XipIvOverlay Reference Manual

Generated by Doxygen 1.5.3

Fri Oct 5 14:12:39 2007

Contents

Chapter 1

XipIvOverlay Hierarchical Index

1.1 XipIvOverlay Class Hierarchy

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

Chapter 2

XipIvOverlay Class Index

2.1 XipIvOverlay Class List

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

Chapter 3

XipIvOverlay File Index

3.1 XipIvOverlay File List

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

Chapter 4

XipIvOverlay Class Documentation

4.1 SbXipOverlaySettings Class Reference

`#include <SoXipOverlayElement.h>`

4.1.1 Detailed Description

Container for overlay settings

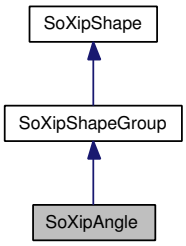
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipOverlayElement.h
- C:/home/gein/xip/src/database/overlay/SoXipOverlayElement.cpp

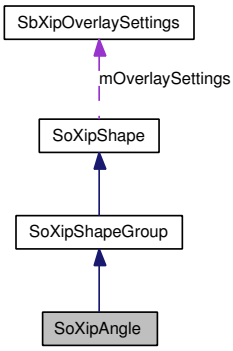
4.2 SoXipAngle Class Reference

`#include <SoXipAngle.h>`

Inheritance diagram for SoXipAngle:



Collaboration diagram for SoXipAngle:



Public Member Functions

- SoXipAngle ()
  Constructor.
- virtual void setRank (int rank)
  set the rank (enumeration) of the shape
- virtual void setCaption (const SbString &caption)
  set the caption (annotation) of the shape

Static Public Member Functions

- static void initClass ()
  Module initialization.

Protected Member Functions

- ~SoXipAngle ()

Destructor.

- virtual void computeAngle ()
  compute the angle given the two lines defined in the shape group

4.2.1 Detailed Description

This node is used to perform an angle measurement in the current view. It is a permanent overlay. User may use multiple instances of this node to perform multiple measures, or use this class as a parameter of the overlay manager (cf. SoXipOverlayManager)

This node combines two line overlays (cf. SoXipLine). The parent class SoXipShapeGroup is used to group the two lines into one single overlay node. These two lines are not yet exposed to the user.

See also:

SoXipOverlayManager
SoXipLine

4.2.2 Member Function Documentation

4.2.2.1 void SoXipAngle::setRank (int *rank*) [virtual]

set the rank (enumeration) of the shape

This method overwrites SoXipShape::setRank() as we want the enumeration to be displayed for the second line only.

Parameters:

rank enumeration

Reimplemented from SoXipShapeGroup.

4.2.2.2 void SoXipAngle::setCaption (const SbString & *caption*) [virtual]

set the caption (annotation) of the shape

This method overwrites SoXipShape::setCaption() as we want the annot to be displayed for the second line only.

Parameters:

caption shape annotation

Reimplemented from SoXipShapeGroup.

4.2.2.3 void SoXipAngle::computeAngle () [protected, virtual]

compute the angle given the two lines defined in the shape group

The computed angle is used as an annotation.

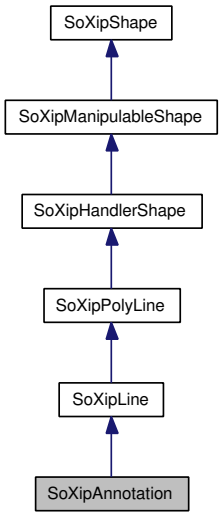
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipAngle.h
- C:/home/gein/xip/src/database/overlay/SoXipAngle.cpp

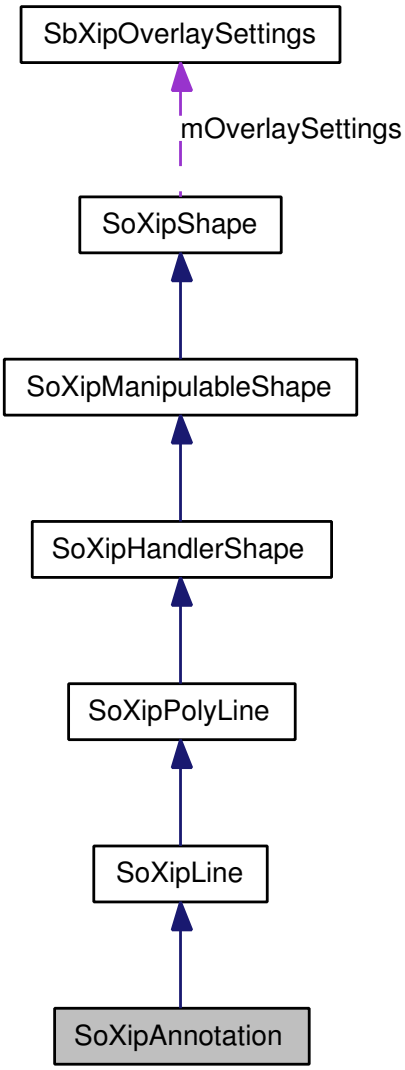
4.3 SoXipAnnotation Class Reference

`#include <SoXipAnnotation.h>`

Inheritance diagram for SoXipAnnotation:



Collaboration diagram for SoXipAnnotation:



Public Member Functions

- SoXipAnnotation ()

  Constructor.

- virtual void setViewTransform (const SbMatrix &viewMatrix)

  Set temporary a transformation.

- virtual void transform (const SbMatrix &matrix)

  Apply the given transformation.

- virtual void updateEnumerationPosition (SoSFVec3f &position)

  Compute the position of the enumeration.

- virtual void updateAnnotationPosition (SoSFVec3f &position)

Compute the position of the shape annotation.

Static Public Member Functions

- static void initClass ()
  Open Inventor class initialization.

Protected Member Functions

- ~SoXipAnnotation ()
  Destructor.
- virtual void beforeCreation ()
  Callback function called before the shape creation.
- virtual void extractLinePoints (SoMFVec3f &points)
  Extract points from the shape characteristics.
- virtual void extractLineSegments (SoMFInt32 &index)
  Extract line indices referring to the shape points.
- virtual void mouseUp (const SbVec3f &pos)
  Send a Mouse Up event to the annotation.
- virtual void getHandlerPoints (SoMFVec3f &points)
  Retrieve handlers point that are used to manipulate the shape.

4.3.1 Detailed Description

This node is used to annotate an object in the current view. It is a permanent overlay. User may use multiple instances of this node to have as many annotations as required, or use this class as a parameter of the overlay manager (cf. SoXipOverlayManager).

See also:

SoXipOverlayManager

4.3.2 Member Function Documentation

4.3.2.1 void SoXipAnnotation::setViewTransform (const SbMatrix & *viewMatrix*) [virtual]

Set temporary a transformation.

Parameters:

matrix transformation matrix

Reimplemented from SoXipManipulableShape.

4.3.2.2 void SoXipAnnotation::transform (const SbMatrix & *matrix*) [virtual]

Apply the given transformation.

Parameters:

matrix transformation matrix

Reimplemented from SoXipPolyLine.

4.3.2.3 void SoXipAnnotation::updateEnumerationPosition (SoSFVec3f & *position*) [virtual]

Compute the position of the enumeration.

Parameters:

position computed position

Reimplemented from SoXipPolyLine.

4.3.2.4 void SoXipAnnotation::updateAnnotationPosition (SoSFVec3f & *position*) [virtual]

Compute the position of the shape annotation.

Parameters:

position computed position

Reimplemented from SoXipPolyLine.

4.3.2.5 void SoXipAnnotation::extractLinePoints (SoMFVec3f & *points*) [protected, virtual]

Extract points from the shape characteristics.

This method is called by the base class to retrieve the segments defining the annotation.

Parameters:

points output array of points

Reimplemented from SoXipPolyLine.

4.3.2.6 void SoXipAnnotation::extractLineSegments (SoMFInt32 & *index*) [protected, virtual]

Extract line indices referring to the shape points.

This method is called by the base class to retrieve the segments defining the annotation.

Parameters:

index output array of indices

Reimplemented from SoXipManipulableShape.

4.3.2.7 void SoXipAnnotation::mouseUp (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Up event to the annotation.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipLine.

4.3.2.8 void SoXipAnnotation::getHandlerPoints (SoMFVec3f & *points*) [protected, virtual]

Retrieve handlers point that are used to manipulate the shape.

This method is called by the base class (SoXipHandlerShape) to retrieve handler points used for manipulation (see SoXipOverlayHandlerManip)

Parameters:

points output array of points

Reimplemented from SoXipHandlerShape.

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

- C:/home/gein/xip/src/database/overlay/SoXipAnnotation.h
- C:/home/gein/xip/src/database/overlay/SoXipAnnotation.cpp

4.4 SoXipBox Class Reference

`#include <SoXipBox.h>`

Public Member Functions

- SoXipBox ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoSFBool on

  Is the tool active ?

- SoSFBool showCenter

  Enable/disable the display of the box center.

- SoMFVec3f point

  Output the box coordinates.

- SoSFVec3f padding

  Padding vector used to create a volume of interest from the ROI.

- SoSFMatrix volumeOfInterest

  Output volume of interest.

Protected Member Functions

- virtual ~SoXipBox ()

  Destructor.

Protected Attributes

- SoGroup * mSeparator

  Box geometries separator.

- SoCoordinate3 * mCoords

  Box coordinates.

- SbPlaneProjector * mPlaneProj

  Plane projector uesd to project mouse coordinates to world coordinates.

- SbViewVolume mViewVolume

  View volume.

- SbViewportRegion mViewport

  Viewport.

- SbVec3f mFirstCoordinates

  Coordinates of the top-left corner of the box.

- SbBool mDrawing

  Is the box being drawn ?

4.4.1 Detailed Description

Tool used to draw a rectangular ROI. The output volume of interest is extracted from this ROI and a user given padding.

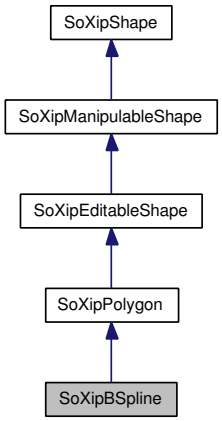
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipBox.h
- C:/home/gein/xip/src/database/overlay/SoXipBox.cpp

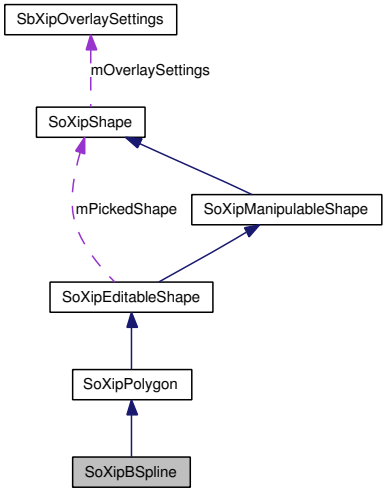
4.5 SoXipBSpline Class Reference

`#include <SoXipBSpline.h>`

Inheritance diagram for SoXipBSpline:



Collaboration diagram for SoXipBSpline:



Public Member Functions

- SoXipBSpline ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoSFShort degree
  Order of the spline.
- SoSFInt32 alpha
  Number of interpolated points between two control points.

Protected Member Functions

- ~SoXipBSpline ()
  Destructor.
- virtual void extractLinePoints (SoMFVec3f &linePoints)
  Extract points from the shape characteristics.
- virtual int getPreviousControlPoint (int id) const
  Map a displayed line index to its preceding control point.

4.5.1 Detailed Description

This node is used to create/ edit a b-spline contour. It is a permanent overlay. User may use multiple instances of this node to perform multiple measures, or use this class as a parameter of the overlay manager (cf. SoXipOverlayManager)

See also:

SoXipOverlayManager

4.5.2 Member Function Documentation

4.5.2.1 void SoXipBSpline::extractLinePoints (SoMFVec3f & *linePoints*) [protected, virtual]

Extract points from the shape characteristics.

This method is called by the base class to retrieve the segments defining the spline.

Parameters:

points output array of points

Reimplemented from SoXipEditableShape.

4.5.2.2 int SoXipBSpline::getPreviousControlPoint (int *id*) const [protected, virtual]

Map a displayed line index to its preceding control point.

This method is called when the b-spline is edited and a control point needs to be added. The editor only knows about the extracted point, not the control points.

Parameters:

id index

Returns:

index of the preceding control point

Reimplemented from SoXipEditableShape.

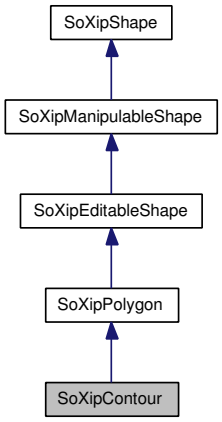
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipBSpline.h
- C:/home/gein/xip/src/database/overlay/SoXipBSpline.cpp

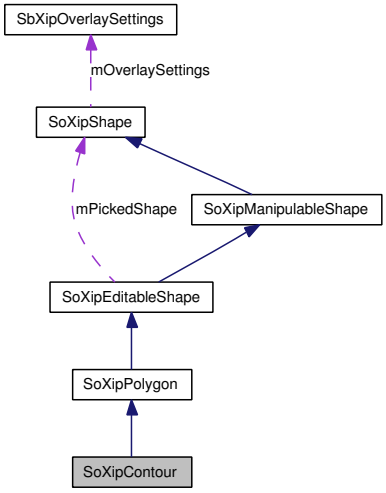
4.6 SoXipContour Class Reference

`#include <SoXipContour.h>`

Inheritance diagram for SoXipContour:



Collaboration diagram for SoXipContour:



Public Member Functions

- SoXipContour ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Protected Member Functions

- ~SoXipContour ()

  Destructor.

- virtual void mouseMove (const SbVec3f &pos)

  Send a Mouse Move event to the ellipse.

- virtual SbBool canClose (const SbVec3f &pos) const

  Can the contour be closed at the given position ?

4.6.1 Detailed Description

This node is used to draw a free-hand contour in the current view. It is a permanent overlay. User may use multiple instances of this node to perform multiple measures, or use this class as a parameter of the overlay manager (cf. SoXipOverlayManager)

Points keep being added to the contour on Mouse Down. On Mouse Click, SoXipContour inherits its behavior from SoXipPolygon.

This overlay can be used as a base class for measurements.

See also:

SoXipOverlayManager

4.6.2 Member Function Documentation

4.6.2.1 void SoXipContour::mouseMove (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Move event to the ellipse.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipPolygon.

4.6.2.2 SbBool SoXipContour::canClose (const SbVec3f & *pos*) const [protected, virtual]

Can the contour be closed at the given position ?

Tell wheter the cursor is close enough from the contour starting point. If so, the user mouse position, gets anchored to the starting point, and the contour can the be closed with a Mouse Click.

Parameters:

pos position

Reimplemented from SoXipPolygon.

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

- C:/home/gein/xip/src/database/overlay/SoXipContour.h
- C:/home/gein/xip/src/database/overlay/SoXipContour.cpp

4.7 SoXipDropShadowElement Class Reference

`#include <SoXipDropShadowElement.h>`

Public Member Functions

- virtual void init (SoState *state)

  Initializes element (no constructor for elements).

- virtual SoElement * copyMatchInfo () const

  Copy element information into a new element.

- virtual SbBool matches (const SoElement *elt) const

  Returns wheter the element matches the given argument.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

- static void set (SoState *state, SoNode *node, SbBool on, const SbColor &color, float transparency, const SbVec2s &pixelOffset)

  Sets the drop shadows properties (offset, color, transparency).

- static SbBool isOn (SoState *state)

  Are the drop shadows activated ?

- static SbColor getColor (SoState *state)

  Get the drop shadows color.

- static float getTransparency (SoState *state)

  Get the drop shadows transparency.

- static SbVec2s getPixelOffset (SoState *state)

  get the offset (in screen coordinates), where the drop shadows should be rendered.

Protected Member Functions

- virtual ~SoXipDropShadowElement ()

  Destructor.

Protected Attributes

- SbBool mOn

  Activation flag.

- SbColor mColor

  Drop shadows color.

- float mTransparency

  Drop shadows transparency.

- SbVec2s mPixelOffset

  Drop shadows offset in screen coordinates.

4.7.1 Detailed Description

This element defines the drop shadows properties for 2D annotations (e.g. SoXipEllipse, SoXipPolygon, etc.) as well as text annotations.

See also:

SoXipDropShadowStyle
SoXipManipulableShape
SoXipText2

4.7.2 Member Function Documentation

4.7.2.1 SbVec2s SoXipDropShadowElement::getPixelOffset (SoState * *state*) [static]

get the offset (in screen coordinates), where the drop shadows should be rendered.

The user can project this offset on the current plane using a SbPlaneProjector to get the offset in world coordinates.

Parameters:

state state of the action

Returns:

a 2D point

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

- C:/home/gein/xip/src/database/overlay/SoXipDropShadowElement.h
- C:/home/gein/xip/src/database/overlay/SoXipDropShadowElement.cpp

4.8 SoXipDropShadowStyle Class Reference

`#include <SoXipDropShadowStyle.h>`

Public Member Functions

- SoXipDropShadowStyle ()
  Constructor.

Static Public Member Functions

- static void initClass ()
  Class initialization.

Public Attributes

- SoSFBool on
  Enable/disable drop shadows.
- SoSFVec2s pixelOffset
  Drop shadows offset in screen coordinates.
- SoSFColor color
  Drop shadows color.
- SoSFFloat transparency
  Drop shadows transparency.

Protected Member Functions

- ~SoXipDropShadowStyle ()
  Destructor.

4.8.1 Detailed Description

This node allows the user to set the properties of drop shadows that affect 2D annotations (text, ROIs, etc.) Those properties are propagated in the scene via a SoXipDropShadowElement

See also:

SoXipDropShadowElement

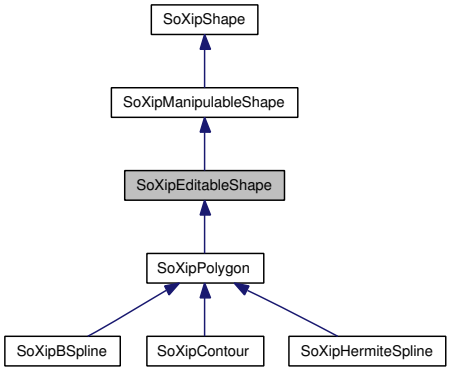
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipDropShadowStyle.h
- C:/home/gein/xip/src/database/overlay/SoXipDropShadowStyle.cpp

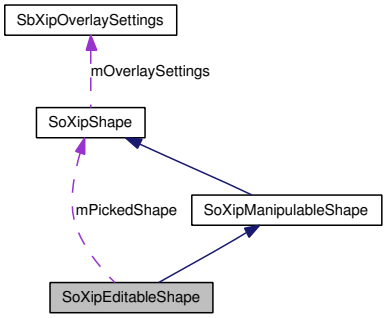
4.9 SoXipEditableShape Class Reference

`#include <SoXipEditableShape.h>`

Inheritance diagram for SoXipEditableShape:



Collaboration diagram for SoXipEditableShape:



Public Member Functions

- SoXipEditableShape ()

  Constructor.

- virtual SbBool isClosed () const

  is the shape opened or closed?

- virtual void updateGeometries ()

  Extract the shape geometries based on the shape characteristics.

- virtual void updateAnnotationPosition (SoSFVec3f &position)

  Compute the position of the annotation.

- virtual void updateEnumerationPosition (SoSFVec3f &position)

  Compute the position of the enumeration.

- virtual void transform (const SbMatrix &matrix)

Apply the given transformation to the editable points.

- virtual int pickControlPoint (SoHandleEventAction *action)

  Pick a control point. Return its id.

- virtual void addPoint (SoNode *node, const SbVec2s &screenPt)

  Add a control point on the editable shape.

- virtual void deletePoint (int id)

  Delete a control point from the editable shape at given position.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoSFBool closeOnDblClick

  should the shape be closed on double click?

