Reference Manual

2.3.1pre-

Generated by Doxygen 1.8.13

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

Namespace Documentation

4.1 zstr Namespace Reference

Typedefs

- typedef std::ifstream **ifstream**
- typedef std::ofstream **ofstream**

4.1.1 Detailed Description

This namespace defines wrappers for std::ifstream, std::ofstream, and std::fstream objects. The wrappers perform the following steps:

- check the open modes make sense
- check that the call to open() is successful
- (for input streams) check that the opened file is peek-able
- turn on the badbit in the exception mask

Chapter 5

Class Documentation

5.1 Alphas Class Reference

Static Public Member Functions

- static void **SetMz** (double Mz)
- static double **GetMz** ()
- static void **SetAlphasMz** (double alphas)
- static double **GetAlphasMz** ()
- static void **SetNf** (int nf)
- static int **GetNf** ()
- static void **SetNLoop** (int nLoop)
- static int **GetNLoop** ()
- static void **SetFlavorMatchingOn** (bool FlavorMatching)
- static bool **GetFlavorMatchingOn** ()
- static void **SetFlavorMatchingThresholds** (double th1, double th2, double th3, double th4, double th5, double th6)
- static void **GetFlavorMatchingThresholds** (double &th1, double &th2, double &th3, double &th4, double &th5, double &th6)
- static double **CalcAlphasMu** (double mu, double alphasMz=0, int nLoop=0, int nFlavors=0)
- static double **CalcAlphasMuFixedNf** (double mu, int nf)
- static int **CalcNf** (double mu)
- static void **PrintInfo** ()

Static Public Attributes

- static double **fMz** = 91.1876
- static double **fAlphasMz** = 0.1184
- static int **fNf** = 5
- static int **fnLoop** = 2
- static bool **bFlavorMatching** = false
- static double **fTh** [6] = {0.0023, 0.0048, 0.095, 1.275, 4.18, 173.5}

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

- fastnlotoolkit/include/fastnlotk/Alphas.h
- fastnlotoolkit/Alphas.cc

5.2 AsmMS Struct Reference

Public Attributes

- double **Asexact**
- double **mMSexact**

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

- fastnlotoolkit/include/fastnlotk/CRunDec.h

5.3 CRunDec Class Reference

Public Member Functions

- **CRunDec** (int)
- int **GetNf** ()
- void **SetNf** (int nf)
- double **LamExpl** (double asmu, double mu, int nloops)
- double **LamImpl** (double asmu, double mu, int nloops)
- double **AlphasLam** (double Lambda, double mu, int nloops)
- double **AlphasExact** (double asmu0, double mu0, double mu1, int nloops)
- double **mMS2mMS** (double mu0, double asmu0, double asmu1, int nloops)
- double **mMS2mOS** (double MS, double mq[4], double asmu, double mu, int nloops)
- double **mOS2mMS** (double mOS, double mq[ ], double asmu, double mu, int nloops)
- double **mMS2mSI** (double mMS, double asmu, double mu, int nloops)
- double **mRI2mMS** (double mRI, double asmu, int nloops)
- double **mMS2mRGI** (double mMS, double asmu, int nloops)
- double **mRGI2mMS** (double mRGI, double asmu, int nloops)
- double **mOS2mSI** (double mOS, double mq[ ], double asM, int nloops)
- double **mOS2mMSrun** (double mOS, double mq[ ], double asmu, double mu, int nloops)
- double **mMS2mOSrun** (double mMS, double mq[ ], double asmu, double mu, int nloops)
- double **mMS2mRI** (double mMS, double asmu, int nloops)
- double **mOS2mMSit** (double mOS, double mq[ ], double asmu, double mu, int nloops)
- double **mMS2mRGImod** (double mMS, double asmu, int nloops)
- AsmMS **AsmMSrunexact** (double mmu, double asmu0, double mu0, double mu1, int nloops)
- double **DecAsDownOS** (double asmu, double massth, double muth, int nloops)
- double **DecAsUpOS** (double asmu, double massth, double muth, int nloops)
- double **DecMqUpOS** (double mq, double asmu, double massth, double muth, int nloops)
- double **DecMqDownOS** (double mq, double asmu, double massth, double muth, int nloops)
- double **AIL2AIH** (double asl, double mu1, TriplenfMmu decpar[ ], double mu2, int nloops)
- double **AIH2AIL** (double ash, double mu1, TriplenfMmu decpar[ ], double mu2, int nloops)
- double **mL2mH** (double mql, double asl, double mu1, TriplenfMmu decpar[ ], double mu2, int nloops)
- double **mH2mL** (double mqh, double ash, double mu1, TriplenfMmu decpar[ ], double mu2, int nloops)
- double **LamExpl** (double asmu, double mu, int nf, int nloops)
- double **LamImpl** (double asmu, double mu, int nf, int nloops)
- double **AlphasLam** (double Lambda, double mu, int nf, int nloops)
- double **AlphasExact** (double asmu0, double mu0, double mu1, int nf, int nloops)
- double **mMS2mMS** (double mu0, double asmu1, double asmu0, int nf, int nloops)
- AsmMS **AsmMSrunexact** (double mmu, double asmu0, double mu0, double mu1, int nf, int nloops)
- double **mMS2mOS** (double MS, double mq[ ], double asmu, double mu, int nf, int nloops)

- double **mOS2mMS** (double mOS, double mq[ ], double asmu, double mu, int nf, int nloops)
- double **mMS2mSI** (double mMS, double asmu, double mu, int nf, int nloops)
- double **mRI2mMS** (double mRI, double asmu, int nf, int nloops)
- double **mMS2mRGI** (double mMS, double asmu, int nf, int nloops)
- double **mRGI2mMS** (double mRGI, double asmu, int nf, int nloops)
- double **mOS2mSI** (double mOS, double mq[ ], double asM, int nf, int nloops)
- double **mOS2mMSrun** (double mOS, double mq[ ], double asmu, double mu, int nf, int nloops)
- double **mMS2mOSrun** (double mMS, double mq[ ], double asmu, double mu, int nf, int nloops)
- double **mMS2mRI** (double mMS, double asmu, int nf, int nloops)
- double **mOS2mMSit** (double mOS, double mq[ ], double asmu, double mu, int nf, int nloops)
- double **mMS2mRGImod** (double mMS, double asmu, int nf, int nloops)
- double **DecAsDownOS** (double asmu, double massth, double muth, int nf, int nloops)
- double **DecAsUpOS** (double asmu, double massth, double muth, int nf, int nloops)
- double **DecMqUpOS** (double mq, double asmu, double massth, double muth, int nf, int nloops)
- double **DecMqDownOS** (double mq, double asmu, double massth, double muth, int nf, int nloops)

Public Attributes

- double **mq** [4]
- TriplenfMmu **nfMmu** [4]
- AsmMS **AM**

Friends

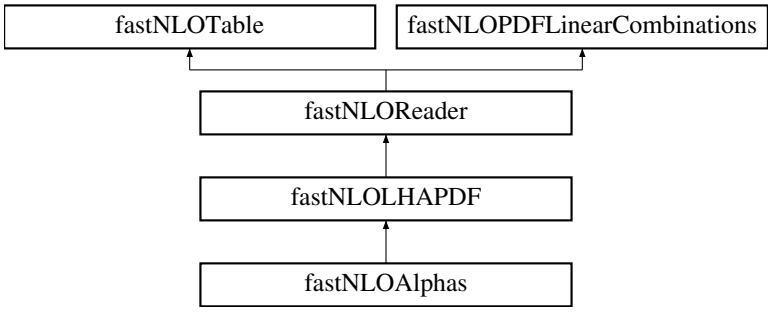
- double **fSetdydx** (CRunDec S, double x, double A, int nl)
- double **fSetdydxa1** (CRunDec S, double x, double A)
- double **fSetdydxM1** (CRunDec S, double A, double M)
- double **fSetdydxa2** (CRunDec S, double x, double A)
- double **fSetdydxM2** (CRunDec S, double A, double M)
- double **fSetdydxa3** (CRunDec S, double x, double A)
- double **fSetdydxM3** (CRunDec S, double A, double M)
- double **fSetdydxa4** (CRunDec S, double x, double A)
- double **fSetdydxM4** (CRunDec S, double A, double M)

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

- fastnlotoolkit/include/fastnlotk/CRunDec.h
- fastnlotoolkit/CRunDec.cc

5.4 fastNLOAlphas Class Reference

Inheritance diagram for fastNLOAlphas:



Public Member Functions

- **fastNLOAlphas** (std::string name)
- **fastNLOAlphas** (std::string name, std::string LHAPDFFile, int PDFSet)
- void **SetMz** (double Mz)
- void **SetNFlavor** (int nflavor)
- void **SetNLoop** (int nloop)
- void **SetAlphasMz** (double AlphasMz, bool ReCalcCrossSection=false)
- double **GetAlphasMz** () const
- void **SetGRVtoPDG2012_2loop** ()

Protected Member Functions

- double **EvolveAlphas** (double Q) const

Protected Attributes

- double **fAlphasMz**

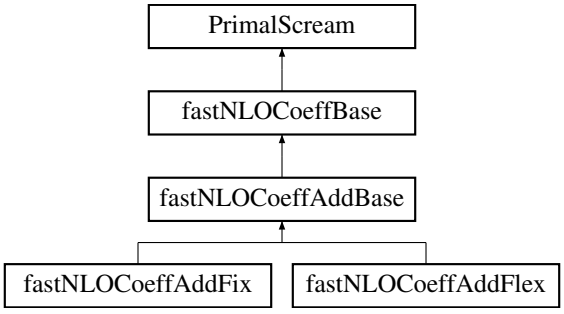
Additional Inherited Members

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

- fastnlotoolkit/include/fastnlotk/fastNLOAlphas.h
- fastnlotoolkit/fastNLOAlphas.cc

5.5 fastNLOCoeffAddBase Class Reference

Inheritance diagram for fastNLOCoeffAddBase:



Public Member Functions

- **fastNLOCoeffAddBase** (int NObsBin)
- **fastNLOCoeffAddBase** (const fastNLOCoeffBase &base)
- virtual fastNLOCoeffAddBase * Clone () const

  returns 'new' copy of this instance.
- void Read (std::istream &table)

  _____ //
- virtual void **Write** (std::ostream &table, int ITabVersionWrite)
- virtual void **Add** (const fastNLOCoeffAddBase &other, fastNLO::EMerge moption=fastNLO::kMerge)
- virtual void **Print** (int iprint) const
- virtual void Clear ()

  Clear all coefficients and event counters.
- virtual void NormalizeCoefficients (double wgt=1)

  Set number of events to unity and normalize coefficients accordingly.
- virtual void **NormalizeCoefficients** (const std::vector< std::vector< double > > &wgtProcBin)
- virtual void **MultiplyCoefficientsByConstant** (double fact)
- virtual void MultiplyBin (unsigned int iObsIdx, double fact)

  Multiply all coefficients of all bins by a constant factor.
- virtual void MultiplyBinProc (unsigned int iObsIdx, unsigned int iProc, double fact)

  Multiply coefficients of one observable bin a factor.
- virtual void EraseBin (unsigned int iObsIdx)

  Multiply coefficients of one observable bin a factor (idx starting from 0)
- virtual void CatBin (const fastNLOCoeffAddBase &other, unsigned int iObsIdx)

  Catenate observable to table.
- int **GetIRef** () const
- void **SetIRef** (int iref=1)
- double **GetNevt** () const
- double **GetNevt** (int NObsBin, int NSubproc) const
- void SetNevt (double nevt)

  Set number of events.
- int **GetNxmax** (int Obsbin) const
- int **GetXIndex** (int Obsbin, int x1bin, int x2bin=0) const
- int **GetNSubproc** () const
- int **GetIScaleDep** () const
- int **GetNPDF** () const
- int **GetPDFPDG** (int iPDF) const
- int **GetNPDFDim** () const
- int **GetIPDFdef1** () const
- int **GetIPDFdef2** () const
- int **GetIPDFdef3** () const
- int **GetNpow** () const
- int **GetNScales** () const
- int **GetNScaleDim** () const
- std::string **GetScaleDescription** (int iScale=0) const
- std::vector< std::vector< std::string > > **GetScaleDescr** () const
- int **GetNxtot1** (int iBin) const
- int **GetNxtot2** (int iBin) const
- double **GetXNode1** (int iObsBin, int iNode) const
- double **GetXNode2** (int iObsBin, int iNode) const
- double **GetX1** (int iObsBin, int iXnode) const
- double GetX2 (int iObsBin, int iXnode) const

return x value of pdf1 for x-node 1

- std::vector< double > GetXNodes1 (int iObsBin) const

  return x value of pdf1 for x-node 1
- std::vector< double > **GetXNodes2** (int iObsBin) const
- bool **IsReference** () const
- bool **IsCompatible** (const fastNLOCoeffAddBase &other) const
- bool **IsCatenable** (const fastNLOCoeffAddBase &other) const
- const std::vector< std::vector< std::pair< int, int > > > & **GetPDFCoeff** () const
- const fastNLO::WgtStat & GetWgtStat () const

  Get weight and event counts.
- fastNLO::WgtStat & AccessWgtStat ()

  Get weight and event counts.
- double GetMergeWeight (fastNLO::EMerge moption, int proc, int bin) const

  Get merge weight for a given bin and subprocess.

Static Public Member Functions

- static bool **CheckCoeffConstants** (const fastNLOCoeffBase *c, bool quiet=false)

Protected Member Functions

- void **ReadCoeffAddBase** (std::istream &table)
- int **GetScaledimfromvar** (int scalevar) const

Protected Attributes

- int **IRef** = 0
- int **IScaleDep** = 0
- double **Nevt** = 0
- int **Npow** = 0
- std::vector< int > **NPDFPDG**
- int **NPDFDim** = 0
- std::vector< int > **NFFPDG**
- int **NFFDim** = 0
- int **NSubproc** = 0
- int **IPDFdef1** = 0
- int **IPDFdef2** = 0
- int **IPDFdef3** = 0
- std::vector< std::vector< std::pair< int, int > > > **fPDFCoeff**
- std::vector< double > Hxlim1

  fPDFCoeff[iSubProc][iPartonPair][pair]
- fastNLO::v2d **XNode1**
- std::vector< double > **Hxlim2**
- fastNLO::v2d **XNode2**
- std::vector< int > **Nztot**
- std::vector< double > **Hzlim**
- fastNLO::v2d **ZNode**
- int **NScales** = 0
- int **NScaleDim** = 0
- std::vector< int > **Iscale**
- std::vector< std::vector< std::string > > **ScaleDescript**
- fastNLO::WgtStat fWgt

  event and weight counts

Friends

- class **fastNLOTable**
- class **fastNLOCreate**

Additional Inherited Members

5.5.1 Member Function Documentation

5.5.1.1 Clear()

```
void fastNLOCoeffAddBase::Clear ( )  [virtual]
```

Clear all coefficients and event counters.

Clear all coefficients and event counts

Reimplemented in fastNLOCoeffAddFix, and fastNLOCoeffAddFlex.

5.5.1.2 Clone()

```
fastNLOCoeffAddBase * fastNLOCoeffAddBase::Clone ( ) const  [virtual]
```

returns 'new' copy of this instance.

Use has to take care to delete this object later

Reimplemented from fastNLOCoeffBase.

Reimplemented in fastNLOCoeffAddFix, and fastNLOCoeffAddFlex.

5.5.1.3 EraseBin()

```
void fastNLOCoeffAddBase::EraseBin (
            unsigned int iObsIdx )  [virtual]
```

Multiply coefficients of one observable bin a factor (idx starting from 0)

Erase observable bin from table

Reimplemented from fastNLOCoeffBase.

Reimplemented in fastNLOCoeffAddFix, and fastNLOCoeffAddFlex.

5.5.1.4 GetMergeWeight()

```
double fastNLOCoeffAddBase::GetMergeWeight (
            fastNLO::EMerge moption,
            int proc,
            int bin ) const
```

Get merge weight for a given bin and subprocess.

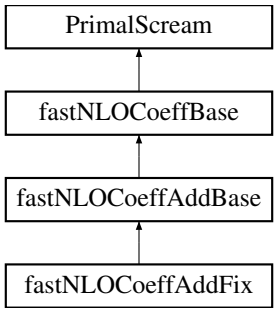
< Get a bin and subprocess dependent weight for merging puprposes.

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

- fastnlotoolkit/include/fastnlotk/fastNLOCoeffAddBase.h
- fastnlotoolkit/fastNLOCoeffAddBase.cc

5.6 fastNLOCoeffAddFix Class Reference

Inheritance diagram for fastNLOCoeffAddFix:



Public Member Functions

- **fastNLOCoeffAddFix** (int NObsBin)
- **fastNLOCoeffAddFix** (const fastNLOCoeffBase &)
- virtual fastNLOCoeffAddFix * Clone () const

  returns 'new' copy of this instance.
- virtual void Read (std::istream &table)

  _____↩
  _____ //
- void **ReadRest** (std::istream &table)
- virtual void Write (std::ostream &table, int ITabVersionWrite)
- virtual void Add (const fastNLOCoeffAddBase &other, fastNLO::EMerge moption=fastNLO::kMerge)
- virtual void **Print** (int iprint) const
- virtual void Clear ()

  Clear all coefficients and event counters.
- virtual void NormalizeCoefficients (double wgt=1)

  a Set number of events to wgt and re-normalize coefficients accordingly
- virtual void NormalizeCoefficients (const std::vector< std::vector< double > > &wgtProcBin)
- virtual void MultiplyCoefficientsByConstant (double fact)

  Multiply all coefficients of all bins by a constant factor.

- virtual void MultiplyBin (unsigned int iObsIdx, double fact)

  Multiply coefficients of one bin a factor.
- virtual void MultiplyBinProc (unsigned int iObsIdx, unsigned int iProc, double fact)

  Multiply coefficients of one bin and subprocess by a factor.
- virtual void EraseBin (unsigned int iObsIdx)

  Multiply coefficients of one observable bin a factor (idx starting from 0)
- virtual void **CatBin** (const fastNLOCoeffAddFix &other, unsigned int iObsIdx)
- int GetTotalScalevars () const
- int GetTotalScalenodes () const
- int **GetNScaleNode** () const
- int **GetNScalevar** () const
- fastNLO::v1d **GetAvailableScaleFactors** () const
- double **GetScaleFactor** (int iVar) const
- double **GetSigmaTilde** (int iObs, int iSvar, int ix, int is, int iN) const
- double **GetScaleNode** (int iObs, int iSvar, int iNode) const
- std::vector< double > **GetScaleNodes** (int iObs, int iSvar) const
- void ResizePdfLC ()
- void ResizePdfSplLC ()
- void ResizeSigmaTilde ()
- bool IsCompatible (const fastNLOCoeffAddFix &other) const

  Check for compatibility of two contributions for merging/adding.
- bool IsCatenable (const fastNLOCoeffAddFix &other) const

  Check for compatibility of two contributions for merging/adding.

Static Public Member Functions

- static bool **CheckCoeffConstants** (const fastNLOCoeffBase *c, bool quiet=false)

Public Attributes

- fastNLO::v2d **AlphasTwoPi_v20**
- fastNLO::v4d **PdfLc**
- fastNLO::v4d **PdfSplLc1**
- fastNLO::v4d **PdfSplLc2**

Protected Member Functions

- void **ReadCoeffAddFix** (std::istream &table)

Protected Attributes

- std::vector< int > **Nscalevar**
- fastNLO::v2d **ScaleFac**
- fastNLO::v4d **ScaleNode**
- fastNLO::v5d **SigmaTilde**

Friends

- class **fastNLOTable**
- class **fastNLOReader**
- class **fastNLOCreate**

5.6.1 Member Function Documentation

5.6.1.1 Add()

```
void fastNLOCoeffAddFix::Add (
            const fastNLOCoeffAddBase & other,
            fastNLO::EMerge moption = fastNLO::kMerge )  [virtual]
```

Add another coefficient table to this table

Reimplemented from fastNLOCoeffAddBase.

5.6.1.2 Clear()

```
void fastNLOCoeffAddFix::Clear ( )  [virtual]
```

Clear all coefficients and event counters.

Set all elelments of SigmaTilde to zero.

Reimplemented from fastNLOCoeffAddBase.

5.6.1.3 Clone()

```
fastNLOCoeffAddFix * fastNLOCoeffAddFix::Clone ( ) const  [virtual]
```

returns 'new' copy of this instance.

Use has to take care to delete this object later

Reimplemented from fastNLOCoeffAddBase.

5.6.1.4 EraseBin()

```
void fastNLOCoeffAddFix::EraseBin (
            unsigned int iObsIdx )  [virtual]
```

Multiply coefficients of one observable bin a factor (idx starting from 0)

Erase observable bin from table Erase observable bin

Reimplemented from fastNLOCoeffAddBase.

5.6.1.5 GetTotalScalenodes()

```
int fastNLOCoeffAddFix::GetTotalScalenodes ( ) const
```

Get number of scale nodes

5.6.1.6 GetTotalScalevars()

```
int fastNLOCoeffAddFix::GetTotalScalevars ( ) const
```

Get nuber of scale-variations

5.6.1.7 IsCatenable()

```
bool fastNLOCoeffAddFix::IsCatenable (
            const fastNLOCoeffAddFix & other ) const
```

Check for compatibility of two contributions for merging/adding.

Check for compatibility of catenating observable bins

5.6.1.8 IsCompatible()

```
bool fastNLOCoeffAddFix::IsCompatible (
            const fastNLOCoeffAddFix & other ) const
```

Check for compatibility of two contributions for merging/adding.

Check for compatibility for merging/adding of two contributions

5.6.1.9 MultiplyBin()

```
void fastNLOCoeffAddFix::MultiplyBin (
            unsigned int iObsIdx,
            double fact )  [virtual]
```

Multiply coefficients of one bin a factor.

Multiply observable bin

Reimplemented from fastNLOCoeffAddBase.

