# HepPDT Reference Manual

3.03.00

Generated by Doxygen 1.5.1-3

Fri Oct 24 13:41:35 2008

# Contents

# Chapter 1

# HepPDT Directory Hierarchy

## 1.1 HepPDT Directories

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

# Chapter 2

# HepPDT Namespace Index

## 2.1 HepPDT Namespace List

Here is a list of all namespaces with brief descriptions:

# Chapter 3

# HepPDT Hierarchical Index

## 3.1 HepPDT Class Hierarchy

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

# Chapter 4

# HepPDT Class Index

## 4.1 HepPDT Class List

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

# Chapter 5

# HepPDT File Index

## 5.1 HepPDT File List

Here is a list of all files with brief descriptions:

# Chapter 6

# HepPDT Directory Documentation

## 6.1 /home/cepa01/garren/lcg/heppdt/HepPDT-3.03.00/examples/ Directory Reference

### Directories

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

### Files

- file **list_of_examples.cc**

## 6.2 /home/cepa01/garren/lcg/heppdt/HepPDT-3.03.00/tests/HepPDT/ Directory Reference

### Files

- file **listEvtGenNames.cc.in**
- file **listPDGNames.cc.in**
- file **listPythiaNames.cc.in**
- file **testHepPDT.cc**
- file **TestNuclearFragment.hh**
- file **testPID.cc**
- file **testReadEvtGen.cc.in**
- file **testReadIsajet.cc.in**
- file **testReadParticleTable.cc.in**
- file **testReadQQ.cc.in**

## 6.3 /home/cepa01/garren/lcg/heppdt/HepPDT-3.03.00/examples/HepPDT/ Directory Reference

### Files

- file **examMyPDT.cc**

## 6.4 /home/cepa01/garren/lcg/heppdt/HepPDT-3.03.00/include/HepPDT/ Directory Reference

### Files

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

## 6.5 /home/cepa01/garren/lcg/heppdt/HepPDT-3.03.00/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-3.03.00/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-3.03.00/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-3.03.00/tests/HepPID/ Directory Reference

### Files

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

## 6.9 /home/cepa01/garren/lcg/heppdt/HepPDT-3.03.00/examples/HepPID/ Directory Reference

### Files

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

## 6.10 /home/cepa01/garren/lcg/heppdt/HepPDT-3.03.00/include/ Directory Reference

### Directories

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

## 6.11 /home/cepa01/garren/lcg/heppdt/HepPDT-3.03.00/src/ Directory Reference

### Directories

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

## 6.12 /home/cepa01/garren/lcg/heppdt/HepPDT-3.03.00/tests/ Directory Reference

### Directories

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

### Files

- file **list_of_tests.cc**

# Chapter 7

# HepPDT Namespace Documentation

## 7.1 HepPDT Namespace Reference

**HepPDT** (p. 25) is a Particle Data Table namespace.

### Classes

- class **Constituent**

  *The **Constituent** (*p. *65) class has information about constituent particles.*

- class **DefTable**

  *The **DefTable** (*p. *68) class holds EvtGen definitions.*

- class **HeavyIonUnknownID**

  *The **HeavyIonUnknownID** (*p. *71) class inherits from **ProcessUnknownID** (*p. *110).*

- class **Measurement**

  *The **Measurement** (*p. *73) class defines a value with its error.*

- class **ParticleData**

  *The **ParticleData** (*p. *76) class holds data for a single particle in the table.*

- class **ParticleDataTable**

  *The **ParticleDataTable** (*p. *88) class is the core of **HepPDT** (*p. *25).*

- class **ParticleDataTableComparison**

  *The **ParticleDataTableComparison** (*p. *95) class provides a utility for sorting the PDT.*

- struct **Quarks**

  *constituent quarks*

- class **ParticleID**

  *The **ParticleID** (*p. *96) has various utilities to extract information from the particle ID.*

- class **ProcessUnknownID**

*The **ProcessUnknownID** (p. 110) class is abstract.*

- class **ResonanceStructure**

  *The **ResonanceStructure** (p. 114) class is holds mass and width information.*

- class **SimpleProcessUnknownID**

  *The **SimpleProcessUnknownID** (p. 118) class inherits from **ProcessUnknownID** (p. 110).*

- class **SpinState**

  *The **SpinState** (p. 119) class holds spin information.*

- class **TableBuilder**

  *The **TableBuilder** (p. 122) class is used to construct a **ParticleDataTable** (p. 88).*

- 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**

  *The **TestNuclearFragment** (p. 136) class inherits from **ProcessUnknownID** (p. 110).*

## Namespaces

- namespace **detail**

  ***HepPDT::detail** (p. 35) is for internal use.*

## Typedefs

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

  *useful typedef*

## Enumerations

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

## Functions

- void **swap** (**Constituent** &first, **Constituent** &second)
- void **swap** (**Measurement** &first, **Measurement** &second)
- double **NaN** ()
- void **swap** (**ParticleData** &first, **ParticleData** &second)
- bool **writePDGStream** (std::ostream &os, const **ParticleDataTable** &table)
- bool **writePythiaStream** (std::ostream &os, const **ParticleDataTable** &table)
- bool **writeHerwigStream** (std::ostream &os, const **ParticleDataTable** &table)
- bool **writeIsajetStream** (std::ostream &os, const **ParticleDataTable** &table)
- bool **writeQQStream** (std::ostream &os, const **ParticleDataTable** &table)
- bool **writeEvtGenStream** (std::ostream &os, const **ParticleDataTable** &table)
- double **spinitod** (int js)

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

- int **spindtoi** (double spin)

  *convert an actual spin to 2J+1*

- void **swap** (**ParticleID** &first, **ParticleID** &second)
- void **swap** (**ResonanceStructure** &first, **ResonanceStructure** &second)
- void **swap** (**SpinState** &first, **SpinState** &second)
- double **stringtodouble** (std::string &numb)

  *extract a double from a string*

- bool **addPDGParticles** (std::istream &pdfile, **TableBuilder** &tb)

  *read PDG input and add particles to the table*

- bool **addPythiaParticles** (std::istream &pdfile, **TableBuilder** &tb)

  *read Pythia input and add particles to the table*

- bool **addHerwigParticles** (std::istream &pdfile, **TableBuilder** &tb)
- bool **addIsajetParticles** (std::istream &pdfile, **TableBuilder** &tb)

  *read Isajet particle input and add particles to the table*

- bool **addIsajetDecay** (std::istream &pdfile, **TableBuilder** &tb)

  *read Isajet decay input and add decay information to the table*

- bool **addQQParticles** (std::istream &pdfile, **TableBuilder** &tb)

  *read QQ input and add particles to the table*

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

  *read EvtGen input and add particles to the table*

- bool **addParticleTable** (std::istream &pdfile, **TableBuilder** &tb, bool validate=false)
- double **calculateWidthFromLifetime** (double)
- void **swap** (**TempParticleData** &first, **TempParticleData** &second)
- void **version** ()

  *print **HepPDT** (p. 25) version*

- void **writeVersion** (std::ostream &os)

*write **HepPDT** (p. 25) version to os*

- std::string **versionName** ()

  *return **HepPDT** (p. 25) version*

- bool **getEvtGenLineType** (std::string &ltype, int &id, std::string &name, const std::string &pdline)
- void **parseEvtGenLine** (**TempParticleData** &tpd, const std::string &pdline)
- void **parseEvtGenAlias** (**TempAliasData** &tad, const std::string &pdline)
- bool **parseEvtGenDecayLine** (**TempParticleData** &tpd, const std::string &pdline)
- bool **parseEvtGenAliasDecayLine** (**TempAliasData** &tad, const std::string &pdline)
- void **parseEvtGenConj** (std::string &cname, const std::string &pdline)
- void **parseEvtGenDefinition** (std::string &def, double &val, const std::string &pdline)
- bool **getQQLineType** (std::string &ltype, 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 Detailed Description

**HepPDT** (p. 25) is a Particle Data Table namespace.

The **HepPDT** (p. 25) classes are used to create a Particle Data Table.

## 7.1.2 Typedef Documentation

### 7.1.2.1 typedef std::vector<TempDecayData> HepPDT::TDDlist

useful typedef

Definition at line 45 of file TempParticleData.hh.

## 7.1.3 Enumeration Type Documentation

### 7.1.3.1 enum HepPDT::location

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

**Enumerator:**

***nj***

***nq3***

***nq2***

***nq1***

***nl***

***nr***

***n***

***n8***

***n9***

***n10***

Definition at line 36 of file ParticleID.hh.

## 7.1.4 Function Documentation

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

read EvtGen input and add particles to the table

**Examples:**

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

Definition at line 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().

Referenced by main().

### 7.1.4.2 bool HepPDT::addHerwigParticles (std::istream & *pdfile*, TableBuilder & *tb*)

Definition at line 13 of file addHerwigParticles.cc.

### 7.1.4.3 bool HepPDT::addIsajetDecay (std::istream & *pdfile*, TableBuilder & *tb*)

read Isajet decay input and add decay information to the table

### 7.1.4.4 bool HepPDT::addIsajetParticles (std::istream & *pdfile*, TableBuilder & *tb*)

read Isajet particle input and add particles to the table

**Examples:**

**testReadIsajet.cc.in**.

Definition at line 14 of file addIsajetParticles.cc.

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

Referenced by main().

### 7.1.4.5 bool HepPDT::addParticleTable (std::istream & *pdfile*, TableBuilder & *tb*, bool *validate* = `false`)

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

validate=true => verify that the **ParticleID** (p. 96) is valid

**Examples:**

**testReadParticleTable.cc.in**.

Definition at line 21 of file addParticleTable.cc.

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

Referenced by main().

#### 7.1.4.6 bool HepPDT::addPDGParticles (std::istream & *pdfile*, TableBuilder & *tb*)

read PDG input and add particles to the table

**Examples:**

**listPDGNames.cc.in**, and **testHepPDT.cc**.

Definition at line 22 of file addPDGParticles.cc.

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

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

#### 7.1.4.7 bool HepPDT::addPythiaParticles (std::istream & *pdfile*, TableBuilder & *tb*)

read Pythia input and add particles to the table

**Examples:**

**listPythiaNames.cc.in**.

Definition at line 20 of file addPythiaParticles.cc.

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

Referenced by main().

#### 7.1.4.8 bool HepPDT::addQQParticles (std::istream & *pdfile*, TableBuilder & *tb*)

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().

Referenced by main().

#### 7.1.4.9 double HepPDT::calculateWidthFromLifetime (double)

free function Given the lifetime, calculate the width.

Definition at line 13 of file calculateWidthFromLifetime.cc.

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

#### 7.1.4.10 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().

#### 7.1.4.11 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().

#### 7.1.4.12 double HepPDT::NaN () `[inline]`

Definition at line 11 of file Measurement.icc.

#### 7.1.4.13 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().

#### 7.1.4.14 bool HepPDT::parseEvtGenAliasDecayLine (TempAliasData & *tad*, const std::string & *pdline*)

Definition at line 201 of file addEvtGenParticles.cc.

Referenced by addEvtGenParticles().

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

Definition at line 239 of file addEvtGenParticles.cc.

Referenced by addEvtGenParticles().

#### 7.1.4.16 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().

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

Definition at line 254 of file addEvtGenParticles.cc.

Referenced by addEvtGenParticles().

#### 7.1.4.18 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().

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

Definition at line 158 of file addQQParticles.cc.

Referenced by addQQParticles().

#### 7.1.4.20 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().

#### 7.1.4.21 int HepPDT::spindtoi (double *spin*)

convert an actual spin to 2J+1

Definition at line 13 of file spindtoi.cc.

#### 7.1.4.22 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().

#### 7.1.4.23 double HepPDT::stringtodouble (std::string & *numb*)

extract a double from a string

Definition at line 15 of file stringtodouble.cc.

#### 7.1.4.24 void HepPDT::swap (TempParticleData & *first*, TempParticleData & *second*) `[inline]`

Definition at line 106 of file TempParticleData.hh.

References HepPDT::TempParticleData::swap().

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

#### 7.1.4.25 void HepPDT::swap (SpinState & *first*, SpinState & *second*) `[inline]`

Definition at line 69 of file SpinState.hh.

References HepPDT::SpinState::swap().

#### 7.1.4.26 void HepPDT::swap (ResonanceStructure & *first*, ResonanceStructure & *second*) `[inline]`

Definition at line 83 of file ResonanceStructure.hh.

References HepPDT::ResonanceStructure::swap().

#### 7.1.4.27 void HepPDT::swap (ParticleID & *first*, ParticleID & *second*) `[inline]`

Definition at line 159 of file ParticleID.hh.

References HepPDT::ParticleID::swap().

#### 7.1.4.28 void HepPDT::swap (ParticleData & *first*, ParticleData & *second*) `[inline]`

Definition at line 173 of file ParticleData.hh.

References HepPDT::ParticleData::swap().

#### 7.1.4.29 void HepPDT::swap (Measurement & *first*, Measurement & *second*) `[inline]`

Definition at line 45 of file Measurement.hh.

References HepPDT::Measurement::swap().

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

Definition at line 77 of file Constituent.hh.

References HepPDT::Constituent::swap().

#### 7.1.4.31 void HepPDT::version ()

print **HepPDT** (p. 25) version

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

References versionName().

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

#### 7.1.4.32 std::string HepPDT::versionName ()

return **HepPDT** (p. 25) version

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

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

#### 7.1.4.33 bool HepPDT::writeEvtGenStream (std::ostream & *os*, const ParticleDataTable & *table*)

#### 7.1.4.34 bool HepPDT::writeHerwigStream (std::ostream & *os*, const ParticleDataTable & *table*)

#### 7.1.4.35 bool HepPDT::writeIsajetStream (std::ostream & *os*, const ParticleDataTable & *table*)

#### 7.1.4.36 bool HepPDT::writePDGStream (std::ostream & *os*, const ParticleDataTable & *table*)

#### 7.1.4.37 bool HepPDT::writePythiaStream (std::ostream & *os*, const ParticleDataTable & *table*)

#### 7.1.4.38 bool HepPDT::writeQQStream (std::ostream & *os*, const ParticleDataTable & *table*)

#### 7.1.4.39 void HepPDT::writeVersion (std::ostream & *os*)

write **HepPDT** (p. 25) version to os

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

References versionName().

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

## 7.2 HepPDT::detail Namespace Reference

**HepPDT::detail** (p. 35) is for internal use.

### Functions

- void **getPDGpid** (std::vector< int > &idlist, std::string &pdline)

  *for internal use*

- void **getPDGnames** (std::vector< std::string > &namelst, std::string &pdline)

  *for internal use*

- void **parsePDGline** (**TempParticleData** &tpd, std::string &pdline)

  *for internal use*

- bool **CheckPDGEntry** (**TempParticleData** &tpd, const std::string &, double, double)

  *for internal use*

- bool **getPythiaid** (int &id, const std::string &pdline)

  *for internal use*

- void **parsePythiaLine** (**TempParticleData** &tpd, int &anti, std::string &aname, const std::string &pdline)

  *for internal use*

- void **parsePythiaDecayLine** (**TempParticleData** &tpd, const std::string &pdline)

  *for internal use*

- **TempDecayData** **getPythiaDecay** (const std::string &pdline)

  *for internal use*

- bool **getIsajetID** (int &id, const std::string &pdline)

  *for internal use*

- void **parseIsajetLine** (**TempParticleData** &tpd, const std::string &pdline)

  *for internal use*

- void **parseIsajetDecayLine** (**TempParticleData** &tpd, const std::string &pdline, **TableBuilder** &tb)

  *for internal use*

- bool **getParticleID** (int &id, const std::string &pdline)

  *for internal use*

- void **parseParticleLine** (**TempParticleData** &tpd, const std::string &pdline)

  *for internal use*

## 7.2.1 Detailed Description

**HepPDT::detail** (p. 35) is for internal use.

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

## 7.2.2 Function Documentation

### 7.2.2.1 bool HepPDT::detail::CheckPDGEntry (TempParticleData & *tpd*, const std::string &, double, double)

for internal use

Definition at line 67 of file addPDGParticles.cc.

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

Referenced by parsePDGline().

### 7.2.2.2 bool HepPDT::detail::getIsajetID (int & *id*, const std::string & *pdline*)

for internal use

Definition at line 17 of file getIsajetID.cc.

Referenced by HepPDT::addIsajetParticles().

### 7.2.2.3 bool HepPDT::detail::getParticleID (int & *id*, const std::string & *pdline*)

for internal use

Definition at line 50 of file addParticleTable.cc.

Referenced by HepPDT::addParticleTable().

### 7.2.2.4 void HepPDT::detail::getPDGnames (std::vector< std::string > & *namelst*, std::string & *pdline*)

for internal use

Definition at line 40 of file getPDGpid.cc.

Referenced by HepPDT::addPDGParticles().

### 7.2.2.5 void HepPDT::detail::getPDGpid (std::vector< int > & *idlist*, std::string & *pdline*)

for internal use

Definition at line 20 of file getPDGpid.cc.

Referenced by HepPDT::addPDGParticles().

### 7.2.2.6 TempDecayData HepPDT::detail::getPythiaDecay (const std::string & *pdline*)

for internal use

#### 7.2.2.7 bool HepPDT::detail::getPythiaid (int & *id*, const std::string & *pdline*)

for internal use

Definition at line 20 of file getPythiaid.cc.

Referenced by HepPDT::addPythiaParticles().

#### 7.2.2.8 void HepPDT::detail::parseIsajetDecayLine (TempParticleData & *tpd*, const std::string & *pdline*, TableBuilder & *tb*)

for internal use

#### 7.2.2.9 void HepPDT::detail::parseIsajetLine (TempParticleData & *tpd*, const std::string & *pdline*)

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().

#### 7.2.2.10 void HepPDT::detail::parseParticleLine (TempParticleData & *tpd*, const std::string & *pdline*)

for internal use

Definition at line 70 of file addParticleTable.cc.

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

Referenced by HepPDT::addParticleTable().

#### 7.2.2.11 void HepPDT::detail::parsePDGline (TempParticleData & *tpd*, std::string & *pdline*)

for internal use

Definition at line 51 of file addPDGParticles.cc.

References CheckPDGEntry().

Referenced by HepPDT::addPDGParticles().

#### 7.2.2.12 void HepPDT::detail::parsePythiaDecayLine (TempParticleData & *tpd*, const std::string & *pdline*)

for internal use

Definition at line 98 of file addPythiaParticles.cc.

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

Referenced by HepPDT::addPythiaParticles().

#### 7.2.2.13 void HepPDT::detail::parsePythiaLine (TempParticleData & *tpd*, int & *anti*, std::string & *aname*, const std::string & *pdline*)

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().

## 7.3 HepPID Namespace Reference

The **HepPID** (p. 39) namespace has independent particle ID translation methods.

### Classes

- class **ParticleNameMap**

### Typedefs

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

### Functions

- unsigned short **digit** (**location** loc, const int &pid)

  *return the digit at a named location in the PID*

- int **A** (const int &pid)
- int **Z** (const int &pid)
- int **lambda** (const int &pid)
- int **abspid** (const int &pid)

  *absolute value of particle ID*

- int **fundamentalID** (const int &pid)

  *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)
- bool **isValid** (const int &pid)

  *is this a valid ID?*

- bool **isMeson** (const int &pid)

  *is this a valid meson ID?*

- bool **isBaryon** (const int &pid)

  *is this a valid baryon ID?*

- bool **isDiQuark** (const int &pid)

  *is this a valid diquark ID?*

- bool **isHadron** (const int &pid)

  *is this a valid hadron ID?*

- bool **isLepton** (const int &pid)

  *is this a valid lepton ID?*

- bool **isNucleus** (const int &pid)

  *is this a valid ion ID?*

- bool **isPentaquark** (const int &pid)

  *is this a valid pentaquark ID?*

- bool **isSUSY** (const int &pid)

  *is this a valid SUSY ID?*

- bool **isRhadron** (const int &pid)

  *is this a valid R-hadron ID?*

- bool **hasUp** (const int &pid)

  *does this particle contain an up quark?*

- bool **hasDown** (const int &pid)

  *does this particle contain a down quark?*

- bool **hasStrange** (const int &pid)

  *does this particle contain a strange quark?*

- bool **hasCharm** (const int &pid)

  *does this particle contain a charm quark?*

- bool **hasBottom** (const int &pid)

  *does this particle contain a bottom quark?*

- bool **hasTop** (const int &pid)

  *does this particle contain a top quark?*

- int **jSpin** (const int &pid)

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

- int **sSpin** (const int &pid)

  *sSpin returns 2S+1, where S is the spin*

- int **lSpin** (const int &pid)

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

- int **threeCharge** (const int &pid)

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

- int **translateHerwigtoPDT** (const int herwigID)

  *translate Herwig to PDG standard*

- int **translatePDTtoHerwig** (const int pid)

  *translate PDG standard to Herwig*

- void **writeHerwigTranslation** (std::ostream &os)

  *output the translation list*

- int **translateIsajettoPDT** (const int isajetID)

  *translate Isajet to PDG standard*

- int **translatePDTtoIsajet** (const int pid)

  *translate PDG standard to Isajet*

- void **writeIsajetTranslation** (std::ostream &os)

  *output the translation list*

- int **translatePythiatoPDT** (const int pythiaID)

  *translate Pythia to PDG standard*

- int **translatePDTtoPythia** (const int pid)

  *translate PDG standard to Pythia*

- void **writePythiaTranslation** (std::ostream &os)

  *output the translation list*

- int **translateEvtGentoPDT** (const int evtGenID)

  *translate EvtGen to PDG standard*

- int **translatePDTtoEvtGen** (const int pid)

  *translate PDG standard to EvtGen*

- void **writeEvtGenTranslation** (std::ostream &os)

  *output the translation list*

- int **translatePDGtabletoPDT** (const int pdgID)
  *translate PDG table to PDG standard*
- int **translatePDTtoPDGtable** (const int pid)
  *translate PDG standard to PDG table*
- void **writePDGTranslation** (std::ostream &os)
  *output the translation list*
- int **translateQQtoPDT** (const int qqID)
  *translate QQ to PDG standard*
- int **translatePDTtoQQ** (const int pid)
  *translate PDG standard to QQ*
- int **translateQQbar** (const int id)
  *QQ helper function.*
- int **translateInverseQQbar** (const int id)
  *QQ helper function.*
- void **writeQQTranslation** (std::ostream &os)
  *output the translation list*
- int **translateGeanttoPDT** (const int geantID)
  *translate Geant3 to PDG standard*
- int **translatePDTtoGeant** (const int pid)
  *translate PDG standard to Geant3*
- std::string **particleName** (const int &)
  *get a known **HepPID** (p. 39) 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. 108) for other purposes*
- void **version** ()

*print **HepPID** (*p. *39) version*

- void **writeVersion** (std::ostream &os)

  *write **HepPID** (*p. *39) version to os*

- std::string **versionName** ()

  *return **HepPID** (*p. *39) version*

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

- **QQbarMap** const & **getQQbarMap** ()
- **InverseQQbarMap** const & **getInverseQQbarMap** ()
- **QQPDTMap** const & **QQPDTMapInit** ()
- **QQbarMap** const & **QQbarMapInit** ()
- **PDTQQMap** const & **PDTQQMapInit** ()
- **InverseQQbarMap** const & **InverseQQbarMapInit** ()
- **QQPDTMap** const & **getQQPDTMap** ()
- **PDTQQMap** const & **getPDTQQMap** ()
- **QQbarMap** const & **getQQbarMap** ()
- **InverseQQbarMap** const & **getInverseQQbarMap** ()

## 7.3.1 Detailed Description

The **HepPID** (p. 39) namespace has independent particle ID translation methods.

