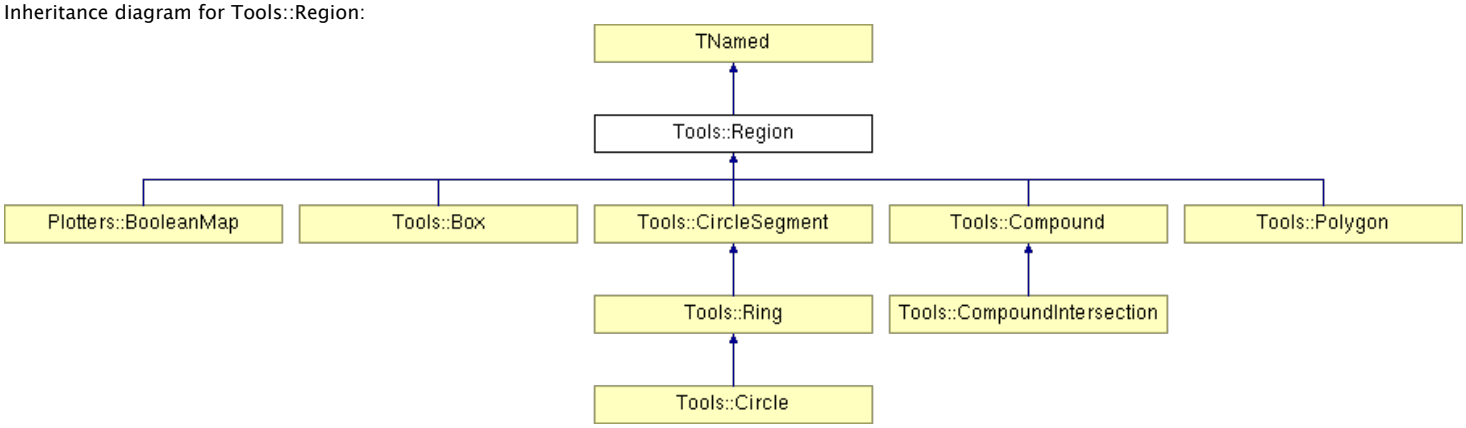
Tools::Region Class Reference
[utilities]

Base class for all regions. More...

`#include <Region.hh>`

Inheritance diagram for Tools::Region:



List of all members.

Public Types

typedef std::vector< const Tools::Region * >	**RegionList**
typedef std::vector< const Tools::Region * > ::const_iterator	**RegionListIterator**
typedef std::vector < std::vector < Stash::Coordinate > >	**VertexList**

Public Member Functions

	Region (const Stash::Coordinate &cor=Stash::Coordinate(0, 0, Stash::System::GetDummySystem()), std::string name="Region", std::string type="", Stash::Lambda phi=Stash::Lambda(0, Stash::Angle::Degrees))
double	GetDistanceToCentre (const Stash::Coordinate &cor) const
Stash::Coordinate	GetTrueCentre () const Get the true center (after rotation is applied) of the region.
Stash::Coordinate	GetCentre () const Get center of the region (before rotation is applied, as given during construction).
const Stash::SystemRef &	GetSystem () const Get coordinate system of the region center.
virtual void	MoveTo (Stash::Coordinate newcen) Move the region to another part of the sky.
bool	Overlaps (const Tools::Region &reg) const
std::string	GetType () String representation of region type.
double	GetEquivalentCircleRadius () const
void	SetAccuracy (double maxerror) accuracy is used in calculations that sample points over the region (e.g. to calculate overlaps or zenith distributions, etc). Increasing it speeds things up at the expense of some accuracy.
Stash::Lambda	GetRotationAngle () const
Stash::Coordinate	GetRotationPoint () const
void	SetRotation (Stash::Lambda phi=Stash::Lambda(0, Stash::Angle::Degrees), Stash::Coordinate point=Stash::Coordinate(0, 0, Stash::System::GetDummySystem()))
void	SetInverted (bool inv=true)
bool	IsInverted () const
virtual Tools::Region *	GrowBoundary (const double degrees, const std::string name="_grown") const =0 Grow the boundary of the region by a given amount of space angle.
bool	Contains (const Stash::Coordinate &dir) const Check if dir is contained in this region.

