

B- \rightarrow DK, D- \rightarrow KK $\pi\pi$, binned fit

Generated by Doxygen 1.9.0

1 Class Index	1
1.1 Class List	1
2 Class Documentation	3
2.1 Amplitude Class Reference	3
2.1.1 Detailed Description	3
2.1.2 Constructor & Destructor Documentation	3
2.1.2.1 Amplitude()	3
2.1.3 Member Function Documentation	3
2.1.3.1 operator()	3
2.2 Bin Class Reference	4
2.2.1 Detailed Description	4
2.2.2 Constructor & Destructor Documentation	4
2.2.2.1 Bin()	4
2.2.3 Member Function Documentation	4
2.2.3.1 AddEvent()	4
2.2.3.2 GetEvents()	5
2.2.3.3 GetNumberEvents()	5
2.3 BinList Class Reference	5
2.3.1 Detailed Description	6
2.3.2 Constructor & Destructor Documentation	6
2.3.2.1 BinList()	6
2.3.3 Member Function Documentation	6
2.3.3.1 AddEvent() [1/2]	6
2.3.3.2 AddEvent() [2/2]	7
2.3.3.3 GetBin()	7
2.3.3.4 GetEvents()	7
2.3.3.5 LoadTTree()	8
2.3.3.6 NumberBins()	8
2.3.3.7 Predict()	8
2.4 CPPParameters Class Reference	9
2.4.1 Detailed Description	9
2.4.2 Constructor & Destructor Documentation	9
2.4.2.1 CPPParameters()	9
2.4.3 Member Function Documentation	9
2.4.3.1 GetCPPParameters()	10
2.4.3.2 GetError()	10
2.4.3.3 SetError()	10
2.5 DDecayParameters Class Reference	11
2.5.1 Detailed Description	11
2.5.2 Constructor & Destructor Documentation	11
2.5.2.1 DDecayParameters()	11

2.5.3 Member Function Documentation	12
2.5.3.1 Getc()	12
2.5.3.2 GetK()	12
2.5.3.3 GetKbar()	12
2.5.3.4 Gets()	13
2.6 Event Class Reference	13
2.6.1 Detailed Description	13
2.6.2 Constructor & Destructor Documentation	13
2.6.2.1 Event() [1/3]	13
2.6.2.2 Event() [2/3]	13
2.6.2.3 Event() [3/3]	14
2.6.3 Member Function Documentation	14
2.6.3.1 GetEvent()	14
2.6.3.2 GetEventVector()	14
2.6.3.3 GetInvMass2()	14
2.6.3.4 GetInvMass3()	15
2.7 EventList Class Reference	15
2.7.1 Detailed Description	16
2.7.2 Constructor & Destructor Documentation	16
2.7.2.1 EventList()	16
2.7.3 Member Function Documentation	16
2.7.3.1 AddEvent()	16
2.7.3.2 GetEvents()	16
2.7.3.3 NumberEvents()	17
2.8 Fitter Class Reference	17
2.8.1 Detailed Description	17
2.8.2 Constructor & Destructor Documentation	17
2.8.2.1 Fitter()	17
2.8.3 Member Function Documentation	17
2.8.3.1 DoFit()	18
2.9 Generator Class Reference	18
2.9.1 Detailed Description	18
2.9.2 Constructor & Destructor Documentation	18
2.9.2.1 Generator()	18
2.9.3 Member Function Documentation	19
2.9.3.1 Generate()	19
2.10 Likelihood Class Reference	19
2.10.1 Detailed Description	19
2.10.2 Constructor & Destructor Documentation	19
2.10.2.1 Likelihood()	19
2.10.3 Member Function Documentation	20
2.10.3.1 operator()()	20

2.11 PhaseSpaceParameterisation Class Reference	20
2.11.1 Detailed Description	20
2.11.2 Constructor & Destructor Documentation	21
2.11.2.1 PhaseSpaceParameterisation()	21
2.11.3 Member Function Documentation	21
2.11.3.1 NumberOfBins()	21
2.11.3.2 WhichBin()	21
Index	23