5.6.1.10 MultiplyBinProc()

```
void fastNLOCoeffAddFix::MultiplyBinProc (
            unsigned int iObsIdx,
            unsigned int iProc,
            double fact )  [virtual]
```

Multiply coefficients of one bin and subprocess by a factor.

Multiply observable bin for a single subprocess

Reimplemented from fastNLOCoeffAddBase.

5.6.1.11 NormalizeCoefficients() [1/2]

```
void fastNLOCoeffAddFix::NormalizeCoefficients (
            double wgt = 1 )  [virtual]
```

a Set number of events to wgt and re-normalize coefficients accordingly

< Set number of events to wgt (default=1) and normalize coefficients accordingly. This means, that the information about the number of events is essentially lost

Reimplemented from fastNLOCoeffAddBase.

5.6.1.12 NormalizeCoefficients() [2/2]

```
void fastNLOCoeffAddFix::NormalizeCoefficients (
            const std::vector< std::vector< double > > & wgtProcBin )  [virtual]
```

< Set number of events to wgtProcBin for each subprocess and bin and normalize coefficients accordingly.

Reimplemented from fastNLOCoeffAddBase.

5.6.1.13 ResizePdfLC()

```
void fastNLOCoeffAddFix::ResizePdfLC ( )
```

resize PdfLC

5.6.1.14 ResizePdfSplLC()

```
void fastNLOCoeffAddFix::ResizePdfSplLC ( )
```

resize PdfSplLC

5.6.1.15 ResizeSigmaTilde()

```
void fastNLOCoeffAddFix::ResizeSigmaTilde ( )
```

resize SigmaTilde and ensure that all entries are empty

5.6.1.16 Write()

```
void fastNLOCoeffAddFix::Write (
            std::ostream & table,
            int ITabVersionWrite )  [virtual]
```

Write coefficient table to disk (ostream)

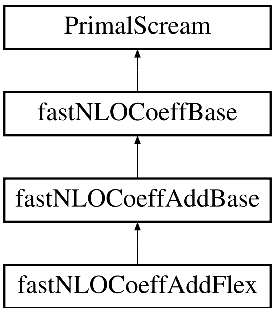
Reimplemented from fastNLOCoeffAddBase.

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

- fastnlotoolkit/include/fastnlotk/fastNLOCoeffAddFix.h
- fastnlotoolkit/fastNLOCoeffAddFix.cc

5.7 fastNLOCoeffAddFlex Class Reference

Inheritance diagram for fastNLOCoeffAddFlex:



Public Member Functions

- **fastNLOCoeffAddFlex** (int NObsBin, int iLOord)
- **fastNLOCoeffAddFlex** (const fastNLOCoeffBase &base, int iLOord)
- virtual fastNLOCoeffAddFlex ∗ Clone () const

  returns 'new' copy of this instance.
- virtual void Read (std::istream &table)

  ____↩
  ______________________ //
- void **ReadRest** (std::istream &table)
- virtual void **Write** (std::ostream &table, int ITabVersionWrite)
- virtual void **Print** (int iprint) const
- virtual void **Add** (const fastNLOCoeffAddBase &other, fastNLO::EMerge moption=fastNLO::kMerge)
- virtual void Clear ()

Clear all coefficients and event counters.

- virtual void NormalizeCoefficients (double wgt=1)

  Set number of events to given wgt and re-normalize coefficients accordingly.
- virtual void NormalizeCoefficients (const std::vector< std::vector< double > > &wgtProcBin)
- virtual void MultiplyCoefficientsByConstant (double fact)

  Multiply all coefficients of all bins by a constant factor.
- virtual void MultiplyBin (unsigned int iObsIdx, double fact)

  Multiply coefficients of one bin a factor (iObsIdx starting with index 0)
- virtual void MultiplyBinProc (unsigned int iObsIdx, unsigned int iProc, double fact)

  Multiply coefficients of one bin and subprocess a factor.
- virtual void EraseBin (unsigned int iObsIdx)

  Erase observable bin from table.
- virtual void CatBin (const fastNLOCoeffAddFlex &other, unsigned int iObsIdx)

  Catenate observable to table.
- unsigned int **GetNScaleNode1** (int iObsBin) const
- unsigned int **GetNScaleNode2** (int iObsBin) const
- double **GetScaleNode1** (int iObsBin, int iNode) const
- double **GetScaleNode2** (int iObsBin, int iNode) const
- std::vector< double > **GetScaleNodes1** (int iObsBin) const
- std::vector< double > **GetScaleNodes2** (int iObsBin) const
- bool IsCompatible (const fastNLOCoeffAddFlex &other) const

  check for compatibilty for adding/merging of two tables
- bool IsCatenable (const fastNLOCoeffAddFlex &other) const

  Check for compatibility of two contributions for merging/adding.
- std::vector< fastNLO::v5d * > **AccessSigmaTildes** ()
- std::vector< const fastNLO::v5d * > GetSigmaTildes () const

  Get access to sigma tilde.

Static Public Member Functions

- static bool **CheckCoeffConstants** (const fastNLOCoeffBase *c, bool quiet=false)

Public Attributes

- fastNLO::v3d **AlphasTwoPi**
- fastNLO::v5d **PdfLcMuVar**
- fastNLO::v5d **PdfXfx**

Protected Member Functions

- void ReadCoeffAddFlex (std::istream &table)

  Get access to sigma tilde.

Protected Attributes

- int **fILOord**
- int **fSTildeDISFormat** = 1
- fastNLO::v5d **SigmaTildeMuIndep**
- fastNLO::v5d **SigmaTildeMuFDep**
- fastNLO::v5d **SigmaTildeMuRDep**
- fastNLO::v5d **SigmaTildeMuRRDep**
- fastNLO::v5d **SigmaTildeMuFFDep**
- fastNLO::v5d **SigmaTildeMuRFDep**
- fastNLO::v2d **SigmaRefMixed**
- fastNLO::v2d **SigmaRef_s1**
- fastNLO::v2d **SigmaRef_s2**
- fastNLO::v2d **ScaleNode1**
- fastNLO::v2d **ScaleNode2**

Friends

- class **fastNLOTable**
- class **fastNLOReader**
- class **fastNLOCreate**

5.7.1 Member Function Documentation

5.7.1.1 Clear()

```
void fastNLOCoeffAddFlex::Clear ( )  [virtual]
```

Clear all coefficients and event counters.

Set all elements of sigma tilde to zero

Reimplemented from fastNLOCoeffAddBase.

5.7.1.2 Clone()

```
fastNLOCoeffAddFlex * fastNLOCoeffAddFlex::Clone ( ) const  [virtual]
```

returns 'new' copy of this instance.

User has to take care to delete this object later

Reimplemented from fastNLOCoeffAddBase.

5.7.1.3 IsCatenable()

```
bool fastNLOCoeffAddFlex::IsCatenable (
            const fastNLOCoeffAddFlex & other ) const
```

Check for compatibility of two contributions for merging/adding.

Check for compatibility of catenating observable bins

5.7.1.4 IsCompatible()

```
bool fastNLOCoeffAddFlex::IsCompatible (
            const fastNLOCoeffAddFlex & other ) const
```

check for compatibilty for adding/merging of two tables

Check for compatibility for merging/adding of two contributions

5.7.1.5 MultiplyBin()

```
void fastNLOCoeffAddFlex::MultiplyBin (
            unsigned int iObsIdx,
            double fact )  [virtual]
```

Multiply coefficients of one bin a factor (iObsIdx starting with index 0)

Multiply observable bin

Reimplemented from fastNLOCoeffAddBase.

5.7.1.6 MultiplyBinProc()

```
void fastNLOCoeffAddFlex::MultiplyBinProc (
            unsigned int iObsIdx,
            unsigned int iProc,
            double fact )  [virtual]
```

Multiply coefficients of one bin and subprocess a factor.

Multiply observable bin

Reimplemented from fastNLOCoeffAddBase.

5.7.1.7 NormalizeCoefficients() [1/2]

```
void fastNLOCoeffAddFlex::NormalizeCoefficients (
            double wgt = 1 )  [virtual]
```

Set number of events to given wgt and re-normalize coefficients accordingly.

< Set number of events to wgt (default=1) and normalize coefficients accordingly.

Reimplemented from fastNLOCoeffAddBase.

5.7.1.8 NormalizeCoefficients() [2/2]

```
void fastNLOCoeffAddFlex::NormalizeCoefficients (
            const std::vector< std::vector< double > > & wgtProcBin )  [virtual]
```

< Change cross sections!!! Warning! This function is only sensible if called by 'MergeTable'!

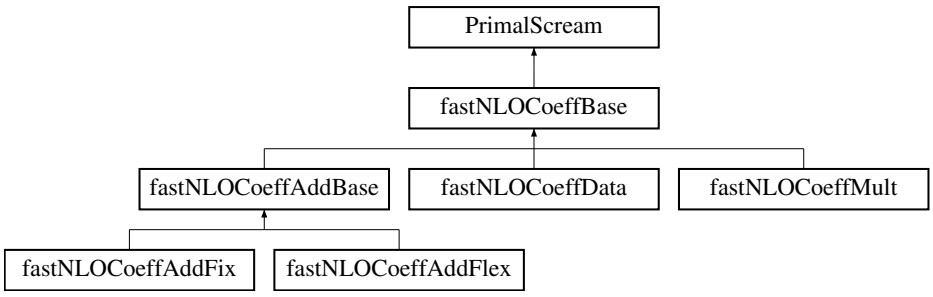
Reimplemented from fastNLOCoeffAddBase.

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

- fastnlotoolkit/include/fastnlotk/fastNLOCoeffAddFlex.h
- fastnlotoolkit/fastNLOCoeffAddFlex.cc

5.8 fastNLOCoeffBase Class Reference

Inheritance diagram for fastNLOCoeffBase:



Public Member Functions

- **fastNLOCoeffBase** (int NObsBin)
- virtual ~fastNLOCoeffBase ()

  Use this constructor.
- virtual fastNLOCoeffBase * Clone () const

  destructor
- virtual void **Read** (std::istream &table)
- virtual void **Write** (std::ostream &table, int ITabVersionWrite)
- virtual void **Print** (int iprint) const
- virtual void **EraseBin** (unsigned int iObsIdx)
- virtual void **MultiplyBin** (unsigned int iObsIdx, double fact)
- virtual void **CatBin** (const fastNLOCoeffBase &other, unsigned int iObsIdx)
- bool **IsCatenable** (const fastNLOCoeffBase &other) const
- void **SetCoeffAddDefaults** ()
- int **GetIDataFlag** () const
- void **SetIDataFlag** (int n)
- int **GetIAddMultFlag** () const
- void **SetIAddMultFlag** (int n)
- int **GetIContrFlag1** () const
- void **SetIContrFlag1** (int n)
- int **GetIContrFlag2** () const
- void **SetIContrFlag2** (int n)
- int **GetNScaleDep** () const
- void **SetNScaleDep** (int n)
- int **GetIXsectUnits** () const
- void **SetIXsectUnits** (int n)
- int **GetNObsBin** () const
- void **SetNObsBin** (unsigned int nObs)
- bool **GetIsFlexibleScale** () const
- std::vector< std::string > **GetContributionDescription** () const
- void **SetContributionDescription** (std::vector< std::string > descr)
- std::vector< std::string > GetCodeDescription () const

  Set contribution description.
- bool **IsLO** () const
- bool **IsNLO** () const
- bool **IsNNLO** () const
- bool **IsCompatible** (const fastNLOCoeffBase &other) const

Protected Member Functions

- void **ReadBase** (std::istream &table)
- void **EndReadCoeff** (std::istream &table)

Protected Attributes

- int **fNObsBins**
- int **IXsectUnits**
- int **IDataFlag**
- int **IAddMultFlag**
- int **IContrFlag1**
- int **IContrFlag2**
- int **NScaleDep**
- int **fVersionRead** = 23000
- std::vector< std::string > **CtrbDescript**
- std::vector< std::string > **CodeDescript**

Friends

- class **fastNLOTable**

Additional Inherited Members

5.8.1 Member Function Documentation

5.8.1.1 Clone()

```
fastNLOCoeffBase * fastNLOCoeffBase::Clone ( ) const  [virtual]
```

destructor

returns 'new' copy of this instance. User has to take care to delete this object later

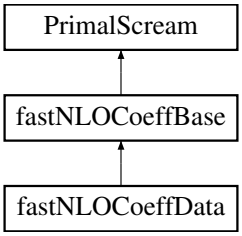
Reimplemented in fastNLOCoeffAddBase, fastNLOCoeffAddFix, fastNLOCoeffAddFlex, fastNLOCoeffMult, and fastNLOCoeffData.

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

- fastnlotoolkit/include/fastnlotk/fastNLOCoeffBase.h
- fastnlotoolkit/fastNLOCoeffBase.cc

5.9 fastNLOCoeffData Class Reference

Inheritance diagram for fastNLOCoeffData:



Public Member Functions

- **fastNLOCoeffData** (int NObsBin)
- **fastNLOCoeffData** (const fastNLOCoeffBase &)
- virtual fastNLOCoeffData ∗ Clone () const

  returns 'new' copy of this instance.
- virtual void Read (std::istream &table)

  _____↩
  _____ //
- virtual void **Write** (std::ostream &table, int ITabVersionWrite)
- virtual void **Print** (int iprint) const
- virtual void **EraseBin** (unsigned int iObsIdx)
- virtual void **MultiplyBin** (unsigned int iObsIdx, double fact)
- virtual void **CatBin** (const fastNLOCoeffData &other, unsigned int iObsIdx)
- bool IsCatenable (const fastNLOCoeffData &other) const
- int **GetNuncorrel** () const
- void **SetNuncorrel** (int n)
- int **GetNcorrel** () const
- void **SetNcorrel** (int n)
- int **GetNErrMatrix** () const
- void **SetNErrMatrix** (int n)

Static Public Member Functions

- static bool **CheckCoeffConstants** (const fastNLOCoeffBase ∗c, bool quiet=false)

Protected Member Functions

- void **ReadCoeffData** (std::istream &table)
- void **ReadRest** (std::istream &table)

Protected Attributes

- int **Nuncorrel**
- std::vector< std::string > **UncDescr**
- int **Ncorrel**
- std::vector< std::string > **CorDescr**
- std::vector< double > **Xcenter**
- std::vector< double > **Value**
- fastNLO::v2d **UncorLo**
- fastNLO::v2d **UncorHi**
- fastNLO::v2d **CorrLo**
- fastNLO::v2d **CorrHi**
- int **NErrMatrix**
- fastNLO::v2d **matrixelement**

Friends

- class **fastNLOTable**

Additional Inherited Members

5.9.1 Member Function Documentation

5.9.1.1 Clone()

```
fastNLOCoeffData * fastNLOCoeffData::Clone ( ) const  [virtual]
```

returns 'new' copy of this instance.

Use has to take care to delete this object later

Reimplemented from fastNLOCoeffBase.

5.9.1.2 IsCatenable()

```
bool fastNLOCoeffData::IsCatenable (
            const fastNLOCoeffData & other ) const
```

Check for compatibility of catenating observable bins

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

- fastnlotoolkit/include/fastnlotk/fastNLOCoeffData.h
- fastnlotoolkit/fastNLOCoeffData.cc

5.10 fastNLOCoefficients Class Reference

Public Member Functions

- **fastNLOCoefficients** (int NObsBin, int iLOord)
- int Read (std::istream *table)
- int Write (std::ostream *table, int option=0)
- int **Copy** (fastNLOCoefficients *other)
- void **StripWhitespace** (std::string &s) const
- void **ResizeTable** (std::vector< double > *v, int dim0)
- void **ResizeTable** (std::vector< std::vector< double > > *v, int dim0, int dim1)
- void **ResizeTable** (std::vector< std::vector< double > > *v, int dim0, int *dim1GetNxmaxFromDimI)
- void **ResizeTable** (std::vector< std::vector< std::vector< double > > > *v, int dim0, int *dim1GetNxmax↩ FromDimI, int dim2)
- void **ResizeTable** (std::vector< std::vector< std::vector< double > > > *v, int dim0, int dim1, int dim2)
- void **ResizeTable** (std::vector< std::vector< std::vector< std::vector< double > > > > *v, int dim0, int dim1, int dim2, int dim3)
- void **ResizeTable** (std::vector< std::vector< std::vector< std::vector< double > > > > *v, int dim0, int dim1, int *dim2GetNxmaxFromDimI, int dim3)

- void **ResizeTable** (std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > *v, int dim0, int dim1, int dim2, int dim3, int dim4)
- void **ResizeTable** (std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > *v, int dim0, int dim1, int dim2, int *dim3GetNxmaxFromDimI, int dim4)
- void **ResizeTable** (std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > *v, int dim0, int *dim1GetNxmaxFromDimI, int dim2, int dim3, int dim4)
- void **ResizeTable** (std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > > *v, int dim0, int dim1, int dim2, int dim3, int dim4, int dim5)
- void **ResizeTable** (std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > > > *v, int dim0, int dim1, int dim2, int dim3, int dim4, int dim5, int dim6)
- void **ResizeTable** (std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > > > *v, int dim0, int dim1, int dim2, int dim3, int dim4, int *dim5GetNxmaxFromDimI, int dim6)
- template<typename T >
  int **ReadTable** (std::vector< T > *v, std::istream *table)
- int **ReadTable** (std::vector< double > *v, std::istream *table)
- template<typename T >
  int **ReadFlexibleVector** (std::vector< T > *v, std::istream *table, bool nProcLast=false)
- int **ReadFlexibleVector** (std::vector< double > *v, std::istream *table, bool nProcLast=false)
- int **WriteTable** (std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > > > *v, std::ostream *table, bool DivByNevt=false, int Nevt=1)
- int **WriteTable** (std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > > *v, std::ostream *table, bool DivByNevt=false, int Nevt=1)
- int **WriteTable** (std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > *v, std::ostream *table, bool DivByNevt=false, int Nevt=1)
- int **WriteTable** (std::vector< std::vector< std::vector< std::vector< double > > > > *v, std::ostream *table, bool DivByNevt=false, int Nevt=1)
- int **WriteTable** (std::vector< std::vector< std::vector< double > > > *v, std::ostream *table, bool DivByNevt=false, int Nevt=1)
- int **WriteTable** (std::vector< std::vector< double > > *v, std::ostream *table, bool DivByNevt=false, int Nevt=1)
- int **WriteTable** (std::vector< double > *v, std::ostream *table, bool DivByNevt=false, int Nevt=1)
- int **WriteFlexibleTable** (std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > > > *v, std::ostream *table, bool DivByNevt=false, int Nevt=1, bool nProcLast=false)
- int **WriteFlexibleTable** (std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > > *v, std::ostream *table, bool DivByNevt=false, int Nevt=1, bool nProcLast=false)
- int **WriteFlexibleTable** (std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > *v, std::ostream *table, bool DivByNevt=false, int Nevt=1, bool nProcLast=false)
- int **WriteFlexibleTable** (std::vector< std::vector< std::vector< std::vector< double > > > > *v, std::ostream *table, bool DivByNevt=false, int Nevt=1, bool nProcLast=false)
- int **WriteFlexibleTable** (std::vector< std::vector< std::vector< double > > > *v, std::ostream *table, bool DivByNevt=false, int Nevt=1, bool nProcLast=false)
- int **WriteFlexibleTable** (std::vector< std::vector< double > > *v, std::ostream *table, bool DivByNevt=false, int Nevt=1, bool nProcLast=false)
- int **WriteFlexibleTable** (std::vector< double > *v, std::ostream *table, bool DivByNevt=false, int Nevt=1, bool nProcLast=false)
- void **AddTableToAnotherTable** (std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > > > *vSum, std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > > > *vAdd, double w1=0, double w2=0)
- void **AddTableToAnotherTable** (std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > > *vSum, std::vector< std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > > *vAdd, double w1=1, double w2=1)
- void **AddTableToAnotherTable** (std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > *vSum, std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > *vAdd, double w1=1, double w2=1)

- void **AddTableToAnotherTable** (std::vector< std::vector< std::vector< std::vector< double > > > > *vSum, std::vector< std::vector< std::vector< std::vector< double > > > > *vAdd, double w1=1, double w2=1)
- void **AddTableToAnotherTable** (std::vector< std::vector< std::vector< double > > > *vSum, std::vector< std::vector< std::vector< double > > > *vAdd, double w1=1, double w2=1)
- void **AddTableToAnotherTable** (std::vector< std::vector< double > > *vSum, std::vector< std::vector< double > > *vAdd, double w1=1, double w2=1)
- void **AddTableToAnotherTable** (std::vector< double > *vSum, std::vector< double > *vAdd, double w1=1, double w2=1)
- void **Print** () const
- int **GetIRef** () const
- int **GetIDataFlag** () const
- int **GetIAddMultFlag** () const
- int **GetIContrFlag1** () const
- int **GetIContrFlag2** () const
- int **GetNScaleDep** () const
- int **GetNpow** () const
- long long int **GetNevt** () const
- int **GetNxmax** (int Obsbin) const
- int **GetXIndex** (int Obsbin, int x1bin, int x2bin=0) const
- void **SetNlojetDefaults** ()
- void **SetIXsectUnits** (int n)
- void **SetIDataFlag** (int n)
- void **SetIAddMultFlag** (int n)
- void **SetIContrFlag1** (int n)
- void **SetIContrFlag2** (int n)
- void **SetNScaleDep** (int n)
- void **SetNlojetDescr** ()
- void **Add** (fastNLOCoefficients *other)
- bool **IsLO** () const
- bool **IsNLO** () const
- bool **IsReference** () const
- int **GetTotalScalevars** () const
- int **GetTotalScalenodes** () const

Static Public Attributes

- static const int **DividebyNevt** = 1

Protected Attributes

- int **fNObsBins**
- int **fILOord**
- int **IXsectUnits**
- int **IDataFlag**
- int **IAddMultFlag**
- int **IContrFlag1**
- int **IContrFlag2**
- int **NScaleDep**
- std::vector< std::string > **CtrbDescript**
- std::vector< std::string > **CodeDescript**
- int **Nuncorrel**
- std::vector< std::string > **UncDescr**

