

GAMETOOLS

ADVANCED TOOLS FOR DEVELOPING HIGHLY REALISTIC COMPUTER GAMES

DUMMY VISIBILITY MODULES

Document identifier:	GameTools-3-D3.1-01-1-0- Dummy Visibility Modules
Date:	29/04/2005
Work package:	WP03: Visibility
Partner(s):	VUT
Leading Partner:	VUT
Document status:	DRAFT

Deliverable identifier: **D3.1**

Abstract: Dummy Modules Report for Visibility



Delivery Slip

	Name	Partner	Date	Signature
From	Michael Wimmer	VUT	29-04-05	
Reviewed by	Moderator and reviewers			
Approved by				

Document Log

Issue	Date	Comment	Author
1-0	29-04-05	First draft	Jiri Bittner (VUT)

Document Change Record

Issue	Item	Reason for Change

Files

Software Products	User files / URL
Word	gametools-ist-2-004363-3-d3.1-01-1-0-dummy visibility modules.doc

Contents

1	Overview	1
1.1	Module Class Hierarchy	1
1.2	Visibility Culling	1
1.3	Visibility Queries	2
1.4	Visibility Preprocessing	2
2	GameTools Visibility Modules Namespace Index	3
2.1	GameTools Visibility Modules Namespace List	3
3	GameTools Visibility Modules Hierarchical Index	5
3.1	GameTools Visibility Modules Class Hierarchy	5
4	GameTools Visibility Modules Compound Index	7
4.1	GameTools Visibility Modules Compound List	7
5	GameTools Visibility Modules File Index	9
5.1	GameTools Visibility Modules File List	9
6	GameTools Visibility Modules Namespace Documentation	11
6.1	GtpVisibility Namespace Reference	11
6.2	GtpVisibilityPreprocessor Namespace Reference	13
6.3	Ogre Namespace Reference	14
6.4	std Namespace Reference	15
7	GameTools Visibility Modules Class Documentation	17
7.1	GtpVisibilityPreprocessor::AxisAlignedBox3 Class Reference	17
7.2	Ogre::BspHierarchyInterface Class Reference	18
7.3	GtpVisibilityPreprocessor::BSPInterior Class Reference	22
7.4	GtpVisibilityPreprocessor::BSPLeaf Class Reference	24
7.5	GtpVisibilityPreprocessor::BSPNode Class Reference	26

7.6	GtpVisibilityPreprocessor::BSPTree Class Reference	28
7.7	GtpVisibility::CoherentHierarchicalCullingManager Class Reference	29
7.8	GtpVisibility::CullingManager Class Reference	31
7.9	GtpVisibility::DummyPreprocessingManager Class Reference	33
7.10	GtpVisibility::DummyQueryManager Class Reference	35
7.11	GtpVisibilityPreprocessor::ExactPreprocessor Class Reference	37
7.12	GtpVisibility::FrustumCullingManager Class Reference	38
7.13	GtpVisibility::GreaterDistance< T > Class Template Reference	39
7.14	GtpVisibility::HierarchyInterface Class Reference	40
7.15	GtpVisibilityPreprocessor::KdInterior Class Reference	46
7.16	GtpVisibilityPreprocessor::KdLeaf Class Reference	48
7.17	GtpVisibilityPreprocessor::KdNode Class Reference	50
7.18	GtpVisibilityPreprocessor::KdTree Class Reference	52
7.19	GtpVisibilityPreprocessor::Mesh Class Reference	54
7.20	GtpVisibility::MeshInfo Class Reference	56
7.21	GtpVisibility::NodeInfo Class Reference	57
7.22	GtpVisibility::OcclusionQuery Class Reference	58
7.23	Ogre::OctreeHierarchyInterface Class Reference	60
7.24	GtpVisibilityPreprocessor::Patch Class Reference	65
7.25	GtpVisibilityPreprocessor::Plane3 Class Reference	66
7.26	Ogre::PlatformHierarchyInterface Class Reference	67
7.27	Ogre::PlatformOcclusionQuery Class Reference	72
7.28	GtpVisibility::PreprocessingManager Class Reference	74
7.29	GtpVisibilityPreprocessor::Preprocessor Class Reference	78
7.30	GtpVisibility::QueryManager Class Reference	82
7.31	GtpVisibilityPreprocessor::SamplingPreprocessor Class Reference	85
7.32	GtpVisibilityPreprocessor::SceneGraph Class Reference	87
7.33	GtpVisibilityPreprocessor::SceneGraphNode Class Reference	88
7.34	Ogre::SceneNodeHierarchyInterface Class Reference	89
7.35	Ogre::SolidHalfBoundingBox Class Reference	93
7.36	GtpVisibility::StopAndWaitCullingManager Class Reference	95
7.37	GtpVisibilityPreprocessor::Vector3 Class Reference	96
7.38	Ogre::VisibilityBspSceneManager Class Reference	97
7.39	Ogre::VisibilityDotSceneManager Class Reference	100
7.40	GtpVisibility::VisibilityEnvironment Class Reference	103
7.41	GtpVisibility::VisibilityManager Class Reference	104

7.42	Ogre::VisibilityOctreeSceneManager Class Reference	106
7.43	Ogre::VisibilitySceneManager Class Reference	109
7.44	Ogre::VisibilityTerrainSceneManager Class Reference	111
8	GameTools Visibility Modules File Documentation	115
8.1	GtpVisibility/include/CoherentHierarchicalCullingManager.h File Reference	115
8.2	GtpVisibility/include/CullingManager.h File Reference	116
8.3	GtpVisibility/include/DistanceQueue.h File Reference	117
8.4	GtpVisibility/include/DummyPreprocessingManager.h File Reference	118
8.5	GtpVisibility/include/DummyQueryManager.h File Reference	119
8.6	GtpVisibility/include/FrustumCullingManager.h File Reference	120
8.7	GtpVisibility/include/HierarchyInterface.h File Reference	121
8.8	GtpVisibility/include/OcclusionQuery.h File Reference	122
8.9	GtpVisibility/include/PreprocessingManager.h File Reference	123
8.10	GtpVisibility/include/QueryManager.h File Reference	124
8.11	GtpVisibility/include/StopAndWaitCullingManager.h File Reference	125
8.12	GtpVisibility/include/VisibilityAxisAlignedBox.h File Reference	126
8.13	GtpVisibility/include/VisibilityCamera.h File Reference	127
8.14	GtpVisibility/include/VisibilityEnvironment.h File Reference	128
8.15	GtpVisibility/include/VisibilityInfo.h File Reference	129
8.16	GtpVisibility/include/VisibilityManager.h File Reference	130
8.17	GtpVisibility/include/VisibilityMesh.h File Reference	131
8.18	GtpVisibility/include/VisibilityRay.h File Reference	132
8.19	GtpVisibility/include/VisibilityVector3.h File Reference	133
8.20	GtpVisibility/src/CoherentHierarchicalCullingManager.cpp File Reference	134
8.21	GtpVisibility/src/CullingManager.cpp File Reference	135
8.22	GtpVisibility/src/DummyPreprocessingManager.cpp File Reference	136
8.23	GtpVisibility/src/DummyQueryManager.cpp File Reference	137
8.24	GtpVisibility/src/FrustumCullingManager.cpp File Reference	138
8.25	GtpVisibility/src/HierarchyInterface.cpp File Reference	139
8.26	GtpVisibility/src/PreprocessingManager.cpp File Reference	140
8.27	GtpVisibility/src/QueryManager.cpp File Reference	141
8.28	GtpVisibility/src/StopAndWaitCullingManager.cpp File Reference	142
8.29	GtpVisibility/src/VisibilityEnvironment.cpp File Reference	143
8.30	GtpVisibility/src/VisibilityManager.cpp File Reference	144
8.31	GtpVisibilityPreprocessor/include/AxisAlignedBox3.h File Reference	145
8.32	GtpVisibilityPreprocessor/include/Containers.h File Reference	146

8.33	GtpVisibilityPreprocessor/include/ExactPreprocessor.h File Reference	147
8.34	GtpVisibilityPreprocessor/include/KdTree.h File Reference	148
8.35	GtpVisibilityPreprocessor/include/Mesh.h File Reference	149
8.36	GtpVisibilityPreprocessor/include/Plane3.h File Reference	150
8.37	GtpVisibilityPreprocessor/include/Preprocessor.h File Reference	151
8.38	GtpVisibilityPreprocessor/include/SamplingPreprocessor.h File Reference	152
8.39	GtpVisibilityPreprocessor/include/SceneGraph.h File Reference	153
8.40	GtpVisibilityPreprocessor/include/Vector3.h File Reference	154
8.41	GtpVisibilityPreprocessor/include/ViewCellBSP.h File Reference	155
8.42	GtpVisibilityPreprocessor/src/ExactPreprocessor.cpp File Reference	156
8.43	GtpVisibilityPreprocessor/src/KdTree.cpp File Reference	157
8.44	GtpVisibilityPreprocessor/src/Preprocessor.cpp File Reference	158
8.45	GtpVisibilityPreprocessor/src/SamplingPreprocessor.cpp File Reference	159
8.46	Ogre/include/OgreBspHierarchyInterface.h File Reference	160
8.47	Ogre/include/OgreOctreeHierarchyInterface.h File Reference	161
8.48	Ogre/include/OgrePlatformHierarchyInterface.h File Reference	162
8.49	Ogre/include/OgrePlatformOcclusionQuery.h File Reference	163
8.50	Ogre/include/OgreSceneNodeHierarchyInterface.h File Reference	164
8.51	Ogre/include/OgreSolidHalfBoundingBox.h File Reference	165
8.52	Ogre/include/OgreVisibilityBspSceneManager.h File Reference	166
8.53	Ogre/include/OgreVisibilityDotSceneManager.h File Reference	167
8.54	Ogre/include/OgreVisibilityOctreeSceneManager.h File Reference	168
8.55	Ogre/include/OgreVisibilitySceneManager.h File Reference	169
8.56	Ogre/include/OgreVisibilityTerrainSceneManager.h File Reference	170
8.57	Ogre/src/OgreBspHierarchyInterface.cpp File Reference	171
8.58	Ogre/src/OgreOctreeHierarchyInterface.cpp File Reference	172
8.59	Ogre/src/OgrePlatformHierarchyInterface.cpp File Reference	173
8.60	Ogre/src/OgrePlatformOcclusionQuery.cpp File Reference	174
8.61	Ogre/src/OgreSceneNodeHierarchyInterface.cpp File Reference	175
8.62	Ogre/src/OgreSolidHalfBoundingBox.cpp File Reference	176
8.63	Ogre/src/OgreVisibilityBspSceneManager.cpp File Reference	177
8.64	Ogre/src/OgreVisibilityDotSceneManager.cpp File Reference	178
8.65	Ogre/src/OgreVisibilityOctreeSceneManager.cpp File Reference	179
8.66	Ogre/src/OgreVisibilitySceneManager.cpp File Reference	180
8.67	Ogre/src/OgreVisibilitySceneManagerDll.cpp File Reference	181
8.68	Ogre/src/OgreVisibilityTerrainSceneManager.cpp File Reference	182

Chapter 1

Overview

The visibility work package aims to develop methods for making use of restricted visibility in large scenes. The major goal of these methods is to avoid wasting of the computational power on currently invisible parts of the scene. To address this goal we develop new methods for online visibility culling and offline visibility preprocessing. The online visibility culling is either used to render only visible part of the scene or to provide queries which determine visible part of the scene in runtime. Visibility preprocessing provides partitioning of the view space into view cells and for each view cell it calculates a potentially visible set.

1.1 Module Class Hierarchy

The visibility work package consist of a module which is integrated into the game engine and an external module which deals with preprocessing. One of the goals of the design of the module class hierarchy has been easy portability of the developed methods to other game engines. The resulting class structure clearly separates the actual engine dependent parts from the game engine independent algorithmic parts. We have used name spaces to separate the engine dependent parts (name space Ogre) from the engine independent parts (name space GtpVisibility). The external visibility preprocessing module defines its own name space (GtpVisibilityPreprocessor).

An overview of the important classes of the visibility work package and their integration into the Ogre engine is depicted in Figure 1.1.

The diagram shows that the module integrated into the engine consists of three main parts (CullingManager, QueryManager, PreprocessingManager) which are encapsulated in a helper class (Visibility Manager). The helper class then deals with the initialization of the contained parts and allows an easy communication between them.

1.2 Visibility Culling

Visibility culling seamlessly integrates into the rendering loop and eliminates most invisible objects from being send to the pipeline. Visibility culling is implemented by instances of the CullingManager class. We will provide several implementations which can be easily switched at runtime and so the best technique for the particular scene and scene representation can be selected. Note that the Ogre integration design allows to exploit all already existing methods for scene management, such as octree, BSP tree, or plain scene graph.

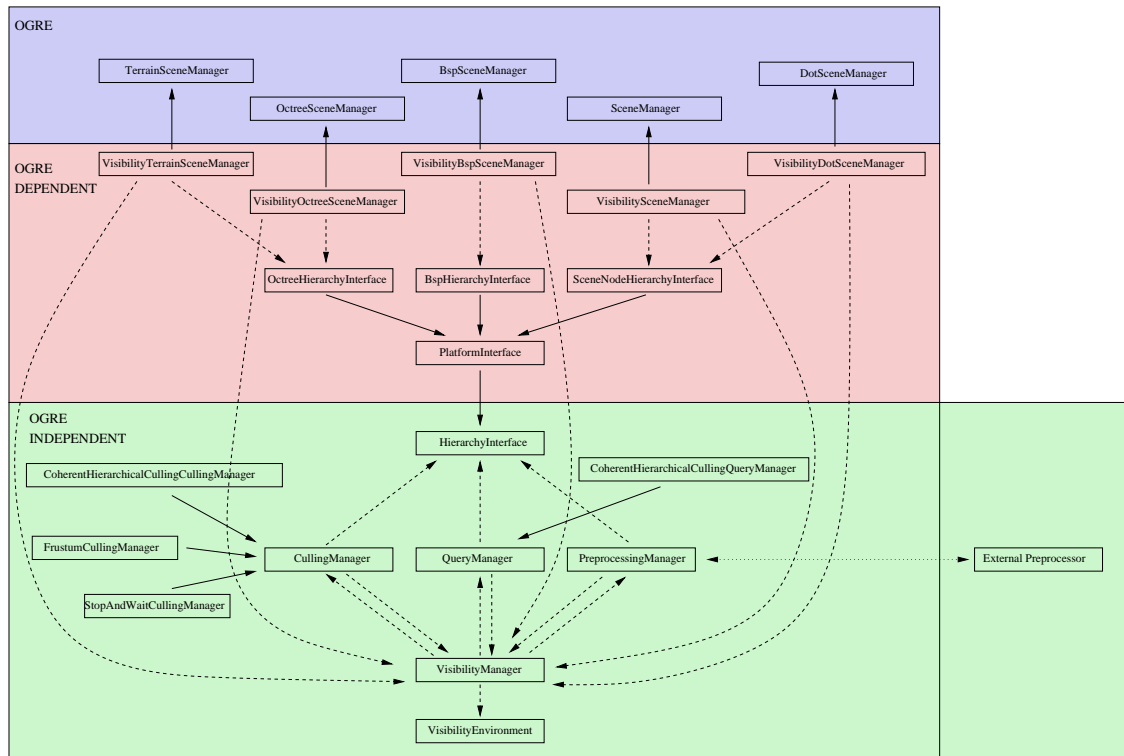


Figure 1.1: Overview of the visibility work packages classes and their integration into the Ogre game engine.

1.3 Visibility Queries

Visibility queries determine the visible geometry for a given view point in the scene. Additionally they can also report the visible scene hierarchy nodes. Visibility queries are implemented by instances of the QueryManager class. Note that some of these instances will make use of the preprocessed visibility data through the use of PreprocessingManager.

1.4 Visibility Preprocessing

Visibility preprocessing precalculates visibility for all viewpoints corresponding to cells of a view space partitioning. The visibility preprocessor is implemented as a standalone module which imports files generated by the PreprocessingManager. The results of the visibility computation will be exported to the file which can then be loaded by the PreprocessingManager and used inside the engine.

Apart from the scene definition file the preprocessor will be able to import the view cell definition. In this case it will assume the view cells have to be described as meshes satisfying set of requirements. Alternatively the module can generate view cells by automatic view space partitioning.

Chapter 2

GameTools Visibility Modules Namespace Index

2.1 GameTools Visibility Modules Namespace List

Here is a list of all namespaces with brief descriptions:

GtpVisibility	11
GtpVisibilityPreprocessor	13
Ogre	14
std	15

Chapter 3

GameTools Visibility Modules Hierarchical Index

3.1 GameTools Visibility Modules Class Hierarchy

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

GtpVisibilityPreprocessor::AxisAlignedBox3	17
GtpVisibilityPreprocessor::BSPNode	26
GtpVisibilityPreprocessor::BSPInterior	22
GtpVisibilityPreprocessor::BSPLeaf	24
GtpVisibilityPreprocessor::BSPTree	28
GtpVisibility::CullingManager	31
GtpVisibility::CoherentHierarchicalCullingManager	29
GtpVisibility::FrustumCullingManager	38
GtpVisibility::StopAndWaitCullingManager	95
GtpVisibility::GreaterDistance< T >	39
GtpVisibility::HierarchyInterface	40
Ogre::PlatformHierarchyInterface	67
Ogre::BspHierarchyInterface	18
Ogre::OctreeHierarchyInterface	60
Ogre::SceneNodeHierarchyInterface	89
GtpVisibilityPreprocessor::KdNode	50
GtpVisibilityPreprocessor::KdInterior	46
GtpVisibilityPreprocessor::KdLeaf	48
GtpVisibilityPreprocessor::KdTree	52
GtpVisibilityPreprocessor::Mesh	54
GtpVisibility::MeshInfo	56
GtpVisibility::NodeInfo	57
GtpVisibility::OcclusionQuery	58
Ogre::PlatformOcclusionQuery	72
GtpVisibilityPreprocessor::Patch	65
GtpVisibilityPreprocessor::Plane3	66
GtpVisibility::PreprocessingManager	74
GtpVisibility::DummyPreprocessingManager	33
GtpVisibilityPreprocessor::Preprocessor	78

GtpVisibilityPreprocessor::ExactPreprocessor	37
GtpVisibilityPreprocessor::SamplingPreprocessor	85
GtpVisibility::QueryManager	82
GtpVisibility::DummyQueryManager	35
GtpVisibilityPreprocessor::SceneGraph	87
GtpVisibilityPreprocessor::SceneGraphNode	88
Ogre::SolidHalfBoundingBox	93
GtpVisibilityPreprocessor::Vector3	96
Ogre::VisibilityBspSceneManager	97
Ogre::VisibilityDotSceneManager	100
GtpVisibility::VisibilityEnvironment	103
GtpVisibility::VisibilityManager	104
Ogre::VisibilityOctreeSceneManager	106
Ogre::VisibilitySceneManager	109
Ogre::VisibilityTerrainSceneManager	111

Chapter 4

GameTools Visibility Modules Compound Index

4.1 GameTools Visibility Modules Compound List

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

GtpVisibilityPreprocessor::AxisAlignedBox3	17
Ogre::BspHierarchyInterface	18
GtpVisibilityPreprocessor::BSPInterior	22
GtpVisibilityPreprocessor::BSPLeaf	24
GtpVisibilityPreprocessor::BSPNode	26
GtpVisibilityPreprocessor::BSPTree	28
GtpVisibility::CoherentHierarchicalCullingManager	29
GtpVisibility::CullingManager	31
GtpVisibility::DummyPreprocessingManager	33
GtpVisibility::DummyQueryManager	35
GtpVisibilityPreprocessor::ExactPreprocessor	37
GtpVisibility::FrustumCullingManager	38
GtpVisibility::GreaterDistance< T >	39
GtpVisibility::HierarchyInterface	40
GtpVisibilityPreprocessor::KdInterior	46
GtpVisibilityPreprocessor::KdLeaf	48
GtpVisibilityPreprocessor::KdNode	50
GtpVisibilityPreprocessor::KdTree	52
GtpVisibilityPreprocessor::Mesh	54
GtpVisibility::MeshInfo	56
GtpVisibility::NodeInfo	57
GtpVisibility::OcclusionQuery	58
Ogre::OctreeHierarchyInterface	60
GtpVisibilityPreprocessor::Patch	65
GtpVisibilityPreprocessor::Plane3	66
Ogre::PlatformHierarchyInterface	67
Ogre::PlatformOcclusionQuery	72
GtpVisibility::PreprocessingManager	74
GtpVisibilityPreprocessor::Preprocessor	78
GtpVisibility::QueryManager	82
GtpVisibilityPreprocessor::SamplingPreprocessor	85

GtpVisibilityPreprocessor::SceneGraph	87
GtpVisibilityPreprocessor::SceneGraphNode	88
Ogre::SceneNodeHierarchyInterface	89
Ogre::SolidHalfBoundingBox	93
GtpVisibility::StopAndWaitCullingManager	95
GtpVisibilityPreprocessor::Vector3	96
Ogre::VisibilityBspSceneManager	97
Ogre::VisibilityDotSceneManager	100
GtpVisibility::VisibilityEnvironment	103
GtpVisibility::VisibilityManager	104
Ogre::VisibilityOctreeSceneManager	106
Ogre::VisibilitySceneManager	109
Ogre::VisibilityTerrainSceneManager	111

Chapter 5

GameTools Visibility Modules File Index

5.1 GameTools Visibility Modules File List

Here is a list of all files with brief descriptions:

GtpVisibility/include/CoherentHierarchicalCullingManager.h	115
GtpVisibility/include/CullingManager.h	116
GtpVisibility/include/DistanceQueue.h	117
GtpVisibility/include/DummyPreprocessingManager.h	118
GtpVisibility/include/DummyQueryManager.h	119
GtpVisibility/include/FrustumCullingManager.h	120
GtpVisibility/include/HierarchyInterface.h	121
GtpVisibility/include/OcclusionQuery.h	122
GtpVisibility/include/PreprocessingManager.h	123
GtpVisibility/include/QueryManager.h	124
GtpVisibility/include/StopAndWaitCullingManager.h	125
GtpVisibility/include/VisibilityAxisAlignedBox.h	126
GtpVisibility/include/VisibilityCamera.h	127
GtpVisibility/include/VisibilityEnvironment.h	128
GtpVisibility/include/VisibilityInfo.h	129
GtpVisibility/include/VisibilityManager.h	130
GtpVisibility/include/VisibilityMesh.h	131
GtpVisibility/include/VisibilityRay.h	132
GtpVisibility/include/VisibilityVector3.h	133
GtpVisibility/src/CoherentHierarchicalCullingManager.cpp	134
GtpVisibility/src/CullingManager.cpp	135
GtpVisibility/src/DummyPreprocessingManager.cpp	136
GtpVisibility/src/DummyQueryManager.cpp	137
GtpVisibility/src/FrustumCullingManager.cpp	138
GtpVisibility/src/HierarchyInterface.cpp	139
GtpVisibility/src/PreprocessingManager.cpp	140
GtpVisibility/src/QueryManager.cpp	141
GtpVisibility/src/StopAndWaitCullingManager.cpp	142
GtpVisibility/src/VisibilityEnvironment.cpp	143
GtpVisibility/src/VisibilityManager.cpp	144
GtpVisibilityPreprocessor/include/AxisAlignedBox3.h	145

GtpVisibilityPreprocessor/include/Containers.h	146
GtpVisibilityPreprocessor/include/ExactPreprocessor.h	147
GtpVisibilityPreprocessor/include/KdTree.h	148
GtpVisibilityPreprocessor/include/Mesh.h	149
GtpVisibilityPreprocessor/include/Plane3.h	150
GtpVisibilityPreprocessor/include/Preprocessor.h	151
GtpVisibilityPreprocessor/include/SamplingPreprocessor.h	152
GtpVisibilityPreprocessor/include/SceneGraph.h	153
GtpVisibilityPreprocessor/include/Vector3.h	154
GtpVisibilityPreprocessor/include/ViewCellBSP.h	155
GtpVisibilityPreprocessor/src/ExactPreprocessor.cpp	156
GtpVisibilityPreprocessor/src/KdTree.cpp	157
GtpVisibilityPreprocessor/src/Preprocessor.cpp	158
GtpVisibilityPreprocessor/src/SamplingPreprocessor.cpp	159
Ogre/include/OgreBspHierarchyInterface.h	160
Ogre/include/OgreOctreeHierarchyInterface.h	161
Ogre/include/OgrePlatformHierarchyInterface.h	162
Ogre/include/OgrePlatformOcclusionQuery.h	163
Ogre/include/OgreSceneNodeHierarchyInterface.h	164
Ogre/include/OgreSolidHalfBoundingBox.h	165
Ogre/include/OgreVisibilityBspSceneManager.h	166
Ogre/include/OgreVisibilityDotSceneManager.h	167
Ogre/include/OgreVisibilityOctreeSceneManager.h	168
Ogre/include/OgreVisibilitySceneManager.h	169
Ogre/include/OgreVisibilityTerrainSceneManager.h	170
Ogre/src/OgreBspHierarchyInterface.cpp	171
Ogre/src/OgreOctreeHierarchyInterface.cpp	172
Ogre/src/OgrePlatformHierarchyInterface.cpp	173
Ogre/src/OgrePlatformOcclusionQuery.cpp	174
Ogre/src/OgreSceneNodeHierarchyInterface.cpp	175
Ogre/src/OgreSolidHalfBoundingBox.cpp	176
Ogre/src/OgreVisibilityBspSceneManager.cpp	177
Ogre/src/OgreVisibilityDotSceneManager.cpp	178
Ogre/src/OgreVisibilityOctreeSceneManager.cpp	179
Ogre/src/OgreVisibilitySceneManager.cpp	180
Ogre/src/OgreVisibilitySceneManagerDll.cpp	181
Ogre/src/OgreVisibilityTerrainSceneManager.cpp	182

Chapter 6

GameTools Visibility Modules Namespace Documentation

6.1 GtpVisibility Namespace Reference

Compounds

- class [CoherentHierarchicalCullingManager](#)
- class [CullingManager](#)
- class [GreaterDistance](#)
- class [DummyPreprocessingManager](#)
- class [DummyQueryManager](#)
- class [FrustumCullingManager](#)
- class [HierarchyInterface](#)
- class [OcclusionQuery](#)
- class [PreprocessingManager](#)
- class [QueryManager](#)
- class [StopAndWaitCullingManager](#)
- class [VisibilityEnvironment](#)
- class [NodeInfo](#)
- class [MeshInfo](#)
- class [VisibilityManager](#)

Typedefs

- typedef std::pair< [HierarchyNode](#) *, [OcclusionQuery](#) * > [QueryPair](#)
- typedef std::queue< [QueryPair](#) > [QueryQueue](#)
- typedef void [HierarchyNode](#)
- typedef std::priority_queue< [HierarchyNode](#) *, std::vector< [HierarchyNode](#) * >, [GreaterDistance](#)< std::vector< [HierarchyNode](#) * >::value_type > > [DistanceQueue](#)
- typedef Ogre::AxisAlignedBox [AxisAlignedBox](#)
- typedef Ogre::Camera [Camera](#)
- typedef Ogre::Mesh [Mesh](#)
- typedef Ogre::Ray [Ray](#)
- typedef Ogre::Vector3 [Vector3](#)

6.1.1 Detailed Description

This namespace includes all classes which are created by the VUT (Vienna University of Technology for the Visibility module of the GTP (GameTools Project) (www.gametools.org), and are not directly derived from an [Ogre](#) class.

6.1.2 Typedef Documentation

6.1.2.1 `typedef std::pair<HierarchyNode *, OcclusionQuery *> GtpVisibility::QueryPair`

6.1.2.2 `typedef std::queue<QueryPair> GtpVisibility::QueryQueue`

6.1.2.3 `typedef void GtpVisibility::HierarchyNode`

6.1.2.4 `typedef std::priority_queue<HierarchyNode *, std::vector<HierarchyNode *>, GreaterDistance<std::vector<HierarchyNode *>::value_type> > GtpVisibility::DistanceQueue`

A priority queue where closer hierarchy nodes are given a higher priority.

6.1.2.5 `typedef Ogre::AxisAlignedBox GtpVisibility::AxisAlignedBox`

This class currently uses the native [Ogre](#) `AxisAlignedBox` when compiled with the [Ogre](#) platform

6.1.2.6 `typedef Ogre::Camera GtpVisibility::Camera`

Camera class currently uses the native [Ogre](#) camera when compiled with the [Ogre](#) platform

6.1.2.7 `typedef Ogre::Mesh GtpVisibility::Mesh`

6.1.2.8 `typedef Ogre::Ray GtpVisibility::Ray`

Ray class currently uses native [Ogre](#) ray when compiled with the [Ogre](#) platform

6.1.2.9 `typedef Ogre::Vector3 GtpVisibility::Vector3`

Vector3 class currently uses the native [Ogre](#) vector when compiled with the [Ogre](#) platform

6.2 GtpVisibilityPreprocessor Namespace Reference

Compounds

- class [AxisAlignedBox3](#)
- class [ExactPreprocessor](#)
- class [KdTree](#)
- class [KdNode](#)
- class [KdInterior](#)
- class [KdLeaf](#)
- class [Patch](#)
- class [Mesh](#)
- class [Plane3](#)
- class [Preprocessor](#)
- class [SamplingPreprocessor](#)
- class [SceneGraphNode](#)
- class [SceneGraph](#)
- class [Vector3](#)
- class [BSPNode](#)
- class [BSPInterior](#)
- class [BSPLeaf](#)
- class [BSPTree](#)

Typedefs

- typedef vector< [Mesh](#) * > [MeshContainer](#)
- typedef vector< [ViewCell](#) * > [ViewCellContainer](#)
- typedef vector< [HierarchyNode](#) * > [NodeContainer](#)

6.2.1 Detailed Description

Namespace for the external visibility preprocessor

This namespace includes all classes which are created by the VUT (Vienna University of Technology) for the External Visibility [Preprocessor](#) of the GTP (GameTools Project) (www.gametools.org).

6.2.2 Typedef Documentation

6.2.2.1 typedef vector<[Mesh](#) *> GtpVisibilityPreprocessor::MeshContainer

Container for [Mesh](#) pointers primarily for the use within the kDTree and BSP hierarchies

6.2.2.2 typedef vector<[ViewCell](#) *> GtpVisibilityPreprocessor::ViewCellContainer

Container for [ViewCell](#) pointers primarily for the use within the kDTree and BSP hierarchies

6.2.2.3 typedef vector<[HierarchyNode](#) *> GtpVisibilityPreprocessor::NodeContainer

Container for [HierarchyNodes](#) pointers primarily for the use within the kDTree and BSP hierarchies

6.3 Ogre Namespace Reference

Compounds

- class [BspHierarchyInterface](#)
- class [OctreeHierarchyInterface](#)
- class [PlatformHierarchyInterface](#)
- class [PlatformOcclusionQuery](#)
- class [SceneNodeHierarchyInterface](#)
- class [SolidHalfBoundingBox](#)
- class [VisibilityBspSceneManager](#)
- class [VisibilityDotSceneManager](#)
- class [VisibilityOctreeSceneManager](#)
- class [VisibilitySceneManager](#)
- class [VisibilityTerrainSceneManager](#)

Functions

- void [dllStartPlugin](#) (void)
- void [dllStopPlugin](#) (void)

Variables

- OcclusionCullingOctreeSceneManager * [cullingOctreePlugin](#)
- OcclusionCullingTerrainSceneManager * [cullingTerrainPlugin](#)
- HeightmapTerrainPageSource * [heightmapTerrainPageSource](#)

6.3.1 Function Documentation

6.3.1.1 void [dllStartPlugin](#) (void)

6.3.1.2 void [dllStopPlugin](#) (void)

6.3.2 Variable Documentation

6.3.2.1 OcclusionCullingOctreeSceneManager* [Ogre::cullingOctreePlugin](#)

6.3.2.2 OcclusionCullingTerrainSceneManager* [Ogre::cullingTerrainPlugin](#)

6.3.2.3 HeightmapTerrainPageSource* [Ogre::heightmapTerrainPageSource](#)

6.4 std Namespace Reference

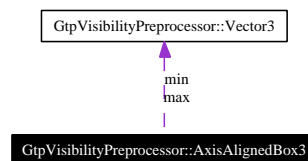
Chapter 7

GameTools Visibility Modules Class Documentation

7.1 GtpVisibilityPreprocessor::AxisAlignedBox3 Class Reference

```
#include <AxisAlignedBox3.h>
```

Collaboration diagram for GtpVisibilityPreprocessor::AxisAlignedBox3:



Protected Attributes

- [Vector3 min](#)
- [Vector3 max](#)

7.1.1 Detailed Description

Axis aligned box

7.1.2 Member Data Documentation

7.1.2.1 [Vector3](#) `GtpVisibilityPreprocessor::AxisAlignedBox3::min` [protected]

7.1.2.2 [Vector3](#) `GtpVisibilityPreprocessor::AxisAlignedBox3::max` [protected]

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

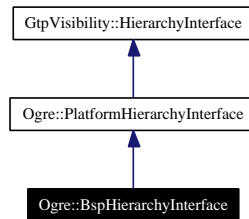
- `GtpVisibilityPreprocessor/include/AxisAlignedBox3.h`

7.2 Ogre::BspHierarchyInterface Class Reference

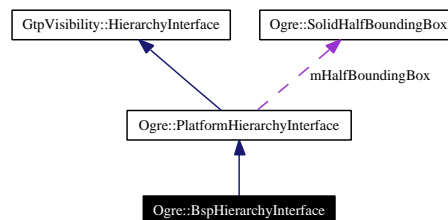
```
#include <OgreBspHierarchyInterface.h>
```

Inherits [Ogre::PlatformHierarchyInterface](#).

