
OpenSceneGraph Version 2.9.6

# osgTerrain::

## Reference Manual

Generated for OpenSceneGraph by Doxygen, Graphviz and MikTex

Wed Jan 6 20:51:04 2010

# Contents

# Chapter 1

# Main Page

---

The OpenSceneGraph exists as a number of modules, each sitting in its own library, enclosed within its own namespace. At the very core lies the osg library. This contains the OpenSceneGraph's central classes and, at the bare minimum, it is all users need to write an OpenSceneGraph program in C++.

Around and alongside this sit other supporting libraries, such as osgUtil (containing visitors for app traversals, cull traversals, scene graph optimizers and so on), osgDB (for handling plug-ins, shared library loading, database reading and writing and the like), osgText, osgParticle, etc.

Extensive online documentation is available from the OSG Support section to help in using Open Scene Graph.

The project's original reference guides generated by Doxygen from the source code may be downloaded as a single file from the OSG Reference Guides section.

To download source code, binaries, dependencies and sample datasets visit the OSG Download page.

For more about dependencies see the OSG Dependencies page.

The documentation you are looking at can be downloaded from www.3draum.ch.

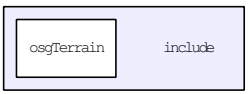
Enjoy!

# Chapter 2

# Directory Documentation

---

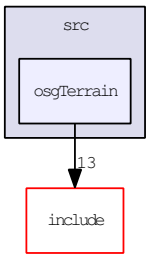
## 2.1 include/ Directory Reference




### Directories

- directory **osgTerrain**

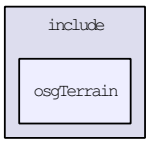
## 2.2 src/osgTerrain/ Directory Reference




### Files

- file **GeometryTechnique.cpp**
- file **Layer.cpp**
- file **Locator.cpp**
- file **Terrain.cpp**
- file **TerrainTechnique.cpp**
- file **TerrainTile.cpp**
- file **Version.cpp**

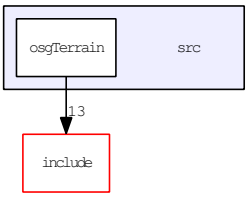
## 2.3 include/osgTerrain/ Directory Reference




### Files

- file **Export**
- file **GeometryTechnique**
- file **Layer**
- file **Locator**
- file **mainpage.h**
- file **Terrain**
- file **TerrainTechnique**
- file **TerrainTile**
- file **ValidDataOperator**
- file **Version**

## 2.4 src/ Directory Reference




### Directories

- directory **osgTerrain**

# Chapter 3

# Namespace Documentation

---

## 3.1 osgTerrain Namespace Reference

The **osgTerrain** (p. 7) library is a NodeKit that provides geospecifc terrain rendering support.

### Classes

- class **CompositeLayer**
- class **ContourLayer**
- class **GeometryTechnique**
- class **HeightFieldLayer**
- class **ImageLayer**
- class **Layer**
- class **Locator**
- struct **NoDataValue**
- class **ProxyLayer**
- class **SwitchLayer**
- class **Terrain**

  ***Terrain*** *(p. 42) provides a framework for loosely coupling height field data with height rendering algorithms.*

- class **TerrainTechnique**
- class **TerrainTile**

  ***Terrain*** *(p. 42) provides a framework for loosely coupling height field data with height rendering algorithms.*

- class **TileID**
- struct **ValidDataOperator**
- struct **ValidRange**
- class **WhiteListTileLoadedCallback**

  *Helper callback for managing optional sets of layers, that loading of is deffered to this callback, with this callback working out which layers to load, and how to create fallback versions of the layers.*

### Functions

- OSGTERRAIN_EXPORT std::string **createCompondSetNameAndFileName** (const std::string &setname, const std::string &filename)

  *Create a compound string in the form set:setname:filename, or just filename if setname is "".*

- OSGTERRAIN_EXPORT void **extractSetNameAndFileName** (const std::string &compoundstring, std::string &setname, std::string &filename)

  *Extact the setname and filename from a compound string in the from set:setname:filename".*

### 3.1.1 Detailed Description

The **osgTerrain** (p. 7) library is a NodeKit that provides geospecifc terrain rendering support.

### 3.1.2 Function Documentation

#### 3.1.2.1 std::string createCompondSetNameAndFileName (const std::string & *setname*, const std::string & *filename*)

Create a compound string in the form set:setname:filename, or just filename if setname is "".

#### 3.1.2.2 void extractSetNameAndFileName (const std::string & *compoundstring*, std::string & *setname*, std::string & *filename*)

Extact the setname and filename from a compound string in the from set:setname:filename". Returns a setname of "" when non set:setname: entry is present.

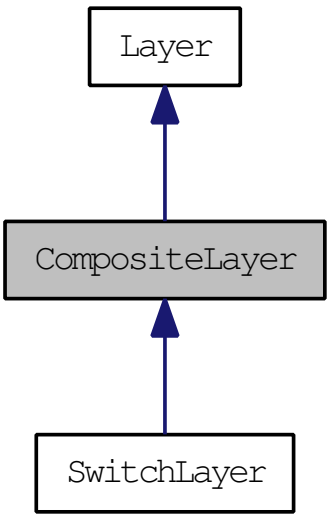
# Chapter 4

# Class Documentation

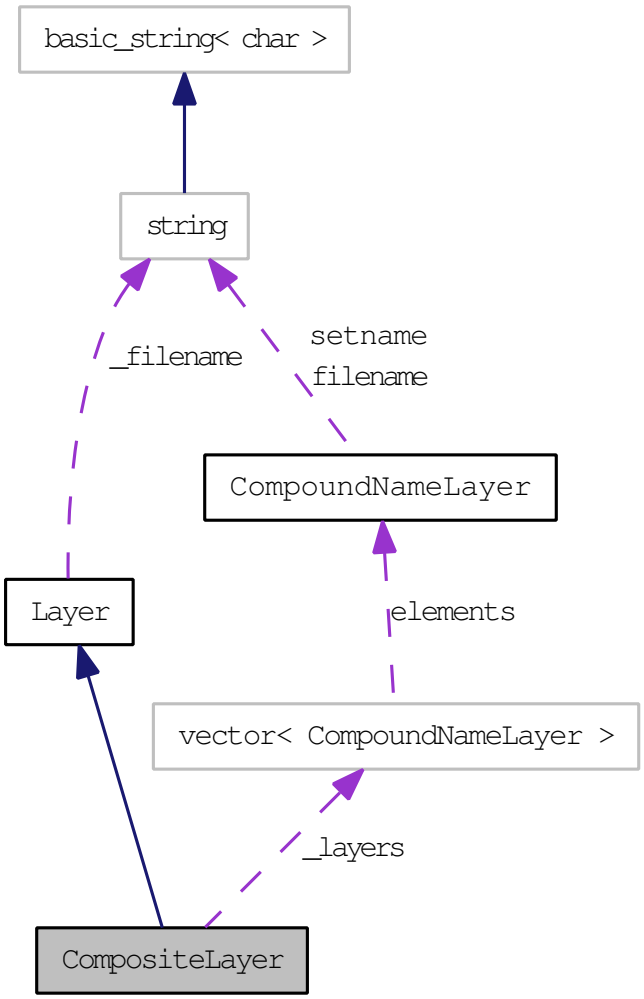
---

## 4.1 CompositeLayer Class Reference

Inheritance diagram for CompositeLayer:



Collaboration diagram for CompositeLayer:




## Classes

- struct **CompoundNameLayer**

## Public Member Functions

- **CompositeLayer** (const **CompositeLayer** &compositeLayer, const osg::CopyOp &copyop=osg::CopyOp::SHALLOW_COPY)

  *Copy constructor using CopyOp to manage deep vs shallow copy.*

- **CompositeLayer** ()
- void **addLayer** (**Layer** *layer)
- void **addLayer** (const std::string &setname, const std::string &filename)
- void **addLayer** (const std::string &compoundname)
- void **clear** ()
- std::string **getCompoundName** (unsigned int i) const
- const std::string & **getFileName** (unsigned int i) const
- const **Layer** * **getLayer** (unsigned int i) const
- **Layer** * **getLayer** (unsigned int i)
- unsigned int **getNumLayers** () const
- const std::string & **getSetName** (unsigned int i) const
- const std::string & **getSetName** () const
- **META_Object** (osgTerrain, **CompositeLayer**)

- void **removeLayer** (unsigned int i)
- void **setCompoundName** (unsigned int i, const std::string &compoundname)
- void **setFileName** (unsigned int i, const std::string &filename)
- void **setLayer** (unsigned int i, **Layer** ∗layer)
- void **setSetName** (unsigned int i, const std::string &setname)
- void **setSetName** (const std::string &setname)

### Protected Types

- typedef std::vector< **CompoundNameLayer** > **Layers**

### Protected Member Functions

- virtual ∼**CompositeLayer** ()

### Protected Attributes

- **Layers _layers**

## 4.1.1 Member Typedef Documentation

#### 4.1.1.1 typedef std::vector< CompoundNameLayer > Layers [protected]

## 4.1.2 Constructor & Destructor Documentation

#### 4.1.2.1 CompositeLayer ()

#### 4.1.2.2 CompositeLayer (const CompositeLayer & *compositeLayer*, const osg::CopyOp & *copyop* = osg::CopyOp::SHALLOW_COPY)

Copy constructor using CopyOp to manage deep vs shallow copy.

#### 4.1.2.3 virtual ∼CompositeLayer () [inline, protected, virtual]

## 4.1.3 Member Function Documentation

#### 4.1.3.1 void addLayer (Layer ∗ *layer*) [inline]

#### 4.1.3.2 void addLayer (const std::string & *setname*, const std::string & *filename*)

#### 4.1.3.3 void addLayer (const std::string & *compoundname*)

#### 4.1.3.4 void clear ()

#### 4.1.3.5 std::string getCompoundName (unsigned int *i*) const

#### 4.1.3.6 const std::string& getFileName (unsigned int *i*) const [inline]

#### 4.1.3.7 const Layer∗ getLayer (unsigned int *i*) const [inline]

#### 4.1.3.8 Layer∗ getLayer (unsigned int *i*) [inline]

#### 4.1.3.9 unsigned int getNumLayers () const [inline]

#### 4.1.3.10 const std::string& getSetName (unsigned int *i*) const [inline]

#### 4.1.3.11 const std::string& getSetName () const [inline]

Reimplemented from **Layer** (p. 29).

#### 4.1.3.12 META_Object (osgTerrain, CompositeLayer)

#### 4.1.3.13 void removeLayer (unsigned int *i*) [inline]

#### 4.1.3.14 void setCompoundName (unsigned int *i*, const std::string & *compoundname*)

#### 4.1.3.15 void setFileName (unsigned int *i*, const std::string & *filename*) [inline]

#### 4.1.3.16 void setLayer (unsigned int *i*, Layer * *layer*) [inline]

#### 4.1.3.17 void setSetName (unsigned int *i*, const std::string & *setname*) [inline]

#### 4.1.3.18 void setSetName (const std::string & *setname*) [inline]

Reimplemented from **Layer** (p. 30).

### 4.1.4 Member Data Documentation

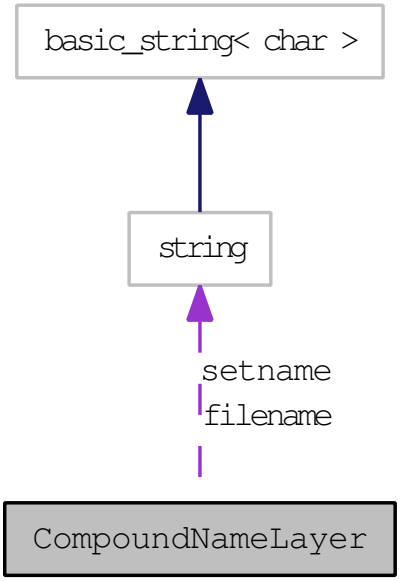
#### 4.1.4.1 Layers _layers [protected]

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

- **Layer**
- **Layer.cpp**

## 4.2 CompoundNameLayer Struct Reference

Collaboration diagram for CompoundNameLayer:




### Public Member Functions

- **CompoundNameLayer** (const std::string &sn, const std::string &fn, **Layer** *l)
- **CompoundNameLayer** (const **CompoundNameLayer** &cnl)
- **CompoundNameLayer** ()
- **CompoundNameLayer** & **operator=** (const **CompoundNameLayer** &cnl)

### Public Attributes

- std::string **filename**
- osg::ref_ptr< **Layer** > **layer**
- std::string **setname**

### 4.2.1 Constructor & Destructor Documentation

#### 4.2.1.1 CompoundNameLayer () [inline]

#### 4.2.1.2 CompoundNameLayer (const CompoundNameLayer & *cnl*) [inline]

#### 4.2.1.3 CompoundNameLayer (const std::string & *sn*, const std::string & *fn*, Layer * *l*) [inline]

### 4.2.2 Member Function Documentation

#### 4.2.2.1 CompoundNameLayer& operator= (const CompoundNameLayer & *cnl*) [inline]

### 4.2.3 Member Data Documentation

#### 4.2.3.1 std::string filename

#### 4.2.3.2 osg::ref_ptr<Layer> layer

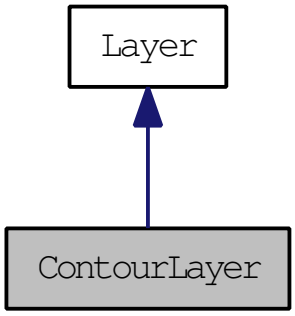
#### 4.2.3.3 std::string setname

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

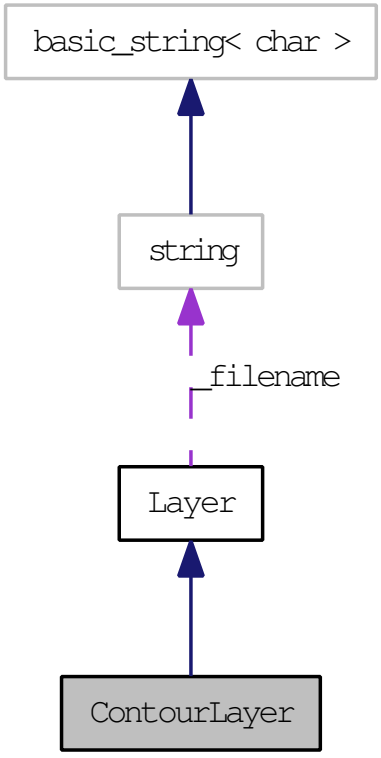
- **Layer**

## 4.3 ContourLayer Class Reference

Inheritance diagram for ContourLayer:




Collaboration diagram for ContourLayer:




### Public Member Functions

- **ContourLayer** (const **ContourLayer** &tfLayer, const osg::CopyOp &copyop=osg::CopyOp::SHALLOW_COPY)

  *Copy constructor using CopyOp to manage deep vs shallow copy.*

- **ContourLayer** (osg::TransferFunction1D *tf=0)
- virtual void **dirty** ()

  *increment the modified count.*

- virtual const osg::Image * **getImage** () const

  *Return const image associated with layer.*

- virtual osg::Image * **getImage** ()

*Return image associated with layer.*

- virtual unsigned int **getModifiedCount** () const
  *Get modified count value.*
- virtual unsigned int **getNumColumns** () const
- virtual unsigned int **getNumRows** () const
- const osg::TransferFunction1D * **getTransferFunction** () const
- osg::TransferFunction1D * **getTransferFunction** ()
- virtual bool **getValue** (unsigned int i, unsigned int j, osg::Vec4 &value) const
- virtual bool **getValue** (unsigned int i, unsigned int j, osg::Vec3 &value) const
- virtual bool **getValue** (unsigned int i, unsigned int j, osg::Vec2 &value) const
- virtual bool **getValue** (unsigned int i, unsigned int j, float &value) const
- **META_Object** (osgTerrain, **ContourLayer**)
- virtual void **setModifiedCount** (unsigned int value)
  *Set the modified count value.*
- void **setTransferFunction** (osg::TransferFunction1D *tf)
- virtual bool **transform** (float offset, float scale)

## Protected Member Functions

- virtual ~**ContourLayer** ()

## Protected Attributes

- osg::ref_ptr< osg::TransferFunction1D > **_tf**

### 4.3.1 Constructor & Destructor Documentation

#### 4.3.1.1 ContourLayer (osg::TransferFunction1D * *tf* = 0)

#### 4.3.1.2 ContourLayer (const ContourLayer & *tfLayer*, const osg::CopyOp & *copyop* = osg::CopyOp::SHALLOW_COPY)

Copy constructor using CopyOp to manage deep vs shallow copy.

#### 4.3.1.3 virtual ~ContourLayer () [inline, protected, virtual]

### 4.3.2 Member Function Documentation

#### 4.3.2.1 void dirty () [virtual]

increment the modified count. "

Reimplemented from **Layer** (p. 28).

#### 4.3.2.2 virtual const osg::Image* getImage () const [inline, virtual]

Return const image associated with layer.

