.12.3.8 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::ParticleData::setTotalWidthFromLifetime().

8.12.3.9 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::ParticleData::setUpperCutoff().

8.12.3.10 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().

8.12.3.11 Measurement const& HepPDT::ResonanceStructure::totalWidth () const [inline]

get the total width

Definition at line 50 of file ResonanceStructure.hh.

Referenced by HepPDT::ParticleData::totalWidth(), and HepPDT::ParticleData::write().

8.12.3.12 double HepPDT::ResonanceStructure::upperCutoff () const [inline]

upper cutoff of allowed width values

Definition at line 56 of file ResonanceStructure.hh.

Referenced by HepPDT::ParticleData::upperCutoff().

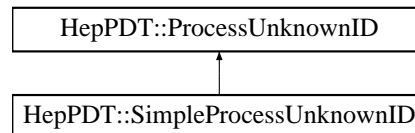
The documentation for this class was generated from the following files:

- **ResonanceStructure.hh**
- **lifetime.cc**
- **ResonanceStructure.cc**

8.13 HepPDT::SimpleProcessUnknownID Class Reference

```
#include <SimpleProcessUnknownID.hh>
```

Inheritance diagram for HepPDT::SimpleProcessUnknownID::



Public Member Functions

- **SimpleProcessUnknownID ()**
- **virtual ParticleData * processUnknownID (ParticleID, const ParticleDataTable &)**

8.13.1 Detailed Description

Author:

Lynn Garren

Definition at line 24 of file SimpleProcessUnknownID.hh.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 HepPDT::SimpleProcessUnknownID::SimpleProcessUnknownID () [inline]

Definition at line 26 of file SimpleProcessUnknownID.hh.

8.13.3 Member Function Documentation

8.13.3.1 virtual ParticleData* HepPDT::SimpleProcessUnknownID::processUnknownID (ParticleID, const ParticleDataTable &) [inline, virtual]

Implements **HepPDT::ProcessUnknownID p.** (classHepPDT₁₁*ProcessUnknownID* ??)

Definition at line 29 of file SimpleProcessUnknownID.hh.

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

- **SimpleProcessUnknownID.hh**

8.14 HepPDT::SpinState Class Reference

```
#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

8.14.1 Detailed Description

Author:

Lynn Garren

Examples:

`examMyPDT.cc.`

Definition at line 20 of file SpinState.hh.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 HepPDT::SpinState::SpinState (double ts = 0 . , double spin = 0 . , double oam = 0 .) [inline]

(default) constructor

8.14.2.2 `HepPDT::SpinState::SpinState (const SpinState & orig)` [inline]

8.14.3 Member Function Documentation

8.14.3.1 `SpinState& HepPDT::SpinState::operator= (const SpinState & rhs)` [inline]

8.14.3.2 `bool HepPDT::SpinState::operator== (const SpinState & rhs) const` [inline]

all three spins must match

8.14.3.3 `double HepPDT::SpinState::orbAngMom () const` [inline]

return the orbital angular momentum

Definition at line 47 of file SpinState.hh.

Referenced by HepPDT::ParticleData::write().

8.14.3.4 `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().

8.14.3.5 `void HepPDT::SpinState::setSpin (double spin)` [inline]

change the spin

Definition at line 54 of file SpinState.hh.

Referenced by HepPDT::TempParticleData::processPID().

8.14.3.6 `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().

8.14.3.7 `double HepPDT::SpinState::spin () const` [inline]

return the spin

Definition at line 45 of file SpinState.hh.

Referenced by HepPDT::ParticleData::write().

8.14.3.8 `void HepPDT::SpinState::swap (SpinState & other)` [inline]

Referenced by HepPDT::swap().

8.14.3.9 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::ParticleData::write().

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

- **SpinState.hh**

8.15 HepPDT::TableBuilder Class Reference

```
#include <TableBuilder.hh>
```

Public Member Functions

- **TableBuilder (ParticleDataTable &table, std::ostream &str=std::cerr)**
create TableBuilder (p. 120) from a ParticleDataTable (p. 86)
- **~TableBuilder ()**
call the ParticleDataTable (p. 86) conversion method upon destruction
- **TempParticleData & getParticleData (ParticleID pid)**
create a TempParticleData (p. 128) from a ParticleID (p. 93)
- **TempParticleData & getParticleData (std::string const &name)**
create a TempParticleData (p. 128) from a particle name
- **TempParticleData & getAntiParticle (ParticleID pid, const std::string &aname)**
create an antiparticle TempParticleData (p. 128) from a ParticleID (p. 93)
- **void addParticle (TempParticleData const &pd)**
add a TempParticleData (p. 128) to the map
- **void removeParticle (ParticleID pid)**
remove a TempParticleData (p. 128) 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

8.15.1 Detailed Description

Author:

Marc Paterno, Walter Brown, Lynn Garren

Examples:

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

Definition at line 42 of file TableBuilder.hh.

8.15.2 Constructor & Destructor Documentation

8.15.2.1 HepPDT::TableBuilder::TableBuilder (ParticleDataTable & table, std::ostream & str = std::cerr) [inline, explicit]

create **TableBuilder** (p. 120) from a **ParticleDataTable** (p. 86)

Definition at line 49 of file TableBuilder.hh.

8.15.2.2 HepPDT::TableBuilder::~~TableBuilder () [inline]

call the **ParticleDataTable** (p. 86) conversion method upon destruction

Definition at line 53 of file TableBuilder.hh.

References HepPDT::ParticleDataTable::convertTemporaryMap().

8.15.3 Member Function Documentation

8.15.3.1 void HepPDT::TableBuilder::addAlias (TempAliasData const & ad) [inline]

add alias information to the alias map

Referenced by HepPDT::addEvtGenParticles().

8.15.3.2 void HepPDT::TableBuilder::addParticle (TempParticleData const & pd) [inline]

add a **TempParticleData** (p. 128) to the map

Examples:

examMyPDT.cc.

Referenced by addData(), HepPDT::addEvtGenParticles(), and HepPDT::addQQParticles().

8.15.3.3 TempAliasData& HepPDT::TableBuilder::aliasData (std::string const & *alias*) [inline]

find an entry in the alias map

Referenced by HepPDT::addEvtGenParticles().

8.15.3.4 int HepPDT::TableBuilder::aliasSize () const [inline]

get size of alias map

Definition at line 87 of file TableBuilder.hh.

Referenced by HepPDT::addEvtGenParticles().

8.15.3.5 double HepPDT::TableBuilder::definition (std::string const & *def*) [inline]

return a parameter definition (for EvtGen)

Definition at line 91 of file TableBuilder.hh.

References HepPDT::DefTable::definition().

8.15.3.6 DefTable& HepPDT::TableBuilder::definitions () [inline]

get the list of definitions (for EvtGen)

Definition at line 89 of file TableBuilder.hh.

Referenced by HepPDT::addEvtGenParticles().

8.15.3.7 TempParticleData& HepPDT::TableBuilder::getAntiParticle (ParticleID *pid*, const std::string & *aname*) [inline]

create an antiparticle **TempParticleData** (p. 128) from a **ParticleID** (p. 93)

Referenced by HepPDT::addPythiaParticles().

8.15.3.8 TempParticleData& HepPDT::TableBuilder::getParticleData (std::string const & *name*) [inline]

create a **TempParticleData** (p. 128) from a particle name

8.15.3.9 TempParticleData& HepPDT::TableBuilder::getParticleData (ParticleID *pid*) [inline]

create a **TempParticleData** (p. 128) from a **ParticleID** (p. 93)

Examples:

examMyPDT.cc.

Referenced by `addData()`, `HepPDT::addEvtGenParticles()`, `HepPDT::addIsajetParticles()`, `HepPDT::addParticleTable()`, `HepPDT::addPDGParticles()`, `HepPDT::addPythiaParticles()`, and `HepPDT::addQQParticles()`.

8.15.3.10 `bool HepPDT::TableBuilder::hasAlias (std::string const & alias) [inline]`

check to see if this alias is already defined

Referenced by `HepPDT::addEvtGenParticles()`.

8.15.3.11 `bool HepPDT::TableBuilder::hasDefinition (std::string const & def) [inline]`

check to see if this particle name is already defined

Definition at line 80 of file `TableBuilder.hh`.

References `HepPDT::DefTable::hasDefinition()`.

8.15.3.12 `bool HepPDT::TableBuilder::hasParticleData (std::string const & name) [inline]`

check to see if this particle is already defined

Referenced by `HepPDT::addEvtGenParticles()`, and `HepPDT::addQQParticles()`.

8.15.3.13 `void HepPDT::TableBuilder::removeParticle (ParticleID pid) [inline]`

remove a `TempParticleData` (p. 128) from the map

Examples:

`examMyPDT.cc`.

Definition at line 68 of file `TableBuilder.hh`.

Referenced by `main()`.

8.15.3.14 `int HepPDT::TableBuilder::size () const [inline]`

get size of particle data map

Definition at line 85 of file `TableBuilder.hh`.

Referenced by `HepPDT::addEvtGenParticles()`, `HepPDT::addIsajetParticles()`, `HepPDT::addParticleTable()`, `HepPDT::addPDGParticles()`, `HepPDT::addPythiaParticles()`, and `HepPDT::addQQParticles()`.

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

- `TableBuilder.hh`

8.16 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

8.16.1 Detailed Description

Hold Alias information from EvtGen.

Definition at line 48 of file TempParticleData.hh.

8.16.2 Constructor & Destructor Documentation

8.16.2.1 HepPDT::TempAliasData::TempAliasData ()

used in a map<>

Definition at line 176 of file TempParticleData.cc.

8.16.3 Member Data Documentation

8.16.3.1 std::string HepPDT::TempAliasData::tempAlias

the alias

Definition at line 54 of file TempParticleData.hh.

Referenced by HepPDT::parseEvtGenAlias().

8.16.3.2 TDDlist HepPDT::TempAliasData::tempAliasDecayList

decay list for the alias

Definition at line 57 of file TempParticleData.hh.

8.16.3.3 std::string HepPDT::TempAliasData::tempAliasedParticle

the "real" particle

Definition at line 55 of file TempParticleData.hh.

Referenced by HepPDT::parseEvtGenAlias().

8.16.3.4 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().

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

- TempParticleData.hh
- TempParticleData.cc

8.17 HepPDT::TempConstituent Struct Reference

Temporary 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**

8.17.1 Detailed Description

Temporary constituent (e.g., quark) information.

Definition at line 25 of file TempParticleData.hh.

8.17.2 Constructor & Destructor Documentation

8.17.2.1 HepPDT::TempConstituent::TempConstituent (ParticleID p = ParticleID(0), int m = -1) [inline]

Definition at line 26 of file TempParticleData.hh.

8.17.3 Member Data Documentation

8.17.3.1 ParticleID HepPDT::TempConstituent::tempConstituentPID

Definition at line 28 of file TempParticleData.hh.

Referenced by HepPDT::TempParticleData::processPID().

8.17.3.2 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**

8.18 HepPDT::TempDecayData Struct Reference

temporary holder for decay data

```
#include <TempParticleData.hh>
```

Public Attributes

- **std::string tempDecayName**
Use string to hold int if necessary.
- **double tempBranchingFraction**
- **std::vector< std::string > tempDaughterList**
list of decay particles
- **std::vector< double > tempDecayParameters**
other decay parameters

8.18.1 Detailed Description

temporary holder for decay data

Definition at line 33 of file TempParticleData.hh.

8.18.2 Member Data Documentation

8.18.2.1 double HepPDT::TempDecayData::tempBranchingFraction

Definition at line 37 of file TempParticleData.hh.

8.18.2.2 std::vector<std::string> HepPDT::TempDecayData::tempDaughterList

list of decay particles

Definition at line 39 of file TempParticleData.hh.

8.18.2.3 std::string HepPDT::TempDecayData::tempDecayName

Use string to hold int if necessary.

Definition at line 36 of file TempParticleData.hh.

8.18.2.4 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**

8.19 HepPDT::TempParticleData Struct Reference

temporary holder for Particle Data information

```
#include <TempParticleData.hh>
```

Public Member Functions

- **TempParticleData ()**
used in a map<>
- **TempParticleData (int id)**
construct a basic TempParticleData (p. 128) from just the particle ID
- **TempParticleData (ParticleID pid)**
construct a basic TempParticleData (p. 128) from just the ParticleID (p. 93)
- **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. 128)
- **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**
- **TDDLlist tempDecayList**

8.19.1 Detailed Description

temporary holder for Particle Data information

Examples:

`examMyPDT.cc.`

Definition at line 61 of file TempParticleData.hh.

8.19.2 Constructor & Destructor Documentation

8.19.2.1 HepPDT::TempParticleData::TempParticleData ()

used in a map<>

Definition at line 15 of file TempParticleData.cc.

8.19.2.2 HepPDT::TempParticleData::TempParticleData (int *id*) [explicit]

construct a basic **TempParticleData** (p. 128) from just the particle ID

Definition at line 32 of file TempParticleData.cc.

References processPID().

8.19.2.3 HepPDT::TempParticleData::TempParticleData (ParticleID *pid*) [explicit]

construct a basic **TempParticleData** (p. 128) from just the **ParticleID** (p. 93)

Definition at line 50 of file TempParticleData.cc.

References processPID().

8.19.2.4 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. 128)

Definition at line 68 of file TempParticleData.cc.

8.19.2.5 HepPDT::TempParticleData::TempParticleData (TempParticleData const & *orig*)

Definition at line 106 of file TempParticleData.cc.

8.19.3 Member Function Documentation

8.19.3.1 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.

8.19.3.2 TempParticleData & HepPDT::TempParticleData::operator= (TempParticleData const & rhs)

Definition at line 122 of file TempParticleData.cc.

References swap().

8.19.3.3 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().

8.19.3.4 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().

8.19.4 Member Data Documentation

8.19.4.1 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().

8.19.4.2 double HepPDT::TempParticleData::tempColorCharge

Definition at line 94 of file TempParticleData.hh.

Referenced by antiparticle(), HepPDT::detail::parsePythiaLine(), and swap().

8.19.4.3 TDDlist HepPDT::TempParticleData::tempDecayList

Definition at line 101 of file TempParticleData.hh.

Referenced by antiparticle(), and swap().

8.19.4.4 double HepPDT::TempParticleData::tempHighCutoff

Definition at line 99 of file TempParticleData.hh.

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

8.19.4.5 ParticleID HepPDT::TempParticleData::tempID

Definition at line 89 of file TempParticleData.hh.

Referenced by antiparticle(), HepPDT::parseEvtGenDecayLine(), HepPDT::parseEvtGenLine(), HepPDT::detail::parseIsajetLine(), HepPDT::detail::parseParticleLine(), HepPDT::detail::parsePythiaDecayLine(), HepPDT::detail::parsePythiaLine(), HepPDT::parseQQParticle(), processPID(), and swap().

8.19.4.6 double HepPDT::TempParticleData::tempLowCutoff

Definition at line 98 of file TempParticleData.hh.

Referenced by antiparticle(), HepPDT::parseQQParticle(), and swap().

8.19.4.7 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(), HepPDT::TestNuclearFragment::processUnknownID(), and swap().

8.19.4.8 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().

8.19.4.9 std::string HepPDT::TempParticleData::tempParticleName

Examples:

examMyPDT.cc.

Definition at line 90 of file TempParticleData.hh.

Referenced by addData(), HepPDT::addQQParticles(), antiparticle(), HepPDT::parseEvtGenLine(), HepPDT::detail::parseIsajetLine(), HepPDT::detail::parseParticleLine(), HepPDT::detail::parsePythiaLine(), HepPDT::parseQQParticle(), HepPDT::TestNuclearFragment::processUnknownID(), and swap().

8.19.4.10 `std::vector<TempConstituent> HepPDT::TempParticleData::tempQuarks`

Definition at line 100 of file TempParticleData.hh.

Referenced by processPID(), and swap().

8.19.4.11 `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().

8.19.4.12 `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().

8.19.4.13 `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().

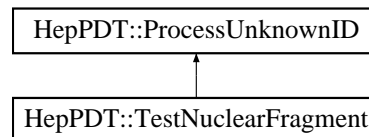
The documentation for this struct was generated from the following files:

- `TempParticleData.hh`
- `TempParticleData.cc`

8.20 HepPDT::TestNuclearFragment Class Reference