Inheritance diagram for `Ogre::BspHierarchyInterface`:



Collaboration diagram for `Ogre::BspHierarchyInterface`:



Public Methods

- [BspHierarchyInterface](#) (`SceneManager *sm`, `RenderSystem *rsys`)
- `void PullUpVisibility` (`GtpVisibility::HierarchyNode *node`)
- `void TraverseNode` (`GtpVisibility::HierarchyNode *node`)
- `void RenderNode` (`GtpVisibility::HierarchyNode *node`)
- `bool IsLeaf` (`GtpVisibility::HierarchyNode *node`)
- `bool HasGeometry` (`GtpVisibility::HierarchyNode *node`)
- `bool HasGreaterDistance` (`GtpVisibility::HierarchyNode *node1`, `GtpVisibility::HierarchyNode *node2`)
- `void SetNodeVisible` (`GtpVisibility::HierarchyNode *node`, `const bool visible`)
- `void SetLastVisited` (`GtpVisibility::HierarchyNode *node`, `const int frameId`)
- `bool IsNodeVisible` (`GtpVisibility::HierarchyNode *node`)
- `int LastVisited` (`GtpVisibility::HierarchyNode *node`)

Protected Methods

- `AxisAlignedBox * GetBoundingBox` (`GtpVisibility::HierarchyNode *node`)
- `Real GetSquaredViewDepth` (`const Camera *cam`, `const AxisAlignedBox *box`) `const`

Protected Attributes

- `unsigned int mNumOctreeNode`

7.2.1 Detailed Description

This class implements the hierarchy interface for the [Ogre](#) bsp hierarchy.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 `Ogre::BspHierarchyInterface::BspHierarchyInterface (SceneManager * sm,
RenderSystem * rsys)`

7.2.3 Member Function Documentation

7.2.3.1 `void Ogre::BspHierarchyInterface::PullUpVisibility (GtpVisibility::HierarchyNode *
node) [virtual]`

Gets the given option for the scene traverser.

Remarks:

See `setOption`

Implements [GtpVisibility::HierarchyInterface](#).

7.2.3.2 `void Ogre::BspHierarchyInterface::TraverseNode (GtpVisibility::HierarchyNode * node)
[virtual]`

Traverses given node.

Parameters:

node current node

Remarks:

pushes children on distance queue

Implements [GtpVisibility::HierarchyInterface](#).

7.2.3.3 `void Ogre::BspHierarchyInterface::RenderNode (GtpVisibility::HierarchyNode * node)
[virtual]`

Renders current scene node .

Parameters:

node current scene node to be rendered

Implements [GtpVisibility::HierarchyInterface](#).

7.2.3.4 `bool Ogre::BspHierarchyInterface::IsLeaf (GtpVisibility::HierarchyNode * node)
[virtual]`

Returns true if current node is leaf of the hierarchy.

Parameters:

node hierarchy node

Returns:

true if node is leaf

Implements [GtpVisibility::HierarchyInterface](#).

7.2.3.5 bool Ogre::BspHierarchyInterface::HasGeometry (GtpVisibility::HierarchyNode * node) [virtual]

Returns true if there is renderable geometry attached to this node

Parameters:

node the current node

Returns:

if the node has renderable geometry

Implements [GtpVisibility::HierarchyInterface](#).

7.2.3.6 bool Ogre::BspHierarchyInterface::HasGreaterDistance (GtpVisibility::HierarchyNode * node1, GtpVisibility::HierarchyNode * node2) [virtual]

Returns true if node 1 has greater distance to the view plane than node 2.

Parameters:

node1 the first node to be compared

node2 the second node to be compared

Implements [GtpVisibility::HierarchyInterface](#).

7.2.3.7 void Ogre::BspHierarchyInterface::SetNodeVisible (GtpVisibility::HierarchyNode * node, const bool visible) [virtual]

Sets the visible flag for this node.

Parameters:

node the current node

visible the visible flag

Implements [GtpVisibility::HierarchyInterface](#).

7.2.3.8 void Ogre::BspHierarchyInterface::SetLastVisited (GtpVisibility::HierarchyNode * node, const int frameId) [virtual]

Sets the last visited frame id for this node.

Parameters:

node the current node

frameId the current frame id

Implements [GtpVisibility::HierarchyInterface](#).

7.2.3.9 `bool Ogre::BspHierarchyInterface::IsNodeVisible (GtpVisibility::HierarchyNode * node)`
[virtual]

Returns true if node has the visible flag set. See set

Implements [GtpVisibility::HierarchyInterface](#).

7.2.3.10 `int Ogre::BspHierarchyInterface::LastVisited (GtpVisibility::HierarchyNode * node)`
[virtual]

Returns frame id when this node was last visited by the traverser. See set

Implements [GtpVisibility::HierarchyInterface](#).

7.2.3.11 `AxisAlignedBox * Ogre::BspHierarchyInterface::GetBoundingBox (GtpVisibility::HierarchyNode * node)` [protected, virtual]

Returns pointer to the bounding box of the node.

Parameters:

node current hierarchy node

Returns:

bounding box of current node

Implements [Ogre::PlatformHierarchyInterface](#).

7.2.3.12 `Real Ogre::BspHierarchyInterface::GetSquaredViewDepth (const Camera * cam, const AxisAlignedBox * box) const` [protected]

Returns squared distance of center of box with respect to the camera .

Parameters:

cam current camera

box axis aligned box

7.2.4 Member Data Documentation

7.2.4.1 `unsigned int Ogre::BspHierarchyInterface::mNumOctreeNode` [protected]

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

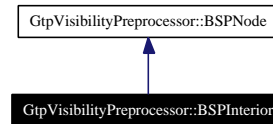
- [Ogre/include/OgreBspHierarchyInterface.h](#)
- [Ogre/src/OgreBspHierarchyInterface.cpp](#)

7.3 GtpVisibilityPreprocessor::BSPInterior Class Reference

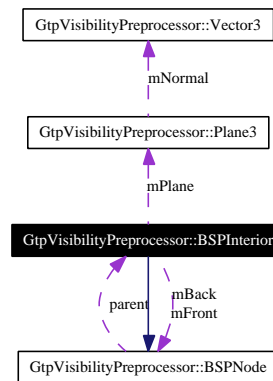
```
#include <ViewCellBSP.h>
```

Inherits [GtpVisibilityPreprocessor::BSPNode](#).

Inheritance diagram for GtpVisibilityPreprocessor::BSPInterior:



Collaboration diagram for GtpVisibilityPreprocessor::BSPInterior:



Protected Attributes

- [Plane3 mPlane](#)
Splitting plane corresponding to this node.
- [BSPNode * mBack](#)
back node
- [BSPNode * mFront](#)
front node

Private Methods

- virtual bool [IsLeaf](#) () const

7.3.1 Detailed Description

BSP interior node implementation

7.3.2 Member Function Documentation

7.3.2.1 `virtual bool GtpVisibilityPreprocessor::BSPInterior::IsLeaf () const` [`inline`, `private`, `virtual`]

Returns:

false since it is an interior node

Implements [GtpVisibilityPreprocessor::BSPNode](#).

7.3.3 Member Data Documentation

7.3.3.1 `Plane3 GtpVisibilityPreprocessor::BSPInterior::mPlane` [`protected`]

Splitting plane corresponding to this node.

7.3.3.2 `BSPNode* GtpVisibilityPreprocessor::BSPInterior::mBack` [`protected`]

back node

7.3.3.3 `BSPNode* GtpVisibilityPreprocessor::BSPInterior::mFront` [`protected`]

front node

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

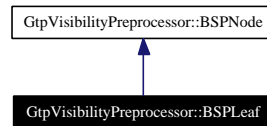
- [GtpVisibilityPreprocessor/include/ViewCellBSP.h](#)

7.4 GtpVisibilityPreprocessor::BSPLeaf Class Reference

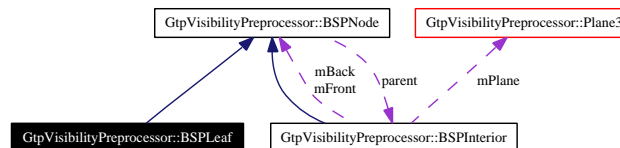
```
#include <ViewCellBSP.h>
```

Inherits [GtpVisibilityPreprocessor::BSPNode](#).

Inheritance diagram for GtpVisibilityPreprocessor::BSPLeaf:



Collaboration diagram for GtpVisibilityPreprocessor::BSPLeaf:



Public Methods

- [BSPLeaf](#) (ViewCell *viewCell=NULL)
- virtual bool [IsLeaf](#) () const

Protected Attributes

- ViewCell * [mViewCell](#)

7.4.1 Detailed Description

BSP leaf node implementation

7.4.2 Constructor & Destructor Documentation

7.4.2.1 [GtpVisibilityPreprocessor::BSPLeaf::BSPLeaf](#) (ViewCell * *viewCell* = NULL) [inline]

7.4.3 Member Function Documentation

7.4.3.1 virtual bool [GtpVisibilityPreprocessor::BSPLeaf::IsLeaf](#) () const [inline, virtual]

Returns:

true since it is an interior node

Implements [GtpVisibilityPreprocessor::BSPNode](#).

7.4.4 Member Data Documentation

7.4.4.1 ViewCell* GtpVisibilityPreprocessor::BSPLeaf::mViewCell [protected]

polygonal representation of this viewcell if NULL this note does not correspond to feasible viewcell

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

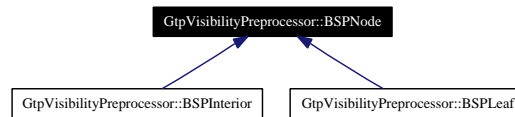
- GtpVisibilityPreprocessor/include/[ViewCellBSP.h](#)

7.5 GtpVisibilityPreprocessor::BSPNode Class Reference

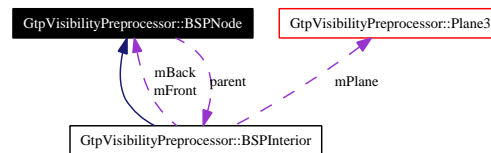
```
#include <ViewCellBSP.h>
```

Inherited by [GtpVisibilityPreprocessor::BSPInterior](#), and [GtpVisibilityPreprocessor::BSPLeaf](#).

Inheritance diagram for GtpVisibilityPreprocessor::BSPNode:



Collaboration diagram for GtpVisibilityPreprocessor::BSPNode:



Public Methods

- virtual bool [IsLeaf](#) () const=0
- virtual bool [IsRoot](#) () const

Protected Attributes

- [BSPInterior](#) * [parent](#)
parent of this node

7.5.1 Detailed Description

[BSPNode](#) abstract class serving for interior and leaf node implementation

7.5.2 Member Function Documentation

7.5.2.1 virtual bool GtpVisibilityPreprocessor::BSPNode::IsLeaf () [pure virtual]

Determines whether this node is a leaf or not

Returns:

true if leaf

Implemented in [GtpVisibilityPreprocessor::BSPInterior](#), and [GtpVisibilityPreprocessor::BSPLeaf](#).

7.5.2.2 `virtual bool GtpVisibilityPreprocessor::BSPNode::IsRoot () const` [inline, virtual]

Determines whether this node is a root

Returns:

true if root

7.5.3 Member Data Documentation

7.5.3.1 `BSPInterior*` `GtpVisibilityPreprocessor::BSPNode::parent` [protected]

parent of this node

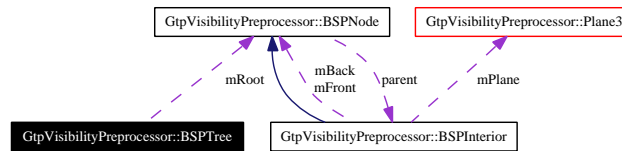
The documentation for this class was generated from the following file:

- [GtpVisibilityPreprocessor/include/ViewCellBSP.h](#)

7.6 GtpVisibilityPreprocessor::BSPTree Class Reference

```
#include <ViewCellBSP.h>
```

Collaboration diagram for GtpVisibilityPreprocessor::BSPTree:



Public Methods

- [BSPTree](#) (ViewCell *cell)

Protected Attributes

- [BSPNode](#) * [mRoot](#)
Pointer to the root of the tree.
- ViewCell * [mRootCell](#)
*Pointer to the root cell of the viewspace */.*

7.6.1 Detailed Description

Implementation of the ViewCell BSP tree

7.6.2 Constructor & Destructor Documentation

7.6.2.1 GtpVisibilityPreprocessor::BSPTree::BSPTree (ViewCell * cell) [inline]

Constructor takes a pointer to the cell corresponding to the whole viewspace

7.6.3 Member Data Documentation

7.6.3.1 [BSPNode](#)* GtpVisibilityPreprocessor::BSPTree::mRoot [protected]

Pointer to the root of the tree.

7.6.3.2 ViewCell* GtpVisibilityPreprocessor::BSPTree::mRootCell [protected]

Pointer to the root cell of the viewspace */.

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

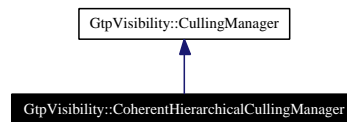
- GtpVisibilityPreprocessor/include/[ViewCellBSP.h](#)

7.7 GtpVisibility::CoherentHierarchicalCullingManager Class Reference

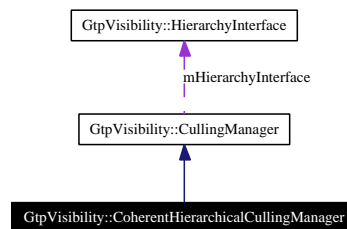
```
#include <CoherentHierarchicalCullingManager.h>
```

Inherits [GtpVisibility::CullingManager](#).

Inheritance diagram for GtpVisibility::CoherentHierarchicalCullingManager:



Collaboration diagram for GtpVisibility::CoherentHierarchicalCullingManager:



Public Methods

- [CoherentHierarchicalCullingManager](#) ([HierarchyInterface](#) *hierarchyInterface)
- void [RenderScene](#) ()

7.7.1 Detailed Description

Renders the scene with the coherent hierarchical culling algorithm.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 [GtpVisibility::CoherentHierarchicalCullingManager::CoherentHierarchicalCullingManager](#) ([HierarchyInterface](#) * *hierarchyInterface*)

7.7.3 Member Function Documentation

7.7.3.1 void [GtpVisibility::CoherentHierarchicalCullingManager::RenderScene](#) () [virtual]

Renders the scene using a specific occlusion culling algorithm, e.g., coherent hierarchical culling or stop and wait.

Implements [GtpVisibility::CullingManager](#).

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

- GtpVisibility/include/[CoherentHierarchicalCullingManager.h](#)

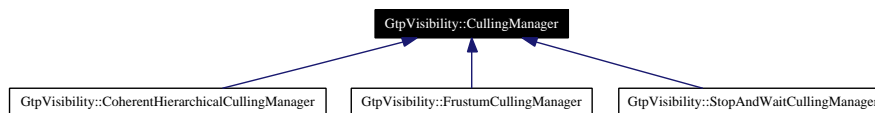
- [GtpVisibility/src/CoherentHierarchicalCullingManager.cpp](#)

7.8 GtpVisibility::CullingManager Class Reference

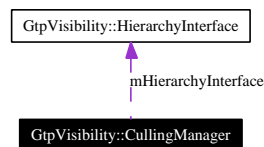
```
#include <CullingManager.h>
```

Inherited by [GtpVisibility::CoherentHierarchicalCullingManager](#), [GtpVisibility::FrustumCullingManager](#), and [GtpVisibility::StopAndWaitCullingManager](#).

Inheritance diagram for GtpVisibility::CullingManager:



Collaboration diagram for GtpVisibility::CullingManager:



Public Methods

- [CullingManager](#) ([HierarchyInterface](#) *hierarchyInterface)
- virtual void [RenderScene](#) ()=0
- void [SetHierarchyInterface](#) ([HierarchyInterface](#) *hierarchyInterface)

Protected Attributes

- unsigned int [mNumQueryCulledNodes](#)
- unsigned int [mNumFrustumCulledNodes](#)
- unsigned int [mVisibilityThreshold](#)
- [HierarchyInterface](#) * [mHierarchyInterface](#)

7.8.1 Detailed Description

This abstract class implements an interface for a specific culling algorithm. The algorithm is either used to render a scene or to make a visibility query.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 GtpVisibility::CullingManager::CullingManager ([HierarchyInterface](#) * hierarchyInterface)

Constructor taking a scene traverser for a specific type of hierarchy as argument.

7.8.3 Member Function Documentation

7.8.3.1 `virtual void GtpVisibility::CullingManager::RenderScene ()` [pure virtual]

Renders the scene using a specific occlusion culling algorithm, e.g., coherent hierarchical culling or stop and wait.

Implemented in [GtpVisibility::CoherentHierarchicalCullingManager](#), [GtpVisibility::FrustumCullingManager](#), and [GtpVisibility::StopAndWaitCullingManager](#).

7.8.3.2 `void GtpVisibility::CullingManager::SetHierarchyInterface (HierarchyInterface * hierarchyInterface)`

Sets the hierarchy interface.

Parameters:

hierarchyInterface

Remarks:

the hierarchy interface encapsulates the hierarchy we are working on

7.8.4 Member Data Documentation

7.8.4.1 `unsigned int GtpVisibility::CullingManager::mNumQueryCulledNodes` [protected]

7.8.4.2 `unsigned int GtpVisibility::CullingManager::mNumFrustumCulledNodes` [protected]

7.8.4.3 `unsigned int GtpVisibility::CullingManager::mVisibilityThreshold` [protected]

7.8.4.4 `HierarchyInterface* GtpVisibility::CullingManager::mHierarchyInterface` [protected]

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

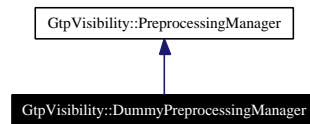
- [GtpVisibility/include/CullingManager.h](#)
- [GtpVisibility/src/CullingManager.cpp](#)

7.9 GtpVisibility::DummyPreprocessingManager Class Reference

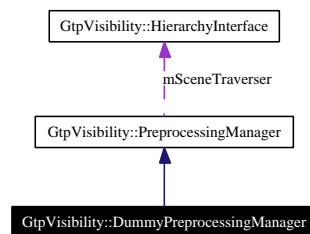
```
#include <DummyPreprocessingManager.h>
```

Inherits [GtpVisibility::PreprocessingManager](#).

Inheritance diagram for GtpVisibility::DummyPreprocessingManager:



Collaboration diagram for GtpVisibility::DummyPreprocessingManager:



Public Methods

- [DummyPreprocessingManager](#) ([HierarchyInterface](#) *hierarchyInterface)
- virtual bool [ExportScene](#) (const string filename)
- virtual bool [LoadPreprocessedData](#) (const string filename)

Protected Methods

- virtual bool [LocateViewCellIds](#) (const [Vector3](#) ¢er, const float radius, vector< int > *viewCellIds)
- virtual int [AddViewCellPVS](#) (const int cellID, InfoContainer< [NodeInfo](#) > *visibleNodes, InfoContainer< [MeshInfo](#) > *visibleMeshes)

7.9.1 Detailed Description

Implements a dummy interface to the external visibility preprocessing.

7.9.2 Constructor & Destructor Documentation

7.9.2.1 GtpVisibility::DummyPreprocessingManager::DummyPreprocessingManager ([HierarchyInterface](#) * hierarchyInterface)

Constructor taking a scene traverser for a specific type of hierarchyInterface as argument.

7.9.3 Member Function Documentation

7.9.3.1 `bool GtpVisibility::DummyPreprocessingManager::ExportScene (const string filename) [virtual]`

Export the scene for visibility preprocessing. Exports the hierarchyInterface including its bounding boxes + mesh data for occluders. The viewcell data will either be supplied directly to the visibility preprocessing standalone module

Implements [GtpVisibility::PreprocessingManager](#).

7.9.3.2 `bool GtpVisibility::DummyPreprocessingManager::LoadPreprocessedData (const string filename) [virtual]`

Load preprocessed visibility information

Implements [GtpVisibility::PreprocessingManager](#).

7.9.3.3 `bool GtpVisibility::DummyPreprocessingManager::LocateViewCellIds (const Vector3 & center, const float radius, vector< int > * viewCellIds) [protected, virtual]`

Find viewcells intersecting a spherical neighborhood of a point. Returns false if no viewcell was found.

Implements [GtpVisibility::PreprocessingManager](#).

7.9.3.4 `int GtpVisibility::DummyPreprocessingManager::AddViewCellPVS (const int cellID, InfoContainer< NodeInfo > * visibleNodes, InfoContainer< MeshInfo > * visibleMeshes) [protected, virtual]`

Add a PVS of the given viewcell to the already evaluated PVS by computing a union with visibleNodes and visibleMeshes. Returns the number of added entries.

Implements [GtpVisibility::PreprocessingManager](#).

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

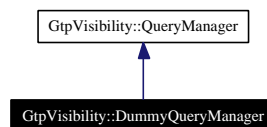
- [GtpVisibility/include/DummyPreprocessingManager.h](#)
- [GtpVisibility/src/DummyPreprocessingManager.cpp](#)

7.10 GtpVisibility::DummyQueryManager Class Reference

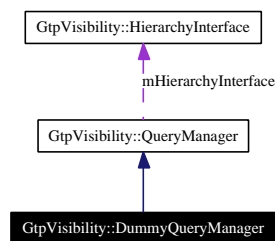
```
#include <DummyQueryManager.h>
```

Inherits [GtpVisibility::QueryManager](#).

Inheritance diagram for GtpVisibility::DummyQueryManager:



Collaboration diagram for GtpVisibility::DummyQueryManager:



Public Methods

- [DummyQueryManager](#) ([HierarchyInterface](#) *hierarchyInterface)
- virtual void [ComputeCameraVisibility](#) (const [Camera](#) &camera, [InfoContainer](#)< [NodeInfo](#) > *visibleNodes, [InfoContainer](#)< [MeshInfo](#) > *visibleGeometry, bool relativeVisibility=false)
- virtual void [ComputeFromPointVisibility](#) (const [Vector3](#) &point, [InfoContainer](#)< [NodeInfo](#) > *visibleNodes, [InfoContainer](#)< [MeshInfo](#) > *visibleGeometry, bool relativeVisibility=false)

7.10.1 Detailed Description

This implements dummy visibility queries. The queries return only the root of the hierarchy as visible node.

7.10.2 Constructor & Destructor Documentation

7.10.2.1 GtpVisibility::DummyQueryManager::DummyQueryManager ([HierarchyInterface](#) *hierarchyInterface) [inline]

Constructor taking a scene traverser for a specific type of hierarchyInterface as argument.

7.10.3 Member Function Documentation

7.10.3.1 `void GtpVisibility::DummyQueryManager::ComputeCameraVisibility (const Camera & camera, InfoContainer< NodeInfo > * visibleNodes, InfoContainer< MeshInfo > * visibleGeometry, bool relativeVisibility = false) [virtual]`

See also:

[QueryManager::ComputeCameraVisibility\(\)](#)

Implements [GtpVisibility::QueryManager](#).

7.10.3.2 `void GtpVisibility::DummyQueryManager::ComputeFromPointVisibility (const Vector3 & point, InfoContainer< NodeInfo > * visibleNodes, InfoContainer< MeshInfo > * visibleGeometry, bool relativeVisibility = false) [virtual]`

See also:

[QueryManager::ComputeFromPointVisibility\(\)](#)

Implements [GtpVisibility::QueryManager](#).

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

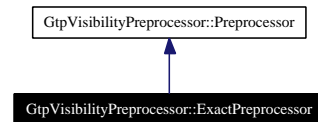
- [GtpVisibility/include/DummyQueryManager.h](#)
- [GtpVisibility/src/DummyQueryManager.cpp](#)

7.11 GtpVisibilityPreprocessor::ExactPreprocessor Class Reference

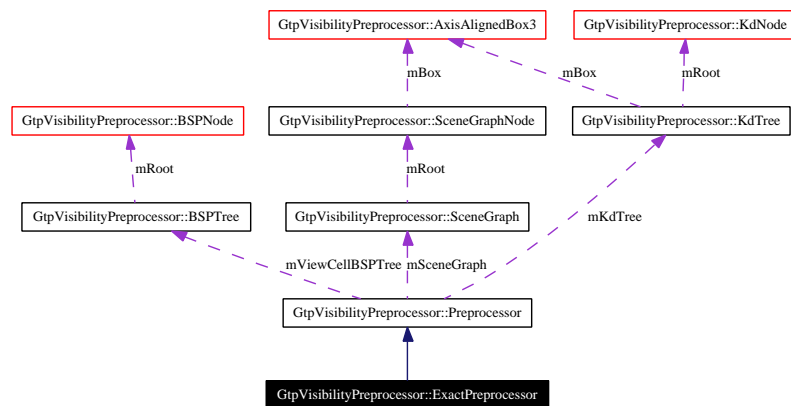
```
#include <ExactPreprocessor.h>
```

Inherits [GtpVisibilityPreprocessor::Preprocessor](#).

Inheritance diagram for GtpVisibilityPreprocessor::ExactPreprocessor:



Collaboration diagram for GtpVisibilityPreprocessor::ExactPreprocessor:



Public Methods

- virtual bool [ComputeVisibility](#) ()

7.11.1 Detailed Description

Exact visibility preprocessing. The implementation is based on 6D BSP and Plucker coordinates

7.11.2 Member Function Documentation

7.11.2.1 bool GtpVisibilityPreprocessor::ExactPreprocessor::ComputeVisibility () [virtual]

Compute visibility method. This method has to be reimplemented by the actual [Preprocessor](#) implementation (e.g. [SamplingPreprocessor](#), [ExactPreprocessor](#), [GlobalSamplingpreprocessor](#))

Implements [GtpVisibilityPreprocessor::Preprocessor](#).

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

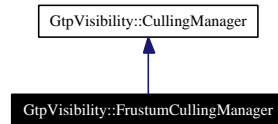
- GtpVisibilityPreprocessor/include/[ExactPreprocessor.h](#)
- GtpVisibilityPreprocessor/src/[ExactPreprocessor.cpp](#)

7.12 GtpVisibility::FrustumCullingManager Class Reference

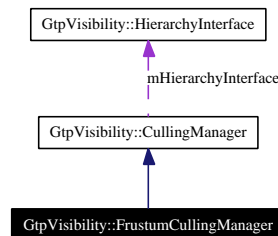
```
#include <FrustumCullingManager.h>
```

Inherits [GtpVisibility::CullingManager](#).

Inheritance diagram for GtpVisibility::FrustumCullingManager:



Collaboration diagram for GtpVisibility::FrustumCullingManager:



Public Methods

- [FrustumCullingManager](#) ([HierarchyInterface](#) *hierarchyInterface)
- void [RenderScene](#) ()

7.12.1 Detailed Description

Renders the scene, applies only view frustum culling.

7.12.2 Constructor & Destructor Documentation

- 7.12.2.1 [GtpVisibility::FrustumCullingManager::FrustumCullingManager](#) ([HierarchyInterface](#) **hierarchyInterface*)

7.12.3 Member Function Documentation

- 7.12.3.1 void [GtpVisibility::FrustumCullingManager::RenderScene](#) () [virtual]

Renders the scene using a specific occlusion culling algorithm, e.g., coherent hierarchical culling or stop and wait.

Implements [GtpVisibility::CullingManager](#).

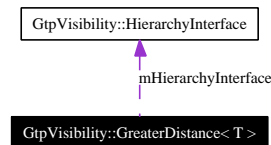
The documentation for this class was generated from the following files:

- GtpVisibility/include/[FrustumCullingManager.h](#)
- GtpVisibility/src/[FrustumCullingManager.cpp](#)

7.13 GtpVisibility::GreaterDistance< T > Class Template Reference

```
#include <DistanceQueue.h>
```

Collaboration diagram for GtpVisibility::GreaterDistance< T >:



Public Methods

- [GreaterDistance](#) ([HierarchyInterface](#) *hierarchyInterface)
- `bool operator()` (T v1, T v2) const

Private Attributes

- [HierarchyInterface](#) * [mHierarchyInterface](#)

7.13.1 Detailed Description

```
template<typename T> class GtpVisibility::GreaterDistance< T >
```

This class implements the less operator for the priority queue, i.e., a greater distance has a lower priority in the queue.

7.13.2 Constructor & Destructor Documentation

7.13.2.1 `template<typename T> GtpVisibility::GreaterDistance< T >::GreaterDistance`
([HierarchyInterface](#) * *hierarchyInterface*) [`inline`]

7.13.3 Member Function Documentation

7.13.3.1 `template<typename T> bool GtpVisibility::GreaterDistance< T >::operator()` (T v1, T v2) const [`inline`]

7.13.4 Member Data Documentation

7.13.4.1 `template<typename T> HierarchyInterface*` `GtpVisibility::GreaterDistance< T >::mHierarchyInterface` [`private`]

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

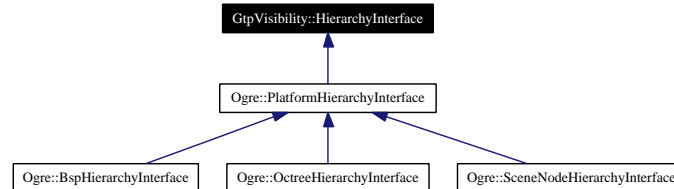
- GtpVisibility/include/[DistanceQueue.h](#)

7.14 GtpVisibility::HierarchyInterface Class Reference

#include <HierarchyInterface.h>

Inherited by [Ogre::PlatformHierarchyInterface](#).

Inheritance diagram for GtpVisibility::HierarchyInterface:



Public Methods

- [HierarchyInterface](#) ()
- virtual [~HierarchyInterface](#) ()
- virtual bool [IsLeaf](#) ([HierarchyNode](#) *node)=0
- virtual void [TraverseNode](#) ([HierarchyNode](#) *node)=0
- virtual void [RenderNode](#) ([HierarchyNode](#) *node)=0
- virtual void [PullUpVisibility](#) ([HierarchyNode](#) *node)=0
- virtual [OcclusionQuery](#) * [IssueOcclusionQuery](#) ([HierarchyNode](#) *node)=0
- void [SetSceneRoot](#) ([HierarchyNode](#) *root)
- [HierarchyNode](#) * [GetSceneRoot](#) () const
- void [InitFrame](#) ([HierarchyNode](#) *root)
- int [GetFrameId](#) ()
- [DistanceQueue](#) * [GetQueue](#) ()
- virtual bool [HasGreaterDistance](#) ([HierarchyNode](#) *node1, [HierarchyNode](#) *node2)=0
- virtual bool [CheckFrustumVisible](#) ([HierarchyNode](#) *node, bool &intersects)=0
- virtual [OcclusionQuery](#) * [GetNextOcclusionQuery](#) ()=0
- virtual bool [HasGeometry](#) ([HierarchyNode](#) *node)=0
- virtual void [SetNodeVisible](#) ([HierarchyNode](#) *node, const bool visible)=0
- virtual bool [IsNodeVisible](#) ([HierarchyNode](#) *node)=0
- virtual void [SetLastVisited](#) ([HierarchyNode](#) *node, const int frameId)=0
- virtual int [LastVisited](#) ([HierarchyNode](#) *node)=0

Protected Attributes

- unsigned int [mFrameId](#)
- int [mCurrentTestIdx](#)
- unsigned int [mNumSceneNodes](#)
- unsigned int [mNumTraversedNodes](#)
- unsigned int [mNumRenderedGeometry](#)
- unsigned int [mNumRenderedNodes](#)
- [DistanceQueue](#) * [mDistanceQueue](#)
- [HierarchyNode](#) * [mSceneRoot](#)

7.14.1 Detailed Description

Class which implements a hierarchy interface for a scene hierarchy.

7.14.2 Constructor & Destructor Documentation

7.14.2.1 GtpVisibility::HierarchyInterface::HierarchyInterface ()

Default constructor.

7.14.2.2 GtpVisibility::HierarchyInterface::~~HierarchyInterface () [virtual]

7.14.3 Member Function Documentation

7.14.3.1 virtual bool GtpVisibility::HierarchyInterface::IsLeaf ([HierarchyNode](#) * *node*) [pure virtual]

Returns true if current node is leaf of the hierarchy.

Parameters:

node hierarchy node

Returns:

true if node is leaf

Implemented in [Ogre::BspHierarchyInterface](#), [Ogre::OctreeHierarchyInterface](#), and [Ogre::SceneNodeHierarchyInterface](#).

7.14.3.2 virtual void GtpVisibility::HierarchyInterface::TraverseNode ([HierarchyNode](#) * *node*) [pure virtual]

Traverses the given node.

Parameters:

node the hierarchy node

Implemented in [Ogre::BspHierarchyInterface](#), [Ogre::OctreeHierarchyInterface](#), and [Ogre::SceneNodeHierarchyInterface](#).