Reimplemented from **Layer** (p. 28).

#### 4.3.2.3 virtual osg::Image* getImage () [inline, virtual]

Return image associated with layer.

Reimplemented from **Layer** (p. 28).

#### 4.3.2.4 unsigned int getModifiedCount () const [virtual]

Get modified count value.

Reimplemented from **Layer** (p. 29).

#### 4.3.2.5 virtual unsigned int getNumColumns () const [inline, virtual]

Reimplemented from **Layer** (p. 29).

#### 4.3.2.6 virtual unsigned int getNumRows () const [inline, virtual]

Reimplemented from **Layer** (p. 29).

#### 4.3.2.7 const osg::TransferFunction1D* getTransferFunction () const [inline]

#### 4.3.2.8 osg::TransferFunction1D* getTransferFunction () [inline]

#### 4.3.2.9 bool getValue (unsigned int *i*, unsigned int *j*, osg::Vec4 & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.3.2.10 bool getValue (unsigned int *i*, unsigned int *j*, osg::Vec3 & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.3.2.11 bool getValue (unsigned int *i*, unsigned int *j*, osg::Vec2 & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.3.2.12 bool getValue (unsigned int *i*, unsigned int *j*, float & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.3.2.13 META_Object (osgTerrain, ContourLayer)

#### 4.3.2.14 void setModifiedCount (unsigned int) [virtual]

Set the modified count value.

Reimplemented from **Layer** (p. 30).

#### 4.3.2.15 void setTransferFunction (osg::TransferFunction1D * *tf*)

#### 4.3.2.16 bool transform (float *offset*, float *scale*) [virtual]

Reimplemented from **Layer** (p. 30).

### 4.3.3 Member Data Documentation

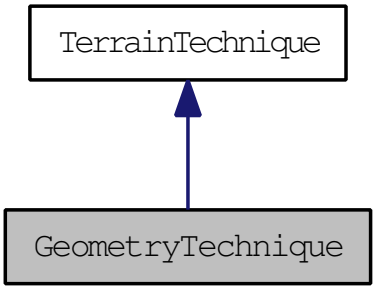
#### 4.3.3.1 osg::ref_ptr<osg::TransferFunction1D> _tf [protected]

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

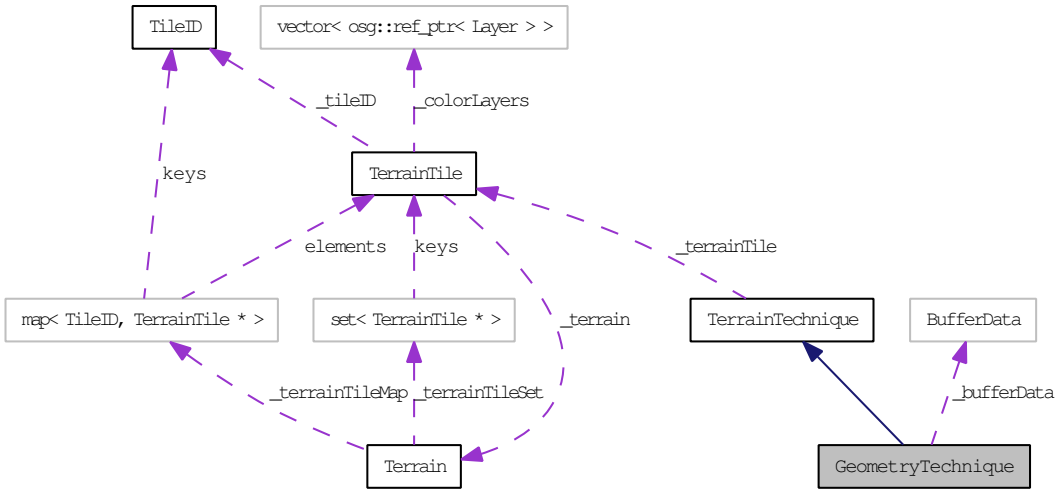
- **Layer**
- **Layer.cpp**

## 4.4 GeometryTechnique Class Reference

Inheritance diagram for GeometryTechnique:




Collaboration diagram for GeometryTechnique:




### Classes

- struct **BufferData**

### Public Types

- enum **FilterType** { **GAUSSIAN**, **SMOOTH**, **SHARPEN** }

### Public Member Functions

- **GeometryTechnique** (const **GeometryTechnique** &, const osg::CopyOp &copyop=osg::CopyOp::SHALLOW_COPY)

  *Copy constructor using CopyOp to manage deep vs shallow copy.*

- **GeometryTechnique** ()
- virtual void **applyColorLayers** ()
- virtual void **applyTransparency** ()
- virtual void **cleanSceneGraph** ()

  *Clean scene graph from any terrain technique specific nodes.*

- virtual osg::Vec3d **computeCenterModel** (**Locator** *masterLocator)
- virtual **Locator** * **computeMasterLocator** ()
- virtual void **cull** (osgUtil::CullVisitor *nv)

- virtual void **generateGeometry** (**Locator** ∗masterLocator, const osg::Vec3d &centerModel)
- float **getFilterBias** () const
- const osg::Matrix3 & **getFilterMatrix** () const
- osg::Matrix3 & **getFilterMatrix** ()
- float **getFilterWidth** () const
- virtual void **init** ()
- **META_Object** (osgTerrain, **GeometryTechnique**)
- virtual void **releaseGLObjects** (osg::State ∗=0) const

  *If State is non-zero, this function releases any associated OpenGL objects for the specified graphics context.*

- void **setFilterBias** (float filterBias)
- void **setFilterMatrix** (const osg::Matrix3 &matrix)
- void **setFilterMatrixAs** (**FilterType** filterType)
- void **setFilterWidth** (float filterWidth)
- virtual void **smoothGeometry** ()
- virtual void **traverse** (osg::NodeVisitor &nv)

  *Traverse the terain subgraph.*

- virtual void **update** (osgUtil::UpdateVisitor ∗nv)

### 4.4.1 Member Enumeration Documentation

#### 4.4.1.1 enum FilterType

**Enumerator:**

***GAUSSIAN***

***SMOOTH***

***SHARPEN***

### 4.4.2 Constructor & Destructor Documentation

#### 4.4.2.1 GeometryTechnique ()

#### 4.4.2.2 GeometryTechnique (const GeometryTechnique & *gt*, const osg::CopyOp & *copyop* = `osg::CopyOp::SHALLOW_COPY`)

Copy constructor using CopyOp to manage deep vs shallow copy.

### 4.4.3 Member Function Documentation

#### 4.4.3.1 void applyColorLayers () `[virtual]`

#### 4.4.3.2 void applyTransparency () `[virtual]`

#### 4.4.3.3 void cleanSceneGraph () `[virtual]`

Clean scene graph from any terrain technique specific nodes.

Reimplemented from **TerrainTechnique** (p. 46).

#### 4.4.3.4 osg::Vec3d computeCenterModel (Locator ∗ *masterLocator*) `[virtual]`

#### 4.4.3.5 Locator ∗ computeMasterLocator () `[virtual]`

#### 4.4.3.6 void cull (osgUtil::CullVisitor ∗ *nv*) `[virtual]`

Reimplemented from **TerrainTechnique** (p. 46).

#### 4.4.3.7 void generateGeometry (Locator * *masterLocator*, const osg::Vec3d & *centerModel*) [virtual]

#### 4.4.3.8 float getFilterBias () const [inline]

#### 4.4.3.9 const osg::Matrix3& getFilterMatrix () const [inline]

#### 4.4.3.10 osg::Matrix3& getFilterMatrix () [inline]

#### 4.4.3.11 float getFilterWidth () const [inline]

#### 4.4.3.12 void init () [virtual]

Reimplemented from **TerrainTechnique** (p. 46).

#### 4.4.3.13 META_Object (osgTerrain, GeometryTechnique)

#### 4.4.3.14 void releaseGLObjects (osg::State * *state* = 0) const [virtual]

If State is non-zero, this function releases any associated OpenGL objects for the specified graphics context. Otherwise, releases OpenGL objects for all graphics contexts.

Reimplemented from **TerrainTechnique** (p. 46).

#### 4.4.3.15 void setFilterBias (float *filterBias*)

#### 4.4.3.16 void setFilterMatrix (const osg::Matrix3 & *matrix*)

#### 4.4.3.17 void setFilterMatrixAs (FilterType *filterType*)

#### 4.4.3.18 void setFilterWidth (float *filterWidth*)

#### 4.4.3.19 void smoothGeometry () [virtual]

#### 4.4.3.20 void traverse (osg::NodeVisitor & *nv*) [virtual]

Traverse the terain subgraph.

Reimplemented from **TerrainTechnique** (p. 46).

#### 4.4.3.21 void update (osgUtil::UpdateVisitor * *nv*) [virtual]

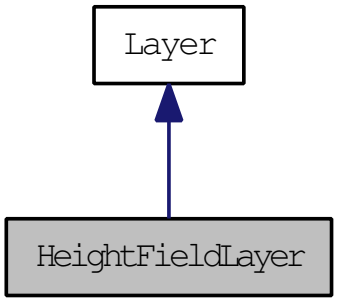
Reimplemented from **TerrainTechnique** (p. 46).

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

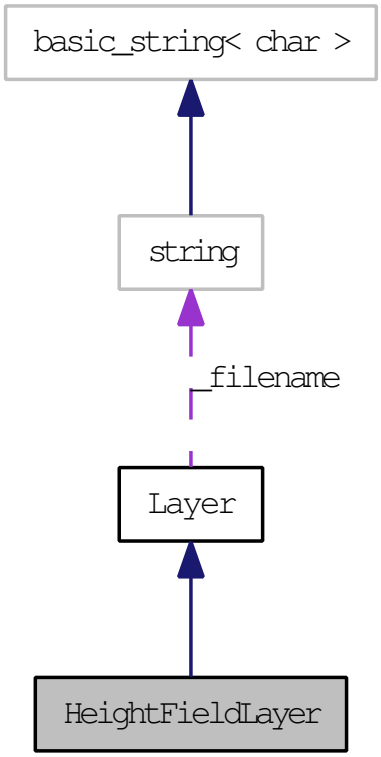
- **GeometryTechnique**
- **GeometryTechnique.cpp**

## 4.5 HeightFieldLayer Class Reference

Inheritance diagram for HeightFieldLayer:




Collaboration diagram for HeightFieldLayer:




### Public Member Functions

- **HeightFieldLayer** (const **HeightFieldLayer** &hfLayer, const osg::CopyOp &copyop=osg::CopyOp::SHALLOW_COPY)

  *Copy constructor using CopyOp to manage deep vs shallow copy.*

- **HeightFieldLayer** (osg::HeightField *hf=0)
- virtual void **dirty** ()

  *increment the modified count.*

- virtual const std::string & **getFileName** () const

  *Get the file name of the layer.*

- const osg::HeightField * **getHeightField** () const

- osg::HeightField * **getHeightField** ()
- virtual unsigned int **getModifiedCount** () const

  *Get modified count value.*

- virtual unsigned int **getNumColumns** () const
- virtual unsigned int **getNumRows** () const
- virtual bool **getValue** (unsigned int i, unsigned int j, osg::Vec4 &value) const
- virtual bool **getValue** (unsigned int i, unsigned int j, osg::Vec3 &value) const
- virtual bool **getValue** (unsigned int i, unsigned int j, osg::Vec2 &value) const
- virtual bool **getValue** (unsigned int i, unsigned int j, float &value) const
- **META_Object** (osgTerrain, **HeightFieldLayer**)
- void **setFileName** (const std::string &filename)

  *Set the file name of the data associated with this layer.*

- void **setHeightField** (osg::HeightField *hf)
- virtual void **setModifiedCount** (unsigned int value)

  *Set the modified count value.*

- virtual bool **transform** (float offset, float scale)

## Protected Member Functions

- virtual ~**HeightFieldLayer** ()

## Protected Attributes

- osg::ref_ptr< osg::HeightField > **_heightField**
- unsigned int **_modifiedCount**

### 4.5.1 Constructor & Destructor Documentation

#### 4.5.1.1 HeightFieldLayer (osg::HeightField * *hf* = 0)

#### 4.5.1.2 HeightFieldLayer (const HeightFieldLayer & *hfLayer*, const osg::CopyOp & *copyop* = osg::CopyOp::SHALLOW_COPY)

Copy constructor using CopyOp to manage deep vs shallow copy.

#### 4.5.1.3 virtual ~HeightFieldLayer () [inline, protected, virtual]

### 4.5.2 Member Function Documentation

#### 4.5.2.1 void dirty () [virtual]

increment the modified count. "

Reimplemented from **Layer** (p. 28).

#### 4.5.2.2 virtual const std::string& getFileName () const [inline, virtual]

Get the file name of the layer.

Reimplemented from **Layer** (p. 28).

#### 4.5.2.3 const osg::HeightField* getHeightField () const [inline]

#### 4.5.2.4 osg::HeightField* getHeightField () [inline]

#### 4.5.2.5 unsigned int getModifiedCount () const [virtual]

Get modified count value.

Reimplemented from **Layer** (p. 29).

#### 4.5.2.6 virtual unsigned int getNumColumns () const [inline, virtual]

Reimplemented from **Layer** (p. 29).

#### 4.5.2.7 virtual unsigned int getNumRows () const [inline, virtual]

Reimplemented from **Layer** (p. 29).

#### 4.5.2.8 bool getValue (unsigned int *i*, unsigned int *j*, osg::Vec4 & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.5.2.9 bool getValue (unsigned int *i*, unsigned int *j*, osg::Vec3 & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.5.2.10 bool getValue (unsigned int *i*, unsigned int *j*, osg::Vec2 & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.5.2.11 bool getValue (unsigned int *i*, unsigned int *j*, float & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.5.2.12 META_Object (osgTerrain, HeightFieldLayer)

#### 4.5.2.13 void setFileName (const std::string & *filename*) [inline, virtual]

Set the file name of the data associated with this layer.

Reimplemented from **Layer** (p. 29).

#### 4.5.2.14 void setHeightField (osg::HeightField * *hf*)

#### 4.5.2.15 void setModifiedCount (unsigned int) [virtual]

Set the modified count value.

Reimplemented from **Layer** (p. 30).

#### 4.5.2.16 bool transform (float *offset*, float *scale*) [virtual]

Reimplemented from **Layer** (p. 30).

### 4.5.3 Member Data Documentation

#### 4.5.3.1 osg::ref_ptr<osg::HeightField> _heightField [protected]

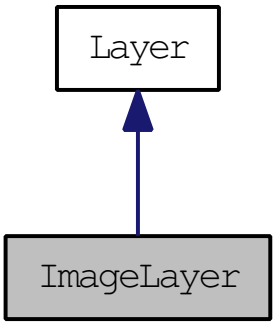
#### 4.5.3.2 unsigned int _modifiedCount [protected]

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

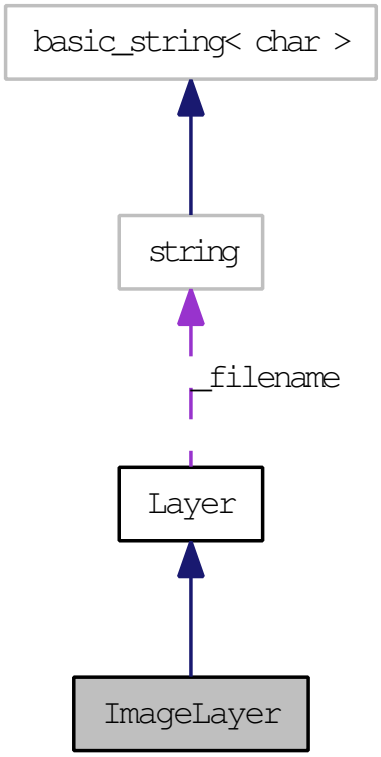
- **Layer**
- **Layer.cpp**

## 4.6 ImageLayer Class Reference

Inheritance diagram for ImageLayer:




Collaboration diagram for ImageLayer:




### Public Member Functions

- **ImageLayer** (const **ImageLayer** &imageLayer, const osg::CopyOp &copyop=osg::CopyOp::SHALLOW_COPY)

  *Copy constructor using CopyOp to manage deep vs shallow copy.*

- **ImageLayer** (osg::Image *image=0)
- virtual void **dirty** ()

  *increment the modified count.*

- virtual const std::string & **getFileName** () const

  *Get the file name of the layer.*

- virtual const osg::Image * **getImage** () const
  *Return const image associated with layer.*
- virtual osg::Image * **getImage** ()
  *Return image associated with layer.*
- virtual unsigned int **getModifiedCount** () const
  *Get modified count value.*
- virtual unsigned int **getNumColumns** () const
- virtual unsigned int **getNumRows** () const
- virtual bool **getValue** (unsigned int i, unsigned int j, osg::Vec4 &value) const
- virtual bool **getValue** (unsigned int i, unsigned int j, osg::Vec3 &value) const
- virtual bool **getValue** (unsigned int i, unsigned int j, osg::Vec2 &value) const
- virtual bool **getValue** (unsigned int i, unsigned int j, float &value) const
- **META_Object** (osgTerrain, **ImageLayer**)
- void **setFileName** (const std::string &filename)
  *Set the file name of the data associated with this layer.*
- void **setImage** (osg::Image *image)
- virtual void **setModifiedCount** (unsigned int value)
  *Set the modified count value.*
- virtual bool **transform** (float offset, float scale)

### Protected Member Functions

- virtual ~**ImageLayer** ()

### Protected Attributes

- osg::ref_ptr< osg::Image > **_image**

## 4.6.1 Constructor & Destructor Documentation

#### 4.6.1.1 ImageLayer (osg::Image * *image* = 0)

#### 4.6.1.2 ImageLayer (const ImageLayer & *imageLayer*, const osg::CopyOp & *copyop* = osg::CopyOp::SHALLOW_COPY)

Copy constructor using CopyOp to manage deep vs shallow copy.