```
#include <TestNuclearFragment.hh>
```

Inheritance diagram for HepPDT::TestNuclearFragment::



Public Member Functions

- `TestNuclearFragment ()`
- `virtual ParticleData * processUnknownID (ParticleID, const ParticleDataTable &pd)`

8.20.1 Detailed Description

Author:

Lynn Garren

Examples:

`testHepPDT.cc.`

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

8.20.2 Constructor & Destructor Documentation

8.20.2.1 HepPDT::TestNuclearFragment::TestNuclearFragment () [inline]

Definition at line 31 of file `TestNuclearFragment.hh`.

8.20.3 Member Function Documentation

8.20.3.1 ParticleData * HepPDT::TestNuclearFragment::processUnknownID (ParticleID, const ParticleDataTable &pd) [inline, virtual]

Implements `HepPDT::ProcessUnknownID p.` (`classHepPDT11ProcessUnknownID ??`)

Definition at line 39 of file `TestNuclearFragment.hh`.

References `HepPDT::ParticleID::isNucleus()`, `HepPDT::ParticleData::mass()`, `HepPDT::ParticleDataTable::particle()`, `HepPDT::TempParticleData::tempMass`, and `HepPDT::TempParticleData::tempParticleName`.

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

- `TestNuclearFragment.hh`

Chapter 9

HepPDT File Documentation

9.1 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** (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)
- **bool HepPDT::addEvtGenParticles** (std::istream &, TableBuilder &)
read EvtGen input and add particles to the table

9.2 addHerwigParticles.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/TableBuilder.hh"
```

Namespaces

- namespace **HepPDT**

Functions

- bool **HepPDT::addHerwigParticles** (std::istream &, TableBuilder &)

9.3 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 &, TableBuilder &)
read Isajet particle input and add particles to the table
- **void HepPDT::detail::parseIsajetLine** (TempParticleData &, const std::string &)
for internal use

9.4 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 &, TableBuilder &, bool validate=false)
validate=true => verify that the ParticleID (p. 93) is valid
- **bool HepPDT::detail::getParticleID** (int &id, const std::string &)
for internal use
- **void HepPDT::detail::parseParticleLine** (TempParticleData &, const std::string &)
for internal use

9.5 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 &, TableBuilder &)
read PDG input and add particles to the table
- **void HepPDT::detail::parsePDGline** (TempParticleData &, std::string &)
for internal use
- **bool HepPDT::detail::CheckPDGEntry** (TempParticleData &, const std::string &, double, double)
for internal use

9.6 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 &, TableBuilder &)
read Pythia input and add particles to the table
- **void HepPDT::detail::parsePythiaLine** (TempParticleData &, int &, std::string &, const std::string &)
for internal use
- **void HepPDT::detail::parsePythiaDecayLine** (TempParticleData &, const std::string &)
for internal use

9.7 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"
```

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)
- **void HepPDT::parseQQParticle** (TempParticleData &tpd, const std::string &pdline)
- **bool HepPDT::addQQParticles** (std::istream &, TableBuilder &)
read QQ input and add particles to the table

9.8 calculateWidthFromLifetime.cc File Reference

```
#include "HepPDT/defs.h"
#include "HepPDT/TempParticleData.hh"
```

Namespaces

- namespace **HepPDT**

Functions

- double **HepPDT::calculateWidthFromLifetime** (double)
Given the lifetime, calculate the width.

9.9 Constituent.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/Constituent.hh"
```

Namespaces

- namespace **HepPDT**

9.10 Constituent.hh File Reference

```
#include <algorithm>
#include "HepPDT/ParticleID.hh"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::Constituent**

Functions

- void **HepPDT::swap** (**Constituent &first**, **Constituent &second**)

9.11 convertTemporaryMap.cc File Reference

```
#include "HepPDT/defs.h"
#include "HepPDT/ParticleID.hh"
#include "HepPDT/ParticleDataTable.hh"
#include "HepPDT/ParticleData.hh"
#include "HepPDT/TempParticleData.hh"
#include "HepPDT/Version.hh"
```

Namespaces

- namespace **HepPDT**

9.12 DefTable.cc File Reference

```
#include <iostream>
#include "HepPDT/defs.h"
#include "HepPDT/DefTable.hh"
```

Namespaces

- namespace **HepPDT**

9.13 DefTable.hh File Reference

```
#include <string>
#include <map>
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::DefTable**

9.14 examListHerwig.cc File Reference

```
#include <fstream>
#include <iostream>
#include <cstdlib>
#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 ()`

9.14.1 Function Documentation

9.14.1.1 `void get_herwig_name_ (int *ihwg, int *id, char *name)`

Examples:

`examListHerwig.cc.`

Referenced by `main()`.

9.14.1.2 `void get_list_size_ (int *)`

Examples:

`examListHerwig.cc.`

Referenced by `main()`.

9.14.1.3 `void list_herwig_end_ ()`

Examples:

`examListHerwig.cc.`

Referenced by `main()`.

9.14.1.4 `void list_herwig_init_ (int *nevt)`

Author:

Lynn Garren

Examples:

examListHerwig.cc.

Referenced by main().

9.14.1.5 int main ()

Definition at line 30 of file examListHerwig.cc.

References `get_herwig_name_()`, `get_list_size_()`, `list_herwig_end_()`, `list_herwig_init_()`, `HepPID::particleName()`, `HepPID::translateHerwigtoPDT()`, and `HepPID::writeVersion()`.

9.15 examListIsajet.cc File Reference

```
#include <fstream>
#include <iostream>
#include <cstdlib>
#include <cstring>
#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 ()`

9.15.1 Function Documentation

9.15.1.1 void `flavor_ (int *, int *, int *, int *, int *, int *)`

Examples:

`examListIsajet.cc`.

Referenced by `main()`.

9.15.1.2 void `get_label_ (int *id, char *name)`

Examples:

`examListIsajet.cc`.

Referenced by `main()`.

9.15.1.3 void `list_isajet_init_ ()`

Author:

Lynn Garren

Examples:

`examListIsajet.cc`.

Referenced by `main()`.

9.15.1.4 int main ()

Definition at line 29 of file examListIsajet.cc.

References `flavor_()`, `get_label_()`, `list_isajet_init_()`, `HepPID::particleName()`, `HepPID::translateIsajettoPDT()`, and `HepPID::writeVersion()`.

9.16 examListPythia.cc File Reference

```
#include <fstream>
#include <string>
#include <cstdlib>
#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 ()`

9.16.1 Function Documentation

9.16.1.1 void `getkf_ (int *, int *)`

Examples:

`examListPythia.cc.`

Referenced by `main()`.

9.16.1.2 void `getpyname_ (int *, int *, char * name)`

Examples:

`examListPythia.cc.`

Referenced by `main()`.

9.16.1.3 void `list_pythia_()`

Author:

Lynn Garren

Examples:

`examListPythia.cc.`

Referenced by `main()`.

9.16.1.4 int main ()

Definition at line 32 of file examListPythia.cc.

References `getkf_()`, `getpyname_()`, `HepPID::isValid()`, `list_pythia_()`, `HepPID::particleName()`, `HepPID::translatePythiatoPDT()`, `writeLine()`, and `HepPID::writeVersion()`.

9.16.1.5 void writeLine (int & *i*, int & *kf*, int & *id*, std::string & *name*, std::string & *pn*, std::ofstream & *os*)

Examples:

`examListPythia.cc`.

Definition at line 79 of file examListPythia.cc.

Referenced by `main()`.

9.17 examMyPDT.cc File Reference

```
#include "HepPDT/defs.h"
#include <fstream>
#include <cstdlib>
#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 ()

9.17.1 Function Documentation

9.17.1.1 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

Examples:

examMyPDT.cc.

Definition at line 68 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().

9.17.1.2 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, testReadParticleTable.cc.in, and testReadQQ.cc.in.

Definition at line 24 of file examMyPDT.cc.

References addData(), HepPDT::ParticleData::name(), HepPDT::ParticleDataTable::particle(), HepPDT::TableBuilder::removeParticle(), and HepPDT::ParticleDataTable::writeParticleData().

9.18 getIsajetID.cc File Reference

```
#include <iostream>
#include <string>
#include <sstream>
```

Namespaces

- namespace **HepPDT**
- namespace **HepPDT::detail**

Functions

- bool **HepPDT::detail::getIsajetID** (int &, const std::string &)
for internal use

9.19 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 > &, std::string &)
for internal use
- void **HepPDT::detail::getPDGnames** (std::vector< std::string > &, std::string &)
for internal use

9.20 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 &, const std::string &)
for internal use

9.21 hasMethods.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/ParticleData.hh"
```

Namespaces

- namespace **HepPDT**

9.22 HeavyIonUnknownID.cc File Reference

```
#include "HepPDT/HeavyIonUnknownID.hh"  
#include "HepPDT/ParticleDataTable.hh"
```

Namespaces

- namespace **HepPDT**

9.23 HeavyIonUnknownID.hh File Reference

```
#include "HepPDT/ProcessUnknownID.hh"  
#include "HepPDT/ParticleData.hh"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::HeavyIonUnknownID**

9.24 lifetime.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/ResonanceStructure.hh"
```

Namespaces

- namespace **HepPDT**

9.25 list_of_examples.cc File Reference

9.26 list_of_tests.cc File Reference

9.27 listEvtGenTranslation.cc File Reference

```
#include <fstream>
#include <iostream>
#include <cstdlib>
#include "HepPID/ParticleIDTranslations.hh"
```

Functions

- `int main ()`

9.27.1 Function Documentation

9.27.1.1 `int main ()`

Definition at line 15 of file listEvtGenTranslation.cc.

References HepPID::writeEvtGenTranslation().

9.28 listHerwigTranslation.cc File Reference

```
#include <fstream>
#include <iostream>
#include <cstdlib>
#include "HepPID/ParticleIDTranslations.hh"
```

Functions

- `int main ()`

9.28.1 Function Documentation

9.28.1.1 `int main ()`

Definition at line 15 of file listHerwigTranslation.cc.

References `HepPID::writeHerwigTranslation()`.

9.29 listIsajetTranslation.cc File Reference

```
#include <fstream>
#include <iostream>
#include <cstdlib>
#include "HepPID/ParticleIDTranslations.hh"
```

Functions

- `int main ()`

9.29.1 Function Documentation

9.29.1.1 `int main ()`

Definition at line 15 of file `listIsajetTranslation.cc`.

References `HepPID::writeIsajetTranslation()`.

9.30 listParticleNames.cc File Reference

```
#include <fstream>
#include <iostream>
#include <cstdlib>
#include "HepPID/ParticleName.hh"
```

Functions

- `int main ()`

9.30.1 Function Documentation

9.30.1.1 `int main ()`

Definition at line 17 of file listParticleNames.cc.

References HepPID::listParticleNames().

9.31 listPDGTranslation.cc File Reference

```
#include <fstream>
#include <iostream>
#include <cstdlib>
#include "HepPID/ParticleIDTranslations.hh"
```

Functions

- `int main ()`

9.31.1 Function Documentation

9.31.1.1 `int main ()`

Definition at line 15 of file listPDGTranslation.cc.

References `HepPID::writePDGTranslation()`.

9.32 listPythiaTranslation.cc File Reference

```
#include <fstream>
#include <iostream>
#include <cstdlib>
#include "HepPID/ParticleIDTranslations.hh"
```

Functions

- `int main ()`

9.32.1 Function Documentation

9.32.1.1 `int main ()`

Definition at line 15 of file listPythiaTranslation.cc.

References `HepPID::writePythiaTranslation()`.

9.33 listQQTranslation.cc File Reference

```
#include <fstream>
#include <iostream>
#include <cstdlib>
#include "HepPID/ParticleIDTranslations.hh"
```

Functions

- `int main ()`

9.33.1 Function Documentation

9.33.1.1 `int main ()`

Definition at line 15 of file listQQTranslation.cc.

References `HepPID::writeQQTranslation()`.

9.34 Measurement.hh File Reference

```
#include "HepPDT/Measurement.icc"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::Measurement**

Functions

- void **HepPDT::swap** (**Measurement &first**, **Measurement &second**)

9.35 ParticleData.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/ParticleData.icc"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::ParticleData**

Functions

- void **HepPDT::swap** (ParticleData &first, ParticleData &second)

9.36 ParticleDataTable.cc File Reference

```
#include <iostream>
#include <string>
#include <map>
#include "HepPDT/defs.h"
#include "HepPDT/ParticleDataTable.hh"
```

Namespaces

- namespace **HepPDT**

9.37 ParticleDataTable.hh File Reference

```
#include <iostream>
#include <string>
#include <map>
#include "HepPDT/ParticleID.hh"
#include "HepPDT/ParticleData.hh"
#include "HepPDT/ParticleDataTableComparison.hh"
#include "HepPDT/ProcessUnknownID.hh"
#include "HepPDT/SimpleProcessUnknownID.hh"
#include "HepPDT/Version.hh"
#include "HepPDT/ParticleDataTable.icc"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::ParticleDataTable**

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)**

9.38 ParticleDataTableComparison.hh File Reference

```
#include "HepPDT/ParticleID.hh"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::ParticleDataTableComparison**

9.39 ParticleID.cc File Reference

```
#include <stdlib.h>
#include <cmath>
#include "HepPDT/defs.h"
#include "HepPDT/ParticleID.hh"
#include "HepPID/ParticleIDMethods.hh"
```

Namespaces

- namespace **HepPDT**

9.40 ParticleID.hh File Reference

```
#include <string>
#include <algorithm>
#include "HepPID/ParticleName.hh"
#include "HepPID/ParticleIDTranslations.hh"
```

Namespaces

- namespace **HepPDT**

Classes

- struct **HepPDT::Quarks**
constituent quarks
- class **HepPDT::ParticleID**

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** }
 The location enum provides a convenient index into the PID.

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)

9.41 ParticleIDMethods.cc File Reference

```
#include <cmath>
#include "HepPID/ParticleIDMethods.hh"
#include "HepPID/ParticleName.hh"
```

Namespaces

- namespace **HepPID**

Functions

- **bool HepPID::findQ (const int &pid, const int &q)**
- **int HepPID::abspid (const int &pid)**
absolute value of particle ID
- **int HepPID::extraBits (const int &pid)**
(e.g. outside the standard numbering scheme)
- **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)**
extract fundamental ID (1-100) if this is a "fundamental" particle
- **int HepPID::Z (const int &pid)**
Ion numbers are +/- 10LZZZAAAI.
- **int HepPID::A (const int &pid)**
Ion numbers are +/- 10LZZZAAAI.
- **int HepPID::lambda (const int &pid)**
Ion numbers are +/- 10LZZZAAAI.
- **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::isDyon (const int &pid)**
is this a valid Dyon (magnetic monopole) ID?
- **bool HepPID::isQBall (const int &pid)**
Ad-hoc numbering for such particles is 100xxxx0, where xxxx is the charge in tenths.
- **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 $2L+I$, where L is the orbital angular momentum

- **int HepPID::threeCharge (const int &pid)**

If this is a Q-ball, return 30 times the charge.

- **double HepPID::charge (const int &pid)**

return the actual charge

9.42 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** }

The location enum provides a convenient index into the PID.

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)
Ion numbers are +/- 10LZZZAAAI.
- int **HepPID::Z** (const int &pid)
Ion numbers are +/- 10LZZZAAAI.
- int **HepPID::lambda** (const int &pid)
Ion numbers are +/- 10LZZZAAAI.
- int **HepPID::abspid** (const int &pid)
absolute value of particle ID
- int **HepPID::fundamentalID** (const int &pid)
extract fundamental ID (1-100) 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)
(e.g. outside the standard numbering scheme)
- 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::isDyon (const int &pid)**

is this a valid Dyon (magnetic monopole) ID?

- **bool HepPID::isQBall (const int &pid)**

Ad-hoc numbering for such particles is 100xxxx0, where xxxx is the charge in tenths.