virtual bool	**IsCompletelyContained** (const **Tools::Region** &reg) const Check if another region is completely contained in this region.
virtual **Stash::Coordinate**	**GetCentreOfGravity** () const
virtual void	**FillOffsetDistribution** (const **Stash::Coordinate** &pos, TH1 &, double psicut=-1) const
virtual void	**FillZenithOrAzimuthDistribution** (TH1 &, const **Sash::Time**, const **Sash::Time**, bool fillZenith=true) const
virtual void	**FillZenithOffsetDistribution** (const **Stash::Coordinate** &pos, TH2 &, const **Sash::Time**, const **Sash::Time**, double psicut=-1) const
virtual void	**Fill2AxesDistanceDistribution** (const **Stash::Coordinate** &pos, TH2 &, double psicut=-1) const
virtual double	**GetArea** () const =0 Returns area of region in square degrees.
virtual double	**GetBound2** () const =0 Return width of bounding box.
virtual double	**GetBound1** () const =0 Return length of bounding box.
virtual double	**GetRadialBound** () const =0 Return radius of **circle** containing reg.
Tools::Box *	**GetBoundingBox** () const
Tools::Circle *	**GetBoundingCircle** () const
double	**GetMonteCarloArea** (int samples_per_square_deg=1e6) const
virtual std::string	**AsFITSRegionString** () const Return FITS region string representation of this region.
virtual VertexList	**AsVertexList** () const Return VertexList representation of this region.

Protected Member Functions

std::string	**GetFITSSystemName** () const
Stash::Coordinate	**Transform** (const **Stash::Coordinate** &src) const
Stash::Coordinate	**InverseTransform** (const **Stash::Coordinate** &src) const
Stash::Coordinate	**GetPointAtOffsetFromCenter** (double offsX, double offsY) const
void	**GetOffsetOfPointFromCenter** (**Stash::Coordinate** point, double &offsX, double &offsY, bool untransformed=false) const
Stash::Coordinate	**GetPointAtRadius** (**Stash::Beta** rad, **Stash::Lambda** phi) const
int	**GetNumSpaceSteps** (double radius) const
int	**GetNumTimeSteps** () const
virtual bool	**IsInside** (const **Stash::Coordinate** &dir) const =0 Check if dir is inside this region.

Protected Attributes

Stash::Coordinate	**fCentre** Centre of **Region**.
double	**fAccuracy** accuracy of grid or monte-carlo operations 1.0=normal, larger is more accurate (more steps used in loops)

Detailed Description

Base class for all regions.

This is a base class that defines a physical region on the sky, with methods to test if a point is inside (**Region::Contains**(point)) or if regions overlap (Region::Overlaps(region)). The base-class is abstract, and defines a set of virtual functions that may be implemented by sub-classes, such as **Tools::Box**, or **Tools::Circle**.

Regions can be:

- rotated (about an arbitrary point) via **SetRotation()**
- inverted (rejecting points inside and accepting points outside) via **SetInverted()**
- grouped into **Tools::Compound** regions (using inverted regions, you may construct complex shapes)

Note:
Since **Tools::Region** defines an internal rotation transformation (about an arbitrary coordinate), sub-classes do not need to deal with rotations directly, and can assume that the region is fixed at the center position and aligned with the X-Y plane.
that this means that internally, fCentre can be assumed to be unrotated, and should be used in methods like **IsInside()**, while **GetCentre()** returns the true region center on the sky (after applying the rotation transformation).

Author:
Heidelberg team + CEA team

Definition at line **49** of file **Region.hh**.

Member Function Documentation

virtual std::string Tools::Region::AsFITSRegionString ( ) const [inline, virtual]

Return FITS region string representation of this region.

Subclasses should implement this such that it returns a valid FITS region string. Used by the output streamer and Print()

Reimplemented in Plotters::BooleanMap, Tools::Polygon, Tools::CircleSegment, Tools::Ring, Tools::Circle, Tools::Box, and Tools::Compound.

Definition at line 164 of file Region.hh.

virtual VertexList Tools::Region::AsVertexList ( ) const [inline, virtual]

Return VertexList representation of this region.

To be implemented by subclasses: shold return a representation/approximatin of the region as a continuous list of coordinates that follows the boundaries of the region (this is used for drawing the regions in the correct transformation)

The vertex list returned is in the native system of the region.

Reimplemented in Tools::Polygon, Tools::CircleSegment, Tools::Box, and Tools::Compound.

Definition at line 179 of file Region.hh.

Referenced by Plotters::SkyHist::DrawRegion().

void Tools::Region::Fill2AxesDistanceDistribution (const Stash::Coordinate & refpos,
TH2 & hist2,
double psicut = -1
) const [virtual]

Fills histogram off 2d offsets in the sky from

Parameters:
psicut - will be ignored, if set to -1.

Definition at line 546 of file Region.C.