#### 4.6.1.3 virtual ~ImageLayer () [inline, protected, virtual]

## 4.6.2 Member Function Documentation

#### 4.6.2.1 void dirty () [virtual]

increment the modified count. "

Reimplemented from **Layer** (p. 28).

#### 4.6.2.2 virtual const std::string& getFileName () const [inline, virtual]

Get the file name of the layer.

Reimplemented from **Layer** (p. 28).

#### 4.6.2.3 virtual const osg::Image* getImage () const [inline, virtual]

Return const image associated with layer.

Reimplemented from **Layer** (p. 28).

#### 4.6.2.4 virtual osg::Image* getImage () [inline, virtual]

Return image associated with layer.

Reimplemented from **Layer** (p. 28).

#### 4.6.2.5 unsigned int getModifiedCount () const [virtual]

Get modified count value.

Reimplemented from **Layer** (p. 29).

#### 4.6.2.6 virtual unsigned int getNumColumns () const [inline, virtual]

Reimplemented from **Layer** (p. 29).

#### 4.6.2.7 virtual unsigned int getNumRows () const [inline, virtual]

Reimplemented from **Layer** (p. 29).

#### 4.6.2.8 bool getValue (unsigned int *i*, unsigned int *j*, osg::Vec4 & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.6.2.9 bool getValue (unsigned int *i*, unsigned int *j*, osg::Vec3 & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.6.2.10 bool getValue (unsigned int *i*, unsigned int *j*, osg::Vec2 & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.6.2.11 bool getValue (unsigned int *i*, unsigned int *j*, float & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.6.2.12 META_Object (osgTerrain, ImageLayer)

#### 4.6.2.13 void setFileName (const std::string & *filename*) [inline, virtual]

Set the file name of the data associated with this layer.

Reimplemented from **Layer** (p. 29).

#### 4.6.2.14 void setImage (osg::Image * *image*)

#### 4.6.2.15 void setModifiedCount (unsigned int) [virtual]

Set the modified count value.

Reimplemented from **Layer** (p. 30).

#### 4.6.2.16 bool transform (float *offset*, float *scale*) [virtual]

Reimplemented from **Layer** (p. 30).

### 4.6.3 Member Data Documentation

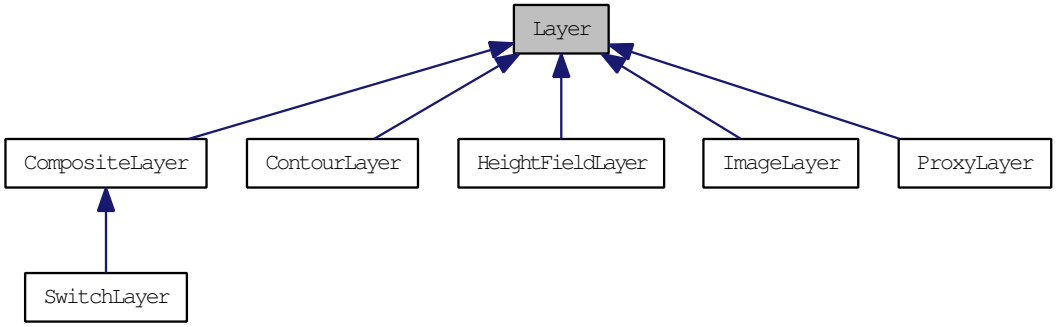
#### 4.6.3.1 osg::ref_ptr<osg::Image> _image [protected]

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

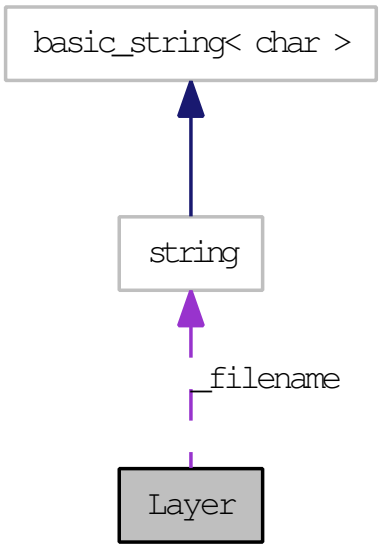
- **Layer**
- **Layer.cpp**

## 4.7 Layer Class Reference

Inheritance diagram for Layer:




Collaboration diagram for Layer:




### Public Member Functions

- **Layer** (const **Layer** &, const osg::CopyOp &copyop=osg::CopyOp::SHALLOW_COPY)
  *Copy constructor using CopyOp to manage deep vs shallow copy.*
- **Layer** ()
- virtual osg::BoundingSphere **computeBound** (bool treatAsElevationLayer) const
- void **computeIndices** (double ndc_x, double ndc_y, unsigned int &i, unsigned int &j, double &ir, double &jr)
- virtual void **dirty** ()
  *increment the modified count.*
- std::string **getCompoundName** () const
  *Return the compound name of the layer in the form set::name::filename string.*
- const osg::Vec4 & **getDefaultValue** () const
- virtual const std::string & **getFileName** () const
  *Get the file name of the layer.*
- virtual const osg::Image * **getImage** () const
  *Return const image associated with layer if supported.*

- virtual osg::Image * **getImage** ()

  *Return image associated with layer if supported.*

- bool **getInterpolatedValue** (double ndc_x, double ndc_y, float &value)
- const **Locator** * **getLocator** () const
- **Locator** * **getLocator** ()
- osg::Texture::FilterMode **getMagFilter** () const

  *Get the magnification texture filter to use when do texture associated with this layer.*

- unsigned int **getMaxLevel** () const
- osg::Texture::FilterMode **getMinFilter** () const

  *Get the minification texture filter to use when do texture associated with this layer.*

- unsigned int **getMinLevel** () const
- virtual unsigned int **getModifiedCount** () const

  *Get modified count value.*

- virtual unsigned int **getNumColumns** () const
- virtual unsigned int **getNumRows** () const
- const std::string & **getSetName** () const
- const **ValidDataOperator** * **getValidDataOperator** () const
- **ValidDataOperator** * **getValidDataOperator** ()
- bool **getValidValue** (unsigned int i, unsigned int j, osg::Vec4 &value) const
- bool **getValidValue** (unsigned int i, unsigned int j, osg::Vec3 &value) const
- bool **getValidValue** (unsigned int i, unsigned int j, osg::Vec2 &value) const
- bool **getValidValue** (unsigned int i, unsigned int j, float &value) const
- virtual bool **getValue** (unsigned int, unsigned int, osg::Vec4 &) const
- virtual bool **getValue** (unsigned int, unsigned int, osg::Vec3 &) const
- virtual bool **getValue** (unsigned int, unsigned int, osg::Vec2 &) const
- virtual bool **getValue** (unsigned int, unsigned int, float &) const
- **META_Object** (osgTerrain, **Layer**)
- void **setDefaultValue** (const osg::Vec4 &value)
- virtual void **setFileName** (const std::string &filename)

  *Set the file name of the data associated with this layer.*

- void **setLocator** (**Locator** *locator)
- void **setMagFilter** (osg::Texture::FilterMode filter)

  *Set the magniification texture filter to use when do texture associated with this layer.*

- void **setMaxLevel** (unsigned int maxLevel)
- void **setMinFilter** (osg::Texture::FilterMode filter)

  *Set the minification texture filter to use when do texture associated with this layer.*

- void **setMinLevel** (unsigned int minLevel)
- virtual void **setModifiedCount** (unsigned int)

  *Set the modified count value.*

- void **setSetName** (const std::string &setname)
- void **setValidDataOperator** (**ValidDataOperator** *validDataOp)
- virtual bool **transform** (float, float)

## Protected Member Functions

- virtual ~**Layer** ()

### Protected Attributes

- osg::Vec4 **_defaultValue**
- std::string **_filename**
- osg::ref_ptr< **Locator** > **_locator**
- osg::Texture::FilterMode **_magFilter**
- unsigned int **_maxLevel**
- osg::Texture::FilterMode **_minFilter**
- unsigned int **_minLevel**
- osg::ref_ptr< **ValidDataOperator** > **_validDataOperator**

## 4.7.1 Constructor & Destructor Documentation

#### 4.7.1.1 Layer ()

#### 4.7.1.2 Layer (const Layer & *layer*, const osg::CopyOp & *copyop* = `osg::CopyOp::SHALLOW_COPY`)

Copy constructor using CopyOp to manage deep vs shallow copy.

#### 4.7.1.3 ~Layer () `[protected, virtual]`

## 4.7.2 Member Function Documentation

#### 4.7.2.1 osg::BoundingSphere computeBound (bool *treatAsElevationLayer*) const `[virtual]`

Reimplemented in **ProxyLayer** (p. 37).

#### 4.7.2.2 void computeIndices (double *ndc_x*, double *ndc_y*, unsigned int & *i*, unsigned int & *j*, double & *ir*, double & *jr*) `[inline]`

#### 4.7.2.3 virtual void dirty () `[inline, virtual]`

increment the modified count. "

Reimplemented in **ImageLayer** (p. 24), **ContourLayer** (p. 15), **HeightFieldLayer** (p. 21), and **ProxyLayer** (p. 38).

#### 4.7.2.4 std::string getCompoundName () const `[inline]`

Return the compound name of the layer in the form set::name::filename string.

#### 4.7.2.5 const osg::Vec4& getDefaultValue () const `[inline]`

#### 4.7.2.6 virtual const std::string& getFileName () const `[inline, virtual]`

Get the file name of the layer.

Reimplemented in **ImageLayer** (p. 24), **HeightFieldLayer** (p. 21), and **ProxyLayer** (p. 38).

#### 4.7.2.7 virtual const osg::Image* getImage () const `[inline, virtual]`

Return const image associated with layer if supported.

Reimplemented in **ImageLayer** (p. 24), **ContourLayer** (p. 15), **ProxyLayer** (p. 38), and **SwitchLayer** (p. 41).

#### 4.7.2.8 virtual osg::Image* getImage () `[inline, virtual]`

Return image associated with layer if supported.

Reimplemented in **ImageLayer** (p. 25), **ContourLayer** (p. 15), **ProxyLayer** (p. 38), and **SwitchLayer** (p. 41).

#### 4.7.2.9 bool getInterpolatedValue (double *ndc_x*, double *ndc_y*, float & *value*) `[inline]`

#### 4.7.2.10 const Locator* getLocator () const `[inline]`

#### 4.7.2.11 Locator* getLocator () `[inline]`

#### 4.7.2.12 osg::Texture::FilterMode getMagFilter () const `[inline]`

Get the magnification texture filter to use when do texture associated with this layer.

#### 4.7.2.13 unsigned int getMaxLevel () const [inline]

#### 4.7.2.14 osg::Texture::FilterMode getMinFilter () const [inline]

Get the minification texture filter to use when do texture associated with this layer.

#### 4.7.2.15 unsigned int getMinLevel () const [inline]

#### 4.7.2.16 virtual unsigned int getModifiedCount () const [inline, virtual]

Get modified count value.

Reimplemented in **ImageLayer** (p. 25), **ContourLayer** (p. 15), **HeightFieldLayer** (p. 21), and **ProxyLayer** (p. 38).

#### 4.7.2.17 virtual unsigned int getNumColumns () const [inline, virtual]

Reimplemented in **ImageLayer** (p. 25), **ContourLayer** (p. 15), **HeightFieldLayer** (p. 21), and **ProxyLayer** (p. 38).

#### 4.7.2.18 virtual unsigned int getNumRows () const [inline, virtual]

Reimplemented in **ImageLayer** (p. 25), **ContourLayer** (p. 16), **HeightFieldLayer** (p. 22), and **ProxyLayer** (p. 38).

#### 4.7.2.19 const std::string& getSetName () const [inline]

Reimplemented in **CompositeLayer** (p. 11).

#### 4.7.2.20 const ValidDataOperator* getValidDataOperator () const [inline]

#### 4.7.2.21 ValidDataOperator* getValidDataOperator () [inline]

#### 4.7.2.22 bool getValidValue (unsigned int *i*, unsigned int *j*, osg::Vec4 & *value*) const [inline]

#### 4.7.2.23 bool getValidValue (unsigned int *i*, unsigned int *j*, osg::Vec3 & *value*) const [inline]

#### 4.7.2.24 bool getValidValue (unsigned int *i*, unsigned int *j*, osg::Vec2 & *value*) const [inline]

#### 4.7.2.25 bool getValidValue (unsigned int *i*, unsigned int *j*, float & *value*) const [inline]

#### 4.7.2.26 virtual bool getValue (unsigned int, unsigned int, osg::Vec4 &) const [inline, virtual]

Reimplemented in **ImageLayer** (p. 25), **ContourLayer** (p. 16), **HeightFieldLayer** (p. 22), and **ProxyLayer** (p. 38).

#### 4.7.2.27 virtual bool getValue (unsigned int, unsigned int, osg::Vec3 &) const [inline, virtual]

Reimplemented in **ImageLayer** (p. 25), **ContourLayer** (p. 16), **HeightFieldLayer** (p. 22), and **ProxyLayer** (p. 38).

#### 4.7.2.28 virtual bool getValue (unsigned int, unsigned int, osg::Vec2 &) const [inline, virtual]

Reimplemented in **ImageLayer** (p. 25), **ContourLayer** (p. 16), **HeightFieldLayer** (p. 22), and **ProxyLayer** (p. 38).

#### 4.7.2.29 virtual bool getValue (unsigned int, unsigned int, float &) const [inline, virtual]

Reimplemented in **ImageLayer** (p. 25), **ContourLayer** (p. 16), **HeightFieldLayer** (p. 22), and **ProxyLayer** (p. 38).

#### 4.7.2.30 META_Object (osgTerrain, Layer)

#### 4.7.2.31 void setDefaultValue (const osg::Vec4 & *value*) [inline]

#### 4.7.2.32 virtual void setFileName (const std::string & *filename*) [inline, virtual]

Set the file name of the data associated with this layer.

Reimplemented in **ImageLayer** (p. 25), **HeightFieldLayer** (p. 22), and **ProxyLayer** (p. 38).

#### 4.7.2.33 void setLocator (Locator * *locator*) [inline]

#### 4.7.2.34 void setMagFilter (osg::Texture::FilterMode *filter*) [inline]

Set the magniification texture filter to use when do texture associated with this layer.

#### 4.7.2.35 void setMaxLevel (unsigned int *maxLevel*) [inline]

#### 4.7.2.36 void setMinFilter (osg::Texture::FilterMode *filter*) [inline]

Set the minification texture filter to use when do texture associated with this layer.

#### 4.7.2.37 void setMinLevel (unsigned int *minLevel*) [inline]

#### 4.7.2.38 virtual void setModifiedCount (unsigned int) [inline, virtual]

Set the modified count value.

Reimplemented in **ImageLayer** (p. 25), **ContourLayer** (p. 16), **HeightFieldLayer** (p. 22), and **ProxyLayer** (p. 39).

#### 4.7.2.39 void setSetName (const std::string & *setname*) [inline]

Reimplemented in **CompositeLayer** (p. 12).

#### 4.7.2.40 void setValidDataOperator (ValidDataOperator * *validDataOp*) [inline]

#### 4.7.2.41 virtual bool transform (float, float) [inline, virtual]

Reimplemented in **ImageLayer** (p. 25), **ContourLayer** (p. 16), **HeightFieldLayer** (p. 22), and **ProxyLayer** (p. 39).