Chapter 1

Class Index

1.1 Class List

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

Amplitude	3
Bin	4
BinList	5
CParameters	9
DDecayParameters	11
Event	13
EventList	15
Fitter	17
Generator	18
Likelihood	19
PhaseSpaceParameterisation	20

Chapter 2

Class Documentation

2.1 Amplitude Class Reference

```
#include <Amplitude.h>
```

Public Member Functions

- [Amplitude](#) (const std::string &Damplitude, const std::string &DBARamplitude)
- std::complex< double > [operator\(\)](#) (const std::vector< double > &event, int conj)

2.1.1 Detailed Description

[Amplitude](#) is a class that loads the shared libraries generated by AmpGen and calculates the amplitude of events

2.1.2 Constructor & Destructor Documentation

2.1.2.1 Amplitude()

```
Amplitude::Amplitude (
    const std::string & Damplitude,
    const std::string & DBARamplitude )
```

Constructor that loads the shared library for D and Dbar decay amplitudes

2.1.3 Member Function Documentation

2.1.3.1 operator()()

```
std::complex< double > Amplitude::operator() (
    const std::vector< double > & event,
    int conj )
```

Overload () operator to easily access amplitude

Parameters

<i>event</i>	Vector of four-momenta of event
<i>conj</i>	Set to +1 for D^0 decay and -1 for \overline{D}^0 decay

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

- /data/lhcb/users/tat/KKpipi_Binned_Fit/include/Amplitude.h
- /data/lhcb/users/tat/KKpipi_Binned_Fit/src/Amplitude.cpp

2.2 Bin Class Reference

```
#include <Bin.h>
```

Public Member Functions

- [Bin](#) ()
- void [AddEvent](#) ([Event](#) event, int charge)
- int [GetNumberEvents](#) (int charge) const
- [EventList](#) [GetEvents](#) (int charge)

2.2.1 Detailed Description

[Bin](#) is a class for a bin in phase space

2.2.2 Constructor & Destructor Documentation

2.2.2.1 Bin()

```
Bin::Bin ( )
```

Default constructor that creates an empty [EventList](#)

2.2.3 Member Function Documentation

2.2.3.1 AddEvent()

```
void Bin::AddEvent (
    Event event,
    int charge )
```

Function for adding an event

Parameters

<i>event</i>	Event to add
<i>charge</i>	+1 for B+, -1 for B-

2.2.3.2 GetEvents()

```
EventList Bin::GetEvents (
    int charge )
```

Function for betting [EventList](#) object

Parameters

<i>charge</i>	+1 for B+, -1 for B-
---------------	----------------------

Returns

eventlist [EventList](#) object

2.2.3.3 GetNumberEvents()

```
int Bin::GetNumberEvents (
    int charge ) const
```

Function for getting number of events in this bin

Parameters

<i>charge</i>	+1 for B+, -1 for B-
---------------	----------------------

Returns

Number of events in this bin

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

- /data/lhcb/users/tat/KKpipi_Binned_Fit/include/Bin.h
- /data/lhcb/users/tat/KKpipi_Binned_Fit/src/Bin.cpp

2.3 BinList Class Reference

```
#include <BinList.h>
```

Public Member Functions

- [BinList](#) ([PhaseSpaceParameterisation](#) php)
- void [AddEvent](#) ([Event](#) event, int charge)
- void [AddEvent](#) ([Event](#) event, int charge, int maxevents)
- void [LoadTTree](#) (TTree *tree, int charge)
- int [NumberBins](#) ()
- std::vector< int > [GetEvents](#) (int charge) const
- [Bin](#) [GetBin](#) (int i)
- void [Predict](#) (const [DDecayParameters](#) &ddparameters, const [CPPParameters](#) &cppparameters, std::vector< double > &BplusEvents, std::vector< double > &BminusEvents, int totalBplus, int totalBminus)

2.3.1 Detailed Description

[BinList](#) is a class that contains all the bins in phase space [BinList](#) also loads the input data and puts it in their respective bins

2.3.2 Constructor & Destructor Documentation

2.3.2.1 [BinList](#)()

```
BinList::BinList (
    PhaseSpaceParameterisation php )
```