Protected Member Functions

- ~SoXipEditableShape ()

  Destructor.

- virtual bool belongsToContour (const SbVec3f &worldPt, int &id)

  return wheter a point belong to the contour or not

- virtual int getPreviousControlPoint (int id) const

  Map a displayed line index to its preceding control point.

- virtual void startEditing ()

  Callback function called before the shape editing.

- virtual void finishEditing ()

  Callback function called after the shape editing.

- virtual void beforeCreation ()

  Callback function called before the shape creation.

- virtual void extractControlPoints (SoMFVec3f &points)

  Extract control points that are used to edit the shape.

- virtual void extractLinePoints (SoMFVec3f &points)

  Extract points from the shape characteristics.

- virtual void extractLineSegments (SoMFInt32 &index)

  Extract line indices referring to the shape points.

Protected Attributes

- SoXipShape * mPickedShape

  Picked shape.

- int mControlPointId

  Id of the control point picked.

4.9.1 Detailed Description

Base class for all editable shapes

4.9.2 Member Function Documentation

4.9.2.1 void SoXipEditableShape::updateAnnotationPosition (SoSFVec3f & *position*) [virtual]

Compute the position of the annotation.

Parameters:

computed position

Reimplemented from SoXipManipulableShape.

4.9.2.2 void SoXipEditableShape::updateEnumerationPosition (SoSFVec3f & *position*) [virtual]

Compute the position of the enumeration.

Parameters:

computed position

Reimplemented from SoXipManipulableShape.

4.9.2.3 int SoXipEditableShape::getPreviousControlPoint (int *id*) const [protected, virtual]

Map a displayed line index to its preceding control point.

This method is called when the shape is edited and a control point needs to be added. The editor only knows about the extracted point, not the control points.

Parameters:

id index

Returns:

index of the preceding control point

Reimplemented in SoXipBSpline, and SoXipHermiteSpline.

4.9.2.4 void SoXipEditableShape::extractControlPoints (SoMFVec3f & *points*) [protected, virtual]

Extract control points that are used to edit the shape.

Parameters:

points output array of points

4.9.2.5 void SoXipEditableShape::extractLinePoints (SoMFVec3f & *points*) [protected, virtual]

Extract points from the shape characteristics.

This method is called by the base class to retrieve the segments defining the shape.

Parameters:

points output array of points

Implements SoXipManipulableShape.

Reimplemented in SoXipBSpline, and SoXipHermiteSpline.

4.9.2.6 void SoXipEditableShape::extractLineSegments (SoMFInt32 & *index*) [protected, virtual]

Extract line indices referring to the shape points.

This method is called by the base class to retrieve the segments defining the shape.

Parameters:

index output array of indices

Reimplemented from SoXipManipulableShape.

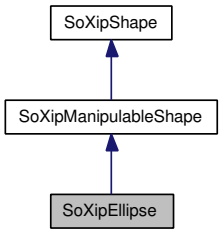
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipEditableShape.h
- C:/home/gein/xip/src/database/overlay/SoXipEditableShape.cpp

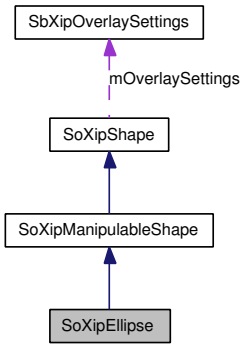
4.10 SoXipEllipse Class Reference

`#include <SoXipEllipse.h>`

Inheritance diagram for SoXipEllipse:



Collaboration diagram for SoXipEllipse:



Public Member Functions

- SoXipEllipse ()

  Constructor.

- virtual SbBool isClosed () const

  Returns wheter the shape is closed or not. Always true for an ellipse.

- virtual void transform (const SbMatrix &matrix)

  Apply the given transformation.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Protected Member Functions

- ~SoXipEllipse ()

Destructor.

- virtual void beforeCreation ()

  Callback function called before the shape creation.

- virtual void extractLinePoints (SoMFVec3f &points)

  Extract points from the ellipse characteristics.

- virtual void extractLineSegments (SoMFInt32 &index)

  Extract line indices referring to the ellipse points.

- virtual void mouseDown (const SbVec3f &pos)

  Send a Mouse Down event to the ellipse.

- virtual void mouseMove (const SbVec3f &pos)

  Send a Mouse Move event to the ellipse.

- virtual void mouseUp (const SbVec3f &pos)

  Send a Mouse Move event to the ellipse.

Protected Attributes

- SbVec3f mFirstPoint
- bool mHasFirstPoint

  Tell wheter the first point has been defined or not.

4.10.1 Detailed Description

This node is used to draw an ellipse in the current view. It is a permanent overlay. User may use multiple instances of this node to perform multiple measures, or use this class as a parameter of the overlay manager (cf. SoXipOverlayManager)

This overlay can be used as a base class for measurements.

See also:

SoXipOverlayManager

4.10.2 Member Function Documentation

4.10.2.1 SbBool SoXipEllipse::isClosed () const [virtual]

Returns wheter the shape is closed or not. Always true for an ellipse.

Returns:

true if the shape is closed, false otherwise

Reimplemented from SoXipManipulableShape.

4.10.2.2 void SoXipEllipse::transform (const SbMatrix & *matrix*) [virtual]

Apply the given transformation.

The transformation is applied to the ellipse's center and the scale and rotation extracted from the matrix are applied to the radius vectors.

Parameters:

matrix transformation matrix

Implements SoXipManipulableShape.

4.10.2.3 void SoXipEllipse::extractLinePoints (SoMFVec3f & *points*) [protected, virtual]

Extract points from the ellipse characteristics.

This method is called by the base class to retrieve the segments defining the ellipse.

Parameters:

points output array of points

Implements SoXipManipulableShape.

4.10.2.4 void SoXipEllipse::extractLineSegments (SoMFInt32 & *index*) [protected, virtual]

Extract line indices referring to the ellipse points.

This method is called by the base class to retrieve the segments defining the ellipse.

Parameters:

index output array of indices

Reimplemented from SoXipManipulableShape.

4.10.2.5 void SoXipEllipse::mouseDown (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Down event to the ellipse.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

4.10.2.6 void SoXipEllipse::mouseMove (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Move event to the ellipse.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

4.10.2.7 void SoXipEllipse::mouseUp (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Move event to the ellipse.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

4.10.3 Member Data Documentation

4.10.3.1 SbVec3f SoXipEllipse::mFirstPoint [protected]

First point defined by the user, used to compute the ellipse's center and radius

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

- C:/home/gein/xip/src/database/overlay/SoXipEllipse.h
- C:/home/gein/xip/src/database/overlay/SoXipEllipse.cpp

4.11 SoXipFontAutoScale Class Reference

`#include <SoXipFontAutoScale.h>`

Public Member Functions

- SoXipFontAutoScale ()
  Constructor.

Static Public Member Functions

- static void initClass ()
  Open Inventor class initialization.

Protected Member Functions

- virtual ~SoXipFontAutoScale ()
  Destructor.

4.11.1 Detailed Description

Automatically sets up the font size that would fit with the viewport size and aspect ratio.

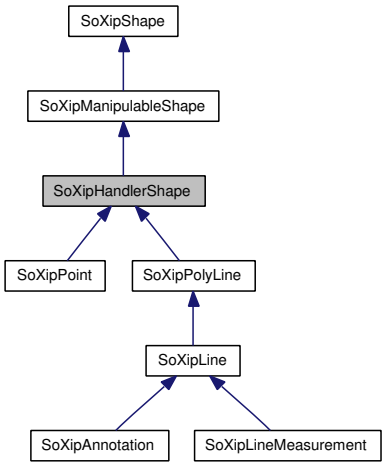
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipFontAutoScale.h
- C:/home/gein/xip/src/database/overlay/SoXipFontAutoScale.cpp

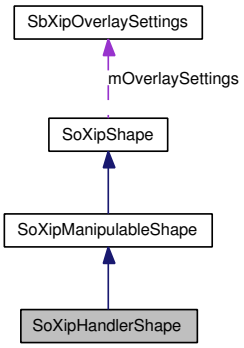
4.12 SoXipHandlerShape Class Reference

`#include <SoXipHandlerShape.h>`

Inheritance diagram for SoXipHandlerShape:



Collaboration diagram for SoXipHandlerShape:



Public Member Functions

- SoXipHandlerShape ()

  Constructor.

- virtual void getHandlerPoints (SoMFVec3f &points_out)

  Get a list of handler points, i.e. points that can be manipulated by a SoXipOverlayHandlerManip.

- virtual void moveHandlerPoint (int id, const SbVec3f &newPos)=0

  Callback triggered when one handler point is being moved by a a SoXipOverlayHandlerManip.

Static Public Member Functions

- static void initClass ()

Class initialization.

Protected Member Functions

- ~SoXipHandlerShape ()

  Destructor.

4.12.1 Detailed Description

Base class for all shapes that'd better be manipulated point by point (SoXipOverlayHandlerManip), rather than with a transform box (SoXipOverlayTransformBoxManip).

See also:

SoXipOverlayHandlerManip

4.12.2 Member Function Documentation

4.12.2.1 void SoXipHandlerShape::getHandlerPoints (SoMFVec3f & *points_out*) [virtual]

Get a list of handler points, i.e. points that can be manipulated by a SoXipOverlayHandlerManip.

Parameters:

points_out list of handler points

Reimplemented in SoXipAnnotation.

4.12.2.2 virtual void SoXipHandlerShape::moveHandlerPoint (int *id*, const SbVec3f & *newPos*) [pure virtual]

Callback triggered when one handler point is being moved by a a SoXipOverlayHandlerManip.

Parameters:

id index of the handler point being moved

newPos new position of the handler point

Implemented in SoXipLineMeasurement, SoXipPoint, and SoXipPolyLine.

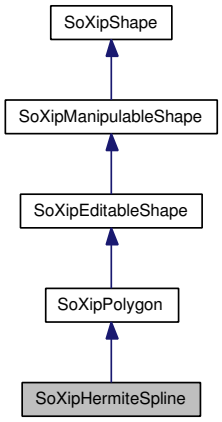
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipHandlerShape.h
- C:/home/gein/xip/src/database/overlay/SoXipHandlerShape.cpp

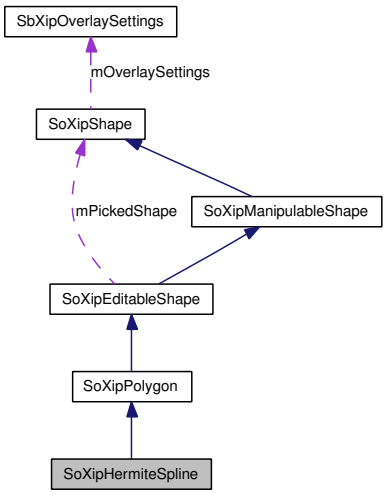
4.13 SoXipHermiteSpline Class Reference

`#include <SoXipHermiteSpline.h>`

Inheritance diagram for SoXipHermiteSpline:



Collaboration diagram for SoXipHermiteSpline:



Public Member Functions

- SoXipHermiteSpline ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoSFInt32 alpha

  Number of interpolated points per segment.

Protected Member Functions

- ~SoXipHermiteSpline ()

  Destructor.

- virtual void extractLinePoints (SoMFVec3f &linePoints)

  Extract points from the shape characteristics.

- virtual int getPreviousControlPoint (int id) const

  Map a displayed line index to its preceding control point.

4.13.1 Detailed Description

This node is used to create/ edit an hermitian spline contour. It is a permanent overlay. User may use multiple instances of this node to perform multiple measures, or use this class as a parameter of the overlay manager (cf. SoXipOverlayManager)

See also:

SoXipOverlayManager

4.13.2 Member Function Documentation

4.13.2.1 void SoXipHermiteSpline::extractLinePoints (SoMFVec3f & *linePoints*) [protected, virtual]

Extract points from the shape characteristics.

This method is called by the base class to retrieve the segments defining the hermitian spline.

Parameters:

points output array of points

Reimplemented from SoXipEditableShape.

4.13.2.2 int SoXipHermiteSpline::getPreviousControlPoint (int *id*) const [protected, virtual]

Map a displayed line index to its preceding control point.

This method is called when the shape is edited and a control point needs to be added. The editor only knows about the extracted point, not the control points.

Parameters:

id index

Returns:

index of the preceding control point

Reimplemented from SoXipEditableShape.

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

- C:/home/gein/xip/src/database/overlay/SoXipHermiteSpline.h
- C:/home/gein/xip/src/database/overlay/SoXipHermiteSpline.cpp

4.14 SoXipIcon Class Reference

`#include <SoXipIcon.h>`

Public Types

- enum IconType
  Type of the icon.

Public Member Functions

- SoXipIcon ()
  Constructor.

Static Public Member Functions

- static void initClass ()
  Open Inventor class initialization.

Public Attributes

- SoSFEnum icon
  Type of the icon displayed.
- SoSFBool isTransparent
  Does the icon texture have a transparent color.
- SoSFColor transparentColor
  Transparent color to be looked up for in the texture.

Protected Member Functions

- ~SoXipIcon ()
  Destructor.
- virtual void fieldChanged (SoField *whichField)
  Force the update of the internal texture.

Protected Attributes

- class SoTexture2 * mTexture
  Internal texture used to render the icon.

4.14.1 Detailed Description

Display a toolbar icon.

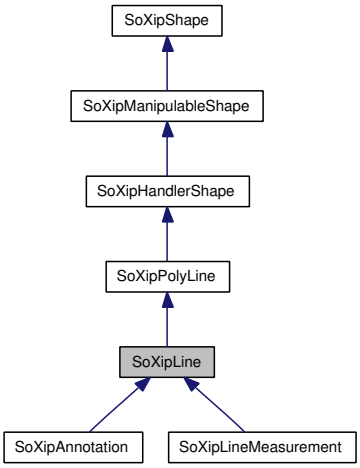
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipIcon.h
- C:/home/gein/xip/src/database/overlay/SoXipIcon.cpp

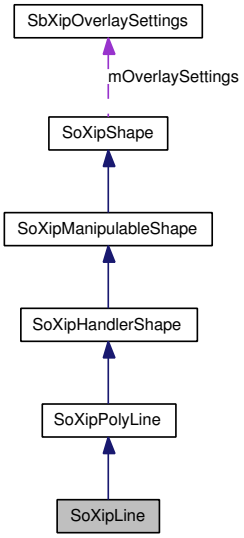
4.15 SoXipLine Class Reference

`#include <SoXipLine.h>`

Inheritance diagram for SoXipLine:



Collaboration diagram for SoXipLine:



Public Member Functions

- SoXipLine ()

  Constructor.

Static Public Member Functions

- static void initClass ()

Module initialization.

Protected Member Functions

- ~SoXipLine ()
  Destructor.
- virtual void mouseDown (const SbVec3f &pos)
  Implementation of the MouseDown event.
- virtual void mouseUp (const SbVec3f &pos)
  Implementation of the MouseUp event.

4.15.1 Detailed Description

This node is used to create a line overlay in the current view. It is a permanent overlay. User may use multiple instances of this node to have as many lines as required, or use this class as a parameter of the overlay manager (cf. SoXipOverlayManager).

See also:

SoXipOverlayManager

4.15.2 Member Function Documentation

4.15.2.1 void SoXipLine::mouseDown (const SbVec3f & *pos*) [protected, virtual]

Implementation of the MouseDown event.

Parameters:

pos projection of the mouse position in 3D.

Reimplemented from SoXipPolyLine.

4.15.2.2 void SoXipLine::mouseUp (const SbVec3f & *pos*) [protected, virtual]

Implementation of the MouseUp event.

Parameters:

pos projection of the mouse position in 3D.

Reimplemented from SoXipPolyLine.

Reimplemented in SoXipAnnotation.

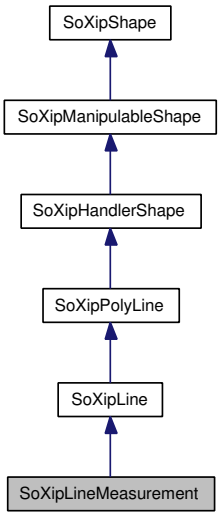
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipLine.h
- C:/home/gein/xip/src/database/overlay/SoXipLine.cpp

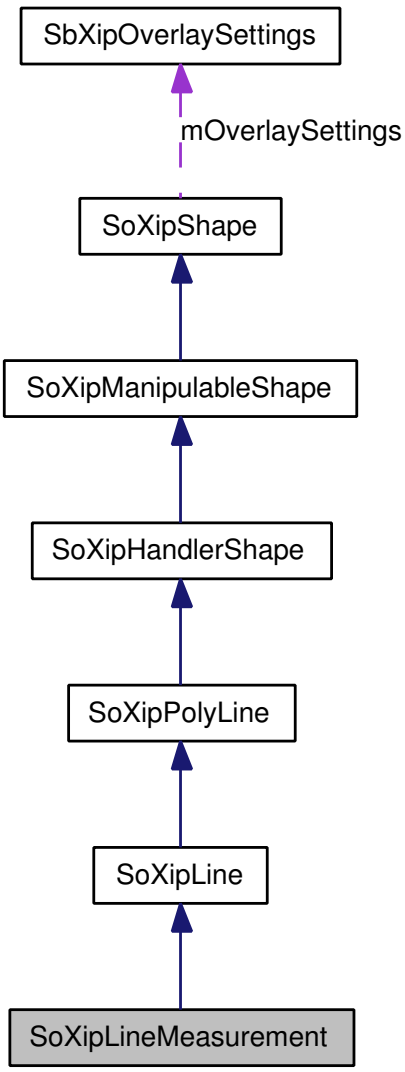
4.16 SoXipLineMeasurement Class Reference

`#include <SoXipLineMeasurement.h>`

Inheritance diagram for SoXipLineMeasurement:



Collaboration diagram for SoXipLineMeasurement:



Public Member Functions

- SoXipLineMeasurement ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Class Initialization.

Protected Member Functions

- ~SoXipLineMeasurement ()
  Destructor.
- virtual void mouseMove (const SbVec3f &pos)
  Implementation of the MouseMove event.
- virtual void moveHandlerPoint (int id, const SbVec3f &newPos)
  Callback triggered when one handler point is being moved by a a SoXipOverlayHandlerManip.

4.16.1 Detailed Description

This node is used to perform a distance measurement in the current view. It is a permanent overlay. User may use multiple instances of this node to have as many distance lines as required, or use this class as a parameter of the overlay manager (cf. SoXipOverlayManager).

See also:

SoXipOverlayManager

4.16.2 Member Function Documentation

4.16.2.1 void SoXipLineMeasurement::mouseMove (const SbVec3f & *pos*) [protected, virtual]

Implementation of the MouseMove event.

Parameters:

pos projection of the mouse position in 3D.

Reimplemented from SoXipPolyLine.

4.16.2.2 void SoXipLineMeasurement::moveHandlerPoint (int *id*, const SbVec3f & *newPos*) [protected, virtual]

Callback triggered when one handler point is being moved by a a SoXipOverlayHandlerManip.

Parameters:

id index of the handler point being moved

newPos new position of the handler point

Reimplemented from SoXipPolyLine.

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

- C:/home/gein/xip/src/database/overlay/SoXipLineMeasurement.h
- C:/home/gein/xip/src/database/overlay/SoXipLineMeasurement.cpp

4.17 SoXipLoadOverlay Class Reference

`#include <SoXipLoadOverlay.h>`

Public Member Functions

- SoXipLoadOverlay ()
  Constructor.

Static Public Member Functions

- static void initClass ()
  Open Inventor class initialization.

Public Attributes

- SoSFString filename
  Filename.
- SoEngineOutput overlays
  Overlays found in the given file.

Protected Member Functions

- ~SoXipLoadOverlay ()
  Destructor.
- virtual void inputChanged (SoField *whichField)
  Engines inputChanged method.
- virtual void evaluate ()
  Engines evaluate method.

