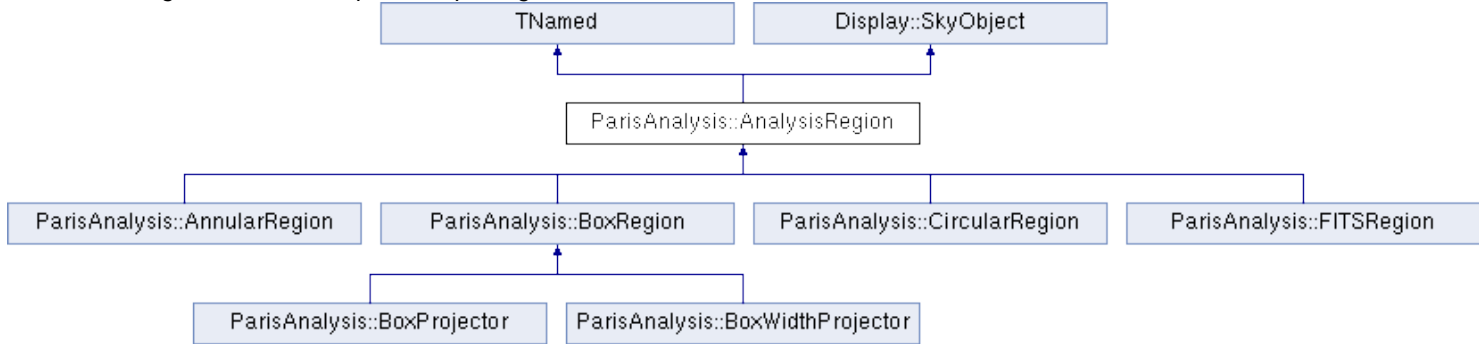


ParisAnalysis::AnalysisRegion Class Reference

[Arbitrary region shapes]

Inheritance diagram for ParisAnalysis::AnalysisRegion:



Detailed Description

Base class for [Arbitrary region shapes](#).

Defines an arbitrary region in the sky to be excluded from the background regions used in map constructions (because it might contain gammas)

Author:

Mathieu de Naurois

Definition at line 39 of file [AnalysisRegion.hh](#).

[List of all members.](#)

Public Types

enum **RegionType** { [TargetRegion](#), [ExcludedRegion](#), [OFFRegion](#), [RegionOfInterest](#) }
Type of region.

[More...](#)

Public Member Functions

AnalysisRegion (const char *name="Region", RegionType type=TargetRegion)
default constructor

AnalysisRegion (const [AnalysisRegion](#) &)
Analysis Region copy constructor.

AnalysisRegion (const [Stash::Coordinate](#) &pos, const [Stash::Lambda](#) &size, const char *name="Region", RegionType type=TargetRegion)
Analysis region constructor.

virtual const [Stash::Coordinate](#) & **GetPos** () const
Center of gravity.

virtual const [Stash::Coordinate](#) & **GetNomPos** () const
Center of gravity, in Nominal System.

virtual [Stash::Lambda](#) **GetRegionSize** () const
Maximum Radius of region, from center of gravity.

virtual Double_t **GetSolidAngle** () const =0
Region solid angle.

[Stash::Lambda](#) **GetRotationAngle** (const [Stash::System](#) &) const
Computes the rotation angle between the current system and a new one.

[Stash::Lambda](#) **GetRotationAngle** () const

virtual void **Rotate** (const [Stash::Coordinate](#) ¢er, const [Stash::Angle](#) &a)
Rotates analysis region in the sky around a direction.