The **HepPID** (p. 39) namespace contains a set of independent particle ID translation methods

## 7.3.2 Typedef Documentation

### 7.3.2.1 typedef std::map< int, int > HepPID::EvtGenPDTMap

Definition at line 33 of file translateEvtGen.cc.

### 7.3.2.2 typedef std::map< int, int > HepPID::HerwigPDTMap

Definition at line 33 of file translateHerwig.cc.

### 7.3.2.3 typedef std::map< int, int > HepPID::InverseQQbarMap

Definition at line 40 of file translateQQ.cc.

### 7.3.2.4 typedef std::map< int, int > HepPID::IsajetPDTMap

Definition at line 36 of file translateIsajet.cc.

### 7.3.2.5 typedef std::map< int, std::string > HepPID::PartcleIdMap

Definition at line 33 of file ParticleName.cc.

### 7.3.2.6 typedef std::map< std::string, int > HepPID::ParticleLookupMap

Definition at line 34 of file ParticleName.cc.

### 7.3.2.7 typedef std::map< int, int > HepPID::PDGtoPDTMap

Definition at line 33 of file translatePDG.cc.

#### 7.3.2.8 typedef std::map< int, int > HepPID::PDTEvtGenMap

Definition at line 34 of file translateEvtGen.cc.

#### 7.3.2.9 typedef std::map< int, int > HepPID::PDTHerwigMap

Definition at line 34 of file translateHerwig.cc.

#### 7.3.2.10 typedef std::map< int, int > HepPID::PDTIsajetMap

Definition at line 37 of file translateIsajet.cc.

#### 7.3.2.11 typedef std::map< int, int > HepPID::PDTPythiaMap

Definition at line 34 of file translatePythia.cc.

#### 7.3.2.12 typedef std::map< int, int > HepPID::PDTQQMap

Definition at line 38 of file translateQQ.cc.

#### 7.3.2.13 typedef std::map< int, int > HepPID::PDTtoPDGMap

Definition at line 34 of file translatePDG.cc.

#### 7.3.2.14 typedef std::map< int, int > HepPID::PythiaPDTMap

Definition at line 33 of file translatePythia.cc.

#### 7.3.2.15 typedef std::map< int, int > HepPID::QQbarMap

Definition at line 39 of file translateQQ.cc.

#### 7.3.2.16 typedef std::map< int, int > HepPID::QQPDTMap

Definition at line 37 of file translateQQ.cc.

### 7.3.3 Enumeration Type Documentation

#### 7.3.3.1 enum HepPID::location

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

**Enumerator:**

***nj***

***nq3***

***nq2***

***nq1***

***nl***

***nr***

***n***

***n8***

***n9***

***n10***

Definition at line 24 of file ParticleIDMethods.hh.

## 7.3.4 Function Documentation

### 7.3.4.1 int HepPID::A (const int & *pid*)

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

Definition at line 86 of file ParticleIDMethods.cc.

References abspid(), and isNucleus().

Referenced by main().

### 7.3.4.2 int HepPID::abspid (const int & *pid*)

absolute value of particle ID

Definition at line 42 of file ParticleIDMethods.cc.

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

### 7.3.4.3 int HepPID::@63::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().

### 7.3.4.4 int HepPID::@63::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().

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

return the digit at a named location in the PID

Definition at line 54 of file ParticleIDMethods.cc.

References abspid().

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

#### 7.3.4.6 EvtGenPDTMap const& HepPID::@60::EvtGenPDTMapInit () `[static]`

Definition at line 41 of file translateEvtGen.cc.

Referenced by getEvtGenPDTMap().

#### 7.3.4.7 int HepPID::extraBits (const int & *pid*)

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

Definition at line 48 of file ParticleIDMethods.cc.

References abspid().

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

#### 7.3.4.8 bool HepPID::@58::findQ (const int & *pid*, const int & *q*) `[static]`

Definition at line 17 of file ParticleIDMethods.cc.

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

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

#### 7.3.4.9 int HepPID::fundamentalID (const int & *pid*)

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

Definition at line 64 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().

#### 7.3.4.10 EvtGenPDTMap const& HepPID::@60::getEvtGenPDTMap () `[static]`

Definition at line 608 of file translateEvtGen.cc.

References EvtGenPDTMapInit().

#### 7.3.4.11 EvtGenPDTMap const& HepPID::@60::getEvtGenPDTMap () `[static]`

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

#### 7.3.4.12 HerwigPDTMap const& HepPID::@62::getHerwigPDTMap () `[static]`

Definition at line 500 of file translateHerwig.cc.

References HerwigPDTMapInit().

#### 7.3.4.13 HerwigPDTMap const& HepPID::@62::getHerwigPDTMap () `[static]`

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

#### 7.3.4.14 InverseQQbarMap const& HepPID::@67::getInverseQQbarMap () `[static]`

Definition at line 560 of file translateQQ.cc.

References InverseQQbarMapInit().

#### 7.3.4.15 InverseQQbarMap const& HepPID::@67::getInverseQQbarMap () `[static]`

Referenced by translateInverseQQbar().

#### 7.3.4.16 IsajetPDTMap const& HepPID::@63::getIsajetPDTMap () `[static]`

Definition at line 891 of file translateIsajet.cc.

References IsajetPDTMapInit().

#### 7.3.4.17 IsajetPDTMap const& HepPID::@63::getIsajetPDTMap () `[static]`

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

#### 7.3.4.18 ParticleNameMap const & HepPID::getParticleNameMap ()

access the **ParticleNameMap** (p. 108) for other purposes

Definition at line 1657 of file ParticleName.cc.

References ParticleNameInit().

Referenced by particleName(), and validParticleName().

#### 7.3.4.19 PDGtoPDTMap const& HepPID::@64::getPDGtoPDTMap () `[static]`

Definition at line 369 of file translatePDG.cc.

References PDGtoPDTMapInit().

#### 7.3.4.20 PDGtoPDTMap const& HepPID::@64::getPDGtoPDTMap () `[static]`

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

#### 7.3.4.21 PDTEvtGenMap const& HepPID::@60::getPDTEvtGenMap () `[static]`

Definition at line 617 of file translateEvtGen.cc.

References PDTEvtGenMapInit().

#### 7.3.4.22 PDTEvtGenMap const& HepPID::@60::getPDTEvtGenMap () `[static]`

Referenced by translatePDTtoEvtGen().

#### 7.3.4.23 PDTHerwigMap const& HepPID::@62::getPDTHerwigMap () `[static]`

Definition at line 509 of file translateHerwig.cc.

References PDTHerwigMapInit().

#### 7.3.4.24 PDTHerwigMap const& HepPID::@62::getPDTHerwigMap () `[static]`

Referenced by translatePDTtoHerwig().

#### 7.3.4.25 PDTIsajetMap const& HepPID::@63::getPDTIsajetMap () `[static]`

Definition at line 900 of file translateIsajet.cc.

References PDTIsajetMapInit().

#### 7.3.4.26 PDTIsajetMap const& HepPID::@63::getPDTIsajetMap () `[static]`

Referenced by translatePDTtoIsajet().

#### 7.3.4.27 PDTPythiaMap const& HepPID::@66::getPDTPythiaMap () `[static]`

Definition at line 628 of file translatePythia.cc.

References PDTPythiaMapInit().

#### 7.3.4.28 PDTPythiaMap const& HepPID::@66::getPDTPythiaMap () `[static]`

Referenced by translatePDTtoPythia().

#### 7.3.4.29 PDTQQMap const& HepPID::@67::getPDTQQMap () `[static]`

Definition at line 543 of file translateQQ.cc.

References PDTQQMapInit().

#### 7.3.4.30 PDTQQMap const& HepPID::@67::getPDTQQMap () `[static]`

Referenced by translatePDTtoQQ().

#### 7.3.4.31 PDTtoPDGMap const& HepPID::@64::getPDTtoPDGMap () `[static]`

Definition at line 378 of file translatePDG.cc.

References PDTtoPDGMapInit().

#### 7.3.4.32 PDTtoPDGMap const& HepPID::@64::getPDTtoPDGMap () `[static]`

Referenced by translatePDTtoPDGtable().

#### 7.3.4.33 PythiaPDTMap const& HepPID::@66::getPythiaPDTMap () `[static]`

Definition at line 619 of file translatePythia.cc.

References PythiaPDTMapInit().

#### 7.3.4.34 PythiaPDTMap const& HepPID::@66::getPythiaPDTMap () `[static]`

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

#### 7.3.4.35 QQbarMap const& HepPID::@67::getQQbarMap () `[static]`

Definition at line 551 of file translateQQ.cc.

References QQbarMapInit().

#### 7.3.4.36 QQbarMap const& HepPID::@67::getQQbarMap () `[static]`

Referenced by InverseQQbarMapInit(), and translateQQbar().

#### 7.3.4.37 QQPDTMap const& HepPID::@67::getQQPDTMap () `[static]`

Definition at line 534 of file translateQQ.cc.

References QQPDTMapInit().

#### 7.3.4.38 QQPDTMap const& HepPID::@67::getQQPDTMap () `[static]`

Referenced by PDTQQMapInit(), and translateQQtoPDT().

#### 7.3.4.39 bool HepPID::hasBottom (const int & *pid*)

does this particle contain a bottom quark?

Definition at line 325 of file ParticleIDMethods.cc.

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

Referenced by main().

#### 7.3.4.40 bool HepPID::hasCharm (const int & *pid*)

does this particle contain a charm quark?

Definition at line 318 of file ParticleIDMethods.cc.

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

Referenced by main().

#### 7.3.4.41 bool HepPID::hasDown (const int & *pid*)

does this particle contain a down quark?

Definition at line 304 of file ParticleIDMethods.cc.

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

Referenced by main().

#### 7.3.4.42 bool HepPID::hasFundamentalAnti (const int & *pid*)

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

Definition at line 139 of file ParticleIDMethods.cc.

References fundamentalID(), and validParticleName().

Referenced by isValid().

#### 7.3.4.43 bool HepPID::hasStrange (const int & *pid*)

does this particle contain a strange quark?

Definition at line 311 of file ParticleIDMethods.cc.

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

Referenced by main().

#### 7.3.4.44 bool HepPID::hasTop (const int & *pid*)

does this particle contain a top quark?

Definition at line 332 of file ParticleIDMethods.cc.

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

Referenced by main().

#### 7.3.4.45 bool HepPID::hasUp (const int & *pid*)

does this particle contain an up quark?

Definition at line 297 of file ParticleIDMethods.cc.

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

Referenced by main().

#### 7.3.4.46 HerwigPDTMap const& HepPID::@62::HerwigPDTMapInit () `[static]`

Definition at line 41 of file translateHerwig.cc.

Referenced by getHerwigPDTMap().

#### 7.3.4.47 InverseQQbarMap const& HepPID::@67::InverseQQbarMapInit () `[static]`

Definition at line 520 of file translateQQ.cc.

References getQQbarMap().

Referenced by getInverseQQbarMap().

#### 7.3.4.48 IsajetPDTMap const& HepPID::@63::IsajetPDTMapInit () `[static]`

Definition at line 44 of file translateIsajet.cc.

Referenced by getIsajetPDTMap().

#### 7.3.4.49 bool HepPID::isBaryon (const int & *pid*)

is this a valid baryon ID?

Definition at line 176 of file ParticleIDMethods.cc.

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

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

#### 7.3.4.50 bool HepPID::isDiQuark (const int & *pid*)

is this a valid diquark ID?

Definition at line 190 of file ParticleIDMethods.cc.

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

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

#### 7.3.4.51 bool HepPID::isHadron (const int & *pid*)

is this a valid hadron ID?

Definition at line 209 of file ParticleIDMethods.cc.

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

Referenced by main().

#### 7.3.4.52 bool HepPID::isLepton (const int & *pid*)

is this a valid lepton ID?

Definition at line 219 of file ParticleIDMethods.cc.

References extraBits(), and fundamentalID().

Referenced by main().

#### 7.3.4.53 bool HepPID::isMeson (const int & *pid*)

is this a valid meson ID?

Definition at line 151 of file ParticleIDMethods.cc.

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

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

#### 7.3.4.54 bool HepPID::isNucleus (const int & *pid*)

is this a valid ion ID?

Definition at line 233 of file ParticleIDMethods.cc.

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

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

#### 7.3.4.55 bool HepPID::isPentaquark (const int & *pid*)

is this a valid pentaquark ID?

Definition at line 247 of file ParticleIDMethods.cc.

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

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

#### 7.3.4.56 bool HepPID::isRhadron (const int & *pid*)

is this a valid R-hadron ID?

Definition at line 279 of file ParticleIDMethods.cc.

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

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

#### 7.3.4.57 bool HepPID::isSUSY (const int & *pid*)

is this a valid SUSY ID?

Definition at line 267 of file ParticleIDMethods.cc.

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

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

#### 7.3.4.58 bool HepPID::isValid (const int & *pid*)

is this a valid ID?

**Examples:**

**examListPythia.cc**.

Definition at line 109 of file ParticleIDMethods.cc.

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

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

#### 7.3.4.59 int HepPID::jSpin (const int & *pid*)

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

Definition at line 342 of file ParticleIDMethods.cc.

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

Referenced by main().

#### 7.3.4.60 int HepPID::lambda (const int & *pid*)

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

Definition at line 96 of file ParticleIDMethods.cc.

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

Referenced by main().

#### 7.3.4.61 void HepPID::listParticleNames (std::ostream & *os*)

list all known names

**Examples:**

**listParticleNames.cc**.

Definition at line 1704 of file ParticleName.cc.

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

Referenced by main().

#### 7.3.4.62 int HepPID::lSpin (const int & *pid*)

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

Definition at line 383 of file ParticleIDMethods.cc.

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

Referenced by main().

#### 7.3.4.63 int HepPID::particleName (const std::string &)

lookup a known ID

Definition at line 1692 of file ParticleName.cc.

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

#### 7.3.4.64 std::string HepPID::particleName (const int &)

get a known **HepPID** (p. 39) Particle name

**Examples:**

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

Definition at line 1682 of file ParticleName.cc.

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

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

#### 7.3.4.65 ParticleNameMap const& HepPID::@59::ParticleNameInit () `[static]`

Definition at line 75 of file ParticleName.cc.

Referenced by getParticleNameMap().

#### 7.3.4.66 PDGtoPDTMap const& HepPID::@64::PDGtoPDTMapInit () `[static]`

Definition at line 41 of file translatePDG.cc.

Referenced by getPDGtoPDTMap().

#### 7.3.4.67 PDTEvtGenMap const& HepPID::@60::PDTEvtGenMapInit () `[static]`

Definition at line 594 of file translateEvtGen.cc.

References getEvtGenPDTMap().

Referenced by getPDTEvtGenMap().

#### 7.3.4.68 PDTHerwigMap const& HepPID::@62::PDTHerwigMapInit () `[static]`

Definition at line 486 of file translateHerwig.cc.

References getHerwigPDTMap().

Referenced by getPDTHerwigMap().

#### 7.3.4.69 PDTIsajetMap const& HepPID::@63::PDTIsajetMapInit () [static]

Definition at line 677 of file translateIsajet.cc.

References getIsajetPDTMap().

Referenced by getPDTIsajetMap().

#### 7.3.4.70 PDTPythiaMap const& HepPID::@66::PDTPythiaMapInit () [static]

Definition at line 605 of file translatePythia.cc.

References getPythiaPDTMap().

Referenced by getPDTPythiaMap().

#### 7.3.4.71 PDTQQMap const& HepPID::@67::PDTQQMapInit () [static]

Definition at line 509 of file translateQQ.cc.

References getQQPDTMap().

Referenced by getPDTQQMap().

#### 7.3.4.72 PDTtoPDGMap const& HepPID::@64::PDTtoPDGMapInit () [static]

Definition at line 355 of file translatePDG.cc.

References getPDGtoPDTMap().

Referenced by getPDTtoPDGMap().

#### 7.3.4.73 PythiaPDTMap const& HepPID::@66::PythiaPDTMapInit () [static]

Definition at line 41 of file translatePythia.cc.

Referenced by getPythiaPDTMap().

#### 7.3.4.74 QQbarMap const& HepPID::@67::QQbarMapInit () [static]

Definition at line 455 of file translateQQ.cc.

Referenced by getQQbarMap().

#### 7.3.4.75 QQPDTMap const& HepPID::@67::QQPDTMapInit () [static]

Definition at line 49 of file translateQQ.cc.

Referenced by getQQPDTMap().

#### 7.3.4.76 int HepPID::sSpin (const int & *pid*)

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

Definition at line 358 of file ParticleIDMethods.cc.

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

Referenced by main().

#### 7.3.4.77 int HepPID::threeCharge (const int & *pid*)

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

Definition at line 432 of file ParticleIDMethods.cc.

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

Referenced by main().

#### 7.3.4.78 int HepPID::translateEvtGentoPDT (const int *evtGenID*)

translate EvtGen to PDG standard

Definition at line 625 of file translateEvtGen.cc.

References getEvtGenPDTMap(), and isValid().

Referenced by HepPDT::addEvtGenParticles().

#### 7.3.4.79 int HepPID::translateGeanttoPDT (const int *geantID*)

translate Geant3 to PDG standard

Definition at line 20 of file translateGeanttoPDT.cc.

References IDMAX.

#### 7.3.4.80 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().

#### 7.3.4.81 int HepPID::translateInverseQQbar (const int *id*)

QQ helper function.

Definition at line 579 of file translateQQ.cc.

References getInverseQQbarMap().

Referenced by writeQQTranslation().

#### 7.3.4.82 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().

#### 7.3.4.83 int HepPID::translatePDGtabletoPDT (const int *pdgID*)

translate PDG table to PDG standard

Definition at line 386 of file translatePDG.cc.

References getPDGtoPDTMap(), and isValid().

#### 7.3.4.84 int HepPID::translatePDTtoEvtGen (const int *pid*)

translate PDG standard to EvtGen

Definition at line 638 of file translateEvtGen.cc.

References getPDTEvtGenMap(), and isValid().

Referenced by writeEvtGenTranslationLine().

#### 7.3.4.85 int HepPID::translatePDTtoGeant (const int *pid*)

translate PDG standard to Geant3

Definition at line 22 of file translatePDTtoGeant.cc.

References abspid(), and IDMAX.

#### 7.3.4.86 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().

#### 7.3.4.87 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().

#### 7.3.4.88 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().

#### 7.3.4.89 int HepPID::translatePDTtoPythia (const int *pid*)

translate PDG standard to Pythia

Definition at line 649 of file translatePythia.cc.

References getPDTPythiaMap(), and isValid().

Referenced by writePythiaTranslationLine().

#### 7.3.4.90 int HepPID::translatePDTtoQQ (const int *pid*)

translate PDG standard to QQ

Definition at line 601 of file translateQQ.cc.

References getPDTQQMap().

Referenced by writeQQTranslation().

#### 7.3.4.91 int HepPID::translatePythiatoPDT (const int *pythiaID*)

translate Pythia to PDG standard

**Examples:**

**examListPythia.cc**.

Definition at line 636 of file translatePythia.cc.

References getPythiaPDTMap(), and isValid().

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

#### 7.3.4.92 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().

#### 7.3.4.93 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().

#### 7.3.4.94 bool HepPID::validParticleName (const std::string &)

verify that this string has a valid id

Definition at line 1673 of file ParticleName.cc.

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

#### 7.3.4.95 bool HepPID::validParticleName (const int &)

verify that this number has a valid name

Definition at line 1663 of file ParticleName.cc.

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

Referenced by hasFundamentalAnti(), and writeParticleNameLine().

#### 7.3.4.96 void HepPID::version ()

print **HepPID** (p. 39) version

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

References versionName().

#### 7.3.4.97 std::string HepPID::versionName ()

return **HepPID** (p. 39) version

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

Referenced by version(), and writeVersion().

#### 7.3.4.98 void HepPID::writeEvtGenTranslation (std::ostream & *os*)

output the translation list

**Examples:**

**listEvtGenTranslation.cc**.

Definition at line 675 of file translateEvtGen.cc.

References writeEvtGenTranslationLine(), and writeVersion().

Referenced by main().

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

Definition at line 651 of file translateEvtGen.cc.

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

Referenced by writeEvtGenTranslation().