4.17.1 Detailed Description

Engine that can load overlays from an external file

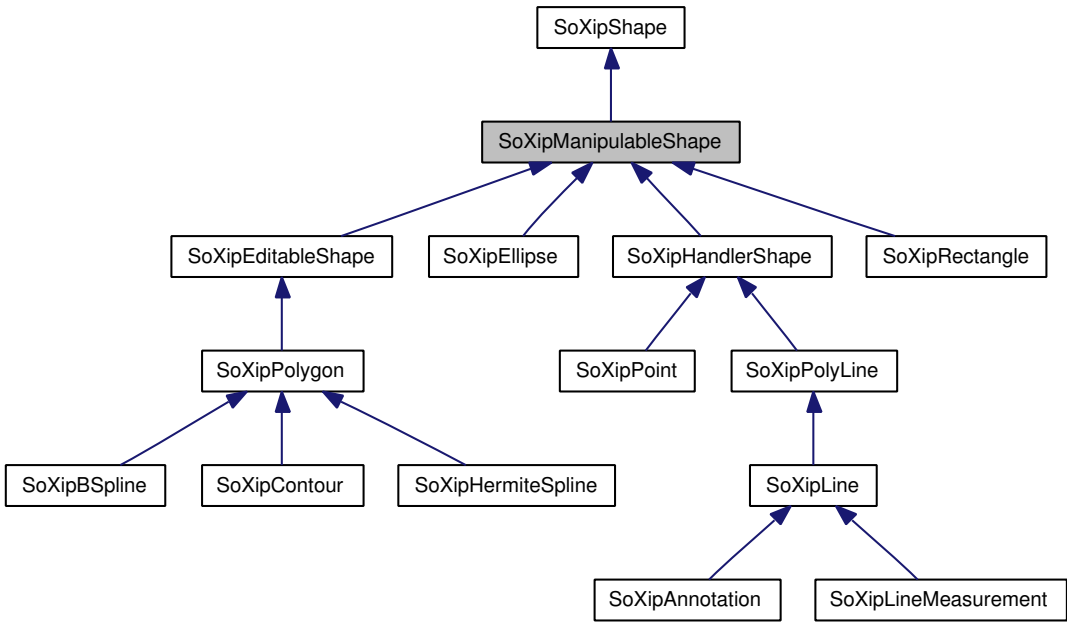
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipLoadOverlay.h
- C:/home/gein/xip/src/database/overlay/SoXipLoadOverlay.cpp

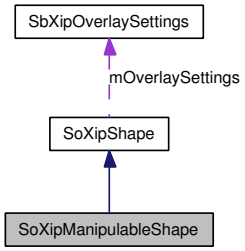
4.18 SoXipManipulableShape Class Reference

`#include <SoXipManipulableShape.h>`

Inheritance diagram for SoXipManipulableShape:



Collaboration diagram for SoXipManipulableShape:



Public Member Functions

- SoXipManipulableShape ()

  Constructor.

- virtual SbBool isButtonPressed () const

  Is the mouse button pressed ?

- virtual SbBool isTextPicked () const

  Has the text been picked ?

- virtual SbBool isTextAnchored () const
- virtual SbBool isClosed () const

Is the shape closed or opened?

- virtual const SbMatrix & getTransform () const

  Get the view transform.

- virtual void setViewTransform (const SbMatrix &viewMatrix)

  Add a transformation matrix in the view before the shape geometries.

- virtual void applyViewTransform (const SbMatrix &viewMatrix)

  Apply the given transformation to the shape geometries.

- virtual void computeXBoundingBox (SbXfBox3f &bbox)

  Compute the shape in-plane bounding box.

- void invalidateGeometries ()

  Invalidate the shape geometries.

- virtual void extractLinePoints (SoMFVec3f &points)=0

  Extract points from the shape characteristics.

- virtual void extractLineSegments (SoMFInt32 &index)

  Extract line indices referring to the shape points.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

- static void invalidateGeometriesCB (void ∗userData, SoSensor ∗)

  Callback for invalidating the shape geometries (e.g a field sensor attached to one of the shape field will be directed here).

Public Attributes

- SoSFBool textAnchored

  Is the text anchored to the shape ?

- SoSFVec3f textPosition

  Position of the text (world coordinates) if not anchored.

Protected Member Functions

- virtual void updateGeometries ()

  Extract the shape geometries based on the shape characteristics.

- virtual void updateViewDependentGeometries ()

update the annotation position and enumeration position (see updateAnnotationPosition and updateEnumerationPosition)

- virtual void updateAnnotationPosition (SoSFVec3f &position)

  Compute the position of the annotation.

- virtual void updateEnumerationPosition (SoSFVec3f &position)

  Compute the position of the enumeration.

- ~SoXipManipulableShape ()

  Destructor.

- virtual void toggleEnumerationOnOff (SbBool flag)

  togle on/off the display of the shape enumeration

- virtual void toggleAnnotationOnOff (SbBool flag)

  togle on/off the display of the shape annotation

- virtual void mouseDown (const SbVec3f &pos)

  Send a Mouse Down event to the shape.

- virtual void mouseMove (const SbVec3f &pos)

  Send a Mouse Move event to the shape.

- virtual void mouseUp (const SbVec3f &pos)

  Send a Mouse Up event to the shape.

- virtual void mouseDouble (const SbVec3f &pos)

  Send a Mouse Double event to the shape.

- virtual void beforeCreation ()

  Callback function called before the shape creation.

- virtual void afterCreation ()

  Callback function called after the shape creation.

Protected Attributes

- SbBool mShowEnumeration

  Should the shape enumeration be displayed ?

- SbBool mShowAnnotation

  Should the shape annotation be displayed ?

- SbBool mUpdateGeometries

  Should the geometries be updated ?

- SbBool mIsViewInitialized

  Is the view initialized ?

- SbBool mIsButtonPressed

  Is the mouse button pressed ?

- SbBool mIsManipulated

  Is the shape currently being manipulated ?

- SbBool mIsTextPicked

  Has the shape annotation been picked ?

- SbBool mIsEnumPicked

  Has the shape enumeration been picked ?

- SbVec2s mMouseDownPos

  Position of the mouse (screen coordinates) when the mouse button was pressed.

- SbTime mMouseDownTime

  Time when the mouse button was pressed.

- SbVec2s mMouseUpPos

  Position of the mouse (screen coordinates) when the mouse button was released.

- SbTime mMouseUpTime

  Time when the mouse button was released.

Static Protected Attributes

- static int mDoubleClickMaxJump = 2
- static SbTime mDoubleClickTime = SbTime(0.5)

  Maximum time between two mouse clicks.

- static SbTime mClickTime = SbTime(0.1)

  Time for one mouse click.

4.18.1 Detailed Description

Base class for all manipulable shapes

4.18.2 Member Function Documentation

4.18.2.1 SbBool SoXipManipulableShape::isTextAnchored () const [virtual]

Is the text anchored (attached) to the shape, or does it have and independent position

4.18.2.2 void SoXipManipulableShape::setViewTransform (const SbMatrix & *viewMatrix*) [virtual]

Add a transformation matrix in the view before the shape geometries.

Parameters:

viewMatrix transformation matrix

Reimplemented in SoXipAnnotation.

4.18.2.3 void SoXipManipulableShape::applyViewTransform (const SbMatrix & *viewMatrix*) [virtual]

Apply the given transformation to the shape geometries.

Parameters:

matrix transformation matrix

4.18.2.4 void SoXipManipulableShape::computeXBoundingBox (SbXfBox3f & *bbox*) [virtual]

Compute the shape in-plane bounding box.

Parameters:

bbox bounding box with attached transformation

4.18.2.5 void SoXipManipulableShape::invalidateGeometries ()

Invalidate the shape geometries.

This will result in a call to updateGeometries in the next rendering loop.

4.18.2.6 void SoXipManipulableShape::updateAnnotationPosition (SoSFVec3f & *position*) [protected, virtual]

Compute the position of the annotation.

Parameters:

computed position

Reimplemented in SoXipAnnotation, SoXipEditableShape, SoXipPoint, and SoXipPolyLine.

4.18.2.7 void SoXipManipulableShape::updateEnumerationPosition (SoSFVec3f & *position*) [protected, virtual]

Compute the position of the enumeration.

Parameters:

computed position

Reimplemented in SoXipAnnotation, SoXipEditableShape, SoXipPoint, and SoXipPolyLine.

4.18.2.8 virtual void SoXipManipulableShape::extractLinePoints (SoMFVec3f & *points*) [pure virtual]

Extract points from the shape characteristics.

This method is called by the base class to retrieve the segments defining the shape.

Parameters:

points output array of points

Implemented in SoXipAnnotation, SoXipBSpline, SoXipEditableShape, SoXipEllipse, SoXipHermiteSpline, SoXipPoint, SoXipPolyLine, and SoXipRectangle.

4.18.2.9 void SoXipManipulableShape::extractLineSegments (SoMFInt32 & *index*) [virtual]

Extract line indices referring to the shape points.

This method is called by the base class to retrieve the segments defining the shape.

Parameters:

index output array of indices

Reimplemented in SoXipAnnotation, SoXipEditableShape, SoXipEllipse, SoXipPoint, and SoXipRectangle.

4.18.2.10 virtual void SoXipManipulableShape::mouseDown (const SbVec3f & *pos*) [inline, protected, virtual]

Send a Mouse Down event to the shape.

Parameters:

pos projection of the mouse position in the view

Reimplemented in SoXipEllipse, SoXipLine, SoXipPoint, SoXipPolygon, SoXipPolyLine, and SoXipRectangle.

4.18.2.11 virtual void SoXipManipulableShape::mouseMove (const SbVec3f & *pos*) [inline, protected, virtual]

Send a Mouse Move event to the shape.

Parameters:

pos projection of the mouse position in the view

Reimplemented in SoXipContour, SoXipEllipse, SoXipLineMeasurement, SoXipPolygon, SoXipPolyLine, and SoXipRectangle.

4.18.2.12 virtual void SoXipManipulableShape::mouseUp (const SbVec3f & *pos*) [inline, protected, virtual]

Send a Mouse Up event to the shape.

Parameters:

pos projection of the mouse position in the view

Reimplemented in SoXipAnnotation, SoXipEllipse, SoXipLine, SoXipPolygon, SoXipPolyLine, and SoXipRectangle.

4.18.2.13 virtual void SoXipManipulableShape::mouseDouble (const SbVec3f & *pos*) [inline, protected, virtual]

Send a Mouse Double event to the shape.

Parameters:

pos projection of the mouse position in the view

Reimplemented in SoXipPolygon, and SoXipPolyLine.

4.18.3 Member Data Documentation

4.18.3.1 int SoXipManipulableShape::mDoubleClickMaxJump = 2 [static, protected]

Distance beyond which two separate mouse clicks are interpreted as independant mouse down events

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

- C:/home/gein/xip/src/database/overlay/SoXipManipulableShape.h
- C:/home/gein/xip/src/database/overlay/SoXipManipulableShape.cpp

4.19 SoXipMeasPixelLens Class Reference

`#include <SoXipMeasPixelLens.h>`

Public Member Functions

- SoXipMeasPixelLens ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoSFBool on

  turn on/off the pixel lens

- SoXipSFDataImage image

  image where to read the intensities from

- SoSFInt32 offset

  apply an offset to the read intensity ?

- SoSFVec3f position

  Current position analyzed by the pixel lens.

Protected Member Functions

- virtual ~SoXipMeasPixelLens ()

  Destructor.

- virtual SbBool getVoxelValue (const SbVec3f &position, float &value)

  Return the voxel value at the given position.

4.19.1 Detailed Description

Draw a line in the current view and display the actual distance

Read the voxel intensity at the location pointed at by the mouse. This coordinate node will also push an empty coordinate vector in the traversal state. Typically needed for seed points etc.

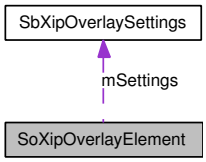
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipMeasPixelLens.h
- C:/home/gein/xip/src/database/overlay/SoXipMeasPixelLens.cpp

4.20 SoXipOverlayElement Class Reference

`#include <SoXipOverlayElement.h>`

Collaboration diagram for SoXipOverlayElement:



Public Types

- enum LocationType

  Location (where to get the overlay points).

Public Member Functions

- virtual void init (SoState *state)

  Elements init method.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

- static void set (SoState *state, SoNode *node, SbXipOverlaySettings settings)

  Set the overlay settings.

Protected Member Functions

- virtual ~SoXipOverlayElement ()

  Destructor.

Protected Attributes

- SbXipOverlaySettings mSettings

  Overlay settings.

4.20.1 Detailed Description

Set the color of one or more overlay object that have the specified label

This element is used to associate an overlay label with a color/ alpha. This comes really handy when you need to set the color of an overlay nested in an overlay manager

This element contains some overlay properties that apply to all the overlays in the scene graph where the element is set.

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

- C:/home/gein/xip/src/database/overlay/SoXipOverlayElement.h
- C:/home/gein/xip/src/database/overlay/SoXipOverlayElement.cpp

4.21 SoXipOverlayExtractContour Class Reference

`#include <SoXipOverlayExtractContour.h>`

Public Member Functions

- SoXipOverlayExtractContour ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoMFNode overlays

  List of overlays.

- SoSFTrigger extract

  Extract the contour information.

- SoEngineOutput point

  Contour points.

- SoEngineOutput coordIndex

  Contour indices.

Protected Member Functions

- ~SoXipOverlayExtractContour ()

  Destructor.

- virtual void inputChanged (SoField *whichField)

  Engines inputChanged method.

- virtual void evaluate ()

  Engines evaluate method.

- virtual void appendContour (class SoXipManipulableShape *shape, SoMFVec3f &accumulatePoint, SoMFInt32 &accumulateCoordIndex)

Protected Attributes

- bool m_doExtract

  Shoudl the extraction be performed?

4.21.1 Detailed Description

Extract a list of contour, defined by a list of point and a list of point index, from a list of overlay. Each contour is separated by -1 in the list of index.

4.21.2 Member Function Documentation

4.21.2.1 void SoXipOverlayExtractContour::appendContour (class SoXipManipulableShape * *shape*, SoMFVec3f & *accumulatePoint*, SoMFInt32 & *accumulateCoordIndex*) [protected, virtual]

Append the contour points and indices of one shape to a point and index accumulation buffers

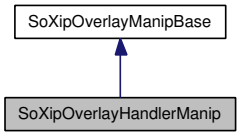
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipOverlayExtractContour.h
- C:/home/gein/xip/src/database/overlay/SoXipOverlayExtractContour.cpp

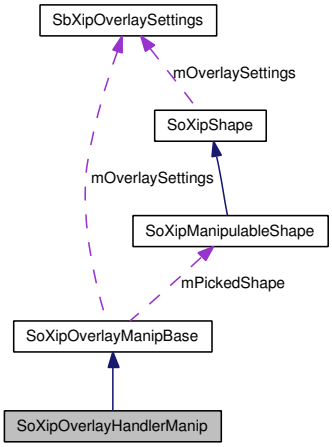
4.22 SoXipOverlayHandlerManip Class Reference

`#include <SoXipOverlayHandlerManip.h>`

Inheritance diagram for SoXipOverlayHandlerManip:



Collaboration diagram for SoXipOverlayHandlerManip:



Public Types

- enum ModeType

  Mode type.

Public Member Functions

- SoXipOverlayHandlerManip ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoSFEnum mode

  mode

Protected Member Functions

- ~SoXipOverlayHandlerManip ()

  Destructor.

- virtual void create ()

  Create the manipulator geometries.

- virtual int pickControlPoint (SoHandleEventAction *action)

  Return the id of the picked handler point, -1 if none.

Protected Attributes

- int mControlPointId

  id of the picked handler point

4.22.1 Detailed Description

Manipulator that can manipulate one (and only one) instance of SoXipHandlerShape derived shape. This manipulator creates an handler point for each point returned by SoXipHandlerShape::getHandlerPoints(). Those handlers can be dragged to set a new position

See also:

SoXipHandlerShape

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

- C:/home/gein/xip/src/database/overlay/SoXipOverlayHandlerManip.h
- C:/home/gein/xip/src/database/overlay/SoXipOverlayHandlerManip.cpp

4.23 SoXipOverlayManager Class Reference

`#include <SoXipOverlayManager.h>`

Public Member Functions

- SoXipOverlayManager ()

  Constructor.

- SoXipShape * createNewShape ()

  Create a new shape given shapeClassName, shapeLabel and shapeCaption.

- virtual class SoXipShapeList * getCurrentList ()
- virtual int getListNumChildren () const

  Get the shape list number of children.

- virtual SoNode * getListChild (int position) const

  Get the shape list child at given position.

- virtual void insertListChild (int position, SoNode *node)

  Insert node into the shape list at given position.

- virtual void removeListChild (SoNode *node)

  Remove node from the shape list.

- virtual void removeListChild (int position)

  Remove the child at given position from the shape list.

- virtual void removeAllListChildren ()

  Remove all the shape list children.

- virtual int findListChild (SoNode *node)

  Find a node in the shape list and returns its position. Returns -1 if not found.

Static Public Member Functions

- static void initClass ()

  Class initialization.

Public Attributes

- SoSFBool menuEnabled

  Should a context menu be used.

- SoSFBool create

  Is the manager in creation mode.

- SoSFString shapeClassName

  Class name of the next shape to be created.

- SoSFString shapeLabel

  Label of the next shape to be created.

- SoSFString shapeCaption

  Caption of the next shape to be created.

- SoSFBool multipleShapes

  Single or multiple shapes.

Protected Member Functions

- ~SoXipOverlayManager ()

  Destructor.

- virtual int getNextRank ()
- virtual void initContextMenu ()

  Create the context menu.

- virtual SbBool setUpConnections (SbBool onOff, SbBool doItAlways=FALSE)

  Set up the sensors on internal nodes.

- virtual void handleEvent (SoHandleEventAction *action)

  Event manager.

- SoXipShape * isShapePicked (SoHandleEventAction *action)

  Returns the picked shape, null if none were picked.

- virtual void onMenuItemClick ()

  Call when a menu item is clicked.

- virtual void onMenuEnable ()

  Call when the menu is enabled.

- virtual void onCopy ()

  Call from the menu or when the user press CTRL+C.

- virtual void onPaste ()

  Call from the menu or when the user press CTRL+V.

- virtual void onDelete ()

  Call from the menu or when the user press CTRL+X.

Protected Attributes

- SoNodeList mSelection

  Current overlay selection.

4.23.1 Detailed Description

This node is used to handle multiple overlay objects, of different type.

4.23.2 Member Function Documentation

4.23.2.1 SoXipShapeList * SoXipOverlayManager::getCurrentList () [virtual]

Internal. Get a pointer to the internal list of shape.

4.23.2.2 int SoXipOverlayManager::getNextRank () [protected, virtual]

Compute the next rank available. Each overlay is enumerated starting from 1. Return the smallest available index.

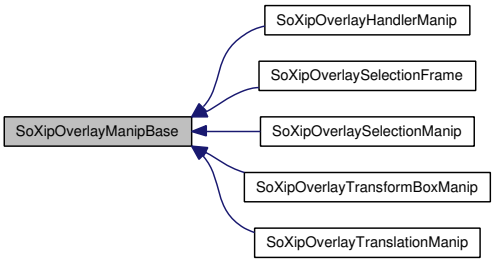
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipOverlayManager.h
- C:/home/gein/xip/src/database/overlay/SoXipOverlayManager.cpp

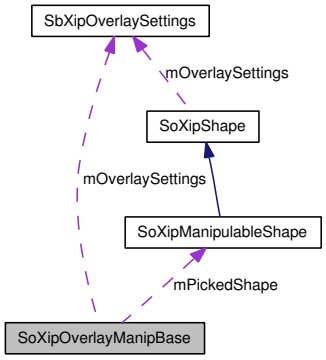
4.24 SoXipOverlayManipBase Class Reference

`#include <SoXipOverlayManipBase.h>`

Inheritance diagram for SoXipOverlayManipBase:



Collaboration diagram for SoXipOverlayManipBase:



Public Member Functions

- SoXipOverlayManipBase ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoSFBool on

  turn on/off the manipulator

- SoSFInt32 numNodesUpToContainer

Protected Member Functions

- ~SoXipOverlayManipBase ()

  Destructor.