- **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)**
If this is a Q -ball, return 30 times the charge.
- **double HepPID::charge (const int &pid)**
return the actual charge

9.43 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::translateGeantttoPDT (const int geantID)**
translate Geant3 to PDG standard
- **int HepPID::translatePDTtoGeant (const int pid)**
translate PDG standard to Geant3

9.44 ParticleName.cc File Reference

```
#include <string>
#include <map>
#include <iostream>
#include <sstream>
#include <iomanip>
#include <utility>
#include "HepPID/ParticleName.hh"
#include "HepPID/ParticleIDMethods.hh"
#include "HepPID/Version.hh"
```

Namespaces

- namespace **HepPID**

Classes

- class **HepPID::ParticleNameMap**

Typedefs

- typedef std::map< int, std::string > **HepPID::ParticleIdMap**
- typedef std::map< std::string, int > **HepPID::ParticleLookupMap**

Functions

- ParticleNameMap const & **HepPID::ParticleNameInit** ()
- void **HepPID::writeParticleNameLine** (int i, std::ostream &os)
- std::string **HepPID::dyonName** (const int &pid)
- std::string **HepPID::qballName** (const int &pid)
- int **HepPID::checkForSpecialParticle** (const std::string &s)
- ParticleNameMap const & **HepPID::getParticleNameMap** ()
access the ParticleNameMap (p. 106) for other purposes
- bool **HepPID::validParticleName** (const int &)
verify that this number has a valid name
- bool **HepPID::validParticleName** (const std::string &)
verify that this string has a valid id
- std::string **HepPID::particleName** (const int &)
get a known HepPID (p. 37) Particle name
- int **HepPID::particleName** (const std::string &)

lookup a known ID

- **void HepPID::listParticleNames (std::ostream &os)**

list all known names

9.45 ParticleName.hh File Reference

```
#include <string>
#include <map>
#include <iostream>
```

Namespaces

- namespace **HepPID**

Functions

- **std::string HepPID::particleName (const int &)**
get a known HepPID (p. 37) Particle name
- **int HepPID::particleName (const std::string &)**
lookup a known ID
- **void HepPID::listParticleNames (std::ostream &os)**
list all known names
- **bool HepPID::validParticleName (const int &)**
verify that this number has a valid name
- **bool HepPID::validParticleName (const std::string &)**
verify that this string has a valid id
- **ParticleNameMap const & HepPID::getParticleNameMap ()**
access the ParticleNameMap (p. 106) for other purposes

9.46 ProcessUnknownID.cc File Reference

```
#include "HepPDT/ProcessUnknownID.hh"  
#include "HepPDT/ParticleDataTable.hh"
```

Namespaces

- namespace **HepPDT**

9.47 ProcessUnknownID.hh File Reference

```
#include "HepPDT/ParticleID.hh"  
#include "HepPDT/ParticleData.hh"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::ProcessUnknownID**

9.48 quarks.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/ParticleID.hh"
```

Namespaces

- namespace **HepPDT**

9.49 ResonanceStructure.cc File Reference

```
#include "HepPDT/defs.h"  
#include "HepPDT/ResonanceStructure.hh"
```

Namespaces

- namespace **HepPDT**

9.50 ResonanceStructure.hh File Reference

```
#include <algorithm>
#include "HepPDT/Measurement.hh"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::ResonanceStructure**

Functions

- void **HepPDT::swap (ResonanceStructure &first, ResonanceStructure &second)**

9.51 SimpleProcessUnknownID.hh File Reference

```
#include "HepPDT/ProcessUnknownID.hh"  
#include "HepPDT/ParticleData.hh"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::SimpleProcessUnknownID**

9.52 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 $2J+1$

9.53 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

9.54 SpinState.hh File Reference

```
#include "HepPDT/SpinState.icc"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::SpinState**

Functions

- void **HepPDT::swap** (SpinState &first, SpinState &second)

9.55 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

9.56 stringtodouble.hh File Reference

```
#include <string>
```

Namespaces

- namespace **HepPDT**

Functions

- double **HepPDT::stringtodouble** (std::string &numb)
extract a double from a string

9.57 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**

Functions

- **bool HepPDT::addPDGParticles (std::istream &, TableBuilder &)**
read PDG input and add particles to the table
- **bool HepPDT::addPythiaParticles (std::istream &, TableBuilder &)**
read Pythia input and add particles to the table
- **bool HepPDT::addHerwigParticles (std::istream &, TableBuilder &)**
- **bool HepPDT::addIsajetParticles (std::istream &, TableBuilder &)**
read Isajet particle input and add particles to the table
- **bool HepPDT::addIsajetDecay (std::istream &, TableBuilder &)**
read Isajet decay input and add decay information to the table
- **bool HepPDT::addQQParticles (std::istream &, TableBuilder &)**
read QQ input and add particles to the table
- **bool HepPDT::addEvtGenParticles (std::istream &, TableBuilder &)**
read EvtGen input and add particles to the table
- **bool HepPDT::addParticleTable (std::istream &, TableBuilder &, bool validate=false)**
validate=true => verify that the ParticleID (p. 93) is valid
- **void HepPDT::detail::getPDGpid (std::vector< int > &, std::string &)**
for internal use

- **void HepPDT::detail::getPDGnames** (std::vector< std::string > &, std::string &)
for internal use
- **void HepPDT::detail::parsePDGline** (TempParticleData &, std::string &)
for internal use
- **bool HepPDT::detail::CheckPDGEntry** (TempParticleData &, const std::string &, double, double)
for internal use
- **bool HepPDT::detail::getPythiaid** (int &, const std::string &)
for internal use
- **void HepPDT::detail::parsePythiaLine** (TempParticleData &, int &, std::string &, const std::string &)
for internal use
- **void HepPDT::detail::parsePythiaDecayLine** (TempParticleData &, const std::string &)
for internal use
- **TempDecayData HepPDT::detail::getPythiaDecay** (const std::string &)
for internal use
- **bool HepPDT::detail::getIsajetID** (int &, const std::string &)
for internal use
- **void HepPDT::detail::parseIsajetLine** (TempParticleData &, const std::string &)
for internal use
- **void HepPDT::detail::parseIsajetDecayLine** (TempParticleData &, const std::string &, TableBuilder &)
for internal use
- **bool HepPDT::detail::getParticleID** (int &id, const std::string &)
for internal use
- **void HepPDT::detail::parseParticleLine** (TempParticleData &, const std::string &)
for internal use

9.58 TempParticleData.cc File Reference

```
#include <algorithm>
#include "HepPDT/defs.h"
#include "HepPDT/TempParticleData.hh"
```

Namespaces

- namespace **HepPDT**

9.59 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**
Temporary 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)
Given the lifetime, calculate the width.
- void **HepPDT::swap** (TempParticleData &first, TempParticleData &second)

9.60 testHepPDT.cc File Reference

```
#include <fstream>
#include <iomanip>
#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
#include "HepPDT/HeavyIonUnknownID.hh"
#include "TestNuclearFragment.hh"
```

Functions

- void **pdtSimpleTest** (char[300], std::ofstream &)
- void **pdtFragmentTest** (char[300], std::ofstream &)
- void **duplicateFragmentTest** (char[300], std::ofstream &)
- void **testPDMMethods** (HepPDT::ParticleDataTable &, std::ofstream &)
- int **main** ()

9.60.1 Function Documentation

9.60.1.1 void duplicateFragmentTest (char[300], std::ofstream &)

Examples:

testHepPDT.cc.

Definition at line 140 of file testHepPDT.cc.

References HepPDT::addPDGParticles(), HepPDT::ParticleDataTable::particle(), HepPDT::ParticleData::write(), HepPDT::ParticleDataTable::writeParticleData(), and HepPDT::ParticleDataTable::writeParticleStatus().

Referenced by main().

9.60.1.2 int main ()

Definition at line 27 of file testHepPDT.cc.

References duplicateFragmentTest(), pdtFragmentTest(), and pdtSimpleTest().

9.60.1.3 void pdtFragmentTest (char[300], std::ofstream &)

Examples:

testHepPDT.cc.

Definition at line 108 of file testHepPDT.cc.

References HepPDT::addPDGParticles(), HepPDT::ParticleDataTable::particle(), and HepPDT::ParticleData::write().

Referenced by main().

9.60.1.4 void pdtSimpleTest (char[300], std::ofstream &)

Examples:

testHepPDT.cc.

Definition at line 49 of file testHepPDT.cc.

References HepPDT::addPDGParticles(), HepPDT::ParticleData::lowerCutoff(), HepPDT::ParticleDataTable::particle(), testPDMMethods(), HepPDT::ParticleData::totalWidth(), HepPDT::ParticleData::upperCutoff(), HepPDT::ParticleData::write(), HepPDT::ParticleDataTable::writeParticleData(), and HepPDT::ParticleDataTable::writeParticleInfo().

Referenced by main().

9.60.1.5 void testPDMMethods (HepPDT::ParticleDataTable &, std::ofstream &)

Examples:

testHepPDT.cc.

Definition at line 194 of file testHepPDT.cc.

References HepPDT::ParticleData::hasBottom(), HepPDT::ParticleData::hasCharm(), HepPDT::ParticleData::hasDown(), HepPDT::ParticleData::hasStrange(), HepPDT::ParticleData::hasTop(), HepPDT::ParticleData::hasUp(), HepPDT::ParticleData::name(), HepPDT::ParticleDataTable::particle(), and HepPDT::ParticleData::pid().

Referenced by pdtSimpleTest().

9.61 TestNuclearFragment.hh File Reference

```
#include <sstream>
#include "HepPDT/ProcessUnknownID.hh"
#include "HepPDT/ParticleData.hh"
```

Namespaces

- namespace **HepPDT**

Classes

- class **HepPDT::TestNuclearFragment**

9.62 testParticleIDMethods.cc File Reference

```
#include <fstream>
#include <iostream>
#include <iomanip>
#include <cstdlib>
#include "HepPID/ParticleIDMethods.hh"
#include "HepPID/ParticleName.hh"
```

Functions

- `int main ()`

9.62.1 Function Documentation

9.62.1.1 `int main ()`

Definition at line 19 of file testParticleIDMethods.cc.

References `HepPID::A()`, `HepPID::abspid()`, `HepPID::charge()`, `HepPID::digit()`, `HepPID::extraBits()`, `HepPID::fundamentalID()`, `HepPID::hasBottom()`, `HepPID::hasCharm()`, `HepPID::hasDown()`, `HepPID::hasStrange()`, `HepPID::hasTop()`, `HepPID::hasUp()`, `HepPID::isBaryon()`, `HepPID::isDiQuark()`, `HepPID::isDyon()`, `HepPID::isHadron()`, `HepPID::isLepton()`, `HepPID::isMeson()`, `HepPID::isNucleus()`, `HepPID::isPentaquark()`, `HepPID::isQBall()`, `HepPID::isRhadron()`, `HepPID::isSUSY()`, `HepPID::isValid()`, `HepPID::jSpin()`, `HepPID::lambda()`, `HepPID::lSpin()`, `HepPID::n`, `HepPID::n10`, `HepPID::nj`, `HepPID::nl`, `HepPID::nq1`, `HepPID::nq2`, `HepPID::nq3`, `HepPID::nr`, `HepPDT::nr`, `HepPID::particleName()`, `HepPID::sSpin()`, `HepPID::threeCharge()`, and `HepPID::Z()`.

9.63 testPID.cc File Reference

```
#include <iostream>
#include <iomanip>
#include "HepPDT/defs.h"
#include "HepPDT/ParticleID.hh"
```

Functions

- void testHadron (HepPDT::ParticleID &, int &)
- void testLepton (HepPDT::ParticleID &, int &)
- void testDiQuark (HepPDT::ParticleID &, int &)
- void testNucleus (HepPDT::ParticleID &, int &)
- void testUnknown (HepPDT::ParticleID &, int &)
- void testHasMethods (HepPDT::ParticleID &, int &)
- void testValid (HepPDT::ParticleID &)
- int main ()

9.63.1 Function Documentation

9.63.1.1 int main ()

Definition at line 23 of file testPID.cc.

References HepPDT::n, HepPDT::nj, HepPDT::nl, HepPDT::nq1, HepPDT::nq2, HepPDT::nq3, HepPDT::nr, testDiQuark(), testHadron(), testHasMethods(), testLepton(), testNucleus(), testUnknown(), and testValid().

9.63.1.2 void testDiQuark (HepPDT::ParticleID &, int &)

Examples:

testPID.cc.

Definition at line 119 of file testPID.cc.

References HepPDT::ParticleID::digit(), HepPDT::ParticleID::extraBits(), HepPDT::ParticleID::fundamentalID(), HepPDT::ParticleID::isDiQuark(), HepPDT::ParticleID::jSpin(), HepPDT::ParticleID::lSpin(), HepPDT::n, HepPDT::Quarks::nq1, HepPDT::Quarks::nq2, HepPDT::Quarks::nq3, HepPDT::nr, HepPDT::ParticleID::quarks(), and HepPDT::ParticleID::threeCharge().

Referenced by main().

9.63.1.3 void testHadron (HepPDT::ParticleID &, int &)

Examples:

testPID.cc.

Definition at line 67 of file testPID.cc.

References HepPDT::ParticleID::digit(), HepPDT::ParticleID::extraBits(), HepPDT::ParticleID::fundamentalID(), HepPDT::ParticleID::isBaryon(), HepPDT::ParticleID::isHadron(), HepPDT::ParticleID::isMeson(), HepPDT::ParticleID::isRhadron(), HepPDT::ParticleID::jSpin(), HepPDT::ParticleID::lSpin(), HepPDT::n, HepPDT::Quarks::nq1, HepPDT::Quarks::nq2, HepPDT::Quarks::nq3, HepPDT::nr, HepPDT::ParticleID::pid(), HepPDT::ParticleID::quarks(), and HepPDT::ParticleID::threeCharge().

Referenced by main().

9.63.1.4 void testHasMethods (HepPDT::ParticleID &, int &)

Examples:

testPID.cc.

Definition at line 214 of file testPID.cc.

References HepPDT::ParticleID::hasBottom(), HepPDT::ParticleID::hasCharm(), HepPDT::ParticleID::hasDown(), HepPDT::ParticleID::hasStrange(), HepPDT::ParticleID::hasTop(), HepPDT::ParticleID::hasUp(), HepPDT::ParticleID::PDname(), and HepPDT::ParticleID::pid().

Referenced by main().

9.63.1.5 void testLepton (HepPDT::ParticleID &, int &)

Examples:

testPID.cc.

Definition at line 103 of file testPID.cc.

References HepPDT::ParticleID::digit(), HepPDT::ParticleID::extraBits(), HepPDT::ParticleID::fundamentalID(), HepPDT::ParticleID::isLepton(), HepPDT::ParticleID::jSpin(), HepPDT::ParticleID::lSpin(), HepPDT::n, HepPDT::Quarks::nq1, HepPDT::Quarks::nq2, HepPDT::Quarks::nq3, HepPDT::nr, HepPDT::ParticleID::quarks(), and HepPDT::ParticleID::threeCharge().

Referenced by main().

9.63.1.6 void testNucleus (HepPDT::ParticleID &, int &)

Examples:

testPID.cc.

Definition at line 135 of file testPID.cc.

References HepPDT::ParticleID::A(), HepPDT::ParticleID::digit(), HepPDT::ParticleID::extraBits(), HepPDT::ParticleID::fundamentalID(), HepPDT::ParticleID::isNucleus(), HepPDT::ParticleID::jSpin(), HepPDT::ParticleID::lambda(), HepPDT::n10, HepPDT::ParticleID::threeCharge(), and HepPDT::ParticleID::Z().

Referenced by main().

9.63.1.7 void testUnknown (HepPDT::ParticleID &, int &)

Examples:

testPID.cc.

Definition at line 150 of file testPID.cc.

References HepPDT::ParticleID::abspid(), HepPDT::ParticleID::digit(), HepPDT::ParticleID::extraBits(), HepPDT::ParticleID::fundamentalID(), HepPDT::ParticleID::isDiQuark(), HepPDT::ParticleID::isDyon(), HepPDT::ParticleID::isHadron(), HepPDT::ParticleID::isLepton(), HepPDT::ParticleID::isNucleus(), HepPDT::ParticleID::isSUSY(), HepPDT::ParticleID::jSpin(), HepPDT::ParticleID::lSpin(), HepPDT::n, HepPDT::Quarks::nq1, HepPDT::Quarks::nq2, HepPDT::Quarks::nq3, HepPDT::nr, HepPDT::ParticleID::quarks(), and HepPDT::ParticleID::threeCharge().

Referenced by main().

9.63.1.8 void testValid (HepPDT::ParticleID &)

Examples:

testPID.cc.

Definition at line 198 of file testPID.cc.

References HepPDT::ParticleID::charge(), HepPDT::ParticleID::isValid(), HepPDT::ParticleID::jSpin(), HepPDT::ParticleID::lSpin(), HepPDT::Quarks::nq1, HepPDT::Quarks::nq2, HepPDT::Quarks::nq3, HepPDT::ParticleID::quarks(), HepPDT::spinitod(), HepPDT::ParticleID::sSpin(), and HepPDT::ParticleID::threeCharge().

Referenced by main().

9.64 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

9.65 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

9.65.1 Define Documentation

9.65.1.1 `#define IDMAX 49`

Definition at line 16 of file translateGeanttoPDT.cc.

Referenced by `HepPID::translateGeanttoPDT()`, and `HepPID::translatePDTtoGeant()`.

9.66 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

9.67 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

9.68 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

9.69 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

9.69.1 Define Documentation

9.69.1.1 `#define IDMAX 49`

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

9.70 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

9.71 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

9.72 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

9.73 Version.cc File Reference