### 4.7.3 Member Data Documentation

#### 4.7.3.1 osg::Vec4 _defaultValue [protected]

#### 4.7.3.2 std::string _filename [protected]

#### 4.7.3.3 osg::ref_ptr<Locator> _locator [protected]

#### 4.7.3.4 osg::Texture::FilterMode _magFilter [protected]

#### 4.7.3.5 unsigned int _maxLevel [protected]

#### 4.7.3.6 osg::Texture::FilterMode _minFilter [protected]

#### 4.7.3.7 unsigned int _minLevel [protected]

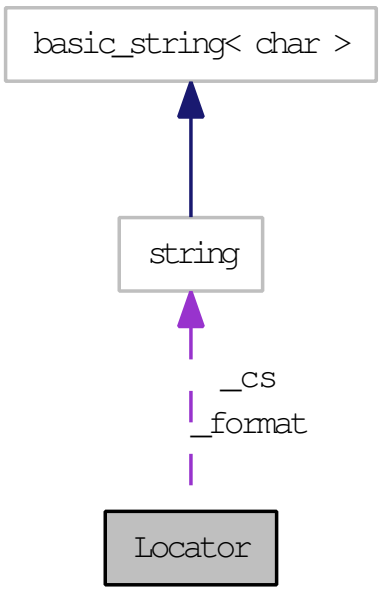
#### 4.7.3.8 osg::ref_ptr<ValidDataOperator> _validDataOperator [protected]

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

- **Layer**
- **Layer.cpp**

## 4.8 Locator Class Reference

Collaboration diagram for Locator:




### Public Types

- enum **CoordinateSystemType** { **GEOCENTRIC**, **GEOGRAPHIC**, **PROJECTED** }
  *CoordinateSystemType provides the classification of the type coordinate system represented.*

### Public Member Functions

- **Locator** (const **Locator** &, const osg::CopyOp &copyop=osg::CopyOp::SHALLOW_COPY)
  *Copy constructor using CopyOp to manage deep vs shallow copy.*
- **Locator** ()
- bool **computeLocalBounds** (**Locator** &source, osg::Vec3d &bottomLeft, osg::Vec3d &topRight) const
- virtual bool **convertLocalToModel** (const osg::Vec3d &local, osg::Vec3d &world) const
- virtual bool **convertModelToLocal** (const osg::Vec3d &world, osg::Vec3d &local) const
- const std::string & **getCoordinateSystem** () const
  *Get the CoordinateSystem reference string.*
- **CoordinateSystemType getCoordinateSystemType** () const
  *Get the CoordinatesSyetemType.*
- bool **getDefinedInFile** () const
- const osg::EllipsoidModel * **getEllipsoidModel** () const
  *Get the const EllipsoidModel.*
- osg::EllipsoidModel * **getEllipsoidModel** ()
  *Get the EllipsoidModel.*
- const std::string & **getFormat** () const
  *Get the coordinate system format string.*
- const osg::Matrixd & **getTransform** () const
  *Set the transformation from local coordinates to model coordinates.*

- bool **getTransformScaledByResolution** () const
- **META_Object** (osgTerrain, **Locator**)
- virtual bool **orientationOpenGL** () const
- void **setCoordinateSystem** (const std::string &cs)

  *Set the CoordinateSystem reference string, should be stored in a form consistent with the Format.*

- void **setCoordinateSystemType** (**CoordinateSystemType** type)

  *Set the CoordinatesSyetemType.*

- void **setDefinedInFile** (bool flag)
- void **setEllipsoidModel** (osg::EllipsoidModel ∗ellipsode)

  *Set EllipsoidModel to describe the model used to map lat, long and height into geocentric XYZ and back.*

- void **setFormat** (const std::string &format)

  *Set the coordinate system format string.*

- void **setTransform** (const osg::Matrixd &transform)

  *Set the transformation from local coordinates to model coordinates.*

- void **setTransformAsExtents** (double minX, double minY, double maxX, double maxY)

  *Set the extents of the local coords.*

- void **setTransformScaledByResolution** (bool scaledByResolution)

### Static Public Member Functions

- static bool **convertLocalCoordBetween** (const **Locator** &source, const osg::Vec3d &sourceNDC, const **Locator** &destination, osg::Vec3d &destinationNDC)

### Protected Member Functions

- virtual ∼**Locator** ()

### Protected Attributes

- **CoordinateSystemType _coordinateSystemType**
- std::string **_cs**
- bool **_definedInFile**
- osg::ref_ptr< osg::EllipsoidModel > **_ellipsoidModel**
- std::string **_format**
- osg::Matrixd **_inverse**
- osg::Matrixd **_transform**
- bool **_transformScaledByResolution**

## 4.8.1 Member Enumeration Documentation

### 4.8.1.1 enum CoordinateSystemType

CoordinateSystemType provides the classification of the type coordinate system represented.

**Enumerator:**

***GEOCENTRIC*** GEOCENTRIC coordinate systems are ones mapped to the around the ellipsoid, i.e. whole earth.

***GEOGRAPHIC*** GEOGRAPHIC coordinate systems are ones mapped to latitude and longitude.

***PROJECTED*** PROJECTED coordinate systems are ones projected to a local projected coordinate system i.e. UTMs.

### 4.8.2 Constructor & Destructor Documentation

#### 4.8.2.1 Locator ()

#### 4.8.2.2 Locator (const Locator & *locator*, const osg::CopyOp & *copyop* = `osg::CopyOp::SHALLOW_COPY`)

Copy constructor using CopyOp to manage deep vs shallow copy.

#### 4.8.2.3 ∼Locator () `[protected, virtual]`

### 4.8.3 Member Function Documentation

#### 4.8.3.1 bool computeLocalBounds (Locator & *source*, osg::Vec3d & *bottomLeft*, osg::Vec3d & *topRight*) const

#### 4.8.3.2 static bool convertLocalCoordBetween (const Locator & *source*, const osg::Vec3d & *sourceNDC*, const Locator & *destination*, osg::Vec3d & *destinationNDC*) `[inline, static]`

#### 4.8.3.3 bool convertLocalToModel (const osg::Vec3d & *local*, osg::Vec3d & *world*) const `[virtual]`

#### 4.8.3.4 bool convertModelToLocal (const osg::Vec3d & *world*, osg::Vec3d & *local*) const `[virtual]`

#### 4.8.3.5 const std::string& getCoordinateSystem () const `[inline]`

Get the CoordinateSystem reference string.

#### 4.8.3.6 CoordinateSystemType getCoordinateSystemType () const `[inline]`

Get the CoordinatesSyetemType.

#### 4.8.3.7 bool getDefinedInFile () const `[inline]`

#### 4.8.3.8 const osg::EllipsoidModel∗ getEllipsoidModel () const `[inline]`

Get the const EllipsoidModel.

#### 4.8.3.9 osg::EllipsoidModel∗ getEllipsoidModel () `[inline]`

Get the EllipsoidModel.

#### 4.8.3.10 const std::string& getFormat () const `[inline]`

Get the coordinate system format string.

#### 4.8.3.11 const osg::Matrixd& getTransform () const `[inline]`

Set the transformation from local coordinates to model coordinates.

#### 4.8.3.12 bool getTransformScaledByResolution () const `[inline]`

#### 4.8.3.13 META_Object (osgTerrain, Locator)

#### 4.8.3.14 bool orientationOpenGL () const `[virtual]`

#### 4.8.3.15 void setCoordinateSystem (const std::string & *cs*) `[inline]`

Set the CoordinateSystem reference string, should be stored in a form consistent with the Format.

#### 4.8.3.16 void setCoordinateSystemType (CoordinateSystemType *type*) `[inline]`

Set the CoordinatesSyetemType. Note, the user must keep the CoordinateSystemString consistent with the type of the CoordinateSystem.

#### 4.8.3.17 void setDefinedInFile (bool *flag*) `[inline]`

#### 4.8.3.18 void setEllipsoidModel (osg::EllipsoidModel ∗ *ellipsode*) `[inline]`

Set EllipsoidModel to describe the model used to map lat, long and height into geocentric XYZ and back.

#### 4.8.3.19 void setFormat (const std::string & *format*) `[inline]`

Set the coordinate system format string. Typical values would be WKT, PROJ4, USGS etc.

#### 4.8.3.20 void setTransform (const osg::Matrixd & *transform*) `[inline]`

Set the transformation from local coordinates to model coordinates.

#### 4.8.3.21 void setTransformAsExtents (double *minX*, double *minY*, double *maxX*, double *maxY*)

Set the extents of the local coords.

#### 4.8.3.22 void setTransformScaledByResolution (bool *scaledByResolution*) [inline]

### 4.8.4 Member Data Documentation

#### 4.8.4.1 CoordinateSystemType _coordinateSystemType [protected]

#### 4.8.4.2 std::string _cs [protected]

#### 4.8.4.3 bool _definedInFile [protected]

#### 4.8.4.4 osg::ref_ptr<osg::EllipsoidModel> _ellipsoidModel [protected]

#### 4.8.4.5 std::string _format [protected]

#### 4.8.4.6 osg::Matrixd _inverse [protected]

#### 4.8.4.7 osg::Matrixd _transform [protected]

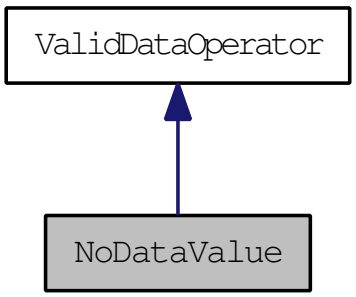
#### 4.8.4.8 bool _transformScaledByResolution [protected]

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

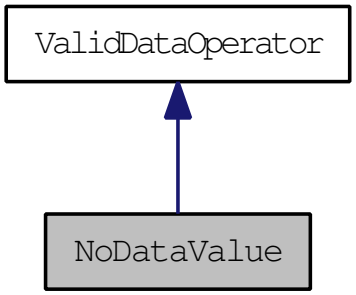
- **Locator**
- **Locator.cpp**

## 4.9 NoDataValue Struct Reference

Inheritance diagram for NoDataValue:




Collaboration diagram for NoDataValue:




### Public Member Functions

- **NoDataValue** (float value)
- float **getValue** () const
- virtual bool **operator()** (float value) const
- void **setNoDataValue** (float value)

### Public Attributes

- float **_value**

### 4.9.1 Constructor & Destructor Documentation

#### 4.9.1.1 NoDataValue (float *value*) [inline]

### 4.9.2 Member Function Documentation

#### 4.9.2.1 float getValue () const [inline]

#### 4.9.2.2 virtual bool operator() (float *value*) const [inline, virtual]

Reimplemented from **ValidDataOperator** (p. 57).

#### 4.9.2.3 void setNoDataValue (float *value*) [inline]

### 4.9.3 Member Data Documentation

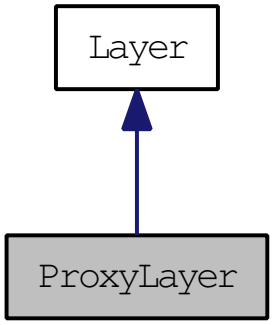
#### 4.9.3.1 float _value

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

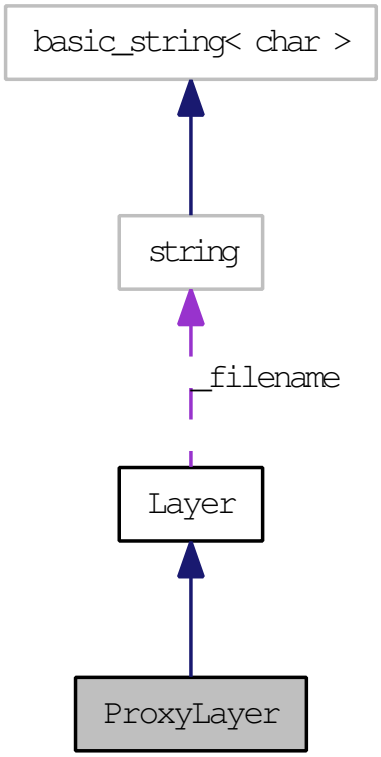
- **ValidDataOperator**

## 4.10 ProxyLayer Class Reference

Inheritance diagram for ProxyLayer:




Collaboration diagram for ProxyLayer:




### Public Member Functions

- **ProxyLayer** (const **ProxyLayer** &proxyLayer, const osg::CopyOp &copyop=osg::CopyOp::SHALLOW_-COPY)

  *Copy constructor using CopyOp to manage deep vs shallow copy.*

- **ProxyLayer** ()
- virtual osg::BoundingSphere **computeBound** (bool treatAsElevationLayer) const
- virtual void **dirty** ()

  *increment the modified count.*

- virtual const std::string & **getFileName** () const

  *Get the file name of the layer.*

- virtual const osg::Image * **getImage** () const
  *Return const image associated with layer if supported.*
- virtual osg::Image * **getImage** ()
  *Return image associated with layer if supported.*
- const **Layer** * **getImplementation** () const
  *Get the const implementation layer that does the actual work.*
- **Layer** * **getImplementation** ()
  *Get the implementation layer that does the actual work.*
- virtual unsigned int **getModifiedCount** () const
  *Get modified count value.*
- virtual unsigned int **getNumColumns** () const
- virtual unsigned int **getNumRows** () const
- virtual bool **getValue** (unsigned int i, unsigned int j, osg::Vec4 &value) const
- virtual bool **getValue** (unsigned int i, unsigned int j, osg::Vec3 &value) const
- virtual bool **getValue** (unsigned int i, unsigned int j, osg::Vec2 &value) const
- virtual bool **getValue** (unsigned int i, unsigned int j, float &value) const
- **META_Object** (osgTerrain, **ProxyLayer**)
- virtual void **setFileName** (const std::string &filename)
  *Set the file name of the data associated with this layer.*
- void **setImplementation** (**Layer** *layer)
  *Set the implementation layer that does the actual work.*
- virtual void **setModifiedCount** (unsigned int value)
  *Set the modified count value.*
- virtual bool **transform** (float offset, float scale)

## Protected Member Functions

- virtual ~**ProxyLayer** ()

## Protected Attributes

- osg::ref_ptr< **Layer** > **_implementation**

### 4.10.1 Constructor & Destructor Documentation

#### 4.10.1.1 ProxyLayer ()

#### 4.10.1.2 ProxyLayer (const ProxyLayer & *proxyLayer*, const osg::CopyOp & *copyop* = `osg::CopyOp::SHALLOW_COPY`)

Copy constructor using CopyOp to manage deep vs shallow copy.

#### 4.10.1.3 ~ProxyLayer () `[protected, virtual]`

### 4.10.2 Member Function Documentation

#### 4.10.2.1 osg::BoundingSphere computeBound (bool *treatAsElevationLayer*) const `[virtual]`

Reimplemented from **Layer** (p. 28).

#### 4.10.2.2 void dirty () [virtual]

increment the modified count. "

Reimplemented from **Layer** (p. 28).

#### 4.10.2.3 virtual const std::string& getFileName () const [inline, virtual]

Get the file name of the layer.

Reimplemented from **Layer** (p. 28).

#### 4.10.2.4 virtual const osg::Image* getImage () const [inline, virtual]

Return const image associated with layer if supported.

Reimplemented from **Layer** (p. 28).

#### 4.10.2.5 virtual osg::Image* getImage () [inline, virtual]

Return image associated with layer if supported.

Reimplemented from **Layer** (p. 28).

#### 4.10.2.6 const Layer* getImplementation () const [inline]

Get the const implementation layer that does the actual work.

#### 4.10.2.7 Layer* getImplementation () [inline]

Get the implementation layer that does the actual work.

#### 4.10.2.8 unsigned int getModifiedCount () const [virtual]

Get modified count value.

Reimplemented from **Layer** (p. 29).

#### 4.10.2.9 unsigned int getNumColumns () const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.10.2.10 unsigned int getNumRows () const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.10.2.11 bool getValue (unsigned int *i*, unsigned int *j*, osg::Vec4 & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.10.2.12 bool getValue (unsigned int *i*, unsigned int *j*, osg::Vec3 & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.10.2.13 bool getValue (unsigned int *i*, unsigned int *j*, osg::Vec2 & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.10.2.14 bool getValue (unsigned int *i*, unsigned int *j*, float & *value*) const [virtual]

Reimplemented from **Layer** (p. 29).

#### 4.10.2.15 META_Object (osgTerrain, ProxyLayer)

#### 4.10.2.16 void setFileName (const std::string & *filename*) [virtual]

Set the file name of the data associated with this layer.

Reimplemented from **Layer** (p. 29).

#### 4.10.2.17 void setImplementation (Layer * *layer*) [inline]

Set the implementation layer that does the actual work.

#### 4.10.2.18 void setModifiedCount (unsigned int) [virtual]

Set the modified count value.

Reimplemented from **Layer** (p. 30).

#### 4.10.2.19 bool transform (float *offset*, float *scale*) [virtual]

Reimplemented from **Layer** (p. 30).