#### 7.3.4.100 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().

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

Definition at line 543 of file translateHerwig.cc.

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

Referenced by writeHerwigTranslation().

#### 7.3.4.102 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().

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

Definition at line 930 of file translateIsajet.cc.

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

Referenced by writeIsajetTranslation().

#### 7.3.4.104 void HepPID::@59::writeParticleNameLine (int *i*, std::ostream & *os*) `[static]`

Definition at line 1640 of file ParticleName.cc.

References particleName(), and validParticleName().

Referenced by listParticleNames().

#### 7.3.4.105 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().

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

Definition at line 412 of file translatePDG.cc.

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

Referenced by writePDGTranslation().

#### 7.3.4.107 void HepPID::writePythiaTranslation (std::ostream & *os*)

output the translation list

**Examples:**

**listPythiaTranslation.cc**.

Definition at line 686 of file translatePythia.cc.

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

Referenced by main().

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

Definition at line 662 of file translatePythia.cc.

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

Referenced by writePythiaTranslation().

#### 7.3.4.109 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().

#### 7.3.4.110 void HepPID::writeVersion (std::ostream & *os*)

write **HepPID** (p. 39) 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().

#### 7.3.4.111 int HepPID::Z (const int & *pid*)

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

Definition at line 77 of file ParticleIDMethods.cc.

References abspid(), and isNucleus().

Referenced by main().

## 7.4 std Namespace Reference

# Chapter 8

# HepPDT Class Documentation

## 8.1 HepPDT::Constituent Class Reference

The **Constituent** (p. 65) class has information about constituent particles.

`#include <Constituent.hh>`

### Public Member Functions

- **Constituent** (**ParticleID** p=**ParticleID**(0), int m=-1)
- **Constituent** (**Constituent** const &orig)
- **Constituent** & **operator=** (**Constituent** const &rhs)
- void **swap** (**Constituent** &other)
- int **multiplicity** () const

  *how many of this constituent are there?*

- **ParticleID pid** () const

  ***ParticleID** (p. 96) 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

The **Constituent** (p. 65) class has information about constituent particles.

**Author:**

Lynn Garren

Holds a particle constituent (e.g. quark type and number of quarks of this type)

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

Definition at line 38 of file Constituent.hh.

References swap().

#### 8.1.3.2 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().

#### 8.1.3.3 int HepPDT::Constituent::multiplicity () const `[inline]`

how many of this constituent are there?

Definition at line 50 of file Constituent.hh.

#### 8.1.3.4 ParticleID HepPDT::Constituent::pid () const `[inline]`

**ParticleID** (p. 96) of this constituent.

Definition at line 52 of file Constituent.hh.

#### 8.1.3.5 bool HepPDT::Constituent::isUp () const

is this an up quark?

Definition at line 12 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

#### 8.1.3.6 bool HepPDT::Constituent::isDown () const

is this a down quark?

Definition at line 18 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

#### 8.1.3.7 bool HepPDT::Constituent::isStrange () const

is this a strange quark?

Definition at line 24 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

#### 8.1.3.8 bool HepPDT::Constituent::isCharm () const

is this a charm quark?

Definition at line 30 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

#### 8.1.3.9 bool HepPDT::Constituent::isBottom () const

is this a bottom quark?

Definition at line 36 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

#### 8.1.3.10 bool HepPDT::Constituent::isTop () const

is this a top quark?

Definition at line 42 of file Constituent.cc.

References HepPDT::ParticleID::fundamentalID().

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

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

## 8.2 HepPDT::DefTable Class Reference

The **DefTable** (p. 68) class holds EvtGen definitions.

`#include <DefTable.hh>`

### Public Types

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

### Public Member Functions

- **DefTable** ()
- ∼**DefTable** ()
- void **addDefinition** (std::string const &def, double val)

  *add a definition to the map*
- bool **hasDefinition** (std::string const &def) const

  *is this definition already defined?*
- int **size** () const

  *get the size of the definition map*
- double **definition** (std::string const &def)

  *return the definition of this parameter*
- void **writeDefinitions** () const

  *use for diagnostics*
- **iterator begin** ()

  *begin iterating over the definition map*
- **const_iterator begin** () const

  *begin iterating over the definition map*
- **iterator end** ()

  *end iterating over the definition map*
- **const_iterator end** () const

  *end iterating over the definition map*

### 8.2.1 Detailed Description

The **DefTable** (p. 68) class holds EvtGen definitions.

**Author:**

Lynn Garren

This is temporary information storage used when reading EvtGen input.

Definition at line 23 of file DefTable.hh.

## 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 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().

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

get the size of the definition map

Definition at line 46 of file DefTable.hh.

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

use for diagnostics

Definition at line 41 of file DefTable.cc.

References begin(), and end().

#### 8.2.4.6 iterator HepPDT::DefTable::begin () `[inline]`

begin iterating over the definition map

Definition at line 53 of file DefTable.hh.

Referenced by writeDefinitions().

#### 8.2.4.7 const_iterator HepPDT::DefTable::begin () const `[inline]`

begin iterating over the definition map

Definition at line 55 of file DefTable.hh.

#### 8.2.4.8 iterator HepPDT::DefTable::end () `[inline]`

end iterating over the definition map

Definition at line 58 of file DefTable.hh.

Referenced by writeDefinitions().

#### 8.2.4.9 const_iterator HepPDT::DefTable::end () const `[inline]`

end iterating over the definition map

Definition at line 60 of file DefTable.hh.

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

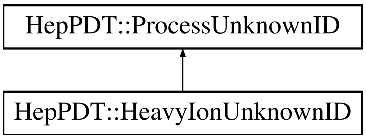
- **DefTable.hh**
- **DefTable.cc**

## 8.3 HepPDT::HeavyIonUnknownID Class Reference

The **HeavyIonUnknownID** (p. 71) class inherits from **ProcessUnknownID** (p. 110).

`#include <HeavyIonUnknownID.hh>`

Inheritance diagram for HepPDT::HeavyIonUnknownID::



### Public Member Functions

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

### 8.3.1 Detailed Description

The **HeavyIonUnknownID** (p. 71) class inherits from **ProcessUnknownID** (p. 110).

**Author:**

Lynn Garren

Create and return a pointer to a **ParticleData** (p. 76) object if invoked by a valid Heavy Ion. HeavyIonUnknownID::processUnknownID( ParticleID ) calculates both charge and an approximate mass for the particle.

**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. 110).

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

The **Measurement** (p. 73) class defines a value with its error.

`#include <Measurement.hh>`

## Public Member Functions

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

  *compare the value, ignore the error*

- bool **operator==** (**Measurement** const &other) const

  *compare the value, ignore the error*

- double **value** () const
- double **sigma** () const
- **operator double** () const

## 8.4.1 Detailed Description

The **Measurement** (p. 73) class defines a value with its error.

**Author:**

Mark Fischler

Contain a value with associated errors. Provide simple comparisons.

**Examples:**

**examMyPDT.cc**.

Definition at line 20 of file Measurement.hh.

## 8.4.2 Constructor & Destructor Documentation

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

Definition at line 33 of file Measurement.icc.

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

Definition at line 30 of file Measurement.icc.

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

Definition at line 35 of file Measurement.icc.

### 8.4.3 Member Function Documentation

#### 8.4.3.1 void HepPDT::Measurement::swap (Measurement & *other*) `[inline]`

Definition at line 38 of file Measurement.icc.

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

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

#### 8.4.3.2 Measurement & HepPDT::Measurement::operator= (Measurement const & *rhs*) `[inline]`

Definition at line 44 of file Measurement.icc.

References swap().

#### 8.4.3.3 bool HepPDT::Measurement::operator< (Measurement const & *other*) const `[inline]`

compare the value, ignore the error

Definition at line 51 of file Measurement.icc.

References val.

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

compare the value, ignore the error

Definition at line 56 of file Measurement.icc.

References val.

#### 8.4.3.5 double HepPDT::Measurement::value () const `[inline]`

Definition at line 61 of file Measurement.icc.

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

#### 8.4.3.6 double HepPDT::Measurement::sigma () const `[inline]`

Definition at line 62 of file Measurement.icc.

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

#### 8.4.3.7 HepPDT::Measurement::operator double () const `[inline]`

Definition at line 64 of file Measurement.icc.

References value().

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

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

## 8.5 HepPDT::ParticleData Class Reference

The **ParticleData** (p. 76) class holds data for a single particle in the table.

`#include <ParticleData.hh>`

### Public Member Functions

- **ParticleData** (const **TempParticleData** &tpd)

  *create **ParticleData** (p. 76) 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 **PDTname** () const

  *return PDG particle name*

- const std::string & **source** () const

  *the name of the input source*

- **ParticleID ID** () const

  *return the **ParticleID** (p. 96)*

- 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. *96) 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
- 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 **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 &lt)

  *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

The **ParticleData** (p. 76) class holds data for a single particle in the table.

**Author:**

Lynn Garren

This class holds the basic paritcle data - name, ID, spin, etc. Methods are provided to get various information about the particle. All the particle information is accessed via this class.

**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*) `[inline]`

create **ParticleData** (p. 76) from the temporary information

Definition at line 12 of file ParticleData.icc.

### 8.5.2.2 HepPDT::ParticleData::∼ParticleData () `[inline]`

Definition at line 26 of file ParticleData.icc.

### 8.5.2.3 HepPDT::ParticleData::ParticleData (const ParticleData & *orig*) `[inline]`

Definition at line 42 of file ParticleData.icc.

## 8.5.3 Member Function Documentation

### 8.5.3.1 void HepPDT::ParticleData::swap (ParticleData & *rhs*) `[inline]`

Definition at line 29 of file ParticleData.icc.

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

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

### 8.5.3.2 ParticleData & HepPDT::ParticleData::operator= (const ParticleData & *rhs*) `[inline]`

Definition at line 54 of file ParticleData.icc.

References swap().

#### 8.5.3.3 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().

#### 8.5.3.4 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().

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

the name of the input source

Definition at line 55 of file ParticleData.hh.

#### 8.5.3.6 ParticleID HepPDT::ParticleData::ID () const [inline]

return the **ParticleID** (p. 96)

Definition at line 57 of file ParticleData.hh.

#### 8.5.3.7 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().

#### 8.5.3.8 int HepPDT::ParticleData::originalID () const [inline]

return untranslated integer ID

Definition at line 61 of file ParticleData.hh.

#### 8.5.3.9 double HepPDT::ParticleData::charge () const `[inline]`

return charge

Definition at line 63 of file ParticleData.hh.

#### 8.5.3.10 double HepPDT::ParticleData::color () const `[inline]`

color information

Definition at line 65 of file ParticleData.hh.

#### 8.5.3.11 SpinState HepPDT::ParticleData::spin () const `[inline]`

spin information

Definition at line 67 of file ParticleData.hh.

Referenced by setSpin().

#### 8.5.3.12 Measurement HepPDT::ParticleData::mass () const `[inline]`

mass

Definition at line 69 of file ParticleData.hh.

References HepPDT::ResonanceStructure::mass().

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

#### 8.5.3.13 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 isStable(), and pdtSimpleTest().

#### 8.5.3.14 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().

#### 8.5.3.15 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().

#### 8.5.3.16 Measurement HepPDT::ParticleData::lifetime () const `[inline]`

calculate the lifetime

Definition at line 77 of file ParticleData.hh.

References HepPDT::ResonanceStructure::lifetime().

Referenced by isStable(), and write().

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

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

Definition at line 79 of file ParticleData.hh.

#### 8.5.3.18 Constituent HepPDT::ParticleData::constituent (unsigned int *i*) const `[inline]`

constituent information

return this constituent if it exists

Definition at line 98 of file ParticleData.icc.

#### 8.5.3.19 ParticleID HepPDT::ParticleData::constituentParticle (unsigned int *i*) const `[inline]`

**ParticleID** (p. 96) for a constituent particle.

Definition at line 109 of file ParticleData.icc.

#### 8.5.3.20 ResonanceStructure const HepPDT::ParticleData::resonance () const `[inline]`

resonance (width) information

Definition at line 85 of file ParticleData.hh.

#### 8.5.3.21 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(), main(), pdtFragmentTest(), pdtSimpleTest(), and writeParticleInfo().

### 8.5.3.22 void HepPDT::ParticleData::writeParticleInfo (std::ostream & *os*) const

output all information about a particle EXCEPT its decays This method is redundant with **write()** (p. 82), but retained for backwards compatibility.

Definition at line 68 of file write.cc.

References write().

### 8.5.3.23 void HepPDT::ParticleData::writeParticleTranslation (std::ostream & *os*) const

output the translation information for this particle

Definition at line 73 of file write.cc.

References PDTname(), and pid().

### 8.5.3.24 bool HepPDT::ParticleData::isMeson () const [inline]

is this a valid meson?

Definition at line 98 of file ParticleData.hh.

References HepPDT::ParticleID::isMeson().

### 8.5.3.25 bool HepPDT::ParticleData::isBaryon () const [inline]

is this a valid baryon?

Definition at line 100 of file ParticleData.hh.

References HepPDT::ParticleID::isBaryon().

### 8.5.3.26 bool HepPDT::ParticleData::isDiQuark () const [inline]

is this a valid diquark?

Definition at line 102 of file ParticleData.hh.

References HepPDT::ParticleID::isDiQuark().

#### 8.5.3.27 bool HepPDT::ParticleData::isHadron () const [inline]

is this a valid hadron?

Definition at line 104 of file ParticleData.hh.

References HepPDT::ParticleID::isHadron().

#### 8.5.3.28 bool HepPDT::ParticleData::isLepton () const [inline]

is this a valid lepton?

Definition at line 106 of file ParticleData.hh.

References HepPDT::ParticleID::isLepton().

#### 8.5.3.29 bool HepPDT::ParticleData::isNucleus () const [inline]

is this a valid ion?

Definition at line 108 of file ParticleData.hh.

References HepPDT::ParticleID::isNucleus().

#### 8.5.3.30 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().

#### 8.5.3.31 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().

#### 8.5.3.32 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().

#### 8.5.3.33 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().

#### 8.5.3.34 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().

#### 8.5.3.35 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().

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

is this particle allowed to decay?

Definition at line 61 of file ParticleData.icc.

References lifetime(), and totalWidth().

#### 8.5.3.37 bool HepPDT::ParticleData::operator< (const ParticleData & *other*) const `[inline]`

compare masses

Definition at line 68 of file ParticleData.icc.

References mass().

#### 8.5.3.38 bool HepPDT::ParticleData::operator== (const ParticleData & *other*) const `[inline]`

use PID and ignore everything else

Definition at line 73 of file ParticleData.icc.

References itsID.

#### 8.5.3.39 void HepPDT::ParticleData::setCharge (double *chg*) `[inline]`

change the charge

Definition at line 133 of file ParticleData.hh.

#### 8.5.3.40 void HepPDT::ParticleData::setColor (double *col*) `[inline]`

change color information

Definition at line 135 of file ParticleData.hh.

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

change spin information

Definition at line 137 of file ParticleData.hh.

References spin().

#### 8.5.3.42 void HepPDT::ParticleData::addConstituent (Constituent *c*) `[inline]`

add a constituent particle

Definition at line 139 of file ParticleData.hh.

#### 8.5.3.43 void HepPDT::ParticleData::setMass (Measurement const & *mass*) `[inline]`

change the mass

Definition at line 141 of file ParticleData.hh.

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

#### 8.5.3.44 void HepPDT::ParticleData::setTotalWidth (Measurement const & *width*) `[inline]`

change the total width

Definition at line 143 of file ParticleData.hh.

References HepPDT::ResonanceStructure::setTotalWidth().

#### 8.5.3.45 void HepPDT::ParticleData::setTotalWidthFromLifetime (Measurement const & *lt*) `[inline]`

change the total width using a lifetime

Definition at line 145 of file ParticleData.hh.

References HepPDT::ResonanceStructure::setTotalWidthFromLifetime().

#### 8.5.3.46 void HepPDT::ParticleData::setLowerCutoff (double *cut*) `[inline]`

change the total width lower cutoff

Definition at line 147 of file ParticleData.hh.

References HepPDT::ResonanceStructure::setLowerCutoff().

#### 8.5.3.47 void HepPDT::ParticleData::setUpperCutoff (double *cut*) `[inline]`

change the total width upper cutoff

Definition at line 149 of file ParticleData.hh.

References HepPDT::ResonanceStructure::setUpperCutoff().

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

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

## 8.6 HepPDT::ParticleDataTable Class Reference

The **ParticleDataTable** (p. 88) class is the core of **HepPDT** (p. 25).

`#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**)
- ∼**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. *96)*

- **ParticleData** * **particle** (**ParticleID**)

  *access particle information via **ParticleID** (*p. *96)*

- **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. *96)*

- **ParticleData** const ∗ **operator[ ]** (**ParticleID**) const

  *access particle information via* ***ParticleID*** *(*p. *96)*

- **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. *122) destructor to fill the PDT*

### 8.6.1 Detailed Description

The **ParticleDataTable** (p. 88) class is the core of **HepPDT** (p. 25).

**Author:**

Lynn Garren, Walter Brown

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

**Examples:**

**examMyPDT.cc**, **listEvtGenNames.cc.in**, **listPDGNames.cc.in**, **listPythiaNames.cc.in**, **testHepPDT.cc**, **testReadEvtGen.cc.in**, **testReadIsajet.cc.in**, **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 ParticleData HepPDT::ParticleDataTable::CPD

Definition at line 47 of file ParticleDataTable.hh.

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

Definition at line 49 of file ParticleDataTable.hh.

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

Definition at line 50 of file ParticleDataTable.hh.

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

Definition at line 51 of file ParticleDataTable.hh.

### 8.6.2.5 typedef PDTMap::const_iterator HepPDT::ParticleDataTable::const_iterator

Definition at line 53 of file ParticleDataTable.hh.

### 8.6.2.6 typedef PDTNameMap::const_iterator HepPDT::ParticleDataTable::const_iteratorByName

Definition at line 54 of file ParticleDataTable.hh.

## 8.6.3 Constructor & Destructor Documentation

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

Construct a particle data table with an identifying name. 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 () [inline]

Definition at line 10 of file ParticleDataTable.icc.

## 8.6.4 Member Function Documentation

### 8.6.4.1 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().

#### 8.6.4.2 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().

#### 8.6.4.3 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().

#### 8.6.4.4 int HepPDT::ParticleDataTable::sizeNameMap () const [inline]

size of the map of particle names

Definition at line 72 of file ParticleDataTable.hh.

#### 8.6.4.5 const_iteratorByName HepPDT::ParticleDataTable::beginNameMap () const [inline]

begin iterating over the map of particle names

Definition at line 74 of file ParticleDataTable.hh.

#### 8.6.4.6 const_iteratorByName HepPDT::ParticleDataTable::endNameMap () const [inline]

end iterating over the map of particle names

Definition at line 76 of file ParticleDataTable.hh.

#### 8.6.4.7 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().

#### 8.6.4.8 ParticleData const ∗ HepPDT::ParticleDataTable::particle (ParticleID) const

access particle information via **ParticleID** (p. 96)

**Examples:**

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

Definition at line 153 of file ParticleDataTable.cc.

Referenced by duplicateFragmentTest(), main(), operator[ ](), particle(), pdtFragmentTest(), pdtSimpleTest(), HepPDT::TestNuclearFragment::processUnknownID(), HepPDT::HeavyIonUnknownID::processUnknownID(), and testPDMethods().

#### 8.6.4.9 ParticleData ∗ HepPDT::ParticleDataTable::particle (ParticleID)

access particle information via **ParticleID** (p. 96)

Definition at line 141 of file ParticleDataTable.cc.

#### 8.6.4.10 ParticleData const ∗ HepPDT::ParticleDataTable::particle (std::string) const

access particle information via a particle name

Definition at line 187 of file ParticleDataTable.cc.

References particle().

#### 8.6.4.11 ParticleData ∗ HepPDT::ParticleDataTable::particle (std::string)

access particle information via a particle name

Definition at line 176 of file ParticleDataTable.cc.

References particle().

#### 8.6.4.12 ParticleData ∗ HepPDT::ParticleDataTable::operator[ ] (ParticleID) `[inline]`

access particle information via **ParticleID** (p. 96)

Definition at line 13 of file ParticleDataTable.icc.

References particle().

#### 8.6.4.13 ParticleData const ∗ HepPDT::ParticleDataTable::operator[ ] (ParticleID) const `[inline]`

access particle information via **ParticleID** (p. 96)

Definition at line 18 of file ParticleDataTable.icc.

References particle().

#### 8.6.4.14 ParticleData ∗ HepPDT::ParticleDataTable::operator[ ] (std::string) `[inline]`

access particle information via a particle name

Definition at line 23 of file ParticleDataTable.icc.

References particle().

#### 8.6.4.15 ParticleData const ∗ HepPDT::ParticleDataTable::operator[ ] (std::string) const `[inline]`

access particle information via a particle name

Definition at line 28 of file ParticleDataTable.icc.

References particle().

#### 8.6.4.16 void HepPDT::ParticleDataTable::writeParticleData (std::ostream & *outstr*)

output all information in the PDT

**Examples:**

> **examMyPDT.cc**, **testHepPDT.cc**, **testReadEvtGen.cc.in**, **testReadIsajet.cc.in**, **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().

#### 8.6.4.17 void HepPDT::ParticleDataTable::writeParticleInfo (std::ostream & *outstr*)

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 main(), and pdtSimpleTest().

#### 8.6.4.18 void HepPDT::ParticleDataTable::writeParticleTranslation (std::ostream & *outstr*)

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().

Referenced by main().

#### 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(), and main().

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

used by the **TableBuilder** (p. 122) destructor to fill the PDT

Definition at line 17 of file convertTemporaryMap.cc.

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

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

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

## 8.7 HepPDT::ParticleDataTableComparison Class Reference

The **ParticleDataTableComparison** (p. 95) class provides a utility for sorting the PDT.