7.14.3.3 virtual void GtpVisibility::HierarchyInterface::RenderNode ([HierarchyNode](#) * *node*) [pure virtual]

Renders current scene node .

Parameters:

node current scene node to be rendered

Implemented in [Ogre::BspHierarchyInterface](#), [Ogre::OctreeHierarchyInterface](#), and [Ogre::SceneNodeHierarchyInterface](#).

7.14.3.4 `virtual void GtpVisibility::HierarchyInterface::PullUpVisibility (HierarchyNode * node)`
`[pure virtual]`

Pulls up the visibility from the current node recursively to the parent nodes.

Parameters:

node the current node

Implemented in [Ogre::BspHierarchyInterface](#), [Ogre::OctreeHierarchyInterface](#), and [Ogre::SceneNode-HierarchyInterface](#).

7.14.3.5 `virtual OcclusionQuery* GtpVisibility::HierarchyInterface::IssueOcclusionQuery (HierarchyNode * node)`
`[pure virtual]`

Issue a occlusion query for this node.

Parameters:

node the current hierarchy node

Returns:

occlusion query for this node

Implemented in [Ogre::PlatformHierarchyInterface](#).

7.14.3.6 `void GtpVisibility::HierarchyInterface::SetSceneRoot (HierarchyNode * root)`

Sets the root of the scene hierarchy.

Parameters:

root the hierarchy root

7.14.3.7 `HierarchyNode* GtpVisibility::HierarchyInterface::GetSceneRoot () const` `[inline]`

Get the root of the scene hierarchy.

Returns:

the hierarchy root

7.14.3.8 `void GtpVisibility::HierarchyInterface::InitFrame (HierarchyNode * root)`

Sets the scene root and initialises this scene traverser for a traversal.

Parameters:

root current scene root

Remarks:

initialises some parameters, and also the statistics.

7.14.3.9 int GtpVisibility::HierarchyInterface::GetFrameId ()

Returns current frame id.

Returns:

frame id

7.14.3.10 DistanceQueue * GtpVisibility::HierarchyInterface::GetQueue ()

Returns the current distance queue.

Returns:

current distance queue

7.14.3.11 virtual bool GtpVisibility::HierarchyInterface::HasGreaterDistance (HierarchyNode * node1, HierarchyNode * node2) [pure virtual]

Returns true if node 1 has greater distance to the view plane than node 2.

Parameters:

node1 the first node to be compared

node2 the second node to be compared

Implemented in [Ogre::BspHierarchyInterface](#), [Ogre::OctreeHierarchyInterface](#), and [Ogre::SceneNodeHierarchyInterface](#).

7.14.3.12 virtual bool GtpVisibility::HierarchyInterface::CheckFrustumVisible (HierarchyNode * node, bool & intersects) [pure virtual]

Checks if the node is visible from the current view frustum.

Parameters:

node the current node

intersects returns true if the current node intersects the near plane

Implemented in [Ogre::PlatformHierarchyInterface](#).

7.14.3.13 virtual OcclusionQuery* GtpVisibility::HierarchyInterface::GetNextOcclusionQuery () [pure virtual]

Returns next available occlusion query or creates new one.

Returns:

the next occlusion query

Implemented in [Ogre::PlatformHierarchyInterface](#).

7.14.3.14 `virtual bool GtpVisibility::HierarchyInterface::HasGeometry (HierarchyNode * node)`
 [pure virtual]

Returns true if there is renderable geometry attached to this node

Parameters:

node the current node

Returns:

if the node has renderable geometry

Implemented in [Ogre::BspHierarchyInterface](#), [Ogre::OctreeHierarchyInterface](#), and [Ogre::SceneNodeHierarchyInterface](#).

7.14.3.15 `virtual void GtpVisibility::HierarchyInterface::SetNodeVisible (HierarchyNode * node,
 const bool visible)` [pure virtual]

Sets the visible flag for this node.

Parameters:

node the current node

visible the visible flag

Implemented in [Ogre::BspHierarchyInterface](#), [Ogre::OctreeHierarchyInterface](#), and [Ogre::SceneNodeHierarchyInterface](#).

7.14.3.16 `virtual bool GtpVisibility::HierarchyInterface::IsNodeVisible (HierarchyNode * node)`
 [pure virtual]

Returns true if node has the visible flag set. See set

Implemented in [Ogre::BspHierarchyInterface](#), [Ogre::OctreeHierarchyInterface](#), and [Ogre::SceneNodeHierarchyInterface](#).

7.14.3.17 `virtual void GtpVisibility::HierarchyInterface::SetLastVisited (HierarchyNode * node,
 const int frameId)` [pure virtual]

Sets the last visited frame id for this node.

Parameters:

node the current node

frameId the current frame id

Implemented in [Ogre::BspHierarchyInterface](#), [Ogre::OctreeHierarchyInterface](#), and [Ogre::SceneNodeHierarchyInterface](#).

7.14.3.18 `virtual int GtpVisibility::HierarchyInterface::LastVisited (HierarchyNode * node)`
 [pure virtual]

Returns frame id when this node was last visited by the traverser. See set

Implemented in [Ogre::BspHierarchyInterface](#), [Ogre::OctreeHierarchyInterface](#), and [Ogre::SceneNodeHierarchyInterface](#).

7.14.4 Member Data Documentation

7.14.4.1 **unsigned int GtpVisibility::HierarchyInterface::mFrameId** [protected]

7.14.4.2 **int GtpVisibility::HierarchyInterface::mCurrentTestIdx** [protected]

7.14.4.3 **unsigned int GtpVisibility::HierarchyInterface::mNumSceneNodes** [protected]

7.14.4.4 **unsigned int GtpVisibility::HierarchyInterface::mNumTraversedNodes** [protected]

7.14.4.5 **unsigned int GtpVisibility::HierarchyInterface::mNumRenderedGeometry**
[protected]

7.14.4.6 **unsigned int GtpVisibility::HierarchyInterface::mNumRenderedNodes** [protected]

7.14.4.7 **DistanceQueue*** **GtpVisibility::HierarchyInterface::mDistanceQueue** [protected]

7.14.4.8 **HierarchyNode*** **GtpVisibility::HierarchyInterface::mSceneRoot** [protected]

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

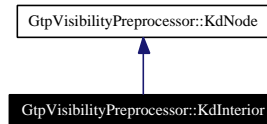
- GtpVisibility/include/[HierarchyInterface.h](#)
- GtpVisibility/src/[HierarchyInterface.cpp](#)

7.15 GtpVisibilityPreprocessor::KdInterior Class Reference

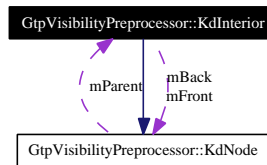
```
#include <KdTree.h>
```

Inherits [GtpVisibilityPreprocessor::KdNode](#).

Inheritance diagram for GtpVisibilityPreprocessor::KdInterior:



Collaboration diagram for GtpVisibilityPreprocessor::KdInterior:



Public Methods

- virtual bool [IsLeaf](#) () const

Protected Attributes

- int [axis](#)
- float [mPosition](#)
- [KdNode](#) * [mBack](#)
- [KdNode](#) * [mFront](#)

7.15.1 Detailed Description

Implementation of the kd-tree interior node

7.15.2 Member Function Documentation

7.15.2.1 virtual bool [GtpVisibilityPreprocessor::KdInterior::IsLeaf](#) () const [inline, virtual]

See also:

[KdNode::IsLeaf\(\)](#)

Implements [GtpVisibilityPreprocessor::KdNode](#).

7.15.3 Member Data Documentation

7.15.3.1 `int GtpVisibilityPreprocessor::KdInterior::axis` [protected]

splitting axis

7.15.3.2 `float GtpVisibilityPreprocessor::KdInterior::mPosition` [protected]

splitting position, absolute position within the bounding box of this node

7.15.3.3 `KdNode* GtpVisibilityPreprocessor::KdInterior::mBack` [protected]

back node

7.15.3.4 `KdNode* GtpVisibilityPreprocessor::KdInterior::mFront` [protected]

front node

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

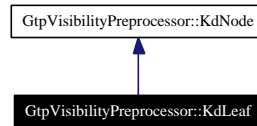
- `GtpVisibilityPreprocessor/include/KdTree.h`

7.16 GtpVisibilityPreprocessor::KdLeaf Class Reference

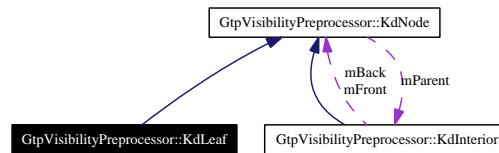
```
#include <KdTree.h>
```

Inherits [GtpVisibilityPreprocessor::KdNode](#).

Inheritance diagram for GtpVisibilityPreprocessor::KdLeaf:



Collaboration diagram for GtpVisibilityPreprocessor::KdLeaf:



Public Methods

- virtual bool [IsLeaf](#) () const

Protected Attributes

- [MeshContainer](#) mOccluders
- [MeshContainer](#) mOccludees
- [ViewCellContainer](#) mViewCells

7.16.1 Detailed Description

Implementation of the kd-tree leaf node

7.16.2 Member Function Documentation

7.16.2.1 virtual bool [GtpVisibilityPreprocessor::KdLeaf::IsLeaf](#) () const [inline, virtual]

See also:

[KdNode::IsLeaf\(\)](#)

Implements [GtpVisibilityPreprocessor::KdNode](#).

7.16.3 Member Data Documentation

7.16.3.1 [MeshContainer](#) [GtpVisibilityPreprocessor::KdLeaf::mOccluders](#) [protected]

pointers to occluders contained in this node

7.16.3.2 MeshContainer GtpVisibilityPreprocessor::KdLeaf::mOccludees [protected]

pointers to occludees contained in this node

7.16.3.3 ViewCellContainer GtpVisibilityPreprocessor::KdLeaf::mViewCells [protected]

pointers to viewcells contained in this node

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

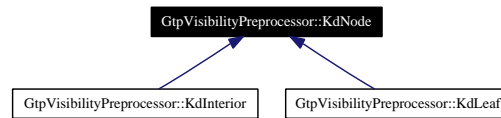
- GtpVisibilityPreprocessor/include/[KdTree.h](#)

7.17 GtpVisibilityPreprocessor::KdNode Class Reference

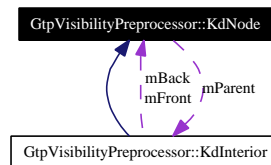
```
#include <KdTree.h>
```

Inherited by [GtpVisibilityPreprocessor::KdInterior](#), and [GtpVisibilityPreprocessor::KdLeaf](#).

Inheritance diagram for GtpVisibilityPreprocessor::KdNode:



Collaboration diagram for GtpVisibilityPreprocessor::KdNode:



Protected Attributes

- [KdInterior](#) * [mParent](#)

Private Methods

- virtual bool [IsLeaf](#) () const=0
- virtual bool [IsRoot](#) () const

7.17.1 Detailed Description

Abstract class for kd-tree node

7.17.2 Member Function Documentation

7.17.2.1 virtual bool GtpVisibilityPreprocessor::KdNode::IsLeaf () [private, pure virtual]

Determines whether this node is a leaf or interior node

Returns:

true if leaf

Implemented in [GtpVisibilityPreprocessor::KdInterior](#), and [GtpVisibilityPreprocessor::KdLeaf](#).

7.17.2.2 `virtual bool GtpVisibilityPreprocessor::KdNode::IsRoot () const` [inline, private, virtual]

Determines whether this node is the root of the tree

Returns:

true if root

7.17.3 Member Data Documentation

7.17.3.1 `KdInterior*` `GtpVisibilityPreprocessor::KdNode::mParent` [protected]

Parent of the node - the parent is a little overhead for maintenance of the tree, but allows various optimizations of tree traversal algorithms

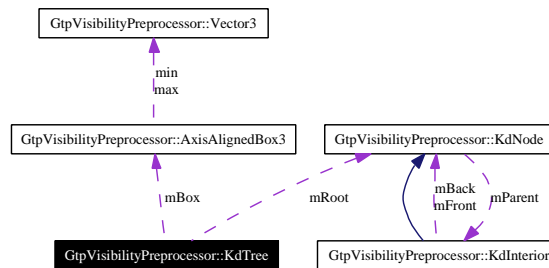
The documentation for this class was generated from the following file:

- `GtpVisibilityPreprocessor/include/KdTree.h`

7.18 GtpVisibilityPreprocessor::KdTree Class Reference

```
#include <KdTree.h>
```

Collaboration diagram for GtpVisibilityPreprocessor::KdTree:



Public Methods

- virtual void [InsertOccluder](#) ([Mesh](#) *occluder)
- virtual void [InsertOccludee](#) ([Mesh](#) *occludee)
- virtual void [InsertViewCell](#) ([ViewCell](#) *viewCell)
- virtual bool [Subdivide](#) ([KdNode](#) *subtree)
- [KdNode](#) * [GetRoot](#) () const

Protected Attributes

- [KdNode](#) * [mRoot](#)
root of the tree
- [AxisAlignedBox3](#) [mBox](#)
bounding box of the tree root

7.18.1 Detailed Description

[KdTree](#) for indexing scene entities - occluders/occludees/viewcells

7.18.2 Member Function Documentation

7.18.2.1 virtual void [GtpVisibilityPreprocessor::KdTree::InsertOccluder](#) ([Mesh](#) * *occluder*)
[inline, virtual]

Insert occluder into the tree

7.18.2.2 virtual void [GtpVisibilityPreprocessor::KdTree::InsertOccludee](#) ([Mesh](#) * *occludee*)
[inline, virtual]

Insert occludee into the tree

7.18.2.3 `virtual void GtpVisibilityPreprocessor::KdTree::InsertViewCell (ViewCell * viewCell)` [inline, virtual]

Insert view cell into the tree

7.18.2.4 `bool GtpVisibilityPreprocessor::KdTree::Subdivide (KdNode * subtree)` [virtual]

Check whether subdivision criteria are met for the given subtree. If not subdivide the leafs of the subtree. The criteria are specified in the environment as well as the subdivision method. By default surface area heuristics is used.

Parameters:

subtree root of the subtree

Returns:

true if subdivision was performed, false if subdivision criteria were already met

7.18.2.5 `KdNode* GtpVisibilityPreprocessor::KdTree::GetRoot () const` [inline]

Get the root of the tree

7.18.3 Member Data Documentation

7.18.3.1 `KdNode* GtpVisibilityPreprocessor::KdTree::mRoot` [protected]

root of the tree

7.18.3.2 `AxisAlignedBox3 GtpVisibilityPreprocessor::KdTree::mBox` [protected]

bounding box of the tree root

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

- [GtpVisibilityPreprocessor/include/KdTree.h](#)
- [GtpVisibilityPreprocessor/src/KdTree.cpp](#)

7.19 GtpVisibilityPreprocessor::Mesh Class Reference

```
#include <Mesh.h>
```

Public Types

- typedef std::vector< [Vector3](#) > [VertexContainer](#)
default vertex container for [Mesh](#)
- typedef std::vector< [Patch *](#) > [PatchContainer](#)
default patch container for [Mesh](#)

Public Methods

- [Mesh](#) ()
Default constructor.
- [Mesh](#) (const int vertices, const int patches)
Constructor with container preallocation.

Protected Attributes

- [VertexContainer](#) mVertices
- [PatchContainer](#) mPatches

7.19.1 Detailed Description

[Mesh](#) containing polygonal patches

7.19.2 Member Typedef Documentation

7.19.2.1 typedef std::vector<[Vector3](#)> GtpVisibilityPreprocessor::Mesh::VertexContainer

default vertex container for [Mesh](#)

7.19.2.2 typedef std::vector<[Patch *](#)> GtpVisibilityPreprocessor::Mesh::PatchContainer

default patch container for [Mesh](#)

7.19.3 Constructor & Destructor Documentation

7.19.3.1 GtpVisibilityPreprocessor::Mesh::Mesh () [inline]

Default constructor.

7.19.3.2 GtpVisibilityPreprocessor::Mesh::Mesh (const int *vertices*, const int *patches*) [inline]

Constructor with container preallocation.

7.19.4 Member Data Documentation

7.19.4.1 **VertexContainer** GtpVisibilityPreprocessor::Mesh::mVertices [protected]

Vertices forming the mesh

7.19.4.2 **PatchContainer** GtpVisibilityPreprocessor::Mesh::mPatches [protected]

Patches forming the mesh

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

- GtpVisibilityPreprocessor/include/[Mesh.h](#)

7.20 GtpVisibility::MeshInfo Class Reference

```
#include <VisibilityInfo.h>
```

Public Methods

- [MeshInfo](#) ([Mesh](#) *mesh, const float v)

Protected Attributes

- [Mesh](#) * mMesh
- float mVisibility

7.20.1 Detailed Description

Class storing the visibility information of a mesh.

7.20.2 Constructor & Destructor Documentation

7.20.2.1 [GtpVisibility::MeshInfo::MeshInfo](#) ([Mesh](#) * mesh, const float v) [inline]

7.20.3 Member Data Documentation

7.20.3.1 [Mesh](#)* [GtpVisibility::MeshInfo::mMesh](#) [protected]

pointer to the scene node

7.20.3.2 float [GtpVisibility::MeshInfo::mVisibility](#) [protected]

node visibility can either be a number of visible pixels or relative number of visible pixels (if the hardware queries will provide the total number of rasterized pixels)

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

- [GtpVisibility/include/VisibilityInfo.h](#)

7.21 GtpVisibility::NodeInfo Class Reference

```
#include <VisibilityInfo.h>
```

Public Methods

- [NodeInfo](#) ([HierarchyNode](#) *node, const float v)

Protected Attributes

- [HierarchyNode](#) * mNode
- float [mVisibility](#)

7.21.1 Detailed Description

Class storing the visibility information of a scene node.

7.21.2 Constructor & Destructor Documentation

7.21.2.1 [GtpVisibility::NodeInfo::NodeInfo](#) ([HierarchyNode](#) * node, const float v) [inline]

7.21.3 Member Data Documentation

7.21.3.1 [HierarchyNode](#)* [GtpVisibility::NodeInfo::mNode](#) [protected]

pointer to the scene node

7.21.3.2 float [GtpVisibility::NodeInfo::mVisibility](#) [protected]

node visibility can either be a number of visible pixels or relative number of visible pixels (if the hardware queries will provide the total number of rasterized pixels)

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

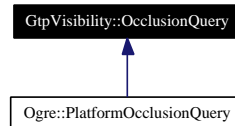
- [GtpVisibility/include/VisibilityInfo.h](#)

7.22 GtpVisibility::OcclusionQuery Class Reference

```
#include <OcclusionQuery.h>
```

Inherited by [Ogre::PlatformOcclusionQuery](#).

Inheritance diagram for GtpVisibility::OcclusionQuery:



Public Methods

- virtual `~OcclusionQuery ()`
- virtual unsigned int `GetQueryResult () const=0`
- virtual bool `ResultAvailable () const=0`
- virtual void `BeginQuery () const=0`
- virtual void `EndQuery () const=0`

7.22.1 Detailed Description

This class is an abstract interface for occlusion queries.

7.22.2 Constructor & Destructor Documentation

7.22.2.1 `virtual GtpVisibility::OcclusionQuery::~~OcclusionQuery () [inline, virtual]`

7.22.3 Member Function Documentation

7.22.3.1 `virtual unsigned int GtpVisibility::OcclusionQuery::GetQueryResult () [pure virtual]`

Returns the result of an occlusion query in terms of visible pixels.

Returns:

number of visible pixels

Implemented in [Ogre::PlatformOcclusionQuery](#).

7.22.3.2 `virtual bool GtpVisibility::OcclusionQuery::ResultAvailable () [pure virtual]`

Returns true if the result of the query is available, false otherwise.

Returns:

if result is available

Implemented in [Ogre::PlatformOcclusionQuery](#).

7.22.3.3 virtual void GtpVisibility::OcclusionQuery::BeginQuery () [pure virtual]

Begins occlusion query.

Remarks:

the query counts the number of visible pixels between it's begin and end

Implemented in [Ogre::PlatformOcclusionQuery](#).

7.22.3.4 virtual void GtpVisibility::OcclusionQuery::EndQuery () [pure virtual]

Ends occlusion query.

Implemented in [Ogre::PlatformOcclusionQuery](#).

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

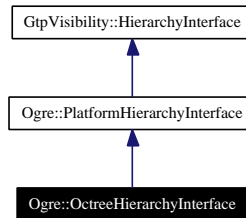
- [GtpVisibility/include/OcclusionQuery.h](#)

7.23 Ogre::OctreeHierarchyInterface Class Reference

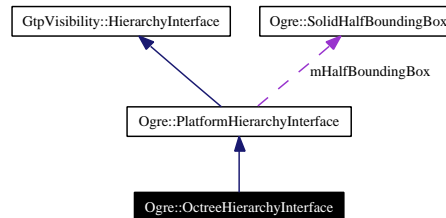
```
#include <OgreOctreeHierarchyInterface.h>
```

Inherits [Ogre::PlatformHierarchyInterface](#).

Inheritance diagram for `Ogre::OctreeHierarchyInterface`:



Collaboration diagram for `Ogre::OctreeHierarchyInterface`:



Public Methods

- [OctreeHierarchyInterface](#) (`SceneManager *sm`, `RenderSystem *rsys`)
- void [SetNumOctreeNodes](#) (`unsigned int num`)
- void [PullUpVisibility](#) (`GtpVisibility::HierarchyNode *node`)
- void [TraverseNode](#) (`GtpVisibility::HierarchyNode *node`)
- void [RenderNode](#) (`GtpVisibility::HierarchyNode *node`)
- bool [IsLeaf](#) (`GtpVisibility::HierarchyNode *node`)
- bool [HasGeometry](#) (`GtpVisibility::HierarchyNode *node`)
- bool [HasGreaterDistance](#) (`GtpVisibility::HierarchyNode *node1`, `GtpVisibility::HierarchyNode *node2`)
- void [SetNodeVisible](#) (`GtpVisibility::HierarchyNode *node`, `const bool visible`)
- void [SetLastVisited](#) (`GtpVisibility::HierarchyNode *node`, `const int frameId`)
- bool [IsNodeVisible](#) (`GtpVisibility::HierarchyNode *node`)
- int [LastVisited](#) (`GtpVisibility::HierarchyNode *node`)

Protected Methods

- `AxisAlignedBox *` [GetBoundingBox](#) (`GtpVisibility::HierarchyNode *node`)
- Real [GetSquaredViewDepth](#) (`const Camera *cam`, `const AxisAlignedBox *box`) `const`

Protected Attributes

- unsigned int [mNumOctreeNodes](#)

7.23.1 Detailed Description

This class implements the hierarchy interface for the [Ogre](#) octree hierarchy.

7.23.2 Constructor & Destructor Documentation

7.23.2.1 `Ogre::OctreeHierarchyInterface::OctreeHierarchyInterface (SceneManager * sm, RenderSystem * rsys)`

7.23.3 Member Function Documentation

7.23.3.1 `void Ogre::OctreeHierarchyInterface::SetNumOctreeNode (unsigned int num)`

Sets the number of nodes in this octree

Remarks:

do not confuse this with the OctreeNode class which is derived from SceneNode

Parameters:

num number of nodes in the octree

7.23.3.2 `void Ogre::OctreeHierarchyInterface::PullUpVisibility (GtpVisibility::HierarchyNode * node) [virtual]`

Gets the given option for the scene traverser.

Remarks:

See setOption

Implements [GtpVisibility::HierarchyInterface](#).

7.23.3.3 `void Ogre::OctreeHierarchyInterface::TraverseNode (GtpVisibility::HierarchyNode * node) [virtual]`

Traverses given node.

Parameters:

node current node

Remarks:

pushes children on distance queue

Implements [GtpVisibility::HierarchyInterface](#).

7.23.3.4 `void Ogre::OctreeHierarchyInterface::RenderNode (GtpVisibility::HierarchyNode * node) [virtual]`

Renders current scene node .

Parameters:

node current scene node to be rendered

Implements [GtpVisibility::HierarchyInterface](#).

7.23.3.5 `bool Ogre::OctreeHierarchyInterface::IsLeaf (GtpVisibility::HierarchyNode * node)`
[virtual]

Returns true if current node is leaf of the hierarchy.

Parameters:

node hierarchy node

Returns:

true if node is leaf

Implements [GtpVisibility::HierarchyInterface](#).

7.23.3.6 `bool Ogre::OctreeHierarchyInterface::HasGeometry (GtpVisibility::HierarchyNode * node)` [virtual]

Returns true if there is renderable geometry attached to this node

Parameters:

node the current node

Returns:

if the node has renderable geometry

Implements [GtpVisibility::HierarchyInterface](#).

7.23.3.7 `bool Ogre::OctreeHierarchyInterface::HasGreaterDistance (GtpVisibility::HierarchyNode * node1, GtpVisibility::HierarchyNode * node2)`
[virtual]

Returns true if node 1 has greater distance to the view plane than node 2.

Parameters:

node1 the first node to be compared

node2 the second node to be compared

Implements [GtpVisibility::HierarchyInterface](#).

7.23.3.8 `void Ogre::OctreeHierarchyInterface::SetNodeVisible (GtpVisibility::HierarchyNode * node, const bool visible)` [virtual]

Sets the visible flag for this node.

Parameters:

node the current node

visible the visible flag

Implements [GtpVisibility::HierarchyInterface](#).

7.23.3.9 `void Ogre::OctreeHierarchyInterface::SetLastVisited (GtpVisibility::HierarchyNode * node, const int frameId) [virtual]`

Sets the last visited frame id for this node.

Parameters:

node the current node

frameId the current frame id

Implements [GtpVisibility::HierarchyInterface](#).

7.23.3.10 `bool Ogre::OctreeHierarchyInterface::IsNodeVisible (GtpVisibility::HierarchyNode * node) [virtual]`

Returns true if node has the visible flag set. See set

Implements [GtpVisibility::HierarchyInterface](#).

7.23.3.11 `int Ogre::OctreeHierarchyInterface::LastVisited (GtpVisibility::HierarchyNode * node) [virtual]`

Returns frame id when this node was last visited by the traverser. See set

Implements [GtpVisibility::HierarchyInterface](#).

7.23.3.12 `AxisAlignedBox * Ogre::OctreeHierarchyInterface::GetBoundingBox (GtpVisibility::HierarchyNode * node) [protected, virtual]`

Returns pointer to the bounding box of the node.

Parameters:

node current hierarchy node

Returns:

bounding box of current node

Implements [Ogre::PlatformHierarchyInterface](#).

7.23.3.13 `Real Ogre::OctreeHierarchyInterface::GetSquaredViewDepth (const Camera * cam, const AxisAlignedBox * box) const [protected]`

Returns squared distance of center of box with respect to the camera .

Parameters:

cam current camera

box axis aligned box

7.23.4 Member Data Documentation

7.23.4.1 `unsigned int Ogre::OctreeHierarchyInterface::mNumOctreeNodes [protected]`

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

- [Ogre/include/OgreOctreeHierarchyInterface.h](#)
- [Ogre/src/OgreOctreeHierarchyInterface.cpp](#)

7.24 GtpVisibilityPreprocessor::Patch Class Reference

```
#include <Mesh.h>
```

Public Methods

- [Patch \(\)](#)

Protected Attributes

- `vector< Vector3 * > mVertices`
list of vertex pointers

7.24.1 Detailed Description

[Patch](#) used as an element of the mesh

7.24.2 Constructor & Destructor Documentation

7.24.2.1 `GtpVisibilityPreprocessor::Patch::Patch ()` [`inline`]

7.24.3 Member Data Documentation

7.24.3.1 `vector<Vector3 *> GtpVisibilityPreprocessor::Patch::mVertices` [`protected`]

list of vertex pointers

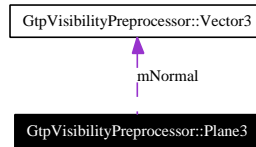
The documentation for this class was generated from the following file:

- [GtpVisibilityPreprocessor/include/Mesh.h](#)

7.25 GtpVisibilityPreprocessor::Plane3 Class Reference

```
#include <Plane3.h>
```

Collaboration diagram for GtpVisibilityPreprocessor::Plane3:



Public Attributes

- [Vector3 mNormal](#)
- [float mD](#)

7.25.1 Detailed Description

3D Plane

7.25.2 Member Data Documentation

7.25.2.1 [Vector3](#) GtpVisibilityPreprocessor::Plane3::mNormal

7.25.2.2 [float](#) GtpVisibilityPreprocessor::Plane3::mD

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

- [GtpVisibilityPreprocessor/include/Plane3.h](#)

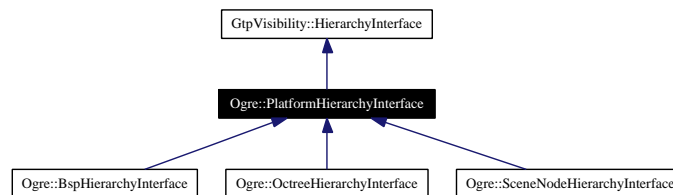
7.26 Ogre::PlatformHierarchyInterface Class Reference

```
#include <OgrePlatformHierarchyInterface.h>
```

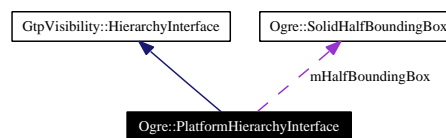
Inherits [GtpVisibility::HierarchyInterface](#).

Inherited by [Ogre::BspHierarchyInterface](#), [Ogre::OctreeHierarchyInterface](#), and [Ogre::SceneNodeHierarchyInterface](#).

Inheritance diagram for `Ogre::PlatformHierarchyInterface`:



Collaboration diagram for `Ogre::PlatformHierarchyInterface`:



Public Methods

- [PlatformHierarchyInterface](#) (`SceneManager *sm`, `RenderSystem *rsys`)
- `~PlatformHierarchyInterface` ()
- `GtpVisibility::OcclusionQuery * GetNextOcclusionQuery` ()
- `void SetCamera` (`Camera *cam`)
- `void InitFrame` (`GtpVisibility::HierarchyNode *root`, `Ogre::Camera *cam`)
- `bool CheckFrustumVisible` (`GtpVisibility::HierarchyNode *node`, `bool &intersects`)
- `void SetSceneManager` (`SceneManager *sm`)
- `void SetRenderSystem` (`RenderSystem *rsys`)
- `virtual AxisAlignedBox * GetBoundingBox` (`GtpVisibility::HierarchyNode *node`)=0
- `GtpVisibility::OcclusionQuery * IssueOcclusionQuery` (`GtpVisibility::HierarchyNode *node`)

Protected Methods

- `void DeleteQueries` ()
- `void RenderBoundingBox` (`AxisAlignedBox *box`)
- `SolidHalfBoundingBox * GetSolidHalfBoundingBox` (`int half`)

Protected Attributes

- `SolidHalfBoundingBox * mHalfBoundingBox` [2]
- `SceneManager * mSceneManager`
- `RenderSystem * mRenderSystem`

- Camera * [mCamera](#)
- AxisAlignedBox [mBox](#)
- std::vector< [PlatformOcclusionQuery](#) * > [mOcclusionQueries](#)

7.26.1 Detailed Description

Class which implements a hierarchy interface for a specific type of hierarchy.

Remarks:

also provides methods for using occlusion queries on the hierarchy nodes

7.26.2 Constructor & Destructor Documentation

7.26.2.1 [Ogre::PlatformHierarchyInterface::PlatformHierarchyInterface](#) ([SceneManager](#) * *sm*, [RenderSystem](#) * *rsys*)

Construction taking the current scene manager and the current rendersystem as argument

Parameters:

- sm* current scene manager
- rsys* current render system

7.26.2.2 [Ogre::PlatformHierarchyInterface::~~PlatformHierarchyInterface](#) ()

7.26.3 Member Function Documentation

7.26.3.1 [GtpVisibility::OcclusionQuery](#) * [Ogre::PlatformHierarchyInterface::GetNextOcclusionQuery](#) () [virtual]

Returns next available occlusion query or creates new one.

Returns:

the next occlusion query

Implements [GtpVisibility::HierarchyInterface](#).

7.26.3.2 void [Ogre::PlatformHierarchyInterface::SetCamera](#) ([Camera](#) * *cam*)

Sets the current camera.

Parameters:

- cam* the current camera

7.26.3.3 void [Ogre::PlatformHierarchyInterface::InitFrame](#) ([GtpVisibility::HierarchyNode](#) * *root*, [Ogre::Camera](#) * *cam*)

Initialises this scene traverser for the current frame.

Parameters:

root root of the hierarchy
cam current camera

Remarks:

convenience method wich calls `VisibilitySceneTraverser::initFrame`, sets the current camera, and initialises the distance queue.

7.26.3.4 bool Ogre::PlatformHierarchyInterface::CheckFrustumVisible (GtpVisibility::HierarchyNode * node, bool & intersects) [virtual]

Checks if the node is visible from the current view frustum.

Parameters:

node the current node
intersects returns true if the current node intersects the near plane

Implements [GtpVisibility::HierarchyInterface](#).

7.26.3.5 void Ogre::PlatformHierarchyInterface::SetSceneManager (SceneManager * sm)

Sets pointer to the current scene manager.