References Contains(), Stash::Angle::Degrees, Stash::Coordinate::GetAngularDistance(), Stash::Coordinate::GetBeta(), GetCentre(), Stash::Coordinate::GetCoordinate(), Stash::Coordinate::GetLambda(), GetNumSpaceSteps(), GetRadialBound(), GetSystem(), and GetTrueCentre().

void Tools::Region::FillOffsetDistribution (const Stash::Coordinate & refpos,
TH1 & hist,
double psicut = -1
) const [virtual]

Fills a histogram of the Offset distribution from the given reference position

Parameters:
psicut - will be ignored, if set to -1.

Reimplemented in Tools::Compound.

Definition at line 191 of file Region.C.

References Contains(), Stash::Angle::Degrees, Stash::Coordinate::GetAngularDistance(), Stash::Coordinate::GetBeta(), Stash::Coordinate::GetCoordinate(), Stash::Coordinate::GetLambda(), GetNumSpaceSteps(), GetRadialBound(), GetSystem(), and GetTrueCentre().

void Tools::Region::FillZenithOffsetDistribution (const Stash::Coordinate & refpos,
TH2 & hist,
const Sash::Time starttime,
const Sash::Time endtime,
double psicut = -1
) const [virtual]

Generates the zenith angle / offset to obs. pos 2D - distribution of a given region over the specified time range.

Parameters:
hist - histogram to fill
starttime - starting time of the run
endtime - ending time of the run
psicut - will be ignored, if set to -1.

Reimplemented in Tools::Compound.

Definition at line 463 of file Region.C.

References Contains(), Stash::Angle::Degrees, Sash::HESSArray::GetAltAzSystem(), Stash::Coordinate::GetAngularDistance(), Stash::Coordinate::GetBeta(), Stash::Coordinate::GetCoordinate(), Sash::HESSArray::GetHESSArray(), Stash::Coordinate::GetLambda(),

GetNumSpaceSteps(), GetNumTimeSteps(), GetRadialBound(), GetSystem(), and GetTrueCentre().

void Tools::Region::FillZenithOrAzimuthDistribution (TH1 & hist, const Sash::Time starttime, const Sash::Time endtime, bool fillZenith = true) const [virtual]

Generates the zenith or azimuth angle distribution of a given region over the specified time range.

Parameters:

hist - histogram to fill
starttime - starting time of the run
endtime - ending time of the run
fillZenith - decide whether the zenith (true, default) or azimuth (false) angle will be filled

Todo:
Document the sampling method implemented here!

Reimplemented in Tools::Compound.

Definition at line 380 of file Region.C.

References Contains(), Stash::Angle::Degrees, Sash::HESSArray::GetAltAzSystem(), Stash::Coordinate::GetBeta(), Stash::Coordinate::GetCoordinate(), Sash::HESSArray::GetHESSArray(), Stash::Coordinate::GetLambda(), GetNumSpaceSteps(), GetNumTimeSteps(), GetRadialBound(), GetSystem(), and GetTrueCentre().

Tools::Box * Tools::Region::GetBoundingBox ( ) const

Returns a Tools::Box that bounds the region

Definition at line 614 of file Region.C.

References Stash::Angle::Degrees, GetBound1(), GetBound2(), and GetTrueCentre().

Stash::Coordinate Tools::Region::GetCentreOfGravity ( ) const [virtual]

Loops over the region and calculates an approximate center of gravity. This can be re-implemented in sub-classes where the center of gravity is known mathematically, to increase speed.

Definition at line 139 of file Region.C.

References Contains(), Stash::Angle::Degrees, Stash::Coordinate::GetBeta(), Stash::Angle::GetDegrees(), Stash::Coordinate::GetLambda(), GetNumSpaceSteps(), GetRadialBound(), GetSystem(), and GetTrueCentre().

Referenced by Tools::Polygon::GetBound2(), Plotters::ReadRegionsFromFile(), Tools::Compound::RecenterOnCOG(), and Tools::Polygon::RecenterOnCOG().

double Tools::Region::GetDistanceToCentre (const Stash::Coordinate & cor) const

Returns:
distance of the given coordinate to the true center-point of the region.

Definition at line 128 of file Region.C.

References Stash::Coordinate::GetAngularDistance(), Stash::Coordinate::GetCoordinate(), GetSystem(), and GetTrueCentre().

double Tools::Region::GetEquivalentCircleRadius ( ) const

Returns the radius of a circle with the same area as the region

Definition at line 179 of file Region.C.

References GetArea().

Referenced by Background::SimpleBgMaker::BeginRun().

string Tools::Region::GetFITSSystemName ( ) const [protected]

Returns:
a string representing the coordinate system for use with AsFITSRegionString()

Definition at line 240 of file Region.C.

References GetSystem().