Constructor that takes a [PhaseSpaceParameterisation](#) object and creates the bins

Parameters

<i>php</i>	A PhaseSpaceParameterisation object that defines the bins in the 5D phase space
------------	---

2.3.3 Member Function Documentation

2.3.3.1 [AddEvent](#)() [1/2]

```
void BinList::AddEvent (
    Event event,
    int charge )
```

Function for adding an event to the correct bin

Parameters

<i>event</i>	Event object to be added to the correct bin
<i>charge</i>	+1 for B+, -1 for B-

2.3.3.2 AddEvent() [2/2]

```
void BinList::AddEvent (
    Event event,
    int charge,
    int maxevents )
```

Function for adding an event to the correct bin, if the number of events in that bin is less than the maximum

Parameters

<i>event</i>	Event object to be added to the correct bin
<i>charge</i>	+1 for B+, -1 for B-
<i>maxEvents</i>	Maximum number of events in each bin

2.3.3.3 GetBin()

```
Bin BinList::GetBin (
    int i )
```

Function for getting [Bin](#) object

Parameters

<i>i</i>	Bin number
----------	----------------------------

Returns

[Bin](#) object

2.3.3.4 GetEvents()

```
std::vector< int > BinList::GetEvents (
    int charge ) const
```

Function for getting the number of events in each bin

Parameters

<i>charge</i>	+1 for B+, -1 for B-
---------------	----------------------

Returns

A vector of the number of events in each bin

2.3.3.5 LoadTTree()

```
void BinList::LoadTTree (
    TTree * tree,
    int charge )
```

Function for loading events from input data into their respective bins

Parameters

<i>tree</i>	A ROOT TTree in the AmpGen format containing all the input data events
<i>charge</i>	+1 for B+, -1 for B-

2.3.3.6 NumberBins()

```
int BinList::NumberBins ( )
```

Function for getting number of bins

2.3.3.7 Predict()

```
void BinList::Predict (
    const DDecayParameters & ddparameters,
    const CPPParameters & cpparameters,
    std::vector< double > & BplusEvents,
    std::vector< double > & BminusEvents,
    int totalBplus,
    int totalBminus )
```

Function for calculating the number of events in each bin, given the D decay parameters and the CP parameters

Parameters

<i>ddparameters</i>	A DDecayParameters object that describes the D meson decay
<i>cpparameters</i>	A CPPParameters object that describes the CP violation in the B meson decay
<i>BplusEvents</i>	Vector of predicted number of B+ events
<i>BminusEvents</i>	Vector of predicted number of B- events
<i>totalBplus</i>	Total number of B+ events
<i>totalBminus</i>	Total number of B- events

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

- /data/lhcb/users/tat/KKpipi_Binned_Fit/include/BinList.h
- /data/lhcb/users/tat/KKpipi_Binned_Fit/src/BinList.cpp

2.4 CParameters Class Reference

```
#include <CParameters.h>
```

Public Member Functions

- [CParameters](#) (double *xplus*, double *xminus*, double *yplus*, double *yminus*)
- void [GetCParameters](#) (double &*xplus*, double &*xminus*, double &*yplus*, double &*yminus*) const
- void [SetError](#) (double *xplus*, double *xminus*, double *yplus*, double *yminus*)
- void [GetError](#) (double &*xplus*, double &*xminus*, double &*yplus*, double &*yminus*) const

2.4.1 Detailed Description

[CParameters](#) is a class that contains the CP parameters *x* and *y*

2.4.2 Constructor & Destructor Documentation

2.4.2.1 CParameters()

```
CParameters::CParameters (
    double xplus,
    double xminus,
    double yplus,
    double yminus )
```

Constructor that takes the CP parameters, *x* and *y*

Parameters

<i>xplus</i>	$r_{\text{Bcos}}(\delta_B + \gamma)$ for Bplus decays
<i>xminus</i>	$r_{\text{Bcos}}(\delta_B - \gamma)$ for Bminus decays
<i>yplus</i>	$r_{\text{Bsin}}(\delta_B + \gamma)$ for Bplus decays
<i>yminus</i>	$r_{\text{Bsin}}(\delta_B - \gamma)$ for Bminus decays