`#include <ParticleDataTableComparison.hh>`

### Public Member Functions

- **ParticleDataTableComparison** ()

  *stateless class*

- bool **operator()** (const **ParticleID** &, const **ParticleID** &) const

### 8.7.1 Detailed Description

The **ParticleDataTableComparison** (p. 95) class provides a utility for sorting the PDT.

**Author:**

Lynn Garren

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

Definition at line 23 of file ParticleDataTableComparison.hh.

### 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]`

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

Definition at line 36 of file ParticleDataTableComparison.hh.

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

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

- **ParticleDataTableComparison.hh**

## 8.8 HepPDT::ParticleID Class Reference

The **ParticleID** (p. 96) has various utilities to extract information from the particle ID.

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

### Public Member Functions

- **ParticleID** (int pid=0)

  *create from an integer ID*

- **ParticleID** (const **ParticleID** &orig)
- **ParticleID** & **operator=** (const **ParticleID** &)
- void **swap** (**ParticleID** &other)
- bool **operator<** (**ParticleID** const &other) const
- bool **operator==** (**ParticleID** const &other) const
- int **pid** () const

  *get the integer ID*

- int **abspid** () const

  *get the absolute value*

- bool **isValid** () const

  *is this a valid ID?*

- bool **isMeson** () const

  *is this a valid meson ID?*

- bool **isBaryon** () const

  *is this a valid baryon ID?*

- bool **isDiQuark** () const

  *is this a valid diquark ID?*

- bool **isHadron** () const

  *is this a valid hadron ID?*

- bool **isLepton** () const

  *is this a valid lepton ID?*

- bool **isNucleus** () const

  *is this a valid ion ID?*

- bool **isPentaquark** () const

  *is this a valid pentaquark ID?*

- bool **isSUSY** () const

  *is this a valid SUSY ID?*

- bool **isRhadron** () const

*is this a valid R-hadron ID?*

- bool **hasUp** () const

  *does this particle contain an up quark?*

- bool **hasDown** () const

  *does this particle contain a down quark?*

- bool **hasStrange** () const

  *does this particle contain a strange quark?*

- bool **hasCharm** () const

  *does this particle contain a charm quark?*

- bool **hasBottom** () const

  *does this particle contain a bottom quark?*

- bool **hasTop** () const

  *does this particle contain a top quark?*

- int **jSpin** () const

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

- int **sSpin** () const

  *sSpin returns 2S+1, where S is the spin*

- int **lSpin** () const

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

- int **fundamentalID** () const

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

- int **extraBits** () const
- **Quarks quarks** () const

  *returns a list of 3 constituent quarks*

- int **threeCharge** () const
- int **A** () const

  *if this is a nucleus (ion), get A*

- int **Z** () const

  *if this is a nucleus (ion), get Z*

- int **lambda** () const

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

- unsigned short **digit** (**location**) const

  *return the digit at a named location in the PID*

- const std::string **PDTname** () const

  *standard particle name*

### 8.8.1 Detailed Description

The **ParticleID** (p. 96) has various utilities to extract information from the particle ID.

**Author:**

Lynn Garren

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

**Examples:**

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

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 16 of file ParticleID.cc.

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

Definition at line 20 of file ParticleID.cc.

### 8.8.3 Member Function Documentation

#### 8.8.3.1 ParticleID & HepPDT::ParticleID::operator= (const ParticleID &)

Definition at line 24 of file ParticleID.cc.

References swap().

#### 8.8.3.2 void HepPDT::ParticleID::swap (ParticleID & *other*)

Definition at line 31 of file ParticleID.cc.

References itsPID, and HepPDT::swap().

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

#### 8.8.3.3 bool HepPDT::ParticleID::operator< (ParticleID const & *other*) const

Definition at line 36 of file ParticleID.cc.

References itsPID.

#### 8.8.3.4 bool HepPDT::ParticleID::operator== (ParticleID const & *other*) const

Definition at line 41 of file ParticleID.cc.

References itsPID.

#### 8.8.3.5 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(), HepPDT::TableBuilder::getAntiParticle(), 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().

#### 8.8.3.6 int HepPDT::ParticleID::abspid () const

get the absolute value

return a value greater than or equal to zero

Definition at line 46 of file ParticleID.cc.

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

#### 8.8.3.7 bool HepPDT::ParticleID::isValid () const

is this a valid ID?

**Examples:**

**testPID.cc**.

Definition at line 137 of file ParticleID.cc.

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

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

#### 8.8.3.8 bool HepPDT::ParticleID::isMeson () const

is this a valid meson ID?

**Examples:**

**testPID.cc**.

Definition at line 256 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().

#### 8.8.3.9 bool HepPDT::ParticleID::isBaryon () const

is this a valid baryon 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::isBaryon(), isHadron(), isValid(), and testHadron().

#### 8.8.3.10 bool HepPDT::ParticleID::isDiQuark () const

is this a valid diquark ID?

**Examples:**

**testPID.cc**.

Definition at line 298 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().

#### 8.8.3.11 bool HepPDT::ParticleID::isHadron () const

is this a valid hadron ID?

**Examples:**

**testPID.cc**.

Definition at line 288 of file ParticleID.cc.

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

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

#### 8.8.3.12 bool HepPDT::ParticleID::isLepton () const

is this a valid lepton ID?

**Examples:**

**testPID.cc**.

Definition at line 280 of file ParticleID.cc.

References extraBits(), and fundamentalID().

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

#### 8.8.3.13 bool HepPDT::ParticleID::isNucleus () const

is this a valid ion ID?

**Examples:**

**testPID.cc**.

Definition at line 216 of file ParticleID.cc.

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

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

#### 8.8.3.14 bool HepPDT::ParticleID::isPentaquark () const

is this a valid pentaquark ID?

Definition at line 162 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(), and isValid().

#### 8.8.3.15 bool HepPDT::ParticleID::isSUSY () const

is this a valid SUSY ID?

Definition at line 199 of file ParticleID.cc.

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

Referenced by isRhadron(), and isValid().

#### 8.8.3.16 bool HepPDT::ParticleID::isRhadron () const

is this a valid R-hadron ID?

Definition at line 182 of file ParticleID.cc.

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

Referenced by isValid().

#### 8.8.3.17 bool HepPDT::ParticleID::hasUp () const

does this particle contain an up quark?

**Examples:**

**testPID.cc**.

Definition at line 82 of file ParticleID.cc.

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

Referenced by testHasMethods().

#### 8.8.3.18 bool HepPDT::ParticleID::hasDown () const

does this particle contain a down quark?

**Examples:**

**testPID.cc**.

Definition at line 91 of file ParticleID.cc.

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

Referenced by testHasMethods().

#### 8.8.3.19 bool HepPDT::ParticleID::hasStrange () const

does this particle contain a strange quark?

**Examples:**

**testPID.cc**.

Definition at line 100 of file ParticleID.cc.

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

Referenced by testHasMethods().

#### 8.8.3.20 bool HepPDT::ParticleID::hasCharm () const

does this particle contain a charm quark?

**Examples:**

**testPID.cc**.

Definition at line 109 of file ParticleID.cc.

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

Referenced by testHasMethods().

### 8.8.3.21 bool HepPDT::ParticleID::hasBottom () const

does this particle contain a bottom quark?

**Examples:**

**testPID.cc**.

Definition at line 118 of file ParticleID.cc.

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

Referenced by testHasMethods().

### 8.8.3.22 bool HepPDT::ParticleID::hasTop () const

does this particle contain a top quark?

**Examples:**

**testPID.cc**.

Definition at line 127 of file ParticleID.cc.

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

Referenced by testHasMethods().

### 8.8.3.23 int HepPDT::ParticleID::jSpin () const

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

**Examples:**

**testPID.cc**.

Definition at line 327 of file ParticleID.cc.

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

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

#### 8.8.3.24 int HepPDT::ParticleID::sSpin () const

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

**Examples:**

**testPID.cc**.

Definition at line 394 of file ParticleID.cc.

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

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

#### 8.8.3.25 int HepPDT::ParticleID::lSpin () const

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

**Examples:**

**testPID.cc**.

Definition at line 344 of file ParticleID.cc.

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

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

#### 8.8.3.26 int HepPDT::ParticleID::fundamentalID () const

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

**Examples:**

**testPID.cc**.

Definition at line 70 of file ParticleID.cc.

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

Referenced by hasBottom(), hasCharm(), hasDown(), hasStrange(), hasTop(), hasUp(), isBaryon(), HepPDT::Constituent::isBottom(), HepPDT::Constituent::isCharm(), isDiQuark(), HepPDT::Constituent::isDown(), isLepton(), isMeson(), HepPDT::Constituent::isStrange(), isSUSY(), HepPDT::Constituent::isTop(), HepPDT::Constituent::isUp(), isValid(), jSpin(), quarks(), testDiQuark(), testHadron(), testLepton(), testNucleus(), testUnknown(), and threeCharge().

#### 8.8.3.27 int HepPDT::ParticleID::extraBits () const

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

**Examples:**

**testPID.cc**.

Definition at line 62 of file ParticleID.cc.

References abspid().

Referenced by hasBottom(), hasCharm(), hasDown(), hasStrange(), hasTop(), hasUp(), isBaryon(), isDiQuark(), isHadron(), isLepton(), isMeson(), isPentaquark(), isRhadron(), isSUSY(), isValid(), jSpin(), quarks(), testDiQuark(), testHadron(), testLepton(), testNucleus(), testUnknown(), and threeCharge().

#### 8.8.3.28 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(), HepPDT::nq1, HepPDT::Quarks::nq1, HepPDT::nq2, HepPDT::Quarks::nq2, HepPDT::nq3, and HepPDT::Quarks::nq3.

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

#### 8.8.3.29 int HepPDT::ParticleID::threeCharge () const

this is mostly for use by functions like addPDGParticles that have to figure out the charge from the PID

**Examples:**

**testPID.cc**.

Definition at line 419 of file ParticleID.cc.

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

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

#### 8.8.3.30 int HepPDT::ParticleID::A () const

if this is a nucleus (ion), get A

**Examples:**

**testPID.cc**.

Definition at line 229 of file ParticleID.cc.

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

Referenced by isNucleus(), and testNucleus().

#### 8.8.3.31 int HepPDT::ParticleID::Z () const

if this is a nucleus (ion), get Z

**Examples:**

**testPID.cc**.

Definition at line 238 of file ParticleID.cc.

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

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

#### 8.8.3.32 int HepPDT::ParticleID::lambda () const

if this is a nucleus (ion), get nLambda

**Examples:**

**testPID.cc**.

Definition at line 247 of file ParticleID.cc.

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

Referenced by testNucleus().

#### 8.8.3.33 unsigned short HepPDT::ParticleID::digit (location) const

return the digit at a named location in the PID

**Examples:**

**testPID.cc**.

Definition at line 53 of file ParticleID.cc.

References abspid().

Referenced by A(), fundamentalID(), hasBottom(), hasCharm(), hasDown(), hasStrange(), hasTop(), hasUp(), isBaryon(), isDiQuark(), isMeson(), isNucleus(), isPentaquark(), isRhadron(), isSUSY(), lambda(), quarks(), testDiQuark(), testHadron(), testLepton(), testNucleus(), testUnknown(), threeCharge(), and Z().

#### 8.8.3.34 const std::string HepPDT::ParticleID::PDTname () const `[inline]`

standard particle name

**Examples:**

**testPID.cc**.

Definition at line 150 of file ParticleID.hh.

References HepPID::particleName().

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

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

### Public Member Functions

- **ParticleNameMap** (**PartcleIdMap** m1, **ParticleLookupMap** m2)
- ∼**ParticleNameMap** ()
- **PartcleIdMap 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

Used internally to store the static maps

Definition at line 42 of file ParticleName.cc.

### 8.9.2 Member Typedef Documentation

#### 8.9.2.1 typedef PartcleIdMap::const_iterator HepPID::ParticleNameMap::idIterator

Definition at line 46 of file ParticleName.cc.

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

Definition at line 47 of file ParticleName.cc.

### 8.9.3 Constructor & Destructor Documentation

#### 8.9.3.1 HepPID::ParticleNameMap::ParticleNameMap (PartcleIdMap *m1*, ParticleLookupMap *m2*) `[inline]`

Definition at line 49 of file ParticleName.cc.

#### 8.9.3.2 HepPID::ParticleNameMap::∼ParticleNameMap () `[inline]`

Definition at line 51 of file ParticleName.cc.

## 8.9.4 Member Function Documentation

### 8.9.4.1 PartcleIdMap HepPID::ParticleNameMap::nameMap () const [inline]

Definition at line 53 of file ParticleName.cc.

### 8.9.4.2 ParticleLookupMap HepPID::ParticleNameMap::lookupMap () const [inline]

Definition at line 54 of file ParticleName.cc.

### 8.9.4.3 idIterator HepPID::ParticleNameMap::begin () const [inline]

Definition at line 55 of file ParticleName.cc.

### 8.9.4.4 idIterator HepPID::ParticleNameMap::end () const [inline]

Definition at line 56 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 57 of file ParticleName.cc.

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

### 8.9.4.6 nameIterator HepPID::ParticleNameMap::beginLookupMap () const [inline]

Definition at line 58 of file ParticleName.cc.

### 8.9.4.7 nameIterator HepPID::ParticleNameMap::endLookupMap () const [inline]

Definition at line 59 of file ParticleName.cc.

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

### 8.9.4.8 nameIterator HepPID::ParticleNameMap::findString (const std::string & *s*) const [inline]

Definition at line 60 of file ParticleName.cc.

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

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

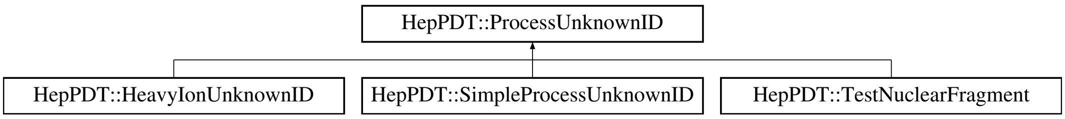
- **ParticleName.cc**

## 8.10 HepPDT::ProcessUnknownID Class Reference

The **ProcessUnknownID** (p. 110) class is abstract.

`#include <ProcessUnknownID.hh>`

Inheritance diagram for HepPDT::ProcessUnknownID::


HepPDT::ProcessUnknownID
HepPDT::HeavyIonUnknownID
HepPDT::SimpleProcessUnknownID
HepPDT::TestNuclearFragment

### Public Member Functions

- **ParticleData** ∗ **callProcessUnknownID** (**ParticleID**, const **ParticleDataTable** &)

  *safety wrapper to avoid secondary calls to processUnknownID*

### Protected Member Functions

- **ProcessUnknownID** ()
- virtual ∼**ProcessUnknownID** ()

### 8.10.1 Detailed Description

The **ProcessUnknownID** (p. 110) class is abstract.

**Author:**

Lynn Garren

This is an abstract class which allows you to define your own methods for handling undefined particle lookups.

Definition at line 28 of file ProcessUnknownID.hh.

### 8.10.2 Constructor & Destructor Documentation

#### 8.10.2.1 HepPDT::ProcessUnknownID::ProcessUnknownID () `[inline, protected]`

Definition at line 36 of file ProcessUnknownID.hh.

#### 8.10.2.2 virtual HepPDT::ProcessUnknownID::∼ProcessUnknownID () `[inline, protected, virtual]`

Definition at line 37 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

The **ResonanceStructure** (p. 114) class is holds mass and width information.

`#include <ResonanceStructure.hh>`

### Public Member Functions

- **ResonanceStructure** (**Measurement** mass=**Measurement**(), **Measurement** width=**Measurement**(), double min=0., double max=0.)

  *construct from mass and total width*

- virtual ~**ResonanceStructure** ()
- **ResonanceStructure** (const **ResonanceStructure** &orig)
- **ResonanceStructure** & **operator=** (const **ResonanceStructure** &rhs)
- void **swap** (**ResonanceStructure** &other)
- **Measurement** const & **mass** () const

  *get the mass*

- **Measurement** const & **totalWidth** () const

  *get the total width*

- **Measurement** **lifetime** () const

  *calculate lifetime from total width*

- double **lowerCutoff** () const

  *lower cutoff of allowed width values*

- double **upperCutoff** () const

  *upper cutoff of allowed width values*

- void **setMass** (**Measurement** const &mass)

  *change the mass*

- void **setTotalWidth** (**Measurement** const &width)

  *change the total width*

- void **setTotalWidthFromLifetime** (**Measurement** const &lt)

  *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

The **ResonanceStructure** (p. 114) class is holds mass and width information.

**Author:**

Lynn Garren

**ResonanceStructure** (p. 114) contains the minimum information for a Breit-Wigner distribution about a given mass.

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

Definition at line 28 of file ResonanceStructure.cc.

References swap().

### 8.12.3.2 void HepPDT::ResonanceStructure::swap (ResonanceStructure & *other*)

Definition at line 35 of file ResonanceStructure.cc.

References itsLowerCutoff, itsMass, itsTotalWidth, itsUpperCutoff, HepPDT::swap(), and HepPDT::Measurement::swap().

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

### 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 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().

#### 8.12.3.5 Measurement HepPDT::ResonanceStructure::lifetime () const

calculate lifetime from total width

Definition at line 13 of file lifetime.cc.

References HepPDT::Measurement::sigma(), and HepPDT::Measurement::value().

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

#### 8.12.3.6 double HepPDT::ResonanceStructure::lowerCutoff () const `[inline]`

lower cutoff of allowed width values

Definition at line 54 of file ResonanceStructure.hh.

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

#### 8.12.3.7 double HepPDT::ResonanceStructure::upperCutoff () const `[inline]`

upper cutoff of allowed width values

Definition at line 56 of file ResonanceStructure.hh.

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

#### 8.12.3.8 void HepPDT::ResonanceStructure::setMass (Measurement const & *mass*) `[inline]`

change the mass

Definition at line 61 of file ResonanceStructure.hh.

References mass().

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

#### 8.12.3.9 void HepPDT::ResonanceStructure::setTotalWidth (Measurement const & *width*) `[inline]`

change the total width

Definition at line 63 of file ResonanceStructure.hh.

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

#### 8.12.3.10 void HepPDT::ResonanceStructure::setTotalWidthFromLifetime (Measurement const & *lt*)

change the total width using a lifetime

Definition at line 47 of file ResonanceStructure.cc.

References HepPDT::Measurement::sigma(), and HepPDT::Measurement::value().

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

#### 8.12.3.11 void HepPDT::ResonanceStructure::setLowerCutoff (double *cut*) `[inline]`

change the lower cutoff of allowed width values

Definition at line 67 of file ResonanceStructure.hh.

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

#### 8.12.3.12 void HepPDT::ResonanceStructure::setUpperCutoff (double *cut*) `[inline]`

change the upper cutoff of allowed width values

Definition at line 69 of file ResonanceStructure.hh.

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

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

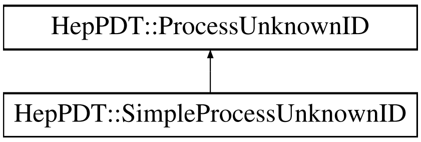
- **ResonanceStructure.hh**
- **lifetime.cc**
- **ResonanceStructure.cc**

## 8.13 HepPDT::SimpleProcessUnknownID Class Reference

The **SimpleProcessUnknownID** (p. 118) class inherits from **ProcessUnknownID** (p. 110).

`#include <SimpleProcessUnknownID.hh>`

Inheritance diagram for HepPDT::SimpleProcessUnknownID::



### Public Member Functions

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

### 8.13.1 Detailed Description

The **SimpleProcessUnknownID** (p. 118) class inherits from **ProcessUnknownID** (p. 110).

**Author:**

Lynn Garren

This provides the default **HepPDT** (p. 25) behaviour and will be invoked by default unless you specify a different behaviour when creating your **ParticleDataTable** (p. 88) object.

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 *key*, const ParticleDataTable & *pdt*) `[inline, virtual]`

Implements **HepPDT::ProcessUnknownID** (p. 110).

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

The **SpinState** (p. 119) class holds spin information.

`#include <SpinState.hh>`

### Public Member Functions

- **SpinState** (double ts=0., double spin=0., double oam=0.)

  *(default) constructor*

- **SpinState** (const **SpinState** &orig)
- **SpinState** & **operator=** (const **SpinState** &rhs)
- void **swap** (**SpinState** &other)
- bool **operator==** (const **SpinState** &rhs) const

  *all three spins must match*

- double **totalSpin** () const

  *return the total spin*

- double **spin** () const

  *return the spin*

- double **orbAngMom** () const

  *return the orbital angular momentum*

- void **setTotalSpin** (double spin)

  *change the total spin*

- void **setSpin** (double spin)

  *change the spin*

- void **setOrbAngMom** (double ang)

  *change the orbital angular momentum*

### 8.14.1 Detailed Description

The **SpinState** (p. 119) class holds spin information.

**Author:**

Lynn Garren

**SpinState** (p. 119) contains total spin, spin, and orbital angular momentum.

**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

Definition at line 11 of file SpinState.icc.

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

Definition at line 17 of file SpinState.icc.

## 8.14.3 Member Function Documentation

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

Definition at line 23 of file SpinState.icc.

References swap().

### 8.14.3.2 void HepPDT::SpinState::swap (SpinState & *other*) `[inline]`

Definition at line 30 of file SpinState.icc.

References itsOrbAngMom, itsSpin, itsTotalSpin, and HepPDT::swap().

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

### 8.14.3.3 bool HepPDT::SpinState::operator== (const SpinState & *rhs*) const `[inline]`

all three spins must match

Definition at line 36 of file SpinState.icc.

References itsOrbAngMom, itsSpin, and itsTotalSpin.

### 8.14.3.4 double HepPDT::SpinState::totalSpin () const `[inline]`

return the total spin

Definition at line 43 of file SpinState.hh.

Referenced by HepPDT::parseEvtGenLine(), HepPDT::parseQQParticle(), and HepPDT::ParticleData::write().

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

return the spin

Definition at line 45 of file SpinState.hh.