```
#include "HepPID/Version.hh"
```

Namespaces

- namespace **HepPID**

Functions

- **std::string HepPID::versionName ()**
return HepPID (p. 37) version
- **void HepPID::version ()**
print HepPID (p. 37) version
- **void HepPID::writeVersion (std::ostream &os)**
write HepPID (p. 37) version to os

9.74 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

9.75 Version.hh File Reference

```
#include <string>
#include <iostream>
```

Namespaces

- namespace **HepPID**

Functions

- void **HepPID::version ()**
print HepPID (p. 37) version
- void **HepPID::writeVersion (std::ostream &os)**
write HepPID (p. 37) version to os
- std::string **HepPID::versionName ()**
return HepPID (p. 37) version

9.76 write.cc File Reference

```
#include <iomanip>
#include <sstream>
#include "HepPDT/defs.h"
#include "HepPDT/ParticleData.hh"
```

Namespaces

- namespace **HepPDT**

Chapter 10

HepPDT Example Documentation

10.1 examListHerwig.cc

```
1 // -----
2 // examListHerwig.cc
3 //
4 // -----
5
13
14 #include <fstream>
15 #include <iostream>
16 #include <cstdlib>
17
18 #include "HepPID/ParticleIDTranslations.hh"
19 #include "HepPID/ParticleName.hh"
20 #include "HepPID/Version.hh"
21
22 extern "C" {
23 // these functions are defined in examListHerwigInterface.F
24 void list_herwig_init_ ( int * nevt );
25 void list_herwig_end_ ( );
26 void get_list_size_ ( int * );
27 void get_herwig_name_( int * ihwg, int * id, char *name );
28 }
29
30 int main()
31 {
32     int nevt=20;
33     int i, j, iend, isize;
34     int hid, id;
35     char cname[10];
36     std::string hname;
37     std::string pn;
38     static char outfile[] = "examListHerwig.out";
39     std::string title = "HepPID listing of Herwig translations";
40
41     // initialize herwig
42     list_herwig_init_ ( & nevt );
43
44     // open the output stream
45     std::ofstream os( outfile );
46     if( !os ) {
47         std::cout << "error opening output file" << std::endl;
48         exit(1);
49     }
50     HepPID::writeVersion(os);
51
52     get_list_size_( & isize );
```

```
53     os << "          " << title << std::endl;
54     os << " number of Herwig particles: " << isize << std::endl;
55
56     for( i=1, iend=isize+1; i<iend; ++i ) {
57         // get info from herwig
58         for( j=0; j<10; ++j) { cname[j] = '\\0'; }
59         get_herwig_name_( & i, & hid, cname );
60         hname = std::string( cname );
61         id = HepPID::translateHerwigtoPDT( hid );
62         pn = HepPID::particleName( id );
63         os << "Herwig: ";
64         os.width(7);
65         os << i ;
66         os.width(12);
67         os << hid << " " << hname;
68         os << " HepPID: " ;
69         os.width(12);
70         os << id << " " << pn << std::endl;
71     }
72
73     list_herwig_end_();
74
75     return 0;
76 }
77
```

10.2 examListHerwigInterface.F

```

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-----

```

10.3 examListIsajet.cc

```

1 // -----
2 // examListIsajet.cc
3 //
4 // -----
5
13
14 #include <fstream>
15 #include <iostream>
16 #include <cstdlib>
17 #include <cstring>
18
19 #include "HepPID/ParticleIDTranslations.hh"
20 #include "HepPID/ParticleName.hh"
21 #include "HepPID/Version.hh"
22
23 extern "C" {
24     void list_isajet_init_ ( );
25     void flavor_( int *, int *, int *, int *, int *, int * );
26     void get_label_( int * id, char *name );
27 }
28
29 int main()
30 {
31     static char outfile[] = "examListIsajet.out";
32     std::string title = "HepPID listing of Isajet translations";
33
34     // initialize isajet
35     list_isajet_init_ ( );
36
37     // open the output stream
38     std::ofstream os( outfile );
39     if( !os ) {
40         std::cout << "error opening output file" << std::endl;
41         exit(1);
42     }
43     HepPID::writeVersion(os);
44
45     os << "          " << title << std::endl;
46
47     int i, j;
48     int id, aid, fl1, fl2, fl3, js, indx;
49     int pid;
50     char cname[10];
51     char acname[10];
52     std::string hname;
53     std::string pn;
54     for( i=1; i<100005; ++i ) {
55         // make sure names are empty
56         for( j=0; j<10; ++j ) { cname[j] = '\0'; }
57         for( j=0; j<10; ++j ) { acname[j] = '\0'; }
58         // get info from isajet
59         id = i;
60         aid = 0;
61         flavor_( &id, &fl1, &fl2, &fl3, &js, &indx );
62         // we need both a valid index and a valid label
63         // check the label only if there is a valid translation
64         if ( indx > 0 ) {
65             get_label_( &id, cname );
66             aid = -id;
67             get_label_( &aid, acname );
68         } else {
69             id = aid = 0;
70         }
71
72         // print particle
73         if( id != 0 ) {

```

```

74         pid = HepPID::translateIsajettoPDT( id );
75         hname = std::string( cname );
76         if ( pid != 0 ) {
77             pn = HepPID::particleName( pid );
78             os << "Isajet: ";
79             os.width(10);
80             os << id << " " << hname;
81             os << "   HepPID: " ;
82             os.width(12);
83             os << pid << " " << pn << std::endl;
84         } else if ( strncmp( cname, "ERR", 3 ) != 0 ) {
85             os << "Isajet: ";
86             os.width(10);
87             os << id << " with name \"" << hname;
88             os << "\" has no HepPID translation " << std::endl;
89         }
90     }
91     // print antiparticle
92     if( aid != 0 ) {
93         hname = std::string( acname );
94         pid = HepPID::translateIsajettoPDT( aid );
95         if ( pid != 0 ) {
96             pn = HepPID::particleName( pid );
97             os << "Isajet: ";
98             os.width(10);
99             os << aid << " " << hname;
100             os << "   HepPID: " ;
101             os.width(12);
102             os << pid << " " << pn << std::endl;
103         } else if ( strncmp( acname, "ERR", 3 ) != 0 ) {
104             os << "Isajet: ";
105             os.width(10);
106             os << aid << " with name \"" << hname;
107             os << "\" has no HepPID translation " << std::endl;
108         }
109     }
110 }
111
112 return 0;
113 }

```

10.4 examListIsajetInterface.F

```

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
```

10.5 examListPythia.cc

```

1 // -----
2 // examListPythia.cc
3 //
4 // -----
5
13
14 #include <fstream>
15 #include <string>
16 #include <cstdlib>
17
18 #include "HepPID/ParticleIDTranslations.hh"
19 #include "HepPID/ParticleIDMethods.hh"
20 #include "HepPID/ParticleName.hh"
21 #include "HepPID/Version.hh"
22
23 extern "C" {
24     void list_pythia_ ( );
25     void getkf_( int *, int * );
26     void getpyname_( int * , int *, char *name );
27 }
28
29 void writeLine( int & i, int & kf, int & id,
30                std::string & name, std::string & pn, std::ofstream & os );
31
32 int main()
33 {
34     int kf,akf,pid,apid,iok;
35     char cname[17],caname[17];
36     std::string name, aname, pn, apn;
37     const char outfile[] = "examListPythia.out";
38     std::string title = "HepPID listing of Pythia translations";
39     // open the output file
40     std::ofstream os( outfile );
41     if( !os ) {
42         std::cerr << "cannot open " << outfile << std::endl;
43         exit(-1);
44     }
45     HepPID::writeVersion(os);
46     // get Pythia listing using the fortran pylist function
47     // write the output of pylist to examListPythia.lpt
48     list_pythia_();
49
50     os << "          " << title << std::endl;
51
52     for(int i=1; i<501; ++i) {
53         getkf_( &i, &kf);
54         if( kf != 0 ) {
55             getpyname_(&kf, &iok, cname);
56             if( iok == 1 ) {
57                 cname[16]='\0';
58                 name = std::string( cname );
59                 pid = HepPID::translatePythiatoPDT( kf );
60                 pn = HepPID::particleName( pid );
61                 writeLine( i, kf, pid, name, pn, os );
62                 if( HepPID::isValid( -pid ) ) {
63                     akf=-kf;
64                     getpyname_(&akf, &iok, caname);
65                     if( iok == 1 ) {
66                         apid = HepPID::translatePythiatoPDT( akf );
67                         apn = HepPID::particleName( apid );
68                         caname[16]='\0';
69                         aname = std::string( caname );
70                         writeLine( i, akf, apid, aname, apn, os );
71                     }
72                 }
73             }
74         }
75     }
76 }

```



```
74     }
75 }
76 return 0;
77 }
78
79 void writeLine( int & i, int & kf, int & id,
80               std::string & name, std::string & pn, std::ofstream & os )
81 {
82     os << "Pythia: ";
83     os.width(7);
84     os << i ;
85     os.width(12);
86     os << kf << " " << name;
87     os << "   HepPID: " ;
88     os.width(12);
89     os << id << " " << pn << std::endl;
90 }
```

10.6 examListPythiaInterface.F

```
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 getpyname(kf,idef,chap)
32 #include "pydat2.inc"
33     integer kf
34     character*16 chap
35     integer idef
36     idef = 1
37     CALL PYNAME(KF,CHAP)
38     if(CHAP.EQ.' ') idef=0
39     return
40     END
41
42
```

10.7 examMyPDT.cc

```

1 // -----
2 // examMyPDT.cc
3 //
4
5
6
7
8
9
10 #include "HepPDT/defs.h"
11 #include <fstream>
12
13 #include <cstdlib>
14 #include <string>
15
16 #include "HepPDT/TableBuilder.hh"
17 #include "HepPDT/ParticleDataTable.hh"
18 #include "HepPDT/TempParticleData.hh"
19
20 void addData( HepPDT::TableBuilder& tb, std::string const & name, int const id,
21              double const mass, double const charge, double const width,
22              double const tspin );
23
24 int main()
25 {
26     const char outfile[] = "examMyPDT.out";
27     // construct empty PDT
28     HepPDT::ParticleDataTable datacol;
29     {
30         // Construct table builder
31         HepPDT::TableBuilder tb(datacol);
32         // create my own particles here
33         addData( tb, "p+", 2212, 0.938, +1.0, -1, .5 );
34         addData( tb, "d", 1, 0., -2./3, -1, .5 );
35         addData( tb, "u~", -2, 0., -1./3, -1, .5 );
36         addData( tb, "W-", -24, 80.396, -1.0, 2.06, 1.0 );
37         addData( tb, "gamma", 22, 0., 0., -1, 1.0 );
38         addData( tb, "badgamma", 122, 0., 0., -1, 1.0 );
39         tb.removeParticle( 122 );
40     } // the tb destructor fills datacol
41     std::ofstream wpdfile( outfile );
42     if( !wpdfile ) {
43         std::cerr << "cannot open " << outfile << std::endl;
44         exit(-1);
45     }
46     datacol.writeParticleData(wpdfile);
47     // access a particle
48     // you get a null pointer if you request an undefined particle
49     HepPDT::ParticleData * pd = datacol.particle( HepPDT::ParticleID(22) );
50     if( pd ) {
51         std::cout << "particle " << pd->name() << " is defined" << std::endl;
52     } else {
53         std::cout << "ERROR: particle is not in particle data table" << std::endl;
54     }
55     pd = datacol[ HepPDT::ParticleID(-24) ];
56     // we expect this next line to produce an error
57     std::cout << "the error is expected" << std::endl;
58     if( datacol[ HepPDT::ParticleID(111) ] ) {
59         std::cout << "particle " << datacol[ HepPDT::ParticleID(111) ]->name() << " is defined" << std::endl;
60     } else {
61         std::cout << "ERROR: particle " << HepPDT::ParticleID(111).pid()
62                 << " is not in particle data table" << std::endl;
63     }
64
65     return 0;
66 }
67
68 void addData( HepPDT::TableBuilder& tb, std::string const & name, int const id,
69              double const mass, double const charge, double const width,
70              double const tspin )

```

```
71 {  
72     HepPDT::TempParticleData& tpd = tb.getParticleData( HepPDT::ParticleID( id ) );  
73     tpd.tempParticleName = name;  
74     tpd.tempCharge = charge;  
75     tpd.tempMass = HepPDT::Measurement( mass, 0. );  
76     tpd.tempSpin = HepPDT::SpinState( tspin, 0., 0. );  
77     tpd.tempWidth = HepPDT::Measurement( width, 0. );  
78     tb.addParticle( tpd );  
79 }
```

10.8 listEvtGenNames.cc.in

```

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@/examples/data/evt.pdl";
18     const char outfile[] = "listEvtGenNames.out";
19     // open input files
20     std::ifstream pdfile1( infile1 );
21     if( !pdfile1 ) {
22         std::cerr << "cannot open " << infile1 << std::endl;
23         exit(-1);
24     }
25     // construct PDT
26     HepPDT::ParticleDataTable datacol( "EvtGen Table" );
27     {
28         // Construct table builder
29         HepPDT::TableBuilder tb(datacol);
30         // read the input - put as many here as you want
31         if( !addEvtGenParticles( pdfile1, tb ) ) { std::cout << "error reading EvtGen pdt file " << std::endl; }
32     } // the tb destructor fills datacol
33     // open output file
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
42     return 0;
43 }

```

10.9 listEvtGenTranslation.cc

```
1 // -----
2 // listEvtGenTranslation.cc
3 // Author: Lynn Garren
4 //
5 // Usage: listEvtGenTranslation
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11 #include <cstdlib>    // for exit
12
13 #include "HepPID/ParticleIDTranslations.hh"
14
15 int main()
16 {
17     const char outfile[] = "listEvtGenTranslation.out";
18     // open the output file
19     std::ofstream wpdfile( outfile );
20     if( !wpdfile ) {
21         std::cerr << "cannot open " << outfile << std::endl;
22         exit(-1);
23     }
24     // write the particle names
25     HepPID::writeEvtGenTranslation( wpdfile );
26 }
```

10.10 listHerwigTranslation.cc

```
1 // -----
2 // listHerwigTranslation.cc
3 // Author: Lynn Garren
4 //
5 // Usage: listHerwigTranslation
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11 #include <cstdlib>      // for exit
12
13 #include "HepPID/ParticleIDTranslations.hh"
14
15 int main()
16 {
17     const char outfile[] = "listHerwigTranslation.out";
18     // open the output file
19     std::ofstream wpdfile( outfile );
20     if( !wpdfile ) {
21         std::cerr << "cannot open " << outfile << std::endl;
22         exit(-1);
23     }
24     // write the particle names
25     HepPID::writeHerwigTranslation( wpdfile );
26 }
```

10.11 listIsajetTranslation.cc

```
1 // -----
2 // listIsajetTranslation.cc
3 // Author: Lynn Garren
4 //
5 // Usage: listIsajetTranslation
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11 #include <cstdlib>      // for exit
12
13 #include "HepPID/ParticleIDTranslations.hh"
14
15 int main()
16 {
17     const char outfile[] = "listIsajetTranslation.out";
18     // open the output file
19     std::ofstream wpdfile( outfile );
20     if( !wpdfile ) {
21         std::cerr << "cannot open " << outfile << std::endl;
22         exit(-1);
23     }
24     // write the particle names
25     HepPID::writeIsajetTranslation( wpdfile );
26 }
```


10.12 listParticleNames.cc

```
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 #include <cstdlib>    // for exit
14
15 #include "HepPID/ParticleName.hh"
16
17 int main()
18 {
19     const char outfile[] = "listParticleNames.out";
20     // open the output file
21     std::ofstream wpdfile( outfile );
22     if( !wpdfile ) {
23         std::cerr << "cannot open " << outfile << std::endl;
24         exit(-1);
25     }
26     // write the particle names
27     HepPID::listParticleNames( wpdfile );
28 }
```

10.13 listPDGNames.cc.in

```

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 }

```

10.14 listPDGTranslation.cc

```
1 // -----
2 // listPDGTranslation.cc
3 // Author: Lynn Garren
4 //
5 // Usage: listPDGTranslation
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11 #include <cstdlib>      // for exit
12
13 #include "HepPID/ParticleIDTranslations.hh"
14
15 int main()
16 {
17     const char outfile[] = "listPDGTranslation.out";
18     // open the output file
19     std::ofstream wpdfile( outfile );
20     if( !wpdfile ) {
21         std::cerr << "cannot open " << outfile << std::endl;
22         exit(-1);
23     }
24     // write the particle names
25     HepPID::writePDGTranslation( wpdfile );
26 }
```

10.15 listPythiaNames.cc.in