const char * **GetRegionTypeName** () const

	Return string for region type.
virtual void	UpdateSystem (const Stash::System &system) Changes reference system.
virtual void	UpdateNomPos (Sash::HESSArray &fHess, const Stash::Coordinate &ObservationPos) Update positions in Nominal System.
virtual bool	Contains (const Stash::Coordinate &pos) const =0 Checks if region contains a point.
virtual bool	Contains (const Stash::Coordinate &pos, const Stash::Lambda &oversampling) const =0 Checks if region contains a point, taking oversampling into account.
virtual void	Configure (const char *what, const Stash::Coordinate &value) Configuration function.
virtual void	Configure (const char *what, const Stash::Lambda &value) Configuration function.
virtual void	Configure (const char *what, const char *value) Configuration function.
virtual void	Configure (const char *what, double value) Configuration function.
virtual void	PaintSky (const Display::SkyHistogramBase &sky) const Paints the region on the sky.
virtual void	print (std::ostream &os) const Print function.
virtual Stash::Lambda	ViewAngle (const Stash::Coordinate &pos) const Return aperture angle from a position of the region.
virtual double	Project (const Stash::Coordinate &pos) const Return the projection of the event in the zone.
virtual bool	Overlap (ParisAnalysis::AnalysisRegion *region, const Stash::Lambda &oversampling) const Return if a region overlap this region.
virtual bool	OverlapFast (ParisAnalysis::AnalysisRegion *region, const Stash::Lambda &oversampling) const Return if a region overlap this region Fast version, that only takes region size into account; not accurate for non-circular shapes.
virtual Float_t	GetNameOffset (const Display::SkyHistogramBase &sky) const
virtual Float_t	GetNameSize (const Display::SkyHistogramBase &sky) const
virtual Float_t	GetNameAngle (const Display::SkyHistogramBase &sky) const
virtual Color_t	GetLineColor (const Display::SkyHistogramBase &sky) const
virtual Color_t	GetLineStyle (const Display::SkyHistogramBase &sky) const
virtual Color_t	GetLineWidth (const Display::SkyHistogramBase &sky) const
virtual void	SetLineColor (Color_t color)
virtual void	SetLineStyle (Style_t style)
virtual void	SetLineWidth (Width_t width)
virtual void	SetNameOffset (Float_t offset=1)
virtual void	SetNameSize (Float_t size=0)
virtual void	SetNameAngle (Float_t angle=0.)
virtual void	ClearLineColor ()
virtual void	ClearLineStyle ()
virtual void	ClearLineWidth ()
virtual void	ClearNameOffset ()
virtual void	ClearNameSize ()
virtual void	ClearNameAngle ()
UInt_t &	GetGammaEventCount ()
UInt_t	GetGammaEventCount () const
UInt_t &	GetBackgroundEventCount ()
UInt_t	GetBackgroundEventCount () const
Utilities::SimpleStats &	GetNomPosX ()
const Utilities::SimpleStats &	GetNomPosX () const
Utilities::SimpleStats &	GetNomPosY ()
const Utilities::SimpleStats &	GetNomPosY () const
Utilities::SimpleStats &	GetZenithDistri ()
const Utilities::SimpleStats &	GetZenithDistri () const
Utilities::SimpleStats &	GetOffAxisDistri ()
const Utilities::SimpleStats &	GetOffAxisDistri () const
Utilities::SimpleStats &	GetAzimutDistri ()
const Utilities::SimpleStats &	GetAzimutDistri () const

RegionType GetRegionType () const
 void SetRegionType (**RegionType** type)

Protected Member Functions

virtual void Streamer (TBuffer &B)

Protected Attributes

RegionsProperty < Float_t >	fRegionsNameOffset
RegionsProperty < Float_t >	fRegionsNameSize
RegionsProperty < Float_t >	fRegionsNameAngle
RegionsProperty < Color_t >	fRegionsLineColor
RegionsProperty < Style_t >	fRegionsLineStyle
RegionsProperty < Width_t >	fRegionsLineWidth
Bool_t	fInialized Initialization state.
RegionType	fRegionType Type of Region.
Stash::Coordinate	fPos Region position (center of gravity)
Stash::Lambda	fRegionSize Size of region.
Stash::Lambda	fRotationAngle Rotation angle – for events.
Stash::Coordinate	fNomPos Position in Nominal System.
UInt_t	fGammaEventCount
UInt_t	fBackgroundEventCount
Utilities::SimpleStats	fNomPosX
Utilities::SimpleStats	fNomPosY
Utilities::SimpleStats	fZenithDistri
Utilities::SimpleStats	fOffAxisDistri
Utilities::SimpleStats	fAzimutDistri

Member Enumeration Documentation

enum ParisAnalysis::AnalysisRegion::RegionType

Type of region.

Enumerator:

- TargetRegion* Region used for signal determination
- ExcludedRegion* Excluded region, where significant gamma-ray emission is present. These regions are excluded from background determination
- OFFRegion* OFF region, used for background determination
- RegionOfInterest* Region of interest, where emission can be expected due, for instance to multiwavelength data; this type of regions is used to display source candidates

Definition at line 42 of file [AnalysisRegion.hh](#).

Constructor & Destructor Documentation

```
ParisAnalysis::AnalysisRegion::AnalysisRegion ( const char * name = "Region",
                                               RegionType type = TargetRegion
                                               )
```

default constructor

Parameters:

- name** Name of region
- type** Type of region

Definition at line 33 of file [AnalysisRegion.C](#).

```

ParisAnalysis::AnalysisRegion::AnalysisRegion ( const Stash::Coordinate & pos,
                                                const Stash::Lambda & size,
                                                const char * name = "Region",
                                                RegionType type = TargetRegion
                                                )

```

Analysis region constructor.

Parameters:

pos center of region
size region radius
name region name
type Region type

Definition at line 72 of file [AnalysisRegion.C](#).

Member Function Documentation

```

void ParisAnalysis::AnalysisRegion::Configure ( const char * what,
                                                double value
                                                ) [virtual]

```

Configuration function.

Parameters:

what string identifying what has to be configured.
value corresponding value

Reimplemented in [ParisAnalysis::BoxRegion](#).

Definition at line 329 of file [AnalysisRegion.C](#).

```

void ParisAnalysis::AnalysisRegion::Configure ( const char * what,
                                                const Stash::Coordinate & value
                                                ) [virtual]

```

Configuration function.