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

#### 8.14.3.6 double HepPDT::SpinState::orbAngMom () const [inline]

return the orbital angular momentum

Definition at line 47 of file SpinState.hh.

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

#### 8.14.3.7 void HepPDT::SpinState::setTotalSpin (double *spin*) [inline]

change the total spin

Definition at line 52 of file SpinState.hh.

Referenced by HepPDT::parseEvtGenLine(), HepPDT::parseQQParticle(), and HepPDT::TempParticleData::processPID().

#### 8.14.3.8 void HepPDT::SpinState::setSpin (double *spin*) [inline]

change the spin

Definition at line 54 of file SpinState.hh.

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

#### 8.14.3.9 void HepPDT::SpinState::setOrbAngMom (double *ang*) [inline]

change the orbital angular momentum

Definition at line 56 of file SpinState.hh.

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

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

- **SpinState.hh**
- **SpinState.icc**

## 8.15 HepPDT::TableBuilder Class Reference

The **TableBuilder** (p. 122) class is used to construct a **ParticleDataTable** (p. 88).

`#include <TableBuilder.hh>`

### Public Member Functions

- **TableBuilder** (**ParticleDataTable** &table, std::ostream &str=std::cerr)

  *create **TableBuilder** (p. 122) from a **ParticleDataTable** (p. 88)*

- ~**TableBuilder** ()

  *call the **ParticleDataTable** (p. 88) conversion method upon destruction*

- **TempParticleData** & **getParticleData** (**ParticleID** pid)

  *create a **TempParticleData** (p. 131) from a **ParticleID** (p. 96)*

- **TempParticleData** & **getParticleData** (std::string const &name)

  *create a **TempParticleData** (p. 131) from a particle name*

- **TempParticleData** & **getAntiParticle** (**ParticleID** pid, const std::string &aname)

  *create an antiparicle **TempParticleData** (p. 131) from a **ParticleID** (p. 96)*

- void **addParticle** (**TempParticleData** const &pd)

  *add a **TempParticleData** (p. 131) to the map*

- void **removeParticle** (**ParticleID** pid)

  *remove a **TempParticleData** (p. 131) 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

The **TableBuilder** (p. 122) class is used to construct a **ParticleDataTable** (p. 88).

**Author:**

Marc Paterno, Walter Brown, Lynn Garren

Define this class and use the add methods to define a **ParticleDataTable** (p. 88). The destructor fills **ParticleDataTable** (p. 88) from the information in **TableBuilder** (p. 122). See the examples for user code.

**Examples:**

**examMyPDT.cc**, **listEvtGenNames.cc.in**, **listPDGNames.cc.in**, **listPythiaNames.cc.in**, **testHepPDT.cc**, **testReadEvtGen.cc.in**, **testReadIsajet.cc.in**, **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. 122) from a **ParticleDataTable** (p. 88)

Definition at line 49 of file TableBuilder.hh.

#### 8.15.2.2 HepPDT::TableBuilder::∼TableBuilder () `[inline]`

call the **ParticleDataTable** (p. 88) 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 TempParticleData & HepPDT::TableBuilder::getParticleData (ParticleID *pid*) `[inline]`

create a **TempParticleData** (p. 131) from a **ParticleID** (p. 96)

**Examples:**

**examMyPDT.cc**.

Definition at line 21 of file TableBuilder.icc.

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

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

create a **TempParticleData** (p. 131) from a particle name

Definition at line 35 of file TableBuilder.icc.

References getParticleData().

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

create an antiparicle **TempParticleData** (p. 131) from a **ParticleID** (p. 96)

Definition at line 47 of file TableBuilder.icc.

References HepPDT::TempParticleData::antiparticle(), getParticleData(), HepPDT::ParticleID::pid(), and HepPDT::TempParticleData::tempParticleName.

Referenced by HepPDT::addPythiaParticles().

#### 8.15.3.4 void HepPDT::TableBuilder::addParticle (TempParticleData const & *pd*) `[inline]`

add a **TempParticleData** (p. 131) to the map

**Examples:**

**examMyPDT.cc**.

Definition at line 83 of file TableBuilder.icc.

References getParticleData(), hasParticleData(), HepPDT::TempParticleData::tempID, and HepPDT::TempParticleData::tempParticleName.

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

#### 8.15.3.5 void HepPDT::TableBuilder::removeParticle (ParticleID *pid*) `[inline]`

remove a **TempParticleData** (p. 131) from the map

**Examples:**

**examMyPDT.cc**.

Definition at line 68 of file TableBuilder.hh.

Referenced by main().

#### 8.15.3.6 void HepPDT::TableBuilder::addAlias (TempAliasData const & *ad*) `[inline]`

add alias information to the alias map

Definition at line 91 of file TableBuilder.icc.

References HepPDT::TempAliasData::tempAlias.

Referenced by HepPDT::addEvtGenParticles().

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

check to see if this particle is already defined

Definition at line 71 of file TableBuilder.icc.

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

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

check to see if this alias is already defined

Definition at line 77 of file TableBuilder.icc.

Referenced by HepPDT::addEvtGenParticles().

#### 8.15.3.9 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().

#### 8.15.3.10 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().

#### 8.15.3.11 int HepPDT::TableBuilder::aliasSize () const `[inline]`

get size of alias map

Definition at line 87 of file TableBuilder.hh.

Referenced by HepPDT::addEvtGenParticles().

#### 8.15.3.12 DefTable& HepPDT::TableBuilder::definitions () `[inline]`

get the list of definitions (for EvtGen)

Definition at line 89 of file TableBuilder.hh.

Referenced by HepPDT::addEvtGenParticles().

#### 8.15.3.13 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().

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

find an entry in the alias map

Definition at line 97 of file TableBuilder.icc.

Referenced by HepPDT::addEvtGenParticles().

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

- **TableBuilder.hh**
- **TableBuilder.icc**

# 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::TableBuilder::addAlias(), and HepPDT::parseEvtGenAlias().

#### 8.16.3.2 std::string HepPDT::TempAliasData::tempAliasedParticle

the "real" particle

Definition at line 55 of file TempParticleData.hh.

Referenced by HepPDT::parseEvtGenAlias().

#### 8.16.3.3 std::string HepPDT::TempAliasData::tempChargeConj

set if there is a charge conjugate alias

Definition at line 56 of file TempParticleData.hh.

Referenced by HepPDT::addEvtGenParticles().

#### 8.16.3.4 TDDlist HepPDT::TempAliasData::tempAliasDecayList

decay list for the alias

Definition at line 57 of file TempParticleData.hh.

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**
- 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 std::string HepPDT::TempDecayData::tempDecayName

Type of decay. Use string to hold int if necessary.

Definition at line 36 of file TempParticleData.hh.

### 8.18.2.2 double HepPDT::TempDecayData::tempBranchingFraction

Definition at line 37 of file TempParticleData.hh.

### 8.18.2.3 std::vector<std::string> HepPDT::TempDecayData::tempDaughterList

list of decay particles

Definition at line 39 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** ()
- **TempParticleData** (int id)

  *construct a basic* ***TempParticleData*** *(*p. *131) from just the particle ID*

- **TempParticleData** (**ParticleID** pid)

  *construct a basic* ***TempParticleData*** *(*p. *131) from just the* ***ParticleID*** *(*p. *96)*

- **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. *131)*

- **TempParticleData** (**TempParticleData** const &orig)
- **TempParticleData** & **operator=** (**TempParticleData** const &rhs)
- void **swap** (**TempParticleData** &other)
- **TempParticleData antiparticle** (std::string const &name)

  *given a particle definition, create an antiparticle*

- bool **processPID** ()

  *– mutator - get spin state and constituent list from PID*

### Public Attributes

- **ParticleID tempID**
- std::string **tempParticleName**
- std::string **tempSource**
- int **tempOriginalID**
- double **tempCharge**
- double **tempColorCharge**
- **SpinState tempSpin**
- **Measurement tempMass**
- **Measurement tempWidth**
- double **tempLowCutoff**
- double **tempHighCutoff**
- std::vector< **TempConstituent** > **tempQuarks**
- **TDDlist tempDecayList**

## 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. 131) 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. 131) from just the **ParticleID** (p. 96)

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. 131)

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::operator= (TempParticleData const & *rhs*)

Definition at line 122 of file TempParticleData.cc.

References swap().

#### 8.19.3.2 void HepPDT::TempParticleData::swap (TempParticleData & *other*)

Definition at line 129 of file TempParticleData.cc.

References HepPDT::swap(), tempCharge, tempColorCharge, tempDecayList, tempHighCutoff, tempID, tempLowCutoff, tempMass, tempOriginalID, tempParticleName, tempQuarks, tempSource, tempSpin, and tempWidth.

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

#### 8.19.3.3 TempParticleData HepPDT::TempParticleData::antiparticle (std::string const & *name*)

given a particle definition, create an antiparticle

Definition at line 87 of file TempParticleData.cc.

References HepPDT::ParticleID::pid(), processPID(), tempCharge, tempColorCharge, tempDecayList, tempHighCutoff, tempID, tempLowCutoff, tempMass, tempOriginalID, tempParticleName, tempSource, and tempWidth.

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

#### 8.19.3.4 bool HepPDT::TempParticleData::processPID ()

– mutator - get spin state and constituent list from PID

Definition at line 146 of file TempParticleData.cc.

References HepPDT::ParticleID::jSpin(), HepPDT::ParticleID::lSpin(), HepPDT::Quarks::nq1, HepPDT::Quarks::nq2, HepPDT::Quarks::nq3, HepPDT::ParticleID::pid(), HepPDT::ParticleID::quarks(), HepPDT::SpinState::setOrbAngMom(), HepPDT::SpinState::setSpin(), HepPDT::SpinState::setTotalSpin(), HepPDT::spinitod(), HepPDT::ParticleID::sSpin(), tempCharge, HepPDT::TempConstituent::tempConstituentPID, tempID, HepPDT::TempConstituent::tempMultiplicity, tempQuarks, tempSpin, and HepPDT::ParticleID::threeCharge().

Referenced by antiparticle(), and TempParticleData().

### 8.19.4 Member Data Documentation

#### 8.19.4.1 ParticleID HepPDT::TempParticleData::tempID

Definition at line 89 of file TempParticleData.hh.

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

#### 8.19.4.2 std::string HepPDT::TempParticleData::tempParticleName

**Examples:**

**examMyPDT.cc**.

Definition at line 90 of file TempParticleData.hh.

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

#### 8.19.4.3 std::string HepPDT::TempParticleData::tempSource

Definition at line 91 of file TempParticleData.hh.

Referenced by HepPDT::addPDGParticles(), HepPDT::addPythiaParticles(), HepPDT::addQQParticles(), antiparticle(), HepPDT::parseEvtGenLine(), HepPDT::detail::parseParticleLine(), HepPDT::detail::parsePythiaLine(), and swap().

#### 8.19.4.4 int HepPDT::TempParticleData::tempOriginalID

Definition at line 92 of file TempParticleData.hh.

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

#### 8.19.4.5 double HepPDT::TempParticleData::tempCharge

**Examples:**

**examMyPDT.cc**.

Definition at line 93 of file TempParticleData.hh.

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

#### 8.19.4.6 double HepPDT::TempParticleData::tempColorCharge

Definition at line 94 of file TempParticleData.hh.

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

#### 8.19.4.7 SpinState HepPDT::TempParticleData::tempSpin

**Examples:**

**examMyPDT.cc**.

Definition at line 95 of file TempParticleData.hh.

Referenced by addData(), HepPDT::parseEvtGenLine(), HepPDT::detail::parseIsajetLine(), HepPDT::parseQQParticle(), processPID(), and swap().

#### 8.19.4.8 Measurement HepPDT::TempParticleData::tempMass

**Examples:**

**examMyPDT.cc**.

Definition at line 96 of file TempParticleData.hh.

Referenced by addData(), HepPDT::addPythiaParticles(), antiparticle(), HepPDT::detail::CheckPDGEntry(), HepPDT::parseEvtGenLine(), HepPDT::detail::parseIsajetLine(), HepPDT::detail::parseParticleLine(), HepPDT::detail::parsePythiaLine(), HepPDT::parseQQParticle(), and swap().

#### 8.19.4.9 Measurement HepPDT::TempParticleData::tempWidth

**Examples:**

**examMyPDT.cc**.

Definition at line 97 of file TempParticleData.hh.

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

#### 8.19.4.10 double HepPDT::TempParticleData::tempLowCutoff

Definition at line 98 of file TempParticleData.hh.

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

#### 8.19.4.11 double HepPDT::TempParticleData::tempHighCutoff

Definition at line 99 of file TempParticleData.hh.

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

#### 8.19.4.12 std::vector<TempConstituent> HepPDT::TempParticleData::tempQuarks

Definition at line 100 of file TempParticleData.hh.

Referenced by processPID(), and swap().

#### 8.19.4.13 TDDlist HepPDT::TempParticleData::tempDecayList

Definition at line 101 of file TempParticleData.hh.

Referenced by antiparticle(), and swap().

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

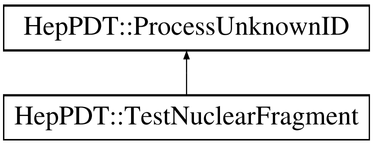
- **TempParticleData.hh**
- **TempParticleData.cc**

## 8.20 HepPDT::TestNuclearFragment Class Reference

The **TestNuclearFragment** (p. 136) class inherits from **ProcessUnknownID** (p. 110).

`#include <TestNuclearFragment.hh>`

Inheritance diagram for HepPDT::TestNuclearFragment::



### Public Member Functions

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

### 8.20.1 Detailed Description

The **TestNuclearFragment** (p. 136) class inherits from **ProcessUnknownID** (p. 110).

**Author:**

Lynn Garren

Create and return a pointer to a **ParticleData** (p. 76) object if invoked by a valid nuclear fragment. TestNuclearFragment::processUnknownID( ParticleID ) calculates both charge and an approximate mass for the particle. This implementation also sends output to cout for code checking

**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 & *pdt*) `[inline, virtual]`

Implements **HepPDT::ProcessUnknownID** (p. 110).

Definition at line 39 of file TestNuclearFragment.hh.

References HepPDT::ParticleID::isNucleus(), HepPDT::ParticleData::mass(), HepPDT::ParticleDataTable::particle(), 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 &ltype, int &id, std::string &name, const std::string &pdline)
- void **HepPDT::parseEvtGenLine** (TempParticleData &tpd, const std::string &pdline)
- void **HepPDT::parseEvtGenAlias** (TempAliasData &tad, const std::string &pdline)
- bool **HepPDT::parseEvtGenDecayLine** (TempParticleData &tpd, const std::string &pdline)
- bool **HepPDT::parseEvtGenAliasDecayLine** (TempAliasData &tad, const std::string &pdline)
- 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 &pdfile, TableBuilder &tb)

    *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 &pdfile, TableBuilder &tb)

## 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 &pdfile, TableBuilder &tb)

  *read Isajet particle input and add particles to the table*

- void **HepPDT::detail::parseIsajetLine** (TempParticleData &tpd, const std::string &pdline)

  *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 &pdfile, TableBuilder &tb, bool validate=false)
- bool **HepPDT::detail::getParticleID** (int &id, const std::string &pdline)

  *for internal use*

- void **HepPDT::detail::parseParticleLine** (TempParticleData &tpd, const std::string &pdline)

  *for internal use*

## 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 &pdfile, TableBuilder &tb)

  *read PDG input and add particles to the table*

- void **HepPDT::detail::parsePDGline** (TempParticleData &tpd, std::string &pdline)

  *for internal use*

- bool **HepPDT::detail::CheckPDGEntry** (TempParticleData &tpd, const std::string &, double, double)

  *for internal use*

## 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 &pdfile, TableBuilder &tb)

  *read Pythia input and add particles to the table*

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

  *for internal use*

- void **HepPDT::detail::parsePythiaDecayLine** (TempParticleData &tpd, const std::string &pdline)

  *for internal use*

## 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 &ltype, 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 &pdfile, TableBuilder &tb)

  *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)

## 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**

  *The **Constituent** (*p. *65) class has information about constituent particles.*

### 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**

  *The* ***DefTable*** *(*p. *68) class holds EvtGen definitions.*

# 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

list Herwig particle ID translations Get ID list directly from Herwig

When mixing C++ and Fortran, the main program must be C++

**Examples:**

**examListHerwig.cc**.

Referenced by main().

#### 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

List the isajet particle ID translations Get ID list directly from isajet

When mixing C++ and Fortran, the main program must be C++

**Examples:**

**examListIsajet.cc**.

Referenced by main().

#### 9.15.1.4 int main ()

Definition at line 29 of file examListIsajet.cc.

References flavor_(), get_label_(), list_isajet_init_(), HepPID::particleName(), HepPID::translateIsajetto-PDT(), 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

list Pythia particle ID translations Get ID list directly from Pythia

When mixing C++ and Fortran, the main program must be C++

**Examples:**

**examListPythia.cc**.

Referenced by main().

#### 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

create a custom PDT using our own definitions and write out the resulting PDT

**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 &id, const std::string &pdline)
  *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 > &idlist, std::string &pdline)

  *for internal use*

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

  *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 &id, const std::string &pdline)

  *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**

  *The* ***HeavyIonUnknownID*** (p. *71) class inherits from* ***ProcessUnknownID*** (p. *110).*

## 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 listEvtGenNames.cc.in File Reference

```
#include <fstream>
#include <iostream>
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

## Functions

- int **main** ()

## 9.27.1 Function Documentation

### 9.27.1.1 int main ()

Definition at line 15 of file listEvtGenNames.cc.in.

References HepPDT::addEvtGenParticles(), and HepPDT::ParticleDataTable::writeParticleTranslation().

# 9.28 listEvtGenTranslation.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 listEvtGenTranslation.cc.

References HepPID::writeEvtGenTranslation().

## 9.29 listHerwigTranslation.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 listHerwigTranslation.cc.

References HepPID::writeHerwigTranslation().

# 9.30 listIsajetTranslation.cc File Reference

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

## Functions

- int **main** ()

## 9.30.1 Function Documentation

### 9.30.1.1 int main ()

Definition at line 15 of file listIsajetTranslation.cc.

References HepPID::writeIsajetTranslation().

# 9.31 listParticleNames.cc File Reference

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

## Functions

- int **main** ()

## 9.31.1 Function Documentation

### 9.31.1.1 int main ()

Definition at line 17 of file listParticleNames.cc.

References HepPID::listParticleNames().

# 9.32 listPDGNames.cc.in File Reference

```
#include <fstream>
#include <iostream>
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

## Functions

- int **main** ()

## 9.32.1 Function Documentation

### 9.32.1.1 int main ()

Definition at line 15 of file listPDGNames.cc.in.

References HepPDT::addPDGParticles(), and HepPDT::ParticleDataTable::writeParticleTranslation().

# 9.33 listPDGTranslation.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 listPDGTranslation.cc.

References HepPID::writePDGTranslation().