4.24.1 Detailed Description

Base class for all overlay manipulators

4.24.2 Member Data Documentation

4.24.2.1 SoSFInt32 SoXipOverlayManipBase::numNodesUpToContainer

Number of nodes upward where overlay manipulators can start searching for overlays.

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

- C:/home/gein/xip/src/database/overlay/SoXipOverlayManipBase.h
- C:/home/gein/xip/src/database/overlay/SoXipOverlayManipBase.cpp

4.25 SoXipOverlayManips Class Reference

`#include <SoXipOverlayManips.h>`

Public Member Functions

- SoXipOverlayManips ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoSFShort numNodesUpToContainer
- SoSFBool multipleSelection

  activate/ disactivate the selection frame manipulator

Protected Member Functions

- ~SoXipOverlayManips ()

  Destructor.

4.25.1 Detailed Description

Kit that groups all the Xip overlay manipulators. As the manipulators all deal with events (mouse interaction), the order matters a lot. This kit makes easy to use overlay manipulators.

4.25.2 Member Data Documentation

4.25.2.1 SoSFShort SoXipOverlayManips::numNodesUpToContainer

Number of nodes upward where overlay manipulators can start searching for overlays.

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

- C:/home/gein/xip/src/database/overlay/SoXipOverlayManips.h
- C:/home/gein/xip/src/database/overlay/SoXipOverlayManips.cpp

4.26 SoXipOverlayManipulatedElement Class Reference

`#include <SoXipOverlayManipulatedElement.h>`

Public Member Functions

- virtual void init (SoState *state)

  Elements init method.

Static Public Member Functions

- static void initClass ()

  Constructor.

Protected Member Functions

- virtual ~SoXipOverlayManipulatedElement ()

  Destructor.

Protected Attributes

- SbBool mIsManipulated

  Have the overlays already been manipulated?

4.26.1 Detailed Description

Element that tells wheter the selected overlays have already been manipulated or not.

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

- C:/home/gein/xip/src/database/overlay/SoXipOverlayManipulatedElement.h
- C:/home/gein/xip/src/database/overlay/SoXipOverlayManipulatedElement.cpp

4.27 SoXipOverlaySearchContour Class Reference

`#include <SoXipOverlaySearchContour.h>`

Public Member Functions

- SoXipOverlaySearchContour ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoSFShort numNodesUpToContainer

  number of nodes upward where to start the search action

- SoSFString label

  shapes should match given label, if not ""

- SoSFString type

  shapes class name should match given type, if not ""

- SoSFBool closed

  shoudl the shapes be closed ?

- SoSFBool selected

  should the shapes be selected ?

- SoSFBool searchAll

  should the search action goes trough kits and groups ?

- SoSFTrigger search

  perform the search action

- SoMFVec3f point

  list of point extracted from matching shapes

- SoMFInt32 coordIndex

Protected Member Functions

- ~SoXipOverlaySearchContour ()

  Destructor.

- virtual SoNodeList selectMatches (const SoNodeList &shapes)

  Select the matching shapes from a list of nodes.

- virtual void appendContour (class SoXipManipulableShape *shape, SoMFVec3f &accumulatePoint, SoMFInt32 &accumulateCoordIndex)

Protected Attributes

- SbBool m_doSearch

  Should the search be performed ?

4.27.1 Detailed Description

Overlay search engine. Retrieve a list of overlays from the specified scene graph and given some search criteria.

Contour search engine. Retrieve a list of point and indices from the specified scene graph and given some search criteria. This is equivalent to SoXipOverlaySearchContour followed by SoXipOverlayExtractContour.

See also:

SoXipOverlaySearchContour
SoXipOverlayExtractContour

4.27.2 Member Function Documentation

4.27.2.1 void SoXipOverlaySearchContour::appendContour (class SoXipManipulableShape * *shape*, SoMFVec3f & *accumulatePoint*, SoMFInt32 & *accumulateCoordIndex*) [protected, virtual]

Append the geometries of a shape to a point and index accumulation buffers

4.27.3 Member Data Documentation

4.27.3.1 SoMFInt32 SoXipOverlaySearchContour::coordIndex

list of point indices, defining multiple contours, extracted from matching shapes

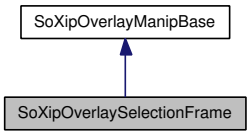
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipOverlaySearchContour.h
- C:/home/gein/xip/src/database/overlay/SoXipOverlaySearchContour.cpp

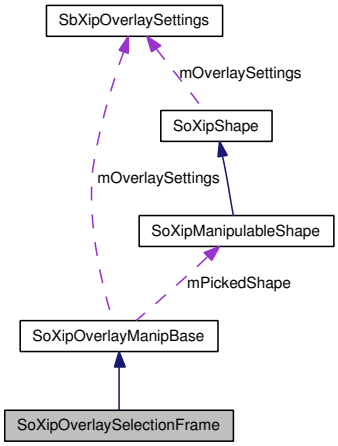
4.28 SoXipOverlaySelectionFrame Class Reference

`#include <SoXipOverlaySelectionFrame.h>`

Inheritance diagram for SoXipOverlaySelectionFrame:



Collaboration diagram for SoXipOverlaySelectionFrame:



Public Member Functions

- SoXipOverlaySelectionFrame ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Protected Member Functions

- ~SoXipOverlaySelectionFrame ()

  Destructor.

- virtual void create ()

  Create the manipulator geometries (semi-transparent box).

- virtual void updateSelection ()

Retrieve the selected shapes and update their selection status.

Protected Attributes

- SbBool mIsDragging

  Is the user dragging ?

4.28.1 Detailed Description

Multiple selection tool. User can draws a semi-transparent box to select multiple shapes at once.

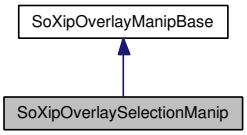
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipOverlaySelectionFrame.h
- C:/home/gein/xip/src/database/overlay/SoXipOverlaySelectionFrame.cpp

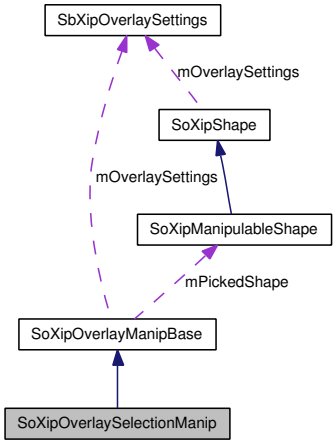
4.29 SoXipOverlaySelectionManip Class Reference

`#include <SoXipOverlaySelectionManip.h>`

Inheritance diagram for SoXipOverlaySelectionManip:



Collaboration diagram for SoXipOverlaySelectionManip:



Public Member Functions

- SoXipOverlaySelectionManip ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Protected Member Functions

- ~SoXipOverlaySelectionManip ()

  Destructor.

- virtual void updateShapesStatus (SoHandleEventAction *action, SbBool unselect=TRUE, SbBool finishEditing=TRUE)

4.29.1 Detailed Description

This class takes care of the shape selection. The Ctrl key can be pressed to add/ remove shape from the selection. As a shape needs to be selected to be translated, the selection manip cannot handle the action (trick).

4.29.2 Member Function Documentation

4.29.2.1 void SoXipOverlaySelectionManip::updateShapesStatus (SoHandleEventAction * *action*, SbBool *unselect* = TRUE, SbBool *finishEditing* = TRUE) [protected, virtual]

Update the selection state of all the shapes found in the scene graph (up to numNodesUpToContainer).

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

- C:/home/gein/xip/src/database/overlay/SoXipOverlaySelectionManip.h
- C:/home/gein/xip/src/database/overlay/SoXipOverlaySelectionManip.cpp

4.30 SoXipOverlaySettings Class Reference

`#include <SoXipOverlaySettings.h>`

Public Types

- enum LocationType
  Location: where to get the points from ?

Public Member Functions

- SoXipOverlaySettings ()
  Constructor.

Static Public Member Functions

- static void initClass ()
  Open Inventor class initialization.

Public Attributes

- SoSFEnum location
  Location.
- SoSFBool showAnnotation
  Should the annotation be displayed ?
- SoSFBool showEnumeration
  Should the enumeration be displayed ?

Protected Member Functions

- ~SoXipOverlaySettings ()
  Destructor.

4.30.1 Detailed Description

Sets some properties for all the overlay

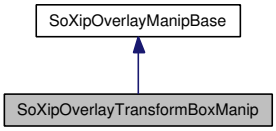
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipOverlaySettings.h
- C:/home/gein/xip/src/database/overlay/SoXipOverlaySettings.cpp

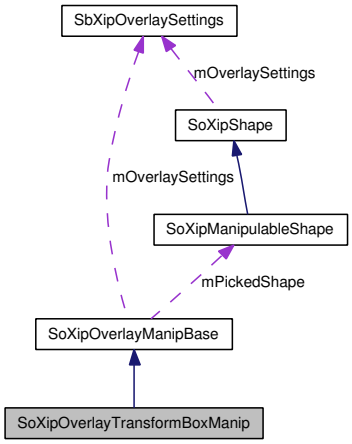
4.31 SoXipOverlayTransformBoxManip Class Reference

`#include <SoXipOverlayTransformBoxManip.h>`

Inheritance diagram for SoXipOverlayTransformBoxManip:



Collaboration diagram for SoXipOverlayTransformBoxManip:



Public Types

- enum ModeType

  Mode of the manipulator.

Public Member Functions

- SoXipOverlayTransformBoxManip ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoSFEnum mode

mode of the manipulator

Protected Member Functions

- ~SoXipOverlayTransformBoxManip ()

  Destructor.

- virtual void create ()

  Create the manipulator geometries (handlers).

- virtual void scale (SoHandleEventAction *action)

  Scale the selected shapes given the last user mouse interactions.

- virtual void rotate (SoHandleEventAction *action)

  Rotate the selected shapes given the last user mouse interactions.

- virtual int pickControlPoint (SoHandleEventAction *action)

  Return the id of the picked handler point, -1 if none were picked.

- virtual SbBool computeSelectionXBoundingBox (SoGLRenderAction *action)

  Compute the in-plane bouding box of the selected shapes.

Protected Attributes

- int mControlPointId

  Id of the lastly picked handler point.

- SbXfBox3f mXBoundingBox

  In-plane bouding box of the selected shapes.

4.31.1 Detailed Description

Compute the in-plane bounding box of the current selection, and display handler points in the 8 corners. Those handlers can be used to apply a scaling to the selected shapes when no key specifiers are pressed. When shift is pressed the 4 corners can be used to apply a rotation around the normal axis.

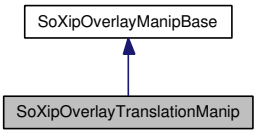
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipOverlayTransformBoxManip.h
- C:/home/gein/xip/src/database/overlay/SoXipOverlayTransformBoxManip.cpp

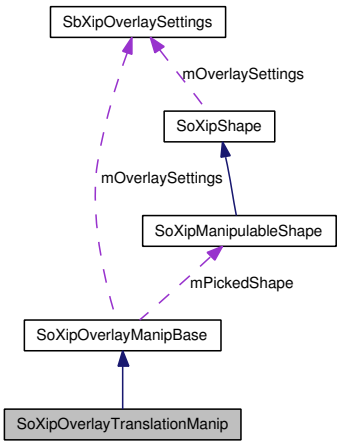
4.32 SoXipOverlayTranslationManip Class Reference

`#include <SoXipOverlayTranslationManip.h>`

Inheritance diagram for SoXipOverlayTranslationManip:



Collaboration diagram for SoXipOverlayTranslationManip:



Public Member Functions

- SoXipOverlayTranslationManip ()
  Constructor.

Static Public Member Functions

- static void initClass ()
  Open Inventor class initialization.

Protected Member Functions

- ~SoXipOverlayTranslationManip ()
  Destructor.

Protected Attributes

- SbBool mCanTranslate

Can translate ?

4.32.1 Detailed Description

Implementation of a translation manipulator. When mouse cursor approaches a shape contour, the manipulator gets activated. If current selection has multiple shapes, then the translation is applied to all.

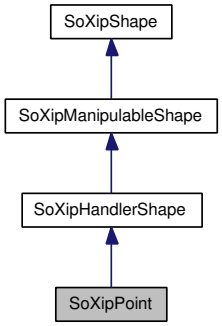
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipOverlayTranslationManip.h
- C:/home/gein/xip/src/database/overlay/SoXipOverlayTranslationManip.cpp

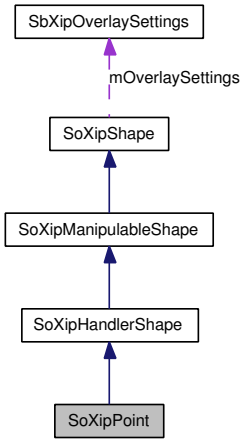
4.33 SoXipPoint Class Reference

`#include <SoXipPoint.h>`

Inheritance diagram for SoXipPoint:



Collaboration diagram for SoXipPoint:



Public Member Functions

- SoXipPoint ()

  Constructor.

- virtual void extractLinePoints (SoMFVec3f &points)

  Extract points from the shape characteristics.

- virtual void extractLineSegments (SoMFInt32 &index)

  Extract line indices referring to the shape points.

- virtual void moveHandlerPoint (int id, const SbVec3f &newPos)

  Callback triggered when one handler point is being moved by a a SoXipOverlayHandlerManip.

- virtual void transform (const SbMatrix &matrix)

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Protected Member Functions

- ~SoXipPoint ()

  Destructor.

- virtual void mouseDown (const SbVec3f &pos)

  Send a Mouse Down event to the shape.

- virtual void beforeCreation ()

  Callback function called before the shape creation.

- virtual void updateAnnotationPosition (SoSFVec3f &position)

  Compute the position of the annotation.

- virtual void updateEnumerationPosition (SoSFVec3f &position)

  Compute the position of the enumeration.

4.33.1 Detailed Description

This node is used to create a point. It is a permanent overlay. User may use multiple instances of this node to perform multiple measures, or use this class as a parameter of the overlay manager (cf. SoXipOverlay-Manager)

See also:

SoXipOverlayManager

4.33.2 Member Function Documentation

4.33.2.1 void SoXipPoint::extractLinePoints (SoMFVec3f & *points*) [virtual]

Extract points from the shape characteristics.

This method is called by the base class to retrieve the segments defining the shape.

Parameters:

points output array of points

Implements SoXipManipulableShape.

4.33.2.2 void SoXipPoint::extractLineSegments (SoMFInt32 & *index*) [virtual]

Extract line indices referring to the shape points.

This method is called by the base class to retrieve the segments defining the shape.

Parameters:

index output array of indices

Reimplemented from SoXipManipulableShape.

4.33.2.3 void SoXipPoint::moveHandlerPoint (int *id*, const SbVec3f & *newPos*) [virtual]

Callback triggered when one handler point is being moved by a a SoXipOverlayHandlerManip.

Parameters:

id index of the handler point being moved

newPos new position of the handler point

Implements SoXipHandlerShape.

4.33.2.4 void SoXipPoint::transform (const SbMatrix & *matrix*) [virtual]

Transform the point by the given transformation matrix

Implements SoXipManipulableShape.

4.33.2.5 void SoXipPoint::mouseDown (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Down event to the shape.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

4.33.2.6 void SoXipPoint::updateAnnotationPosition (SoSFVec3f & *position*) [protected, virtual]

Compute the position of the annotation.

Parameters:

computed position

Reimplemented from SoXipManipulableShape.

4.33.2.7 void SoXipPoint::updateEnumerationPosition (SoSFVec3f & *position*) [protected, virtual]

Compute the position of the enumeration.

Parameters:

computed position

Reimplemented from SoXipManipulableShape.

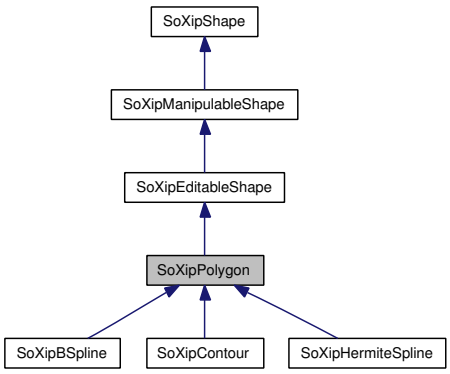
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipPoint.h
- C:/home/gein/xip/src/database/overlay/SoXipPoint.cpp

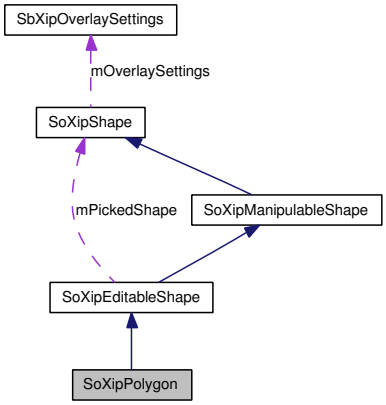
4.34 SoXipPolygon Class Reference

`#include <SoXipPolygon.h>`

Inheritance diagram for SoXipPolygon:



Collaboration diagram for SoXipPolygon:



Public Member Functions

- SoXipPolygon ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Protected Member Functions

- ~SoXipPolygon ()

Destructor.

- virtual void mouseDown (const SbVec3f &pos)
  Send a Mouse Down event to the shape.
- virtual void mouseMove (const SbVec3f &pos)
  Send a Mouse Move event to the shape.
- virtual void mouseUp (const SbVec3f &pos)
  Send a Mouse Up event to the shape.
- virtual void mouseDouble (const SbVec3f &pos)
  Send a Mouse Double event to the shape.
- virtual SbBool canClose (const SbVec3f &pos) const
  Can the polygon be closed at the given position ?
- virtual SbBool isConsistent () const
  Is the polygon consistent ?

4.34.1 Detailed Description

This node is used to create a polygon. It is a permanent overlay. User may use multiple instances of this node to perform multiple measures, or use this class as a parameter of the overlay manager (cf. SoXipOverlayManager)

See also:

SoXipOverlayManager

4.34.2 Member Function Documentation

4.34.2.1 void SoXipPolygon::mouseDown (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Down event to the shape.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

4.34.2.2 void SoXipPolygon::mouseMove (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Move event to the shape.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

Reimplemented in SoXipContour.

4.34.2.3 void SoXipPolygon::mouseUp (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Up event to the shape.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

4.34.2.4 void SoXipPolygon::mouseDouble (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Double event to the shape.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

4.34.2.5 SbBool SoXipPolygon::canClose (const SbVec3f & *pos*) const [protected, virtual]

Can the polygon be closed at the given position ?

Tell wheter the cursor is close enough from the contour starting point. If so, the user mouse position, gets anchored to the starting point, and the contour can the be closed with a Mouse Click.

Parameters:

pos position

Reimplemented in SoXipContour.

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

- C:/home/gein/xip/src/database/overlay/SoXipPolygon.h
- C:/home/gein/xip/src/database/overlay/SoXipPolygon.cpp

4.35 SoXipPolygonArea Class Reference

`#include <SoXipPolygonArea.h>`

Public Member Functions

- SoXipPolygonArea ()
  Constructor.

Static Public Member Functions

- static void initClass ()
  Open Inventor class initialization.

Public Attributes

- SoMFVec3f point
  List of point.
- SoMFInt32 coordIndex
  List of point indices defining the polygon.
- SoEngineOutput area
  Area corresponding to the given polygon.

Protected Member Functions

- ~SoXipPolygonArea ()
  Destructor.
- virtual void evaluate ()
  Engine evaluate method.