```
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 }
```

10.16 listPythiaTranslation.cc

```
1 // -----
2 // listPythiaTranslation.cc
3 // Author: Lynn Garren
4 //
5 // Usage: listPythiaTranslation
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11 #include <cstdlib>      // for exit
12
13 #include "HepPID/ParticleIDTranslations.hh"
14
15 int main()
16 {
17     const char outfile[] = "listPythiaTranslation.out";
18     // open the output file
19     std::ofstream wpdfile( outfile );
20     if( !wpdfile ) {
21         std::cerr << "cannot open " << outfile << std::endl;
22         exit(-1);
23     }
24     // write the particle names
25     HepPID::writePythiaTranslation( wpdfile );
26 }
```

10.17 listQQTranslation.cc

```
1 // -----
2 // listQQTranslation.cc
3 // Author: Lynn Garren
4 //
5 // Usage: listQQTranslation
6 //
7 // -----
8
9 #include <fstream>
10 #include <iostream>
11 #include <cstdlib>      // for exit
12
13 #include "HepPID/ParticleIDTranslations.hh"
14
15 int main()
16 {
17     const char outfile[] = "listQQTranslation.out";
18     // open the output file
19     std::ofstream wpdfile( outfile );
20     if( !wpdfile ) {
21         std::cerr << "cannot open " << outfile << std::endl;
22         exit(-1);
23     }
24     // write the particle names
25     HepPID::writeQQTranslation( wpdfile );
26 }
```

10.18 testHepPDT.cc

```

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 #include <iomanip>
14
15 #include "HepPDT/defs.h"
16 #include "HepPDT/TableBuilder.hh"
17 #include "HepPDT/ParticleDataTable.hh"
18 #include "HepPDT/HeavyIonUnknownID.hh"
19 // local include
20 #include "TestNuclearFragment.hh"
21
22 void pdtSimpleTest( char[300], std::ofstream & );
23 void pdtFragmentTest( char[300], std::ofstream & );
24 void duplicateFragmentTest( char[300], std::ofstream & );
25 void testPDMMethods( HepPDT::ParticleDataTable&, std::ofstream & );
26
27 int main()
28 {
29     char pdgfile[300] = "";
30     const char outfile[] = "testHepPDT.out";
31     std::cin >> pdgfile;
32     // open output file
33     std::ofstream wpdfile( outfile );
34     if( !wpdfile ) {
35         std::cerr << "cannot open " << outfile << std::endl;
36         exit(-1);
37     }
38
39     // construct a default PDT
40     pdtSimpleTest( pdgfile, wpdfile );
41     // now test the nuclear fragment option
42     pdtFragmentTest( pdgfile, wpdfile );
43     // check how we deal with duplicate fragments
44     duplicateFragmentTest( pdgfile, wpdfile );
45
46     return 0;
47 }
48
49 void pdtSimpleTest( char pdgfile[300], std::ofstream & wpdfile )
50 {
51     // open input file
52     std::ifstream pdfile( pdgfile );
53     if( !pdfile ) {
54         std::cerr << "cannot open " << pdgfile << std::endl;
55         exit(-1);
56     }
57     // construct empty PDT
58     HepPDT::ParticleDataTable datacol( "2006 PDG Table" );
59     {
60         // Construct table builder
61         HepPDT::TableBuilder tb(datacol);
62         // read the input - put as many here as you want
63         if( !HepPDT::addPDGParticles( pdfile, tb ) ) {
64             std::cout << "error reading PDG file " << std::endl;
65         }

```

```

66     } // the tb destructor fills datacol
67     // done with pdfile, so close it
68     pdfile.close();
69
70     const char outfile1[] = "testHepPDTtable.out";
71     // open output file
72     std::ofstream wpdt1( outfile1 );
73     if( !wpdt1 ) {
74         std::cerr << "cannot open " << outfile1 << std::endl;
75         exit(-1);
76     }
77     datacol.writeParticleData(wpdt1);
78
79     wpdfile << std::endl;
80
81     // output some pion information
82     HepPDT::ParticleData * pd;
83     pd=datacol.particle(HepPDT::ParticleID(111));
84     // test the ResonanceStructure cutoff methods here
85     if(pd) {
86         pd->write(wpdfile);
87         wpdfile << "Resonance info for 111 "
88             << pd->totalWidth().value() << " "
89             << pd->totalWidth().sigma() << " "
90             << pd->lowerCutoff() << " "
91             << pd->upperCutoff() << std::endl;
92     }
93     // -111 is an illegal particle, no info will be written
94     pd=datacol.particle(HepPDT::ParticleID(-111));
95     if(pd) pd->write(wpdfile);
96     pd=datacol.particle(HepPDT::ParticleID(211));
97     if(pd) pd->write(wpdfile);
98     // string lookup
99     pd=datacol.particle(std::string("pi0"));
100     if(pd) pd->write(wpdfile);
101
102     // particle info
103     datacol.writeParticleInfo(wpdfile);
104
105     testPDMMethods( datacol, wpdfile );
106 }
107
108 void pdtFragmentTest( char pdgfile[300], std::ofstream & wpdfile )
109 {
110     wpdfile << std::endl;
111     wpdfile << " Begin test of HeavyIonUnknownID " << std::endl;
112     // reopen input file
113     std::ifstream pdfile2( pdgfile );
114     if( !pdfile2 ) {
115         std::cerr << "cannot open " << pdgfile << std::endl;
116         exit(-1);
117     }
118     // construct another PDT instance that knows how to deal with unknown heavy ions
119     // NOTE: normally you would construct a single ParticleDataTable with this option
120     HepPDT::ParticleDataTable pdt2( "Handle Heavy Ions",
121                                     new HepPDT::HeavyIonUnknownID );
122     {
123         // Construct table builder
124         HepPDT::TableBuilder tb2(pdt2);
125         // read the input - put as many here as you want
126         if( !HepPDT::addPDGParticles( pdfile2, tb2 ) ) {
127             std::cout << "error reading PDG file " << std::endl;
128         }
129     }
130     // done with pdfile, so close it
131     pdfile2.close();
132

```



```

133     // try a heavy ion
134     HepPDT::ParticleData * pd=pdt2.particle(HepPDT::ParticleID(1000020040));
135     wpdfiler << " Printing information for unknown nuclear fragment "
136         << std::endl;
137     if(pd) pd->write(wpdfiler);
138 }
139
140 void duplicateFragmentTest( char pdgfile[300], std::ofstream & wpdfiler )
141 {
142     wpdfiler << std::endl;
143     wpdfiler << " Begin test of duplicate nuclear fragments " << std::endl;
144     // reopen input file
145     std::ifstream pdfile2( pdgfile );
146     if( !pdfile2 ) {
147         std::cerr << "cannot open " << pdgfile << std::endl;
148         exit(-1);
149     }
150     // this test checks to see if we have actually added a fragment to the table
151     HepPDT::ParticleDataTable pdt( "Duplicate Nuclear Fragments",
152                                     new HepPDT::TestNuclearFragment );
153     {
154         // Construct table builder
155         HepPDT::TableBuilder tb2(pdt);
156         // read the input - put as many here as you want
157         if( !HepPDT::addPDGParticles( pdfile2, tb2 ) ) {
158             std::cout << "error reading PDG file " << std::endl;
159         }
160     }
161     // done with pdfile, so close it
162     pdfile2.close();
163
164     // try a heavy ion
165     wpdfiler << " Printing information for unknown nuclear fragments "
166         << std::endl;
167     HepPDT::ParticleData * pd=pdt.particle(HepPDT::ParticleID(1000020040));
168     if(pd) pd->write(wpdfiler);
169     pd=pdt.particle(HepPDT::ParticleID(1000020040));
170     if(pd) pd->write(wpdfiler);
171     pd=pdt.particle(HepPDT::ParticleID(1000010040));
172     if(pd) pd->write(wpdfiler);
173     pd=pdt.particle(HepPDT::ParticleID(1000020040));
174     if(pd) pd->write(wpdfiler);
175     // what is the state of the table?
176     const char outfile2[] = "testHepPDTfragment.out";
177     std::ofstream wpdt( outfile2 );
178     if( !wpdt ) {
179         std::cerr << "cannot open " << outfile2 << std::endl;
180         exit(-1);
181     }
182     pdt.writeParticleData(wpdt);
183
184     // check isStable
185     const char outfile3[] = "testHepPDTstatus.out";
186     std::ofstream wpdt3( outfile3 );
187     if( !wpdt3 ) {
188         std::cerr << "cannot open " << outfile3 << std::endl;
189         exit(-1);
190     }
191     pdt.writeParticleStatus(wpdt3);
192 }
193
194 void testPDMethods( HepPDT::ParticleDataTable& datacol, std::ofstream & wpdfiler )
195 {
196     wpdfiler << std::endl;
197     wpdfiler << "Begin test of ParticleData methods " << std::endl;
198     HepPDT::ParticleData * pd;
199     int id[27] = { 5, 24, 15, 213, 3214, 10213, 9050225, 541, 129050225,

```

```
200          2000025, 3101, 3301, 2212, 1000020040, 1000060120, 555,
201          5000040, 5100005, 24, 5100024, 5100025, 9221132,
202          4111370, -4120240, 4110050, 10013730, 1000612 };
203  int it;
204  for( it=0; it < 27; it++ ) {
205      pd=dataacol.particle(HepPDT::ParticleID(id[it]));
206      if(pd) {
207          if( pd->hasUp() ) {
208              wpdfiler << "Particle " << pd->name() << " " << pd->pid()
209                  << " has an up quark" << std::endl;
210          }
211          if( pd->hasDown() ) {
212              wpdfiler << "Particle " << pd->name() << " " << pd->pid()
213                  << " has a down quark" << std::endl;
214          }
215          if( pd->hasStrange() ) {
216              wpdfiler << "Particle " << pd->name() << " " << pd->pid()
217                  << " has a strange quark" << std::endl;
218          }
219          if( pd->hasCharm() ) {
220              wpdfiler << "Particle " << pd->name() << " " << pd->pid()
221                  << " has a charmed quark" << std::endl;
222          }
223          if( pd->hasBottom() ) {
224              wpdfiler << "Particle " << pd->name() << " " << pd->pid()
225                  << " has a bottom quark" << std::endl;
226          }
227          if( pd->hasTop() ) {
228              wpdfiler << "Particle " << pd->name() << " " << pd->pid()
229                  << " has a top quark" << std::endl;
230          }
231      }
232  }
233 }
```

10.19 testPID.cc

```

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 void testHadron( HepPDT::ParticleID&, int& );
16 void testLepton( HepPDT::ParticleID&, int& );
17 void testDiQuark( HepPDT::ParticleID&, int& );
18 void testNucleus( HepPDT::ParticleID&, int& );
19 void testUnknown( HepPDT::ParticleID&, int& );
20 void testHasMethods( HepPDT::ParticleID&, int& );
21 void testValid( HepPDT::ParticleID& );
22
23 int main()
24 {
25     int id[40] = { 5, 25, 15, 213, -3214, 10213, 9050225, -200543, 129050225,
26                   2000025, 3101, 3301, -2212, 1000020040, -1000060120, 555,
27                   5000040, 5100005, 24, 5100024, 5100025, 9221132,
28                   4111370, -4120240, 4110050, 10013730,
29                   1000993, 1000612, 1000622, 1000632, 1006213, 1000652,
30                   1009113, 1009213, 1009323,
31                   1093114, 1009333, 1006313, 1092214, 1006223 };
32
33     int it;
34     int nr, nx;
35     int chg, sid, extra;
36     int js, ls;
37     for( it=0; it < 40; it++ ) {
38         HepPDT::ParticleID pid( id[it] );
39         nx = pid.digit( HepPDT::n );
40         nr = pid.digit( HepPDT::nr );
41         extra = pid.extraBits();
42         std::cout << std::endl;
43         std::cout << std::setw(18) << id[it] << ": " << nx << " " << nr
44                   << " " << pid.digit( HepPDT::nl )
45                   << " " << pid.digit( HepPDT::nq1 ) << " "
46                   << pid.digit( HepPDT::nq2 ) << " " << pid.digit( HepPDT::nq3 )
47                   << " " << pid.digit( HepPDT::nj )
48                   << " extra bits " << extra << std::endl;
49         js = pid.jSpin();
50         ls = pid.lSpin();
51         sid = pid.fundamentalID();
52         chg = pid.threeCharge();
53         if( !pid.isValid() ) {
54             std::cout << "**** Invalid PID: " << pid.pid()
55                       << " ****" << std::endl;
56         } else {
57             testHadron( pid, id[it] );
58             testLepton( pid, id[it] );
59             testDiQuark( pid, id[it] );
60             testNucleus( pid, id[it] );
61             testUnknown( pid, id[it] );
62         }
63         testValid( pid );
64         testHasMethods( pid, id[it] );
65     }
66 }

```

```

66
67 void testHadron( HepPDT::ParticleID& pid, int& idn )
68 {
69     int nx = pid.digit(HepPDT::n);
70     int nr = pid.digit(HepPDT::nr);
71     int extra = pid.extraBits();
72     int js = pid.jSpin();
73     int ls = pid.lSpin();
74     int sid = pid.fundamentalID();
75     int chg = pid.threeCharge();
76     HepPDT::Quarks cqks = pid.quarks( );
77     if( pid.isHadron() ) {
78         if( pid.isMeson() ) {
79             std::cout << "meson  " << std::setw(10) << idn << ": " << nx
80                 << " " << nr << " " << ls << " "
81                 << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
82                 << " " << js << " " << sid << " " << chg
83                 << " extra bits " << extra << std::endl;
84         } else if( pid.isRhadron() ) {
85             std::cout << "Rhadron " << std::setw(10) << idn << ": " << nx
86                 << " " << nr << " " << ls << " "
87                 << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
88                 << " " << js << " " << sid << " " << chg
89                 << " extra bits " << extra << std::endl;
90         } else if( pid.isBaryon() ) {
91             std::cout << "baryon  " << std::setw(10) << idn << ": " << nx
92                 << " " << nr << " " << ls << " "
93                 << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
94                 << " " << js << " " << sid << " " << chg
95                 << " extra bits " << extra << std::endl;
96         } else {
97             std::cout << "**** undefined hadron: " << pid.pid()
98                 << " ****" << std::endl;
99         }
100     }
101 }
102
103 void testLepton( HepPDT::ParticleID& pid, int& idn )
104 {
105     HepPDT::Quarks cqks = pid.quarks( );
106     if( pid.isLepton() ) {
107         std::cout << "lepton  " << std::setw(10) << idn
108             << ": " << pid.digit(HepPDT::n)
109             << " " << pid.digit(HepPDT::nr)
110             << " " << pid.lSpin() << " "
111             << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
112             << " " << pid.jSpin()
113             << " " << pid.fundamentalID()
114             << " " << pid.threeCharge()
115             << " extra bits " << pid.extraBits() << std::endl;
116     }
117 }
118
119 void testDiQuark( HepPDT::ParticleID& pid, int& idn )
120 {
121     HepPDT::Quarks cqks = pid.quarks( );
122     if( pid.isDiQuark() ) {
123         std::cout << "diquark " << std::setw(10) << idn
124             << ": " << pid.digit(HepPDT::n)
125             << " " << pid.digit(HepPDT::nr)
126             << " " << pid.lSpin() << " "
127             << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
128             << " " << pid.jSpin()
129             << " " << pid.fundamentalID()
130             << " " << pid.threeCharge()
131             << " extra bits " << pid.extraBits() << std::endl;
132     }

```