### 4.10.3 Member Data Documentation

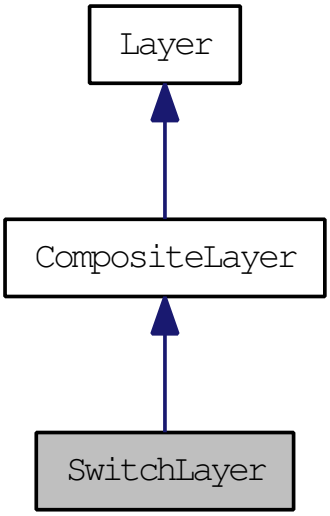
#### 4.10.3.1 osg::ref_ptr<Layer> _implementation [protected]

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

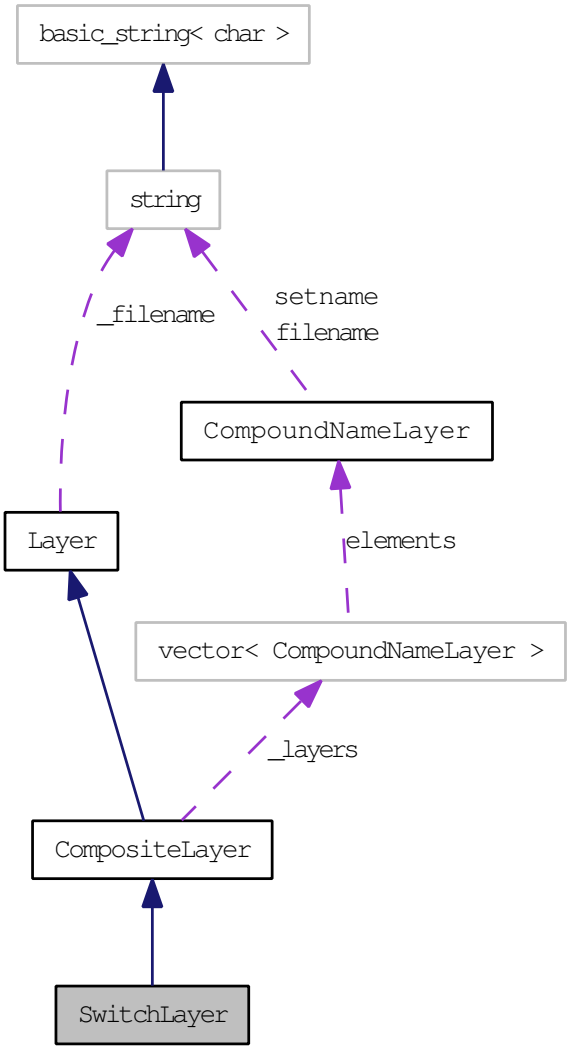
- **Layer**
- **Layer.cpp**

## 4.11 SwitchLayer Class Reference

Inheritance diagram for SwitchLayer:




Collaboration diagram for SwitchLayer:



### Public Member Functions

- **SwitchLayer** (const **SwitchLayer** &switchLayer, const osg::CopyOp &copyop=osg::CopyOp::SHALLOW_-COPY)

  *Copy constructor using CopyOp to manage deep vs shallow copy.*

- **SwitchLayer** ()
- int **getActiveLayer** () const
- virtual const osg::Image ∗ **getImage** () const

  *Return const image associated with layer if supported.*

- virtual osg::Image ∗ **getImage** ()

  *Return image associated with layer if supported.*

- **META_Object** (osgTerrain, **SwitchLayer**)
- void **setActiveLayer** (int i)

### Protected Member Functions

- virtual ∼**SwitchLayer** ()

### Protected Attributes

- int **_activeLayer**

## 4.11.1 Constructor & Destructor Documentation

#### 4.11.1.1 SwitchLayer ()

#### 4.11.1.2 SwitchLayer (const SwitchLayer & *switchLayer*, const osg::CopyOp & *copyop* = `osg::CopyOp::SHALLOW_COPY`)

Copy constructor using CopyOp to manage deep vs shallow copy.

#### 4.11.1.3 virtual ∼SwitchLayer () `[inline, protected, virtual]`

## 4.11.2 Member Function Documentation

#### 4.11.2.1 int getActiveLayer () const `[inline]`

#### 4.11.2.2 virtual const osg::Image∗ getImage () const `[inline, virtual]`

Return const image associated with layer if supported.

Reimplemented from **Layer** (p. 28).

#### 4.11.2.3 virtual osg::Image∗ getImage () `[inline, virtual]`

Return image associated with layer if supported.

Reimplemented from **Layer** (p. 28).

#### 4.11.2.4 META_Object (osgTerrain, SwitchLayer)

#### 4.11.2.5 void setActiveLayer (int *i*) `[inline]`

## 4.11.3 Member Data Documentation

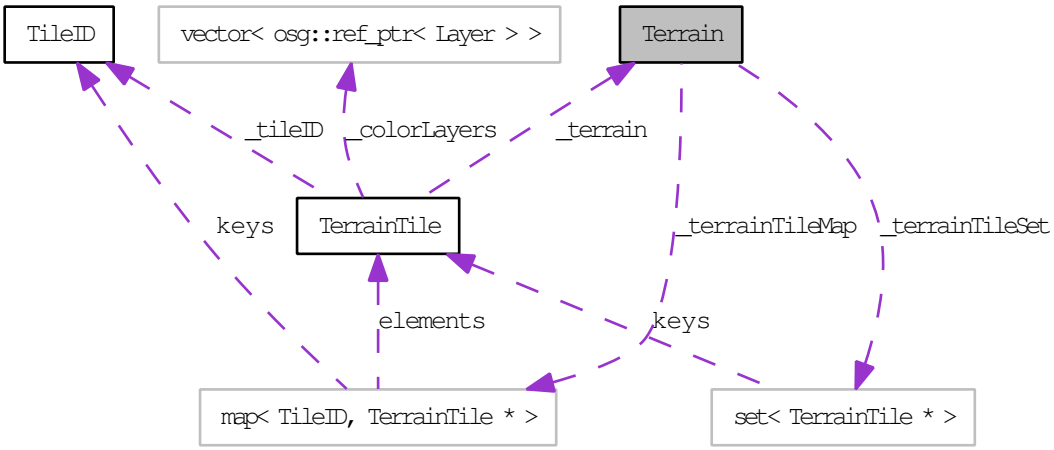
#### 4.11.3.1 int _activeLayer `[protected]`

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

- **Layer**
- **Layer.cpp**

## 4.12 Terrain Class Reference

**Terrain** (p. 42) provides a framework for loosely coupling height field data with height rendering algorithms. Collaboration diagram for Terrain:




### Public Member Functions

- **Terrain** (const **Terrain** &, const osg::CopyOp &copyop=osg::CopyOp::SHALLOW_COPY)
  *Copy constructor using CopyOp to manage deep vs shallow copy.*
- **Terrain** ()
- float **getSampleRatio** () const
  *Get the sample ratio hint.*
- const **TerrainTechnique** * **getTerrainTechniquePrototype** () const
  *Get the const **TerrainTechnique** (p. 45) protype.*
- **TerrainTechnique** * **getTerrainTechniquePrototype** ()
  *Get the **TerrainTechnique** (p. 45) prototype.*
- const **TerrainTile** * **getTile** (const **TileID** &tileID) const
  *Get the const **TerrainTile** (p. 48) for a given **TileID** (p. 54).*
- **TerrainTile** * **getTile** (const **TileID** &tileID)
  *Get the **TerrainTile** (p. 48) for a given **TileID** (p. 54).*
- float **getVerticalScale** () const
  *Get the vertical scale hint.*
- **META_Node** (osgTerrain, **Terrain**)
- void **setSampleRatio** (float ratio)
  *Set the sample ratio hint that **TerrainTile** (p. 48) should use when building geometry.*
- void **setTerrainTechniquePrototype** (**TerrainTechnique** *technique)
  *Set the **TerrainTechnique** (p. 45) prototype from which TerrainTiles can clone the techniques from.*
- void **setVerticalScale** (float scale)
  *Set the vertical scale hint.*
- virtual void **traverse** (osg::NodeVisitor &nv)

### Protected Types

- typedef std::map< **TileID**, **TerrainTile** * > **TerrainTileMap**
- typedef std::set< **TerrainTile** * > **TerrainTileSet**

### Protected Member Functions

- virtual ~**Terrain** ()
- void **dirtyRegisteredTiles** ()
- void **registerTerrainTile** (**TerrainTile** *tile)
- void **unregisterTerrainTile** (**TerrainTile** *tile)

### Protected Attributes

- OpenThreads::Mutex **_mutex**
- float **_sampleRatio**
- osg::ref_ptr< **TerrainTechnique** > **_terrainTechnique**
- **TerrainTileMap _terrainTileMap**
- **TerrainTileSet _terrainTileSet**
- float **_verticalScale**

### Friends

- class **TerrainTile**

## 4.12.1 Detailed Description

**Terrain** (p. 42) provides a framework for loosely coupling height field data with height rendering algorithms. This allows TerrainTechniques to be plugged in at runtime.

## 4.12.2 Member Typedef Documentation

#### 4.12.2.1 typedef std::map< TileID, TerrainTile* > TerrainTileMap [protected]

#### 4.12.2.2 typedef std::set< TerrainTile* > TerrainTileSet [protected]

## 4.12.3 Constructor & Destructor Documentation

#### 4.12.3.1 Terrain ()

#### 4.12.3.2 Terrain (const Terrain & *ts*, const osg::CopyOp & *copyop* = `osg::CopyOp::SHALLOW_COPY`)

Copy constructor using CopyOp to manage deep vs shallow copy.

#### 4.12.3.3 ~Terrain () [protected, virtual]

## 4.12.4 Member Function Documentation

#### 4.12.4.1 void dirtyRegisteredTiles () [protected]

#### 4.12.4.2 float getSampleRatio () const [inline]

Get the sample ratio hint.

#### 4.12.4.3 const TerrainTechnique* getTerrainTechniquePrototype () const [inline]

Get the const **TerrainTechnique** (p. 45) protype.

#### 4.12.4.4 TerrainTechnique* getTerrainTechniquePrototype () [inline]

Get the **TerrainTechnique** (p. 45) prototype.

#### 4.12.4.5 const TerrainTile * getTile (const TileID & *tileID*) const

Get the const **TerrainTile** (p. 48) for a given **TileID** (p. 54).

#### 4.12.4.6 TerrainTile * getTile (const TileID & *tileID*)

Get the **TerrainTile** (p. 48) for a given **TileID** (p. 54).

#### 4.12.4.7 float getVerticalScale () const [inline]

Get the vertical scale hint.

#### 4.12.4.8 META_Node (osgTerrain, Terrain)

#### 4.12.4.9 void registerTerrainTile (TerrainTile * *tile*) [protected]

#### 4.12.4.10 void setSampleRatio (float *ratio*) [inline]

Set the sample ratio hint that **TerrainTile** (p. 48) should use when building geometry. Defaults to 1.0, which means use all original sample points.

#### 4.12.4.11 void setTerrainTechniquePrototype (TerrainTechnique * *technique*) [inline]

Set the **TerrainTechnique** (p. 45) prototype from which TerrainTiles can clone the techniques from.

#### 4.12.4.12 void setVerticalScale (float *scale*) [inline]

Set the vertical scale hint.

#### 4.12.4.13 void traverse (osg::NodeVisitor & *nv*) [virtual]

#### 4.12.4.14 void unregisterTerrainTile (TerrainTile * *tile*) [protected]

### 4.12.5 Friends And Related Function Documentation

#### 4.12.5.1 friend class TerrainTile [friend]

### 4.12.6 Member Data Documentation

#### 4.12.6.1 OpenThreads::Mutex _mutex [mutable, protected]

#### 4.12.6.2 float _sampleRatio [protected]

#### 4.12.6.3 osg::ref_ptr<TerrainTechnique> _terrainTechnique [protected]

#### 4.12.6.4 TerrainTileMap _terrainTileMap [protected]

#### 4.12.6.5 TerrainTileSet _terrainTileSet [protected]

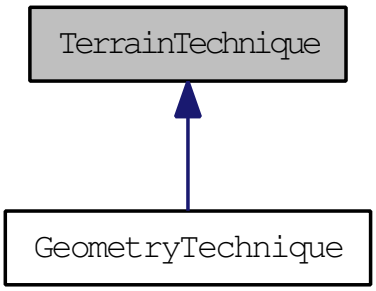
#### 4.12.6.6 float _verticalScale [protected]

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

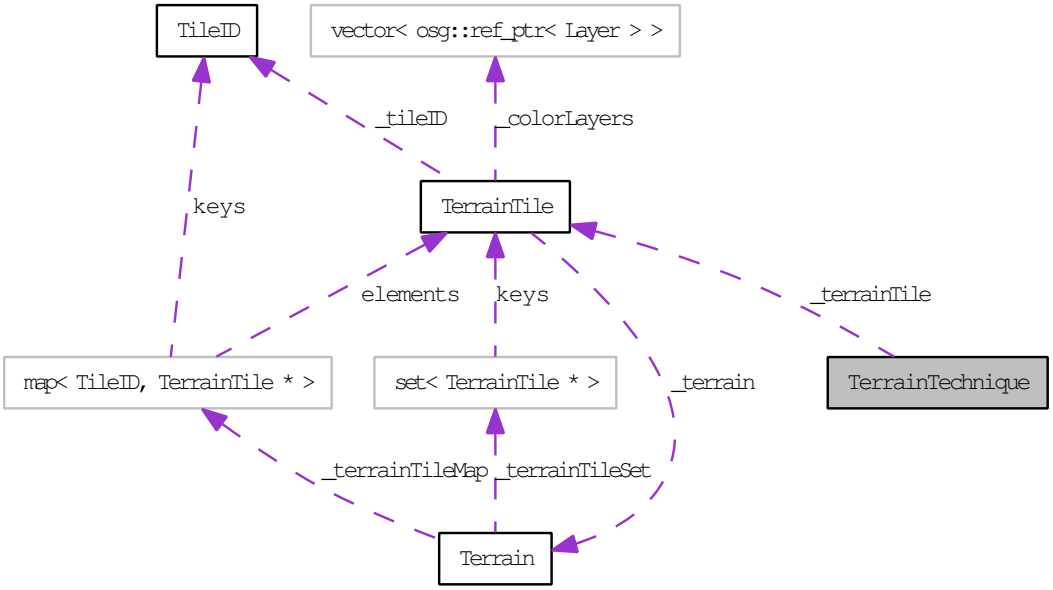
- **Terrain**
- **Terrain.cpp**

## 4.13 TerrainTechnique Class Reference

Inheritance diagram for TerrainTechnique:




Collaboration diagram for TerrainTechnique:




### Public Member Functions

- **TerrainTechnique** (const **TerrainTechnique** &, const osg::CopyOp &copyop=osg::CopyOp::SHALLOW_COPY)

  *Copy constructor using CopyOp to manage deep vs shallow copy.*

- **TerrainTechnique** ()
- virtual void **cleanSceneGraph** ()

  *Clean scene graph from any terrain technique specific nodes.*

- virtual void **cull** (osgUtil::CullVisitor ∗nv)
- const **TerrainTile** ∗ **getTerrainTile** () const
- **TerrainTile** ∗ **getTerrainTile** ()
- virtual void **init** ()
- **META_Object** (osgTerrain, **TerrainTechnique**)
- virtual void **releaseGLObjects** (osg::State ∗=0) const

  *If State is non-zero, this function releases any associated OpenGL objects for the specified graphics context.*

- virtual void **traverse** (osg::NodeVisitor &nv)

*Traverse the terrain subgraph.*

- virtual void **update** (osgUtil::UpdateVisitor ∗nv)

### Protected Member Functions

- virtual ∼**TerrainTechnique** ()
- void **setDirty** (bool dirty)

### Protected Attributes

- **TerrainTile** ∗ **_terrainTile**

### Friends

- class **osgTerrain::TerrainTile**

## 4.13.1 Constructor & Destructor Documentation

#### 4.13.1.1 TerrainTechnique ()

#### 4.13.1.2 TerrainTechnique (const TerrainTechnique & *TerrainTechnique*, const osg::CopyOp & *copyop* = `osg::CopyOp::SHALLOW_COPY`)

Copy constructor using CopyOp to manage deep vs shallow copy.

#### 4.13.1.3 ∼TerrainTechnique () `[protected, virtual]`

## 4.13.2 Member Function Documentation

#### 4.13.2.1 void cleanSceneGraph () `[virtual]`

Clean scene graph from any terrain technique specific nodes.

Reimplemented in **GeometryTechnique** (p. 18).

#### 4.13.2.2 void cull (osgUtil::CullVisitor ∗ *nv*) `[virtual]`

Reimplemented in **GeometryTechnique** (p. 18).

#### 4.13.2.3 const TerrainTile∗ getTerrainTile () const `[inline]`

#### 4.13.2.4 TerrainTile∗ getTerrainTile () `[inline]`

#### 4.13.2.5 void init () `[virtual]`

Reimplemented in **GeometryTechnique** (p. 19).

#### 4.13.2.6 META_Object (osgTerrain, TerrainTechnique)

#### 4.13.2.7 virtual void releaseGLObjects (osg::State ∗ = `0`) const `[inline, virtual]`

If State is non-zero, this function releases any associated OpenGL objects for the specified graphics context. Otherwise, releases OpenGL objects for all graphics contexts.

Reimplemented in **GeometryTechnique** (p. 19).

#### 4.13.2.8 void setDirty (bool *dirty*) `[protected]`

#### 4.13.2.9 void traverse (osg::NodeVisitor & *nv*) `[virtual]`

Traverse the terrain subgraph.

Reimplemented in **GeometryTechnique** (p. 19).

#### 4.13.2.10 void update (osgUtil::UpdateVisitor ∗ *nv*) `[virtual]`

Reimplemented in **GeometryTechnique** (p. 19).