4.35.1 Detailed Description

This engine computes the area of an arbitrary polygon

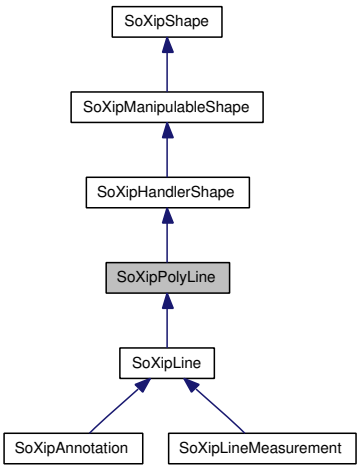
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipPolygonArea.h
- C:/home/gein/xip/src/database/overlay/SoXipPolygonArea.cpp

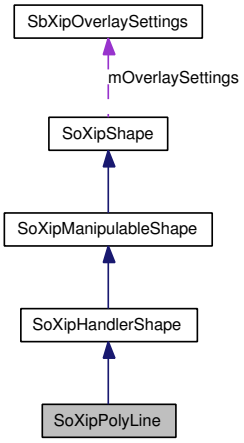
4.36 SoXipPolyLine Class Reference

`#include <SoXipPolyLine.h>`

Inheritance diagram for SoXipPolyLine:



Collaboration diagram for SoXipPolyLine:



Public Member Functions

- SoXipPolyLine ()

  Constructor.

- virtual void extractLinePoints (SoMFVec3f &linePoints)

  Extract points from the shape characteristics.

- virtual void moveHandlerPoint (int id, const SbVec3f &newPos)

  Callback triggered when one handler point is being moved by a a SoXipOverlayHandlerManip.

- virtual void transform (const SbMatrix &matrix)

Apply the given transformation.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Protected Member Functions

- ~SoXipPolyLine ()

  Destructor.

- virtual void mouseDown (const SbVec3f &pos)

  Send a Mouse Down event to the shape.

- virtual void mouseMove (const SbVec3f &pos)

  Send a Mouse Move event to the shape.

- virtual void mouseUp (const SbVec3f &pos)

  Send a Mouse Up event to the shape.

- virtual void mouseDouble (const SbVec3f &pos)

  Send a Mouse Double event to the shape.

- virtual void beforeCreation ()

  Callback function called before the shape creation.

- virtual void updateAnnotationPosition (SoSFVec3f &position)

  Compute the position of the annotation.

- virtual void updateEnumerationPosition (SoSFVec3f &position)

  Compute the position of the enumeration.

4.36.1 Detailed Description

This node is used to create a set of connected lines. It is a permanent overlay. User may use multiple instances of this node to perform multiple measures, or use this class as a parameter of the overlay manager (cf. SoXipOverlayManager)

See also:

SoXipOverlayManager

4.36.2 Member Function Documentation

4.36.2.1 void SoXipPolyLine::extractLinePoints (SoMFVec3f & *linePoints*) [virtual]

Extract points from the shape characteristics.

This method is called by the base class to retrieve the segments defining the polyline.

Parameters:

points output array of points

Implements SoXipManipulableShape.

Reimplemented in SoXipAnnotation.

4.36.2.2 void SoXipPolyLine::moveHandlerPoint (int *id*, const SbVec3f & *newPos*) [virtual]

Callback triggered when one handler point is being moved by a a SoXipOverlayHandlerManip.

Parameters:

id index of the handler point being moved

newPos new position of the handler point

Implements SoXipHandlerShape.

Reimplemented in SoXipLineMeasurement.

4.36.2.3 void SoXipPolyLine::transform (const SbMatrix & *matrix*) [virtual]

Apply the given transformation.

Parameters:

matrix transformation matrix

Implements SoXipManipulableShape.

Reimplemented in SoXipAnnotation.

4.36.2.4 void SoXipPolyLine::mouseDown (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Down event to the shape.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

Reimplemented in SoXipLine.

4.36.2.5 void SoXipPolyLine::mouseMove (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Move event to the shape.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

Reimplemented in SoXipLineMeasurement.

4.36.2.6 void SoXipPolyLine::mouseUp (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Up event to the shape.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

Reimplemented in SoXipAnnotation, and SoXipLine.

4.36.2.7 void SoXipPolyLine::mouseDouble (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Double event to the shape.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

4.36.2.8 void SoXipPolyLine::updateAnnotationPosition (SoSFVec3f & *position*) [protected, virtual]

Compute the position of the annotation.

Parameters:

computed position

Reimplemented from SoXipManipulableShape.

Reimplemented in SoXipAnnotation.

4.36.2.9 void SoXipPolyLine::updateEnumerationPosition (SoSFVec3f & *position*) [protected, virtual]

Compute the position of the enumeration.

Parameters:

computed position

Reimplemented from SoXipManipulableShape.

Reimplemented in SoXipAnnotation.

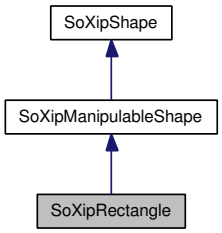
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipPolyLine.h
- C:/home/gein/xip/src/database/overlay/SoXipPolyLine.cpp

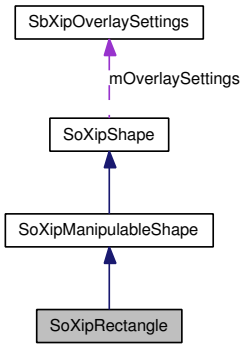
4.37 SoXipRectangle Class Reference

`#include <SoXipRectangle.h>`

Inheritance diagram for SoXipRectangle:



Collaboration diagram for SoXipRectangle:



Public Member Functions

- SoXipRectangle ()

  Constructor.

- virtual SbBool isClosed () const

  Is the shape closed ? (always true for rectangle).

- virtual void transform (const SbMatrix &matrix)

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Protected Member Functions

- ~SoXipRectangle ()

  Destructor.

- virtual void beforeCreation ()

  Callback function called before the shape creation.

- virtual void extractLinePoints (SoMFVec3f &points)

  Extract points from the shape characteristics.

- virtual void extractLineSegments (SoMFInt32 &index)

  Extract line indices referring to the shape points.

- virtual void mouseDown (const SbVec3f &pos)

  Send a Mouse Down event to the rectangle.

- virtual void mouseMove (const SbVec3f &pos)

  Send a Mouse Move event to the rectangle.

- virtual void mouseUp (const SbVec3f &pos)

  Send a Mouse Up event to the rectangle.

Protected Attributes

- bool mHasFirstPoint

  Has the first point already been defined ?

4.37.1 Detailed Description

This node is used to create a rectangle. It is a permanent overlay. User may use multiple instances of this node to perform multiple measures, or use this class as a parameter of the overlay manager (cf. SoXipOverlayManager)

See also:

SoXipOverlayManager

4.37.2 Member Function Documentation

4.37.2.1 void SoXipRectangle::transform (const SbMatrix & *matrix*) [virtual]

Transform the rectangle coordinates by the given transformation matrix

Implements SoXipManipulableShape.

4.37.2.2 void SoXipRectangle::extractLinePoints (SoMFVec3f & *points*) [protected, virtual]

Extract points from the shape characteristics.

This method is called by the base class to retrieve the segments defining the rectangle.

Parameters:

points output array of points

Implements SoXipManipulableShape.

4.37.2.3 void SoXipRectangle::extractLineSegments (SoMFInt32 & *index*) [protected, virtual]

Extract line indices referring to the shape points.

This method is called by the base class to retrieve the segments defining the rectangle.

Parameters:

index output array of indices

Reimplemented from SoXipManipulableShape.

4.37.2.4 void SoXipRectangle::mouseDown (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Down event to the rectangle.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

4.37.2.5 void SoXipRectangle::mouseMove (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Move event to the rectangle.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

4.37.2.6 void SoXipRectangle::mouseUp (const SbVec3f & *pos*) [protected, virtual]

Send a Mouse Up event to the rectangle.

Parameters:

pos projection of the mouse position in the view

Reimplemented from SoXipManipulableShape.

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

- C:/home/gein/xip/src/database/overlay/SoXipRectangle.h
- C:/home/gein/xip/src/database/overlay/SoXipRectangle.cpp

4.38 SoXipSaveOverlay Class Reference

Engine used to save overlays to an external file.

`#include <SoXipSaveOverlay.h>`

Public Member Functions

- SoXipSaveOverlay ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoSFString filename

  filename

- SoMFNode overlays

  list of overlays to be saved

- SoSFTrigger save

  perform the save operation

- SoEngineOutput status

  Engine dummy output (to allow serialization to file).

Protected Member Functions

- ~SoXipSaveOverlay ()

  Destructor.

- virtual void inputChanged (SoField *whichField)

  Engine inputChanged method.

- virtual void evaluate ()

  Engine evaluate method.

4.38.1 Detailed Description

Engine used to save overlays to an external file.

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

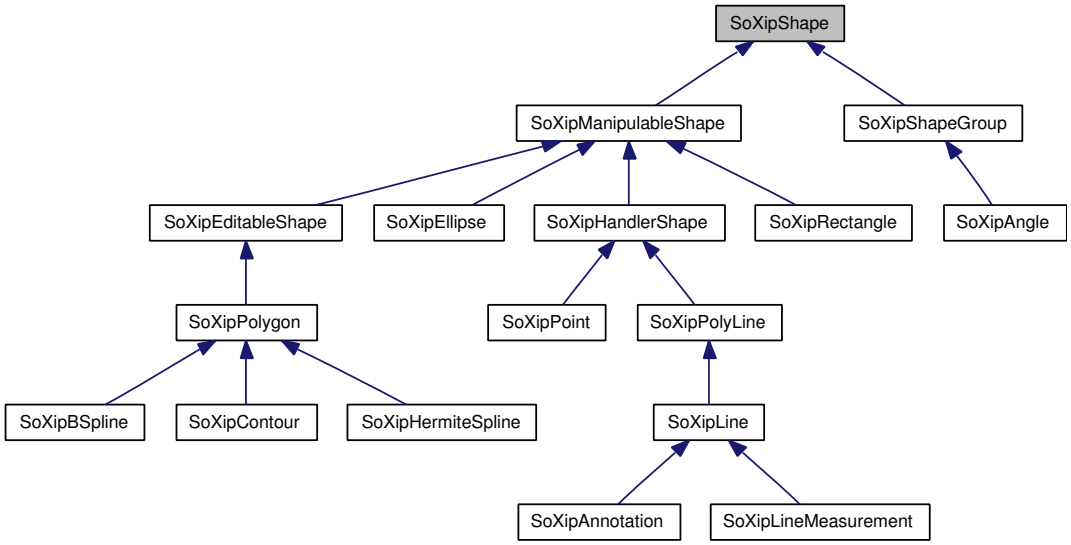
- C:/home/gein/xip/src/database/overlay/SoXipSaveOverlay.h
- C:/home/gein/xip/src/database/overlay/SoXipSaveOverlay.cpp

4.39 SoXipShape Class Reference

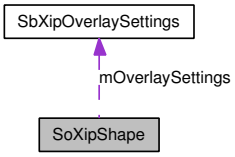
Base class for all the shapes.

`#include <SoXipShape.h>`

Inheritance diagram for SoXipShape:



Collaboration diagram for SoXipShape:



Public Types

- enum StatusType

  Status of the shape.

Public Member Functions

- SoXipShape ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoSFEnum status

  Shape status.

- SoSFInt32 order

  Shape order (back to front).

- SoSFInt32 rank

  Shape rank (enumeration).

- SoSFString label

  Shape label (name).

- SoSFString caption

  Shape caption (annotation).

Protected Member Functions

- ~SoXipShape ()

  Destructor.

- virtual void startEditing ()

  Callback function called before the shape editing.

- virtual void finishEditing ()

  Callback function called after the shape editing.

- virtual void beforeCreation ()

  Callback function called before the shape creation.

- virtual void afterCreation ()

  Callback function called after the shape creation.

4.39.1 Detailed Description

Base class for all the shapes.

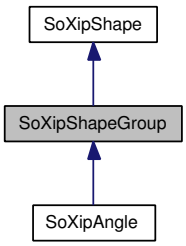
The documentation for this class was generated from the following files:

- C:/home/gein/xip/src/database/overlay/SoXipShape.h
- C:/home/gein/xip/src/database/overlay/SoXipShape.cpp

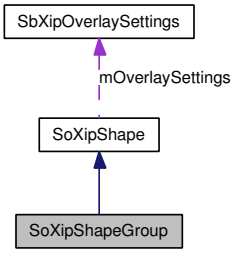
4.40 SoXipShapeGroup Class Reference

`#include <SoXipShapeGroup.h>`

Inheritance diagram for SoXipShapeGroup:



Collaboration diagram for SoXipShapeGroup:



Public Member Functions

- SoXipShapeGroup ()
  Constructor.
- virtual SbBool isEditing () const
  Is the shape group being edited ?
- virtual SbBool isCreating () const
  Is the shape group being created ?
- virtual SbBool isSelected () const
  Is the shape group selected ?
- void addChild (SoNode *child)
  Add a child shape to the group.
- void insertChild (SoNode *child, int newChildIndex)
  Insert a child shape to the group at the given position.
- void removeChild (int index)
  Remove child from the group at the given position.
- void removeAllChildren ()

Remove all child shapes from the group.

- void replaceChild (int index, SoNode *newChild)

  Replace the child at given position by given node.

- void setCreationChild (int index)

  Set the index of the child that will terminate the group creation.

- void setCreationChild (SoNode *child)

  Set the child that will terminate the group creation.

- virtual void setOrder (int order)

  Set the order of the group (back to front).

- virtual void setLabel (const SbString &label)

  Set the label (name) of the group.

- virtual void setRank (int rank)

  Set the rank (enumeration) of the group.

- virtual void setCaption (const SbString &caption)

  Set the caption (annotation) of the group.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoSFBool childrenLinked
- SoMFNode children

  List of children.

Protected Member Functions

- virtual ~SoXipShapeGroup ()

  Destructor.

- virtual void beforeCreation ()

  Callback function called before the shape creation.

- virtual void afterCreation ()

  Callback function called after the shape creation.

4.40.1 Detailed Description

This class can be used as base class for all the shapes that are a composition of one or more SoXipShape. The shape group is handled by the framework as one overlay (same id, etc.) although its children can be manipulated individually.

4.40.2 Member Data Documentation

4.40.2.1 SoSFBool SoXipShapeGroup::childrenLinked

If children are linked, then selecting/unselecting one child of this group is like selecting/unselecting all the children

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

- C:/home/gein/xip/src/database/overlay/SoXipShapeGroup.h
- C:/home/gein/xip/src/database/overlay/SoXipShapeGroup.cpp

4.41 SoXipText2 Class Reference

`#include <SoXipText2.h>`

Inherited by SoXipEditText2.

Public Types

- enum fontType

  Font type.

- enum alignmentType

  Horizontal and vertical alignment types.

Public Member Functions

- SoXipText2 ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Public Attributes

- SoSFString string

  String to be displayed.

- SoSFEnum justification

  Horizontal alignement.

- SoSFEnum vAlignment

  Vertical alignement.

- SoSFEnum type

  Font type.

Protected Member Functions

- virtual ~SoXipText2 ()

  Destructor.

4.41.1 Detailed Description

Text node

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

- C:/home/gein/xip/src/database/overlay/SoXipText2.h
- C:/home/gein/xip/src/database/overlay/SoXipText2.cpp

4.42 SoXipWidgetCamera Class Reference

`#include <SoXipWidgetCamera.h>`

Public Member Functions

- SoXipWidgetCamera ()

  Constructor.

Static Public Member Functions

- static void initClass ()

  Open Inventor class initialization.

Protected Member Functions

- virtual ~SoXipWidgetCamera ()

  Destructor.

- virtual void updateScale ()

  Compute the scale factor used to keep the menu size constant, independently of the viewport size and aspect ratio.

- virtual void saveViewInformation (SoAction *action)

  Save the view information.

4.42.1 Detailed Description

Simple camera for 2D objects (preserve scale)

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

- C:/home/gein/xip/src/database/overlay/SoXipWidgetCamera.h
- C:/home/gein/xip/src/database/overlay/SoXipWidgetCamera.cpp

4.43 XipBSpline Class Reference

`#include <XipBSpline.h>`

Public Member Functions

- XipBSpline (const SoMFVec3f &controlPoints, int order, SbBool closed)
  Constructor.
- ~XipBSpline ()
  Destructor.
- void interpolateContour (int numPointsPerSegment, SoMFVec3f &output)
  Interpolate a contour given a list of control points.

Protected Member Functions

- void computeKnots ()
  Compute the knot vector.
- double blend (int k, int order, double v)
  Compute the blend factor.
- SbVec3f interpolate (double value)
  Compute curve point at the given location.

Protected Attributes

- SoMFVec3f mControlPoints
  spline control points
- int mOrder
  Order of the spline.
- SbBool mClosed
  Is the generated output contour closed ?
- int * mKnots
  Knot vector.

4.43.1 Detailed Description

This class can be used to interpolate a contour given a list of control points, using a B-spline interpolation. Use of this class can be curve fitting (interpolated contour points will not pass by the control points), contour representation, etc.

4.43.2 Member Function Documentation

4.43.2.1 void XipBSpline::interpolateContour (int *numPointsPerSegment*, SoMFVec3f & *output*)

Interpolate a contour given a list of control points.

Parameters:

numPointsPerSegment number of points per interpolated segment

output output contour

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

- C:/home/gein/xip/src/database/overlay/XipBSpline.h
- C:/home/gein/xip/src/database/overlay/XipBSpline.cpp

4.44 XipHermiteSpline Class Reference

`#include <XipHermiteSpline.h>`

Public Member Functions

- XipHermiteSpline (const SoMFVec3f &controlPoints, SbBool closed)

  Constructor.

- ~XipHermiteSpline ()

  Destructor.

- void interpolateContour (int numPointsPerSegment, SoMFVec3f &output)

  Interpolate a contour given a list of control points.

Static Public Member Functions

- static void interpolateSegment (const SbVec3f &p0, const SbVec3f &p1, const SbVec3f &m0, const SbVec3f &m1, SbVec3f *pointsPtr, int numPoints)

  Interpolate points along a given segment, represented by two control points.

Protected Attributes

- SoMFVec3f mControlPoints

  Spline control points.

- SbBool mClosed

  Is the generated output contour closed ?

4.44.1 Detailed Description

This class can be used to interpolate points given a segment and two tangents, or interpolate a contour given a set of control points (the tangents are in this case generated)

4.44.2 Member Function Documentation

4.44.2.1 void XipHermiteSpline::interpolateSegment (const SbVec3f & *p0*, const SbVec3f & *p1*, const SbVec3f & *m0*, const SbVec3f & *m1*, SbVec3f * *pointsPtr*, int *numPoints*) [static]

Interpolate points along a given segment, represented by two control points.

Interpolation is made so that the generated curve has in its two extremities, tangents that match the tangents specified by the user.

Parameters:

p0 first control point

p1 second control point

m0 tangent associated to the first control point

m1 tangent associated to the second control point

pointsPtr output point buffer

numPoints number of points to interpolate between p0 and p1 (both included)

4.44.2.2 void XipHermiteSpline::interpolateContour (int *numPointsPerSegment*, SoMFVec3f & *output*)

Interpolate a contour given a list of control points.

Parameters:

numPointsPerSegment number of points per interpolated segment

output output contour

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

- C:/home/gein/xip/src/database/overlay/XipHermiteSpline.h
- C:/home/gein/xip/src/database/overlay/XipHermiteSpline.cpp

Chapter 5

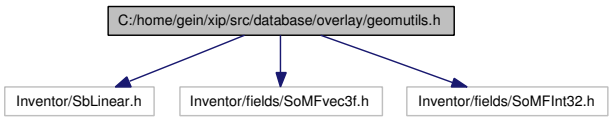
XipIvOverlay File Documentation

5.1 C:/home/gein/xip/src/database/overlay/geomutils.h File Reference

Contains geometry utility functions.

```
#include <Inventor/SbLinear.h>
#include <Inventor/fields/SoMFvec3f.h>
#include <Inventor/fields/SoMFInt32.h>
```

Include dependency graph for geomutils.h:



5.1.1 Detailed Description

Contains geometry utility functions.