```

133 }
134
135 void testNucleus( HepPDT::ParticleID& pid, int& idn )
136 {
137     if( pid.isNucleus() ) {
138         std::cout << "ion      " << std::setw(11) << idn
139             << ": " << pid.digit( HepPDT::n10 )
140             << " " << std::setw(3) << pid.A()
141             << " " << std::setw(3) << pid.Z()
142             << " " << std::setw(3) << pid.lambda()
143             << " " << pid.jSpin()
144             << " " << pid.fundamentalID()
145             << " " << pid.threeCharge()
146             << " extra bits " << pid.extraBits() << std::endl;
147     }
148 }
149
150 void testUnknown( HepPDT::ParticleID& pid, int& idn )
151 {
152     HepPDT::Quarks cqks = pid.quarks( );
153     if( pid.isHadron() || pid.isLepton() ||
154         pid.isDiQuark() || pid.isNucleus() ) {
155     } else if( pid.isSUSY() ) {
156         std::cout << "SUSY      " << std::setw(10) << idn
157             << ": " << pid.digit( HepPDT::n )
158             << " " << pid.digit( HepPDT::nr )
159             << " " << pid.lSpin() << " "
160             << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
161             << " " << pid.jSpin()
162             << " " << pid.fundamentalID()
163             << " " << pid.threeCharge()
164             << " extra bits " << pid.extraBits() << std::endl;
165     } else if( pid.isDyon() ) {
166         std::cout << "Dyon      " << std::setw(10) << idn
167             << ": " << pid.digit( HepPDT::n )
168             << " " << pid.digit( HepPDT::nr )
169             << " " << pid.lSpin() << " "
170             << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
171             << " " << pid.jSpin()
172             << " " << pid.fundamentalID()
173             << " " << pid.threeCharge()
174             << " extra bits " << pid.extraBits() << std::endl;
175     } else if( pid.abspid() < 100 ) {
176         std::cout << "quark/lepton/boson " << std::setw(2) << idn
177             << ": " << pid.digit( HepPDT::n )
178             << " " << pid.digit( HepPDT::nr )
179             << " " << pid.lSpin() << " "
180             << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
181             << " " << pid.jSpin()
182             << " " << pid.fundamentalID()
183             << " " << pid.threeCharge()
184             << " extra bits " << pid.extraBits() << std::endl;
185     } else {
186         std::cout << "unknown " << std::setw(10) << idn
187             << ": " << pid.digit( HepPDT::n )
188             << " " << pid.digit( HepPDT::nr )
189             << " " << pid.lSpin() << " "
190             << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
191             << " " << pid.jSpin()
192             << " " << pid.fundamentalID()
193             << " " << pid.threeCharge()
194             << " extra bits " << pid.extraBits() << std::endl;
195     }
196 }
197
198 void testValid( HepPDT::ParticleID& pid )
199 {

```

```

200         if( pid.isValid() ) {
201             int js = pid.jSpin();
202             int chg = pid.threeCharge();
203             std::cout << "total spin: " << js << " " << HepPDT::spinitod(js) ;
204             std::cout << " orbital angular momentum: " << pid.lSpin() ;
205             std::cout << " spin: " << pid.sSpin() ;
206             std::cout << " charge: " << chg
207                 << " " << pid.charge() << std::endl;
208             HepPDT::Quarks qlist = pid.quarks( );
209             std::cout << "quarks: " << qlist.nq1
210                 << " " << qlist.nq2 << " " << qlist.nq3 << std::endl;
211         }
212     }
213
214 void testHasMethods( HepPDT::ParticleID& pid, int& idn )
215 {
216     if( pid.hasUp() ) {
217         std::cout << "Particle " << pid.PDTname() << " " << pid.pid()
218             << " has an up quark" << std::endl;
219     }
220     if( pid.hasDown() ) {
221         std::cout << "Particle " << pid.PDTname() << " " << pid.pid()
222             << " has a down quark" << std::endl;
223     }
224     if( pid.hasStrange() ) {
225         std::cout << "Particle " << pid.PDTname() << " " << pid.pid()
226             << " has a strange quark" << std::endl;
227     }
228     if( pid.hasCharm() ) {
229         std::cout << "Particle " << pid.PDTname() << " " << pid.pid()
230             << " has a charmed quark" << std::endl;
231     }
232     if( pid.hasBottom() ) {
233         std::cout << "Particle " << pid.PDTname() << " " << pid.pid()
234             << " has a bottom quark" << std::endl;
235     }
236     if( pid.hasTop() ) {
237         std::cout << "Particle " << pid.PDTname() << " " << pid.pid()
238             << " has a top quark" << std::endl;
239     }
240 }
241

```

10.20 testReadEvtGen.cc.in

```

1 // $Id: testReadEvtGen.cc.in,v 1.5 2009/01/09 20:28:34 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/examples/data/evt.pdl";
18     const char infile2[] = "@top_srcdir/examples/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 }

```

10.21 testReadIsajet.cc.in

```

1 // -----
2 // readIsajet.cc
3 // Author: Lynn Garren
4 //
5 // read the isajet particle table 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 int main()
19 {
20     char pdgfile[300] = "@top_srcdir@/examples/data/isaparticles.dat";
21     const char outfile[] = "testReadIsajet.out";
22     // construct empty PDT
23     HepPDT::ParticleDataTable datacol( "Isajet Table" );
24     {
25         // open input files
26         std::ifstream pdfile( pdgfile );
27         if( !pdfile ) {
28             std::cerr << "cannot open " << pdgfile << std::endl;
29             exit(-1);
30         }
31         // Construct table builder
32         HepPDT::TableBuilder tb(datacol);
33         // read the input - put as many here as you want
34         if( !HepPDT::addIsajetParticles( pdfile, tb ) ) {
35             std::cout << "error reading " << pdgfile << std::endl;
36         }
37     } // the tb destructor fills datacol
38     std::ofstream wpdfile( outfile );
39     if( !wpdfile ) {
40         std::cerr << "cannot open " << outfile << std::endl;
41         exit(-1);
42     }
43     datacol.writeParticleData(wpdfile);
44     wpdfile << std::endl;
45
46     // particle info
47     datacol.writeParticleInfo(wpdfile);
48
49     return 0;
50 }

```


10.22 testReadParticleTable.cc.in

```

1 // $Id: testReadParticleTable.cc.in,v 1.6 2009/11/25 02:20:37 garren Exp $
2 // -----
3 // testReadParticleTable.cc
4 //
5 // read particle.tbl 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[] = "@top_srcdir@/data/particle.tbl";
18     const char infile2[] = "@top_srcdir@/tests/HepPDT/extras.tbl";
19     const char outfile[] = "testReadParticleTable.out";
20     // open input files
21     std::ifstream pdfile( infile );
22     if( !pdfile ) {
23         std::cerr << "cannot open " << infile << 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 empty PDT
32     HepPDT::ParticleDataTable datacol( "Generic Particle Table" );
33     {
34         // Construct table builder
35         HepPDT::TableBuilder tb(datacol);
36         // read the input - put as many here as you want
37         // bool addParticleTable( std::istream&, TableBuilder&,
38         //                          bool validate = false );
39         // where: validate=true => verify that the ParticleID is valid
40         if( !addParticleTable( pdfile, tb, true ) ) {
41             std::cout << "error reading PDG pdt file " << std::endl;
42         }
43         if( !addParticleTable( pdfile2, tb, true ) ) {
44             std::cout << "error reading extra pdt file " << std::endl;
45         }
46     } // the tb destructor fills datacol
47     // open the output stream
48     std::ofstream wfile( outfile );
49     if( !wfile ) {
50         std::cerr << "cannot open " << outfile << std::endl;
51         exit(-1);
52     }
53     // write the data table
54     datacol.writeParticleData(wfile);
55     // try some heavy ions
56     wfile << std::endl;
57     wfile << std::endl;
58     HepPDT::ParticleData * pd;
59     pd=datacol.particle(HepPDT::ParticleID(1000020040));
60     if(pd) pd->write(wfile);
61     pd=datacol.particle(HepPDT::ParticleID(1000050110));
62     if(pd) pd->write(wfile);
63
64     // check isStable
65     const char outfile3[] = "testReadParticleTableStatus.out";

```

```
66     std::ofstream wpdt3( outfile3 );
67     if( !wpdt3 ) {
68         std::cerr << "cannot open " << outfile3 << std::endl;
69         exit(-1);
70     }
71     datacol.writeParticleStatus(wpdt3);
72
73     return 0;
74 }
```

10.23 testReadQQ.cc.in