- int **Ncorrel**
- std::vector< std::string > **CorDescr**
- std::vector< double > **Xcenter**
- std::vector< double > **Value**
- std::vector< std::vector< double > > **UncorLo**
- std::vector< std::vector< double > > **UncorHi**
- std::vector< std::vector< double > > **CorrLo**
- std::vector< std::vector< double > > **CorrHi**
- int **NErrMatrix**
- std::vector< std::vector< double > > **matrixelement**
- std::vector< double > **fact**
- int **IRef**
- int **IScaleDep**
- unsigned long long int **Nevt**
- int **Npow**
- int **NPDF**
- std::vector< int > **NPDFPDG**
- int **NPDFDim**
- int **NFragFunc**
- std::vector< int > **NFFPDG**
- int **NFFDim**
- int **NSubproc**
- int **IPDFdef1**
- int **IPDFdef2**
- int **IPDFdef3**
- std::vector< double > **Hxlim1**
- std::vector< std::vector< double > > **XNode1**
- std::vector< double > **Hxlim2**
- std::vector< std::vector< double > > **XNode2**
- std::vector< int > **Nztot**
- std::vector< double > **Hzlim**
- std::vector< std::vector< double > > **ZNode**
- int **NScales**
- int **NScaleDim**
- std::vector< int > **Iscale**
- std::vector< std::vector< std::string > > **ScaleDescript**
- std::vector< int > **Nscalevar**
- std::vector< int > **Nscalenode**
- std::vector< std::vector< double > > **ScaleFac**
- std::vector< std::vector< std::vector< std::vector< double > > > > **ScaleNode**
- std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > **SigmaTilde**
- std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > **SigmaTildeMuIndep**
- std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > **SigmaTildeMuFDep**
- std::vector< std::vector< std::vector< std::vector< std::vector< double > > > > > **SigmaTildeMuRDep**
- std::vector< std::vector< double > > **SigmaRefMixed**
- std::vector< std::vector< double > > **SigmaRef_s1**
- std::vector< std::vector< double > > **SigmaRef_s2**
- int **NscalenodeScale1**
- int **NscalenodeScale2**
- std::vector< std::vector< double > > **ScaleNode1**
- std::vector< std::vector< double > > **ScaleNode2**

Friends

- class **fastNLOTable**
- class **fastNLOCreate**

5.10.1 Member Function Documentation

5.10.1.1 Read()

```
int fastNLOCoefficients::Read (
            std::istream * table )
```

v2.1 store NScaleDep here. v2.1 ∗table >> NScaleDep;

5.10.1.2 Write()

```
int fastNLOCoefficients::Write (
            std::ostream * table,
            int option = 0 )
```

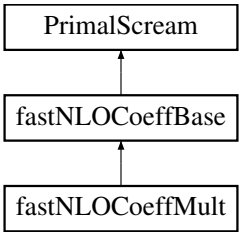
v2.1 store NScaleDep here ∗table << NScaleDep << sep;

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

- fastnlotoolkit/include/fastnlotk/fastNLOCoefficients.h
- fastnlotoolkit/fastNLOCoefficients.cc

5.11 fastNLOCoeffMult Class Reference

Inheritance diagram for fastNLOCoeffMult:



Public Member Functions

- **fastNLOCoeffMult** (int NObsBin)
- **fastNLOCoeffMult** (const fastNLOCoeffBase &)
- virtual fastNLOCoeffMult ∗ Clone () const

  returns 'new' copy of this instance.
- virtual void Read (std::istream &table)

  ____↩
  ____ //
- virtual void **Write** (std::ostream &table, int ITabVersionWrite)
- virtual void **Print** (int iprint) const
- double **GetMultFactor** (int iObs) const
- std::vector< double > **GetMultFactor** () const
- std::vector< std::string > **GetUncDescription** () const
- std::vector< std::string > **GetCorDescription** () const
- fastNLO::v2d **GetUncorLo** () const
- fastNLO::v2d **GetUncorHi** () const
- fastNLO::v2d **GetCorrLo** () const
- fastNLO::v2d **GetCorrHi** () const
- virtual void **EraseBin** (unsigned int iObsIdx)
- virtual void **MultiplyBin** (unsigned int iObsIdx, double fact)
- bool IsCatenable (const fastNLOCoeffMult &other) const
- virtual void **CatBin** (const fastNLOCoeffMult &other, unsigned int iObsIdx)
- int **GetNuncorrel** () const
- void **SetNuncorrel** (int n)
- int **GetNcorrel** () const
- void **SetNcorrel** (int n)

Static Public Member Functions

- static bool **CheckCoeffConstants** (const fastNLOCoeffBase ∗c, bool quiet=false)

Protected Member Functions

- void **ReadCoeffMult** (std::istream &table)
- void **ReadRest** (std::istream &table)

Protected Attributes

- int **Nuncorrel**
- std::vector< std::string > **UncDescr**
- int **Ncorrel**
- std::vector< std::string > **CorDescr**
- fastNLO::v2d **UncorLo**
- fastNLO::v2d **UncorHi**
- fastNLO::v2d **CorrLo**
- fastNLO::v2d **CorrHi**
- fastNLO::v1d **fact**

Friends

- class **fastNLOTable**
- class **fastNLOCreate**

Additional Inherited Members

5.11.1 Member Function Documentation

5.11.1.1 Clone()

```
fastNLOCoeffMult * fastNLOCoeffMult::Clone ( ) const  [virtual]
```

returns 'new' copy of this instance.

Use has to take care to delete this object later

Reimplemented from fastNLOCoeffBase.

5.11.1.2 IsCatenable()

```
bool fastNLOCoeffMult::IsCatenable (
            const fastNLOCoeffMult & other ) const
```

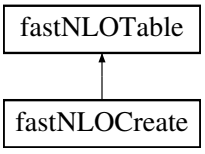
Check for compatibility of catenating observable bins

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

- fastnlotoolkit/include/fastnlotk/fastNLOCoeffMult.h
- fastnlotoolkit/fastNLOCoeffMult.cc

5.12 fastNLOCreate Class Reference

Inheritance diagram for fastNLOCreate:



Classes

- struct fnloStats

Public Member Functions

- fastNLOCreate (const std::string &steerfile, std::string warmupfile="", bool shouldReadSteeringFile=true)
- **fastNLOCreate** (const std::string &steerfile, const fastNLO::GeneratorConstants &GenConsts, const fastNLO::ProcessConstants &ProcConsts)
- **fastNLOCreate** (const std::string &warmupfile, const fastNLO::GeneratorConstants &GenConsts, const fastNLO::ProcessConstants &ProcConsts, const fastNLO::ScenarioConstants &ScenConsts)
- fastNLOCreate (const fastNLO::GeneratorConstants &GenConsts, const fastNLO::ProcessConstants &ProcConsts, const fastNLO::ScenarioConstants &ScenConsts, const fastNLO::WarmupConstants &WarmConsts)
- void SetOrderOfAlphasOfCalculation (unsigned int ord)

  set absolute order of alpha_s
- void SetScenario (const fnloScenario scen)

  set the member fScenario, which will be used when calling Fill()
- void SetEvent (const fnloEvent ev)

  set the member fEvent, which will be used when calling Fill()
- void **SetNumberOfEvents** (double n)
- void SetLoOrder (int LOOrd)

  set number of events. This is only mandatory, before calling WriteTable().
- fastNLOReader * SetIsReferenceTable (fastNLOReader *fnloread=NULL)

  set this table/contribution to become a reference contribution
- void SetGenConstsDefaults ()

  set defaults for generator constants
- void SetProcConstsDefaults ()

  set defaults for process constants
- bool CheckProcConsts ()

  check process constants to be different from defaults
- void Fill (int scalevar=0)

  fill event quantities in fastNLO table. Call it for every subprocess.
- void FillOneSubprocess (const fnloEvent &event, const fnloScenario &scen, int scalevar=0)

  same function as 'Fill()', but uses content of member fScenario and fEvent
- void FillAllSubprocesses (const std::vector< fnloEvent > &events, const fnloScenario &scen, int scalevar=0)

  Fill a selection (std::vector) of events/processes/channels, which all have the identic scenario.
- void FillAllSubprocesses (const std::vector< std::vector< fnloEvent > > &events, const fnloScenario &scen)

  Fill a list of subprocesses for various scale-variations into a fixed-scale table.
- int GetNSubprocesses () const

  The number of subprocesses (channels)
- const std::vector< double > & GetScaleVariations () const

  Get list of scale variations.
- void WriteTable (std::string filename)

  Write fastNLO table to file filename.
- void WriteTable ()

  Write fastNLO table to disk.
- void **WriteTable** (double nEvents)
- void WriteWarmupTable ()

  Write fastNLO table to disk and set number of events.
- void MultiplyCoefficientsByBinSize ()

Multiply all coefficients by bin size.

- void DivideCoefficientsByBinSize ()

  Divide all coefficients by bin size.
- void MultiplyCoefficientsByConstant (double c)

  Multiply all coefficients with a constant factor c.
- void NormalizeCoefficients (double wgt=1)

  Set number of events to wgt and adjust coefficients accordingly.
- void NormalizeCoefficients (const std::vector< std::vector< double > > &wgtProcBin)

  Set number of events to wgt (with wgt differently for each subproc. and bin) resulting Nevt will become 1.
- void PrintWarmupValues ()

  Print the warmup values to the screen.
- std::string GetWarmupTableFilename ()

  Get the filename, which is used for storage of the warmup-table.
- void SetWarmupTableFilename (std::string)

  Set the filename, which is used for storage of the warmup-table (otherwise a default is used)
- bool **GetIsWarmup** () const
- bool GetIsFlexibleScale () const

  Get flag for warmup table.
- void SetWarmupXSafetyMargin (int margin=4)

  Set margin for x-value: First digit in 'e' notation (e.g. margin=4: x=6.6e-3 -> 6.2e-3)
- void SetWarmupNDigitMu1 (int rnd=1)

  Round warmup values for scale 1.
- void SetWarmupNDigitMu2 (int rnd=2)

  Round warmup values for scale 2.
- fastNLOCoeffAddBase * GetTheCoeffTable () const

  Getter for the one (and only) coefficient table.
- bool TestParameterInSteering (const std::string &label) const

  Test on existence of user-defined parameter name in steering card.
- bool GetParameterFromSteering (const std::string &label, bool &val) const

  Get user-defined parameter from steering file.
- bool GetParameterFromSteering (const std::string &label, int &val) const

  Get user-defined parameter from steering file.
- bool GetParameterFromSteering (const std::string &label, double &val) const

  Get user-defined parameter from steering file.
- bool GetParameterFromSteering (const std::string &label, std::string &val) const

  Get user-defined parameter from steering file.
- bool GetParameterFromSteering (const std::string &label, std::vector< int > &val) const

  Get user-defined parameter from steering file.
- bool GetParameterFromSteering (const std::string &label, std::vector< double > &val) const

  Get user-defined parameter from steering file.
- bool GetParameterFromSteering (const std::string &label, std::vector< std::string > &val) const

  Get user-defined parameter from steering file.
- bool GetParameterFromSteering (const std::string &label, std::vector< std::vector< int > > &val) const

  Get user-defined parameter from steering file.
- bool GetParameterFromSteering (const std::string &label, std::vector< std::vector< double > > &val) const

  Get user-defined parameter from steering file.
- void AdjustWarmupValues ()

  Round warmup values to more likely values.
- void **PrintAllSteeringValues** () const
- void Clear ()

  Print all steering values obtained from steering files (of all fastNLOCreate instances);.

- void PrintStats () const

  Clear coefficient table.
- void SetGlobalVerbosity (std::string sverb)

  Set GlobalVerbosity using std::string variable.
- void SetCacheSize (int MaxCache)

  Set maximum number of events in cache. Set to 0 to deactivate caching.
- int GetCacheSize () const

  Get max cache size.

Public Attributes

- fnloEvent fEvent

  Structure, which holds all relevant variables related to event observables.
- fnloScenario fScenario

  Structure, which holds perturbative (wilson) coefficients/weights and x-values.

Protected Member Functions

- fastNLOCreate ()

  don't use the default constructor. fastNLOCreate is only reasonable with input steering.
- void Instantiate ()
- int CreateCoeffTable ()

  Create the one (and only) coefficient table.
- void **ApplyPDFWeight** (std::vector< std::pair< int, double > > &nodes, const double x, const std::vector< double > *grid) const
- double **CalcPDFReweight** (double x) const
- void FillContribution (int scalevar=0)

  fill contribution into table
- void FillContributionFlexHHC (fastNLOCoeffAddFlex *c, int ObsBin)

  fill flexible scale contribution in pp/ppbar
- void FillContributionFlexDIS (fastNLOCoeffAddFlex *c, int ObsBin)

  fill flexible scale contribution in DIS
- void FillContributionFixHHC (fastNLOCoeffAddFix *c, int ObsBin, int scalevar)

  fill fixed scale table in pp/ppbar
- void FillContributionFixDIS (fastNLOCoeffAddFix *c, int ObsBin, int scalevar)

  fill fixed scale contribution in DIS
- void FillRefContribution (int scalevar=0)

  fill contribution if this is a reference table
- void ReadSteering (std::string steerfile, std::string steeringNameSpace="", bool shouldReadSteering↩File=true)

  read steering file
- void ReadGenAndProcConstsFromSteering ()
- void ReadScenarioConstsFromSteering ()

  Read steering values into fScenConsts.
- void **ReadBinning** ()
- void ReadBinningFromScenarioConsts ()
- void ReadBinSize ()
- void **SetBinning1D** (std::vector< double > bgrid, std::string label, unsigned int idiff)
- void **SetBinning1D** (std::vector< double > bgrid, std::string label, unsigned int idiff, double norm)

- void **SetBinning1D** (std::vector< double > bgrid, std::string label, unsigned int idiff, std::vector< double > vnorm)
- void **SetBinning1D** (std::vector< double > blow, std::vector< double > bupp, std::string label, unsigned int idiff)
- void **SetBinning1D** (std::vector< double > blow, std::vector< double > bupp, std::string label, unsigned int idiff, double norm)
- void **SetBinning1D** (std::vector< double > blow, std::vector< double > bupp, std::string label, unsigned int idiff, std::vector< double > vnorm)
- void **SetBinningND** (std::vector< double > bgrid, unsigned int ndim, std::vector< int > idiff)
- void **SetBinningND** (std::vector< std::vector< double > > bgrid, unsigned int ndim, std::vector< int > idiff)
- void **ReadCoefficientSpecificVariables** ()
- void ReadScaleFactors ()
- void **InitVariablesInCoefficientTable** ()
- void InitCoeffTable ()
- void InitInterpolationKernels ()
- fastNLOInterpolBase * MakeInterpolationKernels (std::string KernelName, double xdn, double xup)
- void **InitGrids** ()
- void GetWarmupValues ()
- bool CheckWarmupConsistency ()

  Check consistency of warmup bin-grid and variables with steering values.
- void UseBinGridFromWarmup ()

  Use bin grid as given in the warmup table.
- int CheckWarmupValuesIdenticalWithBinGrid (std::vector< std::pair< double, double > > &wrmmu)

  Check if warmup values are possibly identical with bin grid.
- void RoundValues (std::vector< std::pair< double, double > > &wrmmu, int nth)

  Round values to closes value by at most 1%.
- int **GetNthRelevantDigit** (double val, int n)
- std::vector< std::vector< std::pair< int, int > > > ReadPartonCombinations (int ord, const std::vector< std::vector< int > > &PartonCombinations)

  Read PDFCoeff from steering.
- int GetBin ()

  get bin number from 'scenario' observables
- int GetXIndex (const int &Obsbin, const int &x1bin, const int &x2bin) const

  get x-index in case of two hadrons.
- int GetNxmax (const std::vector< double > *xGrid1, const std::vector< double > *xGrid2)

  get maximum x-index
- bool CheckWeightIsFinite ()

  Check if weight is reasonable.
- void HalfMatrixCheck (double x1, double x2, int &xmin, int &xmax, int &subproc) const

  check x-values in case of half-matrix notation (pp,ppbar), and exchange if necessary.
- void UpdateWarmupArrays ()
- void InitWarmupArrays ()
- void **OutWarmup** (std::ostream &=std::cout)
- std::vector< double > GetColumnFromTable (const std::vector< std::vector< double > > &table, int iCol)

  Get a column from a table.
- std::string **GetWarmupHeader** (int iScale, std::string minmax)
- void FillWeightCache (int scalevar)

  Fill weight into cache, merge weights for identical phase space points.
- void FlushCache ()

  Fill weights from cache into table.

Protected Attributes

- std::string fWarmupFilename

  File name of the warmup table.
- bool fIsWarmup

  is it a warmup run?
- int fWarmupXMargin

  margin for x-value: First digit in 'e' notation (e.g. margin=4: x=6.6e-3 -> 6.2e-3)
- int fWarmupNDigitMu1

  Digits of warmup values for scale 1.
- int fWarmupNDigitMu2

  Digits of warmup values for scale 2.
- int fIOrd

  order of alpha_s of run
- bool fIsFlexibleScale

  is it a flexible scale table?
- bool fApplyPDFReweight

  shall the PDF reweight be applied.
- std::string fSteerfile

  filename of steering file.
- int fObsBin

  ObsBin from 'last' 'Fill()'-call.
- fnloScenario fLastScen

  keep information of scenario from last 'Fill()'-call
- fastNLOReader * fReader

  PDF and alpha_s interface for reference tables.
- fastNLO::GeneratorConstants fGenConsts

  Generator specific constants.
- fastNLO::ProcessConstants fProcConsts

  Process specific constants.
- fastNLO::ScenarioConstants fScenConsts

  Scenario specific constants.
- fastNLO::WarmupConstants fWarmupConsts

  Warmup specific constants.
- std::vector< int > fSymProc

  necessary for half-matrix notation
- std::vector< double > fScaleFac

  Scale factors. Needed for fixed-scale tables.
- std::vector< fastNLOInterpolBase * > fKernX1

  Interpolation kernel for x-interpolation.
- std::vector< fastNLOInterpolBase * > fKernX2

  Interpolation kernel for x-interpolation.
- std::vector< fastNLOInterpolBase * > fKernMu1

  Interpolation kernel for mu1-interpolation.
- std::vector< fastNLOInterpolBase * > fKernMu2

  Interpolation kernel for mu2-interpolation.
- std::vector< std::vector< fastNLOInterpolBase * > > fKernMuS

  Interpolation kernels for each scale var for fixed-scale tables.
- std::vector< std::pair< double, double > > fWMu1

  array of warmup-up values
- std::vector< std::pair< double, double > > fWMu2

array of warmup-values

- std::vector< std::pair< double, double > > fWx

  array of warmup-values
- std::vector< std::pair< double, double > > fWMu1Rnd

  copy of warm-up array for rounding
- std::vector< std::pair< double, double > > fWMu2Rnd

  copy of warm-up array for rounding
- std::vector< std::pair< double, double > > fWxRnd

  copy of warm-up array for rounding
- int **fCacheMax**
- std::vector< std::pair< fnloScenario, fnloEvent > > fWeightCache

  cache for fill-weights
- struct fastNLOCreate::fnloStats **fStats**

Additional Inherited Members

5.12.1 Constructor & Destructor Documentation

5.12.1.1 fastNLOCreate() [1/2]

```
fastNLOCreate::fastNLOCreate (
            const std::string & steerfile,
            std::string warmupfile = "",
            bool shouldReadSteeringFile = true )
```

fastNLOCreate. A class for creating a fastNLO Table which contains exactly one table of coefficients.

Member variables are initialized by reading in a steering file.

Following information has to be obtained from the generator and is NOT obtained from steering:

- Order in alpha_s of leading-order process
- Center of mass energy
- Order of calculation (LO=0, NLO=1, NNLO=2)

5.12.1.2 fastNLOCreate() [2/2]

```
fastNLOCreate::fastNLOCreate (
            const fastNLO::GeneratorConstants & GenConsts,
            const fastNLO::ProcessConstants & ProcConsts,
            const fastNLO::ScenarioConstants & ScenConsts,
            const fastNLO::WarmupConstants & WarmConsts )
```

Constructor of fastNLOCreate

Pass all needed steering paramters through GeneratorConstants, ProcessConstants, ScenarioConstants and WarmupConstants (see GeneratorConstants.h file for details)

No steering or warmup file is read in

Set constants from arguments

No WarmupFile required, a pseudo-WarmupFilename is defined here

5.12.2 Member Function Documentation

5.12.2.1 AdjustWarmupValues()

```
void fastNLOCreate::AdjustWarmupValues ( )
```

Round warmup values to more likely values.

Adjust warmup-values found to supposedly more reasonable values.

Do this ONLY ONCE on COPY of actual values just before writing out to the warm-up table.

1. Round warm-up values up/down, if they are 4% close to the bin boundary -> if more than 70% of all bins are close to the bin boundary, then round all
2. Round values up/down, by mostly 3% to next reasonable value
3. Round lower x-values down by 20%

5.12.2.2 CheckProcConsts()

```
bool fastNLOCreate::CheckProcConsts ( )
```

check process constants to be different from defaults

Check that reasonable values different from the defaults have been set

5.12.2.3 CheckWarmupConsistency()

```
bool fastNLOCreate::CheckWarmupConsistency ( )  [protected]
```

Check consistency of warmup bin-grid and variables with steering values.

check if warmup values are consistent with steering card check if number of bins is consistent

5.12.2.4 CheckWarmupValuesIdenticalWithBinGrid()

```
int fastNLOCreate::CheckWarmupValuesIdenticalWithBinGrid (
            std::vector< std::pair< double, double > > & wrmmu )  [protected]
```

Check if warmup values are possibly identical with bin grid.