2.4.3 Member Function Documentation

2.4.3.1 GetCPPParameters()

```
void CPPParameters::GetCPPParameters (
    double & xplus,
    double & xminus,
    double & yplus,
    double & yminus ) const
```

Function for getting CP parameters

Parameters

<i>xplus</i>	r_Bcos(delta_B + gamma) for Bplus decays
<i>xminus</i>	r_Bcos(delta_B - gamma) for Bminus decays
<i>yplus</i>	r_Bsin(delta_B + gamma) for Bplus decays
<i>yminus</i>	r_Bsin(delta - gamma) for Bminus decays

2.4.3.2 GetError()

```
void CPPParameters::GetError (
    double & xplus,
    double & xminus,
    double & yplus,
    double & yminus ) const
```

Function for getting CP parameter errors

Parameters

<i>xplus</i>	xplus error
<i>xminus</i>	xminus error
<i>yplus</i>	yplus error
<i>yminus</i>	yminus error

2.4.3.3 SetError()

```
void CPPParameters::SetError (
    double xplus,
    double xminus,
    double yplus,
    double yminus )
```

Function for setting CP parameter errors

Parameters

<i>xplus</i>	xplus error
<i>xminus</i>	xminus error
<i>yplus</i>	yplus error
<i>yminus</i>	yminus error

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

- /data/lhcb/users/tat/KKpipi_Binned_Fit/include/CPPParameters.h
- /data/lhcb/users/tat/KKpipi_Binned_Fit/src/CPPParameters.cpp

2.5 DDecayParameters Class Reference

```
#include <DDecayParameters.h>
```

Public Member Functions

- [DDecayParameters](#) (const [PhaseSpaceParameterisation](#) &psp, const double &mass_parent, const double *mass_decay, int events)
- std::vector< double > [GetK](#) () const
- std::vector< double > [GetKbar](#) () const
- std::vector< double > [Getc](#) () const
- std::vector< double > [Gets](#) () const

2.5.1 Detailed Description

[DDecayParameters](#) is the class that calculates and stores the parameters describing the D^0 and \bar{D}^0 decay. These parameters only need to be calculated once because they only depend on the amplitude model.

2.5.2 Constructor & Destructor Documentation

2.5.2.1 DDecayParameters()

```
DDecayParameters::DDecayParameters (
    const PhaseSpaceParameterisation & psp,
    const double & mass_parent,
    const double * mass_decay,
    int events )
```

Constructor that takes in a [PhaseSpaceParameterisation](#) object and calculates the D decay parameters in each bin

Parameters

<i>psp</i>	PhaseSpaceParameterisation object
<i>events</i>	Number of events in each bin for Monte Carlo integration

2.5.3 Member Function Documentation

2.5.3.1 Getc()

```
std::vector< double > DDecayParameters::Getc ( ) const
```

Function for getting cosine of the strong phase

Returns

c Vector of cosine of the strong phases

2.5.3.2 GetK()

```
std::vector< double > DDecayParameters::GetK ( ) const
```

Function for getting fractional yield K_i

Returns

K Vector of fractional yields of D0 events

2.5.3.3 GetKbar()

```
std::vector< double > DDecayParameters::GetKbar ( ) const
```

Function for getting fractional yield K_i

Returns

K Vector of fractional yields of DBAR0 events

2.5.3.4 Gets()

```
std::vector< double > DDecayParameters::Gets ( ) const
```

Function for getting sine of the strong phase

Returns

s Vector of sine of the strong phases

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

- /data/lhcb/users/tat/KKpipi_Binned_Fit/include/DDecayParameters.h
- /data/lhcb/users/tat/KKpipi_Binned_Fit/src/DDecayParameters.cpp

2.6 Event Class Reference

```
#include <Event.h>
```

Public Member Functions

- [Event](#) ()
- [Event](#) (std::vector< double > p)
- std::vector< double > [GetEventVector](#) ()
- [Event](#) (const std::vector< TLorentzVector > &p)
- double [GetInvMass2](#) (int particle1, int particle2) const
- double [GetInvMass3](#) (int particle1, int particle2, int particle3) const
- std::vector< double > [GetEvent](#) () const

2.6.1 Detailed Description

[Event](#) is a class for storing the four-momenta of daughter particles in a D->KKpipi decay.