### 4.13.3 Friends And Related Function Documentation

#### 4.13.3.1 friend class osgTerrain::TerrainTile [friend]

### 4.13.4 Member Data Documentation

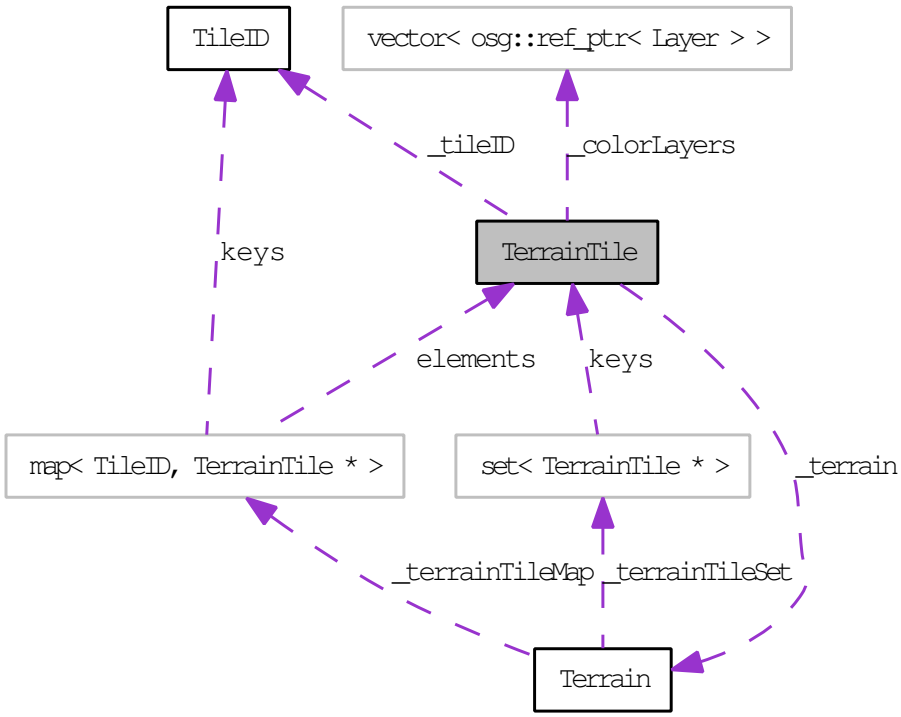
#### 4.13.4.1 TerrainTile* _terrainTile [protected]

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

- **TerrainTechnique**
- **TerrainTechnique.cpp**

## 4.14 TerrainTile Class Reference

**Terrain** (p. 42) provides a framework for loosely coupling height field data with height rendering algorithms. Collaboration diagram for TerrainTile:




### Classes

- struct **TileLoadedCallback**

  *Callback for post processing loaded **TerrainTile** (p. 48), and for filling in missing elements such as external external imagery.*

### Public Member Functions

- **TerrainTile** (const **TerrainTile** &, const osg::CopyOp &copyop=osg::CopyOp::SHALLOW_COPY)

  *Copy constructor using CopyOp to manage deep vs shallow copy.*

- **TerrainTile** ()
- virtual osg::BoundingSphere **computeBound** () const

  *Compute the bounding volume of the terrain by computing the union of the bounding volumes of all layers.*

- const **Layer** * **getColorLayer** (unsigned int i) const

  *Set const color layer with specified layer number.*

- **Layer** * **getColorLayer** (unsigned int i)

  *Get color layer with specified layer number.*

- bool **getDirty** () const

  *return true if the tile is dirty and needs to be updated,*

- const **Layer** * **getElevationLayer** () const

  *Get the const layer to use to define the elevations of the terrain.*

- **Layer** * **getElevationLayer** ()

*Get the layer to use to define the elevations of the terrain.*

- const **Locator** * **getLocator** () const
  *Get the const coordinate frame locator of the terrain node.*
- **Locator** * **getLocator** ()
  *Get the coordinate frame locator of the terrain node.*
- unsigned int **getNumColorLayers** () const
  *Get the number of colour layers.*
- bool **getRequiresNormals** () const
  *Get whether the **TerrainTechnique** (p. 45) should create per vertex normals for lighting purposes.*
- const **Terrain** * **getTerrain** () const
  *Get the const **Terrain** (p. 42) that this **Terrain** (p. 42) tile is a member of.*
- **Terrain** * **getTerrain** ()
  *Get the **Terrain** (p. 42) that this **Terrain** (p. 42) tile is a member of.*
- const **TerrainTechnique** * **getTerrainTechnique** () const
  *Get the const **TerrainTechnique** (p. 45).*
- **TerrainTechnique** * **getTerrainTechnique** ()
  *Get the **TerrainTechnique** (p. 45).*
- const **TileID** & **getTileID** () const
  *Get the **TileID** (p. 54) (layer, x,y) of the **TerrainTile** (p. 48).*
- bool **getTreatBoundariesToValidDataAsDefaultValue** () const
  *Get whether the TeatBoundariesToValidDataAsDefaultValue hint.*
- void **init** ()
  *Call init on any attached **TerrainTechnique** (p. 45).*
- **META_Node** (osgTerrain, **TerrainTile**)
- virtual void **releaseGLObjects** (osg::State *=0) const
  *If State is non-zero, this function releases any associated OpenGL objects for the specified graphics context.*
- void **setColorLayer** (unsigned int i, **Layer** *layer)
  *Set a color layer with specified layer number.*
- void **setDirty** (bool dirty)
  *Set the dirty flag on/off.*
- void **setElevationLayer** (**Layer** *layer)
  *Set the layer to use to define the elevations of the terrain.*
- void **setLocator** (**Locator** *locator)
  *Set the coordinate frame locator of the terrain node.*
- void **setRequiresNormals** (bool flag)
  *Set hint to whether the **TerrainTechnique** (p. 45) should create per vertex normals for lighting purposes.*
- void **setTerrain** (**Terrain** *ts)
  *Set the **Terrain** (p. 42) that this **Terrain** (p. 42) tile is a member of.*

- void **setTerrainTechnique** (**TerrainTechnique** ∗terrainTechnique)

  *Set the **TerrainTechnique** (p. 45).*

- void **setTileID** (const **TileID** &tileID)

  *Set the **TileID** (p. 54) (layer, x,y) of the **TerrainTile** (p. 48).*

- void **setTreatBoundariesToValidDataAsDefaultValue** (bool flag)

  *Set the hint to whether the **TerrainTechnique** (p. 45) should treat the invalid **Layer** (p. 26) entries that at are neighbours to valid entries with the default value.*

- virtual void **traverse** (osg::NodeVisitor &nv)

### Static Public Member Functions

- static osg::ref_ptr< **TileLoadedCallback** > & **getTileLoadedCallback** ()
- static void **setTileLoadedCallback** (**TileLoadedCallback** ∗lc)

### Protected Types

- typedef std::vector< osg::ref_ptr< **Layer** > > **Layers**

### Protected Member Functions

- virtual ∼**TerrainTile** ()

### Protected Attributes

- **Layers _colorLayers**
- bool **_dirty**
- osg::ref_ptr< **Layer** > **_elevationLayer**
- bool **_hasBeenTraversal**
- osg::ref_ptr< **Locator** > **_locator**
- bool **_requiresNormals**
- **Terrain** ∗ **_terrain**
- osg::ref_ptr< **TerrainTechnique** > **_terrainTechnique**
- **TileID _tileID**
- bool **_treatBoundariesToValidDataAsDefaultValue**

### Friends

- class **Terrain**

### 4.14.1 Detailed Description

**Terrain** (p. 42) provides a framework for loosely coupling height field data with height rendering algorithms. This allows TerrainTechnique's to be plugged in at runtime.

### 4.14.2 Member Typedef Documentation

#### 4.14.2.1 typedef std::vector< osg::ref_ptr<Layer> > Layers [protected]

### 4.14.3 Constructor & Destructor Documentation

#### 4.14.3.1 TerrainTile ()

#### 4.14.3.2 TerrainTile (const TerrainTile & *terrain*, const osg::CopyOp & *copyop* = osg::CopyOp::SHALLOW_COPY)

Copy constructor using CopyOp to manage deep vs shallow copy.

#### 4.14.3.3 ~TerrainTile () [protected, virtual]

### 4.14.4 Member Function Documentation

#### 4.14.4.1 osg::BoundingSphere computeBound () const [virtual]

Compute the bounding volume of the terrain by computing the union of the bounding volumes of all layers.

#### 4.14.4.2 const Layer* getColorLayer (unsigned int *i*) const [inline]

Set const color layer with specified layer number.

#### 4.14.4.3 Layer* getColorLayer (unsigned int *i*) [inline]

Get color layer with specified layer number.

#### 4.14.4.4 bool getDirty () const [inline]

return true if the tile is dirty and needs to be updated,

#### 4.14.4.5 const Layer* getElevationLayer () const [inline]

Get the const layer to use to define the elevations of the terrain.

#### 4.14.4.6 Layer* getElevationLayer () [inline]

Get the layer to use to define the elevations of the terrain.

#### 4.14.4.7 const Locator* getLocator () const [inline]

Get the const coordinate frame locator of the terrain node.

#### 4.14.4.8 Locator* getLocator () [inline]

Get the coordinate frame locator of the terrain node.

#### 4.14.4.9 unsigned int getNumColorLayers () const [inline]

Get the number of colour layers.

#### 4.14.4.10 bool getRequiresNormals () const [inline]

Get whether the **TerrainTechnique** (p. 45) should create per vertex normals for lighting purposes.

#### 4.14.4.11 const Terrain* getTerrain () const [inline]

Get the const **Terrain** (p. 42) that this **Terrain** (p. 42) tile is a member of.

#### 4.14.4.12 Terrain* getTerrain () [inline]

Get the **Terrain** (p. 42) that this **Terrain** (p. 42) tile is a member of.

#### 4.14.4.13 const TerrainTechnique* getTerrainTechnique () const [inline]

Get the const **TerrainTechnique** (p. 45).

#### 4.14.4.14 TerrainTechnique* getTerrainTechnique () [inline]

Get the **TerrainTechnique** (p. 45).

#### 4.14.4.15 const TileID& getTileID () const [inline]

Get the **TileID** (p. 54) (layer, x,y) of the **TerrainTile** (p. 48).

#### 4.14.4.16 osg::ref_ptr< TerrainTile::TileLoadedCallback > & getTileLoadedCallback () [static]

#### 4.14.4.17 bool getTreatBoundariesToValidDataAsDefaultValue () const [inline]

Get whether the TeatBoundariesToValidDataAsDefaultValue hint.

#### 4.14.4.18 void init ()

Call init on any attached **TerrainTechnique** (p. 45).

#### 4.14.4.19 META_Node (osgTerrain, TerrainTile)

#### 4.14.4.20 void releaseGLObjects (osg::State * *state* = 0) const [virtual]

If State is non-zero, this function releases any associated OpenGL objects for the specified graphics context. Otherwise, releases OpenGL objects for all graphics contexts.

#### 4.14.4.21 void setColorLayer (unsigned int *i*, Layer * *layer*)

Set a color layer with specified layer number.

#### 4.14.4.22 void setDirty (bool *dirty*)

Set the dirty flag on/off.

#### 4.14.4.23 void setElevationLayer (Layer * *layer*)

Set the layer to use to define the elevations of the terrain.

#### 4.14.4.24 void setLocator (Locator * *locator*) [inline]

Set the coordinate frame locator of the terrain node. The locator takes non-dimensional s,t coordinates into the X,Y,Z world coords and back.

#### 4.14.4.25 void setRequiresNormals (bool *flag*) [inline]

Set hint to whether the **TerrainTechnique** (p. 45) should create per vertex normals for lighting purposes.

#### 4.14.4.26 void setTerrain (Terrain * *ts*)

Set the **Terrain** (p. 42) that this **Terrain** (p. 42) tile is a member of.

#### 4.14.4.27 void setTerrainTechnique (TerrainTechnique * *terrainTechnique*)

Set the **TerrainTechnique** (p. 45).

#### 4.14.4.28 void setTileID (const TileID & *tileID*)

Set the **TileID** (p. 54) (layer, x,y) of the **TerrainTile** (p. 48). The **TileID** (p. 54) is used so it can be located by its neighbours via the enclosing **Terrain** (p. 42) node that manages a map of **TileID** (p. 54) to TerraiTiles.

#### 4.14.4.29 void setTileLoadedCallback (TerrainTile::TileLoadedCallback * *lc*) [static]

#### 4.14.4.30 void setTreatBoundariesToValidDataAsDefaultValue (bool *flag*) [inline]

Set the hint to whether the **TerrainTechnique** (p. 45) should treat the invalid **Layer** (p. 26) entries that at are neigbours to valid entries with the default value.

#### 4.14.4.31 void traverse (osg::NodeVisitor & *nv*) [virtual]

### 4.14.5 Friends And Related Function Documentation

#### 4.14.5.1 friend class Terrain [friend]

### 4.14.6 Member Data Documentation

#### 4.14.6.1 Layers _colorLayers [protected]

#### 4.14.6.2 bool _dirty [protected]

#### 4.14.6.3 osg::ref_ptr<Layer> _elevationLayer [protected]

#### 4.14.6.4 bool _hasBeenTraversal [protected]

#### 4.14.6.5 osg::ref_ptr<Locator> _locator [protected]

#### 4.14.6.6 bool _requiresNormals [protected]

#### 4.14.6.7 Terrain* _terrain [protected]

#### 4.14.6.8 osg::ref_ptr<TerrainTechnique> _terrainTechnique [protected]

#### 4.14.6.9 TileID _tileID [protected]

#### 4.14.6.10 bool _treatBoundariesToValidDataAsDefaultValue [protected]

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

- **TerrainTile**
- **TerrainTile.cpp**

## 4.15 TileID Class Reference

### Public Member Functions

- **TileID** (int in_level, int in_x, int in_y)
- **TileID** ()
- bool **operator!=** (const **TileID** &rhs) const
- bool **operator**< (const **TileID** &rhs) const
- bool **operator==** (const **TileID** &rhs) const
- bool **valid** () const

### Public Attributes

- int **level**
- int **x**
- int **y**

### 4.15.1 Constructor & Destructor Documentation

#### 4.15.1.1 TileID ()

#### 4.15.1.2 TileID (int *in_level*, int *in_x*, int *in_y*)

### 4.15.2 Member Function Documentation

#### 4.15.2.1 bool operator!= (const TileID & *rhs*) const [inline]

#### 4.15.2.2 bool operator< (const TileID & *rhs*) const [inline]

#### 4.15.2.3 bool operator== (const TileID & *rhs*) const [inline]

#### 4.15.2.4 bool valid () const [inline]

### 4.15.3 Member Data Documentation

#### 4.15.3.1 int level

#### 4.15.3.2 int x

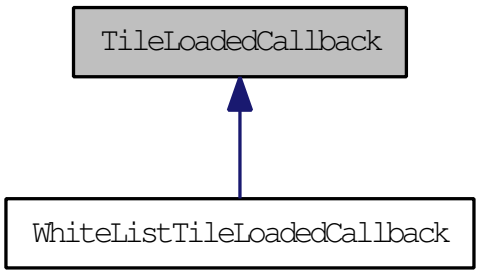
#### 4.15.3.3 int y

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

- **TerrainTile**
- **TerrainTile.cpp**

## 4.16 TileLoadedCallback Struct Reference

Callback for post processing loaded **TerrainTile** (p. 48), and for filling in missing elements such as external external imagery. Inheritance diagram for TileLoadedCallback:




### Public Member Functions

- virtual bool **deferExternalLayerLoading** () const =0
- virtual void **loaded** (**osgTerrain::TerrainTile** ∗tile, const osgDB::ReaderWriter::Options ∗options) const =0

### 4.16.1 Detailed Description

Callback for post processing loaded **TerrainTile** (p. 48), and for filling in missing elements such as external external imagery.

### 4.16.2 Member Function Documentation

#### 4.16.2.1 virtual bool deferExternalLayerLoading () const [pure virtual]

Implemented in **WhiteListTileLoadedCallback** (p. 61).

#### 4.16.2.2 virtual void loaded (osgTerrain::TerrainTile ∗ *tile*, const osgDB::ReaderWriter::Options ∗ *options*) const [pure virtual]

Implemented in **WhiteListTileLoadedCallback** (p. 61).