Referenced by Tools::Box::AsFITSRegionString(), Tools::Circle::AsFITSRegionString(), Tools::Ring::AsFITSRegionString(), Tools::CircleSegment::AsFITSRegionString(), and Tools::Polygon::AsFITSRegionString().

double Tools::Region::GetMonteCarloArea (int samples_per_square_deg = 1e6) const

Gives an approximation of the area using Monte-Carlo integration. This is useful when you have a Compound region with overlapping sub-regions, which would not be properly taken into account in GetArea(), which simply sums the area of all sub-regions. This method is much slower than GetArea, however.

Parameters:
samples_per_square_deg,: approximate sample density to use

Todo:
: this can be optimized to use the region's bounding box, but bounding-boxes needs to be well-tested first.

Note:
this function is used for testing purposes when debugging regions and the Contains() function. It's not intended for production use. Generally one should use the GetArea() function, which does it analytically.

Definition at line 338 of file Region.C.

References Contains(), Tools::Box::GetArea(), GetRadialBound(), and GetTrueCentre().

int Tools::Region::GetNumSpaceSteps (double radius) const [inline, protected]

Returns the optimal number of spatial steps for a region of a given size, taking into accunt the accuracy leve;

Parameters:
radius,: approximate radius of region in deg

Definition at line 201 of file Region.hh.

References fAccuracy.

Referenced by Fill2AxesDistanceDistribution(), FillOffsetDistribution(), FillZenithOffsetDistribution(), FillZenithOrAzimuthDistribution(), GetCentreOfGravity(), IsCompletelyContained(), and Overlaps().

int Tools::Region::GetNumTimeSteps ( ) const [inline, protected]

Returns the optimal number of spatial steps for looping over a time range

Definition at line 209 of file Region.hh.

References fAccuracy.

Referenced by FillZenithOffsetDistribution(), and FillZenithOrAzimuthDistribution().

void Tools::Region::GetOffsetOfPointFromCenter (Stash::Coordinate point, double & offsX, double & offsY, bool untransformed = false) const [protected]

Returns:
offset in sky degrees of point from the region center (similar to GetAngularDistance(), but returning both directions)

Parameters:
untransformed - return the offset from the untransformed center point

Definition at line 1156 of file Region.C.

References fCentre, Stash::Coordinate::GetBeta(), Stash::Coordinate::GetLambda(), Stash::Coordinate::GetSystem(), and GetTrueCentre().

Referenced by Tools::Compound::GetBound1(), Tools::Box::GetBound1(), Tools::Polygon::GetBound1(), Tools::Compound::GetBound2(), Tools::Box::GetBound2(), Tools::Polygon::GetBound2(), GetPointAtOffsetFromCenter(), and Tools::Box::IsInside().

Stash::Coordinate Tools::Region::GetPointAtOffsetFromCenter (double offsX, double offsY) const [protected]

Returns:
a point at a given offset in "sky degrees" (e.g. not RA/Dec degrees).

Definition at line 1121 of file Region.C.

References Stash::Angle::Degrees, fCentre, Stash::Coordinate::GetBeta(), and GetOffsetOfPointFromCenter().

Referenced by Tools::Box::AsVertexList().

Stash::Coordinate Tools::Region::GetPointAtRadius (Stash::Beta rad,
Stash::Lambda phi
) const [protected]

Returns:
a point on a circle with given radius from the Centre position. Used by CircleSegment::AsVertexList.

Note:
rad does need to be a Stash::Beta (not Lambda) in order for the "RA factor" to be taken into account properly)

Definition at line 873 of file Region.C.

References fCentre, Stash::Coordinate::GetDirectionVector(), Stash::Lambda::GetNegative(), GetSystem(), and Stash::DirectionVector::RotateAroundPoint().

Referenced by Tools::CircleSegment::AsVertexList(), and Tools::Polygon::GrowBoundary().

Stash::Coordinate Tools::Region::GetTrueCentre ( ) const [inline]

Get the true center (after rotation is applied) of the region.

This is the inverse-transform of fCentre.

Definition at line 73 of file Region.hh.

References fCentre, and InverseTransform().

Referenced by Tools::Compound::AsFITSRegionString(), Tools::Box::AsFITSRegionString(), Tools::Circle::AsFITSRegionString(), Tools::Ring::AsFITSRegionString(), Tools::CircleSegment::AsFITSRegionString(), Fill2AxesDistanceDistribution(), FillOffsetDistribution(), FillZenithOffsetDistribution(), FillZenithOrAzimuthDistribution(), GetBoundingBox(), GetCentreOfGravity(), GetDistanceToCentre(), GetMonteCarloArea(), GetOffsetOfPointFromCenter(), IsCompletelyContained(), and Overlaps().

virtual Tools::Region* Tools::Region::GrowBoundary (const double degrees,
const std::string name = "_grown"
) const [pure virtual]

Grow the boundary of the region by a given amount of space angle.