Parameters:

sm the scene manager

7.26.3.6 void Ogre::PlatformHierarchyInterface::SetRenderSystem (RenderSystem * rsys)

Sets pointer to the current render system

Parameters:

rsys the rendersystem

7.26.3.7 virtual AxisAlignedBox* Ogre::PlatformHierarchyInterface::GetBoundingBox (GtpVisibility::HierarchyNode * node) [pure virtual]

Returns pointer to bounding box of node.

Parameters:

node current hierarchy node

Returns:

bounding box of current node

Implemented in [Ogre::BspHierarchyInterface](#), [Ogre::OctreeHierarchyInterface](#), and [Ogre::SceneNodeHierarchyInterface](#).

7.26.3.8 [GtpVisibility::OcclusionQuery](#) * [Ogre::PlatformHierarchyInterface::IssueOcclusion-Query](#) ([GtpVisibility::HierarchyNode](#) * *node*) [virtual]

Issue a occlusion query for this node.

Parameters:

node the current hierarchy node

Returns:

occlusion query for this node

Implements [GtpVisibility::HierarchyInterface](#).

7.26.3.9 `void` [Ogre::PlatformHierarchyInterface::DeleteQueries](#) () [protected]

Deletes all occlusion queries.

7.26.3.10 `void` [Ogre::PlatformHierarchyInterface::RenderBoundingBox](#) ([AxisAlignedBox](#) * *box*) [protected]

Renders bounding box of specified node.

Parameters:

box the bounding box of the scene node to be rendered

7.26.3.11 [SolidHalfBoundingBox](#) * [Ogre::PlatformHierarchyInterface::GetSolidHalfBoundingBox](#) (int *half*) [protected]

Returns one half of the bounding box.

Parameters:

half the half index of the bounding box (0 or 1)

7.26.4 Member Data Documentation**7.26.4.1** [SolidHalfBoundingBox](#)* [Ogre::PlatformHierarchyInterface::mHalfBoundingBox](#)[2] [protected]

two halves of an aabb.

7.26.4.2 **SceneManager*** **Ogre::PlatformHierarchyInterface::mSceneManager** [protected]

7.26.4.3 **RenderSystem*** **Ogre::PlatformHierarchyInterface::mRenderSystem** [protected]

7.26.4.4 **Camera*** **Ogre::PlatformHierarchyInterface::mCamera** [protected]

7.26.4.5 **AxisAlignedBox** **Ogre::PlatformHierarchyInterface::mBox** [protected]

7.26.4.6 **std::vector<PlatformOcclusionQuery *>** **Ogre::PlatformHierarchyInterface::mOcclusionQueries** [protected]

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

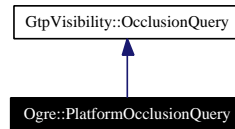
- [Ogre/include/OgrePlatformHierarchyInterface.h](#)
- [Ogre/src/OgrePlatformHierarchyInterface.cpp](#)

7.27 Ogre::PlatformOcclusionQuery Class Reference

```
#include <OgrePlatformOcclusionQuery.h>
```

Inherits [GtpVisibility::OcclusionQuery](#).

Inheritance diagram for `Ogre::PlatformOcclusionQuery`:



Collaboration diagram for `Ogre::PlatformOcclusionQuery`:



Public Methods

- [PlatformOcclusionQuery](#) (RenderSystem *rsys)
- virtual [~PlatformOcclusionQuery](#) ()
- virtual unsigned int [GetQueryResult](#) () const
- virtual bool [ResultAvailable](#) () const
- virtual void [BeginQuery](#) () const
- virtual void [EndQuery](#) () const

Protected Attributes

- HardwareOcclusionQuery * [mHardwareOcclusionQuery](#)

7.27.1 Detailed Description

This class is an implementation for occlusion queries using [Ogre](#).

Remarks:

the class encapsulates [Ogre](#) occlusion query calls.

7.27.2 Constructor & Destructor Documentation

7.27.2.1 `Ogre::PlatformOcclusionQuery::PlatformOcclusionQuery (RenderSystem * rsys)`

7.27.2.2 `Ogre::PlatformOcclusionQuery::~~PlatformOcclusionQuery ()` [virtual]

7.27.3 Member Function Documentation

7.27.3.1 `unsigned int Ogre::PlatformOcclusionQuery::GetQueryResult ()` [virtual]

Returns the result of an occlusion query in terms of visible pixels.

Returns:

number of visible pixels

Implements [GtpVisibility::OcclusionQuery](#).

7.27.3.2 `bool Ogre::PlatformOcclusionQuery::ResultAvailable ()` [virtual]

Returns true if the result of the query is available, false otherwise.

Returns:

if result is available

Implements [GtpVisibility::OcclusionQuery](#).

7.27.3.3 `void Ogre::PlatformOcclusionQuery::BeginQuery ()` [virtual]

Begins occlusion query.

Remarks:

the query counts the number of visible pixels between it's begin and end

Implements [GtpVisibility::OcclusionQuery](#).

7.27.3.4 `void Ogre::PlatformOcclusionQuery::EndQuery ()` [virtual]

Ends occlusion query.

Implements [GtpVisibility::OcclusionQuery](#).

7.27.4 Member Data Documentation

7.27.4.1 `HardwareOcclusionQuery* Ogre::PlatformOcclusionQuery::mHardwareOcclusionQuery` [protected]

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

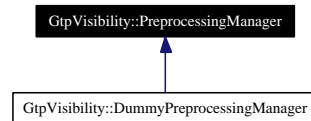
- [Ogre/include/OgrePlatformOcclusionQuery.h](#)
- [Ogre/src/OgrePlatformOcclusionQuery.cpp](#)

7.28 GtpVisibility::PreprocessingManager Class Reference

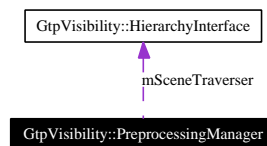
```
#include <PreprocessingManager.h>
```

Inherited by [GtpVisibility::DummyPreprocessingManager](#).

Inheritance diagram for GtpVisibility::PreprocessingManager:



Collaboration diagram for GtpVisibility::PreprocessingManager:



Public Methods

- [PreprocessingManager](#) ([HierarchyInterface](#) *hierarchyInterface)
- virtual [~PreprocessingManager](#) ()
- virtual bool [ExportScene](#) (const string filename)=0
- virtual bool [LoadPreprocessedData](#) (const string filename)=0
- virtual bool [GetPVS](#) (const [Vector3](#) &point, const float radius, InfoContainer< [NodeInfo](#) > *visibleNodes, InfoContainer< [MeshInfo](#) > *visibleMeshes)
- virtual void [SetSceneTraverser](#) ([HierarchyInterface](#) *hierarchyInterface)

Protected Methods

- long [GetOccluderChecksum](#) () const
- long [GetOccludeeChecksum](#) () const
- virtual bool [LocateViewCellIds](#) (const [Vector3](#) ¢er, const float radius, vector< int > *viewCellIds)=0
- virtual int [AddViewCellPVS](#) (const int cellID, InfoContainer< [NodeInfo](#) > *visibleNodes, InfoContainer< [MeshInfo](#) > *visibleMeshes)=0

Protected Attributes

- [HierarchyInterface](#) * [mSceneTraverser](#)

7.28.1 Detailed Description

This class defines an interface to the external visibility preprocessor. It allows to export the static part of the scene to a file in the XML format understood by the preprocessor. By default all the exported meshes

are considered as scene occluders, whereas occludees are formed by their bounding boxes, and bounding boxes of the scene hierarchy.

This class also allows to import the preprocessed data with PVS information for the viewcells (note that the viewcells are either generated automatically by the preprocessor or loaded from a polyhedral definition stored in a file (see documentation for the Preprocessor class).

7.28.2 Constructor & Destructor Documentation

7.28.2.1 GtpVisibility::PreprocessingManager::PreprocessingManager ([HierarchyInterface](#) * *hierarchyInterface*)

Constructor taking a [HierarchyInterface](#) as argument. The [HierarchyInterface](#) makes the [PreprocessingManager](#) independent from the actual scene representation as long as it supports a required set of methods.

7.28.2.2 virtual GtpVisibility::PreprocessingManager::~~PreprocessingManager () [virtual]

Destructor which deletes all data describing static scene visibility.

7.28.3 Member Function Documentation

7.28.3.1 virtual bool GtpVisibility::PreprocessingManager::ExportScene (const string *filename*) [pure virtual]

Export the scene for visibility preprocessing. Exports the hierarchyInterface including its bounding boxes + mesh data. Note that the bounding boxes can be used as occludees by the external preprocessor. The viewcell data will either be supplied directly to the visibility preprocessing standalone module.

Parameters:

filename name of the file to export.

Returns:

true if the export was succesful.

Implemented in [GtpVisibility::DummyPreprocessingManager](#).

7.28.3.2 virtual bool GtpVisibility::PreprocessingManager::LoadPreprocessedData (const string *filename*) [pure virtual]

Load preprocessed visibility information. The loaded data is matched with the current scene graph. The topology of the current scene graph has to match with the loaded data. There is also geometrical check of the bounding boxes. Once loading is succesful the [PreprocessingManager](#) establishes links from its internal visibility representation to the scene graph: no change of this part of the scene graph is allowed; this would violate the static scene assumption!

Parameters:

filename name of the file to load.

Returns:

true if the loading was succesful.

Implemented in [GtpVisibility::DummyPreprocessingManager](#).

7.28.3.3 `bool GtpVisibility::PreprocessingManager::GetPVS (const Vector3 & point, const float radius, InfoContainer< NodeInfo > * visibleNodes, InfoContainer< MeshInfo > * visibleMeshes)` [virtual]

Retrieve a PVS corresponding to the given spherical neighborhood of the point. Typically the implementation of this method will first locate all viewcells intersecting the sphere using efficient logarithmic search enhanced with caching the last query. Then it computes a union of the PVS for the found viewcells.

Parameters:

point the center point of the spherical neighborhood

radius the radius of the spherical neighborhood. Note that if radius==0 a more efficient implementation of the method for this special case can be used.

visibleNodes if not NULL a set of visible hierarchy nodes is added to the visibleNodes container. This set is formed of visible leaves or fully visible interior nodes.

visibleMeshes if not NULL the set of visible meshes is added to the container. Returns true if the corresponding PVS exists.

7.28.3.4 `virtual void GtpVisibility::PreprocessingManager::SetSceneTraverser (HierarchyInterface * hierarchyInterface)` [inline, virtual]

Sets the scene traverser.

Remarks:

the scene traverser is dependent on the type of hierarchyInterface the scene consists of.

7.28.3.5 `long GtpVisibility::PreprocessingManager::GetOccluderChecksum () const` [inline, protected]

Get checksum of the current occluder set in the scene graph. This method is used to check if the preprocessed data matches the current scene graph -> \$\$ could be moved to VisibilitySceneTraverser

7.28.3.6 `long GtpVisibility::PreprocessingManager::GetOccludeeChecksum () const` [inline, protected]

Get checksum of the current occludee set in the scene graph. This method is used to check if the preprocessed data matches the current scene graph -> \$\$ could be moved to VisibilitySceneTraverser

7.28.3.7 `virtual bool GtpVisibility::PreprocessingManager::LocateViewCellIds (const Vector3 & center, const float radius, vector< int > * viewCellIds)` [protected, pure virtual]

Find viewcells intersecting a spherical neighborhood of a point. Returns false if no viewcell was found.

Implemented in [GtpVisibility::DummyPreprocessingManager](#).

7.28.3.8 `virtual int GtpVisibility::PreprocessingManager::AddViewCellPVS (const int cellID, InfoContainer< NodeInfo > * visibleNodes, InfoContainer< MeshInfo > * visibleMeshes)`
[protected, pure virtual]

Add a PVS of the given viewcell to the already evaluated PVS by computing a union with visibleNodes and visibleMeshes. Returns the number of added entries.

Implemented in [GtpVisibility::DummyPreprocessingManager](#).

7.28.4 Member Data Documentation

7.28.4.1 [HierarchyInterface](#)* `GtpVisibility::PreprocessingManager::mSceneTraverser`
[protected]

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

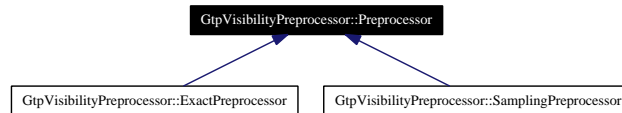
- [GtpVisibility/include/PreprocessingManager.h](#)
- [GtpVisibility/src/PreprocessingManager.cpp](#)

7.29 GtpVisibilityPreprocessor::Preprocessor Class Reference

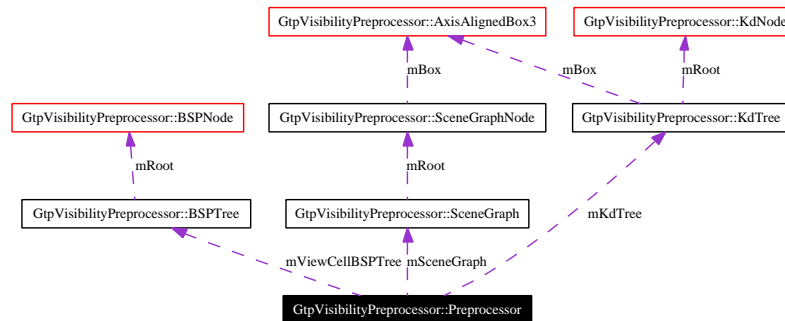
```
#include <Preprocessor.h>
```

Inherited by [GtpVisibilityPreprocessor::ExactPreprocessor](#), and [GtpVisibilityPreprocessor::SamplingPreprocessor](#).

Inheritance diagram for GtpVisibilityPreprocessor::Preprocessor:



Collaboration diagram for GtpVisibilityPreprocessor::Preprocessor:



Public Methods

- virtual bool [LoadScene](#) (const string filename)
- virtual bool [LoadViewcells](#) (const string filename)
- virtual bool [GenerateViewcells](#) ()
- virtual bool [ExportPreprocessedData](#) (const string filename)
- virtual bool [BuildKdTree](#) ()
- virtual bool [ComputeVisibility](#) ()=0

Protected Attributes

- [SceneGraph](#) * [mSceneGraph](#)
scene graph loaded from file
- [BSPTree](#) * [mViewCellBSPTree](#)
BSP tree representing the viewcells.
- [KdTree](#) * [mKdTree](#)
kD-tree organizing the scene graph (occluders + occludees) + viewcells
- [MeshContainer](#) [mOccluders](#)
list of all loaded occluders

- [MeshContainer mOccludees](#)
list of all loaded occludees
- [ViewCellContainer mViewcells](#)
list of all loaded/generated viewcells

7.29.1 Detailed Description

Main class of the visibility preprocessor. Responsible for loading and saving of the input and output files. Initiates construction of the kD-tree, viewcell loading/generation and the visibility computation itself.

7.29.2 Member Function Documentation

7.29.2.1 `bool GtpVisibilityPreprocessor::Preprocessor::LoadScene (const string filename)` [virtual]

Load the input scene.

Parameters:

filename file to load

Returns:

true on success

7.29.2.2 `bool GtpVisibilityPreprocessor::Preprocessor::LoadViewcells (const string filename)` [virtual]

Load the input viewcells. The input viewcells should be given as a collection of meshes. Each mesh is assumed to form a bounded polyhedron defining the interior of the viewcell. The method then builds a BSP tree of these view cells.

Parameters:

filename file to load

Returns:

true on success

7.29.2.3 `bool GtpVisibilityPreprocessor::Preprocessor::GenerateViewcells ()` [virtual]

Generate the viewCells automatically. The particular algorithm to be used depends on the environment setting. Initially the generated viewcells will cover the whole bounding volume of the scene. They can be pruned later depending on the results of visibility computations.

Returns:

true on successful viewcell generation.

7.29.2.4 `bool GtpVisibilityPreprocessor::Preprocessor::ExportPreprocessedData (const string filename) [virtual]`

Export all preprocessed data in a XML format understandable by the PreprocessingInterface of the [Gtp-VisibilityPreprocessor](#) Module. The file can be compressed depending on the environment settings.

Returns:

true on successful export

7.29.2.5 `bool GtpVisibilityPreprocessor::Preprocessor::BuildKdTree () [virtual]`

Build the [KdTree](#) of currently loaded occluders/occludees/viewcells. The construction is driven by the environment settings, which also says which of the three types of entities should be used to drive the heuristical construction (only occluders by default)

7.29.2.6 `virtual bool GtpVisibilityPreprocessor::Preprocessor::ComputeVisibility () [pure virtual]`

Compute visibility method. This method has to be reimplemented by the actual [Preprocessor](#) implementation (e.g. [SamplingPreprocessor](#), [ExactPreprocessor](#), [GlobalSamplingpreprocessor](#))

Implemented in [GtpVisibilityPreprocessor::ExactPreprocessor](#), and [GtpVisibilityPreprocessor::SamplingPreprocessor](#).

7.29.3 Member Data Documentation

7.29.3.1 `SceneGraph* GtpVisibilityPreprocessor::Preprocessor::mSceneGraph [protected]`

scene graph loaded from file

7.29.3.2 `BSPTree* GtpVisibilityPreprocessor::Preprocessor::mViewCellBSPTree [protected]`

BSP tree representing the viewcells.

7.29.3.3 `KdTree* GtpVisibilityPreprocessor::Preprocessor::mKdTree [protected]`

kD-tree organizing the scene graph (occluders + occludees) + viewcells

7.29.3.4 `MeshContainer GtpVisibilityPreprocessor::Preprocessor::mOccluders [protected]`

list of all loaded occluders

7.29.3.5 `MeshContainer GtpVisibilityPreprocessor::Preprocessor::mOccludees [protected]`

list of all loaded occludees

7.29.3.6 **ViewCellContainer** GtpVisibilityPreprocessor::Preprocessor::mViewcells [protected]

list of all loaded/generated viewcells

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

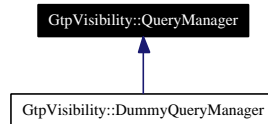
- GtpVisibilityPreprocessor/include/[Preprocessor.h](#)
- GtpVisibilityPreprocessor/src/[Preprocessor.cpp](#)

7.30 GtpVisibility::QueryManager Class Reference

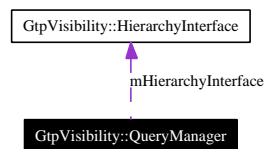
```
#include <QueryManager.h>
```

Inherited by [GtpVisibility::DummyQueryManager](#).

Inheritance diagram for GtpVisibility::QueryManager:



Collaboration diagram for GtpVisibility::QueryManager:



Public Methods

- [QueryManager](#) ([HierarchyInterface](#) *hierarchyInterface)
- virtual void [ComputeCameraVisibility](#) (const [Camera](#) &camera, [InfoContainer](#)< [NodeInfo](#) > *visibleNodes, [InfoContainer](#)< [MeshInfo](#) > *visibleGeometry, bool relativeVisibility=false)=0
- virtual void [ComputeFromPointVisibility](#) (const [Vector3](#) &point, [InfoContainer](#)< [NodeInfo](#) > *visibleNodes, [InfoContainer](#)< [MeshInfo](#) > *visibleGeometry, bool relativeVisibility=false)=0
- virtual bool [ShootRay](#) (const [Ray](#) &ray, [std::vector](#)< [Mesh](#) * > *visibleMeshes, bool isGlobalLine=false)
- void [SetSceneTraverser](#) ([HierarchyInterface](#) *hierarchyInterface)

Protected Attributes

- [HierarchyInterface](#) * mHierarchyInterface

7.30.1 Detailed Description

This abstract class defines interface for a specific visibility query algorithm. The interface supports two from point visibility queries and the ray shooting query. The output of the queries consists of list of visible meshes and hierarchy nodes. The from point queries will be implemented either with item bufferring or based purely on occlusion queries. Note that the actual implementation can also exploit the output of the visibility preprocessor.

7.30.2 Constructor & Destructor Documentation

7.30.2.1 GtpVisibility::QueryManager::QueryManager ([HierarchyInterface](#) * hierarchyInterface)

Constructor taking a hierarchy interface as an argument. This allows to operate onm different hierarchy types, while reusing the implementation of the query methods.

7.30.3 Member Function Documentation

7.30.3.1 `virtual void GtpVisibility::QueryManager::ComputeCameraVisibility (const Camera & camera, InfoContainer< NodeInfo > * visibleNodes, InfoContainer< MeshInfo > * visibleGeometry, bool relativeVisibility = false) [pure virtual]`

Computes restricted visibility from point by using an explicit camera to execute the visibility query.

Parameters:

camera The camera to be used

visibleNodes Pointer to the container where visible nodes should be added. This set is formed of visible leafs or fully visible interior nodes. If NULL no visible nodes are not evaluated.

visibleGeometry Pointer to the container where visible meshes should be added. If NULL no visible meshes are not evaluated.

relativeVisibility If true the visibility member for [NodeInfo](#) and [MeshInfo](#) represent relative visibility; i.e. the number of visible pixels divided by the the number of projected pixels.

Returns:

true if the corresponding PVS exists.

Implemented in [GtpVisibility::DummyQueryManager](#).

7.30.3.2 `virtual void GtpVisibility::QueryManager::ComputeFromPointVisibility (const Vector3 & point, InfoContainer< NodeInfo > * visibleNodes, InfoContainer< MeshInfo > * visibleGeometry, bool relativeVisibility = false) [pure virtual]`

Uses the specified point to execute the visibility query in all directions.

See also:

[ComputeCameraVisibility\(\)](#)

Implemented in [GtpVisibility::DummyQueryManager](#).

7.30.3.3 `bool GtpVisibility::QueryManager::ShootRay (const Ray & ray, std::vector< Mesh * > * visibleMeshes, bool isGlobalLine = false) [virtual]`

Ray shooting interface: finds an intersection with objects in the scene.

Parameters:

ray The given input ray (assuming the ray direction is normalized)

visibleMeshes List of meshes intersecting the ray

isGlobalLine If false only first intersection with opaque object is returned. Otherwise all intersections of the ray with the scene are found.

Returns:

true if there is any intersection.

7.30.3.4 void GtpVisibility::QueryManager::SetSceneTraverser ([HierarchyInterface](#) * *hierarchyInterface*)

Sets the scene traverser.

Remarks:

the scene traverser depends on the type of hierarchyInterface the scene consists of.

7.30.4 Member Data Documentation

7.30.4.1 [HierarchyInterface](#)* GtpVisibility::QueryManager::mHierarchyInterface [protected]

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

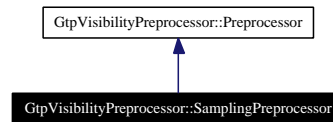
- [GtpVisibility/include/QueryManager.h](#)
- [GtpVisibility/src/QueryManager.cpp](#)

7.31 GtpVisibilityPreprocessor::SamplingPreprocessor Class Reference

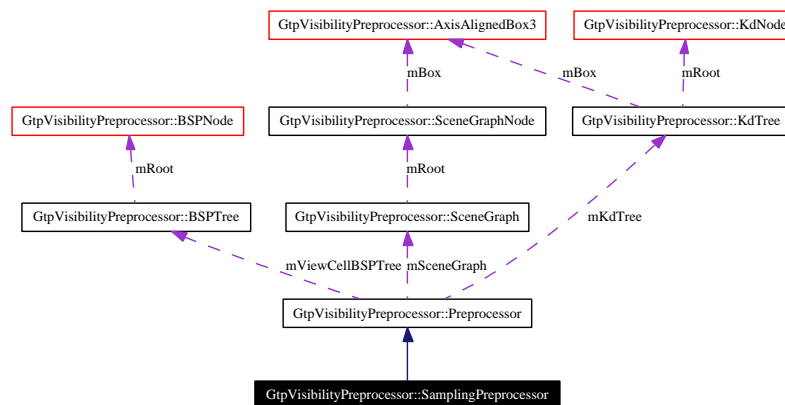
```
#include <SamplingPreprocessor.h>
```

Inherits [GtpVisibilityPreprocessor::Preprocessor](#).

Inheritance diagram for GtpVisibilityPreprocessor::SamplingPreprocessor:



Collaboration diagram for GtpVisibilityPreprocessor::SamplingPreprocessor:



Public Methods

- virtual bool [ComputeVisibility](#) ()

7.31.1 Detailed Description

Sampling based visibility preprocessing. The implementation is based on heuristical sampling of view space

7.31.2 Member Function Documentation

7.31.2.1 bool GtpVisibilityPreprocessor::SamplingPreprocessor::ComputeVisibility () [virtual]

Compute visibility method. This method has to be reimplemented by the actual [Preprocessor](#) implementation (e.g. [SamplingPreprocessor](#), [ExactPreprocessor](#), [GlobalSamplingpreprocessor](#))

Implements [GtpVisibilityPreprocessor::Preprocessor](#).

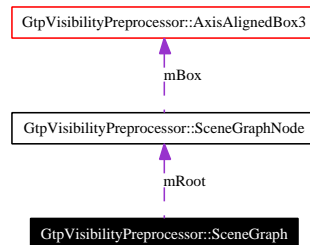
The documentation for this class was generated from the following files:

- [GtpVisibilityPreprocessor/include/SamplingPreprocessor.h](#)
- [GtpVisibilityPreprocessor/src/SamplingPreprocessor.cpp](#)

7.32 GtpVisibilityPreprocessor::SceneGraph Class Reference

```
#include <SceneGraph.h>
```

Collaboration diagram for GtpVisibilityPreprocessor::SceneGraph:



Protected Attributes

- [SceneGraphNode](#) * `mRoot`

7.32.1 Detailed Description

Scene graph class

7.32.2 Member Data Documentation

7.32.2.1 [SceneGraphNode](#)* `GtpVisibilityPreprocessor::SceneGraph::mRoot` [protected]

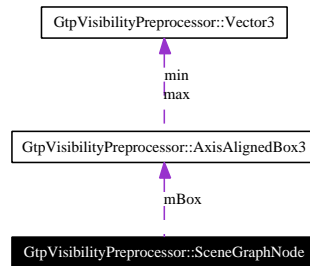
The documentation for this class was generated from the following file:

- `GtpVisibilityPreprocessor/include/SceneGraph.h`

7.33 GtpVisibilityPreprocessor::SceneGraphNode Class Reference

```
#include <SceneGraph.h>
```

Collaboration diagram for GtpVisibilityPreprocessor::SceneGraphNode:



Protected Attributes

- [MeshContainer](#) mGeometry
- [NodeContainer](#) mChildren
- [AxisAlignedBox3](#) mBox

7.33.1 Detailed Description

Basic scene graph node, we are interested only in bounding boxes and topology of the scene graph

7.33.2 Member Data Documentation

7.33.2.1 [MeshContainer](#) GtpVisibilityPreprocessor::SceneGraphNode::mGeometry
[protected]

7.33.2.2 [NodeContainer](#) GtpVisibilityPreprocessor::SceneGraphNode::mChildren
[protected]

7.33.2.3 [AxisAlignedBox3](#) GtpVisibilityPreprocessor::SceneGraphNode::mBox [protected]

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

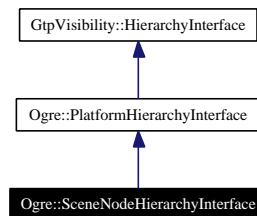
- GtpVisibilityPreprocessor/include/SceneGraph.h

7.34 Ogre::SceneNodeHierarchyInterface Class Reference

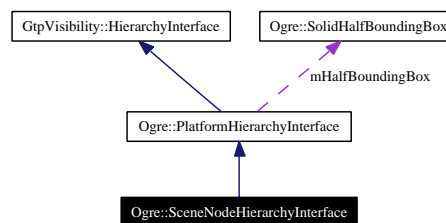
```
#include <OgreSceneNodeHierarchyInterface.h>
```

Inherits [Ogre::PlatformHierarchyInterface](#).

Inheritance diagram for `Ogre::SceneNodeHierarchyInterface`:



Collaboration diagram for `Ogre::SceneNodeHierarchyInterface`:



Public Methods

- [SceneNodeHierarchyInterface](#) (`SceneManager *sm`, `RenderSystem *rsys`)
- `bool IsLeaf (GtpVisibility::HierarchyNode *node)`
- `void TraverseNode (GtpVisibility::HierarchyNode *node)`
- `void RenderNode (GtpVisibility::HierarchyNode *node)`
- `void PullUpVisibility (GtpVisibility::HierarchyNode *node)`
- `bool HasGeometry (GtpVisibility::HierarchyNode *node)`
- `bool HasGreaterDistance (GtpVisibility::HierarchyNode *node1, GtpVisibility::HierarchyNode *node2)`
- `void SetNodeVisible (GtpVisibility::HierarchyNode *node, const bool visible)`
- `void SetLastVisited (GtpVisibility::HierarchyNode *node, const int frameId)`
- `bool IsNodeVisible (GtpVisibility::HierarchyNode *node)`
- `int LastVisited (GtpVisibility::HierarchyNode *node)`
- `AxisAlignedBox * GetBoundingBox (GtpVisibility::HierarchyNode *node)`

7.34.1 Detailed Description

This class implements the hierarchy interface for the [Ogre](#) scene node hierarchy.

7.34.2 Constructor & Destructor Documentation

7.34.2.1 `Ogre::SceneNodeHierarchyInterface::SceneNodeHierarchyInterface (SceneManager * sm, RenderSystem * rsys)`

Construction taking the current scene manager and the current rendersystem as argument

Parameters:

sm current scene manager

rsys current render system

7.34.3 Member Function Documentation

7.34.3.1 `bool Ogre::SceneNodeHierarchyInterface::IsLeaf (GtpVisibility::HierarchyNode * node) [virtual]`

Returns true if current node is leaf of the hierarchy.

Parameters:

node hierarchy node

Returns:

true if node is leaf

Implements [GtpVisibility::HierarchyInterface](#).

7.34.3.2 `void Ogre::SceneNodeHierarchyInterface::TraverseNode (GtpVisibility::HierarchyNode * node) [virtual]`

Traverses the given node.

Parameters:

node the hierarchy node

Implements [GtpVisibility::HierarchyInterface](#).

7.34.3.3 `void Ogre::SceneNodeHierarchyInterface::RenderNode (GtpVisibility::HierarchyNode * node) [virtual]`

Renders current scene node .

Parameters:

node current scene node to be rendered

Implements [GtpVisibility::HierarchyInterface](#).

7.34.3.4 `void Ogre::SceneNodeHierarchyInterface::PullUpVisibility (GtpVisibility::HierarchyNode * node) [virtual]`

Pulls up the visibility from the current node recursively to the parent nodes.

Parameters:

node the current node

Implements [GtpVisibility::HierarchyInterface](#).

7.34.3.5 bool Ogre::SceneNodeHierarchyInterface::HasGeometry (GtpVisibility::HierarchyNode * *node*) [virtual]

Returns true if there is renderable geometry attached to this node

Parameters:

node the current node

Returns:

if the node has renderable geometry

Implements [GtpVisibility::HierarchyInterface](#).

7.34.3.6 bool Ogre::SceneNodeHierarchyInterface::HasGreaterDistance (GtpVisibility::HierarchyNode * *node1*, GtpVisibility::HierarchyNode * *node2*) [virtual]

Returns true if node 1 has greater distance to the view plane than node 2.

Parameters:

node1 the first node to be compared

node2 the second node to be compared

Implements [GtpVisibility::HierarchyInterface](#).

7.34.3.7 void Ogre::SceneNodeHierarchyInterface::SetNodeVisible (GtpVisibility::HierarchyNode * *node*, const bool *visible*) [virtual]

Sets the visible flag for this node.

Parameters:

node the current node

visible the visible flag

Implements [GtpVisibility::HierarchyInterface](#).

7.34.3.8 void Ogre::SceneNodeHierarchyInterface::SetLastVisited (GtpVisibility::HierarchyNode * *node*, const int *frameId*) [virtual]

Sets the last visited frame id for this node.

Parameters:

node the current node

frameId the current frame id

Implements [GtpVisibility::HierarchyInterface](#).

7.34.3.9 `bool Ogre::SceneNodeHierarchyInterface::IsNodeVisible (GtpVisibility::HierarchyNode * node) [virtual]`

Returns true if node has the visible flag set. See `set`

Implements [GtpVisibility::HierarchyInterface](#).

7.34.3.10 `int Ogre::SceneNodeHierarchyInterface::LastVisited (GtpVisibility::HierarchyNode * node) [virtual]`

Returns frame id when this node was last visited by the traverser. See `set`

Implements [GtpVisibility::HierarchyInterface](#).

7.34.3.11 `AxisAlignedBox * Ogre::SceneNodeHierarchyInterface::GetBoundingBox (GtpVisibility::HierarchyNode * node) [virtual]`

Returns pointer to bounding box of node.

Parameters:

node current hierarchy node

Returns:

bounding box of current node

Implements [Ogre::PlatformHierarchyInterface](#).

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

- [Ogre/include/OgreSceneNodeHierarchyInterface.h](#)
- [Ogre/src/OgreSceneNodeHierarchyInterface.cpp](#)

7.35 Ogre::SolidHalfBoundingBox Class Reference

```
#include <OgreSolidHalfBoundingBox.h>
```

Public Methods

- [SolidHalfBoundingBox](#) (bool isFirstHalf)
- void [setupBoundingBox](#) (const AxisAlignedBox &aabb)
- void [getWorldTransforms](#) (Matrix4 *xform) const

Protected Methods

- void [setupBoundingBoxVertices](#) (const AxisAlignedBox &aab)
- void [setOcclusionQueryMaterial](#) ()

Protected Attributes

- bool [mIsFirstHalf](#)

7.35.1 Detailed Description

Allows the rendering of one half of a solid bounding box.