2.6.2 Constructor & Destructor Documentation

2.6.2.1 Event() [1/3]

```
Event::Event ( )
```

Default constructor for D to K+ K- pi+ pi- event with zero momentum

2.6.2.2 Event() [2/3]

```
Event::Event (
    std::vector< double > p )
```

Constructor that takes a vector of four-momenta

Parameters

p	Four-momenta in the form (E, px, py, pz), in the order K+ K- pi+ pi-
-----	--

2.6.2.3 Event() [3/3]

```
Event::Event (
    const std::vector< TLorentzVector > & p )
```

Constructor that takes a vector of four-momenta

Parameters

p	Vector of TLorentzVector objects, in the order K+ K- pi+ pi-
-----	--

2.6.3 Member Function Documentation**2.6.3.1 GetEvent()**

```
std::vector< double > Event::GetEvent ( ) const
```

Function for getting vector of four-momenta of event

Returns

Vector of four-momenta

2.6.3.2 GetEventVector()

```
std::vector< double > Event::GetEventVector ( )
```

Returns the four-momenta of daughter particles as a vector

Returns

Four-momenta of daughter particles in the form (E, px, py, pz), in the order K+ K- pi+ pi-

2.6.3.3 GetInvMass2()

```
double Event::GetInvMass2 (
    int particle1,
    int particle2 ) const
```

Function for getting invariant mass of two particles

Parameters

<i>particle1</i>	Particle 0(K+), 1(K-), 2(pi+), 3(pi-)
<i>particle2</i>	Particle 0(K+), 1(K-), 2(pi+), 3(pi-)

Returns

Returns invariant mass of given particles

2.6.3.4 GetInvMass3()

```
double Event::GetInvMass3 (
    int particle1,
    int particle2,
    int particle3 ) const
```

Function for getting invariant mass of three particles

Parameters

<i>particle1</i>	Particle 0(K+), 1(K-), 2(pi+), 3(pi-)
<i>particle2</i>	Particle 0(K+), 1(K-), 2(pi+), 3(pi-)
<i>particle3</i>	Particle 0(K+), 1(K-), 2(pi+), 3(pi-)

Returns

Returns mass of given particles

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

- /data/lhcb/users/tat/KKpipi_Binned_Fit/include/Event.h
- /data/lhcb/users/tat/KKpipi_Binned_Fit/src/Event.cpp

2.7 EventList Class Reference

```
#include <EventList.h>
```

Public Member Functions

- [EventList](#) ()
- void [AddEvent](#) ([Event](#) event)
- int [NumberEvents](#) () const
- std::vector< [Event](#) > [GetEvents](#) ()

2.7.1 Detailed Description

[EventList](#) is a class that contains all events in a sample

2.7.2 Constructor & Destructor Documentation

2.7.2.1 EventList()

```
EventList::EventList ( )
```

Default constructor that creates an empty [EventList](#)

2.7.3 Member Function Documentation

2.7.3.1 AddEvent()

```
void EventList::AddEvent (
    Event event )
```

Function that adds an [Event](#) to the [EventList](#)

Parameters

<i>event</i>	New Event object to be added to the EventList
--------------	---

2.7.3.2 GetEvents()

```
std::vector< Event > EventList::GetEvents ( )
```

Function that returns the vector of [Event](#) objects

Returns

Vector of [Event](#) objects

2.7.3.3 NumberEvents()

```
int EventList::NumberEvents ( ) const
```

Function that returns total number of events in this [EventList](#)

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

- /data/lhcb/users/tat/KKpipi_Binned_Fit/include/EventList.h
- /data/lhcb/users/tat/KKpipi_Binned_Fit/src/EventList.cpp

2.8 Fitter Class Reference

```
#include <Fitter.h>
```

Public Member Functions

- [Fitter](#) ([BinList](#) binlist, [DDecayParameters](#) ddparameters)
- void [DoFit](#) ([CParameters](#) &cparameters)