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

- **TerrainTile**

## 4.17 TransformOperator Struct Reference

### Public Member Functions

- **TransformOperator** (float offset, float scale)
- void **operator()** (float &v) const
- void **operator()** (int &v) const
- void **operator()** (short &v) const
- void **operator()** (char &v) const
- void **operator()** (unsigned int &v) const
- void **operator()** (unsigned short &v) const
- void **operator()** (unsigned char &v) const

### Public Attributes

- float **_offset**
- float **_scale**

### 4.17.1 Constructor & Destructor Documentation

#### 4.17.1.1 TransformOperator (float *offset*, float *scale*) [inline]

### 4.17.2 Member Function Documentation

#### 4.17.2.1 void operator() (float & *v*) const [inline]

#### 4.17.2.2 void operator() (int & *v*) const [inline]

#### 4.17.2.3 void operator() (short & *v*) const [inline]

#### 4.17.2.4 void operator() (char & *v*) const [inline]

#### 4.17.2.5 void operator() (unsigned int & *v*) const [inline]

#### 4.17.2.6 void operator() (unsigned short & *v*) const [inline]

#### 4.17.2.7 void operator() (unsigned char & *v*) const [inline]

### 4.17.3 Member Data Documentation

#### 4.17.3.1 float _offset

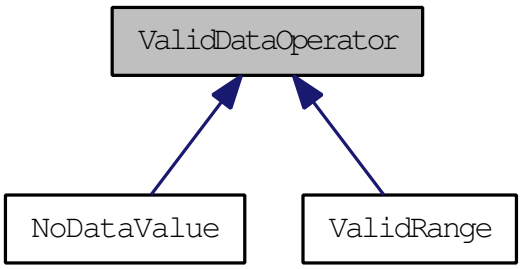
#### 4.17.3.2 float _scale

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

- **Layer.cpp**

## 4.18 ValidDataOperator Struct Reference

Inheritance diagram for ValidDataOperator:




### Public Member Functions

- virtual bool **operator()** (const osg::Vec4 &value) const
- virtual bool **operator()** (const osg::Vec3 &value) const
- virtual bool **operator()** (const osg::Vec2 &value) const
- virtual bool **operator()** (float) const

### 4.18.1 Member Function Documentation

#### 4.18.1.1 virtual bool operator() (const osg::Vec4 & *value*) const [inline, virtual]

#### 4.18.1.2 virtual bool operator() (const osg::Vec3 & *value*) const [inline, virtual]

#### 4.18.1.3 virtual bool operator() (const osg::Vec2 & *value*) const [inline, virtual]

#### 4.18.1.4 virtual bool operator() (float) const [inline, virtual]

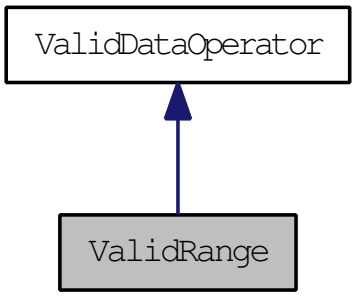
Reimplemented in **ValidRange** (p. 58), and **NoDataValue** (p. 35).

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

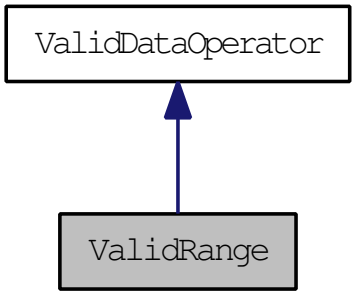
- **ValidDataOperator**

## 4.19 ValidRange Struct Reference

Inheritance diagram for ValidRange:




Collaboration diagram for ValidRange:




### Public Member Functions

- **ValidRange** (float minValue, float maxValue)
- float **getMaxValue** () const
- float **getMinValue** () const
- virtual bool **operator()** (float value) const
- void **setMaxValue** (float maxValue)
- void **setMinValue** (float minValue)
- void **setRange** (float minValue, float maxValue)

### Public Attributes

- float **_maxValue**
- float **_minValue**

### 4.19.1 Constructor & Destructor Documentation

#### 4.19.1.1 ValidRange (float *minValue*, float *maxValue*) [inline]

### 4.19.2 Member Function Documentation

#### 4.19.2.1 float getMaxValue () const [inline]

#### 4.19.2.2 float getMinValue () const [inline]

#### 4.19.2.3 virtual bool operator() (float *value*) const [inline, virtual]

Reimplemented from **ValidDataOperator** (p. 57).

#### 4.19.2.4 void setMaxValue (float *maxValue*) [inline]

#### 4.19.2.5 void setMinValue (float *minValue*) [inline]

#### 4.19.2.6 void setRange (float *minValue*, float *maxValue*) [inline]

### 4.19.3 Member Data Documentation

#### 4.19.3.1 float _maxValue

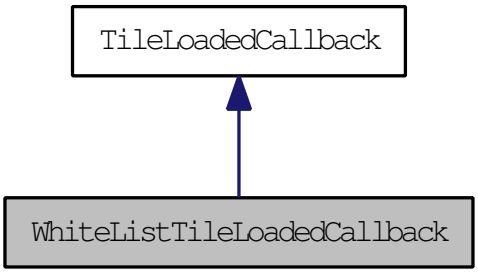
#### 4.19.3.2 float _minValue

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

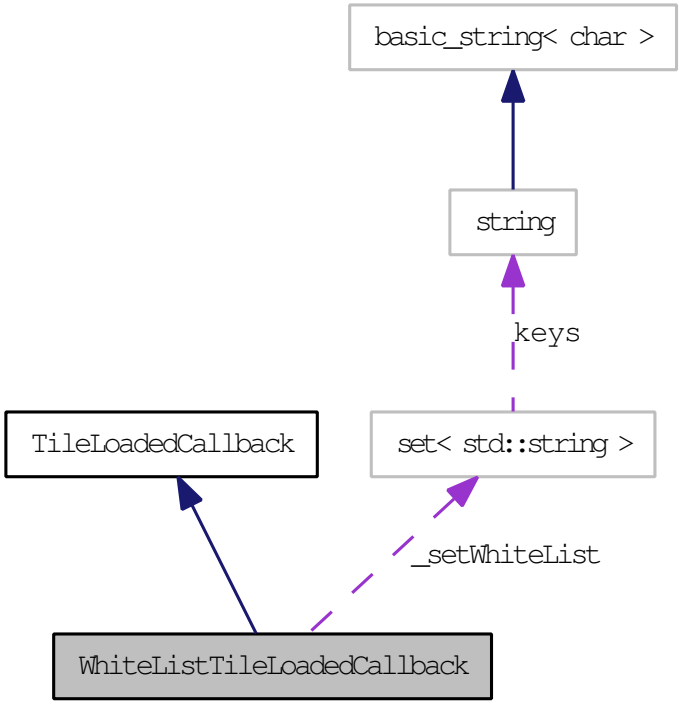
- **ValidDataOperator**

## 4.20 WhiteListTileLoadedCallback Class Reference

Helper callback for managing optional sets of layers, that loading of is deffered to this callback, with this callback working out which layers to load, and how to create fallback versions of the layers. Inheritance diagram for WhiteListTileLoadedCallback:




Collaboration diagram for WhiteListTileLoadedCallback:




### Public Member Functions

- **WhiteListTileLoadedCallback** ()
- void **allow** (const std::string &setname)
- virtual bool **deferExternalLayerLoading** () const
- bool **getAllowAll** () const
- unsigned int **getMinimumNumOfLayers** () const
- bool **getReplaceSwitchLayer** () const
- bool **layerAcceptable** (const std::string &setname) const
- virtual void **loaded** (**osgTerrain::TerrainTile** ∗tile, const osgDB::ReaderWriter::Options ∗options) const
- bool **readImageLayer** (**osgTerrain::ImageLayer** ∗imageLayer, const osgDB::ReaderWriter::Options ∗options) const
- void **setAllowAll** (bool allowAll)
- void **setMinimumNumOfLayers** (unsigned int numLayers)
- void **setReplaceSwitchLayer** (bool replaceSwitchLayer)

### Protected Types

- typedef std::set< std::string > **SetWhiteList**

### Protected Member Functions

- virtual ~**WhiteListTileLoadedCallback** ()

### Protected Attributes

- bool **_allowAll**
- unsigned int **_minumumNumberOfLayers**
- bool **_replaceSwitchLayer**
- **SetWhiteList _setWhiteList**

### 4.20.1 Detailed Description

Helper callback for managing optional sets of layers, that loading of is deffered to this callback, with this callback working out which layers to load, and how to create fallback versions of the layers.

### 4.20.2 Member Typedef Documentation

#### 4.20.2.1 typedef std::set<std::string> SetWhiteList [protected]

### 4.20.3 Constructor & Destructor Documentation

#### 4.20.3.1 WhiteListTileLoadedCallback ()

#### 4.20.3.2 ~WhiteListTileLoadedCallback () [protected, virtual]

### 4.20.4 Member Function Documentation

#### 4.20.4.1 void allow (const std::string & *setname*) [inline]

#### 4.20.4.2 bool deferExternalLayerLoading () const [virtual]

Implements **TileLoadedCallback** (p. 55).

#### 4.20.4.3 bool getAllowAll () const [inline]

#### 4.20.4.4 unsigned int getMinimumNumOfLayers () const [inline]

#### 4.20.4.5 bool getReplaceSwitchLayer () const [inline]

#### 4.20.4.6 bool layerAcceptable (const std::string & *setname*) const

#### 4.20.4.7 void loaded (osgTerrain::TerrainTile * *tile*, const osgDB::ReaderWriter::Options * *options*) const [virtual]

Implements **TileLoadedCallback** (p. 55).

#### 4.20.4.8 bool readImageLayer (osgTerrain::ImageLayer * *imageLayer*, const osgDB::ReaderWriter::Options * *options*) const

#### 4.20.4.9 void setAllowAll (bool *allowAll*) [inline]

#### 4.20.4.10 void setMinimumNumOfLayers (unsigned int *numLayers*) [inline]

#### 4.20.4.11 void setReplaceSwitchLayer (bool *replaceSwitchLayer*) [inline]

### 4.20.5 Member Data Documentation

#### 4.20.5.1 bool _allowAll [protected]

#### 4.20.5.2 unsigned int _minumumNumberOfLayers [protected]

#### 4.20.5.3 bool _replaceSwitchLayer [protected]

#### 4.20.5.4 SetWhiteList _setWhiteList [protected]

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

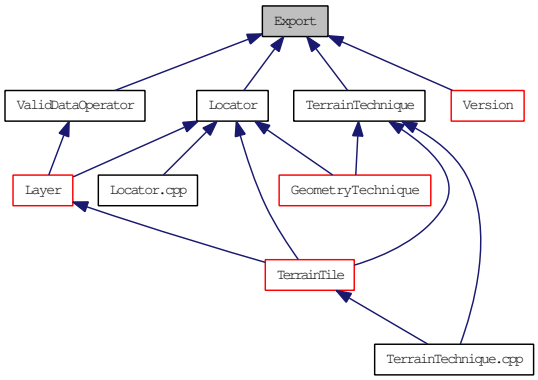
- **TerrainTile**
- **TerrainTile.cpp**

# Chapter 5

# File Documentation

## 5.1 Export File Reference

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




### Namespaces

- namespace **osgTerrain**

  *The **osgTerrain** (p. 7) library is a NodeKit that provides geospecifc terrain rendering support.*

### Defines

- #define **OSGTERRAIN_EXPORT**
- #define **OSGTERRAIN_EXPORT_** 1

### 5.1.1 Define Documentation

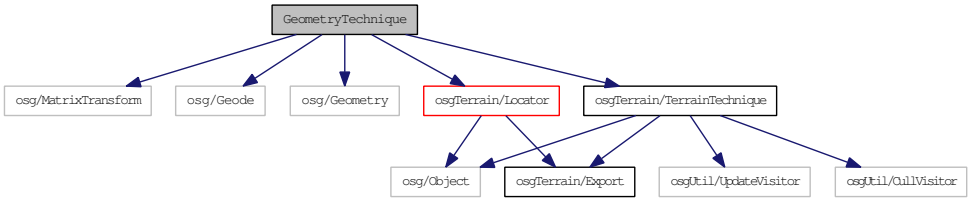
#### 5.1.1.1 #define OSGTERRAIN_EXPORT

#### 5.1.1.2 #define OSGTERRAIN_EXPORT_ 1

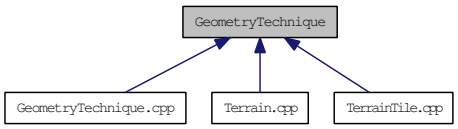
## 5.2 GeometryTechnique File Reference

```
#include <osg/MatrixTransform>
#include <osg/Geode>
#include <osg/Geometry>
#include <osgTerrain/TerrainTechnique>
#include <osgTerrain/Locator>
```

Include dependency graph for GeometryTechnique:




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




### Classes

- struct **BufferData**
- class **GeometryTechnique**

### Namespaces

- namespace **osgTerrain**

  *The **osgTerrain** (p. 7) library is a NodeKit that provides geospecifc terrain rendering support.*

### Defines

- #define **OSGTERRAIN_GEOMETRYTECHNIQUE** 1

### 5.2.1 Define Documentation

#### 5.2.1.1 #define OSGTERRAIN_GEOMETRYTECHNIQUE 1

## 5.3 GeometryTechnique.cpp File Reference

```
#include <osgTerrain/GeometryTechnique>
#include <osgTerrain/TerrainTile>
#include <osgTerrain/Terrain>
#include <osgUtil/SmoothingVisitor>
#include <osgDB/FileUtils>
#include <osg/io_utils>
#include <osg/Texture2D>
#include <osg/Texture1D>
#include <osg/TexEnvCombine>
#include <osg/Program>
#include <osg/Math>
#include <osg/Timer>
```

### Defines

- #define **NEW_COORD_CODE**

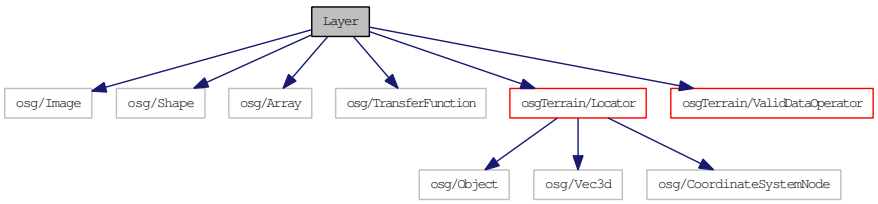
### 5.3.1 Define Documentation

#### 5.3.1.1 #define NEW_COORD_CODE

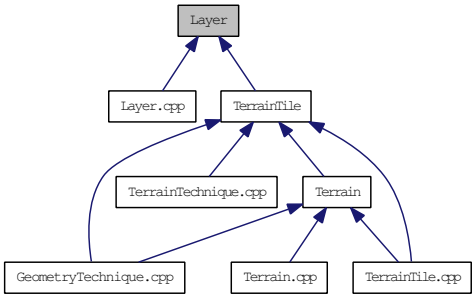
## 5.4 Layer File Reference

```
#include <osg/Image>
#include <osg/Shape>
#include <osg/Array>
#include <osg/TransferFunction>
#include <osgTerrain/Locator>
#include <osgTerrain/ValidDataOperator>
```

Include dependency graph for Layer:




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




### Classes

- class **CompositeLayer**
- struct **CompoundNameLayer**
- class **ContourLayer**
- class **HeightFieldLayer**
- class **ImageLayer**
- class **Layer**
- class **ProxyLayer**
- class **SwitchLayer**

### Namespaces

- namespace **osgTerrain**

  *The **osgTerrain** (p. 7) library is a NodeKit that provides geospecifc terrain rendering support.*

### Defines

- #define **MAXIMUM_NUMBER_OF_LEVELS** 30
- #define **OSGTERRAIN_LAYER** 1

### Functions

- OSGTERRAIN_EXPORT std::string **createCompondSetNameAndFileName** (const std::string &setname, const std::string &filename)

  *Create a compound string in the form set:setname:filename, or just filename if setname is "".*

- OSGTERRAIN_EXPORT void **extractSetNameAndFileName** (const std::string &compoundstring, std::string &setname, std::string &filename)

  *Extact the setname and filename from a compound string in the from set:setname:filename".*

## 5.4.1 Define Documentation

#### 5.4.1.1 #define MAXIMUM_NUMBER_OF_LEVELS 30

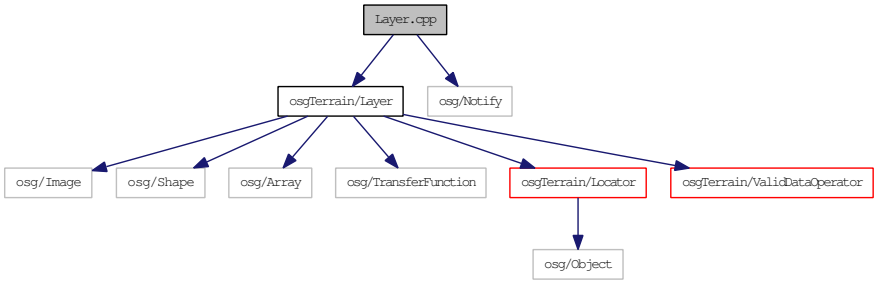
#### 5.4.1.2 #define OSGTERRAIN_LAYER 1

## 5.5 Layer.cpp File Reference

`#include <osgTerrain/Layer>`

`#include <osg/Notify>`

Include dependency graph for Layer.cpp:




### Classes

- struct **TransformOperator**

### Functions

- template<typename T , class O >
  void **_processRow** (unsigned int num, GLenum pixelFormat, T *data, const O &operation)
- template<class O >
  void **processImage** (osg::Image *image, const O &operation)
- template<class O >
  void **processRow** (unsigned int num, GLenum pixelFormat, GLenum dataType, unsigned char *data, const O &operation)