Remarks:

This class builds a wireframe renderable from a given aabb. A pointer to this class can be added to a render queue to display the bounding box of an object.

7.35.2 Constructor & Destructor Documentation

7.35.2.1 [Ogre::SolidHalfBoundingBox::SolidHalfBoundingBox](#) (bool *isFirstHalf*)

7.35.3 Member Function Documentation

7.35.3.1 void [Ogre::SolidHalfBoundingBox::setupBoundingBox](#) (const AxisAlignedBox & *aabb*)

7.35.3.2 void [Ogre::SolidHalfBoundingBox::getWorldTransforms](#) (Matrix4 * *xform*) const

Override this method to prevent parent transforms (rotation,translation,scale) and to make it public.

7.35.3.3 void [Ogre::SolidHalfBoundingBox::setupBoundingBoxVertices](#) (const AxisAlignedBox & *aab*) [protected]

Builds the wireframe line list.

Parameters:

aab the axis aligned bounding box for setting up the list.

7.35.3.4 void Ogre::SolidHalfBoundingBox::setOcclusionQueryMaterial () [protected]

Sets the material used for occlusion queries.

Remarks:

the material is called "OcclusionQuery" and uses no lighting, no depth write, and no colours

7.35.4 Member Data Documentation**7.35.4.1 bool Ogre::SolidHalfBoundingBox::mIsFirstHalf [protected]**

Whether this half box is the first or the second half of the bounding box.

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

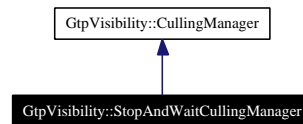
- [Ogre/include/OgreSolidHalfBoundingBox.h](#)
- [Ogre/src/OgreSolidHalfBoundingBox.cpp](#)

7.36 GtpVisibility::StopAndWaitCullingManager Class Reference

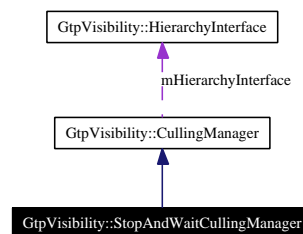
```
#include <StopAndWaitCullingManager.h>
```

Inherits [GtpVisibility::CullingManager](#).

Inheritance diagram for GtpVisibility::StopAndWaitCullingManager:



Collaboration diagram for GtpVisibility::StopAndWaitCullingManager:



Public Methods

- [StopAndWaitCullingManager](#) ([HierarchyInterface](#) *hierarchyInterface)
- void [RenderScene](#) ()

7.36.1 Detailed Description

Renders the scene with the hierarchical stop and wait algorithm.

7.36.2 Constructor & Destructor Documentation

7.36.2.1 GtpVisibility::StopAndWaitCullingManager::StopAndWaitCullingManager ([HierarchyInterface](#) * hierarchyInterface)

7.36.3 Member Function Documentation

7.36.3.1 void GtpVisibility::StopAndWaitCullingManager::RenderScene () [virtual]

Renders the scene using a specific occlusion culling algorithm, e.g., coherent hierarchical culling or stop and wait.

Implements [GtpVisibility::CullingManager](#).

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

- GtpVisibility/include/[StopAndWaitCullingManager.h](#)
- GtpVisibility/src/[StopAndWaitCullingManager.cpp](#)

7.37 GtpVisibilityPreprocessor::Vector3 Class Reference

```
#include <Vector3.h>
```

Public Attributes

- float [x](#)
- float [y](#)
- float [z](#)

7.37.1 Detailed Description

3D vector

7.37.2 Member Data Documentation

7.37.2.1 float [GtpVisibilityPreprocessor::Vector3::x](#)

7.37.2.2 float [GtpVisibilityPreprocessor::Vector3::y](#)

7.37.2.3 float [GtpVisibilityPreprocessor::Vector3::z](#)

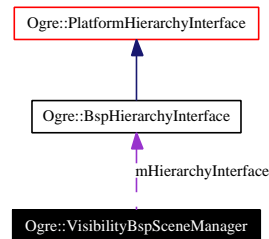
The documentation for this class was generated from the following file:

- [GtpVisibilityPreprocessor/include/Vector3.h](#)

7.38 Ogre::VisibilityBspSceneManager Class Reference

```
#include <OgreVisibilityBspSceneManager.h>
```

Collaboration diagram for Ogre::VisibilityBspSceneManager:



Public Methods

- [VisibilityBspSceneManager](#) (GtpVisibility::Manager *visManager)
- [~VisibilityBspSceneManager](#) ()
- void [_renderVisibleObjects](#) ()
- void [_findVisibleObjects](#) (Camera *cam, bool onlyShadowCasters)
- void [_updateSceneGraph](#) (Camera *cam)
- virtual bool [setOption](#) (const String &, const void *)
- virtual bool [getOption](#) (const String &, void *)
- bool [getOptionValues](#) (const String &key, StringVector &refValueList)
- bool [getOptionKeys](#) (StringVector &refKeys)
- void [setVisibilityManager](#) (GtpVisibility::Manager *visManager)
- GtpVisibility::Manager * [getVisibilityManager](#) (void)

Protected Attributes

- [BspHierarchyInterface](#) * [mHierarchyInterface](#)
- GtpVisibility::Manager * [mVisibilityManager](#)

7.38.1 Detailed Description

This class extends the octree scene manager, using occlusion queries for visibility culling.

7.38.2 Constructor & Destructor Documentation

7.38.2.1 `Ogre::VisibilityBspSceneManager::VisibilityBspSceneManager (GtpVisibility::Manager * visManager)`

7.38.2.2 `Ogre::VisibilityBspSceneManager::~~VisibilityBspSceneManager ()`

7.38.3 Member Function Documentation

7.38.3.1 `void Ogre::VisibilityBspSceneManager::_renderVisibleObjects ()`

7.38.3.2 `void Ogre::VisibilityBspSceneManager::_findVisibleObjects (Camera * cam, bool onlyShadowCasters)`

7.38.3.3 `void Ogre::VisibilityBspSceneManager::_updateSceneGraph (Camera * cam)`

7.38.3.4 `bool Ogre::VisibilityBspSceneManager::setOption (const String &, const void *) [virtual]`

Sets the given option for the SceneManager

Remarks:

Options are: "Algorithm", int *;

7.38.3.5 `bool Ogre::VisibilityBspSceneManager::getOption (const String &, void *) [virtual]`

Gets the given option for the Scene Manager.

Remarks:

See setOption

7.38.3.6 `bool Ogre::VisibilityBspSceneManager::getOptionValues (const String & key, StringVector & refValueList)`

7.38.3.7 `bool Ogre::VisibilityBspSceneManager::getOptionKeys (StringVector & refKeys)`

7.38.3.8 `void Ogre::VisibilityBspSceneManager::setVisibilityManager (GtpVisibility::Manager * visManager)`

Sets the visibility manager.

Parameters:

visManager the visibility manager

7.38.3.9 `GtpVisibility::Manager * Ogre::VisibilityBspSceneManager::getVisibilityManager (void)`

See set.

7.38.4 Member Data Documentation

7.38.4.1 [BspHierarchyInterface](#)* **Ogre::VisibilityBspSceneManager::mHierarchyInterface**
[protected]

7.38.4.2 [GtpVisibility::Manager](#)* **Ogre::VisibilityBspSceneManager::mVisibilityManager**
[protected]

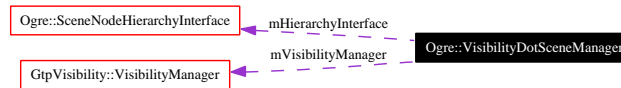
The documentation for this class was generated from the following files:

- [Ogre/include/OgreVisibilityBspSceneManager.h](#)
- [Ogre/src/OgreVisibilityBspSceneManager.cpp](#)

7.39 Ogre::VisibilityDotSceneManager Class Reference

```
#include <OgreVisibilityDotSceneManager.h>
```

Collaboration diagram for Ogre::VisibilityDotSceneManager:



Public Methods

- [VisibilityDotSceneManager](#) ([GtpVisibility::VisibilityManager](#) *visManager)
- [~VisibilityDotSceneManager](#) ()
- void [_renderVisibleObjects](#) ()
- void [_findVisibleObjects](#) (Camera *cam, bool onlyShadowCasters)
- void [_updateSceneGraph](#) (Camera *cam)
- virtual bool [setOption](#) (const String &, const void *)
- virtual bool [getOption](#) (const String &, void *)
- bool [getOptionValues](#) (const String &key, StringVector &refValueList)
- bool [getOptionKeys](#) (StringVector &refKeys)
- void [setVisibilityManager](#) ([GtpVisibility::VisibilityManager](#) *visManager)
- [GtpVisibility::VisibilityManager](#) * [getVisibilityManager](#) ()

Protected Attributes

- [Ogre::SceneNodeHierarchyInterface](#) * mHierarchyInterface
- [GtpVisibility::VisibilityManager](#) * mVisibilityManager

7.39.1 Detailed Description

This class extends the dot scene manager, using occlusion queries for visibility culling.

Remarks:

the scene manager can operate on [Ogre](#) scene descriptions in XML defined by the ".scene" format

7.39.2 Constructor & Destructor Documentation

7.39.2.1 `Ogre::VisibilityDotSceneManager::VisibilityDotSceneManager`
([GtpVisibility::VisibilityManager](#) * *visManager*)

7.39.2.2 `Ogre::VisibilityDotSceneManager::~~VisibilityDotSceneManager` ()

7.39.3 Member Function Documentation

7.39.3.1 `void Ogre::VisibilityDotSceneManager::_renderVisibleObjects` ()

7.39.3.2 `void Ogre::VisibilityDotSceneManager::_findVisibleObjects` (`Camera` * *cam*, `bool` *onlyShadowCasters*)

7.39.3.3 `void Ogre::VisibilityDotSceneManager::_updateSceneGraph` (`Camera` * *cam*)

7.39.3.4 `bool Ogre::VisibilityDotSceneManager::setOption` (`const String &`, `const void *`)
[`virtual`]

Sets the given option for the SceneManager

Remarks:

Options are: "Algorithm", int *;

7.39.3.5 `bool Ogre::VisibilityDotSceneManager::getOption` (`const String &`, `void *`) [`virtual`]

Gets the given option for the Scene VisibilityManager.

Remarks:

See setOption

7.39.3.6 `bool Ogre::VisibilityDotSceneManager::getOptionValues` (`const String &` *key*,
`StringVector &` *refValueList*)

7.39.3.7 `bool Ogre::VisibilityDotSceneManager::getOptionKeys` (`StringVector &` *refKeys*)

7.39.3.8 `void Ogre::VisibilityDotSceneManager::setVisibilityManager`
([GtpVisibility::VisibilityManager](#) * *visManager*)

Sets the visibility manager.

Parameters:

visManager the visibility manager

7.39.3.9 [GtpVisibility::VisibilityManager](#) * `Ogre::VisibilityDotSceneManager::getVisibility-`
`Manager` ()

See set.

7.39.4 Member Data Documentation

7.39.4.1 [Ogre::SceneNodeHierarchyInterface*](#) [Ogre::VisibilityDotSceneManager::mHierarchy-Interface](#) [protected]

7.39.4.2 [GtpVisibility::VisibilityManager*](#) [Ogre::VisibilityDotSceneManager::mVisibility-Manager](#) [protected]

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

- [Ogre/include/OgreVisibilityDotSceneManager.h](#)
- [Ogre/src/OgreVisibilityDotSceneManager.cpp](#)

7.40 GtpVisibility::VisibilityEnvironment Class Reference

```
#include <VisibilityEnvironment.h>
```

Public Types

- enum [CullingManagerType](#) { [FRUSTUM_CULLING](#), [STOP_AND_WAIT](#), [COHERENT_HIERARCHICAL_CULLING](#) }

Public Methods

- [VisibilityEnvironment](#) ()
- void [LoadEnvironment](#) ()

7.40.1 Detailed Description

This class provides different parameters for the visibility manager.

7.40.2 Member Enumeration Documentation

7.40.2.1 enum GtpVisibility::VisibilityEnvironment::CullingManagerType

Different types of occlusion culling algorithms

Enumeration values:

[FRUSTUM_CULLING](#)

[STOP_AND_WAIT](#)

[COHERENT_HIERARCHICAL_CULLING](#)

7.40.3 Constructor & Destructor Documentation

7.40.3.1 GtpVisibility::VisibilityEnvironment::VisibilityEnvironment ()

7.40.4 Member Function Documentation

7.40.4.1 void GtpVisibility::VisibilityEnvironment::LoadEnvironment ()

Loads an environment from disk.

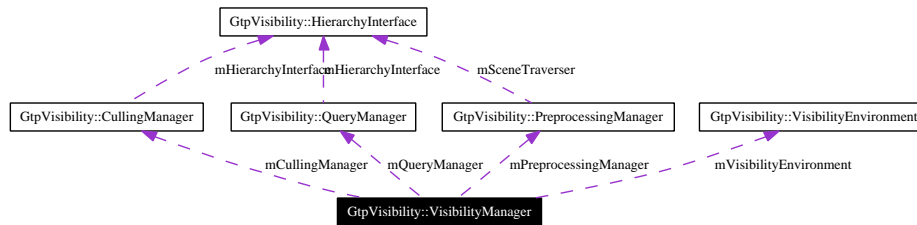
The documentation for this class was generated from the following files:

- GtpVisibility/include/[VisibilityEnvironment.h](#)
- GtpVisibility/src/[VisibilityEnvironment.cpp](#)

7.41 GtpVisibility::VisibilityManager Class Reference

```
#include <VisibilityManager.h>
```

Collaboration diagram for GtpVisibility::VisibilityManager:



Public Methods

- [VisibilityManager](#) ([VisibilityEnvironment](#) *visEnvironment)
- [~VisibilityManager](#) ()
- void [SetCullingManager](#) ([VisibilityEnvironment::CullingManagerType](#) ocmType)
- [CullingManager](#) * [GetCullingManager](#) ()
- void [ApplyVisibilityCulling](#) ()

Protected Attributes

- [CullingManager](#) * [mCullingManager](#)
- [QueryManager](#) * [mQueryManager](#)
- [PreprocessingManager](#) * [mPreprocessingManager](#)
- [VisibilityEnvironment](#) * [mVisibilityEnvironment](#)
- [VisibilityEnvironment::CullingManagerType](#) [mCullingManagerType](#)

7.41.1 Detailed Description

This class manages all forms of visibility. It is the main class of our visibility module and manages online occlusion culling, offline culling, and visibility queries.

7.41.2 Constructor & Destructor Documentation

7.41.2.1 GtpVisibility::VisibilityManager::VisibilityManager ([VisibilityEnvironment](#) *visEnvironment)

Constructor taking the visibility environment object as parameter

Parameters:

visEnvironment the visibility environment

7.41.2.2 **GtpVisibility::VisibilityManager::~~VisibilityManager ()**

7.41.3 Member Function Documentation

7.41.3.1 **void GtpVisibility::VisibilityManager::SetCullingManager**
(**VisibilityEnvironment::CullingManagerType** *ocmType*)

Sets the current online occlusion culling manager, e.g., the stop and wait algorithm or coherent hierarchical culling.

Parameters:

ocmType the online occlusion culling manager type

7.41.3.2 **CullingManager * GtpVisibility::VisibilityManager::GetCullingManager ()**

Returns the current online occlusion culling manager. See set

7.41.3.3 **void GtpVisibility::VisibilityManager::ApplyVisibilityCulling ()**

Applies the online visibility culling algorithm on a scene.

Remarks:

the algorithm depends on the current culling manager.

7.41.4 Member Data Documentation

7.41.4.1 **CullingManager*** **GtpVisibility::VisibilityManager::mCullingManager** [protected]

7.41.4.2 **QueryManager*** **GtpVisibility::VisibilityManager::mQueryManager** [protected]

7.41.4.3 **PreprocessingManager*** **GtpVisibility::VisibilityManager::mPreprocessingManager**
[protected]

7.41.4.4 **VisibilityEnvironment*** **GtpVisibility::VisibilityManager::mVisibilityEnvironment**
[protected]

7.41.4.5 **VisibilityEnvironment::CullingManagerType** **GtpVisibility::VisibilityManager::m-**
CullingManagerType [protected]

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

- [GtpVisibility/include/VisibilityManager.h](#)
- [GtpVisibility/src/VisibilityManager.cpp](#)

7.42 Ogre::VisibilityOctreeSceneManager Class Reference

```
#include <OgreVisibilityOctreeSceneManager.h>
```

Collaboration diagram for Ogre::VisibilityOctreeSceneManager:



Public Methods

- [VisibilityOctreeSceneManager](#) ([GtpVisibility::VisibilityManager](#) *visManager)
- [~VisibilityOctreeSceneManager](#) ()
- void [_renderVisibleObjects](#) ()
- void [_findVisibleObjects](#) (Camera *cam, bool onlyShadowCasters)
- void [_updateSceneGraph](#) (Camera *cam)
- virtual bool [setOption](#) (const String &, const void *)
- virtual bool [getOption](#) (const String &, void *)
- bool [getOptionValues](#) (const String &key, StringVector &refValueList)
- bool [getOptionKeys](#) (StringVector &refKeys)
- void [setVisibilityManager](#) ([GtpVisibility::VisibilityManager](#) *visManager)
- [GtpVisibility::VisibilityManager](#) * [getVisibilityManager](#) (void)

Protected Attributes

- [OctreeHierarchyInterface](#) * [mHierarchyInterface](#)
- [GtpVisibility::VisibilityManager](#) * [mVisibilityManager](#)

7.42.1 Detailed Description

This class extends the octree scene manager, using occlusion queries for visibility culling.

7.42.2 Constructor & Destructor Documentation

7.42.2.1 `Ogre::VisibilityOctreeSceneManager::VisibilityOctreeSceneManager`
([GtpVisibility::VisibilityManager](#) * *visManager*)

7.42.2.2 `Ogre::VisibilityOctreeSceneManager::~~VisibilityOctreeSceneManager` ()

7.42.3 Member Function Documentation

7.42.3.1 `void Ogre::VisibilityOctreeSceneManager::_renderVisibleObjects` ()

7.42.3.2 `void Ogre::VisibilityOctreeSceneManager::_findVisibleObjects` (Camera * *cam*, bool *onlyShadowCasters*)

7.42.3.3 `void Ogre::VisibilityOctreeSceneManager::_updateSceneGraph` (Camera * *cam*)

7.42.3.4 `bool Ogre::VisibilityOctreeSceneManager::setOption` (const String &, const void *)
[virtual]

Sets the given option for the SceneManager

Remarks:

Options are: "Algorithm", int *;

7.42.3.5 `bool Ogre::VisibilityOctreeSceneManager::getOption` (const String &, void *)
[virtual]

Gets the given option for the Scene VisibilityManager.

Remarks:

See setOption

7.42.3.6 `bool Ogre::VisibilityOctreeSceneManager::getOptionValues` (const String & *key*, StringVector & *refValueList*)

7.42.3.7 `bool Ogre::VisibilityOctreeSceneManager::getOptionKeys` (StringVector & *refKeys*)

7.42.3.8 `void Ogre::VisibilityOctreeSceneManager::setVisibilityManager`
([GtpVisibility::VisibilityManager](#) * *visManager*)

Sets the visibility manager.

Parameters:

visManager the visibility manager

7.42.3.9 [GtpVisibility::VisibilityManager](#) * `Ogre::VisibilityOctreeSceneManager::getVisibility-`
`Manager` (void)

See set.

7.42.4 Member Data Documentation

7.42.4.1 [OctreeHierarchyInterface*](#) `Ogre::VisibilityOctreeSceneManager::mHierarchyInterface`
[protected]

7.42.4.2 [GtpVisibility::VisibilityManager*](#) `Ogre::VisibilityOctreeSceneManager::mVisibility-
Manager` [protected]

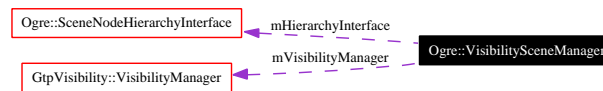
The documentation for this class was generated from the following files:

- [Ogre/include/OgreVisibilityOctreeSceneManager.h](#)
- [Ogre/src/OgreVisibilityOctreeSceneManager.cpp](#)

7.43 Ogre::VisibilitySceneManager Class Reference

```
#include <OgreVisibilitySceneManager.h>
```

Collaboration diagram for Ogre::VisibilitySceneManager:



Public Methods

- [VisibilitySceneManager \(GtpVisibility::VisibilityManager *visManager\)](#)
- [~VisibilitySceneManager \(\)](#)
- [void _renderVisibleObjects \(\)](#)
- [void _findVisibleObjects \(Camera *cam, bool onlyShadowCasters\)](#)
- [void _updateSceneGraph \(Camera *cam\)](#)
- [virtual bool setOption \(const String &, const void *\)](#)
- [virtual bool getOption \(const String &, void *\)](#)
- [bool getOptionValues \(const String &key, StringVector &refValueList\)](#)
- [bool getOptionKeys \(StringVector &refKeys\)](#)
- [void setVisibilityManager \(GtpVisibility::VisibilityManager *visManager\)](#)
- [GtpVisibility::VisibilityManager * getVisibilityManager \(\)](#)

Protected Attributes

- [SceneNodeHierarchyInterface * mHierarchyInterface](#)
- [GtpVisibility::VisibilityManager * mVisibilityManager](#)

7.43.1 Detailed Description

This class implements the default scene manager, using occlusion queries for visibility culling.

7.43.2 Constructor & Destructor Documentation

7.43.2.1 [Ogre::VisibilitySceneManager::VisibilitySceneManager \(GtpVisibility::VisibilityManager * visManager\)](#)

7.43.2.2 [Ogre::VisibilitySceneManager::~~VisibilitySceneManager \(\)](#)

7.43.3 Member Function Documentation

7.43.3.1 [void Ogre::VisibilitySceneManager::_renderVisibleObjects \(\)](#)

Overriden from SceneManager. Renders the scene using occlusion culling.

Remarks:

the type of algorithm is specified by the occlusion culling manager type using by the visibility manager.

7.43.3.2 `void Ogre::VisibilitySceneManager::findVisibleObjects (Camera * cam, bool onlyShadowCasters)`

7.43.3.3 `void Ogre::VisibilitySceneManager::updateSceneGraph (Camera * cam)`

7.43.3.4 `bool Ogre::VisibilitySceneManager::setOption (const String &, const void *)`
[virtual]

Sets the given option for the SceneManager

Remarks:

Options are: "Algorithm", int *;

7.43.3.5 `bool Ogre::VisibilitySceneManager::getOption (const String &, void *)` [virtual]

Gets the given option for the Scene VisibilityManager.

Remarks:

See setOption

7.43.3.6 `bool Ogre::VisibilitySceneManager::getOptionValues (const String & key, StringVector & refValueList)`

7.43.3.7 `bool Ogre::VisibilitySceneManager::getOptionKeys (StringVector & refKeys)`

7.43.3.8 `void Ogre::VisibilitySceneManager::setVisibilityManager (GtpVisibility::VisibilityManager * visManager)`

Sets the visibility manager.

Parameters:

visManager the visibility manager

7.43.3.9 `GtpVisibility::VisibilityManager * Ogre::VisibilitySceneManager::getVisibilityManager ()`

See set.

7.43.4 Member Data Documentation

7.43.4.1 `SceneNodeHierarchyInterface* Ogre::VisibilitySceneManager::mHierarchyInterface`
[protected]

7.43.4.2 `GtpVisibility::VisibilityManager* Ogre::VisibilitySceneManager::mVisibilityManager`
[protected]

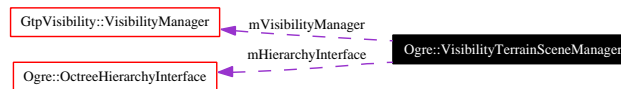
The documentation for this class was generated from the following files:

- [Ogre/include/OgreVisibilitySceneManager.h](#)
- [Ogre/src/OgreVisibilitySceneManager.cpp](#)

7.44 Ogre::VisibilityTerrainSceneManager Class Reference

```
#include <OgreVisibilityTerrainSceneManager.h>
```

Collaboration diagram for Ogre::VisibilityTerrainSceneManager:



Public Methods

- [VisibilityTerrainSceneManager](#) ([GtpVisibility::VisibilityManager](#) *visManager)
- [~VisibilityTerrainSceneManager](#) ()
- void [_renderVisibleObjects](#) (void)
- void [_findVisibleObjects](#) (Camera *cam, bool onlyShadowCasters)
- void [_updateSceneGraph](#) (Camera *cam)
- virtual bool [setOption](#) (const String &, const void *)
- virtual bool [getOption](#) (const String &, void *)
- bool [getOptionValues](#) (const String &key, StringVector &refValueList)
- bool [getOptionKeys](#) (StringVector &refKeys)
- void [setVisibilityManager](#) ([GtpVisibility::VisibilityManager](#) *visManager)
- [GtpVisibility::VisibilityManager](#) * [getVisibilityManager](#) ()

Protected Attributes

- [OctreeHierarchyInterface](#) * mHierarchyInterface
- [GtpVisibility::VisibilityManager](#) * mVisibilityManager

7.44.1 Detailed Description

This class extends the terrain scene manager, using occlusion queries for visibility culling.

7.44.2 Constructor & Destructor Documentation

7.44.2.1 **Ogre::VisibilityTerrainSceneManager::VisibilityTerrainSceneManager**
([GtpVisibility::VisibilityManager](#) * *visManager*)

7.44.2.2 **Ogre::VisibilityTerrainSceneManager::~~VisibilityTerrainSceneManager** ()

7.44.3 Member Function Documentation

7.44.3.1 **void Ogre::VisibilityTerrainSceneManager::_renderVisibleObjects** (void)

7.44.3.2 **void Ogre::VisibilityTerrainSceneManager::_findVisibleObjects** (Camera * *cam*, bool *onlyShadowCasters*)

7.44.3.3 **void Ogre::VisibilityTerrainSceneManager::_updateSceneGraph** (Camera * *cam*)

7.44.3.4 **bool Ogre::VisibilityTerrainSceneManager::setOption** (const String &, const void *)
[virtual]

Sets the given option for the SceneManager

Remarks:

Options are: "Algorithm", int *;

7.44.3.5 **bool Ogre::VisibilityTerrainSceneManager::getOption** (const String &, void *)
[virtual]

Gets the given option for the Scene VisibilityManager.

Remarks:

See setOption

7.44.3.6 **bool Ogre::VisibilityTerrainSceneManager::getOptionValues** (const String & *key*, StringVector & *refValueList*)

7.44.3.7 **bool Ogre::VisibilityTerrainSceneManager::getOptionKeys** (StringVector & *refKeys*)

7.44.3.8 **void Ogre::VisibilityTerrainSceneManager::setVisibilityManager**
([GtpVisibility::VisibilityManager](#) * *visManager*)

Sets the visibility manager.

Parameters:

visManager the visibility manager

7.44.3.9 [GtpVisibility::VisibilityManager](#) * **Ogre::VisibilityTerrainSceneManager::getVisibilityManager** ()

See set.

7.44.4 Member Data Documentation

7.44.4.1 [OctreeHierarchyInterface*](#) `Ogre::VisibilityTerrainSceneManager::mHierarchyInterface`
[protected]

7.44.4.2 [GtpVisibility::VisibilityManager*](#) `Ogre::VisibilityTerrainSceneManager::mVisibility-
Manager` [protected]

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

- [Ogre/include/OgreVisibilityTerrainSceneManager.h](#)
- [Ogre/src/OgreVisibilityTerrainSceneManager.cpp](#)

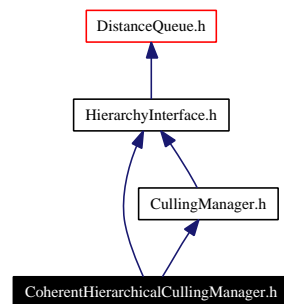
Chapter 8

GameTools Visibility Modules File Documentation

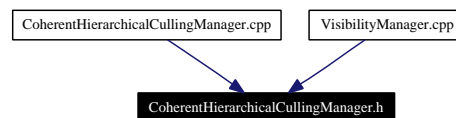
8.1 GtpVisibility/include/CoherentHierarchicalCullingManager.h File Reference

```
#include "CullingManager.h"  
#include "HierarchyInterface.h"
```

Include dependency graph for CoherentHierarchicalCullingManager.h:



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



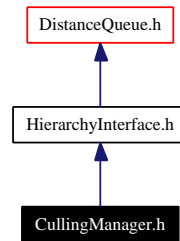
Namespaces

- namespace [GtpVisibility](#)

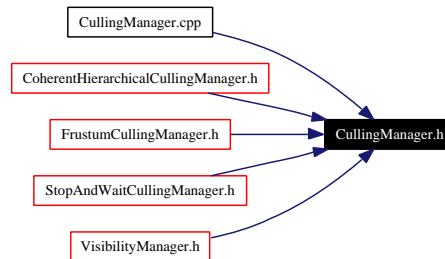
8.2 GtpVisibility/include/CullingManager.h File Reference

```
#include "HierarchyInterface.h"
```

Include dependency graph for CullingManager.h:



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



Namespaces

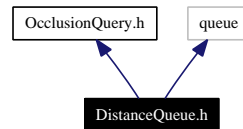
- namespace [GtpVisibility](#)

8.3 GtpVisibility/include/DistanceQueue.h File Reference

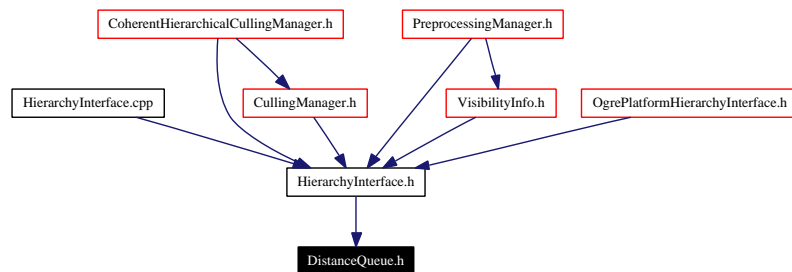
```
#include "OcclusionQuery.h"
```

```
#include <queue>
```

Include dependency graph for DistanceQueue.h:



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



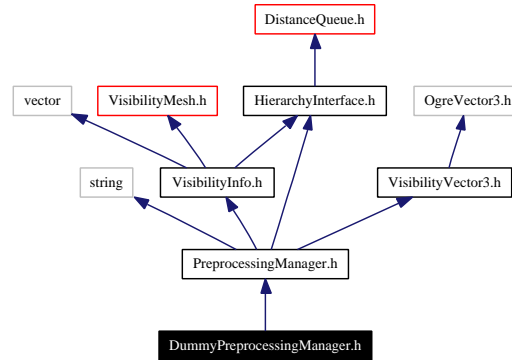
Namespaces

- namespace [GtpVisibility](#)

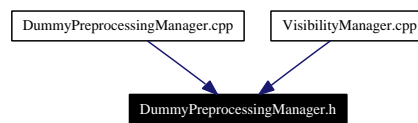
8.4 GtpVisibility/include/DummyPreprocessingManager.h File Reference

```
#include "PreprocessingManager.h"
```

Include dependency graph for DummyPreprocessingManager.h:



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



Namespaces

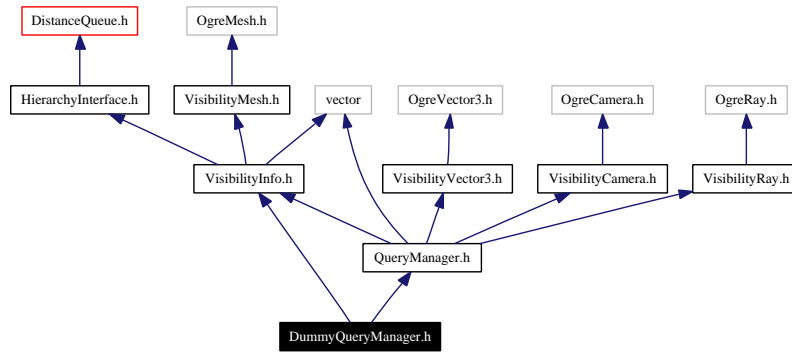
- namespace [GtpVisibility](#)

8.5 GtpVisibility/include/DummyQueryManager.h File Reference

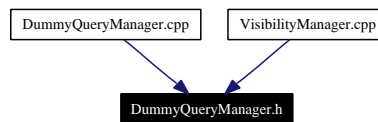
```
#include "VisibilityInfo.h"
```

```
#include "QueryManager.h"
```

Include dependency graph for DummyQueryManager.h:



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



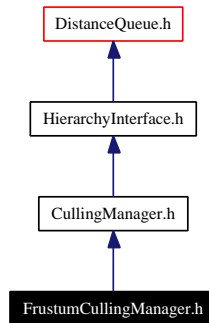
Namespaces

- namespace [GtpVisibility](#)

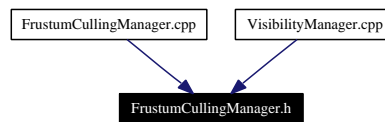
8.6 GtpVisibility/include/FrustumCullingManager.h File Reference

```
#include "CullingManager.h"
```