```
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 }
```

Index

- /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/examples/ Directory Reference, 11
- /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/examples/HepPDT/ Directory Reference, 13
- /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/examples/HepPID/ Directory Reference, 19
- /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/include/ Directory Reference, 20
- /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/include/HepPDT/ Directory Reference, 14
- /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/include/HepPID/ Directory Reference, 16
- /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/src/ Directory Reference, 21
- /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/src/HepPDT/ Directory Reference, 15
- /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/src/HepPID/ Directory Reference, 17
- /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/tests/ Directory Reference, 22
- /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/tests/HepPDT/ Directory Reference, 12
- /home/cepa01/garren/lcg/heppdt/HepPDT-3.04.01/tests/HepPID/ Directory Reference, 18
- ~DefTable
 - HepPDT::DefTable, 69
- ~ParticleData
 - HepPDT::ParticleData, 77
- ~ParticleDataTable
 - HepPDT::ParticleDataTable, 88
- ~ParticleNameMap
 - HepPID::ParticleNameMap, 106
- ~ProcessUnknownID
 - HepPDT::ProcessUnknownID, 108
- ~ResonanceStructure
 - HepPDT::ResonanceStructure, 113
- ~TableBuilder
 - HepPDT::TableBuilder, 121
- A
 - HepPDT::ParticleID, 95
 - HepPID, 44
- abspid
 - HepPDT::ParticleID, 95
 - HepPID, 45
- addAlias
 - HepPDT::TableBuilder, 121
- addConstituent
 - HepPDT::ParticleData, 77
- addData
 - examMyPDT.cc, 154
- addDefinition
 - HepPDT::DefTable, 69
- addEvtGenParticles
 - HepPDT, 29
- addEvtGenParticles.cc, 135
- addHerwigParticles
 - HepPDT, 28
- addHerwigParticles.cc, 136
- addIsajetDecay
 - HepPDT, 28
- addIsajetParticles
 - HepPDT, 28
- addIsajetParticles.cc, 137
- addParticle
 - HepPDT::TableBuilder, 121
- addParticleTable
 - HepPDT, 29
- addParticleTable.cc, 138
- addPDGParticles
 - HepPDT, 27
- addPDGParticles.cc, 139
- addPythiaParticles
 - HepPDT, 28
- addPythiaParticles.cc, 140
- addQQParticles
 - HepPDT, 28
- addQQParticles.cc, 141
- aliasData

- HepPDT::TableBuilder, 121
- aliasSize
 - HepPDT::TableBuilder, 122
- antiparticle
 - HepPDT::TempParticleData, 129
- begin
 - HepPDT::DefTable, 69
 - HepPDT::ParticleDataTable, 88
 - HepPID::ParticleNameMap, 107
- beginLookupMap
 - HepPID::ParticleNameMap, 107
- beginNameMap
 - HepPDT::ParticleDataTable, 88
- calculateWidthFromLifetime
 - HepPDT, 29
- calculateWidthFromLifetime.cc, 142
- callProcessUnknownID
 - HepPDT::ProcessUnknownID, 108
- charge
 - HepPDT::ParticleData, 77
 - HepPDT::ParticleID, 96
 - HepPID, 50
- checkForSpecialParticle
 - HepPID, 57
- CheckPDGEntry
 - HepPDT::detail, 34
- color
 - HepPDT::ParticleData, 77
- const_iterator
 - HepPDT::DefTable, 69
 - HepPDT::ParticleDataTable, 87
- const_iteratorByName
 - HepPDT::ParticleDataTable, 87
- Constituent
 - HepPDT::Constituent, 66
- constituent
 - HepPDT::ParticleData, 78
- Constituent.cc, 143
- Constituent.hh, 144
- constituentParticle
 - HepPDT::ParticleData, 78
- convertTemporaryMap
 - HepPDT::ParticleDataTable, 88
- convertTemporaryMap.cc, 145
- convIsajettoPDT
 - HepPID, 59
- convPDTtoIsajet
 - HepPID, 59
- CPD
 - HepPDT::ParticleDataTable, 88
- definition
 - HepPDT::DefTable, 69
 - HepPDT::TableBuilder, 122
- definitions
 - HepPDT::TableBuilder, 122
- DefTable
 - HepPDT::DefTable, 69
- DefTable.cc, 146
- DefTable.hh, 147
- digit
 - HepPDT::ParticleID, 96
 - HepPID, 44
- duplicateFragmentTest
 - testHepPDT.cc, 204
- dyonName
 - HepPID, 56
- end
 - HepPDT::DefTable, 69, 70
 - HepPDT::ParticleDataTable, 89
 - HepPID::ParticleNameMap, 107
- endLookupMap
 - HepPID::ParticleNameMap, 107
- endNameMap
 - HepPDT::ParticleDataTable, 89
- EvtGenPDTMap
 - HepPID, 42
- EvtGenPDTMapInit
 - HepPID, 57
- examListHerwig.cc, 148
- examListHerwig.cc
 - get_herwig_name_, 148
 - get_list_size_, 148
 - list_herwig_end_, 148
 - list_herwig_init_, 148
 - main, 149
- examListIsajet.cc, 150
- examListIsajet.cc
 - flavor_, 150
 - get_label_, 150
 - list_isajet_init_, 150
 - main, 150
- examListPythia.cc, 152
- examListPythia.cc
 - getkf_, 152
 - getpyname_, 152
 - list_pythia_, 152
 - main, 152
 - writeLine, 153
- examMyPDT.cc, 154
- examMyPDT.cc
 - addData, 154
 - main, 154
- extraBits
 - HepPDT::ParticleID, 96

- HepPID, 45
- find
 - HepPID::ParticleNameMap, 107
- findQ
 - HepPID, 56
- findString
 - HepPID::ParticleNameMap, 107
- flavor_
 - examListIsajet.cc, 150
- fundamentalID
 - HepPDT::ParticleID, 97
 - HepPID, 45
- get_herwig_name_
 - examListHerwig.cc, 148
- get_label_
 - examListIsajet.cc, 150
- get_list_size_
 - examListHerwig.cc, 148
- getAntiParticle
 - HepPDT::TableBuilder, 122
- getEvtGenLineType
 - HepPDT, 30
- getEvtGenPDTMap
 - HepPID, 57
- getHerwigPDTMap
 - HepPID, 58
- getInverseQQbarMap
 - HepPID, 61, 62
- getIsajetID
 - HepPDT::detail, 35
- getIsajetID.cc, 155
- getIsajetPDTMap
 - HepPID, 59
- getkf_
 - examListPythia.cc, 152
- getParticleData
 - HepPDT::TableBuilder, 122
- getParticleID
 - HepPDT::detail, 35
- getParticleNameMap
 - HepPID, 55
- getPDGnames
 - HepPDT::detail, 34
- getPDGpid
 - HepPDT::detail, 33
- getPDGpid.cc, 156
- getPDGtoPDTMap
 - HepPID, 60
- getPDTEvtGenMap
 - HepPID, 57
- getPDTHerwigMap
 - HepPID, 58
- getPDTIsajetMap
 - HepPID, 59
- getPDTPythiaMap
 - HepPID, 60, 61
- getPDTQQMap
 - HepPID, 61, 62
- getPDTtoPDGMap
 - HepPID, 60
- getpyname_
 - examListPythia.cc, 152
- getPythiaDecay
 - HepPDT::detail, 35
- getPythiaid
 - HepPDT::detail, 34
- getPythiaid.cc, 157
- getPythiaPDTMap
 - HepPID, 60, 61
- getQQbarMap
 - HepPID, 61, 62
- getQQLineType
 - HepPDT, 31
- getQQPDTMap
 - HepPID, 61, 62
- hasAlias
 - HepPDT::TableBuilder, 123
- hasBottom
 - HepPDT::ParticleData, 78
 - HepPDT::ParticleID, 97
 - HepPID, 49
- hasCharm
 - HepPDT::ParticleData, 78
 - HepPDT::ParticleID, 97
 - HepPID, 48
- hasDefinition
 - HepPDT::DefTable, 70
 - HepPDT::TableBuilder, 123
- hasDown
 - HepPDT::ParticleData, 78
 - HepPDT::ParticleID, 97
 - HepPID, 48
- hasFundamentalAnti
 - HepPID, 45
- hasMethods.cc, 158
- hasParticleData
 - HepPDT::TableBuilder, 123
- hasStrange
 - HepPDT::ParticleData, 78
 - HepPDT::ParticleID, 98
 - HepPID, 48
- hasTop
 - HepPDT::ParticleData, 79
 - HepPDT::ParticleID, 98
 - HepPID, 49

- hasUp
 - HepPDT::ParticleData, 79
 - HepPDT::ParticleID, 98
 - HepPID, 48
- HeavyIonUnknownID
 - HepPDT::HeavyIonUnknownID, 71
- HeavyIonUnknownID.cc, 159
- HeavyIonUnknownID.hh, 160
- HepPDT, 23
 - n, 26
 - n10, 26
 - n8, 26
 - n9, 26
 - nj, 26
 - nl, 26
 - nq1, 26
 - nq2, 26
 - nq3, 26
 - nr, 26
- HepPDT
 - addEvtGenParticles, 29
 - addHerwigParticles, 28
 - addIsajetDecay, 28
 - addIsajetParticles, 28
 - addParticleTable, 29
 - addPDGParticles, 27
 - addPythiaParticles, 28
 - addQQParticles, 28
 - calculateWidthFromLifetime, 29
 - getEvtGenLineType, 30
 - getQQLineType, 31
 - location, 26
 - parseEvtGenAlias, 30
 - parseEvtGenAliasDecayLine, 31
 - parseEvtGenConj, 31
 - parseEvtGenDecayLine, 31
 - parseEvtGenDefinition, 31
 - parseEvtGenLine, 30
 - parseQQDecayLine, 31
 - parseQQParticle, 31
 - spindtoi, 27
 - spinitod, 27
 - stringtodouble, 27
 - swap, 26, 27, 29
 - TDDlist, 25
 - version, 30
 - versionName, 30
 - writeEvtGenStream, 27
 - writeHerwigStream, 27
 - writeIsajetStream, 27
 - writePDGStream, 26
 - writePythiaStream, 27
 - writeQQStream, 27
 - writeVersion, 30
- HepPDT::Constituent, 65
- HepPDT::Constituent
 - Constituent, 66
 - isBottom, 66
 - isCharm, 66
 - isDown, 66
 - isStrange, 66
 - isTop, 66
 - isUp, 67
 - multiplicity, 67
 - operator=, 67
 - pid, 67
 - swap, 67
- HepPDT::DefTable, 68
- HepPDT::DefTable
 - ~DefTable, 69
 - addDefinition, 69
 - begin, 69
 - const_iterator, 69
 - definition, 69
 - DefTable, 69
 - end, 69, 70
 - hasDefinition, 70
 - iterator, 69
 - size, 70
 - writeDefinitions, 70
- HepPDT::detail, 33
- HepPDT::detail
 - CheckPDGEntry, 34
 - getIsajetID, 35
 - getParticleID, 35
 - getPDGnames, 34
 - getPDGpid, 33
 - getPythiaDecay, 35
 - getPythiaid, 34
 - parseIsajetDecayLine, 35
 - parseIsajetLine, 35
 - parseParticleLine, 35
 - parsePDGline, 34
 - parsePythiaDecayLine, 34
 - parsePythiaLine, 34
- HepPDT::HeavyIonUnknownID, 71
- HepPDT::HeavyIonUnknownID
 - HeavyIonUnknownID, 71
 - processUnknownID, 71
- HepPDT::Measurement, 72
- HepPDT::Measurement
 - Measurement, 72
 - operator double, 72
 - operator<, 72
 - operator=, 72
 - operator==, 73
 - sigma, 73
 - swap, 73

- value, 73
- HepPDT::ParticleData, 74
- HepPDT::ParticleData
 - ~ParticleData, 77
 - addConstituent, 77
 - charge, 77
 - color, 77
 - constituent, 78
 - constituentParticle, 78
 - hasBottom, 78
 - hasCharm, 78
 - hasDown, 78
 - hasStrange, 78
 - hasTop, 79
 - hasUp, 79
 - ID, 79
 - isBaryon, 79
 - isDiQuark, 79
 - isDyon, 79
 - isHadron, 80
 - isLepton, 80
 - isMeson, 80
 - isNucleus, 80
 - isPentaquark, 80
 - isQBall, 80
 - isRhadron, 80
 - isStable, 81
 - isSUSY, 81
 - lifetime, 81
 - lowerCutoff, 81
 - mass, 81
 - name, 81
 - numConstituents, 82
 - operator<, 82
 - operator=, 82
 - operator==, 82
 - originalID, 82
 - ParticleData, 77
 - PDTname, 82
 - pid, 82
 - resonance, 83
 - setCharge, 83
 - setColor, 83
 - setLowerCutoff, 83
 - setMass, 83
 - setSpin, 83
 - setTotalWidth, 83
 - setTotalWidthFromLifetime, 83
 - setUpperCutoff, 84
 - source, 84
 - spin, 84
 - swap, 84
 - totalWidth, 84
 - upperCutoff, 84
 - write, 85
 - writeParticleInfo, 85
 - writeParticleTranslation, 85
- HepPDT::ParticleDataTable, 86
- HepPDT::ParticleDataTable
 - ~ParticleDataTable, 88
 - begin, 88
 - beginNameMap, 88
 - const_iterator, 87
 - const_iteratorByName, 87
 - convertTemporaryMap, 88
 - CPD, 88
 - end, 89
 - endNameMap, 89
 - operator[], 89
 - particle, 89, 90
 - ParticleDataTable, 88
 - PDTMap, 88
 - PDTNameMap, 88
 - size, 90
 - sizeNameMap, 90
 - tableName, 90
 - TempMap, 88
 - writeParticleData, 90
 - writeParticleInfo, 91
 - writeParticleStatus, 91
 - writeParticleTranslation, 91
- HepPDT::ParticleDataTableComparison, 92
- HepPDT::ParticleDataTableComparison
 - operator(), 92
 - ParticleDataTableComparison, 92
- HepPDT::ParticleID, 93
- HepPDT::ParticleID
 - A, 95
 - abspid, 95
 - charge, 96
 - digit, 96
 - extraBits, 96
 - fundamentalID, 97
 - hasBottom, 97
 - hasCharm, 97
 - hasDown, 97
 - hasStrange, 98
 - hasTop, 98
 - hasUp, 98
 - isBaryon, 98
 - isDiQuark, 99
 - isDyon, 99
 - isHadron, 99
 - isLepton, 100
 - isMeson, 100
 - isNucleus, 100
 - isPentaquark, 100
 - isQBall, 101

- isRhadron, 101
- isSUSY, 101
- isValid, 101
- jSpin, 102
- lambda, 102
- lSpin, 102
- operator<, 102
- operator=, 102
- operator==, 103
- ParticleID, 95
- PDTname, 103
- pid, 103
- quarks, 103
- sSpin, 104
- swap, 104
- threeCharge, 104
- Z, 104
- HepPDT::ProcessUnknownID, 108
- HepPDT::ProcessUnknownID
 - ~ProcessUnknownID, 108
 - callProcessUnknownID, 108
 - ProcessUnknownID, 108
- HepPDT::Quarks, 110
- HepPDT::Quarks
 - nq1, 110
 - nq2, 110
 - nq3, 111
 - Quarks, 110
- HepPDT::ResonanceStructure, 112
- HepPDT::ResonanceStructure
 - ~ResonanceStructure, 113
 - lifetime, 113
 - lowerCutoff, 113
 - mass, 113
 - operator=, 113
 - ResonanceStructure, 113
 - setLowerCutoff, 113
 - setMass, 114
 - setTotalWidth, 114
 - setTotalWidthFromLifetime, 114
 - setUpperCutoff, 114
 - swap, 114
 - totalWidth, 114
 - upperCutoff, 115
- HepPDT::SimpleProcessUnknownID, 116
- HepPDT::SimpleProcessUnknownID
 - processUnknownID, 116
 - SimpleProcessUnknownID, 116
- HepPDT::SpinState, 117
- HepPDT::SpinState
 - operator=, 118
 - operator==, 118
 - orbAngMom, 118
 - setOrbAngMom, 118
 - setSpin, 118
 - setTotalSpin, 118
 - spin, 118
 - SpinState, 117
 - swap, 118
 - totalSpin, 118
- HepPDT::TableBuilder, 120
- HepPDT::TableBuilder
 - ~TableBuilder, 121
 - addAlias, 121
 - addParticle, 121
 - aliasData, 121
 - aliasSize, 122
 - definition, 122
 - definitions, 122
 - getAntiParticle, 122
 - getParticleData, 122
 - hasAlias, 123
 - hasDefinition, 123
 - hasParticleData, 123
 - removeParticle, 123
 - size, 123
 - TableBuilder, 121
- HepPDT::TempAliasData, 124
- HepPDT::TempAliasData
 - tempAlias, 124
 - TempAliasData, 124
 - tempAliasDecayList, 124
 - tempAliasedParticle, 125
 - tempChargeConj, 125
- HepPDT::TempConstituent, 126
- HepPDT::TempConstituent
 - TempConstituent, 126
 - tempConstituentPID, 126
 - tempMultiplicity, 126
- HepPDT::TempDecayData, 127
- HepPDT::TempDecayData
 - tempBranchingFraction, 127
 - tempDaughterList, 127
 - tempDecayName, 127
 - tempDecayParameters, 127
- HepPDT::TempParticleData, 128
- HepPDT::TempParticleData
 - antiparticle, 129
 - operator=, 130
 - processPID, 130
 - swap, 130
 - tempCharge, 130
 - tempColorCharge, 130
 - tempDecayList, 130
 - tempHighCutoff, 131
 - tempID, 131
 - tempLowCutoff, 131
 - tempMass, 131

- tempOriginalID, 131
- TempParticleData, 129
- tempParticleName, 131
- tempQuarks, 132
- tempSource, 132
- tempSpin, 132
- tempWidth, 132
- HepPDT::TestNuclearFragment, 133
- HepPDT::TestNuclearFragment
 - processUnknownID, 133
 - TestNuclearFragment, 133
- HepPID, 37
 - n, 44
 - n10, 44
 - n8, 44
 - n9, 44
 - nj, 44
 - nl, 44
 - nq1, 44
 - nq2, 44
 - nq3, 44
 - nr, 44
- HepPID
 - A, 44
 - abspid, 45
 - charge, 50
 - checkForSpecialParticle, 57
 - convIsajettoPDT, 59
 - convPDTtoIsajet, 59
 - digit, 44
 - dyonName, 56
 - EvtGenPDTMap, 42
 - EvtGenPDTMapInit, 57
 - extraBits, 45
 - findQ, 56
 - fundamentalID, 45
 - getEvtGenPDTMap, 57
 - getHerwigPDTMap, 58
 - getInverseQQbarMap, 61, 62
 - getIsajetPDTMap, 59
 - getParticleNameMap, 55
 - getPDGtoPDTMap, 60
 - getPDTEvtGenMap, 57
 - getPDTHerwigMap, 58
 - getPDTIsajetMap, 59
 - getPDTPythiaMap, 60, 61
 - getPDTQQMap, 61, 62
 - getPDTtoPDGMap, 60
 - getPythiaPDTMap, 60, 61
 - getQQbarMap, 61, 62
 - getQQPDTMap, 61, 62
 - hasBottom, 49
 - hasCharm, 48
 - hasDown, 48
 - hasFundamentalAnti, 45
 - hasStrange, 48
 - hasTop, 49
 - hasUp, 48
 - HerwigPDTMap, 42
 - HerwigPDTMapInit, 58
 - InverseQQbarMap, 43
 - InverseQQbarMapInit, 62
 - IsajetPDTMap, 43
 - IsajetPDTMapInit, 59
 - isBaryon, 46
 - isDiQuark, 46
 - isDyon, 47
 - isHadron, 46
 - isLepton, 47
 - isMeson, 46
 - isNucleus, 47
 - isPentaquark, 47
 - isQBall, 48
 - isRhadrion, 47
 - isSUSY, 47
 - isValid, 46
 - jSpin, 49
 - lambda, 45
 - listParticleNames, 55
 - location, 44
 - lSpin, 49
 - ParticleIdMap, 42
 - ParticleLookupMap, 42
 - particleName, 54, 55
 - ParticleNameInit, 56
 - PDGtoPDTMap, 43
 - PDGtoPDTMapInit, 60
 - PDTEvtGenMap, 42
 - PDTEvtGenMapInit, 57
 - PDTHerwigMap, 43
 - PDTHerwigMapInit, 58
 - PDTIsajetMap, 43
 - PDTIsajetMapInit, 59
 - PDTPythiaMap, 43
 - PDTPythiaMapInit, 61
 - PDTQQMap, 43
 - PDTQQMapInit, 62
 - PDTtoPDGMap, 43
 - PDTtoPDGMapInit, 60
 - PythiaPDTMap, 43