Check, where scale variable is identical with measured variable and hence warmu-values should be identical with bin grid.

returns idim, if identity was found returns -1 else.

If more than 70% of all bins are closer than 4% to the bin boundary, then identity is assumed

5.12.2.5 CheckWeightIsFinite()

```
bool fastNLOCreate::CheckWeightIsFinite ( )  [protected]
```

Check if weight is reasonable.

check if weights are finite

5.12.2.6 DivideCoefficientsByBinSize()

```
void fastNLOCreate::DivideCoefficientsByBinSize ( )
```

Divide all coefficients by bin size.

Divide all coefficients by binsize

5.12.2.7 Fill()

```
void fastNLOCreate::Fill (
            int scalevar = 0 )
```

fill event quantities in fastNLO table. Call it for every subprocess.

Fill values, which are stored in 'Event' and 'Scenario' into fastNLO table.

5.12.2.8 FillContribution()

```
void fastNLOCreate::FillContribution (
            int scalevar = 0 )  [protected]
```

fill contribution into table

read information from 'Event' and 'Scenario' do the interpolation and fill into the tables.

5.12.2.9 FillContributionFixDIS()

```
void fastNLOCreate::FillContributionFixDIS (
            fastNLOCoeffAddFix * c,
            int ObsBin,
            int scalevar )  [protected]
```

fill fixed scale contribution in DIS

read information from 'Event' and 'Scenario' do the interpolation and fill into the tables.

5.12.2.10 FillContributionFixHHC()

```
void fastNLOCreate::FillContributionFixHHC (
            fastNLOCoeffAddFix * c,
            int ObsBin,
            int scalevar )  [protected]
```

fill fixed scale table in pp/ppbar

read informatio from 'Event' and 'Scenario' do the interpolation and fill into the tables.

5.12.2.11 FillContributionFlexDIS()

```
void fastNLOCreate::FillContributionFlexDIS (
            fastNLOCoeffAddFlex * c,
            int ObsBin )  [protected]
```

fill flexible scale contribution in DIS

read information from 'Event' and 'Scenario' do the interpolation and fill into the tables.

5.12.2.12 FillContributionFlexHHC()

```
void fastNLOCreate::FillContributionFlexHHC (
            fastNLOCoeffAddFlex * c,
            int ObsBin )  [protected]
```

fill flexible scale contribution in pp/ppbar

read informatio from 'Event' and 'Scenario' do the interpolation and fill into the tables.

5.12.2.13 FillRefContribution()

```
void fastNLOCreate::FillRefContribution (
            int scalevar = 0 )  [protected]
```

fill contribution if this is a reference table

This is a reference table. Fill contribution as it would be a cross section

5.12.2.14 FillWeightCache()

```
void fastNLOCreate::FillWeightCache (
            int scalevar )  [protected]
```

Fill weight into cache, merge weights for identical phase space points.

fill event into cache

5.12.2.15 GetBin()

```
int fastNLOCreate::GetBin ( )  [protected]
```

get bin number from 'scenario' observables

get bin number, using observables from Scenario

5.12.2.16 GetColumnFromTable()

```
std::vector< double > fastNLOCreate::GetColumnFromTable (
            const std::vector< std::vector< double > > & table,
            int iCol )  [protected]
```

Get a column from a table.

Get a column from a table

5.12.2.17 GetIsFlexibleScale()

```
bool fastNLOCreate::GetIsFlexibleScale ( ) const  [inline]
```

Get flag for warmup table.

Get if flexible-scale table is requested

5.12.2.18 GetWarmupValues()

```
void fastNLOCreate::GetWarmupValues ( )  [protected]
```

GetWarmupValues. Checks if warmup-table exists and initialized member variable fIsWarmup

Check if warmup values already set by user

Try to get warmup values from steering

5.12.2.19 GetXIndex()

```
int fastNLOCreate::GetXIndex (
            const int & Obsbin,
            const int & x1bin,
            const int & x2bin ) const  [inline], [protected]
```

get x-index in case of two hadrons.

get index if 1 or two hadrons are involved

5.12.2.20 HalfMatrixCheck()

```
void fastNLOCreate::HalfMatrixCheck (
            double x1,
            double x2,
            int & xmin,
            int & xmax,
            int & subproc ) const  [inline], [protected]
```

check x-values in case of half-matrix notation (pp,ppbar), and exchange if necessary.

check if half-matrix notation if half-matrix notation, and xmin-node is larger than xmax-node exchange suprocesses according to fSymProc and adjust x-nodes.

5.12.2.21 InitCoeffTable()

```
void fastNLOCreate::InitCoeffTable ( )  [protected]
```

create a coeff table

set 'usual' variables for perturbative calculations

read in process specific variables

5.12.2.22 InitInterpolationKernels()

```
void fastNLOCreate::InitInterpolationKernels ( )  [protected]
```

initialize members for interpolation

5.12.2.23 InitWarmupArrays()

```
void fastNLOCreate::InitWarmupArrays ( )  [protected]
```

initialize arrays to store and determine warm-up values including copy for later rounding and write out initialize with reasonable values

5.12.2.24 Instantiate()

```
void fastNLOCreate::Instantiate ( )  [protected]
```

Instantiate all internal members and prepare for filling

5.12.2.25 MakeInterpolationKernels()

```
fastNLOInterpolBase * fastNLOCreate::MakeInterpolationKernels (
            std::string KernelName,
            double xdn,
            double xup )  [protected]
```

This function identifies the string-identifier and creates the corresponding fastNLO Interpolation kernel

5.12.2.26 MultiplyCoefficientsByBinSize()

```
void fastNLOCreate::MultiplyCoefficientsByBinSize ( )
```

Multiply all coefficients by bin size.

Multiply all coefficients by binsize

5.12.2.27 MultiplyCoefficientsByConstant()

```
void fastNLOCreate::MultiplyCoefficientsByConstant (
            double c )
```

Multiply all coefficients with a constant factor c.

Divide all coefficients by binsize

5.12.2.28 NormalizeCoefficients() [1/2]

```
void fastNLOCreate::NormalizeCoefficients (
            double wgt = 1 )
```

Set number of events to wgt and adjust coefficients accordingly.

Set number of events to wgt (default=1) and weight coefficients in sigmatilde accordingly This means, that the information about the number of events is essentially lost (now remained stored in fWgt)

5.12.2.29 NormalizeCoefficients() [2/2]

```
void fastNLOCreate::NormalizeCoefficients (
            const std::vector< std::vector< double > > & wgtProcBin )
```

Set number of events to wgt (with wgt differently for each subproc. and bin) resulting Nevt will become 1.

Set number of events to wgtProcBin[iProc][iBin] sigmatilde is weighted accordingly.

5.12.2.30 PrintStats()

```
void fastNLOCreate::PrintStats ( ) const  [inline]
```

Clear coefficient table.

Print statistics

5.12.2.31 ReadBinningFromScenarioConsts()

```
void fastNLOCreate::ReadBinningFromScenarioConsts ( )  [protected]
```

read in binning from ScenConsts

5.12.2.32 ReadBinSize()

```
void fastNLOCreate::ReadBinSize ( )  [protected]
```

initialize BinSize either from steering or from fScenConsts

5.12.2.33 ReadGenAndProcConstsFromSteering()

```
void fastNLOCreate::ReadGenAndProcConstsFromSteering ( )  [protected]
```

ReadGenAndProcConstsFromSteering() If generator and process constants have not been set in the constructor, then obtain these values from the steering file.

5.12.2.34 ReadPartonCombinations()

```
vector< vector< pair< int, int > > > fastNLOCreate::ReadPartonCombinations (
            int ord,
            const std::vector< std::vector< int > > & PartonCombinations )  [protected]
```

Read PDFCoeff from steering.

Read PDF linear combinations from steering file and convert to internal format

5.12.2.35 ReadScaleFactors()

```
void fastNLOCreate::ReadScaleFactors ( )  [protected]
```

read scale factors from steering and init member fScaleFac

5.12.2.36 ReadSteering()

```
void fastNLOCreate::ReadSteering (
            std::string steerfile,
            std::string steeringNameSpace = "",
            bool shouldReadSteeringFile = true )  [protected]
```

read steering file

read in steering file The filename of the steering file is used as the 'namespace' of keys in read_steer if there is no steeringNameSpace given explicitly

Remove extension from steerfile to define default steering namespace

Set verbosity from steering or to default WARNING

Set WarmupFilename from steering

5.12.2.37 RoundValues()

```
void fastNLOCreate::RoundValues (
            std::vector< std::pair< double, double > > & wrmmu,
            int nth )  [protected]
```

Round values to closes value by at most 1%.

Round warmup values up (down) if third relevant digit is a 9 (0) lower values are only rounded down, upper values are only rounded up

5.12.2.38 SetGenConstsDefaults()

```
void fastNLOCreate::SetGenConstsDefaults ( )
```

set defaults for generator constants

Set default values for generator constants

< X section units of coefficients passed to fastNLO (neg. power of 10: pb->12, fb->15)

5.12.2.39 SetIsReferenceTable()

```
fastNLOReader * fastNLOCreate::SetIsReferenceTable (
            fastNLOReader * fnloread = NULL )
```

set this table/contribution to become a reference contribution

set this table/contribution to become a reference contribution If fnloread is set to NULL, the weights are assumed to be already multiplied by PDF and alpha_s values. If fnloread is provided, then it is assumed that the weights have the same units and format as for the filling of default tables and those need to be multiplied by PDF and alpha_s values.

Function returns the input pointer without changes.

5.12.2.40 SetLoOrder()

```
void fastNLOCreate::SetLoOrder (
            int LOOrd )
```

set number of events. This is only mandatory, before calling WriteTable().

set order of alpha_s for leading order process.

5.12.2.41 SetOrderOfAlphasOfCalculation()

```
void fastNLOCreate::SetOrderOfAlphasOfCalculation (
            unsigned int ord )
```

set absolute order of alpha_s

set order of alpha_s of this calculation it must be: iLeadingOrder + iHigherOrder ; for instance: 3-jet-production in NLO = 4!

5.12.2.42 SetProcConstsDefaults()

```
void fastNLOCreate::SetProcConstsDefaults ( )
```

set defaults for process constants

Set default values for process constants

< Order in alpha_s of leading order process

< No. of PDFs involved

< No. of LO subprocesses

< No. of NLO subprocesses

< No. of NNLO subprocesses

< Flag 1 to define PDF linear combinations of partonic subprocesses (e.g. hh –> jets: 3)

< Flag 2 to define PDF linear combinations (dep. on IPDFdef1; for 3 e.g. 1 for jet specific LCs, 121 for generic 11x11 matrix)

< Flag 3 to define PDF LCs at LO (dep. on IPDFdef1, IPDFdef2; for 3, 1 e.g. 6 subprocesses, ignored for IPD↩Fdef2==121)

< Flag 3 to define PDF LCs at NLO (dep. on IPDFdef1, IPDFdef2; for 3, 1 e.g. 7 subprocesses, ignored for IPDFdef2==121)

< Flag 3 to define PDF LCs at NNLO (dep. on IPDFdef1, IPDFdef2; for 3, 1 e.g. 7 subprocesses, ignored for IPDFdef2==121)

< Define internal storage mode for PDF LCs (dep. on NPDF; e.g. for 1: 0 for linear, for 2: 1 for half- or 2 for full-matrix)

5.12.2.43 TestParameterInSteering()

```
bool fastNLOCreate::TestParameterInSteering (
            const std::string & label ) const
```

Test on existence of user-defined parameter name in steering card.

Get flag if parameter exists in steering card

5.12.2.44 UpdateWarmupArrays()

```
void fastNLOCreate::UpdateWarmupArrays ( )  [protected]
```

Update the warmup-arrays fWMu1, fWx und fWMu2

5.12.2.45 UseBinGridFromWarmup()

```
void fastNLOCreate::UseBinGridFromWarmup ( )  [protected]
```

Use bin grid as given in the warmup table.

initialialize all binning related variables with values stored in the warmup file.

5.12.2.46 WriteTable()

```
void fastNLOCreate::WriteTable ( )  [virtual]
```

Write fastNLO table to disk.

Write fastNLO file to disk

Reimplemented from fastNLOTable.

5.12.2.47 WriteWarmupTable()

```
void fastNLOCreate::WriteWarmupTable ( )
```

Write fastNLO table to disk and set number of events.

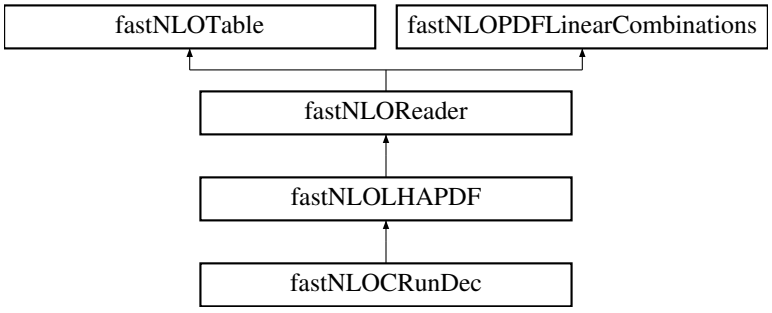
Write the warmup table to disk.

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

- fastnlotoolkit/include/fastnlotk/fastNLOCreate.h
- fastnlotoolkit/fastNLOCreate.cc

5.13 fastNLOCRunDec Class Reference

Inheritance diagram for fastNLOCRunDec:



Public Member Functions

- **fastNLOCRunDec** (std::string name)
- **fastNLOCRunDec** (std::string name, std::string LHAPDFFile, int PDFSet)
- void **SetMz** (double Mz)
- void **SetNFlavor** (int nflavor)
- void **SetNLoop** (int nloop)
- void **SetQMass** (int pdgid, double qmass)
- void **SetAlphasMz** (double AlphasMz, bool ReCalcCrossSection)
- void **SetLHAPDFValues** ()
- void **SetPDGValues** ()
- double **GetMz** () const
- double **GetQMass** (int pdgid) const
- int **GetNFlavor** () const
- int **GetNLoop** () const
- double **GetAlphasMz** () const

Protected Member Functions

- double **EvolveAlphas** (double Q) const
- void **InitCRunDec** ()

Protected Attributes

- CRunDec * **crundec**
- double **fAlphasMz**
- double **fMz**
- int **fnFlavor**
- int **fnLoop**
- double **QMass** [6]

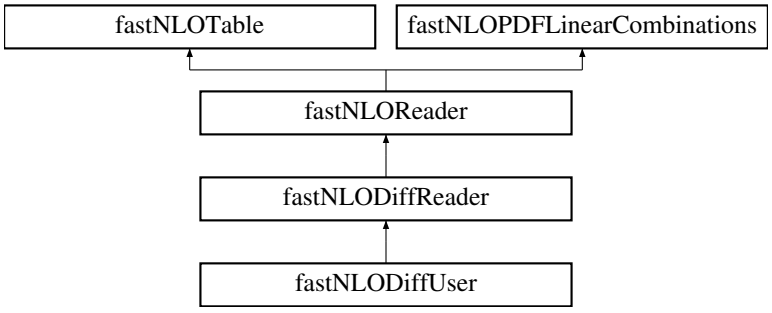
Additional Inherited Members

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

- fastnlotoolkit/include/fastnlotk/fastNLOCRunDec.h
- fastnlotoolkit/fastNLOCRunDec.cc

5.14 fastNLODiffReader Class Reference

Inheritance diagram for fastNLODiffReader:



Public Member Functions

- **fastNLODiffReader** (std::string filename)
- void **SetXPomSlicing** (int nStep, double ∗xpom, double ∗dxpom)
- void **SetXPomLogSlicing** (int nStep, double xpommin, double xpommax)
- void **SetXPomLinSlicing** (int nStep, double xpommin, double xpommax)
- void **SetXPomExpSlicing** (int nStep, double xpommin, double xpommax)
- void **SetZRange** (double zmin, double zmax)
- double **GetZRangeMin** ()
- double **GetZRangeMax** ()
- std::vector< double > **GetCrossSection** ()
- void **CalcCrossSection** ()
- std::vector< double > **GetDiffCrossSection** ()
- void **FillPDFCache** (bool ReCalcCrossSection=false)
- std::vector< double > **GetReferenceCrossSection** ()
- void **PrintCrossSectionsWithReference** ()

Protected Member Functions

- virtual double **EvolveAlphas** (double Q) const =0
- virtual bool **InitPDF** ()=0
- std::vector< double > **GetXFX** (double xp, double muf) const
- virtual std::vector< double > **GetDiffXFX** (double xpom, double zpom, double muf) const =0

Protected Attributes

- double **fxpom**
- double **fzmin**
- double **fzmax**
- std::vector< double > **fxPoms**
- std::vector< double > **fdxPoms**

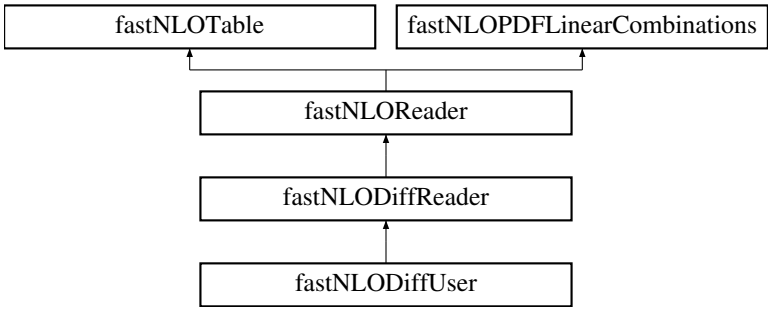
Additional Inherited Members

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

- fastnlotoolkit/include/fastnlotk/fastNLODiffReader.h
- fastnlotoolkit/fastNLODiffReader.cc

5.15 fastNLODiffUser Class Reference

Inheritance diagram for fastNLODiffUser:



Public Member Functions

- **fastNLODiffUser** (std::string filename)

Protected Member Functions

- double **EvolveAlphas** (double Q) const
- bool **InitPDF** ()
- std::vector< double > **GetDiffXFX** (double xpom, double zpom, double muf) const

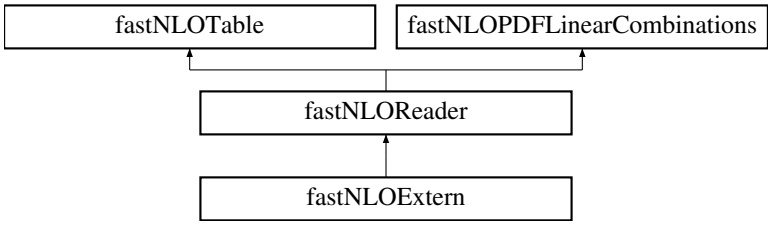
Additional Inherited Members

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

- fastnlotoolkit/include/fastnlotk/fastNLODiffUser.h
- fastnlotoolkit/fastNLODiffUser.cc

5.16 fastNLOExtern Class Reference

Inheritance diagram for fastNLOExtern:



Public Member Functions

- **fastNLOExtern** (std::string tablename)

Protected Member Functions

- virtual bool **InitPDF** ()
- virtual vector< double > **GetXFX** (double x, double muf) const
- virtual double **EvolveAlphas** (double Q) const

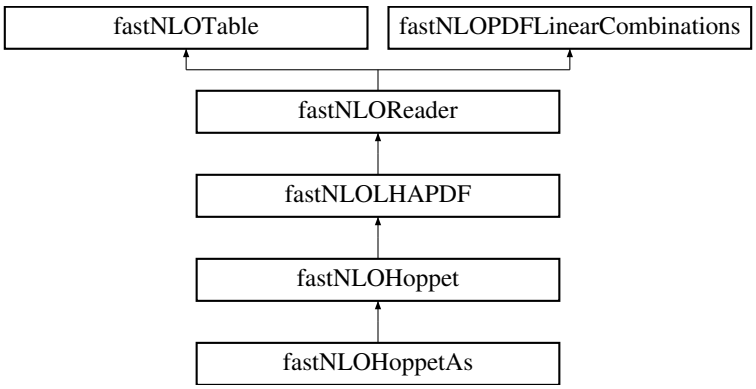
Additional Inherited Members

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

- fortranext/fastNLOFI_cc.cc

5.17 fastNLOHoppet Class Reference

Inheritance diagram for fastNLOHoppet:



Public Member Functions

- **fastNLOHoppet** (std::string name)
- **fastNLOHoppet** (std::string name, std::string LHAPDFFile, int PDFSet)
- void **SetMz** (double Mz)
- void **SetNFlavor** (int nflavor)
- void **SetNLoop** (int nloop)
- void **SetQMass** (int pdgid, double qmass)
- void **SetAlphasMz** (double AlphasMz, bool ReCalcCrossSection=false)
- void **SetLHAPDFValues** ()
- void **SetPDGValues** ()
- virtual bool **InitPDF** ()
- double **GetMz** () const
- double **GetQMass** (int pdgid) const
- int **GetNFlavor** () const
- int **GetNLoop** () const
- double **GetAlphasMz** () const

Protected Member Functions

- virtual double **EvolveAlphas** (double Q) const
- virtual std::vector< double > **GetXFX** (double xp, double muf) const

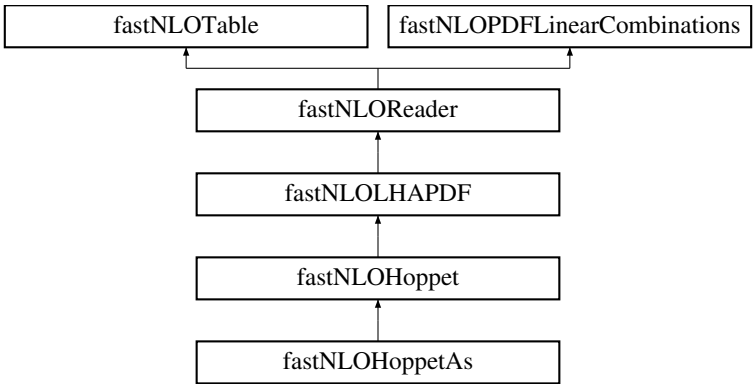
Additional Inherited Members

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

- fastnlotoolkit/include/fastnlotk/fastNLOHoppet.h
- fastnlotoolkit/fastNLOHoppet.cc