2.8.1 Detailed Description

[Fitter](#) is a class for maximising the likelihood and obtaining the CP violation parameters for a B meson decay

2.8.2 Constructor & Destructor Documentation

2.8.2.1 Fitter()

```
Fitter::Fitter (
    BinList binlist,
    DDecayParameters ddparameters )
```

Constructor that takes in a [BinList](#) object of input data and D meson decay parameters

Parameters

<i>binlist</i>	Input data events
<i>ddecayparameters</i>	Parameters describing the D meson decay

2.8.3 Member Function Documentation

2.8.3.1 DoFit()

```
void Fitter::DoFit (
    CPPParameters & cppparameters )
```

Function for doing fit and returning the CP violation parameters (by reference)

Parameters

<i>cppparameters</i>	Initial guess of CP violation parameters, function replaces these with the fitted parameters
----------------------	--

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

- /data/lhcb/users/tat/KKpipi_Binned_Fit/include/Fitter.h
- /data/lhcb/users/tat/KKpipi_Binned_Fit/src/Fitter.cpp

2.9 Generator Class Reference

```
#include <Generator.h>
```

Public Member Functions

- [Generator](#) (const double &mass_parent, const Double_t *mass_decay, Int_t particles)
- std::vector< TLorentzVector > [Generate](#) ()

2.9.1 Detailed Description

[Generator](#) is a class that generates uniformly distributed events in phase space, assuming the parent particle is at rest

2.9.2 Constructor & Destructor Documentation

2.9.2.1 Generator()

```
Generator::Generator (
    const double & mass_parent,
    const Double_t * mass_decay,
    Int_t particles )
```

Constructor that takes in the particle passes and sets up phase space

Parameters

<i>mass_parent</i>	Mass of parent particle
<i>mass_decay</i>	mass of decay particles
<i>particles</i>	Number of particles in the final state

2.9.3 Member Function Documentation

2.9.3.1 Generate()

```
std::vector< TLorentzVector > Generator::Generate ( )
```

Function that generates a random unweighted event

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

- /data/lhcb/users/tat/KKpipi_Binned_Fit/include/Generator.h
- /data/lhcb/users/tat/KKpipi_Binned_Fit/src/Generator.cpp

2.10 Likelihood Class Reference

```
#include <Likelihood.h>
```

Public Member Functions

- [Likelihood](#) ([BinList](#) bins, [DDecayParameters](#) ddparameters)
- double [operator\(\)](#) (const double *cpparameters)

2.10.1 Detailed Description

[Likelihood](#) is a class for calculating the likelihood, given an [EventList](#) of input data and a set of D meson decay parameters and CP violation parameters in B meson decays () operator is overloaded to make the likelihood function easily accessible

2.10.2 Constructor & Destructor Documentation

2.10.2.1 Likelihood()

```
Likelihood::Likelihood (
    BinList bins,
    DDecayParameters ddparameters )
```

Constructor that takes in an [BinList](#) object with input data and a [DDecayParameters](#) object

Parameters

<i>events</i>	BinList object with the input data
<i>ddecayparameters</i>	A DDecayParameters object with the parameters for the D meson decay

2.10.3 Member Function Documentation

2.10.3.1 operator()

```
double Likelihood::operator() (
    const double * cparameters )
```

Operator overload of () to easily access the likelihood function

Parameters

<i>cparameters</i>	A CParameters object with the CP violation parameters for the B meson decay
--------------------	---

Returns

$-2 \cdot \ln(L)$, where L is the likelihood function

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

- /data/lhcb/users/tat/KKpipi_Binned_Fit/include/Likelihood.h
- /data/lhcb/users/tat/KKpipi_Binned_Fit/src/Likelihood.cpp

2.11 PhaseSpaceParameterisation Class Reference

```
#include <PhaseSpaceParameterisation.h>
```

Public Member Functions

- [PhaseSpaceParameterisation](#) ()
- int [WhichBin](#) (const [Event](#) &event)
- int [NumberOfBins](#) ()

2.11.1 Detailed Description

[PhaseSpaceParameterisation](#) is a class that contains the information about how phase space is divided into bins. [PhaseSpaceParameterisation](#) contains a very coarse and arbitrary binning of phase space. A more sophisticated binning can be added by adding a new class that inherits from [PhaseSpaceParameterisation](#).