Include dependency graph for FrustumCullingManager.h:



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



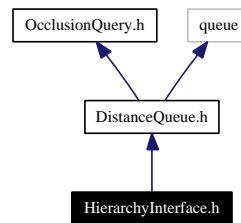
Namespaces

- namespace [GtpVisibility](#)

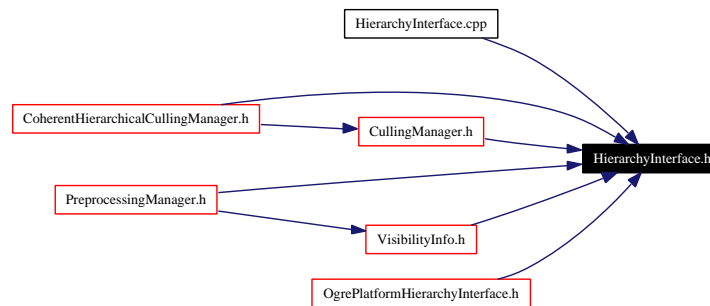
8.7 GtpVisibility/include/HierarchyInterface.h File Reference

```
#include "DistanceQueue.h"
```

Include dependency graph for HierarchyInterface.h:



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

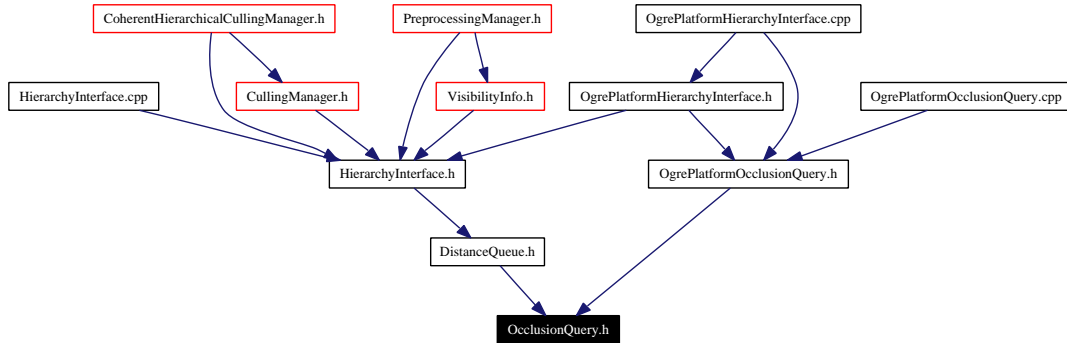


Namespaces

- namespace [GtpVisibility](#)

8.8 GtpVisibility/include/OcclusionQuery.h File Reference

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



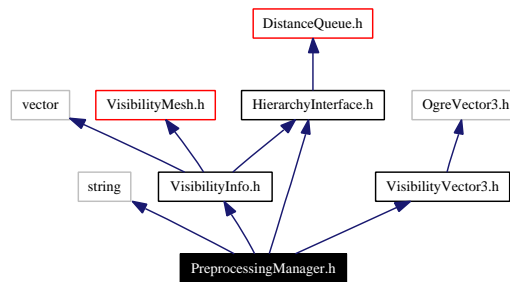
Namespaces

- namespace [GtpVisibility](#)

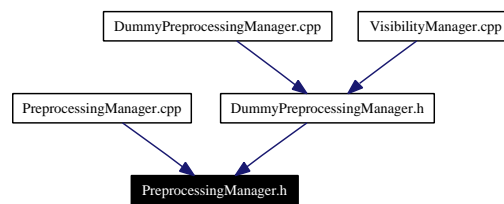
8.9 GtpVisibility/include/PreprocessingManager.h File Reference

```
#include <string>
#include "HierarchyInterface.h"
#include "VisibilityInfo.h"
#include "VisibilityVector3.h"
```

Include dependency graph for PreprocessingManager.h:



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



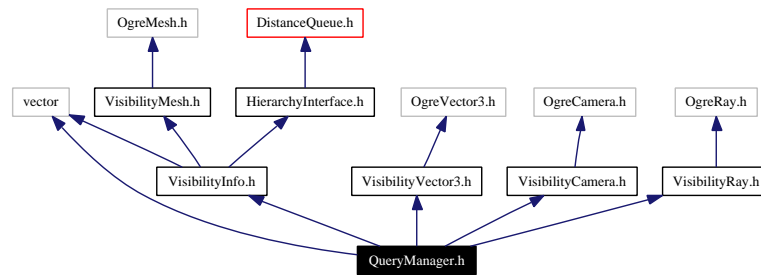
Namespaces

- namespace [GtpVisibility](#)
- namespace [std](#)

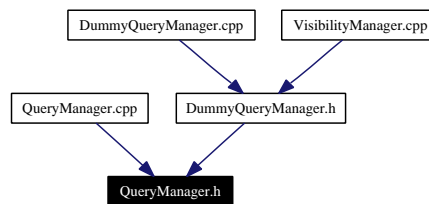
8.10 GtpVisibility/include/QueryManager.h File Reference

```
#include <vector>
#include "VisibilityInfo.h"
#include "VisibilityVector3.h"
#include "VisibilityCamera.h"
#include "VisibilityRay.h"
```

Include dependency graph for QueryManager.h:



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



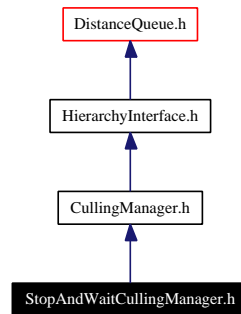
Namespaces

- namespace [GtpVisibility](#)

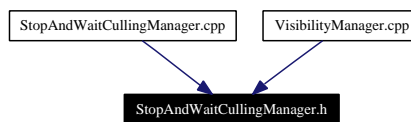
8.11 GtpVisibility/include/StopAndWaitCullingManager.h File Reference

```
#include "CullingManager.h"
```

Include dependency graph for StopAndWaitCullingManager.h:



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



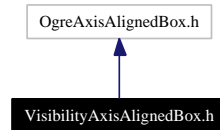
Namespaces

- namespace [GtpVisibility](#)

8.12 GtpVisibility/include/VisibilityAxisAlignedBox.h File Reference

```
#include "OgreAxisAlignedBox.h"
```

Include dependency graph for VisibilityAxisAlignedBox.h:



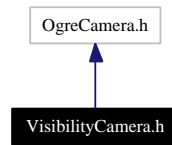
Namespaces

- namespace [GtpVisibility](#)

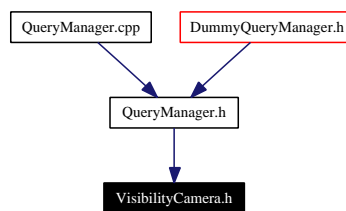
8.13 GtpVisibility/include/VisibilityCamera.h File Reference

```
#include "OgreCamera.h"
```

Include dependency graph for VisibilityCamera.h:



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

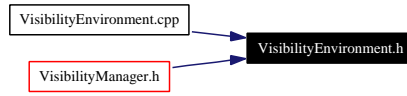


Namespaces

- namespace [GtpVisibility](#)

8.14 GtpVisibility/include/VisibilityEnvironment.h File Reference

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

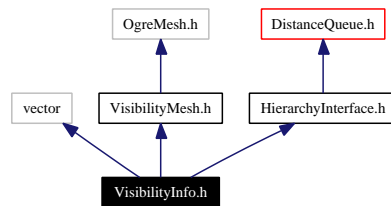


Namespaces

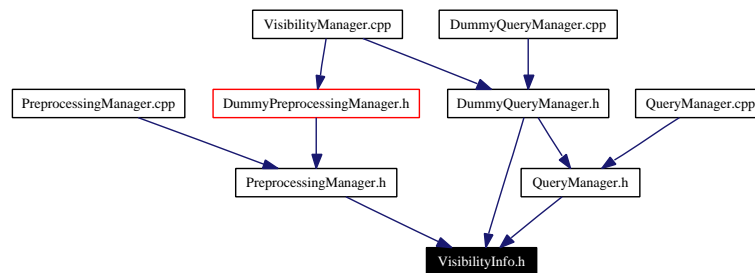
- namespace [GtpVisibility](#)

8.15 GtpVisibility/include/VisibilityInfo.h File Reference

```
#include <vector>
#include "VisibilityMesh.h"
#include "HierarchyInterface.h"
Include dependency graph for VisibilityInfo.h:
```



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



Namespaces

- namespace [GtpVisibility](#)

Defines

- #define [InfoContainer](#) std::vector

8.15.1 Define Documentation

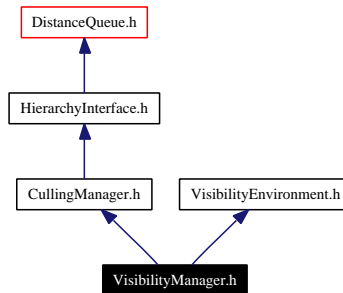
8.15.1.1 #define InfoContainer std::vector

8.16 GtpVisibility/include/VisibilityManager.h File Reference

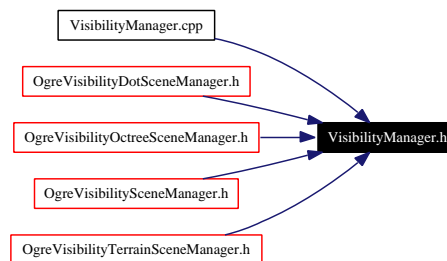
```
#include "CullingManager.h"
```

```
#include "VisibilityEnvironment.h"
```

Include dependency graph for VisibilityManager.h:



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



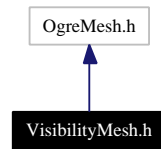
Namespaces

- namespace [GtpVisibility](#)

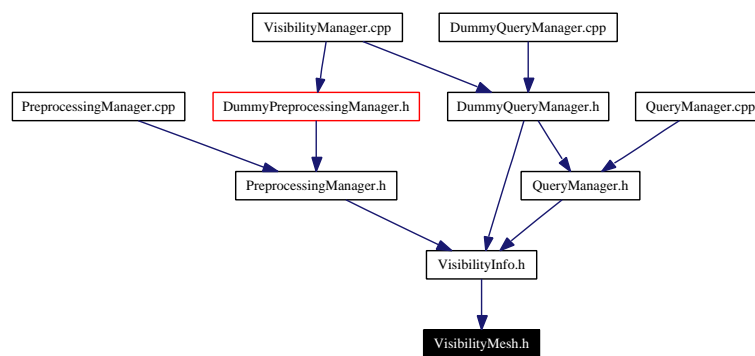
8.17 GtpVisibility/include/VisibilityMesh.h File Reference

```
#include "OgreMesh.h"
```

Include dependency graph for VisibilityMesh.h:



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



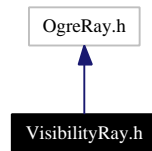
Namespaces

- namespace [GtpVisibility](#)

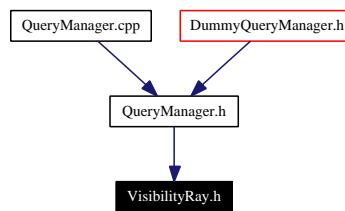
8.18 GtpVisibility/include/VisibilityRay.h File Reference

```
#include "OgreRay.h"
```

Include dependency graph for VisibilityRay.h:



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



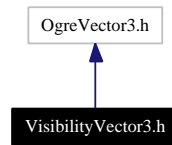
Namespaces

- namespace [GtpVisibility](#)

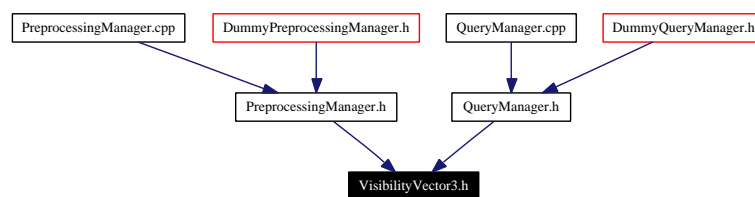
8.19 GtpVisibility/include/VisibilityVector3.h File Reference

```
#include "OgreVector3.h"
```

Include dependency graph for VisibilityVector3.h:



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



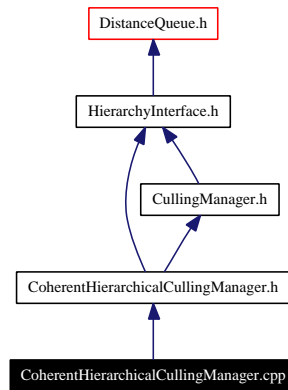
Namespaces

- namespace [GtpVisibility](#)

8.20 GtpVisibility/src/CoherentHierarchicalCullingManager.cpp File Reference

```
#include "CoherentHierarchicalCullingManager.h"
```

Include dependency graph for CoherentHierarchicalCullingManager.cpp:



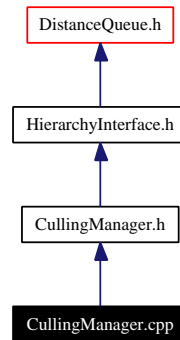
Namespaces

- namespace [GtpVisibility](#)

8.21 GtpVisibility/src/CullingManager.cpp File Reference

```
#include "CullingManager.h"
```

Include dependency graph for CullingManager.cpp:



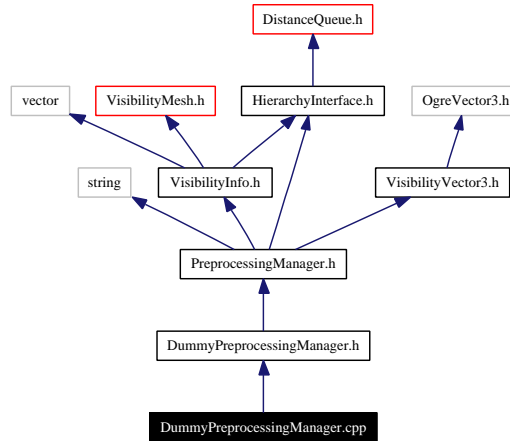
Namespaces

- namespace [GtpVisibility](#)

8.22 GtpVisibility/src/DummyPreprocessingManager.cpp File Reference

```
#include "DummyPreprocessingManager.h"
```

Include dependency graph for DummyPreprocessingManager.cpp:



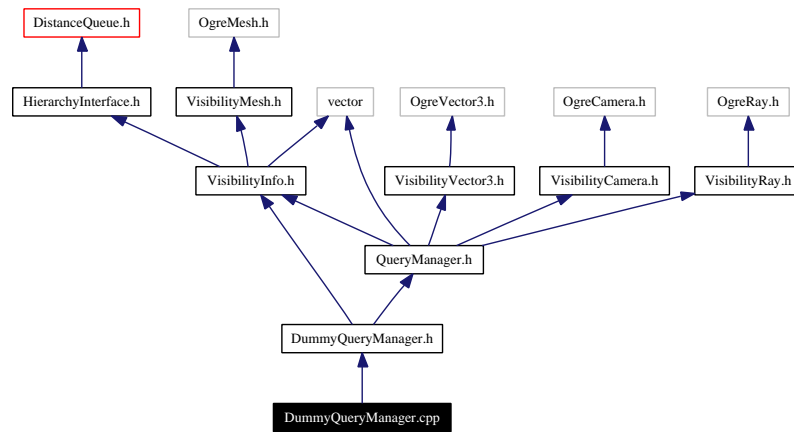
Namespaces

- namespace [GtpVisibility](#)

8.23 GtpVisibility/src/DummyQueryManager.cpp File Reference

```
#include "DummyQueryManager.h"
```

Include dependency graph for DummyQueryManager.cpp:



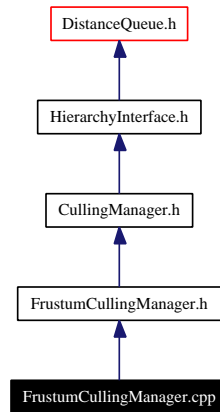
Namespaces

- namespace `GtpVisibility`

8.24 GtpVisibility/src/FrustumCullingManager.cpp File Reference

```
#include "FrustumCullingManager.h"
```

Include dependency graph for FrustumCullingManager.cpp:



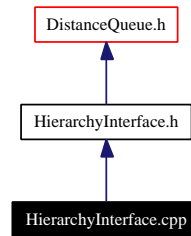
Namespaces

- namespace [GtpVisibility](#)

8.25 GtpVisibility/src/HierarchyInterface.cpp File Reference

```
#include "HierarchyInterface.h"
```

Include dependency graph for HierarchyInterface.cpp:



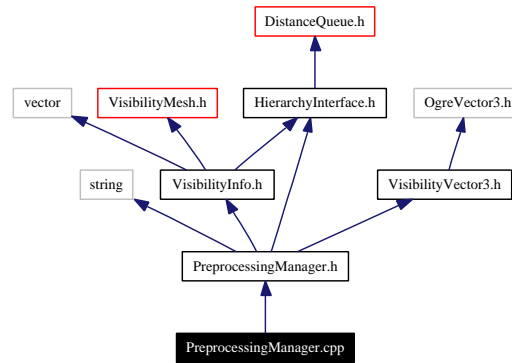
Namespaces

- namespace [GtpVisibility](#)

8.26 GtpVisibility/src/PreprocessingManager.cpp File Reference

```
#include "PreprocessingManager.h"
```

Include dependency graph for PreprocessingManager.cpp:



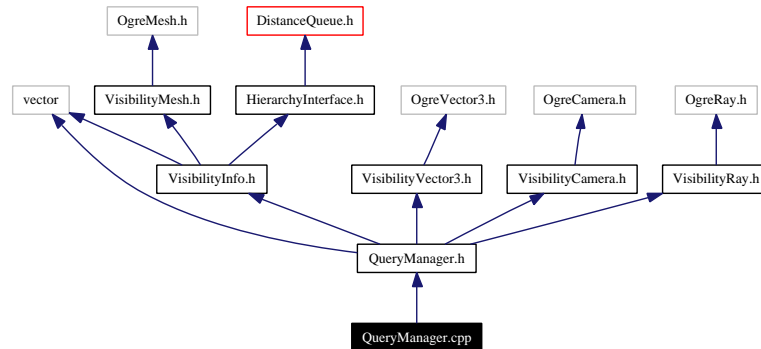
Namespaces

- namespace [GtpVisibility](#)

8.27 GtpVisibility/src/QueryManager.cpp File Reference

```
#include "QueryManager.h"
```

Include dependency graph for QueryManager.cpp:



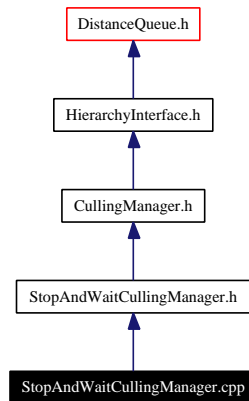
Namespaces

- namespace [GtpVisibility](#)

8.28 GtpVisibility/src/StopAndWaitCullingManager.cpp File Reference

```
#include "StopAndWaitCullingManager.h"
```

Include dependency graph for StopAndWaitCullingManager.cpp:



Namespaces

- namespace [GtpVisibility](#)

8.29 GtpVisibility/src/VisibilityEnvironment.cpp File Reference

```
#include "VisibilityEnvironment.h"
```

Include dependency graph for VisibilityEnvironment.cpp:



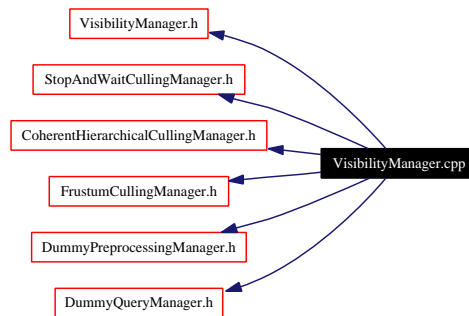
Namespaces

- namespace [GtpVisibility](#)

8.30 GtpVisibility/src/VisibilityManager.cpp File Reference

```
#include "VisibilityManager.h"  
#include "StopAndWaitCullingManager.h"  
#include "CoherentHierarchicalCullingManager.h"  
#include "FrustumCullingManager.h"  
#include "DummyPreprocessingManager.h"  
#include "DummyQueryManager.h"
```

Include dependency graph for VisibilityManager.cpp:



Namespaces

- namespace [GtpVisibility](#)

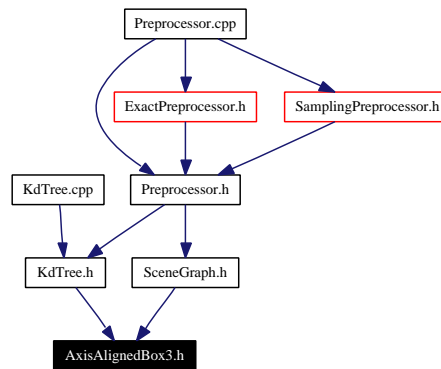
8.31 GtpVisibilityPreprocessor/include/AxisAlignedBox3.h File Reference

```
#include "vector3.h"
```

Include dependency graph for AxisAlignedBox3.h:



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



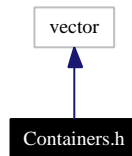
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

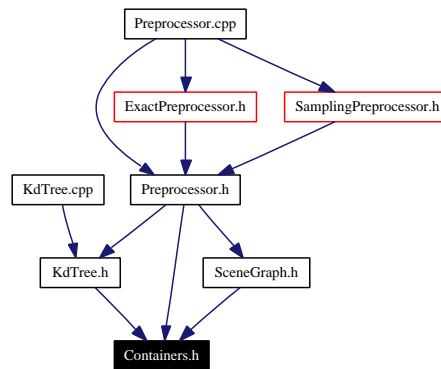
8.32 GtpVisibilityPreprocessor/include/Containers.h File Reference

```
#include <vector>
```

Include dependency graph for Containers.h:



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



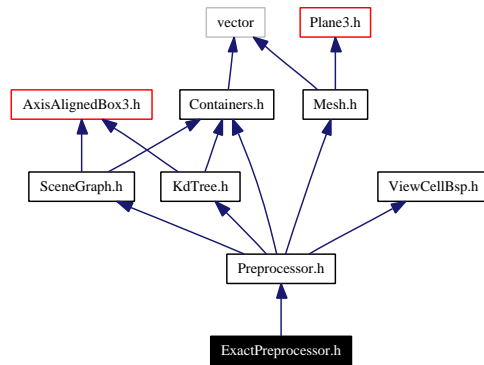
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

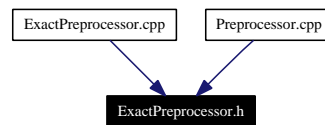
8.33 GtpVisibilityPreprocessor/include/ExactPreprocessor.h File Reference

```
#include "Preprocessor.h"
```

Include dependency graph for ExactPreprocessor.h:



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

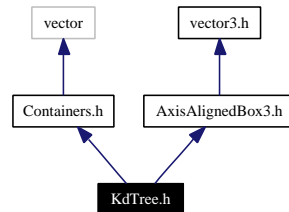


Namespaces

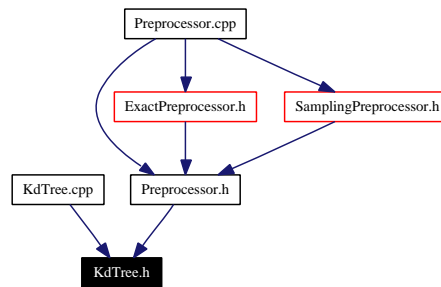
- namespace [GtpVisibilityPreprocessor](#)

8.34 GtpVisibilityPreprocessor/include/KdTree.h File Reference

```
#include "Containers.h"  
#include "AxisAlignedBox3.h"  
Include dependency graph for KdTree.h:
```



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



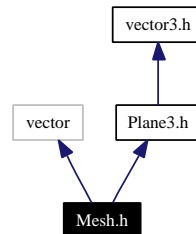
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

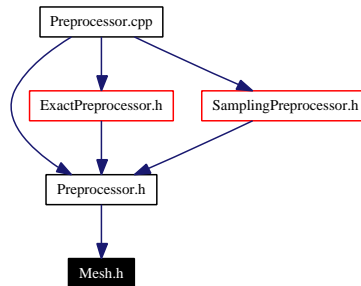
8.35 GtpVisibilityPreprocessor/include/Mesh.h File Reference

```
#include <vector>
#include "Plane3.h"
```

Include dependency graph for Mesh.h:



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



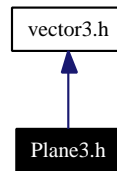
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

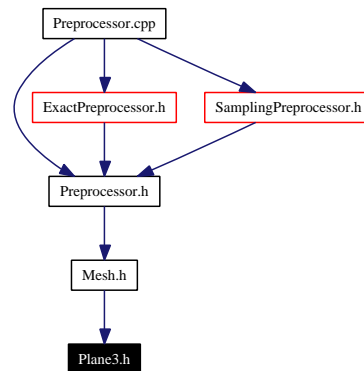
8.36 GtpVisibilityPreprocessor/include/Plane3.h File Reference

```
#include "vector3.h"
```

Include dependency graph for Plane3.h:



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



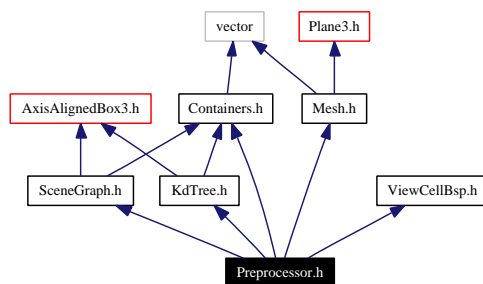
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

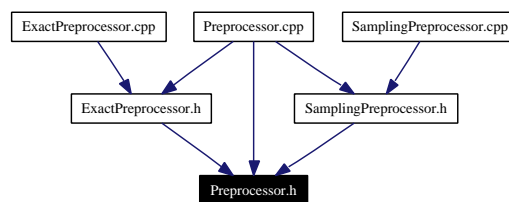
8.37 GtpVisibilityPreprocessor/include/Preprocessor.h File Reference

```
#include "Containers.h"  
#include "Mesh.h"  
#include "KdTree.h"  
#include "ViewCellBsp.h"  
#include "SceneGraph.h"
```

Include dependency graph for Preprocessor.h:



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



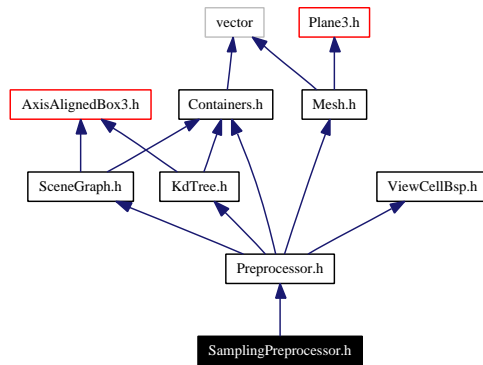
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

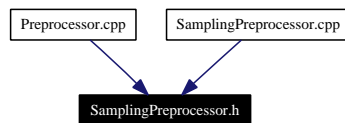
8.38 GtpVisibilityPreprocessor/include/SamplingPreprocessor.h File Reference

```
#include "Preprocessor.h"
```

Include dependency graph for SamplingPreprocessor.h:



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



Namespaces

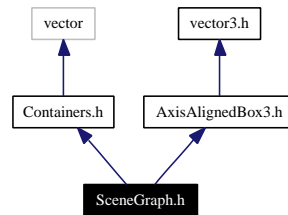
- namespace [GtpVisibilityPreprocessor](#)

8.39 GtpVisibilityPreprocessor/include/SceneGraph.h File Reference

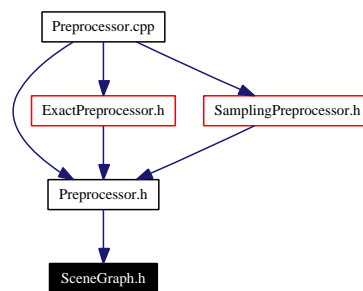
```
#include "Containers.h"
```

```
#include "AxisAlignedBox3.h"
```

Include dependency graph for SceneGraph.h:



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

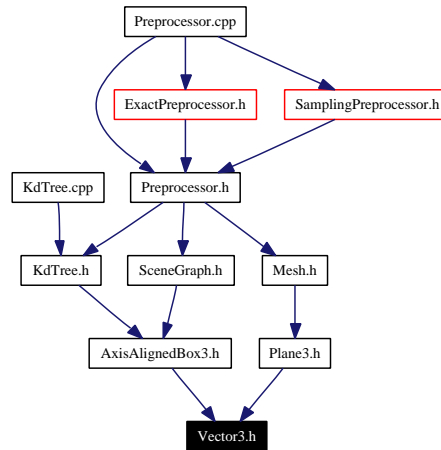


Namespaces

- namespace [GtpVisibilityPreprocessor](#)

8.40 GtpVisibilityPreprocessor/include/Vector3.h File Reference

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

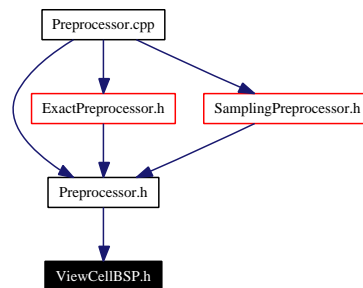


Namespaces

- namespace [GtpVisibilityPreprocessor](#)

8.41 GtpVisibilityPreprocessor/include/ViewCellBSP.h File Reference

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



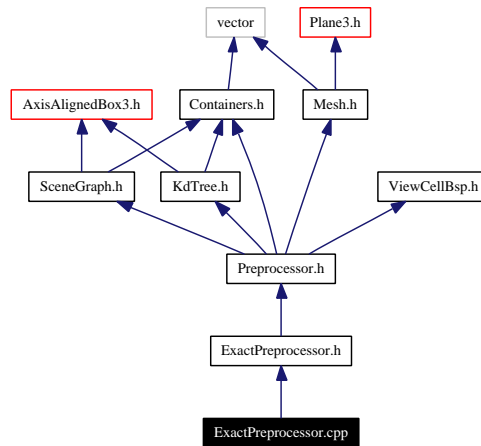
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

8.42 GtpVisibilityPreprocessor/src/ExactPreprocessor.cpp File Reference

```
#include "ExactPreprocessor.h"
```

Include dependency graph for ExactPreprocessor.cpp:



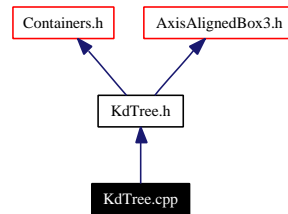
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

8.43 GtpVisibilityPreprocessor/src/KdTree.cpp File Reference

```
#include "KdTree.h"
```

Include dependency graph for KdTree.cpp:



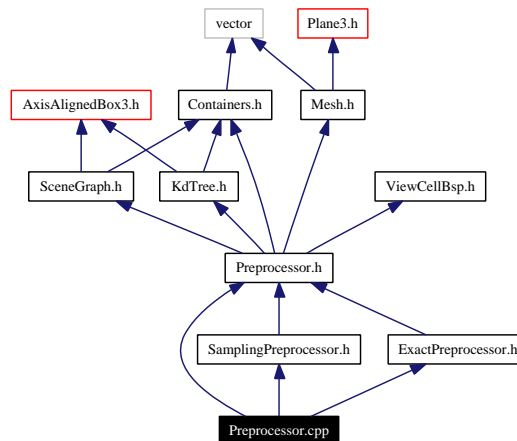
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

8.44 GtpVisibilityPreprocessor/src/Preprocessor.cpp File Reference

```
#include "Preprocessor.h"
#include "SamplingPreprocessor.h"
#include "ExactPreprocessor.h"
```

Include dependency graph for Preprocessor.cpp:



Namespaces

- namespace [GtpVisibilityPreprocessor](#)

Functions

- int [main](#) (int argc, char **argv)

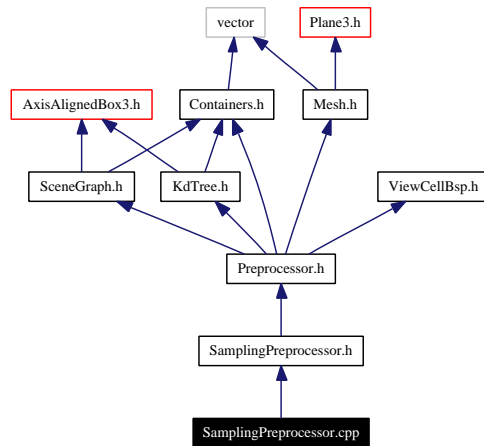
8.44.1 Function Documentation

8.44.1.1 int main (int argc, char ** argv)

8.45 GtpVisibilityPreprocessor/src/SamplingPreprocessor.cpp File Reference

```
#include "SamplingPreprocessor.h"
```

Include dependency graph for SamplingPreprocessor.cpp:



Namespaces

- namespace [GtpVisibilityPreprocessor](#)

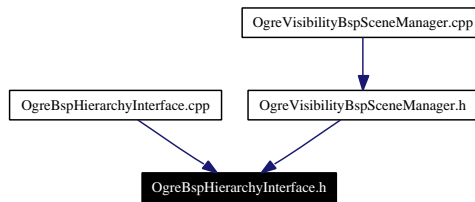
8.46 Ogre/include/OgreBspHierarchyInterface.h File Reference

```
#include "OgrePlatformHierarchyInterface.h"
```