5.18 fastNLOHoppetAs Class Reference

Inheritance diagram for fastNLOHoppetAs:



Public Member Functions

- **fastNLOHoppetAs** (std::string name)
- **fastNLOHoppetAs** (std::string name, std::string LHAPDFFile, int PDFSet)

Protected Member Functions

- virtual std::vector< double > **GetXFX** (double xp, double muf) const

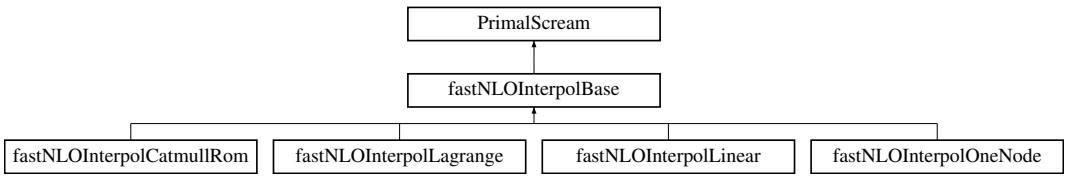
Additional Inherited Members

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

- fastnlotoolkit/include/fastnlotk/fastNLOHoppetAs.h
- fastnlotoolkit/fastNLOHoppetAs.cc

5.19 fastNLOInterpolBase Class Reference

Inheritance diagram for fastNLOInterpolBase:



Public Member Functions

- **fastNLOInterpolBase** (double min, double max, int nMinNodes)
- const std::vector< std::pair< int, double > > & **GetNodeValues** (double val)
- std::vector< std::pair< int, double > > * **GetNodeValuesPtr** (double val)
- void **MakeGrids** (fastNLOGrid::GridType type, int nNodes)
- void **MakeGridsWithNNodesPerMagnitude** (fastNLOGrid::GridType type, int nNodes)
- void **RemoveLastNode** ()
- void **PrintGrid** ()
- const std::vector< double > & **GetGrid** () const
- const std::vector< double > * **GetGridPtr** () const
- const std::vector< double > & **GetHGrid** () const
- double **GetDelta** (double)
- bool **CheckX** (double &)

Static Public Member Functions

- static fastNLOGrid::GridType **TranslateGridType** (std::string in)

Protected Member Functions

- void **SetGrid** (std::vector< double > grid)
- void **SetHGrid** (std::vector< double > grid)
- void **MakeGrids** (double min, double max, int nNodes)
- std::vector< double > **MakeGridFromHGrid** (std::vector< double > g)
- std::vector< double > **MakeLinearGrid** (double min, double max, int nNodes)
- virtual void **CalcNodeValues** (std::vector< std::pair< int, double > > &nodes, double val)=0
- int **FindLargestPossibleNode** (double)
- double **Function_loglog025** (double mu)
- double **Function_loglog025_inv** (double mu)
- double **Function_loglog** (double mu)
- double **Function_loglog_inv** (double mu)
- double **Function_x** (double mu)
- double **Function_x_inv** (double mu)
- double **Function_log10** (double x)
- double **Function_log10_inv** (double x)
- double **Function_sqrtlog10** (double x)
- double **Function_sqrtlog10_inv** (double x)
- double **Function_3rdrtlog10** (double mu)
- double **Function_3rdrtlog10_inv** (double mu)
- double **Function_4thrtlog10** (double mu)
- double **Function_4thrtlog10_inv** (double mu)
- std::vector< double > **HGrid_loglog025_inv** (std::vector< double > grid)
- std::vector< double > **HGrid_loglog_inv** (std::vector< double > grid)
- std::vector< double > **HGrid_log10_inv** (std::vector< double > grid)
- std::vector< double > **HGrid_sqrtlog10_inv** (std::vector< double > grid)
- std::vector< double > **HGrid_4thrtlog10_inv** (std::vector< double > grid)
- std::vector< double > **HGrid_3rdrtlog10_inv** (std::vector< double > grid)
- int **GetNMod** () const
- double **GetHx** (double)

Protected Attributes

- std::vector< std::pair< int, double > > **fNodes**
- int **fNMinNodes**
- double **fvalmin**
- double **fvalmax**
- double **fLastVal**
- bool **fLastGridPointWasRemoved**
- fastNLOGrid::GridType **fdm**
- std::vector< double > **fgrid**
- std::vector< double > **fHgrid**
- int **fnmod**

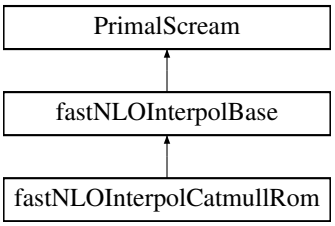
Additional Inherited Members

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

- fastnlotoolkit/include/fastnlotk/fastNLOInterpolBase.h
- fastnlotoolkit/fastNLOInterpolBase.cc

5.20 fastNLOInterpolCatmullRom Class Reference

Inheritance diagram for fastNLOInterpolCatmullRom:



Public Member Functions

- **fastNLOInterpolCatmullRom** (double min, double max)
- void **CalcNodeValues** (std::vector< std::pair< int, double > > &nodes, double val)

Additional Inherited Members

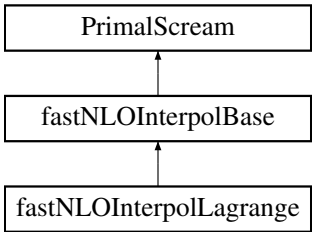
The documentation for this class was generated from the following files:

- fastnlotoolkit/include/fastnlotk/fastNLOInterpolCatmullRom.h
- fastnlotoolkit/fastNLOInterpolCatmullRom.cc

5.21 fastNLOInterpolLagrange Class Reference

`#include <fastnlotoolkit/include/fastnlotk/fastNLOInterpolLagrange.h>`

Inheritance diagram for fastNLOInterpolLagrange:



Public Member Functions

- **fastNLOInterpolLagrange** (double min, double max)
- void CalcNodeValues (std::vector< std::pair< int, double > > &nodes, double val)

Additional Inherited Members

5.21.1 Detailed Description

fastNLOInterpolLagrange

Interpolation routines for lagrange interpolation of second order polynomials.

5.21.2 Member Function Documentation

5.21.2.1 CalcNodeValues()

```
void fastNLOInterpolLagrange::CalcNodeValues (
            std::vector< std::pair< int, double > > & nodes,
            double val )  [virtual]
```

Performs interpolation of value value on grid 'fgrid'. uses distance measure 'fdm' returns for for all relevant grid points

- the integer number of that node
- the vale, that this node obtains.

Implements fastNLOInterpolBase.

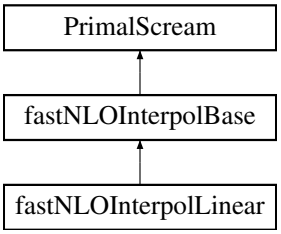
The documentation for this class was generated from the following files:

- fastnlotoolkit/include/fastnlotk/fastNLOInterpolLagrange.h
- fastnlotoolkit/fastNLOInterpolLagrange.cc

5.22 fastNLOInterpolLinear Class Reference

`#include <fastnlotoolkit/include/fastnlotk/fastNLOInterpolLinear.h>`

Inheritance diagram for fastNLOInterpolLinear:



Public Member Functions

- **fastNLOInterpolLinear** (double min, double max)
- void CalcNodeValues (std::vector< std::pair< int, double > > &nodes, double val)

Additional Inherited Members

5.22.1 Detailed Description

fastNLOInterpolLinear

Interpolation routines for linear interpolation.

5.22.2 Member Function Documentation

5.22.2.1 CalcNodeValues()

```
void fastNLOInterpolLinear::CalcNodeValues (
            std::vector< std::pair< int, double > > & nodes,
            double val )  [virtual]
```

Performs interpolation of value value on grid 'fgrid'. uses distance measure 'fdm' returns for for all relevant grid points

- the integer number of that node
- the value, which this node obtains.

Implements fastNLOInterpolBase.

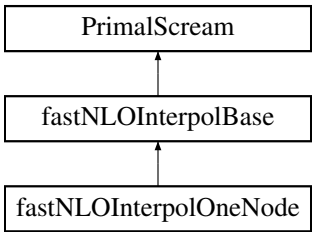
The documentation for this class was generated from the following files:

- fastnlotoolkit/include/fastnlotk/fastNLOInterpolLinear.h
- fastnlotoolkit/fastNLOInterpolLinear.cc

5.23 fastNLOInterpolOneNode Class Reference

`#include <fastnlotoolkit/include/fastnlotk/fastNLOInterpolOneNode.h>`

Inheritance diagram for fastNLOInterpolOneNode:



Public Member Functions

- **fastNLOInterpolOneNode** (double min, double max)
- void **CalcNodeValues** (std::vector< std::pair< int, double > > &nodes, double val)

Additional Inherited Members

5.23.1 Detailed Description

fastNLOInterpolOneNode

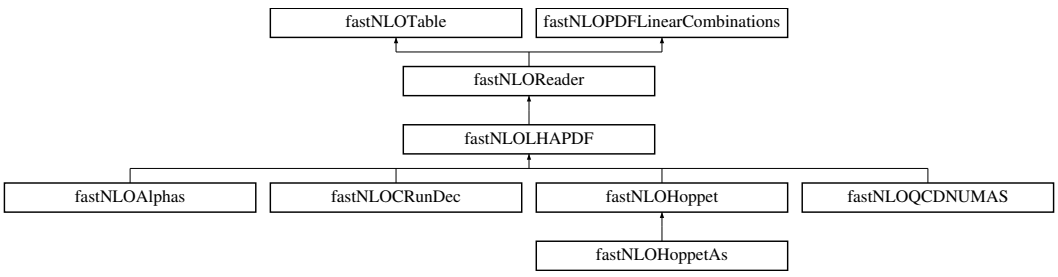
No interpolation is performed, but all values are stored at one single node. The 'x'-value of the node is calculated from the H-function.

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

- fastnlotoolkit/include/fastnlotk/fastNLOInterpolOneNode.h
- fastnlotoolkit/fastNLOInterpolOneNode.cc

5.24 fastNLOLHAPDF Class Reference

Inheritance diagram for fastNLOLHAPDF:



Public Member Functions

- **fastNLOLHAPDF** (std::string name)
- **fastNLOLHAPDF** (const fastNLOTable &)
- **fastNLOLHAPDF** (std::string name, std::string LHAPDFfile, int PDFSet=0)
- **fastNLOLHAPDF** (const fastNLOTable &, std::string LHAPDFfile, int PDFSet=0)
- virtual void **InitEvolveAlphas** ()
- virtual void **SetMz** (double Mz)
- virtual void **SetNFlavor** (int nflavor)
- virtual void **SetNLoop** (int nloop)
- virtual void **SetAlphasMz** (double AlphasMz, bool ReCalcCrossSection=false)
- virtual void **SetQMass** (int pdgid, double mq)
- void **SetLHAPDFFilename** (std::string filename)
- void **SetLHAPDFMember** (int set)
- std::string **GetLHAPDFFilename** () const
- int **GetIPDFMember** () const
- int **GetNPDFMembers** () const
- int **GetNPDFMaxMember** () const
- void **PrintPDFInformation** () const
- virtual double **GetQMass** (int pdgid) const
- int **GetNLoop** () const
- int **GetNFlavor** () const
- double **GetAlphasMz** () const
- XsUncertainty GetAsUncertainty (const fastNLO::EAsUncertaintyStyle eAsUnc)

  Return struct with vectors containing the cross section values and the selected a_s(M_Z) uncertainty.
- XsUncertainty GetAsUncertainty (const fastNLO::EAsUncertaintyStyle eAsUnc, bool lNorm)
- std::vector< std::vector< double > > GetAsUncertaintyVec (const fastNLO::EAsUncertaintyStyle eAsUnc)

  Function for use with pyext (TODO: Clean this up)
- XsUncertainty GetPDFUncertainty (const fastNLO::EPDFUncertaintyStyle ePDFUnc)

  Return struct with vectors containing the cross section values and the selected scale uncertainty.
- XsUncertainty **GetPDFUncertainty** (const fastNLO::EPDFUncertaintyStyle ePDFUnc, bool lNorm)
- std::vector< std::vector< double > > **GetPDFUncertaintyVec** (fastNLO::EPDFUncertaintyStyle)
- virtual double **EvolveAlphas** (double Q) const
- virtual bool **InitPDF** ()
- virtual std::vector< double > **GetXFX** (double xp, double muf) const

Protected Attributes

- std::string **fLHAPDFFilename**
- int **fnPDFs**
- int **fiPDFMember**
- double **fchksum**

Additional Inherited Members

5.24.1 Member Function Documentation

5.24.1.1 GetAsUncertainty()

```
XsUncertainty fastNLOLHAPDF::GetAsUncertainty (
            const fastNLO::EAsUncertaintyStyle eAsUnc,
            bool lNorm )
```

Cross section and absolute uncertainties

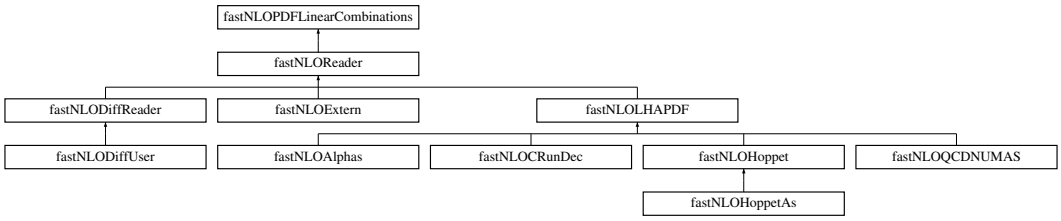
Divide by cross section != 0 to give relative uncertainties

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

- fastnlotoolkit/include/fastnlotk/fastNLOLHAPDF.h
- fastnlotoolkit/fastNLOLHAPDF.cc

5.25 fastNLOPDFLinearCombinations Class Reference

Inheritance diagram for fastNLOPDFLinearCombinations:



Public Member Functions

- std::vector< double > **CalcPDFLinearCombination** (const fastNLOCoeffAddBase *c, const std::vector< double > &pdfx1=std::vector< double >(), const std::vector< double > &pdfx2=std::vector< double >(), bool pdf2IsAntiParticle=false) const

Protected Member Functions

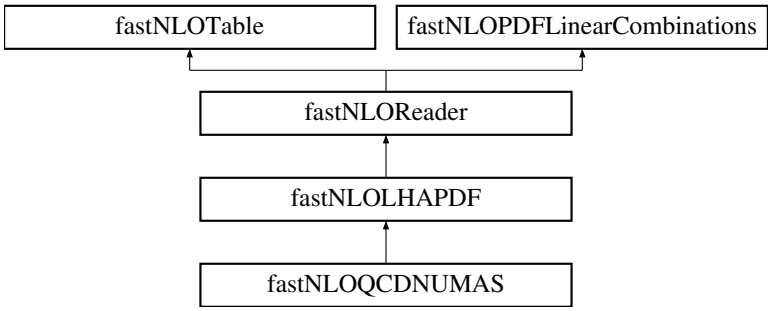
- std::vector< double > **MakeAntiHadron** (const std::vector< double > &hadron) const

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

- fastnlotoolkit/include/fastnlotk/fastNLOPDFLinearCombinations.h
- fastnlotoolkit/fastNLOPDFLinearCombinations.cc

5.26 fastNLOQCDNUMAS Class Reference

Inheritance diagram for fastNLOQCDNUMAS:



Public Member Functions

- **fastNLOQCDNUMAS** (std::string name)
- **fastNLOQCDNUMAS** (std::string name, std::string LHAPDFFile, int PDFSet)
- void **CalcCrossSection** ()
- void **InitEvolveAlphas** ()
- void **SetMz** (double Mz)
- void **SetNFlavor** (int nflavor)
- void **SetNLoop** (int nloop)
- void **SetQMass** (int pdgid, double qmass)
- void **SetAlphasMz** (double AlphasMz, bool ReCalcCrossSection=false)
- void **SetPDGValues** ()
- void **SetLHAPDFValues** ()
- double **GetMz** () const
- double **GetQMass** (int pdgid) const
- int **GetNFlavor** (int nflavor) const
- int **GetNLoop** () const
- double **GetAlphasMz** () const

Protected Member Functions

- double **EvolveAlphas** (double Q) const

Protected Attributes

- double **fAlphasMz**
- double **fMz**
- int **fnFlavor**
- int **fnLoop**
- double **QMass** [6]

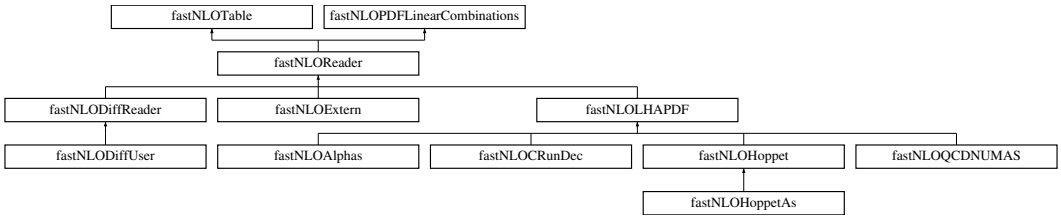
Additional Inherited Members

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

- fastnlotoolkit/include/fastnlotk/fastNLOQCDNUMAS.h
- fastnlotoolkit/fastNLOQCDNUMAS.cc

5.27 fastNLOReader Class Reference

Inheritance diagram for fastNLOReader:



Public Types

- typedef double(∗ mu_func) (double, double)

Public Member Functions

- **fastNLOReader** (std::string filename)
- **fastNLOReader** (const fastNLOTable &)
- fastNLOReader (const fastNLOReader &)
- void **SetFilename** (std::string filename)
- void InitScalevariation ()
- void **SetUnits** (fastNLO::EUnits Unit)
- bool SetContributionON (fastNLO::ESMCalculation eCalc, unsigned int Id, bool SetOn=true)

  Set contribution Id On/Off. Check for Id of a particular contribution with ContrId(...) or use ActivateContribution(...).
- bool ActivateContribution (fastNLO::ESMCalculation eCalc, fastNLO::ESMOrder eOrd, bool SetOn=true)

  Activate first found contribution of type eCalc and order eOrd.
- int ContrId (const fastNLO::ESMCalculation eCalc, const fastNLO::ESMOrder eOrder) const

  Find Id in table of contribution of type eCalc and order eOrd.
- void SetCoefficientUsageDefault ()

  Switch on LO and NLO contributions, deactivate other contributions.
- bool GetIsFlexibleScaleTable (fastNLOCoeffAddBase ∗ctest=NULL) const

  Get, if this table is a 'flexible-scale' table or not.
- void SetMuRFunctionalForm (fastNLO::EScaleFunctionalForm func)

  Set the functional form of Mu_R.
- void SetMuFFunctionalForm (fastNLO::EScaleFunctionalForm func)

  Set the functional form of Mu_F.
- void SetFunctionalForm (fastNLO::EScaleFunctionalForm func, fastNLO::EMuX kMuX)

  Set functional form of MuX.
- bool SetScaleFactorsMuRMuF (double xmur, double xmuf)

  Set scale factors for MuR and MuF.
- void SetExternalFuncForMuR (mu_func)

  Set external function for scale calculation (optional)
- void SetExternalFuncForMuF (mu_func)

  Set external function for scale calculation (optional)
- void SetExternalConstantForMuR (double MuR)

  Set value for mu_r if mu_r is chosen to be a constant value (i.e. m_t, or m_Z)
- void SetExternalConstantForMuF (double MuF)

  Set value for mu_f if mu_f is chosen to be a constant value (i.e. m_t, or m_Z)

- void **UseHoppetScaleVariations** (bool)
- void FillPDFCache (double chksum=0., bool lForce=false)

  Prepare for recalculation of cross section with 'new'/updated pdf.
- virtual std::vector< double > **GetXFX** (double x, double muf) const =0
- virtual bool **InitPDF** ()=0
- virtual double **EvolveAlphas** (double Q) const =0
- void FillAlphasCache (bool lForce=false)

  prepare for recalculation of cross section with new alpha_s value.
- void **ResetCache** ()
- void CalcCrossSection ()
- double RescaleCrossSectionUnits (double binsize, int xunits)
- std::vector< double > GetCrossSection ()

  Return vector with all cross section values.
- std::vector< double > GetCrossSection (bool lNorm)

  Return vector with all cross section values, normalize on request.
- std::vector< double > GetNormCrossSection ()

  Return vector with all normalized cross section values.
- std::vector< std::map< double, double > > **GetCrossSection_vs_x1** ()
- std::vector< std::map< double, double > > GetCrossSection_vs_x2 ()

  Cross section vs. x1 (XSection_vsX1[bin][<x,xs>])
- std::vector< double > GetReferenceCrossSection ()

  Cross section vs. x2 (XSection_vsX1[bin][<x,xs>])
- std::vector< double > GetKFactors ()

  DEPRECATED.
- std::vector< double > GetQScales ()

  Order (power of alpha_s) rel. to LO: 0 –> LO, 1 –> NLO.
- std::vector< std::vector< double > > GetCrossSection2Dim ()
- XsUncertainty GetScaleUncertainty (const fastNLO::EScaleUncertaintyStyle eScaleUnc)

  Return struct with vectors containing the cross section values and the selected scale uncertainty.
- XsUncertainty GetScaleUncertainty (const fastNLO::EScaleUncertaintyStyle eScaleUnc, bool lNorm)
- std::vector< std::vector< double > > GetScaleUncertaintyVec (const fastNLO::EScaleUncertaintyStyle e↩ ScaleUnc)

  Function for use with pyext (TODO: Clean this up)
- fastNLO::EScaleFunctionalForm **GetMuRFunctionalForm** () const
- fastNLO::EScaleFunctionalForm **GetMuFFunctionalForm** () const
- fastNLO::EUnits **GetUnits** () const
- mu_func **GetExternalFuncForMuR** ()
- mu_func **GetExternalFuncForMuF** ()
- double **GetScaleFactorMuR** () const
- double **GetScaleFactorMuF** () const
- int **GetScaleVariation** () const
- std::string GetScaleDescription (const fastNLO::ESMOrder eOrder, int iScale=0) const
- double GetNevt (const fastNLO::ESMOrder eOrder) const
- int GetNScaleVariations () const

  Get number of available scale variations.
- std::vector< double > GetScaleFactors () const

  Get list of available scale factors.
- void Print (int iprint) const

  Print basic info about fastNLO table and its contributions.
- void PrintContributionSummary (int iprint) const
- void PrintCrossSections () const

  Print cross sections (optimized for double-differential tables)

- void PrintCrossSectionsWithReference ()
- void **PrintTableInfo** (const int iprint=0) const
- void **PrintFastNLOTableConstants** (const int iprint=2) const
- void PrintCrossSectionsData () const

  DEPRECATED.
- void PrintCrossSectionsDefault (std::vector< double > kthc=std::vector< double >()) const

  DEPRECATED.
- void RunFastNLODemo ()

  DEPRECATED.
- bool TestXFX ()

  Test if XFX reasonable values.
- bool TestAlphas ()

  Test if EvolvaAlphas returns a reasonable value.