### 5.5.1 Function Documentation

#### 5.5.1.1 void _processRow (unsigned int *num*, GLenum *pixelFormat*, T * *data*, const O & *operation*) [inline]

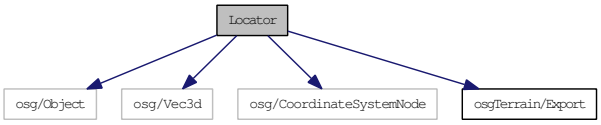
#### 5.5.1.2 void processImage (osg::Image * *image*, const O & *operation*) [inline]

#### 5.5.1.3 void processRow (unsigned int *num*, GLenum *pixelFormat*, GLenum *dataType*, unsigned char * *data*, const O & *operation*) [inline]

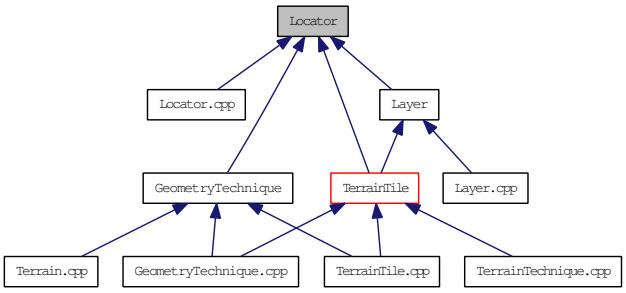
## 5.6 Locator File Reference

```
#include <osg/Object>
#include <osg/Vec3d>
#include <osg/CoordinateSystemNode>
#include <osgTerrain/Export>
```

Include dependency graph for Locator:




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




### Classes

- class **Locator**

### Namespaces

- namespace **osgTerrain**

  *The **osgTerrain** (p. 7) library is a NodeKit that provides geospecifc terrain rendering support.*

### Defines

- #define **OSGTERRAIN_LOCATOR** 1

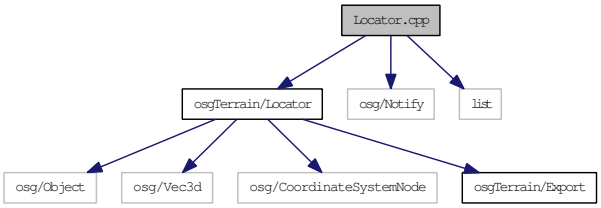
### 5.6.1 Define Documentation

#### 5.6.1.1 #define OSGTERRAIN_LOCATOR 1

## 5.7 Locator.cpp File Reference

```
#include <osgTerrain/Locator>
#include <osg/Notify>
#include <list>
```

Include dependency graph for Locator.cpp:



## 5.8 mainpage.h File Reference

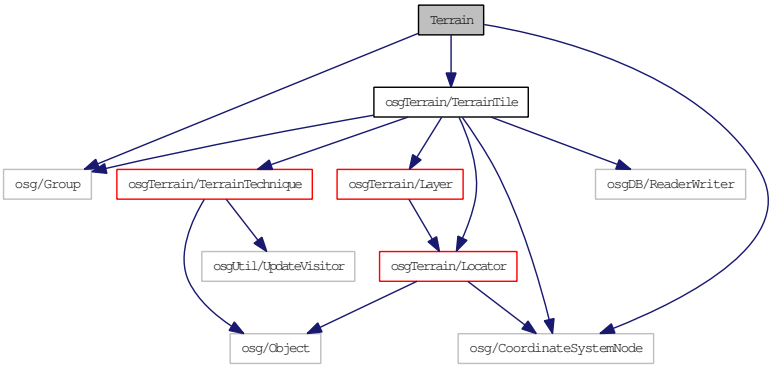
### 5.8.1 Detailed Description

This file contains doxygen special commands and text for the **Main Page** (p. **??**) and some other minor aspects of this documentation. It is not part of the OSG.

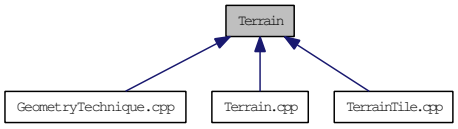
## 5.9 Terrain File Reference

```
#include <osg/Group>
#include <osg/CoordinateSystemNode>
#include <osgTerrain/TerrainTile>
```

Include dependency graph for Terrain:




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




### Classes

- class **Terrain**

  ***Terrain*** (p. 42) *provides a framework for loosely coupling height field data with height rendering algorithms.*

### Namespaces

- namespace **osgTerrain**

  *The* ***osgTerrain*** (p. 7) *library is a NodeKit that provides geospecifc terrain rendering support.*

### Defines

- #define **OSGTerrain** 1

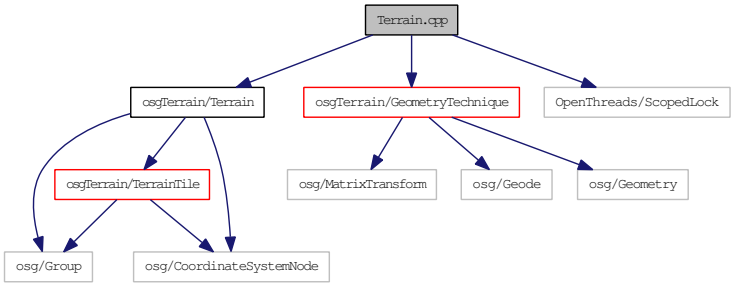
### 5.9.1 Define Documentation

#### 5.9.1.1 #define OSGTerrain 1

## 5.10 Terrain.cpp File Reference

```
#include <osgTerrain/Terrain>
#include <osgTerrain/GeometryTechnique>
#include <OpenThreads/ScopedLock>
```

Include dependency graph for Terrain.cpp:




### Variables

- static unsigned int **s_maxNumTiles** = 0

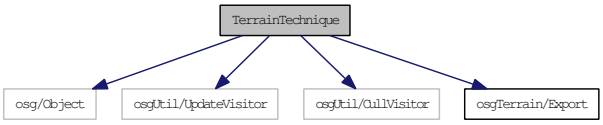
### 5.10.1 Variable Documentation

#### 5.10.1.1 unsigned int s_maxNumTiles = 0 [static]

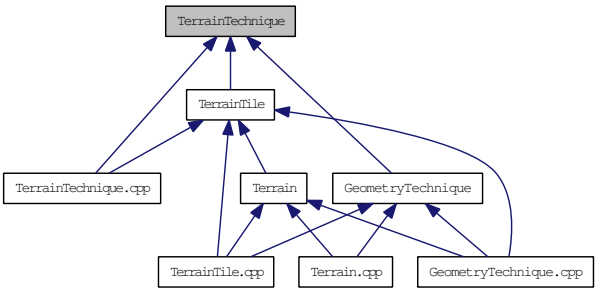
## 5.11 TerrainTechnique File Reference

```
#include <osg/Object>
#include <osgUtil/UpdateVisitor>
#include <osgUtil/CullVisitor>
#include <osgTerrain/Export>
```

Include dependency graph for TerrainTechnique:




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




### Classes

- class **TerrainTechnique**

### Namespaces

- namespace **osgTerrain**

  *The **osgTerrain** (p. 7) library is a NodeKit that provides geospecifc terrain rendering support.*

### Defines

- #define **OSGTERRAIN_TERRAINTECHNIQUE** 1

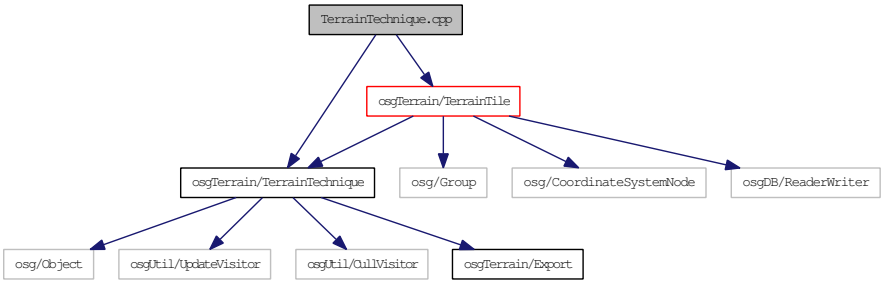
### 5.11.1 Define Documentation

#### 5.11.1.1 #define OSGTERRAIN_TERRAINTECHNIQUE 1

## 5.12 TerrainTechnique.cpp File Reference

```
#include <osgTerrain/TerrainTechnique>
#include <osgTerrain/TerrainTile>
```

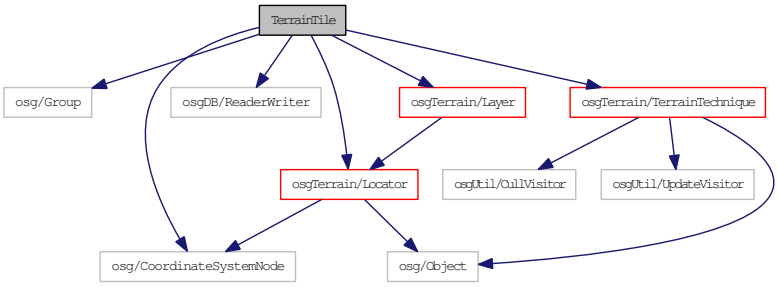
Include dependency graph for TerrainTechnique.cpp:



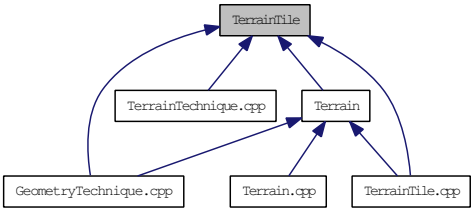
## 5.13 TerrainTile File Reference

```
#include <osg/Group>
#include <osg/CoordinateSystemNode>
#include <osgDB/ReaderWriter>
#include <osgTerrain/TerrainTechnique>
#include <osgTerrain/Layer>
#include <osgTerrain/Locator>
```

Include dependency graph for TerrainTile:




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




### Classes

- class **TerrainTile**

  ***Terrain*** (p. 42) *provides a framework for loosely coupling height field data with height rendering algorithms.*

- class **TileID**
- struct **TileLoadedCallback**

  *Callback for post processing loaded* ***TerrainTile*** (p. 48)*, and for filling in missing elements such as external external imagery.*

- class **WhiteListTileLoadedCallback**

  *Helper callback for managing optional sets of layers, that loading of is deffered to this callback, with this callback working out which layers to load, and how to create fallback versions of the layers.*

### Namespaces

- namespace **osgTerrain**

  *The* ***osgTerrain*** (p. 7) *library is a NodeKit that provides geospecifc terrain rendering support.*

### Defines

- #define **OSGTERRAIN_TERRAINTILE** 1

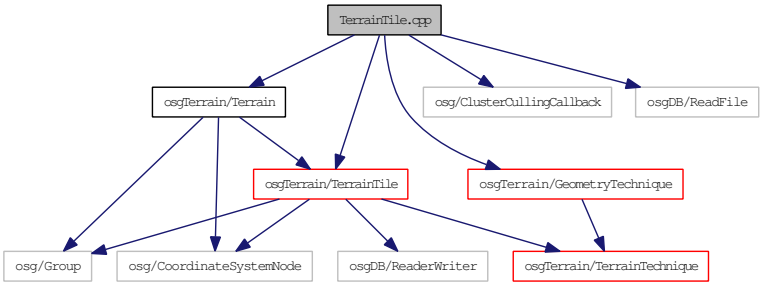
### 5.13.1 Define Documentation

#### 5.13.1.1 #define OSGTERRAIN_TERRAINTILE 1

## 5.14 TerrainTile.cpp File Reference

```
#include <osgTerrain/TerrainTile>
#include <osgTerrain/Terrain>
#include <osgTerrain/GeometryTechnique>
#include <osg/ClusterCullingCallback>
#include <osgDB/ReadFile>
```

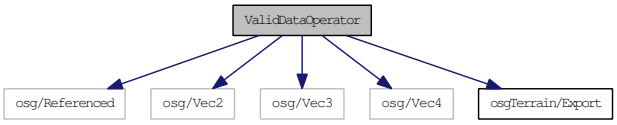
Include dependency graph for TerrainTile.cpp:



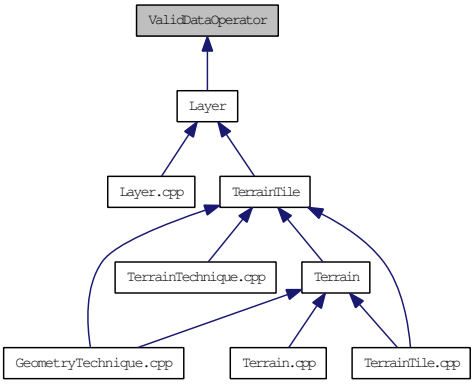
## 5.15 ValidDataOperator File Reference

```
#include <osg/Referenced>
#include <osg/Vec2>
#include <osg/Vec3>
#include <osg/Vec4>
#include <osgTerrain/Export>
```

Include dependency graph for ValidDataOperator:




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




### Classes

- struct **NoDataValue**
- struct **ValidDataOperator**
- struct **ValidRange**

### Namespaces

- namespace **osgTerrain**

  *The **osgTerrain** (p. 7) library is a NodeKit that provides geospecifc terrain rendering support.*

### Defines

- #define **OSGTERRAIN_VALIDDATAOPERATOR** 1

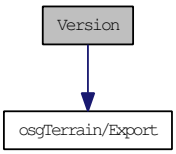
### 5.15.1 Define Documentation

#### 5.15.1.1 #define OSGTERRAIN_VALIDDATAOPERATOR 1

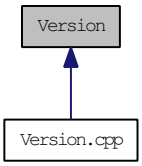
## 5.16 Version File Reference

`#include <osgTerrain/Export>`

Include dependency graph for Version:




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




### Defines

- #define **OSGTERRAIN_VERSION** 1

### Functions

- OSGTERRAIN_EXPORT const char * **osgTerrainGetLibraryName** ()
  ***osgTerrainGetLibraryName()*** *(p. 81) returns the library name in human friendly form.*
- OSGTERRAIN_EXPORT const char * **osgTerrainGetVersion** ()
  ***osgTerrainGetVersion()*** *(p. 81) returns the library version number.*

### 5.16.1 Define Documentation

#### 5.16.1.1 #define OSGTERRAIN_VERSION 1

### 5.16.2 Function Documentation

#### 5.16.2.1 OSGTERRAIN_EXPORT const char* osgTerrainGetLibraryName ()

**osgTerrainGetLibraryName()** (p. 81) returns the library name in human friendly form.

#### 5.16.2.2 OSGTERRAIN_EXPORT const char* osgTerrainGetVersion ()

**osgTerrainGetVersion()** (p. 81) returns the library version number. Numbering convention : OpenSceneGraph-1.0 will return 1.0 from osgTerrainGetVersion.

This C function can be also used to check for the existence of the OpenSceneGraph library using autoconf and its m4 macro AC_CHECK_LIB.

Here is the code to add to your configure.in:

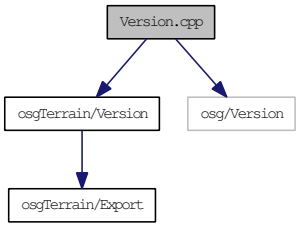
```
#
# Check for the OpenSceneGraph (OSG) Terrain library
#
AC_CHECK_LIB(osg, osgTerrainGetVersion, ,
   [AC_MSG_ERROR(OpenSceneGraph Terrain library not found. See http://www.openscenegraph.org)],)
```

## 5.17 Version.cpp File Reference

`#include <osgTerrain/Version>`

`#include <osg/Version>`

Include dependency graph for Version.cpp:




### Functions

- const char * **osgTerrainGetLibraryName** ()

  ***osgTerrainGetLibraryName()*** *(p. 81) returns the library name in human friendly form.*

- const char * **osgTerrainGetVersion** ()

  ***osgTerrainGetVersion()*** *(p. 81) returns the library version number.*

### 5.17.1 Function Documentation

#### 5.17.1.1 const char* osgTerrainGetLibraryName ()

**osgTerrainGetLibraryName()** (p. 81) returns the library name in human friendly form.

#### 5.17.1.2 const char* osgTerrainGetVersion ()

**osgTerrainGetVersion()** (p. 81) returns the library version number. Numbering convention : OpenSceneGraph-1.0 will return 1.0 from osgTerrainGetVersion.

This C function can be also used to check for the existence of the OpenSceneGraph library using autoconf and its m4 macro AC_CHECK_LIB.

Here is the code to add to your configure.in:

```
#
# Check for the OpenSceneGraph (OSG) Terrain library
#
AC_CHECK_LIB(osg, osgTerrainGetVersion, ,
   [AC_MSG_ERROR(OpenSceneGraph Terrain library not found. See http://www.openscenegraph.org)],)
```

# Index

## - Symbols -

## - A -



## - C -



## - D -



## - E -



## - F -



## - G -

## - H -



## - I -



## - L -

## - M -



## - N -



## - O -

## - P -



## - R -



## - S -

## - T -



## - U -



## - V -



## - W -



## - X -



## - Y -