Include dependency graph for OgreBspHierarchyInterface.h:



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



Namespaces

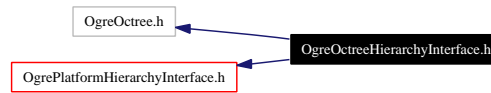
- namespace [Ogre](#)

8.47 Ogre/include/OgreOctreeHierarchyInterface.h File Reference

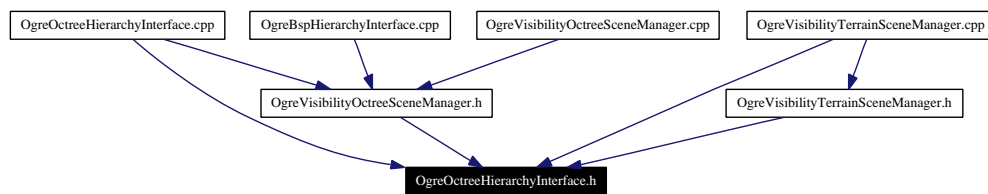
```
#include <OgreOctree.h>
```

```
#include "OgrePlatformHierarchyInterface.h"
```

Include dependency graph for OgreOctreeHierarchyInterface.h:



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



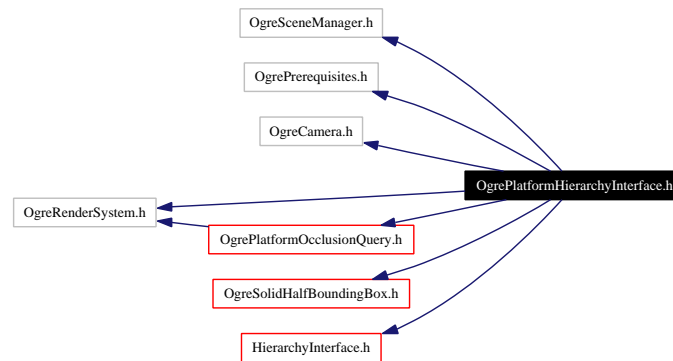
Namespaces

- namespace [Ogre](#)

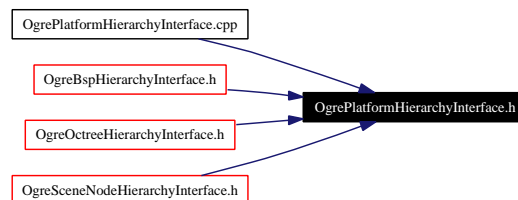
8.48 Ogre/include/OgrePlatformHierarchyInterface.h File Reference

```
#include <OgreSceneManager.h>
#include <OgrePrerequisites.h>
#include <OgreCamera.h>
#include <OgreRenderSystem.h>
#include "OgreSolidHalfBoundingBox.h"
#include "HierarchyInterface.h"
#include "OgrePlatformOcclusionQuery.h"
```

Include dependency graph for OgrePlatformHierarchyInterface.h:



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



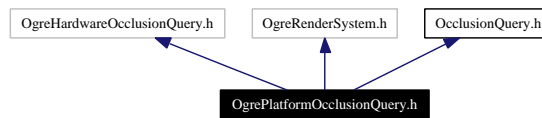
Namespaces

- namespace [Ogre](#)

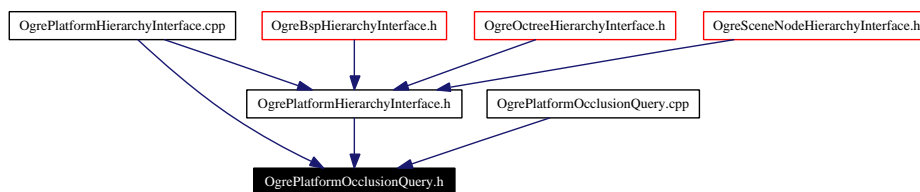
8.49 Ogre/include/OgrePlatformOcclusionQuery.h File Reference

```
#include <OgreHardwareOcclusionQuery.h>
#include <OgreRenderSystem.h>
#include "OcclusionQuery.h"
```

Include dependency graph for OgrePlatformOcclusionQuery.h:



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



Namespaces

- namespace [Ogre](#)

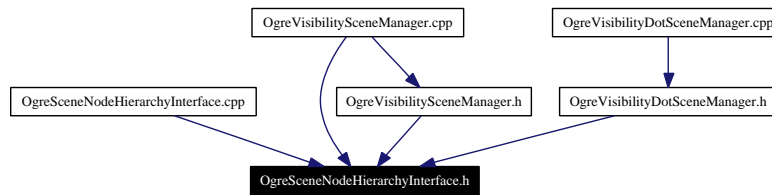
8.50 Ogre/include/OgreSceneNodeHierarchyInterface.h File Reference

```
#include "OgrePlatformHierarchyInterface.h"
```

Include dependency graph for OgreSceneNodeHierarchyInterface.h:



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



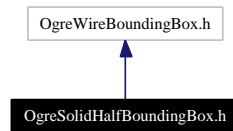
Namespaces

- namespace [Ogre](#)

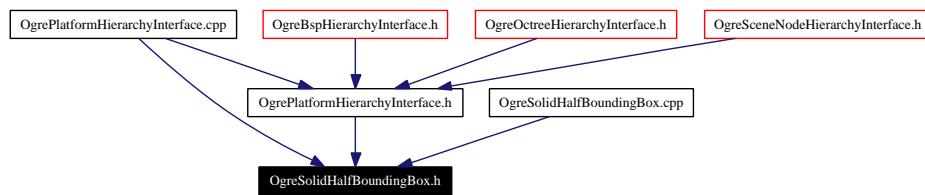
8.51 Ogre/include/OgreSolidHalfBoundingBox.h File Reference

```
#include "OgreWireBoundingBox.h"
```

Include dependency graph for OgreSolidHalfBoundingBox.h:



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



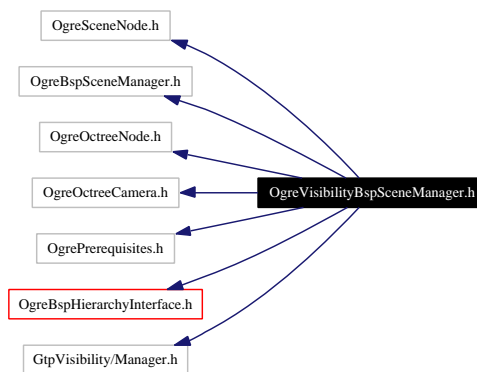
Namespaces

- namespace [Ogre](#)

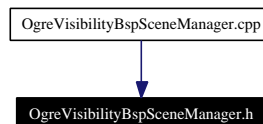
8.52 Ogre/include/OgreVisibilityBspSceneManager.h File Reference

```
#include <OgreSceneNode.h>
#include <OgreBspSceneManager.h>
#include <OgreOctreeNode.h>
#include <OgreOctreeCamera.h>
#include <OgrePrerequisites.h>
#include "OgreBspHierarchyInterface.h"
#include "GtpVisibility/Manager.h"
```

Include dependency graph for OgreVisibilityBspSceneManager.h:



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



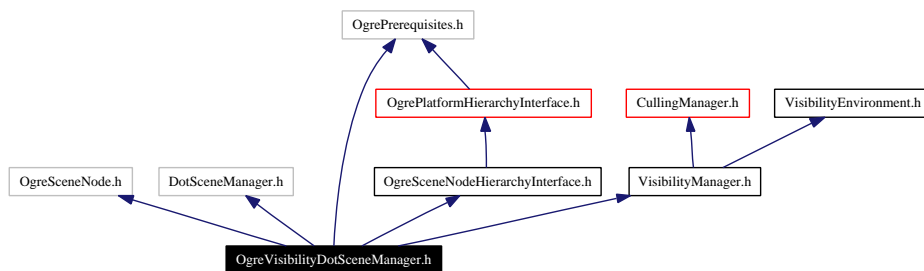
Namespaces

- namespace [Ogre](#)

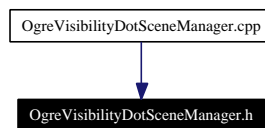
8.53 Ogre/include/OgreVisibilityDotSceneManager.h File Reference

```
#include <OgreSceneNode.h>
#include <DotSceneManager.h>
#include <OgrePrerequisites.h>
#include "VisibilityManager.h"
#include "OgreSceneNodeHierarchyInterface.h"
```

Include dependency graph for OgreVisibilityDotSceneManager.h:



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



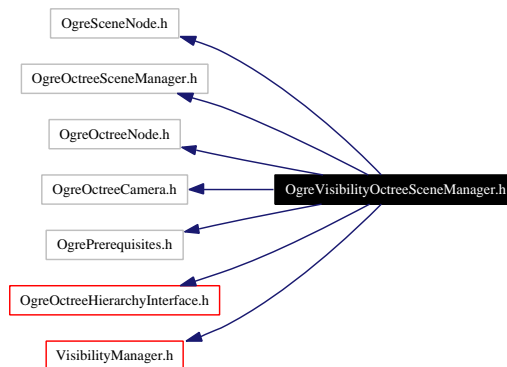
Namespaces

- namespace [Ogre](#)

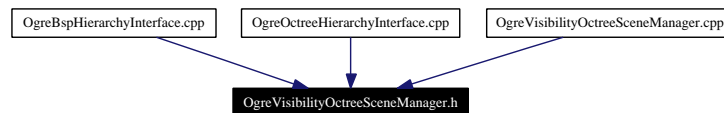
8.54 Ogre/include/OgreVisibilityOctreeSceneManager.h File Reference

```
#include <OgreSceneNode.h>
#include <OgreOctreeSceneManager.h>
#include <OgreOctreeNode.h>
#include <OgreOctreeCamera.h>
#include <OgrePrerequisites.h>
#include "OgreOctreeHierarchyInterface.h"
#include "VisibilityManager.h"
```

Include dependency graph for OgreVisibilityOctreeSceneManager.h:



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



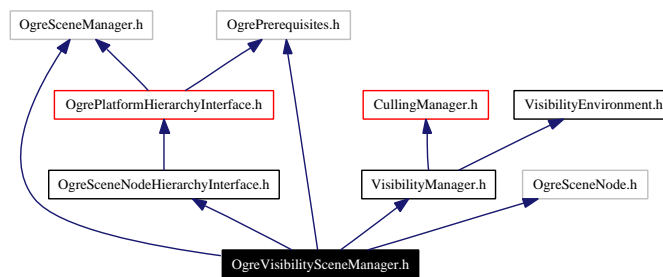
Namespaces

- namespace `Ogre`

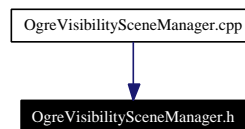
8.55 Ogre/include/OgreVisibilitySceneManager.h File Reference

```
#include "OgreSceneNodeHierarchyInterface.h"
#include "VisibilityManager.h"
#include <OgreSceneNode.h>
#include <OgreSceneManager.h>
#include <OgrePrerequisites.h>
```

Include dependency graph for OgreVisibilitySceneManager.h:



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



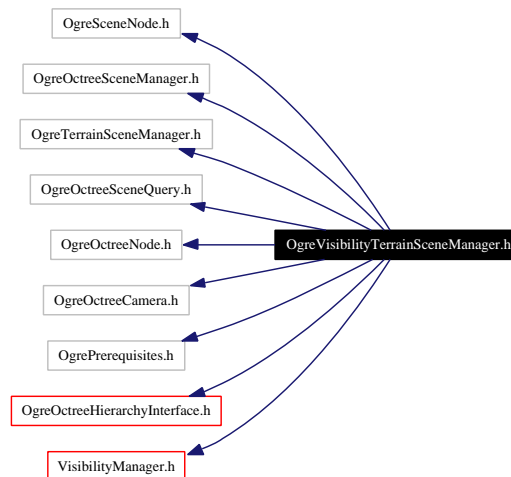
Namespaces

- namespace [Ogre](#)

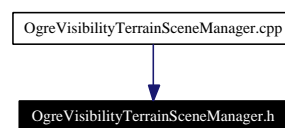
8.56 Ogre/include/OgreVisibilityTerrainSceneManager.h File Reference

```
#include <OgreSceneNode.h>
#include <OgreOctreeSceneManager.h>
#include <OgreTerrainSceneManager.h>
#include <OgreOctreeSceneQuery.h>
#include <OgreOctreeNode.h>
#include <OgreOctreeCamera.h>
#include <OgrePrerequisites.h>
#include "OgreOctreeHierarchyInterface.h"
#include "VisibilityManager.h"
```

Include dependency graph for OgreVisibilityTerrainSceneManager.h:



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



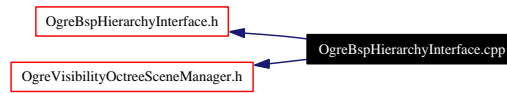
Namespaces

- namespace [Ogre](#)

8.57 Ogre/src/OgreBspHierarchyInterface.cpp File Reference

```
#include "OgreBspHierarchyInterface.h"  
#include "OgreVisibilityOctreeSceneManager.h"
```

Include dependency graph for OgreBspHierarchyInterface.cpp:



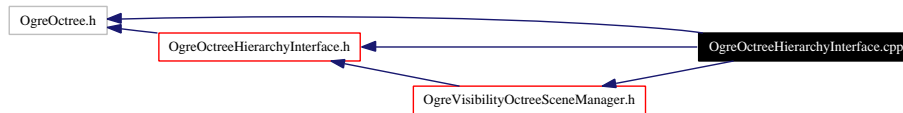
Namespaces

- namespace [Ogre](#)

8.58 Ogre/src/OgreOctreeHierarchyInterface.cpp File Reference

```
#include "OgreOctreeHierarchyInterface.h"  
#include "OgreVisibilityOctreeSceneManager.h"  
#include <OgreOctree.h>
```

Include dependency graph for OgreOctreeHierarchyInterface.cpp:



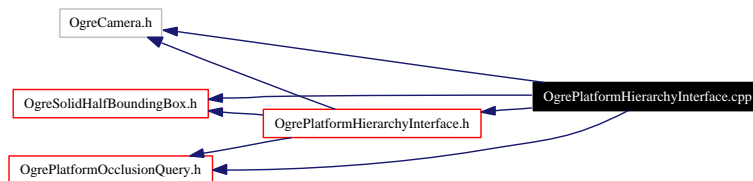
Namespaces

- namespace [Ogre](#)

8.59 Ogre/src/OgrePlatformHierarchyInterface.cpp File Reference

```
#include <OgreCamera.h>
#include "OgreSolidHalfBoundingBox.h"
#include "OgrePlatformHierarchyInterface.h"
#include "OgrePlatformOcclusionQuery.h"
```

Include dependency graph for OgrePlatformHierarchyInterface.cpp:



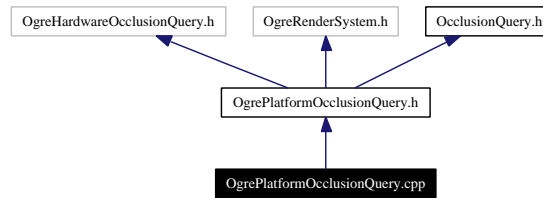
Namespaces

- namespace [Ogre](#)

8.60 Ogre/src/OgrePlatformOcclusionQuery.cpp File Reference

```
#include "OgrePlatformOcclusionQuery.h"
```

Include dependency graph for OgrePlatformOcclusionQuery.cpp:



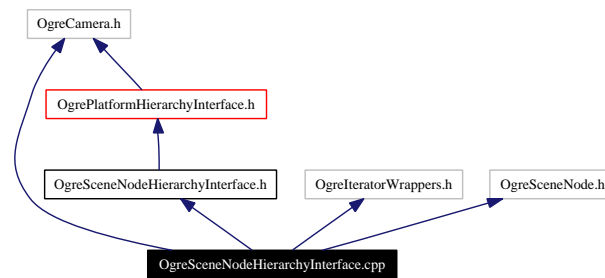
Namespaces

- namespace [Ogre](#)

8.61 Ogre/src/OgreSceneNodeHierarchyInterface.cpp File Reference

```
#include "OgreSceneNodeHierarchyInterface.h"  
#include <OgreIteratorWrappers.h>  
#include <OgreCamera.h>  
#include <OgreSceneNode.h>
```

Include dependency graph for OgreSceneNodeHierarchyInterface.cpp:



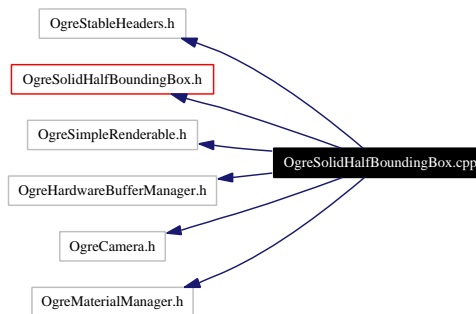
Namespaces

- namespace [Ogre](#)

8.62 Ogre/src/OgreSolidHalfBoundingBox.cpp File Reference

```
#include "OgreStableHeaders.h"  
#include "OgreSolidHalfBoundingBox.h"  
#include "OgreSimpleRenderable.h"  
#include "OgreHardwareBufferManager.h"  
#include "OgreCamera.h"  
#include "OgreMaterialManager.h"
```

Include dependency graph for OgreSolidHalfBoundingBox.cpp:



Namespaces

- namespace [Ogre](#)

Defines

- #define [POSITION_BINDING](#) 0

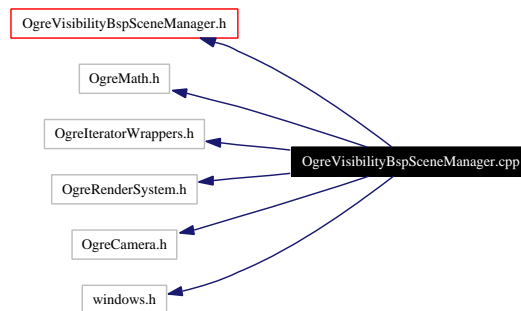
8.62.1 Define Documentation

8.62.1.1 #define [POSITION_BINDING](#) 0

8.63 Ogre/src/OgreVisibilityBspSceneManager.cpp File Reference

```
#include "OgreVisibilityBspSceneManager.h"  
#include <OgreMath.h>  
#include <OgreIteratorWrappers.h>  
#include <OgreRenderSystem.h>  
#include <OgreCamera.h>  
#include <windows.h>
```

Include dependency graph for `OgreVisibilityBspSceneManager.cpp`:



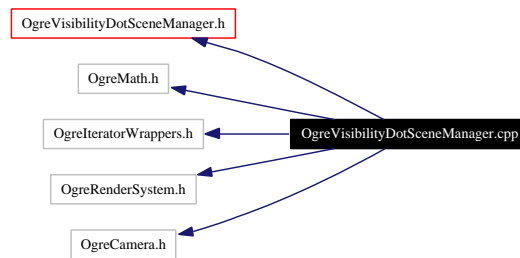
Namespaces

- namespace [Ogre](#)

8.64 Ogre/src/OgreVisibilityDotSceneManager.cpp File Reference

```
#include <OgreVisibilityDotSceneManager.h>
#include <OgreMath.h>
#include <OgreIteratorWrappers.h>
#include <OgreRenderSystem.h>
#include <OgreCamera.h>
```

Include dependency graph for OgreVisibilityDotSceneManager.cpp:



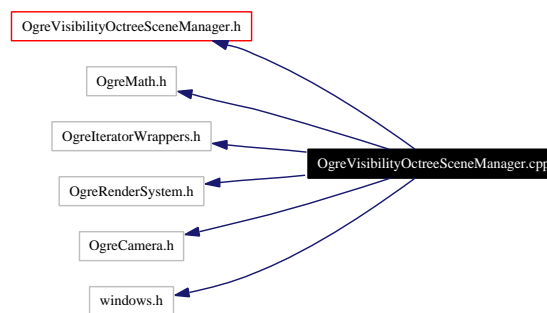
Namespaces

- namespace [Ogre](#)

8.65 Ogre/src/OgreVisibilityOctreeSceneManager.cpp File Reference

```
#include "OgreVisibilityOctreeSceneManager.h"  
#include <OgreMath.h>  
#include <OgreIteratorWrappers.h>  
#include <OgreRenderSystem.h>  
#include <OgreCamera.h>  
#include <windows.h>
```

Include dependency graph for OgreVisibilityOctreeSceneManager.cpp:



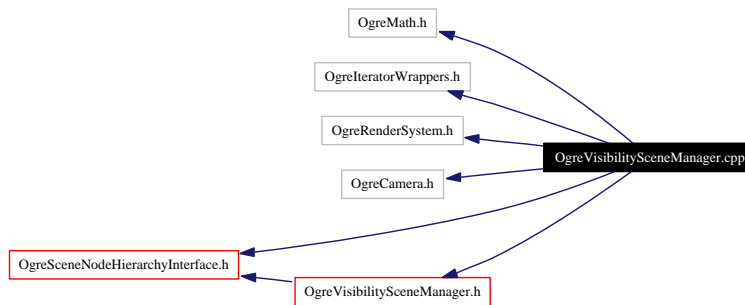
Namespaces

- namespace [Ogre](#)

8.66 Ogre/src/OgreVisibilitySceneManager.cpp File Reference

```
#include <OgreMath.h>
#include <OgreIteratorWrappers.h>
#include <OgreRenderSystem.h>
#include <OgreCamera.h>
#include "OgreVisibilitySceneManager.h"
#include "OgreSceneNodeHierarchyInterface.h"
```

Include dependency graph for OgreVisibilitySceneManager.cpp:



Namespaces

- namespace [Ogre](#)

8.67 Ogre/src/OgreVisibilitySceneManagerDll.cpp File Reference

```
#include <OgreOcclusionCullingTerrainSceneManager.h>
#include <OgreOcclusionCullingOctreeSceneManager.h>
#include <OgreHeightmapTerrainPageSource.h>
#include "GtpVisibilityManager.h"
#include "GtpVisibilityStopAndWaitManager.h"
#include "GtpVisibilityEnvironment.h"
#include <OgreRoot.h>
```

Include dependency graph for OgreVisibilitySceneManagerDll.cpp:



Namespaces

- namespace [Ogre](#)

Variables

- [GtpVisibility::Environment](#) * [visEnv](#)
- [GtpVisibility::Manager](#) * [visManager](#)

8.67.1 Variable Documentation

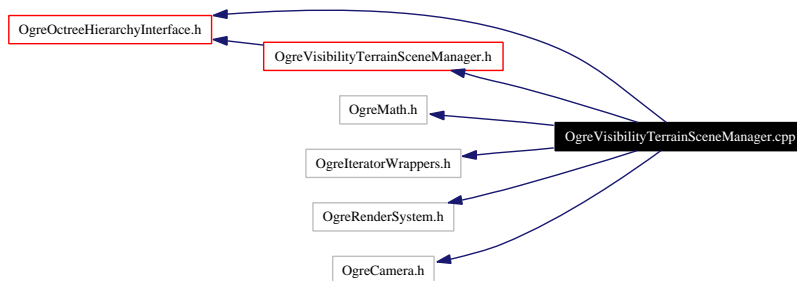
8.67.1.1 [GtpVisibility::Environment](#)* [visEnv](#)

8.67.1.2 [GtpVisibility::Manager](#)* [visManager](#)

8.68 Ogre/src/OgreVisibilityTerrainSceneManager.cpp File Reference

```
#include "OgreVisibilityTerrainSceneManager.h"  
#include "OgreOctreeHierarchyInterface.h"  
#include <OgreMath.h>  
#include <OgreIteratorWrappers.h>  
#include <OgreRenderSystem.h>  
#include <OgreCamera.h>
```

Include dependency graph for OgreVisibilityTerrainSceneManager.cpp:



Namespaces

- namespace [Ogre](#)

Index

- ~HierarchyInterface
 - GtpVisibility::HierarchyInterface, 41
- ~OcclusionQuery
 - GtpVisibility::OcclusionQuery, 58
- ~PlatformHierarchyInterface
 - Ogre::PlatformHierarchyInterface, 68
- ~PlatformOcclusionQuery
 - Ogre::PlatformOcclusionQuery, 73
- ~PreprocessingManager
 - GtpVisibility::PreprocessingManager, 75
- ~VisibilityBspSceneManager
 - Ogre::VisibilityBspSceneManager, 98
- ~VisibilityDotSceneManager
 - Ogre::VisibilityDotSceneManager, 101
- ~VisibilityManager
 - GtpVisibility::VisibilityManager, 104
- ~VisibilityOctreeSceneManager
 - Ogre::VisibilityOctreeSceneManager, 107
- ~VisibilitySceneManager
 - Ogre::VisibilitySceneManager, 109
- ~VisibilityTerrainSceneManager
 - Ogre::VisibilityTerrainSceneManager, 112
- ._findVisibleObjects
 - Ogre::VisibilityBspSceneManager, 98
 - Ogre::VisibilityDotSceneManager, 101
 - Ogre::VisibilityOctreeSceneManager, 107
 - Ogre::VisibilitySceneManager, 109
 - Ogre::VisibilityTerrainSceneManager, 112
- ._renderVisibleObjects
 - Ogre::VisibilityBspSceneManager, 98
 - Ogre::VisibilityDotSceneManager, 101
 - Ogre::VisibilityOctreeSceneManager, 107
 - Ogre::VisibilitySceneManager, 109
 - Ogre::VisibilityTerrainSceneManager, 112
- ._updateSceneGraph
 - Ogre::VisibilityBspSceneManager, 98
 - Ogre::VisibilityDotSceneManager, 101
 - Ogre::VisibilityOctreeSceneManager, 107
 - Ogre::VisibilitySceneManager, 110
 - Ogre::VisibilityTerrainSceneManager, 112
- AddViewCellPVS
 - GtpVisibility::DummyPreprocessing-
Manager, 34
 - GtpVisibility::PreprocessingManager, 76
- ApplyVisibilityCulling
 - GtpVisibility::VisibilityManager, 105
- axis
 - GtpVisibilityPreprocessor::KdInterior, 47
- AxisAlignedBox
 - GtpVisibility, 12
- BeginQuery
 - GtpVisibility::OcclusionQuery, 58
 - Ogre::PlatformOcclusionQuery, 73
- BspHierarchyInterface
 - Ogre::BspHierarchyInterface, 19
- BSPLeaf
 - GtpVisibilityPreprocessor::BSPLeaf, 24
- BSPTree
 - GtpVisibilityPreprocessor::BSPTree, 28
- BuildKdTree
 - GtpVisibilityPreprocessor::Preprocessor,
80
- Camera
 - GtpVisibility, 12
- CheckFrustumVisible
 - GtpVisibility::HierarchyInterface, 43
 - Ogre::PlatformHierarchyInterface, 69
- COHERENT_HIERARCHICAL_CULLING
 - GtpVisibility::VisibilityEnvironment, 103
- CoherentHierarchicalCullingManager
 - GtpVisibility::CoherentHierarchical-
CullingManager, 29
- ComputeCameraVisibility
 - GtpVisibility::DummyQueryManager, 36
 - GtpVisibility::QueryManager, 83
- ComputeFromPointVisibility
 - GtpVisibility::DummyQueryManager, 36
 - GtpVisibility::QueryManager, 83
- ComputeVisibility
 - GtpVisibilityPreprocessor::Exact-
Preprocessor, 37
 - GtpVisibilityPreprocessor::Preprocessor,
80
 - GtpVisibilityPreprocessor::Sampling-
Preprocessor, 85
- CullingManager
 - GtpVisibility::CullingManager, 31

- CullingManagerType
 - GtpVisibility::VisibilityEnvironment, 103
- cullingOctreePlugin
 - Ogre, 14
- cullingTerrainPlugin
 - Ogre, 14
- DeleteQueries
 - Ogre::PlatformHierarchyInterface, 70
- DistanceQueue
 - GtpVisibility, 12
- dllStartPlugin
 - Ogre, 14
- dllStopPlugin
 - Ogre, 14
- DummyPreprocessingManager
 - GtpVisibility::DummyPreprocessing-Manager, 33
- DummyQueryManager
 - GtpVisibility::DummyQueryManager, 35
- EndQuery
 - GtpVisibility::OcclusionQuery, 59
 - Ogre::PlatformOcclusionQuery, 73
- ExportPreprocessedData
 - GtpVisibilityPreprocessor::Preprocessor, 79
- ExportScene
 - GtpVisibility::DummyPreprocessing-Manager, 34
 - GtpVisibility::PreprocessingManager, 75
- FRUSTUM_CULLING
 - GtpVisibility::VisibilityEnvironment, 103
- FrustumCullingManager
 - GtpVisibility::FrustumCullingManager, 38
- GenerateViewcells
 - GtpVisibilityPreprocessor::Preprocessor, 79
- GetBoundingBox
 - Ogre::BspHierarchyInterface, 21
 - Ogre::OctreeHierarchyInterface, 63
 - Ogre::PlatformHierarchyInterface, 69
 - Ogre::SceneNodeHierarchyInterface, 92
- GetCullingManager
 - GtpVisibility::VisibilityManager, 105
- GetFrameId
 - GtpVisibility::HierarchyInterface, 42
- GetNextOcclusionQuery
 - GtpVisibility::HierarchyInterface, 43
 - Ogre::PlatformHierarchyInterface, 68
- GetOccludeeChecksum
 - GtpVisibility::PreprocessingManager, 76
- GetOccluderChecksum
 - GtpVisibility::PreprocessingManager, 76
- getOption
 - Ogre::VisibilityBspSceneManager, 98
 - Ogre::VisibilityDotSceneManager, 101
 - Ogre::VisibilityOctreeSceneManager, 107
 - Ogre::VisibilitySceneManager, 110
 - Ogre::VisibilityTerrainSceneManager, 112
- getOptionKeys
 - Ogre::VisibilityBspSceneManager, 98
 - Ogre::VisibilityDotSceneManager, 101
 - Ogre::VisibilityOctreeSceneManager, 107
 - Ogre::VisibilitySceneManager, 110
 - Ogre::VisibilityTerrainSceneManager, 112
- getOptionValues
 - Ogre::VisibilityBspSceneManager, 98
 - Ogre::VisibilityDotSceneManager, 101
 - Ogre::VisibilityOctreeSceneManager, 107
 - Ogre::VisibilitySceneManager, 110
 - Ogre::VisibilityTerrainSceneManager, 112
- GetPVS
 - GtpVisibility::PreprocessingManager, 75
- GetQueryResult
 - GtpVisibility::OcclusionQuery, 58
 - Ogre::PlatformOcclusionQuery, 73
- GetQueue
 - GtpVisibility::HierarchyInterface, 43
- GetRoot
 - GtpVisibilityPreprocessor::KdTree, 53
- GetSceneRoot
 - GtpVisibility::HierarchyInterface, 42
- GetSolidHalfBoundingBox
 - Ogre::PlatformHierarchyInterface, 70
- GetSquaredViewDepth
 - Ogre::BspHierarchyInterface, 21
 - Ogre::OctreeHierarchyInterface, 63
- getVisibilityManager
 - Ogre::VisibilityBspSceneManager, 98
 - Ogre::VisibilityDotSceneManager, 101
 - Ogre::VisibilityOctreeSceneManager, 107
 - Ogre::VisibilitySceneManager, 110
 - Ogre::VisibilityTerrainSceneManager, 112
- getWorldTransforms
 - Ogre::SolidHalfBoundingBox, 93
- GreaterDistance
 - GtpVisibility::GreaterDistance, 39
- GtpVisibility, 11
- GtpVisibility
 - AxisAlignedBox, 12
 - Camera, 12
 - DistanceQueue, 12
 - HierarchyNode, 12
 - Mesh, 12
 - QueryPair, 12