Public Attributes

- double fConst_MuR

  Constant value for the renormalization scale. Used only for flexible-scale tables and if requested.
- double fConst_MuF

  Constant value for the factorization scale. Used only for flexible-scale tables and if requested.

Protected Member Functions

- void **OrderCoefficients** ()
- void **StripWhitespace** (std::string *s)
- void **PrintScaleSettings** (fastNLO::EMuX kMuX=fastNLO::kMuR)
- void FillBlockBPDFLCsDISv20 (fastNLOCoeffAddFix *B)
- void FillBlockBPDFLCsDISv21 (fastNLOCoeffAddFlex *B, fastNLOCoeffAddFlex *B0=NULL)
- void FillBlockBPDFLCsHHCv20 (fastNLOCoeffAddFix *B)
- void FillBlockBPDFLCsHHCv21 (fastNLOCoeffAddFlex *B)
- void **CalcAposterioriScaleVariationMuR** ()
- void **CalcAposterioriScaleVariationMuF** ()
- void FillAlphasCacheInBlockBv20 (fastNLOCoeffAddFix *B)
- void FillAlphasCacheInBlockBv21 (fastNLOCoeffAddFlex *B)
- double CalcAlphas (double Q)
- double **CalcReferenceAlphas** ()
- double CalcNewPDFChecksum ()
- double CalcChecksum (double mu)
- bool **PrepareCache** ()
- void CalcReferenceCrossSection ()
- double CalcMu (fastNLO::EMuX kMuX, double scale1, double scale2, double scalefactor)
- double **FuncMixedOver1** (double scale1, double scale2)
- double **FuncMixedOver2** (double scale1, double scale2)
- double **FuncMixedOver4** (double scale1, double scale2)
- double **FuncMixed2s2Ov2** (double scale1, double scale2)
- double **FuncMixed2s2Ov4** (double scale1, double scale2)
- double **FuncPow4Sum** (double scale1, double scale2)
- double **FuncWgtAvg** (double scale1, double scale2)
- double **FuncLinearMean** (double scale1, double scale2)
- double **FuncLinearSum** (double scale1, double scale2)
- double **FuncMax** (double scale1, double scale2)

- double **FuncMin** (double scale1, double scale2)
- double **FuncProd** (double scale1, double scale2)
- double **FuncExpProd2** (double scale1, double scale2)
- void CalcCrossSectionv20 (fastNLOCoeffAddFix ∗B)
- void CalcCrossSectionv21 (fastNLOCoeffAddFlex ∗B)
- fastNLOCoeffAddBase ∗ **B_LO** () const
- fastNLOCoeffAddBase ∗ **B_NLO** () const
- fastNLOCoeffAddBase ∗ **B_NNLO** () const
- fastNLOCoeffBase ∗ **B_ThC** (int n=0)
- fastNLOCoeffAddBase ∗ **B_Any** () const
- bool SetScaleVariation (int scalevar)

  Choose the MuF scale variation table.

Protected Attributes

- std::string **ffilename**
- int **fScalevar**
- double **fScaleFacMuR**
- double **fScaleFacMuF**
- fastNLO::EScaleFunctionalForm **fMuRFunc**
- fastNLO::EScaleFunctionalForm **fMuFFunc**
- fastNLO::EUnits **fUnits**
- bool **fPDFSuccess**
- double **fPDFCached**
- double **fAlphasCached**
- mu_func Fct_MuR

  Function, if you define your functional form for your scale external.
- mu_func Fct_MuF

  Function, if you define your functional form for your scale external.
- std::vector< std::vector< bool > > bUseSMCalc

  switch calculations ON/OFF
- bool **fUseHoppet**
- std::vector< std::vector< fastNLOCoeffBase ∗> > BBlocksSMCalc

  BlockB's for SM corrections.
- std::vector< double > **XSection_LO**
- std::vector< double > **XSection**
- std::vector< double > **kFactor**
- std::vector< double > **QScale_LO**
- std::vector< double > **QScale**
- std::vector< std::map< double, double > > **fXSection_vsX1**
- std::vector< std::map< double, double > > fXSection_vsX2

  Cross section vs. x (XSection_vsX1[bin][<x,xs>])
- std::vector< double > **XSectionRef**
- std::vector< double > **XSectionRefMixed**
- std::vector< double > **XSectionRef_s1**
- std::vector< double > **XSectionRef_s2**

Additional Inherited Members

5.27.1 Member Typedef Documentation

5.27.1.1 mu_func

```
typedef double(* fastNLOReader::mu_func) (double, double)
```

fastNLOReader. Abstract base class for evaluation of fastNLO tables. Instantiatians must implement functions for PDF and alpha_s access.

5.27.2 Constructor & Destructor Documentation

5.27.2.1 fastNLOReader()

```
fastNLOReader::fastNLOReader (
            const fastNLOReader & other )
```

Copy constructor

5.27.3 Member Function Documentation

5.27.3.1 CalcAlphas()

```
double fastNLOReader::CalcAlphas (
            double Q )  [protected]
```

Internal method for calculating the alpha_s(mu)

5.27.3.2 CalcChecksum()

```
double fastNLOReader::CalcChecksum (
            double mu )  [protected]
```

caculate a checksum from the PDF in order to check if the PDF has changed. This is mandatory since the old LHAPDF code is written in fortran and PDFs may change without any notice.

5.27.3.3 CalcCrossSection()

```
void fastNLOReader::CalcCrossSection ( )
```

Calculate cross section for all active additive and multiplicative contributions

xs = { sum(all active pert. add. contr.) + sum(all other active add. contr.) } $*$ prod(all active multipl. contr.)

5.27.3.4 CalcCrossSectionv20()

```
void fastNLOReader::CalcCrossSectionv20 (
            fastNLOCoeffAddFix * B )  [protected]
```

Cross section calculation in v2.0 format

Test that alpha_s cache is filled with non-zero values for this contribution

Test that PDF cache is filled with non-zero values for this contribution

5.27.3.5 CalcCrossSectionv21()

```
void fastNLOReader::CalcCrossSectionv21 (
            fastNLOCoeffAddFlex * B )  [protected]
```

Cross section calculation for DIS and HHC tables in v2.1 format

5.27.3.6 CalcMu()

```
double fastNLOReader::CalcMu (
            fastNLO::EMuX kMuX,
            double scale1,
            double scale2,
            double scalefactor )  [protected]
```

Calculate the scales with the defined function and the corresponding prefactor.

5.27.3.7 CalcNewPDFChecksum()

```
double fastNLOReader::CalcNewPDFChecksum ( )  [protected]
```

calculate a PDF checksum to decide, whether PDF cache has to be refilled

5.27.3.8 CalcReferenceCrossSection()

```
void fastNLOReader::CalcReferenceCrossSection ( )  [protected]
```

Initialize the internal arrays for the reference cross sections with the information from the FastNLO file

5.27.3.9 FillAlphasCache()

```
void fastNLOReader::FillAlphasCache (
            bool lForce = false )
```

prepare for recalculation of cross section with new alpha_s value.

Fill the internal alpha_s cache. This is usally called automatically. Only if you make use of ReFillCache==false options, you have to take care of this filling by yourself.

5.27.3.10 FillAlphasCacheInBlockBv20()

```
void fastNLOReader::FillAlphasCacheInBlockBv20 (
            fastNLOCoeffAddFix * B )  [protected]
```

Internal method for filling alpha_s cache

5.27.3.11 FillAlphasCacheInBlockBv21()

```
void fastNLOReader::FillAlphasCacheInBlockBv21 (
            fastNLOCoeffAddFlex * B )  [protected]
```

Internal method for filling alpha_s cache

5.27.3.12 FillBlockBPDFLCsDISv20()

```
void fastNLOReader::FillBlockBPDFLCsDISv20 (
            fastNLOCoeffAddFix * B )  [protected]
```

Fill member variables in fastNLOCoeffAddFix with PDFCache

5.27.3.13 FillBlockBPDFLCsDISv21()

```
void fastNLOReader::FillBlockBPDFLCsDISv21 (
            fastNLOCoeffAddFlex * B,
            fastNLOCoeffAddFlex * B0 = NULL )  [protected]
```

Fill member variables in fastNLOCoeffAddFlex with PDFCache

5.27.3.14 FillBlockBPDFLCsHHCv20()

```
void fastNLOReader::FillBlockBPDFLCsHHCv20 (
            fastNLOCoeffAddFix * B )  [protected]
```

Fill member variables in fastNLOCoeffAddFix with PDFCache

5.27.3.15 FillBlockBPDFLCsHHCv21()

```
void fastNLOReader::FillBlockBPDFLCsHHCv21 (
            fastNLOCoeffAddFlex * B )  [protected]
```

Fill member variables in fastNLOCoeffAddFlex with PDFCache The calculation is improved, if the factorization scale is calculated from only one scale variable (i.e. kScale1 or kScale2)

5.27.3.16 FillPDFCache()

```
void fastNLOReader::FillPDFCache (
            double chksum = 0.,
            bool lForce = false )
```

Prepare for recalculation of cross section with 'new'/updated pdf.

Fill the internal pdf cache. This function has to be called by the user, since the pdf parameters and evolutions are calculated externally.

5.27.3.17 GetCrossSection2Dim()

```
vector< vector< double > > fastNLOReader::GetCrossSection2Dim ( )
```

Get cross section as 2-dimensional vector according to defined binning

5.27.3.18 GetNevt()

```
double fastNLOReader::GetNevt (
            const fastNLO::ESMOrder eOrder ) const
```

Get label of scale iScale for order eOrder of the fixed order calculation.

5.27.3.19 GetScaleDescription()

```
string fastNLOReader::GetScaleDescription (
            const fastNLO::ESMOrder eOrder,
            int iScale = 0 ) const
```

Get label of scale iScale for order eOrder of the fixed order calculation.

5.27.3.20 GetScaleUncertainty()

```
XsUncertainty fastNLOReader::GetScaleUncertainty (
            const fastNLO::EScaleUncertaintyStyle eScaleUnc,
            bool lNorm )
```

Cross section and absolute uncertainties

Divide by cross section != 0 to give relative uncertainties

5.27.3.21 InitScalevariation()

```
void fastNLOReader::InitScalevariation ( )
```

Initialize to scale factors of (MuR,MuF) = (1,1)

5.27.3.22 Print()

```
void fastNLOReader::Print (
            int iprint ) const  [virtual]
```

Print basic info about fastNLO table and its contributions.

this function is inherited from fastNLOTable.

Reimplemented from fastNLOTable.

5.27.3.23 PrintContributionSummary()

```
void fastNLOReader::PrintContributionSummary (
            int iprint ) const
```

this function is inherited from fastNLOTable.

5.27.3.24 PrintCrossSections()

```
void fastNLOReader::PrintCrossSections ( ) const
```

Print cross sections (optimized for double-differential tables)

Print Cross sections in NLO, k-factors and Reference table cross sections

5.27.3.25 PrintCrossSectionsWithReference()

```
void fastNLOReader::PrintCrossSectionsWithReference ( )
```

Print Cross sections in NLO, k-factors and Reference table cross sections

Please mention, that the reference cross section can be easily deviating more than 20% (scales, pdfs, alpha_s, etc...). This does not mean that the table is wrong!

5.27.3.26 RescaleCrossSectionUnits()

```
double fastNLOReader::RescaleCrossSectionUnits (
            double binsize,
            int xunits )
```

This method rescales the stored cross section units according to the chosen Ipublunits and settings for [kAbsolute↩ Units | kPublicationUNits].

5.27.3.27 SetCoefficientUsageDefault()

```
void fastNLOReader::SetCoefficientUsageDefault ( )
```

Switch on LO and NLO contributions, deactivate other contributions.

Switch on LO, NLO, and NNLO contribution. Deactivate all other contributions

5.27.3.28 SetContributionON()

```
bool fastNLOReader::SetContributionON (
            fastNLO::ESMCalculation eCalc,
            unsigned int Id,
            bool SetOn = true )
```

Set contribution Id On/Off. Check for Id of a particular contribution with ContrId(...) or use ActivateContribution(...).

Enable or disable a contribution to be considered in the cross section calculation

- Use SetOn=true, to switch contribution ON,
- Use SetOn=false, to switch a contribution off

Each contribution is identified by an ESMCalculation and by a universal Id. For all available contributions in your table, call PrintContributionSummary().

The LO contribution can be e.g. addressed by (eCalc=fastNLO::kFixedOrder, Id=0); The NLO contribution can be e.g. addressed by (eCalc=fastNLO::kFixedOrder, Id=1);

If an additional additive contribution is switched on, then the PDFCache and AlphasCache are refilled.

5.27.3.29 SetExternalConstantForMuF()

```
void fastNLOReader::SetExternalConstantForMuF (
            double MuF )
```

Set value for mu_f if mu_f is chosen to be a constant value (i.e. m_t, or m_Z)

Set value for mu_r if mu_r is chosen to be constant EScaleFunctionalForm == kConst

5.27.3.30 SetExternalConstantForMuR()

```
void fastNLOReader::SetExternalConstantForMuR (
            double MuR )
```

Set value for mu_r if mu_r is chosen to be a constant value (i.e. m_t, or m_Z)

Set value for mu_r if mu_r is chosen to be constant EScaleFunctionalForm == kConst

5.27.3.31 SetFunctionalForm()

```
void fastNLOReader::SetFunctionalForm (
             fastNLO::EScaleFunctionalForm func,
             fastNLO::EMuX kMuX )
```

Set functional form of MuX.

For MuVar tables this method sets the functional form of the renormalization or the factorization scale. func: Choose a pre-defined function kMuX: is it for mu_r or for mu_f ?

5.27.3.32 SetScaleFactorsMuRMuF()

```
bool fastNLOReader::SetScaleFactorsMuRMuF (
             double xmur,
             double xmuf )
```

Set scale factors for MuR and MuF.

Set renormalization and factorization scale factors simultaneously for scale variations in all v2 tables. You have to ReFill your cache! This is done automatically, but if you want to do it by yourself set ReFillCache = false.

The function aborts the whole program if non-sensical scale factors $<$ 1.E-6 are requested. The function returns true if the requested scale factors can be used with the available table:

If it is NOT a flexibleScaleTable and there is no NLO scalevar table for xmuf, xmur and xmuf are unchanged, a warning is printed and the function returns false! If threshold corrections are selected, then

- only symmetric scale variations, i.e. xmur / xmuf = 1., are allowed,
- the scale variations for xmuf must be stored in IDENTICAL order for the NLO and the threshold corrections (there is only one fScalevar!) If either is not the case, xmur and xmuf are unchanged, a warning is printed and the function returns false!

5.27.3.33 SetScaleVariation()

```
bool fastNLOReader::SetScaleVariation (
             int scalevar )  [protected]
```

Choose the MuF scale variation table.

NEVER call this setter directly, only via the method SetScaleFactorsMuRMuF!

Set the scale variation table to correspond to the selected MuF factor if possible. Usually, v2.0 tables are stored for multiple MuF settings like factors of 0.5, 1.0 and 2.0 times the nominal scale, e.g. scalevar -> scalefactor '0' -> 1.0 '1' -> 0.5 '2' -> 2.0 If tables for multiple MuF factors are present, then they MUST correspond to exactly the same factors in the SAME order for all such contrbutions, e.g. NLO plus 2-loop threshold corrections!

This method returns true if the chosen 'scalevar' table exists for all selected

contributions with extra scale tables.

5.27.3.34 TestAlphas()

```
bool fastNLOReader::TestAlphas ( )
```

Test if EvolvaAlphas returns a reasonable value.

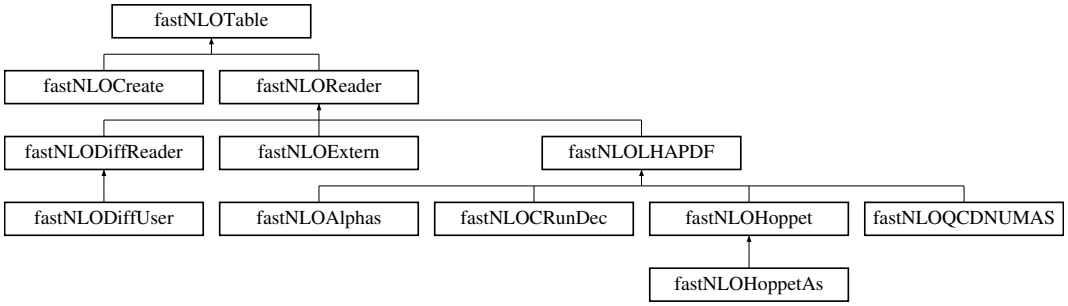
Test if the alpha_s evolution provided by the user yields realistic results.

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

- fastnlotoolkit/include/fastnlotk/fastNLOReader.h
- fastnlotoolkit/fastNLOReader.cc

5.28 fastNLOTable Class Reference

Inheritance diagram for fastNLOTable:



Public Member Functions

- **fastNLOTable** (std::string filename)
- fastNLOTable (const fastNLOTable &)
- virtual void ReadTable ()
- virtual void WriteTable ()
- virtual void **WriteTable** (std::string filename)
- bool **IsCompatible** (const fastNLOTable &other) const
- bool **IsCompatibleScenario** (const fastNLOTable &other) const
- bool **IsCatenable** (const fastNLOTable &other) const
- bool **IsCatenableScenario** (const fastNLOTable &other) const
- void PrintHeader (int iprint) const

  Print header variables (BlockA1) to screen.
- bool IsCompatibleHeader (const fastNLOTable &other) const

  Compare header with header of another table.
- bool IsCatenableHeader (const fastNLOTable &other) const

  Compare header with header of another table.

- std::string **GetFilename** () const
- void **SetFilename** (std::string name)
- int **GetItabversion** () const
- void **SetItabversion** (int version)
- std::string **GetScenName** () const
- void **SetScenName** (std::string name)
- int **GetNmult** () const
- int **GetNcontrib** () const
- int **GetNdata** () const
- int **GetOutputPrecision** () const
- void **SetOutputPrecision** (int precision)
- unsigned int GetNumDiffBin () const

  Get dimensionality of calculation: single-, double-, or triple-differential.
- double GetObsBinLoBound (unsigned int iObs, unsigned int iDim) const

  Getters for linear array of observable bins "ObsBin" running from 0->(NObsBin-1)
- double GetObsBinUpBound (unsigned int iObs, unsigned int iDim) const

  Return upper bin bound for obs. bin iObs in dim. iDim.
- std::vector< double > GetObsBinsLoBounds (unsigned int iDim) const

  Return std::vector of lower bin bounds in dim. iDim for all obs. bins.
- std::vector< double > GetObsBinsUpBounds (unsigned int iDim) const

  Return std::vector of upper bin bounds in dim. iDim for all obs. bins.
- double GetObsBinsLoBoundsMin (unsigned int iDim) const

  Return minimum value of all lower bin bounds for dim. iDim.
- double GetObsBinsUpBoundsMax (unsigned int iDim) const

  Return maximum value of all upper bin bounds for dim. iDim.
- std::vector< std::pair< double, double > > GetObsBinsBounds (unsigned int iDim) const

  Return std::vector of pairs with lower and upper bin bounds in dim. iDim for all obs. bins.
- int GetObsBinNumber (const std::vector< double > &vobs) const

  Return observable bin no. for std::vector of values obs0=var0,obs1=var1,...; -1 if outside range.
- int GetObsBinNumber (double var0) const

  Return observable bin no. for obs0=var0 in 1D binning; -1 if outside range.
- int GetObsBinNumber (double var0, double var1) const

  Return observable bin no. for obs0=var0,obs1=var1 in 2D binning; -1 if outside range.
- int GetObsBinNumber (double var0, double var1, double var2) const

  Return observable bin no. for obs0=var0,obs1=var1,obs2=var2 in 3D binning; -1 if outside range.
- std::vector< std::pair< double, double > > GetDim0BinBounds () const

  Getters for multidimensional binning, here called DimBins.
- std::vector< std::pair< double, double > > GetDim1BinBounds (unsigned int iDim0Bin) const

  Return std::vector of pairs with unique bin bounds of 2nd dim. for 'iDim0Bin' of 1st dim.
- std::vector< std::pair< double, double > > GetDim2BinBounds (unsigned int iDim0Bin, unsigned int iDim1Bin) const

  Return std::vector of pairs with unique bin bounds of 3rd dim. for 'iDim0Bin' and 'iDim1Bin' of 1st two dim.
- std::vector< std::pair< double, double > > GetObsBinDimBounds (unsigned int iObs) const

  Return std::vector of pairs with lower and upper bin bounds for all dimensions for a given obs. bin.
- std::pair< double, double > GetObsBinDimBounds (unsigned int iObs, unsigned int iDim) const

  Return pair with lower and upper bin bounds for given obs. bin and dim. iDim.
- unsigned int GetIDim0Bin (unsigned int iObs) const

  Return bin no. in 1st dim. for obs. bin iObs.
- unsigned int GetIDim1Bin (unsigned int iObs) const

  Return bin no. in 2nd dim. for obs. bin iObs.
- unsigned int GetIDim2Bin (unsigned int iObs) const

Return bin no. in 3rd dim. for obs. bin iObs.