Author:

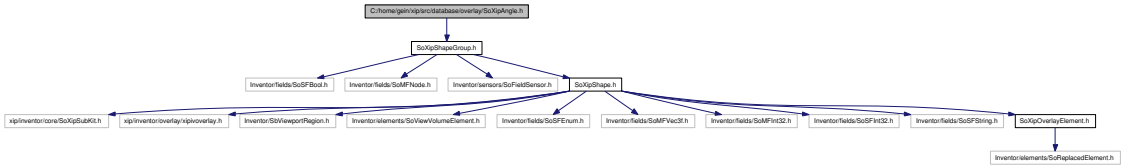
Daphne Yu
Julien Gein
Thomas Moeller

5.2 C:/home/gein/xip/src/database/overlay/SoXipAngle.h File Reference

Node to perform an angle measurement in the current view.

```
#include "SoXipShapeGroup.h"
```

Include dependency graph for SoXipAngle.h:



Classes

- class SoXipAngle

5.2.1 Detailed Description

Node to perform an angle measurement in the current view.

Author:

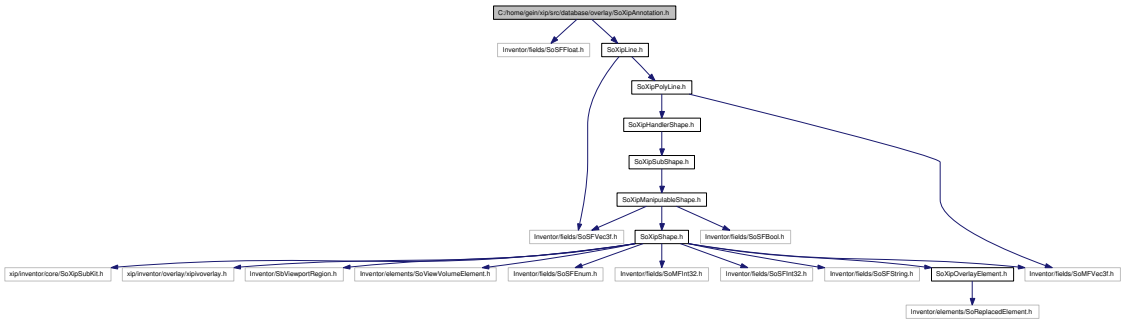
Julien Gein

5.3 C:/home/gein/xip/src/database/overlay/SoXipAnnotation.h File Reference

Node to annotate an object in the current view.

```
#include <Inventor/fields/SoSFFloat.h>
#include "SoXipLine.h"
```

Include dependency graph for SoXipAnnotation.h:



Classes

- class SoXipAnnotation

5.3.1 Detailed Description

Node to annotate an object in the current view.

Author:

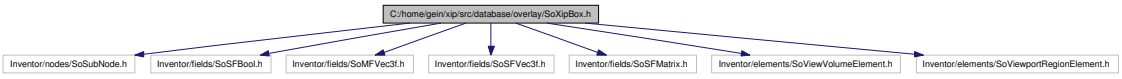
Julien Gein

5.4 C:/home/gein/xip/src/database/overlay/SoXipBox.h File Reference

Declaration of the SoXipBox ROI tool.

```
#include <Inventor/nodes/SoSubNode.h>
#include <Inventor/fields/SoSFBool.h>
#include <Inventor/fields/SoMFVec3f.h>
#include <Inventor/fields/SoSFVec3f.h>
#include <Inventor/fields/SoSFMatrix.h>
#include <Inventor/elements/SoViewVolumeElement.h>
#include <Inventor/elements/SoViewportRegionElement.h>
```

Include dependency graph for SoXipBox.h:



Classes

- class SoXipBox

5.4.1 Detailed Description

Declaration of the SoXipBox ROI tool.

Author:

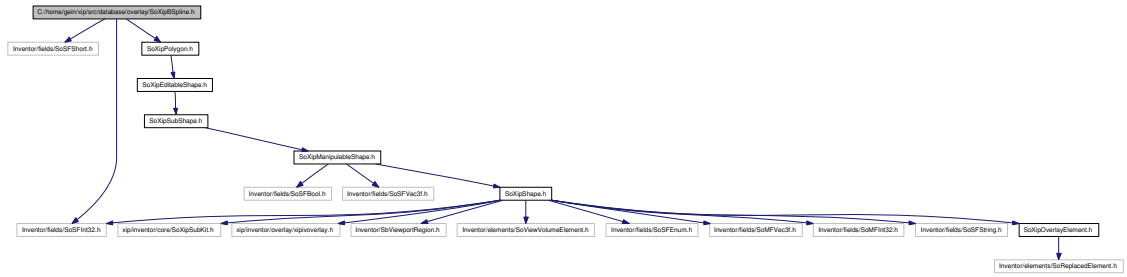
Julien Gein

5.5 C:/home/gein/xip/src/database/overlay/SoXipBSpline.h File Reference

Interactive B-Spline tool.

```
#include <Inventor/fields/SoSFShort.h>
#include <Inventor/fields/SoSFInt32.h>
#include "SoXipPolygon.h"
```

Include dependency graph for SoXipBSpline.h:



Classes

- class SoXipBSpline

5.5.1 Detailed Description

Interactive B-Spline tool.

Author:

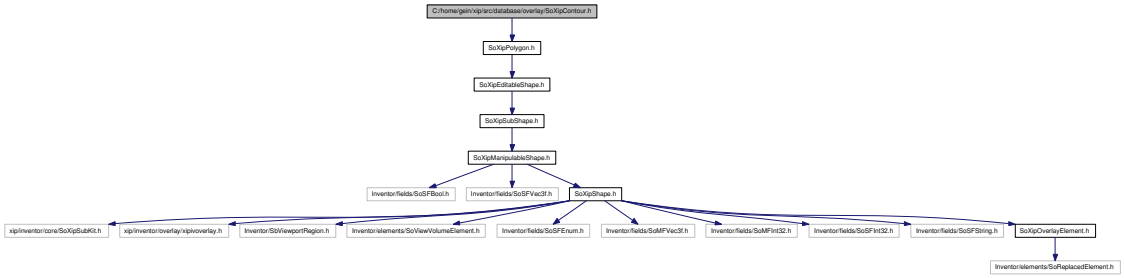
Julien Gein

5.6 C:/home/gein/xip/src/database/overlay/SoXipContour.h File Reference

Declaration of the SoXipContour overlay node.

```
#include "SoXipPolygon.h"
```

Include dependency graph for SoXipContour.h:



Classes

- class SoXipContour

5.6.1 Detailed Description

Declaration of the SoXipContour overlay node.

Author:

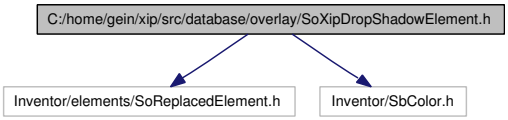
Julien Gein

5.7 C:/home/gein/xip/src/database/overlay/SoXipDropShadowElement.h File Reference

Declaration of the SoXipDropShadowElement element.

```
#include <Inventor/elements/SoReplacedElement.h>
#include <Inventor/SbColor.h>
```

Include dependency graph for SoXipDropShadowElement.h:



Classes

- class SoXipDropShadowElement

5.7.1 Detailed Description

Declaration of the SoXipDropShadowElement element.

Author:

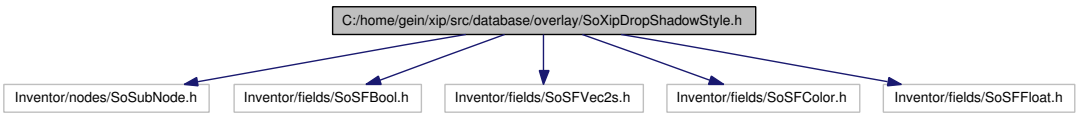
Julien Gein

5.8 C:/home/gein/xip/src/database/overlay/SoXipDropShadowStyle.h File Reference

Declaration of the SoXipDropShadowStyle node.

```
#include <Inventor/nodes/SoSubNode.h>
#include <Inventor/fields/SoSFBool.h>
#include <Inventor/fields/SoSFVec2s.h>
#include <Inventor/fields/SoSFColor.h>
#include <Inventor/fields/SoSFFloat.h>
```

Include dependency graph for SoXipDropShadowStyle.h:



Classes

- class SoXipDropShadowStyle

5.8.1 Detailed Description

Declaration of the SoXipDropShadowStyle node.

Author:

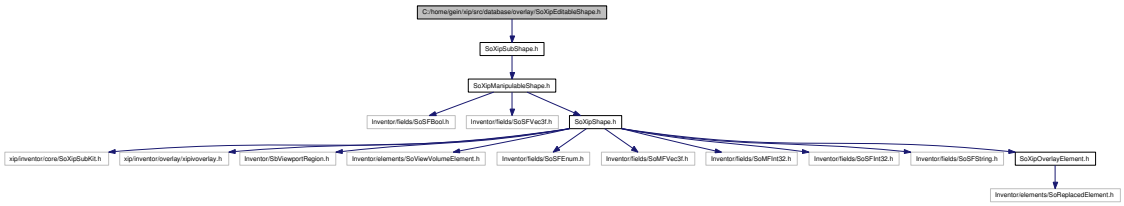
Julien Gein

5.9 C:/home/gein/xip/src/database/overlay/SoXipEditableShape.h File Reference

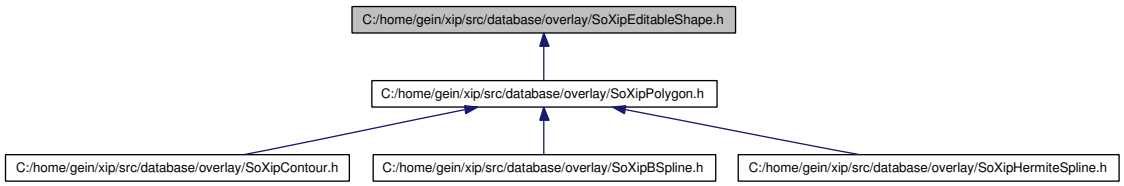
Base class for all editable shapes.

```
#include "SoXipSubShape.h"
```

Include dependency graph for SoXipEditableShape.h:



This graph shows which files directly or indirectly include this file:



Classes

- class SoXipEditableShape

5.9.1 Detailed Description

Base class for all editable shapes.

Author:

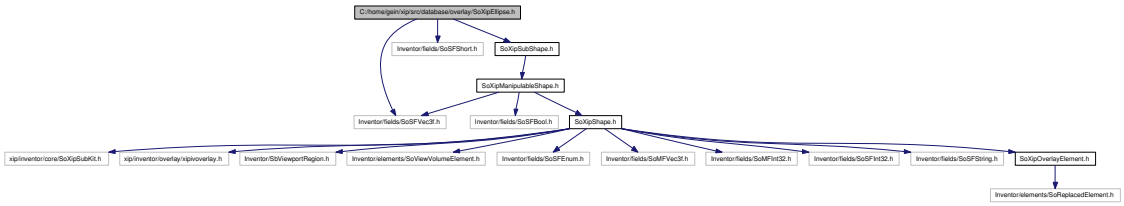
Julien Gein

5.10 C:/home/gein/xip/src/database/overlay/SoXipEllipse.h File Reference

Declaration of the SoXipEllipse overlay module.

```
#include <Inventor/fields/SoSFVec3f.h>
#include <Inventor/fields/SoSFShort.h>
#include "SoXipSubShape.h"
```

Include dependency graph for SoXipEllipse.h:



Classes

- class SoXipEllipse

5.10.1 Detailed Description

Declaration of the SoXipEllipse overlay module.

Author:

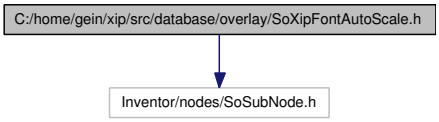
Julien Gein

5.11 C:/home/gein/xip/src/database/overlay/SoXipFontAutoScale.h File Reference

Declaration of the SoXipFontAutoScale class.

```
#include <Inventor/nodes/SoSubNode.h>
```

Include dependency graph for SoXipFontAutoScale.h:



Classes

- class SoXipFontAutoScale

5.11.1 Detailed Description

Declaration of the SoXipFontAutoScale class.

Author:

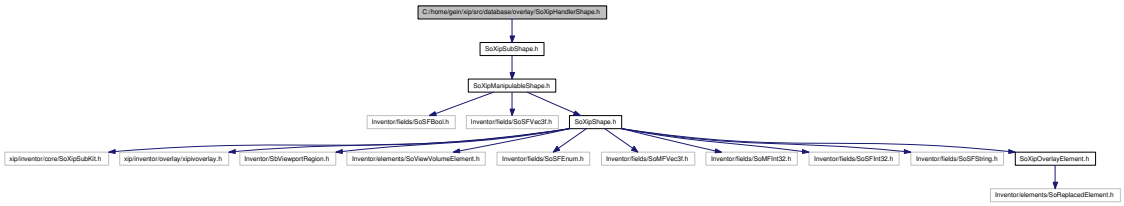
Ravi Kumar

5.12 C:/home/gein/xip/src/database/overlay/SoXipHandlerShape.h File Reference

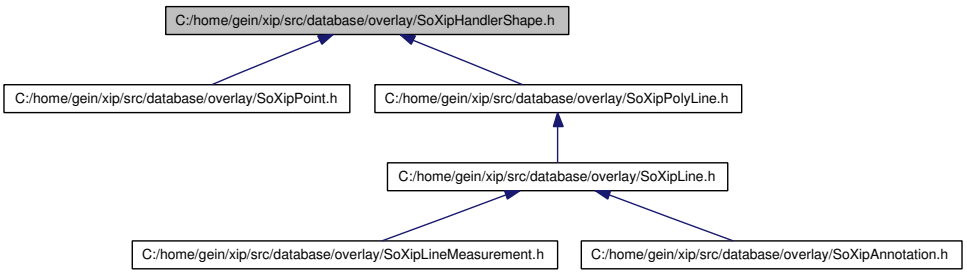
Base class for shapes being manipulated with handlers.

```
#include "SoXipSubShape.h"
```

Include dependency graph for SoXipHandlerShape.h:



This graph shows which files directly or indirectly include this file:



Classes

- class SoXipHandlerShape

5.12.1 Detailed Description

Base class for shapes being manipulated with handlers.

Author:

Julien Gein

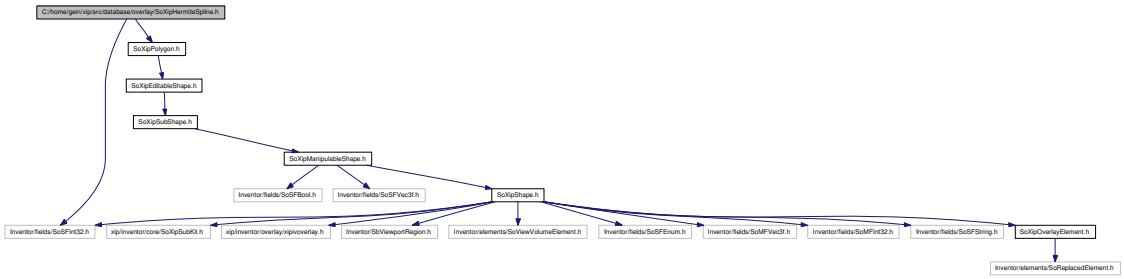
5.13 C:/home/gein/xip/src/database/overlay/SoXipHermiteSpline.h File Reference

Interactive Spline tool.

```
#include <Inventor/fields/SoSFInt32.h>
```

```
#include "SoXipPolygon.h"
```

Include dependency graph for SoXipHermiteSpline.h:



Classes

- class SoXipHermiteSpline

5.13.1 Detailed Description

Interactive Spline tool.

Author:

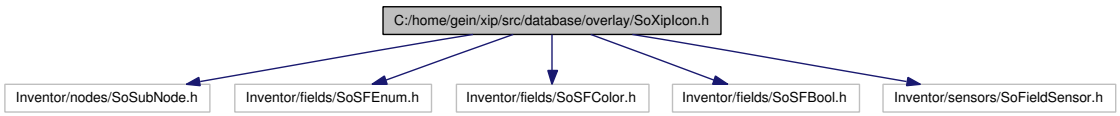
Julien Gein

5.14 C:/home/gein/xip/src/database/overlay/SoXipIcon.h File Reference

Icon class, to be used for 2D GUI purposes.

```
#include <Inventor/nodes/SoSubNode.h>
#include <Inventor/fields/SoSFEnum.h>
#include <Inventor/fields/SoSFColor.h>
#include <Inventor/fields/SoSFBool.h>
#include <Inventor/sensors/SoFieldSensor.h>
```

Include dependency graph for SoXipIcon.h:



Classes

- class SoXipIcon

5.14.1 Detailed Description

Icon class, to be used for 2D GUI purposes.

Author:

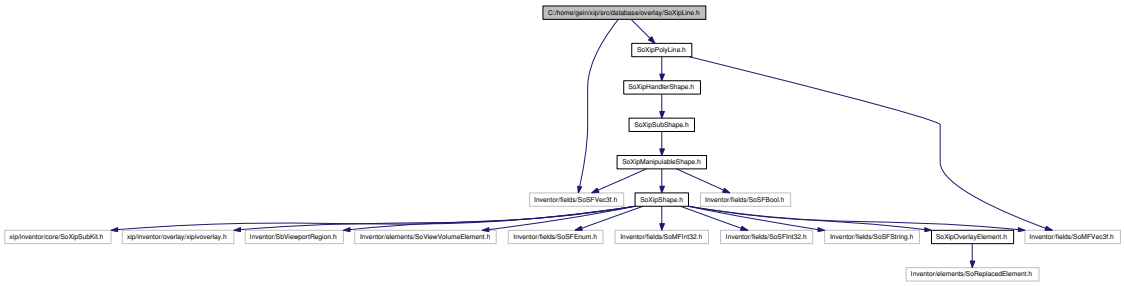
Julien Gein

5.15 C:/home/gein/xip/src/database/overlay/SoXipLine.h File Reference

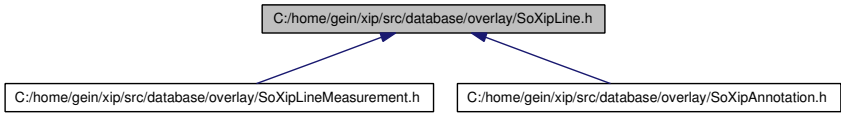
Node to create a permanent line overlay in the current view.

```
#include <Inventor/fields/SoSFVec3f.h>
#include "SoXipPolyLine.h"
```

Include dependency graph for SoXipLine.h:



This graph shows which files directly or indirectly include this file:



Classes

- class SoXipLine

5.15.1 Detailed Description

Node to create a permanent line overlay in the current view.

Author:

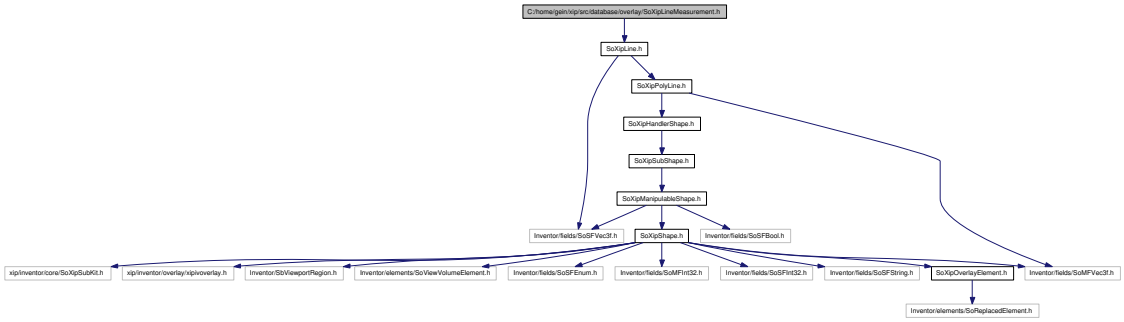
Julien Gein

5.16 C:/home/gein/xip/src/database/overlay/SoXipLineMeasurement.h File Reference

Node to create a permanent line distance measurement in the current view.

```
#include "SoXipLine.h"
```

Include dependency graph for SoXipLineMeasurement.h:



Classes

- class SoXipLineMeasurement

5.16.1 Detailed Description

Node to create a permanent line distance measurement in the current view.