Parameters:

what string identifying what has to be configured.
value corresponding value

The configuration string can be

- **Pos** : [Source](#) Position

Reimplemented in [ParisAnalysis::BoxRegion](#).

Definition at line 286 of file [AnalysisRegion.C](#).

References [Stash::Coordinate::GetCoordinate\(\)](#), and [Stash::System::GetDummySystem\(\)](#).

Referenced by [ParisAnalysis::FITSRegion::Configure\(\)](#), [ParisAnalysis::CircularRegion::Configure\(\)](#), [ParisAnalysis::BoxRegion::Configure\(\)](#), and [ParisAnalysis::AnnularRegion::Configure\(\)](#).

```

void ParisAnalysis::AnalysisRegion::Configure ( const char * what,
                                                const Stash::Lambda & value
                                                ) [virtual]

```

Configuration function.

Parameters:

what string identifying what has to be configured.
value corresponding value

Reimplemented in [ParisAnalysis::BoxRegion](#).

Definition at line 307 of file [AnalysisRegion.C](#).

```
void ParisAnalysis::AnalysisRegion::Configure ( const char * what,
                                               const char * value
                                               )           [virtual]
```

Configuration function.

Parameters:

what string identifying what has to be configured.
value corresponding value

Reimplemented in [ParisAnalysis::BoxRegion](#).

Definition at line 318 of file [AnalysisRegion.C](#).

```
Stash::Lambda ParisAnalysis::AnalysisRegion::GetRotationAngle ( const Stash::System & sys ) const
```

Computes the rotation angle between the current system and a new one.

Parameters:

sys new system

Definition at line 161 of file [AnalysisRegion.C](#).

References [Stash::Angle::Degrees](#), [Stash::Coordinate::GetBeta\(\)](#), [Stash::Coordinate::GetCoordinate\(\)](#), [Stash::Coordinate::GetLambda\(\)](#), [GetRotationAngle\(\)](#), and [Stash::Angle::Radians](#).

Referenced by [GetRotationAngle\(\)](#).

```
void ParisAnalysis::AnalysisRegion::print ( std::ostream & os ) const [virtual]
```

Print function.

Prints the content of the region.

Reimplemented in [ParisAnalysis::AnnularRegion](#), [ParisAnalysis::BoxRegion](#), [ParisAnalysis::CircularRegion](#), and [ParisAnalysis::FITSRegion](#).

Definition at line 95 of file [AnalysisRegion.C](#).

```
double ParisAnalysis::AnalysisRegion::Project ( const Stash::Coordinate & pos ) const [virtual]
```

Return the projection of the event in the zone.

Theta2 if circular, main axis if box region

Reimplemented in [ParisAnalysis::BoxRegion](#), [ParisAnalysis::BoxProjector](#), and [ParisAnalysis::BoxWidthProjector](#).

Definition at line 207 of file [AnalysisRegion.C](#).

References [Stash::Coordinate::GetAngularDistance\(\)](#), and [Stash::Angle::GetDegrees\(\)](#).

```
void ParisAnalysis::AnalysisRegion::Rotate ( const Stash::Coordinate & center,
                                             const Stash::Angle & a
                                             )           [virtual]
```

Rotates analysis region in the sky around a direction.

Parameters:

- center** Center of rotation
- a** Angle of rotation

Reimplemented in [ParisAnalysis::AnnularRegion](#), [ParisAnalysis::BoxRegion](#), and [ParisAnalysis::FITSRegion](#).

Definition at line 183 of file [AnalysisRegion.C](#).

References [Stash::Coordinate::GetDirectionVector\(\)](#), and [Stash::DirectionVector::RotateAroundPoint\(\)](#).

Referenced by [ParisAnalysis::MultipleOffRegionsMaker::ProcessRun\(\)](#), [ParisAnalysis::BoxRegion::Rotate\(\)](#), and [ParisAnalysis::AnnularRegion::Rotate\(\)](#).

```
void ParisAnalysis::AnalysisRegion::UpdateNomPos ( Sash::HESSArray &      fHess,
                                                    const Stash::Coordinate & ObservationPos
                                                    )                                     [virtual]
```

Update positions in Nominal System.

Recalculates the region position in Nominal System.

Definition at line 132 of file [AnalysisRegion.C](#).

References [Stash::Coordinate::GetCoordinate\(\)](#), [Stash::System::GetDummySystem\(\)](#), [Sash::HESSArray::GetHorizonSystem\(\)](#), [Sash::HESSArray::GetNominalSystem\(\)](#), and [Crash::SetRefractionType\(\)](#).

```
void ParisAnalysis::AnalysisRegion::UpdateSystem ( const Stash::System & system ) [virtual]
```

Changes reference system.

Changes the used system.

Reimplemented in [ParisAnalysis::AnnularRegion](#), and [ParisAnalysis::BoxRegion](#).

Definition at line 148 of file [AnalysisRegion.C](#).

References [Stash::System::GetThisSys\(\)](#).

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

- [AnalysisRegion.hh](#)
- [AnalysisRegion.C](#)