2.11.2 Constructor & Destructor Documentation

2.11.2.1 PhaseSpaceParameterisation()

```
PhaseSpaceParameterisation::PhaseSpaceParameterisation ( )
```

Default constructor

2.11.3 Member Function Documentation

2.11.3.1 NumberOfBins()

```
int PhaseSpaceParameterisation::NumberOfBins ( )
```

Function that returns the number of bins in the binning scheme

Returns

Number of bins

2.11.3.2 WhichBin()

```
int PhaseSpaceParameterisation::WhichBin (
    const Event & event )
```

Function that determines which bin an event belongs to

Parameters

<i>event</i>	The event we want to determine the bin of
--------------	---

Returns

Bin number

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

- /data/lhcb/users/tat/KKpipi_Binned_Fit/include/PhaseSpaceParameterisation.h
- /data/lhcb/users/tat/KKpipi_Binned_Fit/src/PhaseSpaceParameterisation.cpp

Index

AddEvent
 Bin, [4](#)
 BinList, [6](#), [7](#)
 EventList, [16](#)

Amplitude, [3](#)
 Amplitude, [3](#)
 operator(), [3](#)

Bin, [4](#)
 AddEvent, [4](#)
 Bin, [4](#)
 GetEvents, [5](#)
 GetNumberEvents, [5](#)

BinList, [5](#)
 AddEvent, [6](#), [7](#)
 BinList, [6](#)
 GetBin, [7](#)
 GetEvents, [7](#)
 LoadTTree, [8](#)
 NumberBins, [8](#)
 Predict, [8](#)

CPPParameters, [9](#)
 CPPParameters, [9](#)
 GetCPPParameters, [9](#)
 GetError, [10](#)
 SetError, [10](#)

DDecayParameters, [11](#)
 DDecayParameters, [11](#)
 Getc, [12](#)
 GetK, [12](#)
 GetKbar, [12](#)
 Gets, [12](#)

DoFit
 Fitter, [17](#)

Event, [13](#)
 Event, [13](#), [14](#)
 GetEvent, [14](#)
 GetEventVector, [14](#)
 GetInvMass2, [14](#)
 GetInvMass3, [15](#)

EventList, [15](#)
 AddEvent, [16](#)
 EventList, [16](#)
 GetEvents, [16](#)
 NumberEvents, [16](#)

Fitter, [17](#)
 DoFit, [17](#)
 Fitter, [17](#)

Generate
 Generator, [19](#)

Generator, [18](#)
 Generate, [19](#)
 Generator, [18](#)

GetBin
 BinList, [7](#)

Getc
 DDecayParameters, [12](#)

GetCPPParameters
 CPPParameters, [9](#)

GetError
 CPPParameters, [10](#)

GetEvent
 Event, [14](#)

GetEvents
 Bin, [5](#)
 BinList, [7](#)
 EventList, [16](#)

GetEventVector
 Event, [14](#)

GetInvMass2
 Event, [14](#)

GetInvMass3
 Event, [15](#)

GetK
 DDecayParameters, [12](#)

GetKbar
 DDecayParameters, [12](#)

GetNumberEvents
 Bin, [5](#)

Gets
 DDecayParameters, [12](#)

Likelihood, [19](#)
 Likelihood, [19](#)
 operator(), [20](#)

LoadTTree
 BinList, [8](#)

NumberBins
 BinList, [8](#)

NumberEvents
 EventList, [16](#)

NumberOfBins
 PhaseSpaceParameterisation, [21](#)

operator()

Amplitude, [3](#)
Likelihood, [20](#)

PhaseSpaceParameterisation, [20](#)
 NumberOfBins, [21](#)
 PhaseSpaceParameterisation, [21](#)
 WhichBin, [21](#)

Predict
 BinList, [8](#)

SetError
 CPPParameters, [10](#)

WhichBin
 PhaseSpaceParameterisation, [21](#)