Author:

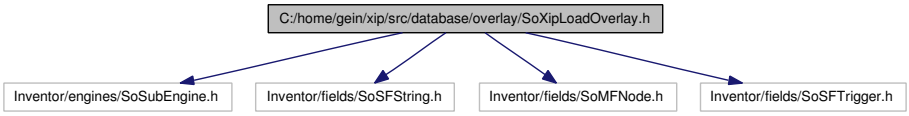
Julien Gein

5.17 C:/home/gein/xip/src/database/overlay/SoXipLoadOverlay.h File Reference

Declaration of the SoXipLoadOverlay class.

```
#include <Inventor/engines/SoSubEngine.h>
#include <Inventor/fields/SoSFString.h>
#include <Inventor/fields/SoMFNode.h>
#include <Inventor/fields/SoSFTrigger.h>
```

Include dependency graph for SoXipLoadOverlay.h:



Classes

- class SoXipLoadOverlay

5.17.1 Detailed Description

Declaration of the SoXipLoadOverlay class.

Author:

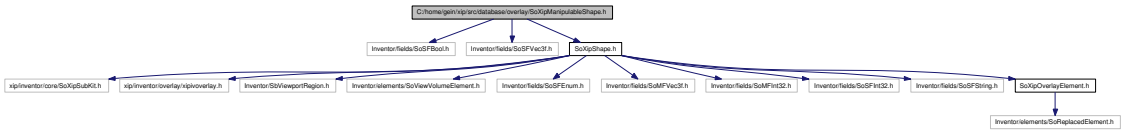
Julien Gein

5.18 C:/home/gein/xip/src/database/overlay/SoXipManipulableShape.h File Reference

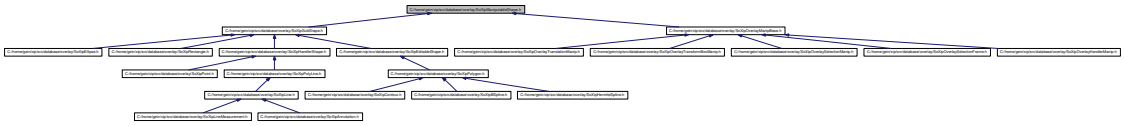
Base class for all manipulable shapes.

```
#include <Inventor/fields/SoSFBool.h>
#include <Inventor/fields/SoSFVec3f.h>
#include "SoXipShape.h"
```

Include dependency graph for SoXipManipulableShape.h:



This graph shows which files directly or indirectly include this file:



Classes

- class SoXipManipulableShape

5.18.1 Detailed Description

Base class for all manipulable shapes.

Author:

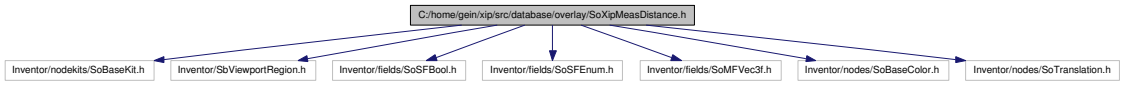
Julien Gein

5.19 C:/home/gein/xip/src/database/overlay/SoXipMeasDistance.h File Reference

Declaration of the SoXipMeasDistance class.

```
#include <Inventor/nodekits/SoBaseKit.h>
#include <Inventor/SbViewportRegion.h>
#include <Inventor/fields/SoSFBool.h>
#include <Inventor/fields/SoSFEnum.h>
#include <Inventor/fields/SoMFVec3f.h>
#include <Inventor/nodes/SoBaseColor.h>
#include <Inventor/nodes/SoTranslation.h>
```

Include dependency graph for SoXipMeasDistance.h:



5.19.1 Detailed Description

Declaration of the SoXipMeasDistance class.

Author:

Thomas Moeller

5.20 C:/home/gein/xip/src/database/overlay/SoXipMeasPixelLens.h File Reference

Declaration of the SoXipMeasPixelLens class.

```
#include <Inventor/nodekits/SoBaseKit.h>
#include <Inventor/SbViewportRegion.h>
#include <Inventor/fields/SoSFBool.h>
#include <Inventor/fields/SoSFInt32.h>
#include <xip/inventor/core/SoXipSFDataImage.h>
#include <xip/inventor/core/SoXipDataImage.h>
#include <xip/inventor/core/SbXipImage.h>
#include <Inventor/nodes/SoBaseColor.h>
#include <Inventor/nodes/SoTranslation.h>
```

Include dependency graph for SoXipMeasPixelLens.h:

Classes

- class SoXipMeasPixelLens

5.20.1 Detailed Description

Declaration of the SoXipMeasPixelLens class.

Author:

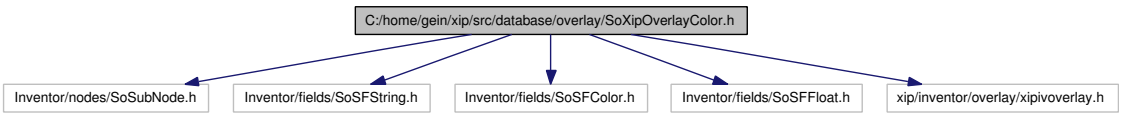
Thomas Moeller

5.21 C:/home/gein/xip/src/database/overlay/SoXipOverlayColor.h File Reference

Declaration of the SoXipOverlayColor class.

```
#include <Inventor/nodes/SoSubNode.h>
#include <Inventor/fields/SoSFString.h>
#include <Inventor/fields/SoSFColor.h>
#include <Inventor/fields/SoSFFloat.h>
#include <xip/inventor/overlay/xipivoverlay.h>
```

Include dependency graph for SoXipOverlayColor.h:



5.21.1 Detailed Description

Declaration of the SoXipOverlayColor class.

Author:

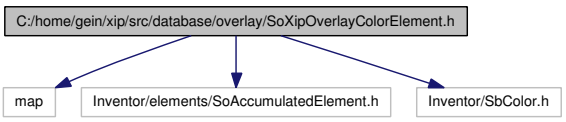
Julien Gein

5.22 C:/home/gein/xip/src/database/overlay/SoXipOverlayColorElement.h File Reference

Declaration of the SoXipOverlayColorElement class.

```
#include <map>
#include <Inventor/elements/SoAccumulatedElement.h>
#include <Inventor/SbColor.h>
```

Include dependency graph for SoXipOverlayColorElement.h:



5.22.1 Detailed Description

Declaration of the SoXipOverlayColorElement class.

Author:

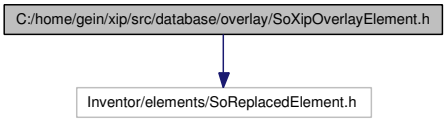
Julien Gein

5.23 C:/home/gein/xip/src/database/overlay/SoXipOverlayElement.h File Reference

Declaration of the SoXipOverlayElement and SbXipOverlaySettings classes.

`#include <Inventor/elements/SoReplacedElement.h>`

Include dependency graph for SoXipOverlayElement.h:



This graph shows which files directly or indirectly include this file:

Classes

- class SbXipOverlaySettings
- class SoXipOverlayElement

5.23.1 Detailed Description

Declaration of the SoXipOverlayElement and SbXipOverlaySettings classes.

Author:

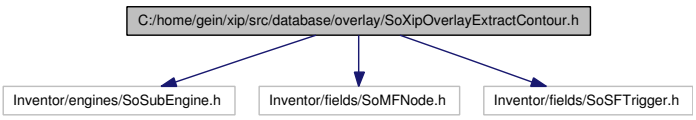
Julien Gein

5.24 C:/home/gein/xip/src/database/overlay/SoXipOverlayExtractContour.h File Reference

Declaration of the SoXipOverlayExtractContour class.

```
#include <Inventor/engines/SoSubEngine.h>
#include <Inventor/fields/SoMFNode.h>
#include <Inventor/fields/SoSFTrigger.h>
```

Include dependency graph for SoXipOverlayExtractContour.h:



Classes

- class SoXipOverlayExtractContour

5.24.1 Detailed Description

Declaration of the SoXipOverlayExtractContour class.

Author:

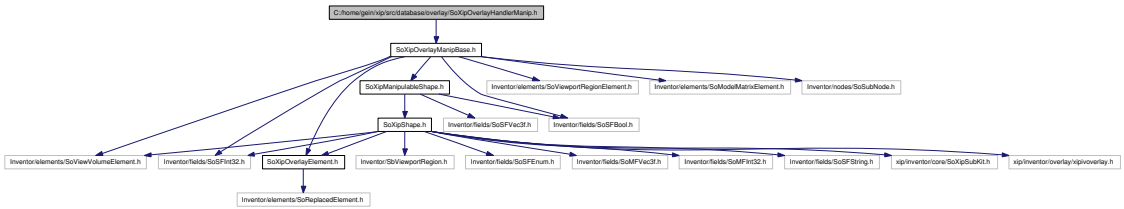
Julien Gein

5.25 C:/home/gein/xip/src/database/overlay/SoXipOverlayHandlerManip.h File Reference

Declaration of the SoXipOverlayHandlerManip class.

```
#include "SoXipOverlayManipBase.h"
```

Include dependency graph for SoXipOverlayHandlerManip.h:



Classes

- class SoXipOverlayHandlerManip

5.25.1 Detailed Description

Declaration of the SoXipOverlayHandlerManip class.

Author:

Julien Gein

5.26 C:/home/gein/xip/src/database/overlay/SoXipOverlayManager.h File Reference

Node responsible for handling of multiple overlays.

```
#include <xip/inventor/core/SoXipList.h>
#include <Inventor/SoLists.h>
#include <Inventor/fields/SoSFString.h>
#include <Inventor/fields/SoSFBool.h>
#include <Inventor/fields/SoSFColor.h>
#include <Inventor/fields/SoSFEnum.h>
#include <Inventor/fields/SoSFTrigger.h>
#include "SoXipShape.h"
```

Include dependency graph for SoXipOverlayManager.h:

Classes

- class SoXipOverlayManager

5.26.1 Detailed Description

Node responsible for handling of multiple overlays.

Author:

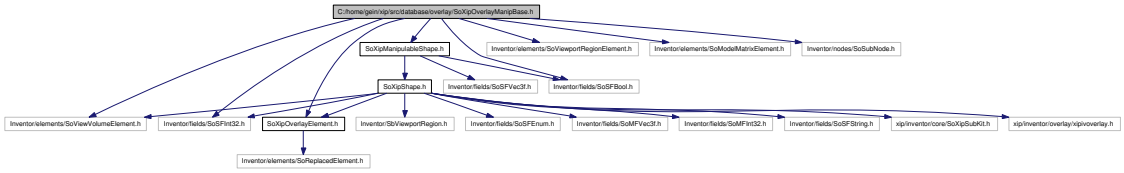
Julien Gein

5.27 C:/home/gein/xip/src/database/overlay/SoXipOverlayManipBase.h File Reference

Declaration of the SoXipOverlayManipBase class.

```
#include <Inventor/elements/SoViewVolumeElement.h>
#include <Inventor/elements/SoViewportRegionElement.h>
#include <Inventor/elements/SoModelMatrixElement.h>
#include <Inventor/fields/SoSFBool.h>
#include <Inventor/fields/SoSFInt32.h>
#include <Inventor/nodes/SoSubNode.h>
#include "SoXipOverlayElement.h"
#include "SoXipManipulableShape.h"
```

Include dependency graph for SoXipOverlayManipBase.h:



This graph shows which files directly or indirectly include this file:

Classes

- class SoXipOverlayManipBase

5.27.1 Detailed Description

Declaration of the SoXipOverlayManipBase class.

Author:

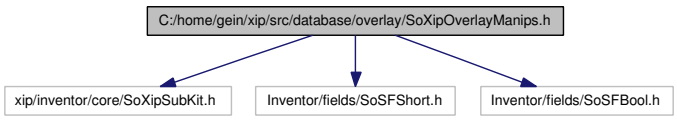
Julien Gein

5.28 C:/home/gein/xip/src/database/overlay/SoXipOverlayManips.h File Reference

Declaration of the SoXipOverlayManips class.

```
#include <xip/inventor/core/SoXipSubKit.h>
#include <Inventor/fields/SoSFShort.h>
#include <Inventor/fields/SoSFBool.h>
```

Include dependency graph for SoXipOverlayManips.h:



Classes

- class SoXipOverlayManips

5.28.1 Detailed Description

Declaration of the SoXipOverlayManips class.

Author:

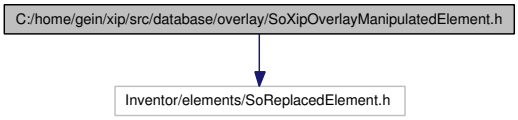
Julien Gein

5.29 C:/home/gein/xip/src/database/overlay/SoXipOverlayManipulatedElement.h File Reference

Declaration of the SoXipOverlayManipulatedElement class.

```
#include <Inventor/elements/SoReplacedElement.h>
```

Include dependency graph for SoXipOverlayManipulatedElement.h:



Classes

- class SoXipOverlayManipulatedElement

5.29.1 Detailed Description

Declaration of the SoXipOverlayManipulatedElement class.

Author:

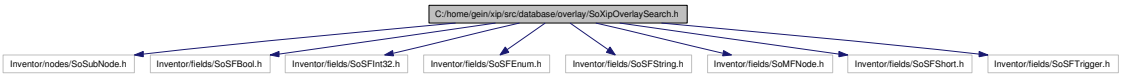
Julien Gein

5.30 C:/home/gein/xip/src/database/overlay/SoXipOverlaySearch.h File Reference

Declaration of the SoXipOverlaySearch class.

```
#include <Inventor/nodes/SoSubNode.h>
#include <Inventor/fields/SoSFBool.h>
#include <Inventor/fields/SoSFInt32.h>
#include <Inventor/fields/SoSFEnum.h>
#include <Inventor/fields/SoSFString.h>
#include <Inventor/fields/SoMFNode.h>
#include <Inventor/fields/SoSFShort.h>
#include <Inventor/fields/SoSFTrigger.h>
```

Include dependency graph for SoXipOverlaySearch.h:



5.30.1 Detailed Description

Declaration of the SoXipOverlaySearch class.

Author:

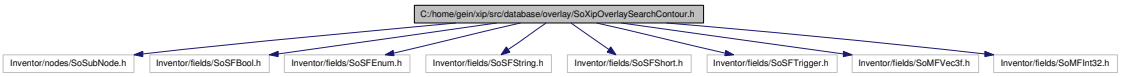
Julien Gein

5.31 C:/home/gein/xip/src/database/overlay/SoXipOverlaySearchContour.h File Reference

Declaration of the SoXipOverlaySearchContour class.

```
#include <Inventor/nodes/SoSubNode.h>
#include <Inventor/fields/SoSFBool.h>
#include <Inventor/fields/SoSFEnum.h>
#include <Inventor/fields/SoSFString.h>
#include <Inventor/fields/SoSFShort.h>
#include <Inventor/fields/SoSFTrigger.h>
#include <Inventor/fields/SoMFVec3f.h>
#include <Inventor/fields/SoMFInt32.h>
```

Include dependency graph for SoXipOverlaySearchContour.h:



Classes

- class SoXipOverlaySearchContour

5.31.1 Detailed Description

Declaration of the SoXipOverlaySearchContour class.

Author:

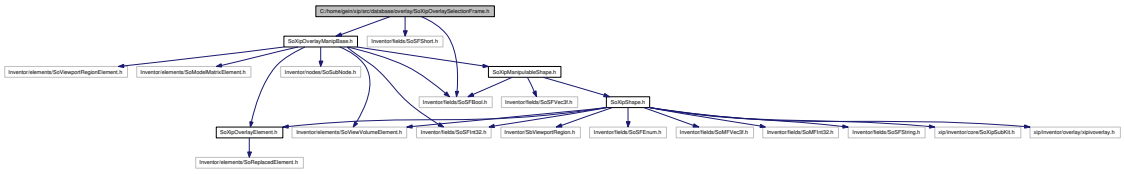
Julien Gein

5.32 C:/home/gein/xip/src/database/overlay/SoXipOverlaySelectionFrame.h File Reference

Declaration of the SoXipOverlaySelectionFrame class.

```
#include <Inventor/fields/SoSFBool.h>
#include <Inventor/fields/SoSFShort.h>
#include "SoXipOverlayManipBase.h"
```

Include dependency graph for SoXipOverlaySelectionFrame.h:



Classes

- class SoXipOverlaySelectionFrame

5.32.1 Detailed Description

Declaration of the SoXipOverlaySelectionFrame class.

Author:

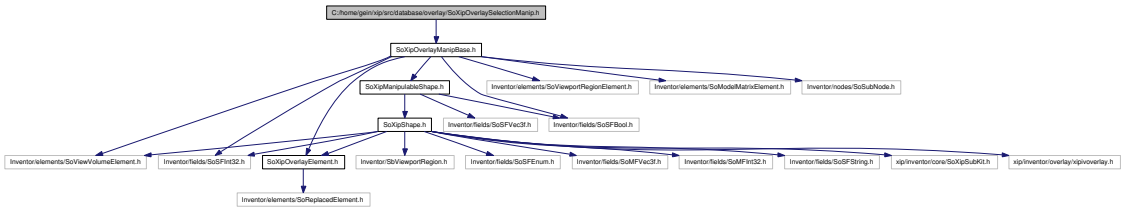
Julien Gein

5.33 C:/home/gein/xip/src/database/overlay/SoXipOverlaySelectionManip.h File Reference

Declaration of the SoXipOverlaySelectionManip class.

```
#include "SoXipOverlayManipBase.h"
```

Include dependency graph for SoXipOverlaySelectionManip.h:



Classes

- class SoXipOverlaySelectionManip

5.33.1 Detailed Description

Declaration of the SoXipOverlaySelectionManip class.

Author:

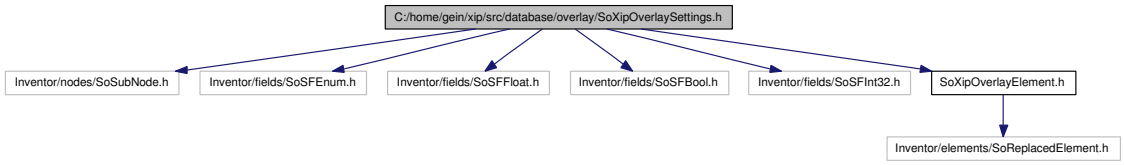
Julien Gein

5.34 C:/home/gein/xip/src/database/overlay/SoXipOverlaySettings.h File Reference

Declaration of the SoXipOverlaySettings class.

```
#include <Inventor/nodes/SoSubNode.h>
#include <Inventor/fields/SoSFEnum.h>
#include <Inventor/fields/SoSFFloat.h>
#include <Inventor/fields/SoSFBool.h>
#include <Inventor/fields/SoSFInt32.h>
#include "SoXipOverlayElement.h"
```

Include dependency graph for SoXipOverlaySettings.h:



Classes

- class SoXipOverlaySettings

5.34.1 Detailed Description

Declaration of the SoXipOverlaySettings class.