- QueryQueue, 12
- Ray, 12
- Vector3, 12
- GtpVisibility/include/CoherentHierarchicalCullingManager.h, 115
- GtpVisibility/include/CullingManager.h, 116
- GtpVisibility/include/DistanceQueue.h, 117
- GtpVisibility/include/DummyPreprocessingManager.h, 118
- GtpVisibility/include/DummyQueryManager.h, 119
- GtpVisibility/include/FrustumCullingManager.h, 120
- GtpVisibility/include/HierarchyInterface.h, 121
- GtpVisibility/include/OcclusionQuery.h, 122
- GtpVisibility/include/PreprocessingManager.h, 123
- GtpVisibility/include/QueryManager.h, 124
- GtpVisibility/include/StopAndWaitCullingManager.h, 125
- GtpVisibility/include/VisibilityAxisAlignedBox.h, 126
- GtpVisibility/include/VisibilityCamera.h, 127
- GtpVisibility/include/VisibilityEnvironment.h, 128
- GtpVisibility/include/VisibilityInfo.h, 129
- GtpVisibility/include/VisibilityManager.h, 130
- GtpVisibility/include/VisibilityMesh.h, 131
- GtpVisibility/include/VisibilityRay.h, 132
- GtpVisibility/include/VisibilityVector3.h, 133
- GtpVisibility/src/CoherentHierarchicalCullingManager.cpp, 134
- GtpVisibility/src/CullingManager.cpp, 135
- GtpVisibility/src/DummyPreprocessingManager.cpp, 136
- GtpVisibility/src/DummyQueryManager.cpp, 137
- GtpVisibility/src/FrustumCullingManager.cpp, 138
- GtpVisibility/src/HierarchyInterface.cpp, 139
- GtpVisibility/src/PreprocessingManager.cpp, 140
- GtpVisibility/src/QueryManager.cpp, 141
- GtpVisibility/src/StopAndWaitCullingManager.cpp, 142
- GtpVisibility/src/VisibilityEnvironment.cpp, 143
- GtpVisibility/src/VisibilityManager.cpp, 144
- GtpVisibility::CoherentHierarchicalCullingManager, 29
- GtpVisibility::CoherentHierarchicalCulling-Manager
 - CoherentHierarchicalCullingManager, 29
 - RenderScene, 29
- GtpVisibility::CullingManager, 31
- GtpVisibility::CullingManager
 - CullingManager, 31
 - mHierarchyInterface, 32
 - mNumFrustumCulledNodes, 32
 - mNumQueryCulledNodes, 32
 - mVisibilityThreshold, 32
 - RenderScene, 32
 - SetHierarchyInterface, 32
- GtpVisibility::DummyPreprocessingManager, 33
- GtpVisibility::DummyPreprocessingManager
 - AddViewCellPVS, 34
 - DummyPreprocessingManager, 33
 - ExportScene, 34
 - LoadPreprocessedData, 34
 - LocateViewCellIds, 34
- GtpVisibility::DummyQueryManager, 35
- GtpVisibility::DummyQueryManager
 - ComputeCameraVisibility, 36
 - ComputeFromPointVisibility, 36
 - DummyQueryManager, 35
- GtpVisibility::FrustumCullingManager, 38
- GtpVisibility::FrustumCullingManager
 - FrustumCullingManager, 38
 - RenderScene, 38
- GtpVisibility::GreaterDistance, 39
- GtpVisibility::GreaterDistance
 - GreaterDistance, 39
 - mHierarchyInterface, 39
 - operator(), 39
- GtpVisibility::HierarchyInterface, 40
- GtpVisibility::HierarchyInterface
 - ~HierarchyInterface, 41
 - CheckFrustumVisible, 43
 - GetFrameId, 42
 - GetNextOcclusionQuery, 43
 - GetQueue, 43
 - GetSceneRoot, 42
 - HasGeometry, 43
 - HasGreaterDistance, 43
 - HierarchyInterface, 41
 - InitFrame, 42
 - IsLeaf, 41
 - IsNodeVisible, 44
 - IssueOcclusionQuery, 42
 - LastVisited, 44
 - mCurrentTestIdx, 45
 - mDistanceQueue, 45
 - mFrameId, 45
 - mNumRenderedGeometry, 45
 - mNumRenderedNodes, 45
 - mNumSceneNodes, 45
 - mNumTraversedNodes, 45

- mSceneRoot, 45
- PullUpVisibility, 41
- RenderNode, 41
- SetLastVisited, 44
- SetNodeVisible, 44
- SetSceneRoot, 42
- TraverseNode, 41
- GtpVisibility::MeshInfo, 56
- GtpVisibility::MeshInfo
 - MeshInfo, 56
 - mMesh, 56
 - mVisibility, 56
- GtpVisibility::NodeInfo, 57
- GtpVisibility::NodeInfo
 - mNode, 57
 - mVisibility, 57
 - NodeInfo, 57
- GtpVisibility::OcclusionQuery, 58
- GtpVisibility::OcclusionQuery
 - ~OcclusionQuery, 58
 - BeginQuery, 58
 - EndQuery, 59
 - GetQueryResult, 58
 - ResultAvailable, 58
- GtpVisibility::PreprocessingManager, 74
- GtpVisibility::PreprocessingManager
 - ~PreprocessingManager, 75
 - AddViewCellPVS, 76
 - ExportScene, 75
 - GetOccludeeChecksum, 76
 - GetOccluderChecksum, 76
 - GetPVS, 75
 - LoadPreprocessedData, 75
 - LocateViewCellIds, 76
 - mSceneTraverser, 77
 - PreprocessingManager, 75
 - SetSceneTraverser, 76
- GtpVisibility::QueryManager, 82
- GtpVisibility::QueryManager
 - ComputeCameraVisibility, 83
 - ComputeFromPointVisibility, 83
 - mHierarchyInterface, 84
 - QueryManager, 82
 - SetSceneTraverser, 83
 - ShootRay, 83
- GtpVisibility::StopAndWaitCullingManager, 95
- GtpVisibility::StopAndWaitCullingManager
 - RenderScene, 95
 - StopAndWaitCullingManager, 95
- GtpVisibility::VisibilityEnvironment, 103
 - COHERENT_HIERARCHICAL_-CULLING, 103
 - FRUSTUM_CULLING, 103
 - STOP_AND_WAIT, 103
- GtpVisibility::VisibilityEnvironment
 - CullingManagerType, 103
 - LoadEnvironment, 103
 - VisibilityEnvironment, 103
- GtpVisibility::VisibilityManager, 104
- GtpVisibility::VisibilityManager
 - ~VisibilityManager, 104
 - ApplyVisibilityCulling, 105
 - GetCullingManager, 105
 - mCullingManager, 105
 - mCullingManagerType, 105
 - mPreprocessingManager, 105
 - mQueryManager, 105
 - mVisibilityEnvironment, 105
 - SetCullingManager, 105
 - VisibilityManager, 104
- GtpVisibilityPreprocessor, 13
- GtpVisibilityPreprocessor
 - MeshContainer, 13
 - NodeContainer, 13
 - ViewCellContainer, 13
- GtpVisibilityPreprocessor/include/AxisAlignedBox3.h, 145
- GtpVisibilityPreprocessor/include/Containers.h, 146
- GtpVisibilityPreprocessor/include/ExactPreprocessor.h, 147
- GtpVisibilityPreprocessor/include/KdTree.h, 148
- GtpVisibilityPreprocessor/include/Mesh.h, 149
- GtpVisibilityPreprocessor/include/Plane3.h, 150
- GtpVisibilityPreprocessor/include/Preprocessor.h, 151
- GtpVisibilityPreprocessor/include/SamplingPreprocessor.h, 152
- GtpVisibilityPreprocessor/include/SceneGraph.h, 153
- GtpVisibilityPreprocessor/include/Vector3.h, 154
- GtpVisibilityPreprocessor/include/ViewCellBSP.h, 155
- GtpVisibilityPreprocessor/src/ExactPreprocessor.cpp, 156
- GtpVisibilityPreprocessor/src/KdTree.cpp, 157
- GtpVisibilityPreprocessor/src/Preprocessor.cpp, 158
- GtpVisibilityPreprocessor/src/SamplingPreprocessor.cpp, 159
- GtpVisibilityPreprocessor::AxisAlignedBox3, 17
- GtpVisibilityPreprocessor::AxisAlignedBox3
 - max, 17
 - min, 17
- GtpVisibilityPreprocessor::BSPInterior, 22

- GtpVisibilityPreprocessor::BSPInterior
 - IsLeaf, 23
 - mBack, 23
 - mFront, 23
 - mPlane, 23
- GtpVisibilityPreprocessor::BSPLeaf, 24
- GtpVisibilityPreprocessor::BSPLeaf
 - BSPLeaf, 24
 - IsLeaf, 24
 - mViewCell, 25
- GtpVisibilityPreprocessor::BSPNode, 26
- GtpVisibilityPreprocessor::BSPNode
 - IsLeaf, 26
 - IsRoot, 26
 - parent, 27
- GtpVisibilityPreprocessor::BSPTree, 28
- GtpVisibilityPreprocessor::BSPTree
 - BSPTree, 28
 - mRoot, 28
 - mRootCell, 28
- GtpVisibilityPreprocessor::ExactPreprocessor, 37
- GtpVisibilityPreprocessor::ExactPreprocessor
 - ComputeVisibility, 37
- GtpVisibilityPreprocessor::KdInterior, 46
- GtpVisibilityPreprocessor::KdInterior
 - axis, 47
 - IsLeaf, 46
 - mBack, 47
 - mFront, 47
 - mPosition, 47
- GtpVisibilityPreprocessor::KdLeaf, 48
- GtpVisibilityPreprocessor::KdLeaf
 - IsLeaf, 48
 - mOccludees, 48
 - mOccluders, 48
 - mViewCells, 49
- GtpVisibilityPreprocessor::KdNode, 50
- GtpVisibilityPreprocessor::KdNode
 - IsLeaf, 50
 - IsRoot, 50
 - mParent, 51
- GtpVisibilityPreprocessor::KdTree, 52
- GtpVisibilityPreprocessor::KdTree
 - GetRoot, 53
 - InsertOccludee, 52
 - InsertOccluder, 52
 - InsertViewCell, 52
 - mBox, 53
 - mRoot, 53
 - Subdivide, 53
- GtpVisibilityPreprocessor::Mesh, 54
- GtpVisibilityPreprocessor::Mesh
 - Mesh, 54
 - mPatches, 55
 - mVertices, 55
 - PatchContainer, 54
 - VertexContainer, 54
- GtpVisibilityPreprocessor::Patch, 65
- GtpVisibilityPreprocessor::Patch
 - mVertices, 65
 - Patch, 65
- GtpVisibilityPreprocessor::Plane3, 66
- GtpVisibilityPreprocessor::Plane3
 - mD, 66
 - mNormal, 66
- GtpVisibilityPreprocessor::Preprocessor, 78
- GtpVisibilityPreprocessor::Preprocessor
 - BuildKdTree, 80
 - ComputeVisibility, 80
 - ExportPreprocessedData, 79
 - GenerateViewcells, 79
 - LoadScene, 79
 - LoadViewcells, 79
 - mKdTree, 80
 - mOccludees, 80
 - mOccluders, 80
 - mSceneGraph, 80
 - mViewCellBSPTree, 80
 - mViewcells, 80
- GtpVisibilityPreprocessor::SamplingPreprocessor, 85
- GtpVisibilityPreprocessor::Sampling-Preprocessor
 - ComputeVisibility, 85
- GtpVisibilityPreprocessor::SceneGraph, 87
- GtpVisibilityPreprocessor::SceneGraph
 - mRoot, 87
- GtpVisibilityPreprocessor::SceneGraphNode, 88
- GtpVisibilityPreprocessor::SceneGraphNode
 - mBox, 88
 - mChildren, 88
 - mGeometry, 88
- GtpVisibilityPreprocessor::Vector3, 96
- GtpVisibilityPreprocessor::Vector3
 - x, 96
 - y, 96
 - z, 96
- HasGeometry
 - GtpVisibility::HierarchyInterface, 43
 - Ogre::BspHierarchyInterface, 20
 - Ogre::OctreeHierarchyInterface, 62
 - Ogre::SceneNodeHierarchyInterface, 91
- HasGreaterDistance
 - GtpVisibility::HierarchyInterface, 43
 - Ogre::BspHierarchyInterface, 20
 - Ogre::OctreeHierarchyInterface, 62

- Ogre::SceneNodeHierarchyInterface, 91
- heightmapTerrainPageSource
 - Ogre, 14
- HierarchyInterface
 - GtpVisibility::HierarchyInterface, 41
- HierarchyNode
 - GtpVisibility, 12
- InfoContainer
 - VisibilityInfo.h, 129
- InitFrame
 - GtpVisibility::HierarchyInterface, 42
 - Ogre::PlatformHierarchyInterface, 68
- InsertOccludee
 - GtpVisibilityPreprocessor::KdTree, 52
- InsertOccluder
 - GtpVisibilityPreprocessor::KdTree, 52
- InsertViewCell
 - GtpVisibilityPreprocessor::KdTree, 52
- IsLeaf
 - GtpVisibility::HierarchyInterface, 41
 - GtpVisibilityPreprocessor::BSPInterior, 23
 - GtpVisibilityPreprocessor::BSPLeaf, 24
 - GtpVisibilityPreprocessor::BSPNode, 26
 - GtpVisibilityPreprocessor::KdInterior, 46
 - GtpVisibilityPreprocessor::KdLeaf, 48
 - GtpVisibilityPreprocessor::KdNode, 50
 - Ogre::BspHierarchyInterface, 19
 - Ogre::OctreeHierarchyInterface, 61
 - Ogre::SceneNodeHierarchyInterface, 90
- IsNodeVisible
 - GtpVisibility::HierarchyInterface, 44
 - Ogre::BspHierarchyInterface, 20
 - Ogre::OctreeHierarchyInterface, 63
 - Ogre::SceneNodeHierarchyInterface, 91
- IsRoot
 - GtpVisibilityPreprocessor::BSPNode, 26
 - GtpVisibilityPreprocessor::KdNode, 50
- IssueOcclusionQuery
 - GtpVisibility::HierarchyInterface, 42
 - Ogre::PlatformHierarchyInterface, 69
- LastVisited
 - GtpVisibility::HierarchyInterface, 44
 - Ogre::BspHierarchyInterface, 21
 - Ogre::OctreeHierarchyInterface, 63
 - Ogre::SceneNodeHierarchyInterface, 92
- LoadEnvironment
 - GtpVisibility::VisibilityEnvironment, 103
- LoadPreprocessedData
 - GtpVisibility::DummyPreprocessing-Manager, 34
 - GtpVisibility::PreprocessingManager, 75
- LoadScene
 - GtpVisibilityPreprocessor::Preprocessor, 79
- LoadViewcells
 - GtpVisibilityPreprocessor::Preprocessor, 79
- LocateViewCellIds
 - GtpVisibility::DummyPreprocessing-Manager, 34
 - GtpVisibility::PreprocessingManager, 76
- main
 - Preprocessor.cpp, 158
- max
 - GtpVisibilityPreprocessor::AxisAligned-Box3, 17
- mBack
 - GtpVisibilityPreprocessor::BSPInterior, 23
 - GtpVisibilityPreprocessor::KdInterior, 47
- mBox
 - GtpVisibilityPreprocessor::KdTree, 53
 - GtpVisibilityPreprocessor::SceneGraph-Node, 88
 - Ogre::PlatformHierarchyInterface, 71
- mCamera
 - Ogre::PlatformHierarchyInterface, 71
- mChildren
 - GtpVisibilityPreprocessor::SceneGraph-Node, 88
- mCullingManager
 - GtpVisibility::VisibilityManager, 105
- mCullingManagerType
 - GtpVisibility::VisibilityManager, 105
- mCurrentTestIdx
 - GtpVisibility::HierarchyInterface, 45
- mD
 - GtpVisibilityPreprocessor::Plane3, 66
- mDistanceQueue
 - GtpVisibility::HierarchyInterface, 45
- Mesh
 - GtpVisibility, 12
 - GtpVisibilityPreprocessor::Mesh, 54
- MeshContainer
 - GtpVisibilityPreprocessor, 13
- MeshInfo
 - GtpVisibility::MeshInfo, 56
- mFrameId
 - GtpVisibility::HierarchyInterface, 45
- mFront
 - GtpVisibilityPreprocessor::BSPInterior, 23
 - GtpVisibilityPreprocessor::KdInterior, 47
- mGeometry
 - GtpVisibilityPreprocessor::SceneGraph-Node, 88
- mHalfBoundingBox

- Ogre::PlatformHierarchyInterface, 70
- mHardwareOcclusionQuery
 - Ogre::PlatformOcclusionQuery, 73
- mHierarchyInterface
 - GtpVisibility::CullingManager, 32
 - GtpVisibility::GreaterDistance, 39
 - GtpVisibility::QueryManager, 84
 - Ogre::VisibilityBspSceneManager, 99
 - Ogre::VisibilityDotSceneManager, 102
 - Ogre::VisibilityOctreeSceneManager, 108
 - Ogre::VisibilitySceneManager, 110
 - Ogre::VisibilityTerrainSceneManager, 113
- min
 - GtpVisibilityPreprocessor::AxisAligned-Box3, 17
- mIsFirstHalf
 - Ogre::SolidHalfBoundingBox, 94
- mKdTree
 - GtpVisibilityPreprocessor::Preprocessor, 80
- mMesh
 - GtpVisibility::MeshInfo, 56
- mNode
 - GtpVisibility::NodeInfo, 57
- mNormal
 - GtpVisibilityPreprocessor::Plane3, 66
- mNumFrustumCulledNodes
 - GtpVisibility::CullingManager, 32
- mNumOctreeNodes
 - Ogre::BspHierarchyInterface, 21
 - Ogre::OctreeHierarchyInterface, 63
- mNumQueryCulledNodes
 - GtpVisibility::CullingManager, 32
- mNumRenderedGeometry
 - GtpVisibility::HierarchyInterface, 45
- mNumRenderedNodes
 - GtpVisibility::HierarchyInterface, 45
- mNumSceneNodes
 - GtpVisibility::HierarchyInterface, 45
- mNumTraversedNodes
 - GtpVisibility::HierarchyInterface, 45
- mOccludees
 - GtpVisibilityPreprocessor::KdLeaf, 48
 - GtpVisibilityPreprocessor::Preprocessor, 80
- mOccluders
 - GtpVisibilityPreprocessor::KdLeaf, 48
 - GtpVisibilityPreprocessor::Preprocessor, 80
- mOcclusionQueries
 - Ogre::PlatformHierarchyInterface, 71
- mParent
 - GtpVisibilityPreprocessor::KdNode, 51
- mPatches
 - GtpVisibilityPreprocessor::Mesh, 55
- mPlane
 - GtpVisibilityPreprocessor::BSPInterior, 23
- mPosition
 - GtpVisibilityPreprocessor::KdInterior, 47
- mPreprocessingManager
 - GtpVisibility::VisibilityManager, 105
- mQueryManager
 - GtpVisibility::VisibilityManager, 105
- mRenderSystem
 - Ogre::PlatformHierarchyInterface, 71
- mRoot
 - GtpVisibilityPreprocessor::BSPTree, 28
 - GtpVisibilityPreprocessor::KdTree, 53
 - GtpVisibilityPreprocessor::SceneGraph, 87
- mRootCell
 - GtpVisibilityPreprocessor::BSPTree, 28
- mSceneGraph
 - GtpVisibilityPreprocessor::Preprocessor, 80
- mSceneManager
 - Ogre::PlatformHierarchyInterface, 70
- mSceneRoot
 - GtpVisibility::HierarchyInterface, 45
- mSceneTraverser
 - GtpVisibility::PreprocessingManager, 77
- mVertices
 - GtpVisibilityPreprocessor::Mesh, 55
 - GtpVisibilityPreprocessor::Patch, 65
- mViewCell
 - GtpVisibilityPreprocessor::BSPLeaf, 25
- mViewCellBSPTree
 - GtpVisibilityPreprocessor::Preprocessor, 80
- mViewCells
 - GtpVisibilityPreprocessor::KdLeaf, 49
- mViewcells
 - GtpVisibilityPreprocessor::Preprocessor, 80
- mVisibility
 - GtpVisibility::MeshInfo, 56
 - GtpVisibility::NodeInfo, 57
- mVisibilityEnvironment
 - GtpVisibility::VisibilityManager, 105
- mVisibilityManager
 - Ogre::VisibilityBspSceneManager, 99
 - Ogre::VisibilityDotSceneManager, 102
 - Ogre::VisibilityOctreeSceneManager, 108
 - Ogre::VisibilitySceneManager, 110
 - Ogre::VisibilityTerrainSceneManager, 113
- mVisibilityThreshold
 - GtpVisibility::CullingManager, 32
- NodeContainer

- GtpVisibilityPreprocessor, 13
- NodeInfo
 - GtpVisibility::NodeInfo, 57
- OctreeHierarchyInterface
 - Ogre::OctreeHierarchyInterface, 61
- Ogre, 14
 - cullingOctreePlugin, 14
 - cullingTerrainPlugin, 14
 - dllStartPlugin, 14
 - dllStopPlugin, 14
 - heightmapTerrainPageSource, 14
- Ogre/include/OgreBspHierarchyInterface.h, 160
- Ogre/include/OgreOctreeHierarchyInterface.h, 161
- Ogre/include/OgrePlatformHierarchyInterface.h, 162
- Ogre/include/OgrePlatformOcclusionQuery.h, 163
- Ogre/include/OgreSceneNodeHierarchyInterface.h, 164
- Ogre/include/OgreSolidHalfBoundingBox.h, 165
- Ogre/include/OgreVisibilityBspSceneManager.h, 166
- Ogre/include/OgreVisibilityDotSceneManager.h, 167
- Ogre/include/OgreVisibilityOctreeSceneManager.h, 168
- Ogre/include/OgreVisibilitySceneManager.h, 169
- Ogre/include/OgreVisibilityTerrainSceneManager.h, 170
- Ogre/src/OgreBspHierarchyInterface.cpp, 171
- Ogre/src/OgreOctreeHierarchyInterface.cpp, 172
- Ogre/src/OgrePlatformHierarchyInterface.cpp, 173
- Ogre/src/OgrePlatformOcclusionQuery.cpp, 174
- Ogre/src/OgreSceneNodeHierarchyInterface.cpp, 175
- Ogre/src/OgreSolidHalfBoundingBox.cpp, 176
- Ogre/src/OgreVisibilityBspSceneManager.cpp, 177
- Ogre/src/OgreVisibilityDotSceneManager.cpp, 178
- Ogre/src/OgreVisibilityOctreeSceneManager.cpp, 179
- Ogre/src/OgreVisibilitySceneManager.cpp, 180
- Ogre/src/OgreVisibilitySceneManagerDll.cpp, 181
- Ogre/src/OgreVisibilityTerrainSceneManager.cpp, 182
- Ogre::BspHierarchyInterface, 18
- Ogre::BspHierarchyInterface
 - BspHierarchyInterface, 19
 - GetBoundingBox, 21
 - GetSquaredViewDepth, 21
 - HasGeometry, 20
 - HasGreaterDistance, 20
 - IsLeaf, 19
 - IsNodeVisible, 20
 - LastVisited, 21
 - mNumOctreeNode, 21
 - PullUpVisibility, 19
 - RenderNode, 19
 - SetLastVisited, 20
 - SetNodeVisible, 20
 - TraverseNode, 19
- Ogre::OctreeHierarchyInterface, 60
- Ogre::OctreeHierarchyInterface
 - GetBoundingBox, 63
 - GetSquaredViewDepth, 63
 - HasGeometry, 62
 - HasGreaterDistance, 62
 - IsLeaf, 61
 - IsNodeVisible, 63
 - LastVisited, 63
 - mNumOctreeNode, 63
 - OctreeHierarchyInterface, 61
 - PullUpVisibility, 61
 - RenderNode, 61
 - SetLastVisited, 62
 - SetNodeVisible, 62
 - SetNumOctreeNode, 61
 - TraverseNode, 61
- Ogre::PlatformHierarchyInterface, 67
- Ogre::PlatformHierarchyInterface
 - ~PlatformHierarchyInterface, 68
 - CheckFrustumVisible, 69
 - DeleteQueries, 70
 - GetBoundingBox, 69
 - GetNextOcclusionQuery, 68
 - GetSolidHalfBoundingBox, 70
 - InitFrame, 68
 - IssueOcclusionQuery, 69
 - mBox, 71
 - mCamera, 71
 - mHalfBoundingBox, 70
 - mOcclusionQueries, 71
 - mRenderSystem, 71
 - mSceneManager, 70
 - PlatformHierarchyInterface, 68
 - RenderBoundingBox, 70
 - SetCamera, 68
 - SetRenderSystem, 69
 - SetSceneManager, 69
- Ogre::PlatformOcclusionQuery, 72

- Ogre::PlatformOcclusionQuery
 - ~PlatformOcclusionQuery, 73
 - BeginQuery, 73
 - EndQuery, 73
 - GetQueryResult, 73
 - mHardwareOcclusionQuery, 73
 - PlatformOcclusionQuery, 73
 - ResultAvailable, 73
- Ogre::SceneNodeHierarchyInterface, 89
- Ogre::SceneNodeHierarchyInterface
 - GetBoundingBox, 92
 - HasGeometry, 91
 - HasGreaterDistance, 91
 - IsLeaf, 90
 - IsNodeVisible, 91
 - LastVisited, 92
 - PullUpVisibility, 90
 - RenderNode, 90
 - SceneNodeHierarchyInterface, 90
 - SetLastVisited, 91
 - SetNodeVisible, 91
 - TraverseNode, 90
- Ogre::SolidHalfBoundingBox, 93
- Ogre::SolidHalfBoundingBox
 - getWorldTransforms, 93
 - mIsFirstHalf, 94
 - setOcclusionQueryMaterial, 93
 - setupBoundingBox, 93
 - setupBoundingBoxVertices, 93
 - SolidHalfBoundingBox, 93
- Ogre::VisibilityBspSceneManager, 97
- Ogre::VisibilityBspSceneManager
 - ~VisibilityBspSceneManager, 98
 - _findVisibleObjects, 98
 - _renderVisibleObjects, 98
 - _updateSceneGraph, 98
 - getOption, 98
 - getOptionKeys, 98
 - getOptionValues, 98
 - getVisibilityManager, 98
 - mHierarchyInterface, 99
 - mVisibilityManager, 99
 - setOption, 98
 - setVisibilityManager, 98
 - VisibilityBspSceneManager, 98
- Ogre::VisibilityDotSceneManager, 100
- Ogre::VisibilityDotSceneManager
 - ~VisibilityDotSceneManager, 101
 - _findVisibleObjects, 101
 - _renderVisibleObjects, 101
 - _updateSceneGraph, 101
 - getOption, 101
 - getOptionKeys, 101
 - getOptionValues, 101
 - getVisibilityManager, 101
 - mHierarchyInterface, 102
 - mVisibilityManager, 102
 - setOption, 101
 - setVisibilityManager, 101
 - VisibilityDotSceneManager, 101
- Ogre::VisibilityOctreeSceneManager, 106
- Ogre::VisibilityOctreeSceneManager
 - ~VisibilityOctreeSceneManager, 107
 - _findVisibleObjects, 107
 - _renderVisibleObjects, 107
 - _updateSceneGraph, 107
 - getOption, 107
 - getOptionKeys, 107
 - getOptionValues, 107
 - getVisibilityManager, 107
 - mHierarchyInterface, 108
 - mVisibilityManager, 108
 - setOption, 107
 - setVisibilityManager, 107
 - VisibilityOctreeSceneManager, 107
- Ogre::VisibilitySceneManager, 109
- Ogre::VisibilitySceneManager
 - ~VisibilitySceneManager, 109
 - _findVisibleObjects, 109
 - _renderVisibleObjects, 109
 - _updateSceneGraph, 110
 - getOption, 110
 - getOptionKeys, 110
 - getOptionValues, 110
 - getVisibilityManager, 110
 - mHierarchyInterface, 110
 - mVisibilityManager, 110
 - setOption, 110
 - setVisibilityManager, 110
 - VisibilitySceneManager, 109
- Ogre::VisibilityTerrainSceneManager, 111
- Ogre::VisibilityTerrainSceneManager
 - ~VisibilityTerrainSceneManager, 112
 - _findVisibleObjects, 112
 - _renderVisibleObjects, 112
 - _updateSceneGraph, 112
 - getOption, 112
 - getOptionKeys, 112
 - getOptionValues, 112
 - getVisibilityManager, 112
 - mHierarchyInterface, 113
 - mVisibilityManager, 113
 - setOption, 112
 - setVisibilityManager, 112
 - VisibilityTerrainSceneManager, 112
- OgreSolidHalfBoundingBox.cpp
 - POSITION_BINDING, 176
- OgreVisibilitySceneManagerDll.cpp

- visEnv, 181
- visManager, 181
- operator()
 - GtpVisibility::GreaterDistance, 39
- parent
 - GtpVisibilityPreprocessor::BSPNode, 27
- Patch
 - GtpVisibilityPreprocessor::Patch, 65
- PatchContainer
 - GtpVisibilityPreprocessor::Mesh, 54
- PlatformHierarchyInterface
 - Ogre::PlatformHierarchyInterface, 68
- PlatformOcclusionQuery
 - Ogre::PlatformOcclusionQuery, 73
- POSITION_BINDING
 - OgreSolidHalfBoundingBox.cpp, 176
- PreprocessingManager
 - GtpVisibility::PreprocessingManager, 75
- Preprocessor.cpp
 - main, 158
- PullUpVisibility
 - GtpVisibility::HierarchyInterface, 41
 - Ogre::BspHierarchyInterface, 19
 - Ogre::OctreeHierarchyInterface, 61
 - Ogre::SceneNodeHierarchyInterface, 90
- QueryManager
 - GtpVisibility::QueryManager, 82
- QueryPair
 - GtpVisibility, 12
- QueryQueue
 - GtpVisibility, 12
- Ray
 - GtpVisibility, 12
- RenderBoundingBox
 - Ogre::PlatformHierarchyInterface, 70
- RenderNode
 - GtpVisibility::HierarchyInterface, 41
 - Ogre::BspHierarchyInterface, 19
 - Ogre::OctreeHierarchyInterface, 61
 - Ogre::SceneNodeHierarchyInterface, 90
- RenderScene
 - GtpVisibility::CoherentHierarchical-CullingManager, 29
 - GtpVisibility::CullingManager, 32
 - GtpVisibility::FrustumCullingManager, 38
 - GtpVisibility::StopAndWaitCulling-Manager, 95
- ResultAvailable
 - GtpVisibility::OcclusionQuery, 58
 - Ogre::PlatformOcclusionQuery, 73
- SceneNodeHierarchyInterface
 - Ogre::SceneNodeHierarchyInterface, 90
- SetCamera
 - Ogre::PlatformHierarchyInterface, 68
- SetCullingManager
 - GtpVisibility::VisibilityManager, 105
- SetHierarchyInterface
 - GtpVisibility::CullingManager, 32
- SetLastVisited
 - GtpVisibility::HierarchyInterface, 44
 - Ogre::BspHierarchyInterface, 20
 - Ogre::OctreeHierarchyInterface, 62
 - Ogre::SceneNodeHierarchyInterface, 91
- SetNodeVisible
 - GtpVisibility::HierarchyInterface, 44
 - Ogre::BspHierarchyInterface, 20
 - Ogre::OctreeHierarchyInterface, 62
 - Ogre::SceneNodeHierarchyInterface, 91
- SetNumOctreeNodeNodes
 - Ogre::OctreeHierarchyInterface, 61
- setOcclusionQueryMaterial
 - Ogre::SolidHalfBoundingBox, 93
- setOption
 - Ogre::VisibilityBspSceneManager, 98
 - Ogre::VisibilityDotSceneManager, 101
 - Ogre::VisibilityOctreeSceneManager, 107
 - Ogre::VisibilitySceneManager, 110
 - Ogre::VisibilityTerrainSceneManager, 112
- SetRenderSystem
 - Ogre::PlatformHierarchyInterface, 69
- SetSceneManager
 - Ogre::PlatformHierarchyInterface, 69
- SetSceneRoot
 - GtpVisibility::HierarchyInterface, 42
- SetSceneTraverser
 - GtpVisibility::PreprocessingManager, 76
 - GtpVisibility::QueryManager, 83
- setupBoundingBox
 - Ogre::SolidHalfBoundingBox, 93
- setupBoundingBoxVertices
 - Ogre::SolidHalfBoundingBox, 93
- setVisibilityManager
 - Ogre::VisibilityBspSceneManager, 98
 - Ogre::VisibilityDotSceneManager, 101
 - Ogre::VisibilityOctreeSceneManager, 107
 - Ogre::VisibilitySceneManager, 110
 - Ogre::VisibilityTerrainSceneManager, 112
- ShootRay
 - GtpVisibility::QueryManager, 83
- SolidHalfBoundingBox
 - Ogre::SolidHalfBoundingBox, 93
- std, 15
- STOP_AND_WAIT
 - GtpVisibility::VisibilityEnvironment, 103
- StopAndWaitCullingManager

- GtpVisibility::StopAndWaitCulling-Manager, [95](#)
- Subdivide
 - GtpVisibilityPreprocessor::KdTree, [53](#)
- TraverseNode
 - GtpVisibility::HierarchyInterface, [41](#)
 - Ogre::BspHierarchyInterface, [19](#)
 - Ogre::OctreeHierarchyInterface, [61](#)
 - Ogre::SceneNodeHierarchyInterface, [90](#)
- Vector3
 - GtpVisibility, [12](#)
- VertexContainer
 - GtpVisibilityPreprocessor::Mesh, [54](#)
- ViewCellContainer
 - GtpVisibilityPreprocessor, [13](#)
- visEnv
 - OgreVisibilitySceneManagerDll.cpp, [181](#)
- VisibilityBspSceneManager
 - Ogre::VisibilityBspSceneManager, [98](#)
- VisibilityDotSceneManager
 - Ogre::VisibilityDotSceneManager, [101](#)
- VisibilityEnvironment
 - GtpVisibility::VisibilityEnvironment, [103](#)
- VisibilityInfo.h
 - InfoContainer, [129](#)
- VisibilityManager
 - GtpVisibility::VisibilityManager, [104](#)
- VisibilityOctreeSceneManager
 - Ogre::VisibilityOctreeSceneManager, [107](#)
- VisibilitySceneManager
 - Ogre::VisibilitySceneManager, [109](#)
- VisibilityTerrainSceneManager
 - Ogre::VisibilityTerrainSceneManager, [112](#)
- visManager
 - OgreVisibilitySceneManagerDll.cpp, [181](#)
- x
 - GtpVisibilityPreprocessor::Vector3, [96](#)
- y
 - GtpVisibilityPreprocessor::Vector3, [96](#)
- z
 - GtpVisibilityPreprocessor::Vector3, [96](#)