# 9.34 listPythiaNames.cc.in File Reference

```
#include <fstream>
#include <iostream>
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

## Functions

- int **main** ()

## 9.34.1 Function Documentation

### 9.34.1.1 int main ()

Definition at line 15 of file listPythiaNames.cc.in.

References HepPDT::addPythiaParticles(), and HepPDT::ParticleDataTable::writeParticleTranslation().

# 9.35 listPythiaTranslation.cc File Reference

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

## Functions

- int **main** ()

## 9.35.1 Function Documentation

### 9.35.1.1 int main ()

Definition at line 15 of file listPythiaTranslation.cc.

References HepPID::writePythiaTranslation().

# 9.36 listQQTranslation.cc File Reference

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

## Functions

- int **main** ()

## 9.36.1 Function Documentation

### 9.36.1.1 int main ()

Definition at line 15 of file listQQTranslation.cc.

References HepPID::writeQQTranslation().

## 9.37 Measurement.hh File Reference

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

### Namespaces

- namespace **HepPDT**

### Classes

- class **HepPDT::Measurement**

  *The **Measurement** (p. 73) class defines a value with its error.*

### Functions

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

## 9.38 Measurement.icc File Reference

```
#include <algorithm>
```

### Namespaces

- namespace **HepPDT**

### Functions

- double **HepPDT::NaN** ()

## 9.39 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**

  *The* ***ParticleData*** (p. *76) class holds data for a single particle in the table.*

### Functions

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

## 9.40 ParticleData.icc File Reference

`#include <algorithm>`

### Namespaces

- namespace **HepPDT**

## 9.41 ParticleDataTable.cc File Reference

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

### Namespaces

- namespace **HepPDT**

## 9.42 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**

  *The* ***ParticleDataTable*** *(*p. *88) class is the core of* ***HepPDT*** *(*p. *25).*

### 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)

## 9.43 ParticleDataTable.icc File Reference

### Namespaces

- namespace **HepPDT**

## 9.44 ParticleDataTableComparison.hh File Reference

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

### Namespaces

- namespace **HepPDT**

### Classes

- class **HepPDT::ParticleDataTableComparison**

  *The **ParticleDataTableComparison** (p. 95) class provides a utility for sorting the PDT.*

# 9.45 ParticleID.cc File Reference

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

## Namespaces

- namespace **HepPDT**

## 9.46 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**

  *The **ParticleID** (p. 96) has various utilities to extract information from the particle ID.*

### Enumerations

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

### Functions

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

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

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

  *convert an actual spin to 2J+1*

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

## 9.47 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)
- 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)
- int **HepPID::A** (const int &pid)
- int **HepPID::lambda** (const int &pid)
- bool **HepPID::isValid** (const int &pid)

  *is this a valid ID?*

- bool **HepPID::hasFundamentalAnti** (const int &pid)

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

- bool **HepPID::isMeson** (const int &pid)

  *is this a valid meson ID?*

- bool **HepPID::isBaryon** (const int &pid)

  *is this a valid baryon ID?*

- bool **HepPID::isDiQuark** (const int &pid)

  *is this a valid diquark ID?*

- bool **HepPID::isHadron** (const int &pid)

  *is this a valid hadron ID?*

- bool **HepPID::isLepton** (const int &pid)

  *is this a valid lepton ID?*

- bool **HepPID::isNucleus** (const int &pid)

*is this a valid ion ID?*

- bool **HepPID::isPentaquark** (const int &pid)

  *is this a valid pentaquark ID?*

- bool **HepPID::isSUSY** (const int &pid)

  *is this a valid SUSY ID?*

- bool **HepPID::isRhadron** (const int &pid)

  *is this a valid R-hadron ID?*

- bool **HepPID::hasUp** (const int &pid)

  *does this particle contain an up quark?*

- bool **HepPID::hasDown** (const int &pid)

  *does this particle contain a down quark?*

- bool **HepPID::hasStrange** (const int &pid)

  *does this particle contain a strange quark?*

- bool **HepPID::hasCharm** (const int &pid)

  *does this particle contain a charm quark?*

- bool **HepPID::hasBottom** (const int &pid)

  *does this particle contain a bottom quark?*

- bool **HepPID::hasTop** (const int &pid)

  *does this particle contain a top quark?*

- int **HepPID::jSpin** (const int &pid)

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

- int **HepPID::sSpin** (const int &pid)

  *sSpin returns 2S+1, where S is the spin*

- int **HepPID::lSpin** (const int &pid)

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

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

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

# 9.48 ParticleIDMethods.hh File Reference

## Namespaces

- namespace **HepPID**

## Enumerations

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

## Functions

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

  *return the digit at a named location in the PID*

- int **HepPID::A** (const int &pid)
- int **HepPID::Z** (const int &pid)
- int **HepPID::lambda** (const int &pid)
- int **HepPID::abspid** (const int &pid)

  *absolute value of particle ID*

- int **HepPID::fundamentalID** (const int &pid)

  *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)
- bool **HepPID::isValid** (const int &pid)

  *is this a valid ID?*

- bool **HepPID::isMeson** (const int &pid)

  *is this a valid meson ID?*

- bool **HepPID::isBaryon** (const int &pid)

  *is this a valid baryon ID?*

- bool **HepPID::isDiQuark** (const int &pid)

  *is this a valid diquark ID?*

- bool **HepPID::isHadron** (const int &pid)

  *is this a valid hadron ID?*

- bool **HepPID::isLepton** (const int &pid)

  *is this a valid lepton ID?*

- bool **HepPID::isNucleus** (const int &pid)

  *is this a valid ion ID?*

- bool **HepPID::isPentaquark** (const int &pid)

  *is this a valid pentaquark ID?*

- bool **HepPID::isSUSY** (const int &pid)

  *is this a valid SUSY ID?*

- bool **HepPID::isRhadron** (const int &pid)

  *is this a valid R-hadron ID?*

- bool **HepPID::hasUp** (const int &pid)

  *does this particle contain an up quark?*

- bool **HepPID::hasDown** (const int &pid)

  *does this particle contain a down quark?*

- bool **HepPID::hasStrange** (const int &pid)

  *does this particle contain a strange quark?*

- bool **HepPID::hasCharm** (const int &pid)

  *does this particle contain a charm quark?*

- bool **HepPID::hasBottom** (const int &pid)

  *does this particle contain a bottom quark?*

- bool **HepPID::hasTop** (const int &pid)

  *does this particle contain a top quark?*

- int **HepPID::jSpin** (const int &pid)

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

- int **HepPID::sSpin** (const int &pid)

  *sSpin returns 2S+1, where S is the spin*

- int **HepPID::lSpin** (const int &pid)

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

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

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

## 9.49 ParticleIDTranslations.hh File Reference

```
#include <iostream>
```

### Namespaces

- namespace **HepPID**

### Functions

- int **HepPID::translateHerwigtoPDT** (const int herwigID)
  *translate Herwig to PDG standard*
- int **HepPID::translatePDTtoHerwig** (const int pid)
  *translate PDG standard to Herwig*
- void **HepPID::writeHerwigTranslation** (std::ostream &os)
  *output the translation list*
- int **HepPID::translateIsajettoPDT** (const int isajetID)
  *translate Isajet to PDG standard*
- int **HepPID::translatePDTtoIsajet** (const int pid)
  *translate PDG standard to Isajet*
- void **HepPID::writeIsajetTranslation** (std::ostream &os)
  *output the translation list*
- int **HepPID::translatePythiatoPDT** (const int pythiaID)
  *translate Pythia to PDG standard*
- int **HepPID::translatePDTtoPythia** (const int pid)
  *translate PDG standard to Pythia*
- void **HepPID::writePythiaTranslation** (std::ostream &os)
  *output the translation list*
- int **HepPID::translateEvtGentoPDT** (const int evtGenID)
  *translate EvtGen to PDG standard*
- int **HepPID::translatePDTtoEvtGen** (const int pid)
  *translate PDG standard to EvtGen*
- void **HepPID::writeEvtGenTranslation** (std::ostream &os)
  *output the translation list*
- int **HepPID::translatePDGtabletoPDT** (const int pdgID)
  *translate PDG table to PDG standard*

- int **HepPID::translatePDTtoPDGtable** (const int pid)
  *translate PDG standard to PDG table*
- void **HepPID::writePDGTranslation** (std::ostream &os)
  *output the translation list*
- int **HepPID::translateQQtoPDT** (const int qqID)
  *translate QQ to PDG standard*
- int **HepPID::translatePDTtoQQ** (const int pid)
  *translate PDG standard to QQ*
- int **HepPID::translateQQbar** (const int id)
  *QQ helper function.*
- int **HepPID::translateInverseQQbar** (const int id)
  *QQ helper function.*
- void **HepPID::writeQQTranslation** (std::ostream &os)
  *output the translation list*
- int **HepPID::translateGeanttoPDT** (const int geantID)
  *translate Geant3 to PDG standard*
- int **HepPID::translatePDTtoGeant** (const int pid)
  *translate PDG standard to Geant3*

## 9.50 ParticleName.cc File Reference

```
#include <string>
#include <map>
#include <iostream>
#include <iomanip>
#include <utility>
#include "HepPID/ParticleName.hh"
#include "HepPID/Version.hh"
```

### Namespaces

- namespace **HepPID**

### Classes

- class **HepPID::ParticleNameMap**

### Typedefs

- typedef std::map< int, std::string > **HepPID::PartcleIdMap**
- typedef std::map< std::string, int > **HepPID::ParticleLookupMap**

### Functions

- ParticleNameMap const & **HepPID::ParticleNameInit** ()
- void **HepPID::writeParticleNameLine** (int i, std::ostream &os)
- ParticleNameMap const & **HepPID::getParticleNameMap** ()

  *access the* ***ParticleNameMap*** *(*p. *108) 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. *39) Particle name*

- int **HepPID::particleName** (const std::string &)

  *lookup a known ID*

- void **HepPID::listParticleNames** (std::ostream &os)

  *list all known names*

## 9.51 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. *39) 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. *108) for other purposes*

# 9.52 ProcessUnknownID.cc File Reference

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

## Namespaces

- namespace **HepPDT**

## 9.53 ProcessUnknownID.hh File Reference

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

### Namespaces

- namespace **HepPDT**

### Classes

- class **HepPDT::ProcessUnknownID**

  *The* ***ProcessUnknownID*** *(*p. *110) class is abstract.*

## 9.54 quarks.cc File Reference

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

### Namespaces

- namespace **HepPDT**

## 9.55 ResonanceStructure.cc File Reference

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

### Namespaces

- namespace **HepPDT**

## 9.56 ResonanceStructure.hh File Reference

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

### Namespaces

- namespace **HepPDT**

### Classes

- class **HepPDT::ResonanceStructure**

  *The **ResonanceStructure** (*p.* 114) class is holds mass and width information.*

### Functions

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

## 9.57 SimpleProcessUnknownID.hh File Reference

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

### Namespaces

- namespace **HepPDT**

### Classes

- class **HepPDT::SimpleProcessUnknownID**

  *The* ***SimpleProcessUnknownID*** *(*p. *118) class inherits from* ***ProcessUnknownID*** *(*p. *110).*

## 9.58 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*

## 9.59 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*

## 9.60 SpinState.hh File Reference

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

### Namespaces

- namespace **HepPDT**

### Classes

- class **HepPDT::SpinState**

  *The **SpinState** (p. 119) class holds spin information.*

### Functions

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

## 9.61 SpinState.icc File Reference

```
#include <algorithm>
```

### Namespaces

- namespace **HepPDT**

## 9.62 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*

## 9.63 stringtodouble.hh File Reference

```
#include <string>
```

### Namespaces

- namespace **HepPDT**

### Functions

- double **HepPDT::stringtodouble** (std::string &numb)
  *extract a double from a string*

# 9.64 TableBuilder.hh File Reference

```
#include <iostream>
#include <string>
#include <map>
#include "HepPDT/ParticleDataTable.hh"
#include "HepPDT/TempParticleData.hh"
#include "HepPDT/DefTable.hh"
#include "HepPDT/stringtodouble.hh"
#include "HepPDT/TableBuilder.icc"
```

## Namespaces

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

## Classes

- class **HepPDT::TableBuilder**

  *The **TableBuilder** (p. 122) class is used to construct a **ParticleDataTable** (p. 88).*

## Functions

- bool **HepPDT::addPDGParticles** (std::istream &pdfile, TableBuilder &tb)

  *read PDG input and add particles to the table*

- bool **HepPDT::addPythiaParticles** (std::istream &pdfile, TableBuilder &tb)

  *read Pythia input and add particles to the table*

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

  *read Isajet particle input and add particles to the table*

- bool **HepPDT::addIsajetDecay** (std::istream &pdfile, TableBuilder &tb)

  *read Isajet decay input and add decay information to the table*

- bool **HepPDT::addQQParticles** (std::istream &pdfile, TableBuilder &tb)

  *read QQ input and add particles to the table*

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

  *read EvtGen input and add particles to the table*

- bool **HepPDT::addParticleTable** (std::istream &pdfile, TableBuilder &tb, bool validate=false)
- void **HepPDT::detail::getPDGpid** (std::vector< int > &idlist, std::string &pdline)

  *for internal use*

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

  *for internal use*

- void **HepPDT::detail::parsePDGline** (TempParticleData &tpd, std::string &pdline)

  *for internal use*

- bool **HepPDT::detail::CheckPDGEntry** (TempParticleData &tpd, const std::string &, double, double)

  *for internal use*

- bool **HepPDT::detail::getPythiaid** (int &id, const std::string &pdline)

  *for internal use*

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

  *for internal use*

- void **HepPDT::detail::parsePythiaDecayLine** (TempParticleData &tpd, const std::string &pdline)

  *for internal use*

- TempDecayData **HepPDT::detail::getPythiaDecay** (const std::string &pdline)

  *for internal use*

- bool **HepPDT::detail::getIsajetID** (int &id, const std::string &pdline)

  *for internal use*

- void **HepPDT::detail::parseIsajetLine** (TempParticleData &tpd, const std::string &pdline)

  *for internal use*

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

  *for internal use*

- bool **HepPDT::detail::getParticleID** (int &id, const std::string &pdline)

  *for internal use*

- void **HepPDT::detail::parseParticleLine** (TempParticleData &tpd, const std::string &pdline)

  *for internal use*

## 9.65 TableBuilder.icc File Reference

```
#include <sstream>
```

### Namespaces

- namespace **HepPDT**

## 9.66 TempParticleData.cc File Reference

```
#include <algorithm>
#include "HepPDT/defs.h"
#include "HepPDT/TempParticleData.hh"
```

### Namespaces

- namespace **HepPDT**

## 9.67 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)
- void **HepPDT::swap** (TempParticleData &first, TempParticleData &second)

## 9.68 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 **testPDMethods** (**HepPDT::ParticleDataTable** &, std::ofstream &)
- int **main** ()

### 9.68.1 Function Documentation

#### 9.68.1.1 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().

#### 9.68.1.2 int main ()

Definition at line 27 of file testHepPDT.cc.

References duplicateFragmentTest(), pdtFragmentTest(), and pdtSimpleTest().

#### 9.68.1.3 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().

#### 9.68.1.4 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(), testPDMethods(), HepPDT::ParticleData::totalWidth(), HepPDT::ParticleData::upperCutoff(), HepPDT::ParticleData::write(), HepPDT::ParticleDataTable::writeParticleData(), and HepPDT::ParticleDataTable::writeParticleInfo().

Referenced by main().

#### 9.68.1.5 void testPDMethods (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().

## 9.69 TestNuclearFragment.hh File Reference

```
#include <sstream>
#include "HepPDT/ProcessUnknownID.hh"
#include "HepPDT/ParticleData.hh"
```

### Namespaces

- namespace **HepPDT**

### Classes

- class **HepPDT::TestNuclearFragment**

  *The* ***TestNuclearFragment*** (p. *136) class inherits from* ***ProcessUnknownID*** (p. *110).*

# 9.70 testParticleIDMethods.cc File Reference

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

## Functions

- int **main** ()

## 9.70.1 Function Documentation

### 9.70.1.1 int main ()

Definition at line 19 of file testParticleIDMethods.cc.

References HepPID::A(), HepPID::abspid(), HepPID::digit(), HepPID::extraBits(), HepPID::fundamentalID(), HepPID::hasBottom(), HepPID::hasCharm(), HepPID::hasDown(), HepPID::hasStrange(), HepPID::hasTop(), HepPID::hasUp(), HepPID::isBaryon(), HepPID::isDiQuark(), HepPID::isHadron(), HepPID::isLepton(), HepPID::isMeson(), HepPID::isNucleus(), HepPID::isPentaquark(), 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, HepPID::particleName(), HepPID::sSpin(), HepPID::threeCharge(), and HepPID::Z().

# 9.71 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** ()

## 9.71.1 Function Documentation

### 9.71.1.1 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, HepPID::nr, testDiQuark(), testHadron(), testHasMethods(), testLepton(), testNucleus(), testUnknown(), and testValid().

### 9.71.1.2 void testDiQuark (HepPDT::ParticleID &, int &)

**Examples:**

**testPID.cc**.

Definition at line 109 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().

### 9.71.1.3 void testHadron (HepPDT::ParticleID &, int &)

**Examples:**

**testPID.cc**.

Definition at line 63 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::jSpin(), HepPDT::ParticleID::lSpin(), HepPDT::n, HepPDT::Quarks::nq1, HepPDT::Quarks::nq2, HepPDT::Quarks::nq3, HepPDT::nr, HepPID::nr, HepPDT::ParticleID::pid(), HepPDT::ParticleID::quarks(), and HepPDT::ParticleID::threeCharge().

Referenced by main().

#### 9.71.1.4 void testHasMethods (HepPDT::ParticleID &, int &)

**Examples:**

**testPID.cc**.

Definition at line 175 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::PDTname(), and HepPDT::ParticleID::pid().

Referenced by main().

#### 9.71.1.5 void testLepton (HepPDT::ParticleID &, int &)

**Examples:**

**testPID.cc**.

Definition at line 93 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().

#### 9.71.1.6 void testNucleus (HepPDT::ParticleID &, int &)

**Examples:**

**testPID.cc**.

Definition at line 125 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::quarks(), HepPDT::ParticleID::threeCharge(), and HepPDT::ParticleID::Z().

Referenced by main().

#### 9.71.1.7 void testUnknown (HepPDT::ParticleID &, int &)

**Examples:**

**testPID.cc**.

Definition at line 141 of file testPID.cc.

References HepPDT::ParticleID::digit(), HepPDT::ParticleID::extraBits(), HepPDT::ParticleID::fundamentalID(), HepPDT::ParticleID::isDiQuark(), HepPDT::ParticleID::isHadron(), HepPDT::ParticleID::isLepton(), HepPDT::ParticleID::isNucleus(), 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().

#### 9.71.1.8 void testValid (HepPDT::ParticleID &)

**Examples:**

**testPID.cc**.

Definition at line 159 of file testPID.cc.

References 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().

## 9.72 testReadEvtGen.cc.in File Reference

```
#include <fstream>
#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

### Functions

- int **main** ()

### 9.72.1 Function Documentation

#### 9.72.1.1 int main ()

Definition at line 15 of file testReadEvtGen.cc.in.

References HepPDT::addEvtGenParticles(), and HepPDT::ParticleDataTable::writeParticleData().

## 9.73 testReadIsajet.cc.in File Reference

```
#include <fstream>
#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

### Functions

- int **main** ()

### 9.73.1 Function Documentation

#### 9.73.1.1 int main ()

Definition at line 18 of file testReadIsajet.cc.in.

References HepPDT::addIsajetParticles(), HepPDT::ParticleDataTable::writeParticleData(), and HepPDT::ParticleDataTable::writeParticleInfo().

# 9.74 testReadParticleTable.cc.in File Reference

```
#include <fstream>
#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

## Functions

- int **main** ()

## 9.74.1 Function Documentation

### 9.74.1.1 int main ()

Definition at line 15 of file testReadParticleTable.cc.in.

References HepPDT::addParticleTable(), HepPDT::ParticleDataTable::particle(), HepPDT::ParticleData::write(), HepPDT::ParticleDataTable::writeParticleData(), and HepPDT::ParticleDataTable::writeParticleStatus().

# 9.75 testReadQQ.cc.in File Reference

```
#include <fstream>
#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
```

## Functions

- int **main** ()

## 9.75.1 Function Documentation

### 9.75.1.1 int main ()

Definition at line 15 of file testReadQQ.cc.in.

References HepPDT::addQQParticles(), HepPDT::ParticleDataTable::writeParticleData(), and HepPDT::ParticleDataTable::writeParticleTranslation().

## 9.76 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*

## 9.77 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*

### 9.77.1 Define Documentation

#### 9.77.1.1 #define IDMAX 49

Definition at line 16 of file translateGeanttoPDT.cc.

Referenced by HepPID::translateGeanttoPDT(), and HepPID::translatePDTtoGeant().

## 9.78 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*

## 9.79 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*

## 9.80 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*

# 9.81 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*

## 9.81.1 Define Documentation

### 9.81.1.1 #define IDMAX 49

Definition at line 18 of file translatePDTtoGeant.cc.

## 9.82 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*

## 9.83 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*

## 9.84 Version.cc File Reference

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

### Namespaces

- namespace **HepPDT**

### Functions

- std::string **HepPDT::versionName** ()

  *return **HepPDT** (*p.*25) version*

- void **HepPDT::version** ()

  *print **HepPDT** (*p.*25) version*

- void **HepPDT::writeVersion** (std::ostream &os)

  *write **HepPDT** (*p.*25) version to os*

## 9.85 Version.cc File Reference

```
#include "HepPID/Version.hh"
```

### Namespaces

- namespace **HepPID**

### Functions

- std::string **HepPID::versionName** ()

  *return **HepPID** (*p.*39) version*

- void **HepPID::version** ()

  *print **HepPID** (*p.*39) version*

- void **HepPID::writeVersion** (std::ostream &os)

  *write **HepPID** (*p.*39) version to os*

## 9.86 Version.hh File Reference

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

### Namespaces

- namespace **HepPDT**

### Functions

- void **HepPDT::version** ()
  *print **HepPDT** (*p.*25) version*
- void **HepPDT::writeVersion** (std::ostream &os)
  *write **HepPDT** (*p.*25) version to os*
- std::string **HepPDT::versionName** ()
  *return **HepPDT** (*p.*25) version*

## 9.87 Version.hh File Reference

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

### Namespaces

- namespace **HepPID**

### Functions

- void **HepPID::version** ()

  *print* ***HepPID*** *(*p.*39) version*

- void **HepPID::writeVersion** (std::ostream &os)

  *write* ***HepPID*** *(*p.*39) version to os*

- std::string **HepPID::versionName** ()

  *return* ***HepPID*** *(*p.*39) version*

## 9.88 write.cc File Reference

```
#include <iomanip>
#include <sstream>
#include "HepPDT/defs.h"
#include "HepPDT/ParticleData.hh"
```

### Namespaces

- namespace **HepPDT**
- namespace **std**

# Chapter 10

# HepPDT Example Documentation

## 10.1 examListHerwig.cc

list Herwig particle ID translations

```
// -------------------------------------------------------------------
// examListHerwig.cc
//
// ----------------------------------------------------------------------

#include <fstream>
#include <iostream>
#include <cstdlib>

#include "HepPID/ParticleIDTranslations.hh"
#include "HepPID/ParticleName.hh"
#include "HepPID/Version.hh"

extern "C" {
// these functions are defined in examListHerwigInterface.F
    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()
{
    int nevt=20;
    int i, j, iend, isize;
    int hid, id;
    char cname[10];
    std::string hname;
    std::string pn;
    static char outfile[] = "examListHerwig.out";
    std::string title = "HepPID listing of Herwig translations";

    // initialize herwig
    list_herwig_init_ ( & nevt );

    // open the output stream
    std::ofstream os( outfile );
    if( !os ) {
        std::cout << "error opening output file"  << std::endl;
        exit(1);
    }
    HepPID::writeVersion(os);

    get_list_size_( & isize );
    os << "      " << title << std::endl;
    os << " number of Herwig particles: " << isize << std::endl;

    for( i=1, iend=isize+1; i<iend; ++i ) {
        // get info from herwig
        for( j=0; j<10; ++j) { cname[j] = ’\0’; }
        get_herwig_name_( & i, & hid, cname );
        hname = std::string( cname );
        id = HepPID::translateHerwigtoPDT( hid );
        pn = HepPID::particleName( id );
        os << "Herwig: ";
        os.width(7);
        os << i ;
        os.width(12);
        os << hid << " " << hname;
        os << "  HepPID: " ;
        os.width(12);
        os << id << " " << pn << std::endl;
    }

    list_herwig_end_();

    return 0;
}