- unsigned int GetNDim0Bins () const

  Return no. of bins in 1st dimension.
- unsigned int GetNDim1Bins (unsigned int iDim0Bin) const

  Return no. of bins in 2nd dimension for given bin in 1st dim.
- unsigned int GetNDim2Bins (unsigned int iDim0Bin, unsigned int iDim1Bin) const

  Return no. of bins in 3rd dimension for given bins in 1st and 2nd dim.
- int GetODim0Bin (double var0) const

  Return bin no. in 1st dim. for obs0=var0; -1 if outside range.
- int GetODim1Bin (double var0, double var1) const

  Return bin no. in 2nd dim. for obs0=var0,obs1=var1; -1 if outside range.
- int GetODim2Bin (double var0, double var1, double var2) const

  Return bin no. in 3rd dim. for obs0=var0,obs1=var1,obs2=var2; -1 if outside range.
- int GetIDiffBin (int bin) const

  Get if dimension is 'truly differential' or bin-integrated (divided by bin width or not)
- double GetBinSize (int bin) const

  Get BinSize for bin = BinSizeDim1 < ∗ BinSizeDim2 >
- std::vector< std::string > GetDimLabels () const

  Get vector of dimensions labels.
- std::string GetDimLabel (int iDim) const

  Get dimension label for dimension iDim.
- int GetINormFlag () const
- bool IsNorm () const

  Get normalization logical (def=false)
- std::string GetDenomTable () const

  Get filename of normalization table for INormFlag<0.
- std::vector< std::string > GetScDescr () const

  get/set scenario description
- void **SetScDescr** (std::vector< std::string > ScDescr)
- int GetIpublunits () const

  get/set cross section units of published results (pb = 12, fb = 15, ...)
- void **SetIpublunits** (int unit)
- double GetEcms () const

  get/set center-of-mass energy in units of GeV
- void **SetEcms** (double E)
- int GetLoOrder () const

  get/set power of alpha_s for LO process
- void **SetLoOrder** (int LOOrd)
- unsigned int GetNObsBin () const

  get/set no. of observable bins
- void **SetNObsBin** (int NObs)
- std::vector< std::vector< std::pair< double, double > > > GetBins () const

  get/set Bin vector
- void **SetBins** (std::vector< std::vector< std::pair< double, double > > >)
- std::vector< double > GetBinSize () const

  get/set BinSize vector
- void **SetBinSize** (std::vector< double >)
- void **EraseBinFromTable** (unsigned int iObsIdx)
- template<typename T >
  void **EraseBin** (std::vector< T > &v, unsigned int idx)
- void **MultiplyBinInTable** (unsigned int iObsIdx, double fact)
- void **MultiplyBinSize** (unsigned int iObsIdx, double fact)

- template<typename T >
  void **MultiplyBin** (std::vector< T > &v, unsigned int idx, double fact)
- void **CatBinToTable** (const fastNLOTable &other, unsigned int iObsIdx, unsigned int table_count)
- void **CatBin** (const fastNLOTable &other, unsigned int iObsIdx, unsigned int table_count)
- std::string GetRivetId () const
- std::string GetXSDescr () const

  Get cross section from analysis description.
- void SetDimLabel (std::string label, unsigned int iDim, bool IsDiff=true)
- void **SetNumDiffBin** (int iDiff)
- void PrintTableInfo (const int iprint=0) const

  Print basic info about fastNLO table and its contributions.
- void **PrintContributionSummary** (int iprint) const
- void PrintFastNLOTableConstants (const int iprint=0) const

  Print (technical) constants of fastNLO table (use iprint) for level of details.
- void **PrintScenario** (int iprint) const
- virtual void **Print** (int iprint) const
- void MergeTable (const fastNLOTable &rhs, fastNLO::EMerge option=fastNLO::kMerge)

  'merge'
- void MergeTables (const std::vector< fastNLOTable *> &tables, fastNLO::EMerge option=fastNLO::kMerge, double cutRMS=0)

  'merge' (also supports 'median' and 'mean')
- void AddTable (const fastNLOTable &rhs, fastNLO::EMerge option=fastNLO::kMerge)

  'merge'
- void DeleteAllCoeffTable ()

  Handle coefficient tables.
- int **CreateCoeffTable** (int no, fastNLOCoeffBase *newcoeff)
- void **CatenateTable** (const fastNLOTable &other)
- fastNLOCoeffBase * **GetCoeffTable** (int no) const
- fastNLOCoeffData * GetDataTable () const

  Returns pointer to data table if available, else returns NULL pointer.
- fastNLOCoeffAddBase * GetReferenceTable (fastNLO::ESMOrder eOrder) const

  Returns pointer to reference table if available, else returns NULL pointer.
- template<typename T >
  void **EraseBin** (vector< T > &v, unsigned int idx)
- template<typename T >
  void **MultiplyBin** (vector< T > &v, unsigned int idx, double fact)

Protected Member Functions

- void PrintWelcomeMessage ()

  Say hello to fastNLO user.
- std::ostream * OpenFileWrite (bool compress=false)

  open std::ofstream for writing tables to ffilename
- std::istream * OpenFileRead ()

  open std::ifstream for reading table
- void WriteHeader (std::ostream &table)

  write (or cout) hader using std::ostream
- int ReadHeader (std::istream &table)

  read header of table (BlockA1)
- void CloseFileWrite (std::ostream &table)
- void CloseFileRead (std::istream &table)
- void **WriteScenario** (std::ostream &table)
- void **ReadScenario** (std::istream &table)
- void ReadCoeffTables (std::istream &table, int nCoeff)
- fastNLOCoeffBase * **ReadRestOfCoeffTable** (const fastNLOCoeffBase &cB, std::istream &table)

Protected Attributes

- std::string **ffilename**
- int **fPrecision**
- int **ItabVersion**
- std::string **ScenName**
- PrimalScream **logger**
- std::vector< fastNLOCoeffBase *> **fCoeff**
- double **Ecms**
- int **ILOord**
- int **IpublUnits**
- std::vector< std::string > **ScDescript**
- unsigned int **NObsBin**
- unsigned int **NDim**
- std::vector< std::string > **DimLabel**
- std::vector< int > **IDiffBin**
- std::vector< std::vector< std::pair< double, double > > > **Bin**
- std::vector< double > **BinSize**
- int **INormFlag**
- std::string **DenomTable**
- std::vector< int > **IDivLoPointer**
- std::vector< int > **IDivUpPointer**

Static Protected Attributes

- static bool **fWelcomeOnce** = false

5.28.1 Constructor & Destructor Documentation

5.28.1.1 fastNLOTable()

```
fastNLOTable::fastNLOTable (
            const fastNLOTable & other )
```

Copy constructor

5.28.2 Member Function Documentation

5.28.2.1 CloseFileRead()

```
void fastNLOTable::CloseFileRead (
            std::istream & table )  [protected]
```

Close file-stream

5.28.2.2 CloseFileWrite()

```
void fastNLOTable::CloseFileWrite (
            std::ostream & table )  [protected]
```

close stream and delete object;

5.28.2.3 GetDim0BinBounds()

```
vector< pair< double, double > > fastNLOTable::GetDim0BinBounds ( ) const
```

Getters for multidimensional binning, here called Dim*Bins.*

Return std::vector of pairs with unique bin bounds of 1st dim.

5.28.2.4 GetIDiffBin()

```
int fastNLOTable::GetIDiffBin (
            int bin ) const  [inline]
```

Get if dimension is 'truly differential' or bin-integrated (divided by bin width or not)

Some more info getters with respect to observable dimensions

5.28.2.5 GetIDim0Bin()

```
unsigned int fastNLOTable::GetIDim0Bin (
            unsigned int iObs ) const
```

Return bin no. in 1st dim. for obs. bin iObs.

Returns bin number in first dimension Valid for up to triple differential binnings

5.28.2.6 GetIDim1Bin()

```
unsigned int fastNLOTable::GetIDim1Bin (
            unsigned int iObs ) const
```

Return bin no. in 2nd dim. for obs. bin iObs.

Returns bin number in second dimension Valid for up to triple differential binnings

5.28.2.7 GetIDim2Bin()

```
unsigned int fastNLOTable::GetIDim2Bin (
            unsigned int iObs ) const
```

Return bin no. in 3rd dim. for obs. bin iObs.

Returns bin number in third dimension Valid for up to triple differential binnings

5.28.2.8 GetINormFlag()

```
int fastNLOTable::GetINormFlag ( ) const  [inline]
```

Some info getters with respect to normalization

Get normalization flag: def=0 -> no norm. 1, 2, 3,... -> normalize to slice in NDim of same table -1,-2,-3,... -> normalize to slice in NDim of other table

5.28.2.9 GetNDim0Bins()

```
unsigned int fastNLOTable::GetNDim0Bins ( ) const
```

Return no. of bins in 1st dimension.

Returns number of bins in first dimension Valid for up to triple differential binnings

5.28.2.10 GetNDim1Bins()

```
unsigned int fastNLOTable::GetNDim1Bins (
            unsigned int iDim0Bin ) const
```

Return no. of bins in 2nd dimension for given bin in 1st dim.

Returns number of bins in second dimension for iDim0Bin in first dimension Valid for up to triple differential binnings

5.28.2.11 GetNDim2Bins()

```
unsigned int fastNLOTable::GetNDim2Bins (
            unsigned int iDim0Bin,
            unsigned int iDim1Bin ) const
```

Return no. of bins in 3rd dimension for given bins in 1st and 2nd dim.

Returns number of bins in third dimension for iDim0Bin in first and iDim1Bin in second dimension Valid for up to triple differential binnings

5.28.2.12 GetNumDiffBin()

```
unsigned int fastNLOTable::GetNumDiffBin ( ) const  [inline]
```

Get dimensionality of calculation: single-, double-, or triple-differential.

Getters for binning structure

5.28.2.13 GetObsBinLoBound()

```
double fastNLOTable::GetObsBinLoBound (
            unsigned int iObs,
            unsigned int iDim ) const
```

Getters for linear array of observable bins "ObsBin" running from 0->(NObsBin-1)

Return lower bin bound for obs. bin iObs in dim. iDim

5.28.2.14 GetObsBinNumber() [1/3]

```
int fastNLOTable::GetObsBinNumber (
            double var0 ) const
```

Return observable bin no. for obs0=var0 in 1D binning; -1 if outside range.

Returns first matching observable bin number for one observation (assumes none or exactly one matching bin!)
Returns -1 if outside range

5.28.2.15 GetObsBinNumber() [2/3]

```
int fastNLOTable::GetObsBinNumber (
            double var0,
            double var1 ) const
```

Return observable bin no. for obs0=var0,obs1=var1 in 2D binning; -1 if outside range.

Returns first matching observable bin number for two observations (assumes none or exactly one matching bin!)
Returns -1 if outside range

5.28.2.16 GetObsBinNumber() [3/3]

```
int fastNLOTable::GetObsBinNumber (
            double var0,
            double var1,
            double var2 ) const
```

Return observable bin no. for obs0=var0,obs1=var1,obs2=var2 in 3D binning; -1 if outside range.

Returns first matching observable bin number for three observations (assumes none or exactly one matching bin!)
Returns -1 if outside range

5.28.2.17 GetObsBinsLoBoundsMin()

```
double fastNLOTable::GetObsBinsLoBoundsMin (
            unsigned int iDim ) const
```

Return minimum value of all lower bin bounds for dim. iDim.

Get lowest bin edge of all observable bins for dimension 'iDim'

5.28.2.18 GetObsBinsUpBoundsMax()

```
double fastNLOTable::GetObsBinsUpBoundsMax (
            unsigned int iDim ) const
```

Return maximum value of all upper bin bounds for dim. iDim.

Get uppermost bin edge of all observable bins for dimension 'iDim'

5.28.2.19 GetRivetId()

```
string fastNLOTable::GetRivetId ( ) const
```

??? Get Rivet ID of analysis

5.28.2.20 GetScDescr()

```
vector< string > fastNLOTable::GetScDescr ( ) const
```

get/set scenario description

Some info getters & setters for table modifications

5.28.2.21 MergeTable()

```
void fastNLOTable::MergeTable (
            const fastNLOTable & rhs,
            fastNLO::EMerge option = fastNLO::kMerge )
```

'merge'

Other useful functions

< Merge another table with the current one. Use the option moption in order to specify the weighting procedure.

- Default option uses 'normalisation' constant Nevt (usually called 'merge')
- Other weighting options are available.
- Weighting which consider weights for individual bins and subprocesses can be chosen.
- Tables can be 'appended', i.e. the sum of the cross sections is calculated.

5.28.2.22 MergeTables()

```
void fastNLOTable::MergeTables (
            const std::vector< fastNLOTable *> & tables,
            fastNLO::EMerge option = fastNLO::kMerge,
            double cutRMS = 0 )
```

'merge' (also supports 'median' and 'mean')

< Merge all other tables with the current one. Warning: data or multiplicative contributions might get lost Warning: Function may require lots of memory, because all contributions are kept in memory.

5.28.2.23 OpenFileRead()

```
std::istream * fastNLOTable::OpenFileRead ( )  [protected]
```

open std::ifstream for reading table

Open file-stream for reading table

5.28.2.24 OpenFileWrite()

```
std::ostream * fastNLOTable::OpenFileWrite (
            bool compress = false )  [protected]
```

open std::ofstream for writing tables to ffilename

open ostream for writing tables do overwrite existing table

5.28.2.25 PrintTableInfo()

```
void fastNLOTable::PrintTableInfo (
            const int iprint = 0 ) const
```

Print basic info about fastNLO table and its contributions.

Info print out functionality

5.28.2.26 ReadCoeffTables()

```
void fastNLOTable::ReadCoeffTables (
            std::istream & table,
            int nCoeff )  [protected]
```

< read nCoeff Coefficient tables (additive, multiplicative and data)

5.28.2.27 ReadHeader()

```
int fastNLOTable::ReadHeader (
            std::istream & table )  [protected]
```

read header of table (BlockA1)

< Read table header (formely named BlockA1 and BlockA2) return number of contributions to follow

5.28.2.28 ReadTable()

```
void fastNLOTable::ReadTable ( )  [virtual]
```

Read file

5.28.2.29 SetDimLabel()

```
void fastNLOTable::SetDimLabel (
            std::string label,
            unsigned int iDim,
            bool IsDiff = true )
```

Set label for dimension In this method, we also set IDiffBin. The IDiffBin flag defines, if this dimension is 0 (not differential, two bin borders required), 1 (pointwise differential, one value required), (not yet completely implemented) 2 (binwise differential, two bin borders required) In case 2 the cross section is divided by the corresponding bin width in this dimension.

5.28.2.30 WriteTable()

```
void fastNLOTable::WriteTable ( )  [virtual]
```

< WriteTable(). writes the full FastNLO table to the previously defined ffilename on disk. Write fastNLO table to file 'ffilename' (member)

Reimplemented in fastNLOCreate.

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

- fastnlotoolkit/include/fastnlotk/fastNLOTable.h
- fastnlotoolkit/fastNLOTable.cc

5.29 fnloEvent Class Reference

Public Member Functions

- void **ResetButX** ()
- void **Reset** ()
- const double & X1 () const

  get x-value of first hadron
- const double & X2 () const

  get x-value of second hadron
- const int & p () const

  get x-value of second hadron
- void SetX (double x)

  set x-value of first hadron (if e.g. DIS)
- void SetX1 (double x)

  setx-value of first hadron
- void SetX2 (double x)

  set x-value of second hadron
- void SetProcessId (int n)

  set identifier of specific subprocess (0<n<NSubproc), according to the corresponding PDF linear combination
- void SetEventCounter (long long int n)
- void SetWeight (double w)

  if not a flexible-scale table
- void SetSigma (double s)

  weight to calculate cross section (i.e. already multiplied by PDF,alpha_s).
- void AddSigma (double s)
- void SetWeight_MuIndependent (double w)

  flexible scale table:
- void SetWeight_log_mur (double w)

  set weight w, which will contribute with log_e(mur^2)∗w
- void SetWeight_log_muf (double w)

  set weight w, which will contribute with log_e(muf^2)∗w
- void SetWeight_log_murr (double w)

  set weight w, which will contribute with log^2_e(mur^2)∗w
- void SetWeight_log_muff (double w)

  set weight w, which will contribute with log^2_e(muf^2)∗w
- void SetWeight_log_murf (double w)

  set weight w, which will contribute with log_e(mur^2)∗log_e(muf^2)∗w
- void AddWeight_MuIndependent (double w)

  weights must be mutliplied with dummypdf (1/x)
- void AddWeight_log_mur (double w)

  set weight w, which will contribute with log_e(mur^2)∗w
- void AddWeight_log_muf (double w)

  set weight w, which will contribute with log_e(muf^2)∗w
- void AddWeight_log_murr (double w)

  set weight w, which will contribute with log^2_e(mur^2)∗w
- void AddWeight_log_muff (double w)

  set weight w, which will contribute with log^2_e(muf^2)∗w
- void AddWeight_log_murf (double w)

  set weight w, which will contribute with log_e(mur^2)∗log_e(muf^2)∗w

Friends

- class fastNLOCreate

5.29.1 Member Function Documentation

5.29.1.1 AddSigma()

```
void fnloEvent::AddSigma (
            double s )  [inline]
```

sigma

5.29.1.2 SetEventCounter()

```
void fnloEvent::SetEventCounter (
            long long int n )  [inline]
```

Set event counter

5.29.1.3 SetWeight()

```
void fnloEvent::SetWeight (
            double w )  [inline]
```

if not a flexible-scale table

weights must be mutliplied with dummypdf (1/x)

5.29.1.4 SetWeight_MuIndependent()

```
void fnloEvent::SetWeight_MuIndependent (
            double w )  [inline]
```

flexible scale table:

weights must be mutliplied with dummypdf (1/x)

5.29.2 Friends And Related Function Documentation

5.29.2.1 fastNLOCreate

```
friend class fastNLOCreate  [friend]
```

useful class to keep all process related variables. e.g x-values, weights, process identifiers, etc...

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

- fastnlotoolkit/include/fastnlotk/fastNLOEvent.h

5.30 fnloScenario Class Reference

Public Member Functions

- void SetObservableDimI (double o, int iDim)

  Set observable of dimension iDim (e.g. in case of multidimensional measurements)
- void SetObservable0 (double o)

  Set observable for '0th' dimension for single-differential calculation.
- void SetObservable1 (double o)

  Set observable for '1st' dimension for single and double-differential calculations.
- void SetObservable2 (double o)

  Set observable for '2nd' dimension for single/double/triple differential calculations.
- void SetObsBin (int iBin)
- void SetObsScale1 (double mu)

  flexible scale table:
- void SetObsScale2 (double mu)
- void SetScale (double mu)

  if not a flexible-scale table

Friends

- class fastNLOCreate

5.30.1 Member Function Documentation

5.30.1.1 SetObsBin()

```
void fnloScenario::SetObsBin (
            int iBin )  [inline]
```

[optional] Set ObsBin (e.g. if binning is performed by generator, no other observables are then needed.)

5.30.1.2 SetObsScale1()

```
void fnloScenario::SetObsScale1 (
            double mu )  [inline]
```

flexible scale table:

For flexible-scale tables. Set scale 1 (should be in 'GeV').

5.30.1.3 SetObsScale2()

```
void fnloScenario::SetObsScale2 (
            double mu )  [inline]
```

For flexible-scale tables. Set scale 2

5.30.1.4 SetScale()

```
void fnloScenario::SetScale (
            double mu )  [inline]
```

if not a flexible-scale table

the ren. and fact. scale (not mu^2)

5.30.2 Friends And Related Function Documentation

5.30.2.1 fastNLOCreate

```
friend class fastNLOCreate  [friend]
```

useful class to keep all scenario specific quantities. e.g. observables and scales

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

- fastnlotoolkit/include/fastnlotk/fastNLOEvent.h

5.31 fastNLOCreate::fnloStats Struct Reference

Public Member Functions

- void **PrintStats** () const

Public Attributes

- time_t _time

  structre to keep track of statisics. Just for fun and information.
- long long int **_nProc**
- long long int **_nEvPS**
- double **_nEv**

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

- fastnlotoolkit/include/fastnlotk/fastNLOCreate.h

5.32 fastNLO::GeneratorConstants Struct Reference

Public Member Functions

- std::vector< std::string > GetCodeDescription ()

  Get 'CodeDescription' usable for fastNLO table.

Public Attributes

- std::string Name

  Name and version of generator.
- std::vector< std::string > References

  References for generator. Include additional information here (e.g. 'run-mode' or process).
- int UnitsOfCoefficients

5.32.1 Member Data Documentation

5.32.1.1 Name

`std::string fastNLO::GeneratorConstants::Name`

Name and version of generator.