 - PythiaPDTMapInit, 61
 - qballName, 57
 - QQbarMap, 43
 - QQbarMapInit, 62
 - QQPDTMap, 43
 - QQPDTMapInit, 62
 - sSpin, 49
 - threeCharge, 49

- translateEvtGentoPDT, 52
- translateGeanttoPDT, 54
- translateHerwigtoPDT, 50
- translateInverseQQbar, 54
- translateIsajettoPDT, 51
- translatePDGtabletoPDT, 52
- translatePDTtoEvtGen, 52
- translatePDTtoGeant, 54
- translatePDTtoHerwig, 50
- translatePDTtoIsajet, 51
- translatePDTtoPDGtable, 53
- translatePDTtoPythia, 51
- translatePDTtoQQ, 53
- translatePythiatoPDT, 51
- translateQQbar, 53
- translateQQtoPDT, 53
- validParticleName, 55
- version, 56
- versionName, 56
- writeEvtGenTranslation, 52
- writeEvtGenTranslationLine, 58
- writeHerwigTranslation, 50
- writeHerwigTranslationLine, 58
- writeIsajetTranslation, 51
- writeIsajetTranslationLine, 59
- writeParticleNameLine, 56
- writePDGTranslation, 53
- writePDGTranslationLine, 60
- writePythiaTranslation, 52
- writePythiaTranslationLine, 61
- writeQQTranslation, 54
- writeVersion, 56
- Z, 44
- HepPID::ParticleNameMap, 106
- HepPID::ParticleNameMap
 - ~ParticleNameMap, 106
 - begin, 107
 - beginLookupMap, 107
 - end, 107
 - endLookupMap, 107
 - find, 107
 - findString, 107
 - idIterator, 106
 - lookupMap, 107
 - nameIterator, 106
 - nameMap, 107
 - ParticleNameMap, 106
- HerwigPDTMap
 - HepPID, 42
- HerwigPDTMapInit
 - HepPID, 58
- ID
 - HepPDT::ParticleData, 79
- idIterator
 - HepPID::ParticleNameMap, 106
- IDMAX
 - translateGeanttoPDT.cc, 212
 - translatePDTtoGeant.cc, 216
- InverseQQbarMap
 - HepPID, 43
- InverseQQbarMapInit
 - HepPID, 62
- IsajetPDTMap
 - HepPID, 43
- IsajetPDTMapInit
 - HepPID, 59
- isBaryon
 - HepPDT::ParticleData, 79
 - HepPDT::ParticleID, 98
 - HepPID, 46
- isBottom
 - HepPDT::Constituent, 66
- isCharm
 - HepPDT::Constituent, 66
- isDiQuark
 - HepPDT::ParticleData, 79
 - HepPDT::ParticleID, 99
 - HepPID, 46
- isDown
 - HepPDT::Constituent, 66
- isDyon
 - HepPDT::ParticleData, 79
 - HepPDT::ParticleID, 99
 - HepPID, 47
- isHadron
 - HepPDT::ParticleData, 80
 - HepPDT::ParticleID, 99
 - HepPID, 46
- isLepton
 - HepPDT::ParticleData, 80
 - HepPDT::ParticleID, 100
 - HepPID, 47
- isMeson
 - HepPDT::ParticleData, 80
 - HepPDT::ParticleID, 100
 - HepPID, 46
- isNucleus
 - HepPDT::ParticleData, 80
 - HepPDT::ParticleID, 100
 - HepPID, 47
- isPentaquark
 - HepPDT::ParticleData, 80
 - HepPDT::ParticleID, 100
 - HepPID, 47
- isQBall
 - HepPDT::ParticleData, 80
 - HepPDT::ParticleID, 101

- HepPID, 48
- isRhadron
 - HepPDT::ParticleData, 80
 - HepPDT::ParticleID, 101
 - HepPID, 47
- isStable
 - HepPDT::ParticleData, 81
- isStrange
 - HepPDT::Constituent, 66
- isSUSY
 - HepPDT::ParticleData, 81
 - HepPDT::ParticleID, 101
 - HepPID, 47
- isTop
 - HepPDT::Constituent, 66
- isUp
 - HepPDT::Constituent, 67
- isValid
 - HepPDT::ParticleID, 101
 - HepPID, 46
- iterator
 - HepPDT::DefTable, 69
- jSpin
 - HepPDT::ParticleID, 102
 - HepPID, 49
- lambda
 - HepPDT::ParticleID, 102
 - HepPID, 45
- lifetime
 - HepPDT::ParticleData, 81
 - HepPDT::ResonanceStructure, 113
- lifetime.cc, 161
- list_herwig_end_
 - examListHerwig.cc, 148
- list_herwig_init_
 - examListHerwig.cc, 148
- list_isajet_init_
 - examListIsajet.cc, 150
- list_of_examples.cc, 162
- list_of_tests.cc, 163
- list_pythia_
 - examListPythia.cc, 152
- listEvtGenTranslation.cc, 164
- listEvtGenTranslation.cc
 - main, 164
- listHerwigTranslation.cc, 165
- listHerwigTranslation.cc
 - main, 165
- listIsajetTranslation.cc, 166
- listIsajetTranslation.cc
 - main, 166
- listParticleNames
 - HepPID, 55
- listParticleNames.cc, 167
- listParticleNames.cc
 - main, 167
- listPDGTranslation.cc, 168
- listPDGTranslation.cc
 - main, 168
- listPythiaTranslation.cc, 169
- listPythiaTranslation.cc
 - main, 169
- listQQTranslation.cc, 170
- listQQTranslation.cc
 - main, 170
- location
 - HepPDT, 26
 - HepPID, 44
- lookupMap
 - HepPID::ParticleNameMap, 107
- lowerCutoff
 - HepPDT::ParticleData, 81
 - HepPDT::ResonanceStructure, 113
- lSpin
 - HepPDT::ParticleID, 102
 - HepPID, 49
- main
 - examListHerwig.cc, 149
 - examListIsajet.cc, 150
 - examListPythia.cc, 152
 - examMyPDT.cc, 154
 - listEvtGenTranslation.cc, 164
 - listHerwigTranslation.cc, 165
 - listIsajetTranslation.cc, 166
 - listParticleNames.cc, 167
 - listPDGTranslation.cc, 168
 - listPythiaTranslation.cc, 169
 - listQQTranslation.cc, 170
 - testHepPDT.cc, 204
 - testParticleIDMethods.cc, 207
 - testPID.cc, 208
- mass
 - HepPDT::ParticleData, 81
 - HepPDT::ResonanceStructure, 113
- Measurement
 - HepPDT::Measurement, 72
- Measurement.hh, 171
- multiplicity
 - HepPDT::Constituent, 67
- n
 - HepPDT, 26
 - HepPID, 44
- n10
 - HepPDT, 26

- HepPID, 44
- n8
 - HepPDT, 26
 - HepPID, 44
- n9
 - HepPDT, 26
 - HepPID, 44
- name
 - HepPDT::ParticleData, 81
- nameIterator
 - HepPID::ParticleNameMap, 106
- nameMap
 - HepPID::ParticleNameMap, 107
- nj
 - HepPDT, 26
 - HepPID, 44
- nl
 - HepPDT, 26
 - HepPID, 44
- nq1
 - HepPDT, 26
 - HepPDT::Quarks, 110
 - HepPID, 44
- nq2
 - HepPDT, 26
 - HepPDT::Quarks, 110
 - HepPID, 44
- nq3
 - HepPDT, 26
 - HepPDT::Quarks, 111
 - HepPID, 44
- nr
 - HepPDT, 26
 - HepPID, 44
- numConstituents
 - HepPDT::ParticleData, 82
- operator double
 - HepPDT::Measurement, 72
- operator()
 - HepPDT::ParticleDataTableComparison, 92
- operator<
 - HepPDT::Measurement, 72
 - HepPDT::ParticleData, 82
 - HepPDT::ParticleID, 102
- operator=
 - HepPDT::Constituent, 67
 - HepPDT::Measurement, 72
 - HepPDT::ParticleData, 82
 - HepPDT::ParticleID, 102
 - HepPDT::ResonanceStructure, 113
 - HepPDT::SpinState, 118
 - HepPDT::TempParticleData, 130
- operator==
 - HepPDT::Measurement, 73
 - HepPDT::ParticleData, 82
 - HepPDT::ParticleID, 103
 - HepPDT::SpinState, 118
- operator[]
 - HepPDT::ParticleDataTable, 89
- orbAngMom
 - HepPDT::SpinState, 118
- originalID
 - HepPDT::ParticleData, 82
- parseEvtGenAlias
 - HepPDT, 30
- parseEvtGenAliasDecayLine
 - HepPDT, 31
- parseEvtGenConj
 - HepPDT, 31
- parseEvtGenDecayLine
 - HepPDT, 31
- parseEvtGenDefinition
 - HepPDT, 31
- parseEvtGenLine
 - HepPDT, 30
- parseIsajetDecayLine
 - HepPDT::detail, 35
- parseIsajetLine
 - HepPDT::detail, 35
- parseParticleLine
 - HepPDT::detail, 35
- parsePDGline
 - HepPDT::detail, 34
- parsePythiaDecayLine
 - HepPDT::detail, 34
- parsePythiaLine
 - HepPDT::detail, 34
- parseQQDecayLine
 - HepPDT, 31
- parseQQParticle
 - HepPDT, 31
- ParticleIdMap
 - HepPID, 42
- particle
 - HepPDT::ParticleDataTable, 89, 90
- ParticleData
 - HepPDT::ParticleData, 77
- ParticleData.hh, 172
- ParticleDataTable
 - HepPDT::ParticleDataTable, 88
- ParticleDataTable.cc, 173
- ParticleDataTable.hh, 174
- ParticleDataTableComparison
 - HepPDT::ParticleDataTableComparison, 92
- ParticleDataTableComparison.hh, 175
- ParticleID

- HepPDT::ParticleID, 95
- ParticleID.cc, 176
- ParticleID.hh, 177
- ParticleIDMethods.cc, 178
- ParticleIDMethods.hh, 181
- ParticleIDTranslations.hh, 184
- ParticleLookupMap
 - HepPID, 42
- particleName
 - HepPID, 54, 55
- ParticleName.cc, 186
- ParticleName.hh, 188
- ParticleNameInit
 - HepPID, 56
- ParticleNameMap
 - HepPID::ParticleNameMap, 106
- PDGtoPDTMap
 - HepPID, 43
- PDGtoPDTMapInit
 - HepPID, 60
- PDTEvtGenMap
 - HepPID, 42
- PDTEvtGenMapInit
 - HepPID, 57
- pdtFragmentTest
 - testHepPDT.cc, 204
- PDTHerwigMap
 - HepPID, 43
- PDTHerwigMapInit
 - HepPID, 58
- PDTIsajetMap
 - HepPID, 43
- PDTIsajetMapInit
 - HepPID, 59
- PDTMap
 - HepPDT::ParticleDataTable, 88
- PDTname
 - HepPDT::ParticleData, 82
 - HepPDT::ParticleID, 103
- PDTNameMap
 - HepPDT::ParticleDataTable, 88
- PDTPythiaMap
 - HepPID, 43
- PDTPythiaMapInit
 - HepPID, 61
- PDTQQMap
 - HepPID, 43
- PDTQQMapInit
 - HepPID, 62
- pdtSimpleTest
 - testHepPDT.cc, 205
- PDTtoPDGMap
 - HepPID, 43
- PDTtoPDGMapInit
 - HepPID, 60
- pid
 - HepPDT::Constituent, 67
 - HepPDT::ParticleData, 82
 - HepPDT::ParticleID, 103
- processPID
 - HepPDT::TempParticleData, 130
- ProcessUnknownID
 - HepPDT::ProcessUnknownID, 108
- processUnknownID
 - HepPDT::HeavyIonUnknownID, 71
 - HepPDT::SimpleProcessUnknownID, 116
 - HepPDT::TestNuclearFragment, 133
- ProcessUnknownID.cc, 189
- ProcessUnknownID.hh, 190
- PythiaPDTMap
 - HepPID, 43
- PythiaPDTMapInit
 - HepPID, 61
- qballName
 - HepPID, 57
- QQbarMap
 - HepPID, 43
- QQbarMapInit
 - HepPID, 62
- QQPDTMap
 - HepPID, 43
- QQPDTMapInit
 - HepPID, 62
- Quarks
 - HepPDT::Quarks, 110
- quarks
 - HepPDT::ParticleID, 103
- quarks.cc, 191
- removeParticle
 - HepPDT::TableBuilder, 123
- resonance
 - HepPDT::ParticleData, 83
- ResonanceStructure
 - HepPDT::ResonanceStructure, 113
- ResonanceStructure.cc, 192
- ResonanceStructure.hh, 193
- setCharge
 - HepPDT::ParticleData, 83
- setColor
 - HepPDT::ParticleData, 83
- setLowerCutoff
 - HepPDT::ParticleData, 83
 - HepPDT::ResonanceStructure, 113
- setMass
 - HepPDT::ParticleData, 83

- HepPDT::ResonanceStructure, 114
- setOrbAngMom
 - HepPDT::SpinState, 118
- setSpin
 - HepPDT::ParticleData, 83
 - HepPDT::SpinState, 118
- setTotalSpin
 - HepPDT::SpinState, 118
- setTotalWidth
 - HepPDT::ParticleData, 83
 - HepPDT::ResonanceStructure, 114
- setTotalWidthFromLifetime
 - HepPDT::ParticleData, 83
 - HepPDT::ResonanceStructure, 114
- setUpperCutoff
 - HepPDT::ParticleData, 84
 - HepPDT::ResonanceStructure, 114
- sigma
 - HepPDT::Measurement, 73
- SimpleProcessUnknownID
 - HepPDT::SimpleProcessUnknownID, 116
- SimpleProcessUnknownID.hh, 194
- size
 - HepPDT::DefTable, 70
 - HepPDT::ParticleDataTable, 90
 - HepPDT::TableBuilder, 123
- sizeNameMap
 - HepPDT::ParticleDataTable, 90
- source
 - HepPDT::ParticleData, 84
- spin
 - HepPDT::ParticleData, 84
 - HepPDT::SpinState, 118
- spindtoi
 - HepPDT, 27
- spindtoi.cc, 195
- spinitod
 - HepPDT, 27
- spinitod.cc, 196
- SpinState
 - HepPDT::SpinState, 117
- SpinState.hh, 197
- sSpin
 - HepPDT::ParticleID, 104
 - HepPID, 49
- stringtodouble
 - HepPDT, 27
- stringtodouble.cc, 198
- stringtodouble.hh, 199
- swap
 - HepPDT, 26, 27, 29
 - HepPDT::Constituent, 67
 - HepPDT::Measurement, 73
 - HepPDT::ParticleData, 84
 - HepPDT::ParticleID, 104
 - HepPDT::ResonanceStructure, 114
 - HepPDT::SpinState, 118
 - HepPDT::TempParticleData, 130
- TableBuilder
 - HepPDT::TableBuilder, 121
- TableBuilder.hh, 200
- tableName
 - HepPDT::ParticleDataTable, 90
- TDDlist
 - HepPDT, 25
- tempAlias
 - HepPDT::TempAliasData, 124
- TempAliasData
 - HepPDT::TempAliasData, 124
- tempAliasDecayList
 - HepPDT::TempAliasData, 124
- tempAliasedParticle
 - HepPDT::TempAliasData, 125
- tempBranchingFraction
 - HepPDT::TempDecayData, 127
- tempCharge
 - HepPDT::TempParticleData, 130
- tempChargeConj
 - HepPDT::TempAliasData, 125
- tempColorCharge
 - HepPDT::TempParticleData, 130
- TempConstituent
 - HepPDT::TempConstituent, 126
- tempConstituentPID
 - HepPDT::TempConstituent, 126
- tempDaughterList
 - HepPDT::TempDecayData, 127
- tempDecayList
 - HepPDT::TempParticleData, 130
- tempDecayName
 - HepPDT::TempDecayData, 127
- tempDecayParameters
 - HepPDT::TempDecayData, 127
- tempHighCutoff
 - HepPDT::TempParticleData, 131
- tempID
 - HepPDT::TempParticleData, 131
- tempLowCutoff
 - HepPDT::TempParticleData, 131
- TempMap
 - HepPDT::ParticleDataTable, 88
- tempMass
 - HepPDT::TempParticleData, 131
- tempMultiplicity
 - HepPDT::TempConstituent, 126
- tempOriginalID
 - HepPDT::TempParticleData, 131

- TempParticleData
 - HepPDT::TempParticleData, 129
- TempParticleData.cc, 202
- TempParticleData.hh, 203
- tempParticleName
 - HepPDT::TempParticleData, 131
- tempQuarks
 - HepPDT::TempParticleData, 132
- tempSource
 - HepPDT::TempParticleData, 132
- tempSpin
 - HepPDT::TempParticleData, 132
- tempWidth
 - HepPDT::TempParticleData, 132
- testDiQuark
 - testPID.cc, 208
- testHadron
 - testPID.cc, 208
- testHasMethods
 - testPID.cc, 209
- testHepPDT.cc, 204
- testHepPDT.cc
 - duplicateFragmentTest, 204
 - main, 204
 - pdTFragmentTest, 204
 - pdTSimpleTest, 205
 - testPDMMethods, 205
- testLepton
 - testPID.cc, 209
- TestNuclearFragment
 - HepPDT::TestNuclearFragment, 133
- TestNuclearFragment.hh, 206
- testNucleus
 - testPID.cc, 209
- testParticleIDMethods.cc, 207
- testParticleIDMethods.cc
 - main, 207
- testPDMMethods
 - testHepPDT.cc, 205
- testPID.cc, 208
- testPID.cc
 - main, 208
 - testDiQuark, 208
 - testHadron, 208
 - testHasMethods, 209
 - testLepton, 209
 - testNucleus, 209
 - testUnknown, 209
 - testValid, 210
- testUnknown
 - testPID.cc, 209
- testValid
 - testPID.cc, 210
- threeCharge
 - HepPDT::ParticleID, 104
- HepPID, 49
- totalSpin
 - HepPDT::SpinState, 118
- totalWidth
 - HepPDT::ParticleData, 84
 - HepPDT::ResonanceStructure, 114
- translateEvtGen.cc, 211
- translateEvtGentoPDT
 - HepPID, 52
- translateGeanttoPDT
 - HepPID, 54
- translateGeanttoPDT.cc, 212
- translateGeanttoPDT.cc
 - IDMAX, 212
- translateHerwig.cc, 213
- translateHerwigtoPDT
 - HepPID, 50
- translateInverseQQbar
 - HepPID, 54
- translateIsajet.cc, 214
- translateIsajettoPDT
 - HepPID, 51
- translatePDG.cc, 215
- translatePDGtabletoPDT
 - HepPID, 52
- translatePDTtoEvtGen
 - HepPID, 52
- translatePDTtoGeant
 - HepPID, 54
- translatePDTtoGeant.cc, 216
- translatePDTtoGeant.cc
 - IDMAX, 216
- translatePDTtoHerwig
 - HepPID, 50
- translatePDTtoIsajet
 - HepPID, 51
- translatePDTtoPDGtable
 - HepPID, 53
- translatePDTtoPythia
 - HepPID, 51
- translatePDTtoQQ
 - HepPID, 53
- translatePythia.cc, 217
- translatePythiatoPDT
 - HepPID, 51
- translateQQ.cc, 218
- translateQQbar
 - HepPID, 53
- translateQQtoPDT
 - HepPID, 53
- upperCutoff
 - HepPDT::ParticleData, 84

- HepPDT::ResonanceStructure, 115
- validParticleName
 - HepPID, 55
- value
 - HepPDT::Measurement, 73
- version
 - HepPDT, 30
 - HepPID, 56
- Version.cc, 220, 221
- Version.hh, 222, 223
- versionName
 - HepPDT, 30
 - HepPID, 56
- write
 - HepPDT::ParticleData, 85
- write.cc, 224
- writeDefinitions
 - HepPDT::DefTable, 70
- writeEvtGenStream
 - HepPDT, 27
- writeEvtGenTranslation
 - HepPID, 52
- writeEvtGenTranslationLine
 - HepPID, 58
- writeHerwigStream
 - HepPDT, 27
- writeHerwigTranslation
 - HepPID, 50
- writeHerwigTranslationLine
 - HepPID, 58
- writeIsajetStream
 - HepPDT, 27
- writeIsajetTranslation
 - HepPID, 51
- writeIsajetTranslationLine
 - HepPID, 59
- writeLine
 - examListPythia.cc, 153
- writeParticleData
 - HepPDT::ParticleDataTable, 90
- writeParticleInfo
 - HepPDT::ParticleData, 85
 - HepPDT::ParticleDataTable, 91
- writeParticleNameLine
 - HepPID, 56
- writeParticleStatus
 - HepPDT::ParticleDataTable, 91
- writeParticleTranslation
 - HepPDT::ParticleData, 85
 - HepPDT::ParticleDataTable, 91
- writePDGStream
 - HepPDT, 26
- writePDGTranslation
 - HepPID, 53
- writePDGTranslationLine
 - HepPID, 60
- writePythiaStream
 - HepPDT, 27
- writePythiaTranslation
 - HepPID, 52
- writePythiaTranslationLine
 - HepPID, 61
- writeQQStream
 - HepPDT, 27
- writeQQTranslation
 - HepPID, 54
- writeVersion
 - HepPDT, 30
 - HepPID, 56
- Z
 - HepPDT::ParticleID, 104
 - HepPID, 44