```

## 10.2 examListHerwigInterface.F

interface to some Herwig Fortran routines

```

      subroutine list_herwig_init(nevt)
c
c  initialization for the herwig C++ listing
c
#include "herwig65.inc"
      integer lnhwrt,lnhrd,lnhout,lnhdcy
      common/heplun/lnhwrt,lnhrd,lnhout,lnhdcy

      external hwudat
      integer n
      integer istr,nevt
C
C initialize HEP logical units
      lnhwrt=0
      lnhrd=0
      lnhdcy=0
      lnhout=22
      lhwout=lnhout
C      open(unit=lnhout,file='examHerwigToStdHep.lpt',status='new')
C
c      call hptrlsth
C
      return
      end

      subroutine get_list_size( isize )
c return the maximum size of herwig's particle list
#include "herwig65.inc"
      integer isize
      isize = NRES
      return
      end

       subroutine get_herwig_name( ihwg, id, name )
c ihwg is the index into herwig's short list
#include "herwig65.inc"
      integer id, ihwg
      character*8 name
      id = 0
      call HWUIDT(2,id,ihwg,name)
      return
      end

      subroutine list_herwig_end
      integer lnhwrt,lnhrd,lnhout,lnhdcy
      common/heplun/lnhwrt,lnhrd,lnhout,lnhdcy
C---terminate elementary process
c      call hwefin
C      close(unit=lnhout)
      return
      end

C----------------------------------------------------------------------
      subroutine hwabeg
C... user's routine for initialization
      end
      subroutine hwaend
C... user's routine for terminal calculations, histogram output, etc
      end
      subroutine hwanal
C... user's routine to analyse data from event
```

```
      end
C----------------------------------------------------------------------
```

## 10.3 examListIsajet.cc

List the Isajet particle ID translations

```
// -------------------------------------------------------------------
// examListIsajet.cc
//
// ----------------------------------------------------------------------

#include <fstream>
#include <iostream>
#include <cstdlib>
#include <cstring>

#include "HepPID/ParticleIDTranslations.hh"
#include "HepPID/ParticleName.hh"
#include "HepPID/Version.hh"

extern "C" {
    void list_isajet_init_ ( );
    void flavor_( int *, int *, int *, int *, int *, int * );
    void get_label_( int * id, char *name );
}

int main()
{
    static char outfile[] = "examListIsajet.out";
    std::string title = "HepPID listing of Isajet translations";

    // initialize isajet
    list_isajet_init_ ( );

    // open the output stream
    std::ofstream os( outfile );
    if( !os ) {
        std::cout << "error opening output file"  << std::endl;
        exit(1);
    }
    HepPID::writeVersion(os);

    os << "      " << title << std::endl;

    int i, j;
    int id, aid, fl1, fl2, fl3, js, indx;
    int pid;
    char cname[10];
    char acname[10];
    std::string hname;
    std::string pn;
    for( i=1; i<100005; ++i ) {
        // make sure names are empty
        for( j=0; j<10; ++j) { cname[j] = ’\0’; }
        for( j=0; j<10; ++j) { acname[j] = ’\0’; }
        // get info from isajet
        id = i;
        aid = 0;
        flavor_(&id, &fl1, &fl2, &fl3, &js, &indx );
        // we need both a valid index and a valid label
        // check the label only if there is a valid translation
        if ( indx > 0 ) {
           get_label_(&id,cname);
           aid = -id;
           get_label_(&aid,acname);
        } else {
           id = aid = 0;
        }

        // print particle
        if( id != 0 ) {
            pid = HepPID::translateIsajettoPDT( id );
            hname = std::string( cname );
            if ( pid != 0 ) {
                pn = HepPID::particleName( pid );
                os << "Isajet: ";
                os.width(10);
                os << id << " " << hname;
                os << "  HepPID: " ;
                os.width(12);
                os << pid << " " << pn << std::endl;
            } else if ( strncmp( cname, "ERR", 3 ) != 0 ) {
                os << "Isajet: ";
                os.width(10);
                os << id << " with name \"" << hname;
                os << "\" has no HepPID translation " << std::endl;
            }
        }
        // print antiparticle
        if( aid != 0 ) {
            hname = std::string( acname );
            pid = HepPID::translateIsajettoPDT( aid );
            if ( pid != 0 ) {
                pn = HepPID::particleName( pid );
                os << "Isajet: ";
                os.width(10);
                os << aid << " " << hname;
                os << "  HepPID: " ;
                os.width(12);
                os << pid << " " << pn << std::endl;
            } else if ( strncmp( acname, "ERR", 3 ) != 0 ) {
                os << "Isajet: ";
                os.width(10);
                os << aid << " with name \"" << hname;
                os << "\" has no HepPID translation " << std::endl;
            }
        }
    }

    return 0;
}
```

## 10.4 examListIsajetInterface.F

interface to some Isajet Fortran routines

```
      subroutine list_isajet_init
C
C     JTDKY = +/- unit number for decay table file.
C                 If it is negative, decay table is not printed.
C     JTEVT = +/- unit number for output event file.
C                 If it is negative, only stable particles are written on it.
C     JTCOM =     unit number for command file.
C     JTLIS =     unit number for listing.
C
      IMPLICIT NONE

C
      integer istr,nevt,itotal
      INTEGER JTDKY,JTEVT,JTCOM,JTLIS,IFL,ILOOP,IPRT,LOK,ILOOP2
      INTEGER INDEC,INDEC2
      CHARACTER*132 ISADEC
      LOGICAL OK,DONE
      SAVE ILOOP,JTDKY,JTEVT,JTCOM,JTLIS

      EXTERNAL ALDATA
C
C          Initialize ISAJET
C
      JTDKY=-1
      JTEVT=23
      JTCOM=21
      JTLIS=22
      nevt=1000
      itotal=0
C  point to standard decay tables
      CALL GETENV( ’ISAJET_DIR’, ISADEC )
      IF ( ISADEC .EQ. ’ ’ ) THEN
         ISADEC = ’isadecay.dat’
      ELSE
         INDEC   = INDEX ( ISADEC , ’ ’ )
         ISADEC(INDEC:INDEC+13)  = ’/isadecay.dat’
         INDEC2  = INDEX ( ISADEC , ’ ’ ) - 1
c         print *,’looking for ’,ISADEC(:INDEC2)
      ENDIF
      OPEN(UNIT=1,FILE=ISADEC,STATUS=’OLD’)
      OPEN(UNIT=JTLIS,FILE=’examListIsajet.lpt’,STATUS=’NEW’)
      CALL ISAINI(JTDKY,JTEVT,JTCOM,JTLIS)
      IPRT=1
C
C print list of defined particles and their translations
C    (you have to call ISAINI first)
C use bogus masses for 4th generation quarks so we can see their definitions
C
      CALL PRTLST(JTLIS,200.,220.)
C

      return
      END

      subroutine get_label( id, lb)

      integer id
      character*8 LB,LABEL

      lb = ’     ’

      if ( id.ne.0) then
```

```
         lb = LABEL(id)
      endif

      return
      end
```

## 10.5 examListPythia.cc

list Pythia particle ID translations

```
// ----------------------------------------------------------------------
// examListPythia.cc
//
// ----------------------------------------------------------------------

#include <fstream>
#include <string>
#include <cstdlib>

#include "HepPID/ParticleIDTranslations.hh"
#include "HepPID/ParticleIDMethods.hh"
#include "HepPID/ParticleName.hh"
#include "HepPID/Version.hh"

extern "C" {
    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()
{
    int kf,akf,pid,apid,iok;
    char cname[17],caname[17];
    std::string name, aname, pn, apn;
    const char outfile[] = "examListPythia.out";
    std::string title = "HepPID listing of Pythia translations";
    // open the output file
    std::ofstream os( outfile );
    if( !os ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    HepPID::writeVersion(os);
    // get Pythia listing using the fortran pylist function
    // write the output of pylist to examListPythia.lpt
    list_pythia_();

    os << "      " << title << std::endl;

    for(int i=1; i<501; ++i) {
       getkf_( &i, &kf);
       if( kf != 0 ) {
           getpyname_(&kf, &iok, cname);
           if( iok == 1 ) {
               cname[16]='\0';
               name = std::string( cname );
               pid = HepPID::translatePythiatoPDT( kf );
               pn = HepPID::particleName( pid );
               writeLine( i, kf, pid, name, pn, os );
               if( HepPID::isValid( -pid ) ) {
                 akf=-kf;
                 getpyname_(&akf, &iok, caname);
                 if( iok == 1 ) {
                     apid = HepPID::translatePythiatoPDT( akf );
                     apn = HepPID::particleName( apid );
                     caname[16]='\0';
                     aname = std::string( caname );
                     writeLine( i, akf, apid, aname, apn, os );
                 }
               }
           }
       }
    }
    return 0;
}

void writeLine( int & i, int & kf, int & id,
                std::string & name, std::string & pn, std::ofstream & os )
{
    os << "Pythia: ";
    os.width(7);
    os << i ;
    os.width(12);
    os << kf << " " << name;
    os << "  HepPID: " ;
    os.width(12);
    os << id << " " << pn << std::endl;
}
```

## 10.6 examListPythiaInterface.F

interface to some Pythia Fortran routines

```
      subroutine list_pythia
C
C     list jetset particle definitions
C
#include "pydat1.inc"

      INTEGER LNHOUT
C
      lnhout=22
      MSTU(11)=LNHOUT
      OPEN(UNIT=LNHOUT,FILE='examListPythia.lpt',STATUS='NEW')
C
C...
      call pylist(11)
C...enable pylist(12) if you need a new pythia decay table
C      call pylist(12)
C
C
c...close output file
      CLOSE(UNIT=LNHOUT)
      return
      END

      subroutine getkf(kc,kf)
#include "pydat2.inc"
      integer kf,kc
      kf = KCHG(KC,4)
      return
      END

      subroutine getpyname(kf,idef,chap)
#include "pydat2.inc"
      integer kf
      character*16 chap
      integer idef
      idef = 1
      CALL PYNAME(KF,CHAP)
      if(CHAP.EQ.' ') idef=0
      return
      END


```

## 10.7 examMyPDT.cc

create a custom PDT using our own definitions and write out the resulting PDT

```
// ----------------------------------------------------------------------
// examMyPDT.cc
//

#include "HepPDT/defs.h"
#include <fstream>

#include <cstdlib>
#include <string>

#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
#include "HepPDT/TempParticleData.hh"

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()
{
    const char outfile[] = "examMyPDT.out";
    // construct empty PDT
    HepPDT::ParticleDataTable datacol;
    {
        // Construct table builder
        HepPDT::TableBuilder  tb(datacol);
        // create my own particles here
        addData( tb,    "p+", 2212,  0.938,  +1.0,   -1,  .5 );
        addData( tb,     "d",    1,     0., -2./3,   -1,  .5 );
        addData( tb,    "u~",   -2,     0., -1./3,   -1,  .5 );
        addData( tb,    "W-",  -24, 80.396,  -1.0, 2.06, 1.0 );
        addData( tb, "gamma",   22,      0.,   0.,   -1, 1.0 );
        addData( tb, "badgamma",   122,      0.,   0.,   -1, 1.0 );
        tb.removeParticle( 122 );
    }   // the tb destructor fills datacol
    std::ofstream wpdfile( outfile );
    if( !wpdfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    datacol.writeParticleData(wpdfile);
    // access a particle
    // you get a null pointer if you reqest an undefined particle
    HepPDT::ParticleData * pd = datacol.particle( HepPDT::ParticleID(22) );
    if( pd )  {
        std::cout << "particle " << pd->name() << " is defined" << std::endl;
    } else {
        std::cout << "ERROR: particle is not in particle data table" << std::endl;
    }
    pd = datacol[ HepPDT::ParticleID(-24) ];
    // we expect this next line to produce an error
    std::cout << "the error is expected" << std::endl;
    if( datacol[ HepPDT::ParticleID(111) ] )  {
        std::cout << "particle " << datacol[ HepPDT::ParticleID(111) ]->name() << " is defined" << std:
    } else {
        std::cout << "ERROR: particle " << HepPDT::ParticleID(111).pid()
                  << " is not in particle data table" << std::endl;
    }

    return 0;
}

```

```
void addData( HepPDT::TableBuilder& tb, std::string const & name, int const id,
               double const mass,  double const charge, double const width,
               double const tspin )
{
    HepPDT::TempParticleData& tpd = tb.getParticleData( HepPDT::ParticleID( id ) );
    tpd.tempParticleName = name;
    tpd.tempCharge = charge;
    tpd.tempMass = HepPDT::Measurement( mass, 0. );
    tpd.tempSpin = HepPDT::SpinState( tspin, 0., 0. );
    tpd.tempWidth = HepPDT::Measurement( width, 0. );
    tb.addParticle( tpd );
}
```

## 10.8 listEvtGenNames.cc.in

The **HepPDT** (p. 25) tests are also useful examples

read EvtGen table and write out translation from EvtGen to **HepPDT** (p. 25)

```
// ----------------------------------------------------------------------
// listEvtGenNames.cc
// Author: Lynn Garren
//
// read EvtGen table and write out translation from EvtGen to HepPDT
//
// ----------------------------------------------------------------------

#include <fstream>
#include <iostream>

#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"

int main()
{
    const char infile1[] = "@top_srcdir@/data/pdt.table";
    const char infile2[] = "@top_srcdir@/data/DECAY.DEC";
    const char outfile[] = "listEvtGenNames.out";
    // open input files
    std::ifstream pdfile1( infile1 );
    if( !pdfile1 ) {
      std::cerr << "cannot open " << infile1 << std::endl;
      exit(-1);
    }
    std::ifstream pdfile2( infile2 );
    if( !pdfile2 ) {
      std::cerr << "cannot open " << infile2 << std::endl;
      exit(-1);
    }
    // construct PDT
    HepPDT::ParticleDataTable datacol( "EvtGen Table" );
    {
        // Construct table builder
        HepPDT::TableBuilder  tb(datacol);
        // read the input - put as many here as you want
        if( !addEvtGenParticles( pdfile1, tb ) ) { std::cout << "error reading EvtGen pdt file " << std
        if( !addEvtGenParticles( pdfile2, tb ) ) { std::cout << "error reading EvtGen decay file " << s
    }   // the tb destructor fills datacol
    // open output file
    std::ofstream wpdfile( outfile );
    if( !wpdfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    // write a translation list
    datacol.writeParticleTranslation( wpdfile );

    return 0;
}
```

## 10.9 listEvtGenTranslation.cc

The **HepPID** (p. 39) tests are also useful examples

write the **HepPID** (p. 39) EvtGen translations

```
// ----------------------------------------------------------------------
// listEvtGenTranslation.cc
// Author: Lynn Garren
//
// Usage:  listEvtGenTranslation
//
// ----------------------------------------------------------------------

#include <fstream>
#include <iostream>
#include <cstdlib>     // for exit

#include "HepPID/ParticleIDTranslations.hh"

int main()
{
    const char outfile[] = "listEvtGenTranslation.out";
    // open the output file
    std::ofstream wpdfile( outfile );
    if( !wpdfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    // write the particle names
    HepPID::writeEvtGenTranslation( wpdfile );
}
```

## 10.10 listHerwigTranslation.cc

The **HepPID** (p. 39) tests are also useful examples

write the **HepPID** (p. 39) Herwig translations

```
// ----------------------------------------------------------------------
// listHerwigTranslation.cc
// Author: Lynn Garren
//
// Usage:  listHerwigTranslation
//
// ----------------------------------------------------------------------

#include <fstream>
#include <iostream>
#include <cstdlib>     // for exit

#include "HepPID/ParticleIDTranslations.hh"

int main()
{
    const char outfile[] = "listHerwigTranslation.out";
    // open the output file
    std::ofstream wpdfile( outfile );
    if( !wpdfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    // write the particle names
    HepPID::writeHerwigTranslation( wpdfile );
}
```

## 10.11 listIsajetTranslation.cc

The **HepPID** (p. 39) tests are also useful examples

write the **HepPID** (p. 39) Isajet translations

```
// ----------------------------------------------------------------------
// listIsajetTranslation.cc
// Author: Lynn Garren
//
// Usage:  listIsajetTranslation
//
// ----------------------------------------------------------------------

#include <fstream>
#include <iostream>
#include <cstdlib>     // for exit

#include "HepPID/ParticleIDTranslations.hh"

int main()
{
    const char outfile[] = "listIsajetTranslation.out";
    // open the output file
    std::ofstream wpdfile( outfile );
    if( !wpdfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    // write the particle names
    HepPID::writeIsajetTranslation( wpdfile );
}
```

## 10.12 listParticleNames.cc

The **HepPID** (p. 39) tests are also useful examples

list all known **HepPID** (p. 39) particle names

```
// ----------------------------------------------------------------------
// listParticleNames.cc
// Author: Lynn Garren
//
// list all known HepPID particle names
//
// Usage:  listParticleNames
//
// ----------------------------------------------------------------------

#include <fstream>
#include <iostream>
#include <cstdlib>     // for exit

#include "HepPID/ParticleName.hh"

int main()
{
    const char outfile[] = "listParticleNames.out";
    // open the output file
    std::ofstream wpdfile( outfile );
    if( !wpdfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    // write the particle names
    HepPID::listParticleNames( wpdfile );
}
```

## 10.13 listPDGNames.cc.in

The **HepPDT** (p. 25) tests are also useful examples

read PDG table and write out translation to **HepPDT** (p. 25)

```
// ----------------------------------------------------------------------
// listPDGNames.cc
// Author: Lynn Garren
//
// read PDG table and write out translation to HepPDT
//
// ----------------------------------------------------------------------

#include <fstream>
#include <iostream>

#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"

int main()
{
    const char infile[] = "@top_srcdir@/data/mass_width_2006.mc";
    const char outfile[] = "listPDGNames.out";
    // open input file
    std::ifstream pdfile( infile );
    if( !pdfile ) {
      std::cerr << "cannot open " << infile << std::endl;
      exit(-1);
    }
    // construct empty PDT
    HepPDT::ParticleDataTable datacol( "PDG Table" );
    {
        // Construct table builder
        HepPDT::TableBuilder  tb(datacol);
        // read the input - put as many here as you want
        if( !addPDGParticles( pdfile, tb ) )
          { std::cout << "error reading PDG file " << std::endl; }
    }   // the tb destructor fills datacol
    // open output file
    std::ofstream wpdfile( outfile );
    if( !wpdfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    // write a translation list
    datacol.writeParticleTranslation( wpdfile );

    return 0;
}
```

## 10.14 listPDGTranslation.cc

The **HepPID** (p. 39) tests are also useful examples

write the **HepPID** (p. 39) PDG translations

```
// ----------------------------------------------------------------------
// listPDGTranslation.cc
// Author: Lynn Garren
//
// Usage:  listPDGTranslation
//
// ----------------------------------------------------------------------

#include <fstream>
#include <iostream>
#include <cstdlib>     // for exit

#include "HepPID/ParticleIDTranslations.hh"

int main()
{
    const char outfile[] = "listPDGTranslation.out";
    // open the output file
    std::ofstream wpdfile( outfile );
    if( !wpdfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    // write the particle names
    HepPID::writePDGTranslation( wpdfile );
}
```

## 10.15 listPythiaNames.cc.in

The **HepPDT** (p. 25) tests are also useful examples

read Pythia table and write out translation from Pythia to **HepPDT** (p. 25)

```
// ----------------------------------------------------------------------
// listPythiaNames.cc
// Author: Lynn Garren
//
// read Pythia table and write out translation from pythia to HepPDT
//
// ----------------------------------------------------------------------

#include <fstream>
#include <iostream>

#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"

int main()
{
    const char infile[] = "@srcdir@/listPythia.tbl";
    const char outfile[] = "listPythiaNames.out";
    // open input file
    std::ifstream pdfile( infile );
    if( !pdfile ) {
      std::cerr << "cannot open " << infile << std::endl;
      exit(-1);
    }
    // construct empty PDT
    HepPDT::ParticleDataTable datacol( "Pythia Table" );
    {
        // Construct table builder
        HepPDT::TableBuilder  tb(datacol);
        // read the input - put as many here as you want
        if( !addPythiaParticles( pdfile, tb ) )
          { std::cout << "error reading pythia file " << std::endl; }
    }   // the tb destructor fills datacol
    // open output file
    std::ofstream wpdfile( outfile );
    if( !wpdfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    // write a translation list
    datacol.writeParticleTranslation( wpdfile );

    return 0;
}
```

## 10.16 listPythiaTranslation.cc

The **HepPID** (p. 39) tests are also useful examples

write the **HepPID** (p. 39) Pythia translations

```
// ----------------------------------------------------------------------
// listPythiaTranslation.cc
// Author: Lynn Garren
//
// Usage:  listPythiaTranslation
//
// ----------------------------------------------------------------------

#include <fstream>
#include <iostream>
#include <cstdlib>     // for exit

#include "HepPID/ParticleIDTranslations.hh"

int main()
{
    const char outfile[] = "listPythiaTranslation.out";
    // open the output file
    std::ofstream wpdfile( outfile );
    if( !wpdfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    // write the particle names
    HepPID::writePythiaTranslation( wpdfile );
}
```

## 10.17 listQQTranslation.cc

The **HepPID** (p. 39) tests are also useful examples

write the **HepPID** (p. 39) QQ translations

```
// ----------------------------------------------------------------------
// listQQTranslation.cc
// Author: Lynn Garren
//
// Usage:  listQQTranslation
//
// ----------------------------------------------------------------------

#include <fstream>
#include <iostream>
#include <cstdlib>     // for exit

#include "HepPID/ParticleIDTranslations.hh"

int main()
{
    const char outfile[] = "listQQTranslation.out";
    // open the output file
    std::ofstream wpdfile( outfile );
    if( !wpdfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    // write the particle names
    HepPID::writeQQTranslation( wpdfile );
}
```