GeneratorConstants

Collection of generator specific constants. These are:

- name and version of generator
- references for generator
- (additional information about generator may be included in References)

5.32.1.2 UnitsOfCoefficients

```
int fastNLO::GeneratorConstants::UnitsOfCoefficients
```

X section units of coefficients passed to fastNLO (neg. power of 10: pb->12, fb->15)

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

- fastnlotoolkit/include/fastnlotk/fastNLOGeneratorConstants.h

5.33 HoppetInterface Class Reference

Static Public Member Functions

- static void **InitHoppet** (fastNLOReader &)
- static std::vector< double > GetSpl (double, double)
- static std::vector< double > GetXFX (double, double)
- static double **EvolveAlphas** (double)

Static Public Attributes

- static bool **IsInitialized** = false
- static double **fAlphasMz** = PDG_ASMZ
- static double **fMz** = PDG_MZ
- static int **fnFlavor** = -1
- static int **fnLoop** = 1
- static double **QMass** [6] = {PDG_MD, PDG_MU, PDG_MS, PDG_MC, PDG_MB, PDG_MT}

5.33.1 Member Function Documentation

5.33.1.1 GetSpl()

```
std::vector< double > HoppetInterface::GetSpl (
            double x,
            double Q )  [static]
```

Returns the splitting functions

5.33.1.2 GetXFX()

```
std::vector< double > HoppetInterface::GetXFX (
            double x,
            double Q )  [static]
```

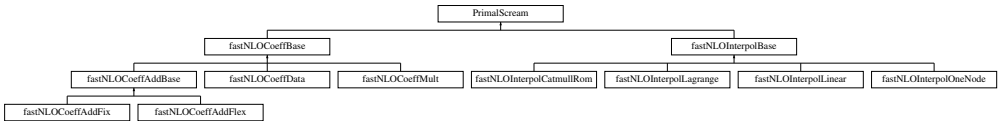
Returns PDF

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

- fastnlotoolkit/include/fastnlotk/HoppetInterface.h
- fastnlotoolkit/HoppetInterface.cc

5.34 PrimalScream Class Reference

Inheritance diagram for PrimalScream:



Public Member Functions

- **PrimalScream** (std::string classname)
- void **SetClassName** (const std::string classname)
- void **SetVerbosity** (say::Verbosity volume)

Public Attributes

- speaker **debug**
- speaker **man**
- speaker **info**
- speaker **warn**
- speaker **error**
- speaker **shout**
- speaker **yell**

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

- fastnlotoolkit/include/fastnlotk/speaker.h
- fastnlotoolkit/speaker.cc

5.35 fastNLO::ProcessConstants Struct Reference

Public Member Functions

- std::vector< std::string > GetProcessDescription ()

Public Attributes

- int LeadingOrder

  Order in alpha_s of leading order process.
- int NPDF

  No. of PDFs involved.
- int NSubProcessesLO

  No. of LO subprocesses.
- int NSubProcessesNLO

  No. of NLO subprocesses.
- int NSubProcessesNNLO

  No. of NNLO subprocesses.
- int IPDFdef1

  Flag 1 to define PDF linear combinations of partonic subprocesses (e.g. hh –> jets: 3)
- int IPDFdef2

  Flag 2 to define PDF linear combinations (dep. on IPDFdef1; for 3 e.g. 1 for jet specific LCs, 121 for generic 11x11 matrix)
- int IPDFdef3LO

  Flag 3 to define PDF LCs at LO (dep. on IPDFdef1, IPDFdef2; for 3, 1 e.g. 6 subprocesses, ignored for IPD↩ Fdef2==121)
- int IPDFdef3NLO

  Flag 3 to define PDF LCs at NLO (dep. on IPDFdef1, IPDFdef2; for 3, 1 e.g. 7 subprocesses, ignored for IPD↩ Fdef2==121)
- int IPDFdef3NNLO

  Flag 3 to define PDF LCs at NNLO (dep. on IPDFdef1, IPDFdef2; for 3, 1 e.g. 7 subprocesses, ignored for IPD↩ Fdef2==121)
- int NPDFDim

  Define internal storage mode for PDF LCs (dep. on NPDF; e.g. for 1: 0 for linear, for 2: 1 for half- or 2 for full-matrix)
- std::vector< std::vector< std::pair< int, int > > > PDFCoeffLO

  PDF Linear combinations for LO calculation (used only if IPDFdef2==0)
- std::vector< std::vector< std::pair< int, int > > > PDFCoeffNLO

  PDF Linear combinations for NLO calculation (used only if IPDFdef2==0)
- std::vector< std::vector< std::pair< int, int > > > PDFCoeffNNLO

  PDF Linear combinations for NNLO calculation (used only if IPDFdef2==0)
- std::vector< std::vector< int > > PDFLiCoInLO

  PDF Linear combinations for LO calculation (used only if IPDFdef2==0) [definition as in steering] (used if PDFCoeffLO is empty)
- std::vector< std::vector< int > > PDFLiCoInNLO

  PDF Linear combinations for NLO calculation (used only if IPDFdef2==0) [definition as in steering].
- std::vector< std::vector< int > > PDFLiCoInNNLO

  PDF Linear combinations for NNLO calculation (used only if IPDFdef2==0) [definition as in steering].
- std::vector< std::pair< int, int > > AsymmetricProcesses

  Specify processes that need to be exchanged in half-matrix notation, when xmin>xmax (only if NPDFDim==1)
- std::string Name

  < More precise description for specific contribution (e.g. LO, pp -> 2 jets; also can add 'run-mode' and further details)
- std::vector< std::string > References

  < References for process (also other plain text lines can be included here)

5.35.1 Member Function Documentation

5.35.1.1 GetProcessDescription()

```
std::vector<std::string > fastNLO::ProcessConstants::GetProcessDescription ( )  [inline]
```

Get 'ContrDescription' usable for fastNLO table

5.35.2 Member Data Documentation

5.35.2.1 LeadingOrder

```
int fastNLO::ProcessConstants::LeadingOrder
```

Order in alpha_s of leading order process.

ProcessConstants

Collection of process specific constants. Please see fastNLO table format definition for a detailed explanation.

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

- fastnlotoolkit/include/fastnlotk/fastNLOGeneratorConstants.h

5.36 read_steer Class Reference

Public Member Functions

- bool **getb** (const std::string &label)
- int **geti** (const std::string &label)
- double **getd** (const std::string &label)
- std::string **gets** (const std::string &label)
- std::vector< bool > **getbf** (const std::string &label)
- std::vector< int > **getif** (const std::string &label)
- std::vector< double > **getdf** (const std::string &label)
- std::vector< std::string > **getsf** (const std::string &label)
- std::vector< std::string > **getsthead** (const std::string &label)
- std::vector< std::vector< int > > **getit** (const std::string &label)
- std::vector< std::vector< double > > **getdt** (const std::string &label)
- std::vector< std::vector< std::string > > **getst** (const std::string &label)
- std::vector< bool > **getbtcol** (const std::string &label, const std::string &col)
- std::vector< int > **getitcol** (const std::string &label, const std::string &col)
- std::vector< double > **getdtcol** (const std::string &label, const std::string &col)
- std::vector< std::string > **getstcol** (const std::string &label, const std::string &col)
- bool **exist** (const std::string &label)
- bool **arrayexist** (const std::string &label)
- bool **arraycontainkey** (const std::string &key, const std::string &label)
- void **arraypushback** (const std::string &value, const std::string &label)
- void **AddLabel** (const std::string &key, const std::string &value)

- template<typename T >
  void **AddLabel** (const std::string &key, T value)
- void **AddArray** (const std::string &key, const std::vector< std::string > &values)
- template<typename T >
  void **AddArray** (const std::string &key, const std::vector< T > &values)
- void **AddTable** (const std::string &key, const std::vector< std::string > &header, const std::vector< std↩
  ::vector< std::string > > &values)
- template<typename T >
  void **AddTable** (const std::string &key, const std::vector< std::string > &header, const std::vector< std↩
  ::vector< T > > &values)
- void **AppendToArray** (const std::string &key, const std::string &entry)
- template<typename T >
  void **AppendToArray** (const std::string &key, const T &entry)
- void AppendToTable (const std::string &key, const std::vector< std::string > &entry)
- template<typename T >
  void **AppendToTable** (const std::string &key, const std::vector< T > &entry)
- int **inits** (std::string filename)
- int **initnmspc** (std::ifstream &strm, std::string filename)
- void **prt** ()
- std::set< std::string > **GetAvailableLabels** () const
- std::set< std::string > **GetAvailableArrrays** () const
- std::set< std::string > **GetAvailableTables** () const

Static Public Member Functions

- static read_steer * **Steering** (std::string steerID=read_steer::stdID)
- static void **destroy** ()
- static void **initnamespace** (std::ifstream &strm, std::string filename, std::string steerID=read_steer::stdID)
- static bool CheckNumber (const std::string &str)
- static bool CheckInt (const std::string &str)
- static int **separatetag** (std::string &vallhs, std::string &valrhs, const std::string &sep)
- static const std::string & **GetDefaultNamespace** ()
- static void **SetDefaultNamespace** (const std::string &nspc)
- static int **readfile** (std::string filename, std::string steerID=read_steer::stdID)
- static void **setVerbosity** (int iVerbosity)
- static int **getVerbosity** ()
- static bool **getbool** (std::string label, std::string steerID=read_steer::stdID)
- static int **getint** (std::string label, std::string steerID=read_steer::stdID)
- static double **getdouble** (std::string label, std::string steerID=read_steer::stdID)
- static std::string **getstring** (std::string label, std::string steerID=read_steer::stdID)
- static std::vector< bool > **getboolarray** (std::string label, std::string steerID=read_steer::stdID)
- static std::vector< int > **getintarray** (std::string label, std::string steerID=read_steer::stdID)
- static std::vector< double > **getdoublearray** (std::string label, std::string steerID=read_steer::stdID)
- static std::vector< std::string > **getstringarray** (std::string label, std::string steerID=read_steer::stdID)
- static std::vector< std::string > **gettableheader** (std::string label, std::string steerID=read_steer::stdID)
- static std::vector< std::vector< int > > **getinttable** (std::string label, std::string steerID=read_steer::stdID)
- static std::vector< std::vector< double > > **getdoubletable** (std::string label, std::string steerID=read_↩
  steer::stdID)
- static std::vector< std::vector< std::string > > **getstringtable** (std::string label, std::string steerID=read_↩
  steer::stdID)
- static std::vector< bool > **getboolcolumn** (std::string label, std::string column, std::string steerID=read_↩
  steer::stdID)
- static std::vector< int > **getintcolumn** (std::string label, std::string column, std::string steerID=read_steer↩
  ::stdID)

- static std::vector< double > **getdoublecolumn** (std::string label, std::string column, std::string steerID=read_steer::stdID)
- static std::vector< std::string > **getstringcolumn** (std::string label, std::string column, std::string steerID=read_steer::stdID)
- static bool **getexist** (const std::string &label, std::string steerID=read_steer::stdID)
- static bool **getarrayexist** (const std::string &label, std::string steerID=read_steer::stdID)
- static bool **getarraycontainkey** (const std::string &key, const std::string &label, std::string steerID=read_steer::stdID)
- static void **arraypushback_steer** (const std::string &value, const std::string &label, std::string steerID=read_steer::stdID)
- template<typename T >
  static void **addvalue** (const std::string &key, const T &val, std::string steerID=read_steer::stdID)
- template<typename T >
  static void **addarray** (const std::string &key, const std::vector< T > &val, std::string steerID=read_steer::stdID)
- template<typename T >
  static void **addtable** (const std::string &key, const std::vector< std::string > &header, const std::vector< std::vector< T > > &values, std::string steerID=read_steer::stdID)
- template<typename T >
  static void **appendtoarray** (const std::string &key, const T &entry, std::string steerID=read_steer::stdID)
- template<typename T >
  static void **appendtotable** (const std::string &key, const std::vector< T > &entry, std::string steerID=read_steer::stdID)
- static void **printall** ()
- static void **print** (std::string steerID=read_steer::stdID)
- static bool parsecommandline (int argc, char **argv)
- static bool **parsecommandline** (std::vector< std::string > argv)

Static Public Attributes

- static int **fVerbosity** = 2

5.36.1 Member Function Documentation

5.36.1.1 AppendToTable()

```
void read_steer::AppendToTable (
            const std::string & key,
            const std::vector< std::string > & entry )
```

append one element to a table

5.36.1.2 CheckInt()

```
bool read_steer::CheckInt (
            const std::string & str )  [static]
```

check if str is an integer number

5.36.1.3 CheckNumber()

```
bool read_steer::CheckNumber (
            const std::string & str )  [static]
```

check if str is a float number

5.36.1.4 parsecommandline()

```
bool read_steer::parsecommandline (
            int argc,
            char ** argv )  [static]
```

use like key=value:namespace

use like namespace:key=value

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

- fastnlotoolkit/include/fastnlotk/read_steer.h
- fastnlotoolkit/read_steer.cc

5.37 fastNLO::ScenarioConstants Struct Reference

Public Member Functions

- void SetDefaults ()

Public Attributes

- std::string ScenarioName

  Name of the scenario.
- std::vector< std::string > ScenarioDescription

  Description of the scenario.
- int PublicationUnits

  Unit of data cross sections (negative power of 10, e.g. 12->pb, 15->fb)
- int DifferentialDimension

  Dimensionality of binning (1: single-differential, 2: double-differential; also decides if SingleDifferentialBinning or DoubleDifferentialBinning is used)
- std::vector< std::string > DimensionLabels

  Labels (symbol and unit) for the measurement dimensions (from outer to inner "loop"), e.g. "|y|" and "p_T [GeV]". This may also help to define the observables to be calculated in an automatized way!
- std::vector< int > DimensionIsDifferential

  Specify for each dimension whether: 0: the cross section is NOT differential, i.e. there are two bin borders (but NO division (normalization) by bin width); 1 : the cross section is point-wise differential, i.e. only one point is given; 2 : the cross section is bin-wise differential, i.e. there are two bin borders and division by bin width.
- bool CalculateBinSize

  Calculate bin width from lower and upper bin boundaries.
- double BinSizeFactor

Possibility to provide additional normalization factor, e.g. of 2 for bins in $|y|$.

- std::vector< double > BinSize

  If 'CalculateBinSize' is 'false' provide table with bin widths 'by hand' for normalization. If the calculation should not be divided by bin width, then use 'DimensionIsDifferential' equal '0', and set 'CalculateBinSize' 'true' for each dimension.
- std::string ScaleDescriptionScale1

  "<pT_1,2>_[GeV]" # This defines the scale to be used (Note: The 1st scale should always be in units of [GeV]!)
- std::string ScaleDescriptionScale2

  "pT_max_[GeV]" # Specify 2nd scale name and unit (ONLY for flexible-scale tables)
- std::vector< double > SingleDifferentialBinning

  Observable binning Use either 'SingleDifferentialBinning' or 'DoubleDifferentialBinning' or 'TripleDifferentialBinning' in accordance with 'DifferentialDimension' above.
- std::vector< std::vector< double > > DoubleDifferentialBinning

  Observable binning.
- std::vector< std::vector< double > > TripleDifferentialBinning

  Observable binning.
- double CenterOfMassEnergy

  Center-of-mass energy in GeV. LHC Next Run II: 13000.
- int PDF1

  PDF of 1st hadron (following PDG convention: proton 2212).
- int PDF2

  PDF of 2nd hadron (following PDG convention: proton 2212).
- std::string OutputFilename

  Filename of fastNLO output table.
- int OutputPrecision

  Number of decimal digits to store in output table (def.=8).
- bool OutputCompression

  If zlib available, gzip output table.
- bool FlexibleScaleTable

  Create table fully flexible in mu_f (larger size, and requires scale independent weights during creation), true, or table with fixed number of mu_f scale factors, def.=false.
- std::vector< double > ScaleVariationFactors

  Factorization scale variations (only needed for fixed-scale tables), List of scale factors must include factor '1', Scale factors will be ordered according to fastNLO convention: (1, min, ... , max). Defaults: {0.5, 1, 2}.
- bool ReadBinningFromSteering

  Specify if binning is read from fScenConst or from warmup.
- bool IgnoreWarmupBinningCheck

  Don't check warmup binning to avoid too many floating precision issues.
- bool ApplyPDFReweighting

  Apply reweighting of pdfs for an optimized interpolation, def.=true.
- bool CheckScaleLimitsAgainstBins

  For warmup-run! Set limits for scale nodes to bin borders, if possible.
- std::string X_Kernel

  Lagrange.
- std::string X_DistanceMeasure

  "sqrtlog10"
- int X_NNodes

  15
- std::string X_NNodeCounting

  "NodesMax","NodesPerBin","NodesPerMagnitude"
- std::string Mu1_Kernel

  Lagrange.

- std::string Mu1_DistanceMeasure

  "loglog025"
- int Mu1_NNodes

  6
- std::string Mu2_Kernel

  Lagrange # Scale2 not used for fixed-scale tables.
- std::string Mu2_DistanceMeasure

  "loglog025"
- int Mu2_NNodes

  6

5.37.1 Member Function Documentation

5.37.1.1 SetDefaults()

```
void fastNLO::ScenarioConstants::SetDefaults ( )  [inline]
```

a safe initialisation

< Lagrange

5.37.2 Member Data Documentation

5.37.2.1 ScenarioName

```
std::string fastNLO::ScenarioConstants::ScenarioName
```

Name of the scenario.

fastNLO Scenario constants Steering parameters for a fastNLO scenario. Contains mostly the binning and description of the specific scenario

5.37.2.2 X_Kernel

```
std::string fastNLO::ScenarioConstants::X_Kernel
```

Lagrange.

Choose fastNLO interpolation kernels and distance measures

- -

Currently implemented interpolation kernels

Catmull

Lagrange

OneNode

Linear

#

Currently implemented distance measures

linear

loglog025 eq. to (log(log(4∗x)))

log10

sqrtlog10 eq. to sqrt(log_10(x))

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

- fastnlotoolkit/include/fastnlotk/fastNLOGeneratorConstants.h

5.38 speaker Class Reference

Public Member Functions

- **speaker** (std::string prefix="", say::Verbosity volume=say::INFO, bool err=false, bool quiet=false)
- **speaker** (const speaker &spk)
- const speaker & **operator=** (const speaker &other)
- std::ostream & **operator[ ]** (const std::string &fct) const
- const speaker & **operator+** (const std::string &fct) const
- const speaker & **prefix** (const std::string &fct) const
- std::ostream & **operator()** (const std::string &fct) const
- template<typename T >
  std::ostream & **operator**<< (const T &arg) const
- std::ostream & **operator**>> (const std::string &arg) const
- std::ostream & **print** (const std::string &mes) const
- void **DoSpeak** (bool loud)
- bool **GetSpeak** () const
- void **SetPrefix** (std::string prefix)
- std::string **GetPrefix** () const
- void **SetClassName** (std::string classname)
- std::string **GetClassName** (void) const
- say::Verbosity **GetVolume** (void) const
- void **SetVolume** (say::Verbosity volume)

Static Public Member Functions

- static int **SetGlobalVerbosity** (say::Verbosity volume)
- static void **ErrorToErrStream** (bool ToCerr)

Protected Attributes

- bool **fquiet**
- std::string **pref**
- bool **errs**
- say::Verbosity **fvol**
- unsigned long **fii**
- std::string **cn**

Static Protected Attributes

- static std::ostream * **weg** = NULL
- static unsigned long **ct** = 0
- static bool **fe2cerr** = true
- static say::Verbosity **fverb** = say::INFO
- static std::map< unsigned long, speaker * > * **list** = NULL

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

- fastnlotoolkit/include/fastnlotk/speaker.h
- fastnlotoolkit/speaker.cc

5.39 TriplenfMmu Struct Reference

Public Attributes

- int **nf**
- double **Mth**
- double **muth**

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

- fastnlotoolkit/include/fastnlotk/CRunDec.h

5.40 fastNLO::WarmupConstants Struct Reference

Public Member Functions

- **WarmupConstants** (const ScenarioConstants &scenario)
- void **Init** ()

Public Attributes

- int OrderInAlphasOfWarmupRunWas
- bool **CheckScaleLimitsAgainstBins**
- std::string **ScaleDescriptionScale1**
- std::string **ScaleDescriptionScale2**
- int **DifferentialDimension**
- std::vector< std::string > **DimensionLabels**
- std::vector< int > **DimensionIsDifferential**
- std::vector< std::vector< double > > **Values**
- std::vector< std::string > **headerValues**
- std::vector< std::vector< double > > **Binning**

5.40.1 Member Data Documentation

5.40.1.1 OrderInAlphasOfWarmupRunWas

```
int fastNLO::WarmupConstants::OrderInAlphasOfWarmupRunWas
```

Variables from warmup-run Initalize WarmupConstants with ScenarioConstants for consistency. Furthermore needed for full initialization:

- OrderInAlphasOfWarmupRunWas
- Binning
- Values
- headerValues

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

- fastnlotoolkit/include/fastnlotk/fastNLOGeneratorConstants.h

5.41 fastNLO::WgtStat Struct Reference

Public Member Functions

- void **Erase** ()
- void **Add** (const WgtStat &other)
- WgtStat & **operator+=** (const WgtStat &other)

Public Attributes

- double WgtNevt = 0

  'number of events', i.e. normalisation as suggested by generator (identical to previously use 'Nevt')
- int NumTable = 1

  Number of tables merged into this table.
- unsigned long long WgtNumEv = 0

  number of entries
- double WgtSumW2 = 0

  Sum of all weight∗∗2.
- double SigSumW2 = 0

  Sum of all sigma∗∗2 (i.e. (wgt∗alpha∗pdf)∗∗2)
- double SigSum = 0

  Sum of all sigma (i.e. (wgt∗alpha∗pdf)∗∗2)
- fastNLO::v2d WgtObsSumW2

  sumw2[proc][obs]
- fastNLO::v2d SigObsSumW2

  sumw2[proc][obs]
- fastNLO::v2d SigObsSum

  sum[proc][obs]
- std::vector< std::vector< unsigned long long > > WgtObsNumEv

  Nentries[proc][obs].

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

- fastnlotoolkit/include/fastnlotk/fastNLOCoeffAddBase.h

5.42 XsUncertainty Struct Reference

Public Attributes

- std::vector< double > xs

  Struct for returning vectors with cross section and relative uncertainty.
- std::vector< double > **dxsl**
- std::vector< double > **dxsu**

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

- fastnlotoolkit/include/fastnlotk/fastNLOReader.h

Index