Author:

Julien Gein

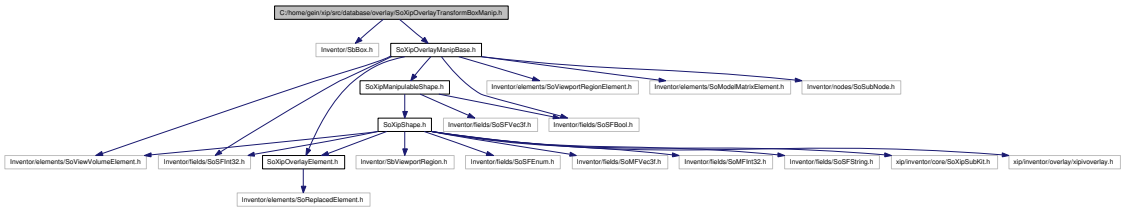
5.35 C:/home/gein/xip/src/database/overlay/SoXipOverlayTransformBoxManip.h File Reference

Declaration of the SoXipOverlayTransformBoxManip class.

```
#include <Inventor/SbBox.h>
```

```
#include "SoXipOverlayManipBase.h"
```

Include dependency graph for SoXipOverlayTransformBoxManip.h:



Classes

- class SoXipOverlayTransformBoxManip

5.35.1 Detailed Description

Declaration of the SoXipOverlayTransformBoxManip class.

Author:

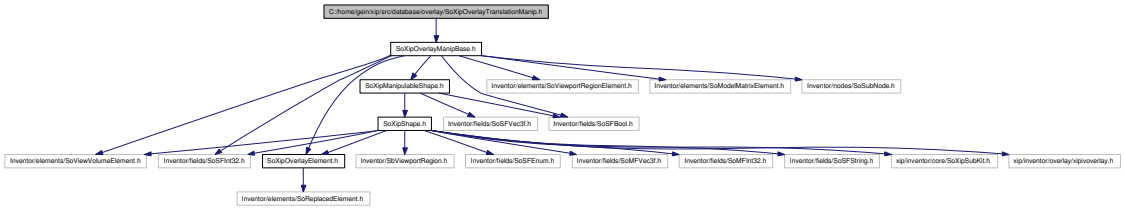
Julien Gein

5.36 C:/home/gein/xip/src/database/overlay/SoXipOverlayTranslationManip.h File Reference

Declaration of the SoXipOverlayTranslationManip class.

```
#include "SoXipOverlayManipBase.h"
```

Include dependency graph for SoXipOverlayTranslationManip.h:



Classes

- class SoXipOverlayTranslationManip

5.36.1 Detailed Description

Declaration of the SoXipOverlayTranslationManip class.

Author:

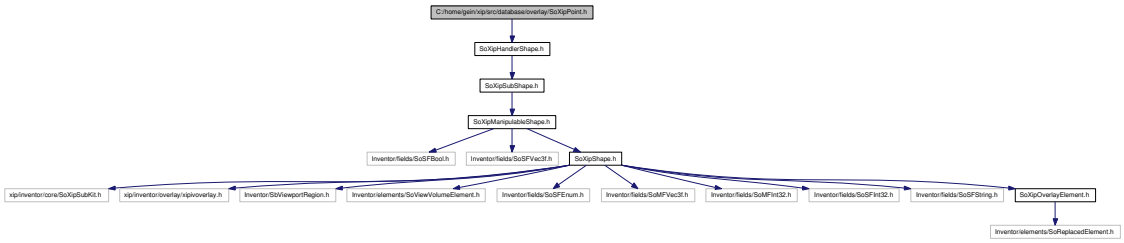
Julien Gein

5.37 C:/home/gein/xip/src/database/overlay/SoXipPoint.h File Reference

Declaration of the SoXipPoint class.

```
#include "SoXipHandlerShape.h"
```

Include dependency graph for SoXipPoint.h:



Classes

- class SoXipPoint

5.37.1 Detailed Description

Declaration of the SoXipPoint class.

Author:

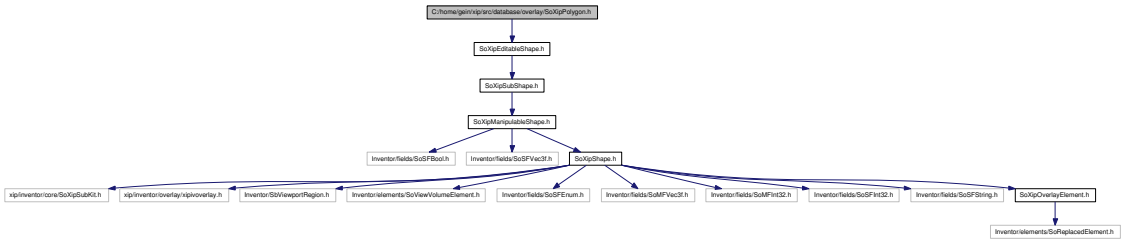
Julien Gein

5.38 C:/home/gein/xip/src/database/overlay/SoXipPolygon.h File Reference

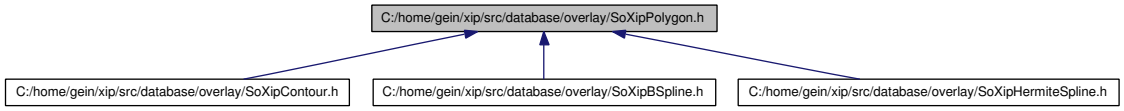
Declaration of the SoXipPolygon class.

```
#include "SoXipEditableShape.h"
```

Include dependency graph for SoXipPolygon.h:



This graph shows which files directly or indirectly include this file:



Classes

- class SoXipPolygon

5.38.1 Detailed Description

Declaration of the SoXipPolygon class.

Author:

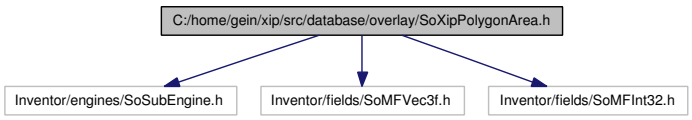
Julien Gein

5.39 C:/home/gein/xip/src/database/overlay/SoXipPolygonArea.h File Reference

Declaration of the SoXipPolygonArea class.

```
#include <Inventor/engines/SoSubEngine.h>
#include <Inventor/fields/SoMFVec3f.h>
#include <Inventor/fields/SoMFInt32.h>
```

Include dependency graph for SoXipPolygonArea.h:



Classes

- class SoXipPolygonArea

5.39.1 Detailed Description

Declaration of the SoXipPolygonArea class.

Author:

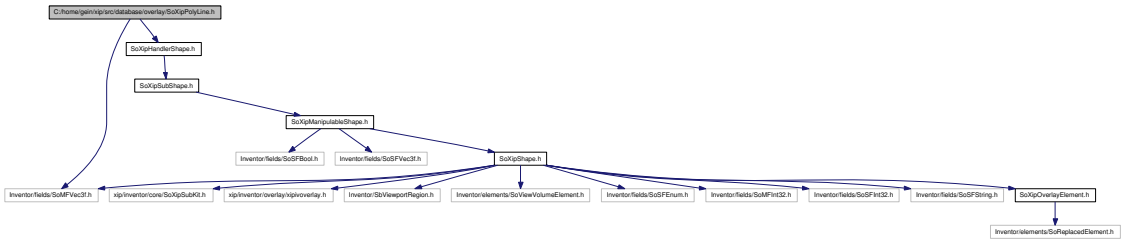
Julien Gein

5.40 C:/home/gein/xip/src/database/overlay/SoXipPolyLine.h File Reference

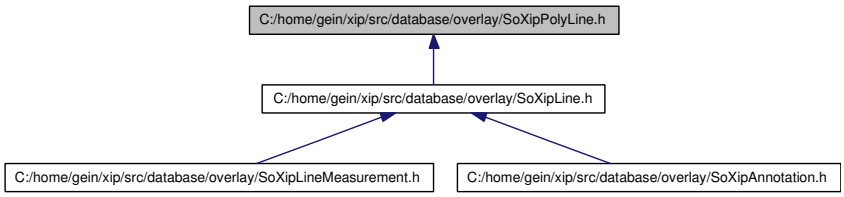
Declaration of the SoXipPolyLine class.

```
#include <Inventor/fields/SoMFVec3f.h>
#include "SoXipHandlerShape.h"
```

Include dependency graph for SoXipPolyLine.h:



This graph shows which files directly or indirectly include this file:



Classes

- class SoXipPolyLine

5.40.1 Detailed Description

Declaration of the SoXipPolyLine class.

Author:

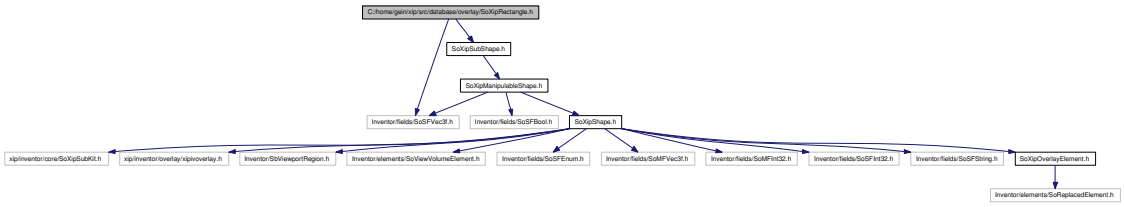
Julien Gein

5.41 C:/home/gein/xip/src/database/overlay/SoXipRectangle.h File Reference

Declaration of the SoXipRectangle class.

```
#include <Inventor/fields/SoSFVec3f.h>
#include "SoXipSubShape.h"
```

Include dependency graph for SoXipRectangle.h:



Classes

- class SoXipRectangle

5.41.1 Detailed Description

Declaration of the SoXipRectangle class.

Author:

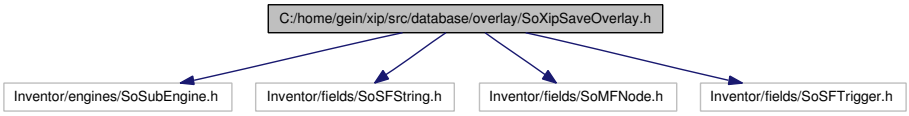
Julien Gein

5.42 C:/home/gein/xip/src/database/overlay/SoXipSaveOverlay.h File Reference

Declaration of the SoXipSaveOverlay engine.

```
#include <Inventor/engines/SoSubEngine.h>
#include <Inventor/fields/SoSFString.h>
#include <Inventor/fields/SoMFNode.h>
#include <Inventor/fields/SoSFTrigger.h>
```

Include dependency graph for SoXipSaveOverlay.h:



Classes

- class SoXipSaveOverlay

  Engine used to save overlays to an external file.

5.42.1 Detailed Description

Declaration of the SoXipSaveOverlay engine.

Author:

Julien Gein

5.43 C:/home/gein/xip/src/database/overlay/SoXipShape.h File Reference

Base class for all the shapes.

```
#include <xip/inventor/core/SoXipSubKit.h>
#include <xip/inventor/overlay/xipivoverlay.h>
#include <Inventor/SbViewportRegion.h>
#include <Inventor/elements/SoViewVolumeElement.h>
#include <Inventor/fields/SoSFEnum.h>
#include <Inventor/fields/SoMFVec3f.h>
#include <Inventor/fields/SoMFInt32.h>
#include <Inventor/fields/SoSFInt32.h>
#include <Inventor/fields/SoSFString.h>
#include "SoXipEditText2.h"
#include "SoXipOverlayElement.h"
```

Include dependency graph for SoXipShape.h:

This graph shows which files directly or indirectly include this file:

Classes

- class SoXipShape

  Base class for all the shapes.

5.43.1 Detailed Description

Base class for all the shapes.

Author:

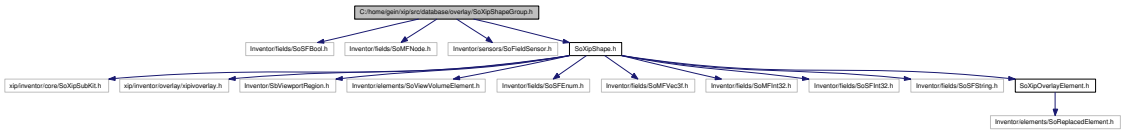
Julien Gein

5.44 C:/home/gein/xip/src/database/overlay/SoXipShapeGroup.h File Reference

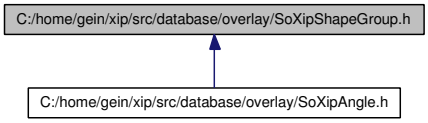
Base class for all compound shapes.

```
#include <Inventor/fields/SoSFBool.h>
#include <Inventor/fields/SoMFNode.h>
#include <Inventor/sensors/SoFieldSensor.h>
#include "SoXipShape.h"
```

Include dependency graph for SoXipShapeGroup.h:



This graph shows which files directly or indirectly include this file:



Classes

- class SoXipShapeGroup

5.44.1 Detailed Description

Base class for all compound shapes.

Author:

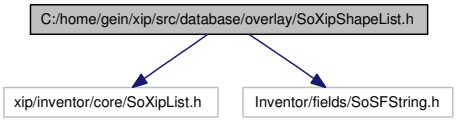
Julien Gein

5.45 C:/home/gein/xip/src/database/overlay/SoXipShapeList.h File Reference

List of shapes.

```
#include <xip/inventor/core/SoXipList.h>
#include <Inventor/fields/SoSFString.h>
```

Include dependency graph for SoXipShapeList.h:



5.45.1 Detailed Description

List of shapes.

Author:

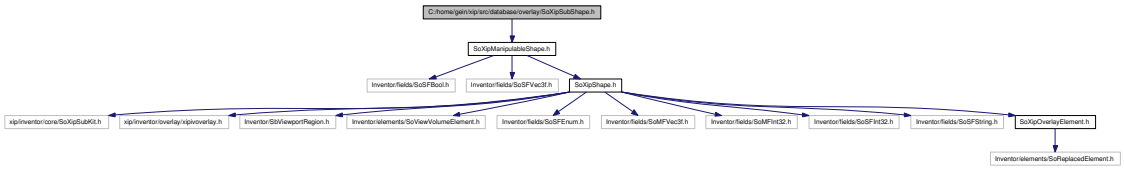
Julien Gein

5.46 C:/home/gein/xip/src/database/overlay/SoXipSubShape.h File Reference

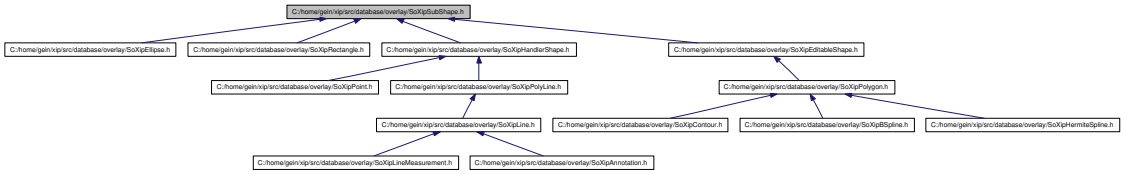
Contains some utility macros for safe creation of nodes derived from SoXipShape.

```
#include "SoXipManipulableShape.h"
```

Include dependency graph for SoXipSubShape.h:



This graph shows which files directly or indirectly include this file:



Defines

- #define SO_XIP_SHAPE_FIELD(FieldName, FieldType)

  Declare a shape field and its associated sensor.

- #define SO_XIP_SHAPE_ADD_FIELD(FieldName, FieldValue)

  Add a field to a rad shape.

5.46.1 Detailed Description

Contains some utility macros for safe creation of nodes derived from SoXipShape.

Author:

Julien Gein

5.46.2 Define Documentation

5.46.2.1 #define SO_XIP_SHAPE_ADD_FIELD(FieldName, FieldValue)

Value:

```
SO_NODE_ADD_FIELD( FieldName, FieldValue ); \
        m##FieldName##Sensor.setFunction( SoXipManipulableShape::invalidateGeometriesCB ); \
        m##FieldName##Sensor.setData( this ); \
```

```
        m##FieldName##Sensor.setPriority(0); \
        m##FieldName##Sensor.attach( &FieldName );
```

Add a field to a rad shape.

This automatically creates sensors to the given field Only call this macro, if the shape geometries need to be updated when the field value is changed.

5.46.2.2 #define SO_XIP_SHAPE_FIELD(FieldName, FieldType)

Value:

```
public: \
                FieldType FieldName; \
        private: \
                SoFieldSensor m##FieldName##Sensor;
```

Declare a shape field and its associated sensor.

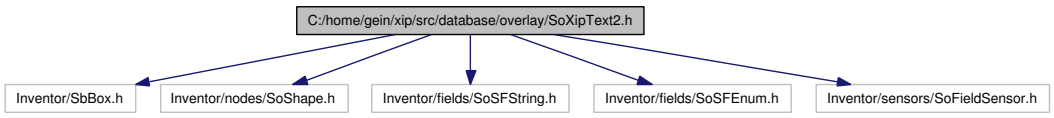
Automatically declares a sensor to be used for the specific shape.

5.47 C:/home/gein/xip/src/database/overlay/SoXipText2.h File Reference

Text node.

```
#include <Inventor/SbBox.h>
#include <Inventor/nodes/SoShape.h>
#include <Inventor/fields/SoSFString.h>
#include <Inventor/fields/SoSFEnum.h>
#include <Inventor/sensors/SoFieldSensor.h>
```

Include dependency graph for SoXipText2.h:



Classes

- class SoXipText2

5.47.1 Detailed Description

Text node.

Author:

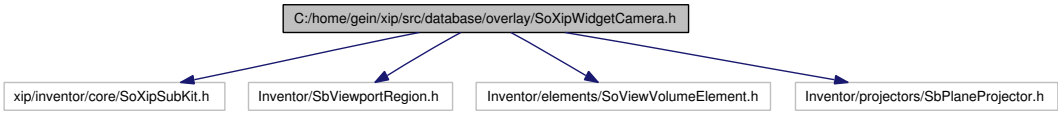
Thomas Moeller

5.48 C:/home/gein/xip/src/database/overlay/SoXipWidgetCamera.h File Reference

Simple camera for 2D objects (preserve scale).

```
#include <xip/inventor/core/SoXipSubKit.h>
#include <Inventor/SbViewportRegion.h>
#include <Inventor/elements/SoViewVolumeElement.h>
#include <Inventor/projectors/SbPlaneProjector.h>
```

Include dependency graph for SoXipWidgetCamera.h:



Classes

- class SoXipWidgetCamera

5.48.1 Detailed Description

Simple camera for 2D objects (preserve scale).

Author:

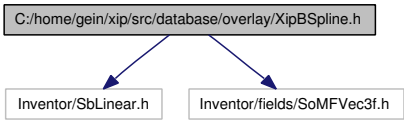
Julien Gein

5.49 C:/home/gein/xip/src/database/overlay/XipBSpline.h File Reference

Contains the declaration of the BSpline utility class.

```
#include <Inventor/SbLinear.h>
#include <Inventor/fields/SoMFVec3f.h>
```

Include dependency graph for XipBSpline.h:



Classes

- class XipBSpline

5.49.1 Detailed Description

Contains the declaration of the BSpline utility class.

Author:

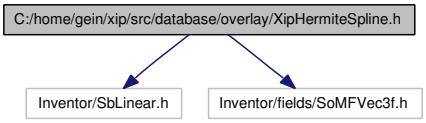
Julien Gein

5.50 C:/home/gein/xip/src/database/overlay/XipHermiteSpline.h File Reference

Contains the declaration of the HermiteSpline utility class.

```
#include <Inventor/SbLinear.h>
#include <Inventor/fields/SoMFVec3f.h>
```

Include dependency graph for XipHermiteSpline.h:



Classes

- class XipHermiteSpline

5.50.1 Detailed Description

Contains the declaration of the HermiteSpline utility class.

Author:

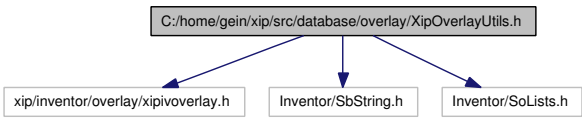
Julien Gein

5.51 C:/home/gein/xip/src/database/overlay/XipOverlayUtils.h File Reference

Contains utilities to facilitate overlay loading, saving, and retrieving.

```
#include <xip/inventor/overlay/xipivoverlay.h>
#include <Inventor/SbString.h>
#include <Inventor/SoLists.h>
```

Include dependency graph for XipOverlayUtils.h:



5.51.1 Detailed Description

Contains utilities to facilitate overlay loading, saving, and retrieving.

Author:

Julien Gein

Index