## 10.18 testHepPDT.cc

The **HepPDT** (p. 25) tests are also useful examples

Test by reading the PDG table. Get filename and location of PDG table from input stream.

```
// ----------------------------------------------------------------------
// testHepPDT.cc
// Author: Lynn Garren
//
// test by reading the PDG table
// get filename and location of PDG table from input stream
//
// Usage:  testHepPDT
//
// ----------------------------------------------------------------------

#include <fstream>
#include <iomanip>

#include "HepPDT/defs.h"
#include "HepPDT/TableBuilder.hh"
#include "HepPDT/ParticleDataTable.hh"
#include "HepPDT/HeavyIonUnknownID.hh"
// local include
#include "TestNuclearFragment.hh"

void pdtSimpleTest( char[300], std::ofstream & );
void pdtFragmentTest( char[300], std::ofstream & );
void duplicateFragmentTest( char[300], std::ofstream & );
void testPDMethods( HepPDT::ParticleDataTable&, std::ofstream & );

int main()
{
    char pdgfile[300] = "";
    const char outfile[] = "testHepPDT.out";
    std::cin >> pdgfile;
    // open output file
    std::ofstream wpdfile( outfile );
    if( !wpdfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }

    // construct a default PDT
    pdtSimpleTest( pdgfile, wpdfile );
    // now test the nuclear fragment option
    pdtFragmentTest( pdgfile, wpdfile );
    // check how we deal with duplicate fragments
    duplicateFragmentTest( pdgfile, wpdfile );

    return 0;
}

void pdtSimpleTest( char pdgfile[300], std::ofstream & wpdfile )
{
    // open input file
    std::ifstream pdfile( pdgfile );
    if( !pdfile ) {
      std::cerr << "cannot open " << pdgfile << std::endl;
      exit(-1);
    }
    // construct empty PDT
    HepPDT::ParticleDataTable datacol( "2006 PDG Table" );
    {
        // Construct table builder
        HepPDT::TableBuilder  tb(datacol);
        // read the input - put as many here as you want
        if( !HepPDT::addPDGParticles( pdfile, tb ) ) {
            std::cout << "error reading PDG file " << std::endl;
        }
    }   // the tb destructor fills datacol
    // done with pdfile, so close it
    pdfile.close();

    const char outfile1[] = "testHepPDTtable.out";
    // open output file
    std::ofstream wpdt1( outfile1 );
    if( !wpdt1 ) {
      std::cerr << "cannot open " << outfile1 << std::endl;
      exit(-1);
    }
    datacol.writeParticleData(wpdt1);

    wpdfile << std::endl;

    // output some pion information
    HepPDT::ParticleData * pd;
    pd=datacol.particle(HepPDT::ParticleID(111));
    // test the ResonanceStructure cutoff methods here
    if(pd) {
        pd->write(wpdfile);
        wpdfile << "Resonance info for 111 "
                << pd->totalWidth().value() << " "
                << pd->totalWidth().sigma() << " "
                << pd->lowerCutoff() << " "
                << pd->upperCutoff() << std::endl;
    }
    //  -111 is an illegal particle, no info will be written
    pd=datacol.particle(HepPDT::ParticleID(-111));
    if(pd) pd->write(wpdfile);
    pd=datacol.particle(HepPDT::ParticleID(211));
    if(pd) pd->write(wpdfile);
    // string lookup
    pd=datacol.particle(std::string("pi0"));
    if(pd) pd->write(wpdfile);

    // particle info
    datacol.writeParticleInfo(wpdfile);

    testPDMethods( datacol, wpdfile );
}

void pdtFragmentTest( char pdgfile[300], std::ofstream & wpdfile )
{
    wpdfile << std::endl;
    wpdfile << " Begin test of HeavyIonUnknownID " << std::endl;
    // reopen input file
    std::ifstream pdfile2( pdgfile );
    if( !pdfile2 ) {
      std::cerr << "cannot open " << pdgfile << std::endl;
      exit(-1);
    }
    // construct another PDT instance that knows how to deal with unknown heavy ions
    // NOTE: normally you would construct a single ParticleDataTable with this option
    HepPDT::ParticleDataTable pdt2( "Handle Heavy Ions",
                                    new HepPDT::HeavyIonUnknownID );
    {
        // Construct table builder
        HepPDT::TableBuilder  tb2(pdt2);
        // read the input - put as many here as you want
        if( !HepPDT::addPDGParticles( pdfile2, tb2 ) ) {
            std::cout << "error reading PDG file " << std::endl;
        }
    }
    // done with pdfile, so close it
    pdfile2.close();

    // try a heavy ion
    HepPDT::ParticleData * pd=pdt2.particle(HepPDT::ParticleID(1000020040));
    wpdfile << " Printing information for unknown nuclear fragment "
            << std::endl;
    if(pd) pd->write(wpdfile);
}

void duplicateFragmentTest( char pdgfile[300], std::ofstream & wpdfile )
{
    wpdfile << std::endl;
    wpdfile << " Begin test of duplicate nuclear fragments " << std::endl;
    // reopen input file
    std::ifstream pdfile2( pdgfile );
    if( !pdfile2 ) {
      std::cerr << "cannot open " << pdgfile << std::endl;
      exit(-1);
    }
    // this test checks to see if we have actually added a fragment to the table
    HepPDT::ParticleDataTable pdt( "Duplicate Nuclear Fragments",
                                    new HepPDT::TestNuclearFragment );
    {
        // Construct table builder
        HepPDT::TableBuilder  tb2(pdt);
        // read the input - put as many here as you want
        if( !HepPDT::addPDGParticles( pdfile2, tb2 ) ) {
            std::cout << "error reading PDG file " << std::endl;
        }
    }
    // done with pdfile, so close it
    pdfile2.close();

    // try a heavy ion
    wpdfile << " Printing information for unknown nuclear fragments "
            << std::endl;
    HepPDT::ParticleData * pd=pdt.particle(HepPDT::ParticleID(1000020040));
    if(pd) pd->write(wpdfile);
    pd=pdt.particle(HepPDT::ParticleID(1000020040));
    if(pd) pd->write(wpdfile);
    pd=pdt.particle(HepPDT::ParticleID(1000010040));
    if(pd) pd->write(wpdfile);
    pd=pdt.particle(HepPDT::ParticleID(1000020040));
    if(pd) pd->write(wpdfile);
    // what is the state of the table?
    const char outfile2[] = "testHepPDTfragment.out";
    std::ofstream wpdt( outfile2 );
    if( !wpdt ) {
      std::cerr << "cannot open " << outfile2 << std::endl;
      exit(-1);
    }
    pdt.writeParticleData(wpdt);

    // check isStable
    const char outfile3[] = "testHepPDTstatus.out";
    std::ofstream wpdt3( outfile3 );
    if( !wpdt3 ) {
      std::cerr << "cannot open " << outfile3 << std::endl;
      exit(-1);
    }
    pdt.writeParticleStatus(wpdt3);
}

void testPDMethods( HepPDT::ParticleDataTable& datacol, std::ofstream & wpdfile )
{
    wpdfile << std::endl;
    wpdfile << "Begin test of ParticleData methods " << std::endl;
    HepPDT::ParticleData * pd;
    int id[16] = { 5, 24, 15, 213, 3214, 10213, 9050225, 541, 129050225,
                   2000025, 3101, 3301, 2212, 1000020040, 1000060120, 555 };
    int it;
    for( it=0; it < 16; it++ ) {
        pd=datacol.particle(HepPDT::ParticleID(id[it]));
        if(pd) {
            if( pd->hasUp() ) {
                wpdfile << "Particle " << pd->name()  << " " << pd->pid()
                          << " has an up quark" << std::endl;
            }
            if( pd->hasDown() ) {
                wpdfile << "Particle " << pd->name()  << " " << pd->pid()
                          << " has a down quark" << std::endl;
            }
            if( pd->hasStrange() ) {
                wpdfile << "Particle " << pd->name()  << " " << pd->pid()
                          << " has a strange quark" << std::endl;
            }
            if( pd->hasCharm() ) {
                wpdfile << "Particle " << pd->name() << " " << pd->pid()
                          << " has a charmed quark" << std::endl;
            }
            if( pd->hasBottom() ) {
                wpdfile << "Particle " << pd->name() << " " << pd->pid()
                          << " has a bottom quark" << std::endl;
            }
            if( pd->hasTop() ) {
                wpdfile << "Particle " << pd->name() << " " << pd->pid()
                          << " has a top quark" << std::endl;
            }
        }
    }
}
```

## 10.19 testPID.cc

The **HepPDT** (p. 25) tests are also useful examples

test ParticleID methods

```
// ----------------------------------------------------------------------
// TestPID.cc
// Author: Lynn Garren
//
// test ParticleID
//
// ----------------------------------------------------------------------

#include <iostream>
#include <iomanip>

#include "HepPDT/defs.h"
#include "HepPDT/ParticleID.hh"

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()
{
    int id[16] = { 5, 25, 15, 213, -3214, 10213, 9050225, -200543, 129050225,
                   2000025, 3101, 3301, -2212, 1000020040, -1000060120, 555 };
    int it;
    int nr, nx;
    int chg, sid, extra;
    int js, ls;
    for( it=0; it < 16; it++ ) {
        HepPDT::ParticleID pid( id[it] );
        nx = pid.digit(HepPDT::n);
        nr = pid.digit(HepPDT::nr);
        extra = pid.extraBits();
        std::cout << std::endl;
        std::cout << std::setw(18) << id[it] << ": " << nx << " " << nr
                     << " " << pid.digit(HepPDT::nl)
                     << " " << pid.digit(HepPDT::nq1) << " "
                     << pid.digit(HepPDT::nq2) << " " << pid.digit(HepPDT::nq3)
                     << " " <<  pid.digit(HepPDT::nj)
                     << " extra bits " << extra << std::endl;
        js = pid.jSpin();
        HepPDT::Quarks cqks = pid.quarks( );
        ls = pid.lSpin();
        sid = pid.fundamentalID();
        chg = pid.threeCharge();
        if( !pid.isValid() ) {
           std::cout << "**** Invalid PID: " << pid.pid()
                     << " ****" << std::endl;
        } else  {
          testHadron( pid, id[it] );
          testLepton( pid, id[it] );
          testDiQuark( pid, id[it] );
          testNucleus( pid, id[it] );
          testUnknown( pid, id[it] );
        }
        testValid( pid );
        testHasMethods( pid, id[it] );
    }
}
```

```

void testHadron( HepPDT::ParticleID& pid, int& idn )
{
        int nx = pid.digit(HepPDT::n);
        int nr = pid.digit(HepPDT::nr);
        int extra = pid.extraBits();
        int js = pid.jSpin();
        int ls = pid.lSpin();
        int sid = pid.fundamentalID();
        int chg = pid.threeCharge();
        HepPDT::Quarks cqks = pid.quarks( );
          if( pid.isHadron() ) {
            if( pid.isMeson() ) {
               std::cout << "meson   " << std::setw(10) << idn << ": " << nx
                         << " " << nr << " "  << ls << " "
                         << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
                         << " " << js << " " << sid << " " << chg
                         << " extra bits " << extra << std::endl;
            } else if( pid.isBaryon() ) {
               std::cout << "baryon  " << std::setw(10) << idn << ": " << nx
                         << " " << nr << " "  << ls << " "
                         << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
                         << " " << js << " " << sid << " " << chg
                         << " extra bits " << extra << std::endl;
            } else {
                std::cout << "**** undefined hadron: " << pid.pid()
                          << " ****" << std::endl;
            }
          }
}

void testLepton( HepPDT::ParticleID& pid, int& idn )
{
        HepPDT::Quarks cqks = pid.quarks( );
          if( pid.isLepton() ) {
           std::cout << "lepton  " << std::setw(10) << idn
                     << ": " << pid.digit(HepPDT::n)
                     << " "  << pid.digit(HepPDT::nr)
                     << " "  << pid.lSpin() << " "
                     << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
                     << " " <<  pid.jSpin()
                     << " " <<  pid.fundamentalID()
                     << " " <<  pid.threeCharge()
                     << " extra bits " <<  pid.extraBits() << std::endl;
          }
}

void testDiQuark( HepPDT::ParticleID& pid, int& idn )
{
        HepPDT::Quarks cqks = pid.quarks( );
          if( pid.isDiQuark() ) {
           std::cout << "diquark " << std::setw(10) << idn
                     << ": " << pid.digit(HepPDT::n)
                     << " "  << pid.digit(HepPDT::nr)
                     << " "  << pid.lSpin() << " "
                     << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
                     << " " <<  pid.jSpin()
                     << " " <<  pid.fundamentalID()
                     << " " <<  pid.threeCharge()
                     << " extra bits " <<  pid.extraBits() << std::endl;
          }
}

void testNucleus( HepPDT::ParticleID& pid, int& idn )
{
        HepPDT::Quarks cqks = pid.quarks( );
          if( pid.isNucleus() ) {
           std::cout << "ion    " << std::setw(11) << idn
                     << ": " << pid.digit(HepPDT::n10)
                     << " " << std::setw(3) << pid.A()
                     << " " << std::setw(3) << pid.Z()
                     << " " << std::setw(3) << pid.lambda()
                     << " " <<  pid.jSpin()
                     << " " <<  pid.fundamentalID()
                     << " " <<  pid.threeCharge()
                     << " extra bits " <<  pid.extraBits() << std::endl;
          }
}

void testUnknown( HepPDT::ParticleID& pid, int& idn )
{
        HepPDT::Quarks cqks = pid.quarks( );
          if( pid.isHadron() || pid.isLepton() ||
              pid.isDiQuark() || pid.isNucleus() ) {
          } else {
           std::cout << "unknown " << std::setw(10) << idn
                     << ": " << pid.digit(HepPDT::n)
                     << " "  << pid.digit(HepPDT::nr)
                     << " "  << pid.lSpin() << " "
                     << cqks.nq1 << " " << cqks.nq2 << " " << cqks.nq3
                     << " " <<  pid.jSpin()
                     << " " <<  pid.fundamentalID()
                     << " " <<  pid.threeCharge()
                     << " extra bits " <<  pid.extraBits() << std::endl;
          }
}

void testValid( HepPDT::ParticleID& pid )
{
        if( pid.isValid() ) {
            int js = pid.jSpin();
            int chg = pid.threeCharge();
            std::cout << "total spin: " << js << " " << HepPDT::spinitod(js) ;
            std::cout << " orbital angular momentum: " <<  pid.lSpin() ;
            std::cout << " spin: " <<  pid.sSpin() ;
            std::cout << " charge: " << chg
                      << " " << double(chg)/3.0 << std::endl;
            HepPDT::Quarks qlist = pid.quarks( );
            std::cout <<  "quarks: " << qlist.nq1
                      << " " << qlist.nq2 << " " << qlist.nq3 << std::endl;
        }
}

void testHasMethods( HepPDT::ParticleID& pid, int& idn )
{
  if( pid.hasUp() ) {
      std::cout << "Particle " << pid.PDTname()  << " " << pid.pid()
                << " has an up quark" << std::endl;
  }
  if( pid.hasDown() ) {
      std::cout << "Particle " << pid.PDTname()  << " " << pid.pid()
                << " has a down quark" << std::endl;
  }
  if( pid.hasStrange() ) {
      std::cout << "Particle " << pid.PDTname()  << " " << pid.pid()
                << " has a strange quark" << std::endl;
  }
  if( pid.hasCharm() ) {
      std::cout << "Particle " << pid.PDTname() << " " << pid.pid()
                << " has a charmed quark" << std::endl;
  }
  if( pid.hasBottom() ) {
      std::cout << "Particle " << pid.PDTname() << " " << pid.pid()
                << " has a bottom quark" << std::endl;
  }
  if( pid.hasTop() ) {
      std::cout << "Particle " << pid.PDTname() << " " << pid.pid()
                << " has a top quark" << std::endl;
  }
}

```

## 10.20 testReadEvtGen.cc.in

The **HepPDT** (p. 25) tests are also useful examples

read EvtGen table and write it out

```
// $Id: testReadEvtGen.cc.in,v 1.3 2007/09/14 19:53:21 garren Exp $
// ----------------------------------------------------------------------
// testReadEvtGen.cc
//
// read EvtGen table and write it out
//
// ----------------------------------------------------------------------

#include <fstream>

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

int main()
{
    const char infile1[] = "@top_srcdir@/data/pdt.table";
    const char infile2[] = "@top_srcdir@/data/DECAY.DEC";
    const char outfile[] = "testReadEvtGen.out";
    // open input files
    std::ifstream pdfile1( infile1 );
    if( !pdfile1 ) {
      std::cerr << "cannot open " << infile1 << std::endl;
      exit(-1);
    }
    // construct empty PDT
    std::ifstream pdfile2( infile2 );
    if( !pdfile2 ) {
      std::cerr << "cannot open " << infile2 << std::endl;
      exit(-1);
    }
    HepPDT::ParticleDataTable datacol( "EvtGen Table" );
    {
        // Construct table builder
        HepPDT::TableBuilder  tb(datacol);
        // read the input - put as many here as you want
        if( !addEvtGenParticles( pdfile1, tb ) ) { std::cout << "error reading EvtGen pdt file " << std
        if( !addEvtGenParticles( pdfile2, tb ) ) { std::cout << "error reading EvtGen decay file " << s
    }   // the tb destructor fills datacol
    std::ofstream wfile( outfile );
    if( !wfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    datacol.writeParticleData(wfile);

    return 0;
}
```

## 10.21 testReadIsajet.cc.in

The **HepPDT** (p. 25) tests are also useful examples

read the isajet particle and decay tables for testing purposes Note that isaparticles.dat was created with PRTLST(...)

```
// ----------------------------------------------------------------------
// readIsajet.cc
// Author: Lynn Garren
//
// read the isajet particle table for testing purposes
// note that isaparticles.dat was created with PRTLST(...)
//
// Usage:  readIsajet
//
// ----------------------------------------------------------------------

#include <fstream>

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

int main()
{
    char pdgfile[300] = "@top_srcdir@/examples/data/isaparticles.dat";
    const char outfile[] = "testReadIsajet.out";
    // construct empty PDT
    HepPDT::ParticleDataTable datacol( "Isajet Table" );
    {
        // open input files
        std::ifstream pdfile( pdgfile );
        if( !pdfile ) {
          std::cerr << "cannot open " << pdgfile << std::endl;
          exit(-1);
        }
        // Construct table builder
        HepPDT::TableBuilder  tb(datacol);
        // read the input - put as many here as you want
        if( !HepPDT::addIsajetParticles( pdfile, tb ) ) {
            std::cout << "error reading " << pdgfile << std::endl;
        }
    }   // the tb destructor fills datacol
    std::ofstream wpdfile( outfile );
    if( !wpdfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    datacol.writeParticleData(wpdfile);
    wpdfile << std::endl;

    // particle info
    datacol.writeParticleInfo(wpdfile);

    return 0;
}
```

## 10.22 testReadParticleTable.cc.in

The **HepPDT** (p. 25) tests are also useful examples

read particle.tbl and write it out Also write out a list of which particles are stable

```
// $Id: testReadParticleTable.cc.in,v 1.5 2008/10/24 16:41:40 garren Exp $
// ----------------------------------------------------------------------
// testReadParticleTable.cc
//
// read particle.tbl and write it out
//
// ----------------------------------------------------------------------

#include <fstream>

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

int main()
{
    const char infile[] = "@top_srcdir@/data/particle.tbl";
    const char outfile[] = "testReadParticleTable.out";
    // open input files
    std::ifstream pdfile( infile );
    if( !pdfile ) {
      std::cerr << "cannot open " << infile << std::endl;
      exit(-1);
    }
    // construct empty PDT
    HepPDT::ParticleDataTable datacol( "Generic Particle Table" );
    {
        // Construct table builder
        HepPDT::TableBuilder  tb(datacol);
        // read the input - put as many here as you want
        // bool  addParticleTable( std::istream&, TableBuilder&,
        //                         bool validate = false );
        // where:  validate=true => verify that the ParticleID is valid
        if( !addParticleTable( pdfile, tb, true ) ) {
            std::cout << "error reading EvtGen pdt file " << std::endl;
        }
    }   // the tb destructor fills datacol
    // open the output stream
    std::ofstream wfile( outfile );
    if( !wfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    // write the data table
    datacol.writeParticleData(wfile);
    // try some heavy ions
    wfile << std::endl;
    wfile << std::endl;
    HepPDT::ParticleData * pd;
    pd=datacol.particle(HepPDT::ParticleID(1000020040));
    if(pd) pd->write(wfile);
    pd=datacol.particle(HepPDT::ParticleID(1000050110));
    if(pd) pd->write(wfile);

    // check isStable
    const char outfile3[] = "testReadParticleTableStatus.out";
    std::ofstream wpdt3( outfile3 );
    if( !wpdt3 ) {
      std::cerr << "cannot open " << outfile3 << std::endl;
      exit(-1);
    }
    datacol.writeParticleStatus(wpdt3);

    return 0;
}
```

## 10.23 testReadQQ.cc.in

The **HepPDT** (p. 25) tests are also useful examples

read QQ table and write it out

```
// $Id: testReadQQ.cc.in,v 1.1 2007/05/22 22:12:23 garren Exp $
// ----------------------------------------------------------------------
// testReadQQ.cc
//
// read QQ table and write it out
//
// ----------------------------------------------------------------------

#include <fstream>

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

int main()
{
    const char infile[] = "@srcdir@/listQQ.dec";
    const char outfile[] = "testReadQQ.out";
    // open input file
    std::ifstream pdfile( infile );
    if( !pdfile ) {
      std::cerr << "cannot open " << infile << std::endl;
      exit(-1);
    }
    // construct empty PDT
    HepPDT::ParticleDataTable datacol( "QQ Table" );
    {
        // Construct table builder
        HepPDT::TableBuilder  tb(datacol);
        // read the input - put as many here as you want
        if( !addQQParticles( pdfile, tb ) )
             { std::cout << "error reading QQ table file " << std::endl; }
    }   // the tb destructor fills the PDT
    std::ofstream wpdfile( outfile );
    if( !wpdfile ) {
      std::cerr << "cannot open " << outfile << std::endl;
      exit(-1);
    }
    // write a translation list
    datacol.writeParticleTranslation( wpdfile );
    // write the particle and decay info
    datacol.writeParticleData( wpdfile );

    return 0;
}
```

# Index