Note:
Has to be implemented by subclasses.

Implemented in Plotters::BooleanMap, Tools::Polygon, Tools::CircleSegment, Tools::Box, and Tools::Compound.

Stash::Coordinate Tools::Region::InverseTransform (const Stash::Coordinate & src) const [protected]

Undo the internal rotation transform of the region (useful for getting back to proper "Sky" coordinates, rather than the internal de-rotated coordinates of the region. It should be used in AsVertexList() to undo the internal transform.

Definition at line 66 of file Region.C.

References Stash::Coordinate::GetCoordinate(), Stash::Coordinate::GetDirectionVector(), Stash::Lambda::GetNegative(), GetSystem(), Stash::Coordinate::GetSystem(), and Stash::DirectionVector::RotateAroundPoint().

Referenced by Tools::Polygon::AsFITSRegionString(), Tools::Compound::AsVertexList(), Tools::Box::AsVertexList(), Tools::CircleSegment::AsVertexList(), Tools::Polygon::AsVertexList(), Tools::Compound::GetBound1(), Tools::Polygon::GetBound1(), Tools::Compound::GetBound2(), Tools::Polygon::GetBound2(), and GetTrueCentre().

virtual bool Tools::Region::IsInside (const Stash::Coordinate & dir) const [protected, pure virtual]

Check if dir is inside this region.

Must be implemented by the sub-classes. This function should return true if the point is inside the *un-rotated* region (the rotation is taken into account automatically by Contains()

Implemented in Plotters::BooleanMap, Tools::Polygon, Tools::CircleSegment, Tools::Ring, Tools::Circle, Tools::Box, Tools::Compound, and Tools::CompoundIntersection.

Referenced by Contains().

bool Tools::Region::Overlaps (const Tools::Region & reg) const

Check whether the given region overlaps this one

Definition at line 293 of file Region.C.

References Contains(), Stash::Angle::Degrees, Stash::Coordinate::GetBeta(), Stash::System::GetDummySystem(), Stash::Coordinate::GetLambda(), GetNumSpaceSteps(), GetRadialBound(), GetSystem(), and GetTrueCentre().

void Tools::Region::SetInverted (bool inv = true) [inline]

Control whether region is inverted or not (inverted means points outside will return true Contains())

Definition at line 121 of file Region.hh.

Referenced by Background::BgMaker::ConstructExclusionRegions(), and Background::BgMaps::CutAwayBorders().

void Tools::Region::SetRotation (Stash::Lambda phi = Stash::Lambda(0,Stash::Angle::Degrees),
Stash::Coordinate point = Stash::Coordinate(0,0,Stash::System::GetDummySystem())
)

Set the rotation of this region. Each region can be rotated about an arbitrary point (if no point is specified, the region center is used). The rotation angle is relative to North in the Region's cordinate System, and increases counter-clockwise (the is the same definition of the "position angle" in astronomy).

```
// Example: rotate a region at the test position about the observation position
Tools::Box *box = new Tools::Box( testpos, wid, len );
box->SetRotation( Stash::Beta(45,Stash::Angle::Degrees), obspos );
```

If you want to apply successive rotations (e.g. rotate first about one point, and then again about another), you should first rotate the region and then add it to a Tools::Compound (which contains its own rotation transform), and apply the second rotation to that.

Definition at line 96 of file Region.C.

References GetCentre(), Stash::Coordinate::GetCoordinate(), Stash::System::GetDummySystem(), and GetSystem().

Referenced by Tools::Compound::AddRegion(), and Tools::Compound::Reset().

Stash::Coordinate Tools::Region::Transform (const Stash::Coordinate & src) const [protected]

Currently Applies a rotation to the given point (for passively rotating the region - this should be called in Contains() after converting to the RegionSystem to get the correct point)

Definition at line 43 of file Region.C.

References Stash::Coordinate::GetCoordinate(), Stash::Coordinate::GetDirectionVector(), Stash::System::GetDummySystem(), Stash::Coordinate::GetSystem(), and Stash::DirectionVector::RotateAroundPoint().

Referenced by Contains(), and Tools::CircleSegment::GetBound1().

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

- /home/hfm/kosack/Data/SASH-Docs/hess/tools/include/Region.hh
- /home/hfm/kosack/Data/SASH-Docs/hess/tools/src/Region.C
