

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 Directory Hierarchy	3
2.1	GameTools Visibility Modules Directories	3
3	GameTools Visibility Modules Namespace Index	5
3.1	GameTools Visibility Modules Namespace List	5
4	GameTools Visibility Modules Hierarchical Index	7
4.1	GameTools Visibility Modules Class Hierarchy	7
5	GameTools Visibility Modules Class Index	9
5.1	GameTools Visibility Modules Class List	9
6	GameTools Visibility Modules File Index	11
6.1	GameTools Visibility Modules File List	11
7	GameTools Visibility Modules Directory Documentation	13
7.1	GtpVisibility/ Directory Reference	13
7.2	GtpVisibilityPreprocessor/ Directory Reference	14
7.3	Ogre/include/ Directory Reference	15
7.4	GtpVisibilityPreprocessor/include/ Directory Reference	16
7.5	GtpVisibility/include/ Directory Reference	17
7.6	Ogre/ Directory Reference	18
7.7	Ogre/src/ Directory Reference	19
7.8	GtpVisibilityPreprocessor/src/ Directory Reference	20

7.9	GtpVisibility/src/ Directory Reference	21
8	GameTools Visibility Modules Namespace Documentation	23
8.1	GtpVisibility Namespace Reference	23
8.2	GtpVisibilityPreprocessor Namespace Reference	25
8.3	Ogre Namespace Reference	26
8.4	std Namespace Reference	27
9	GameTools Visibility Modules Class Documentation	29
9.1	GtpVisibilityPreprocessor::AxisAlignedBox3 Class Reference	29
9.2	Ogre::BspHierarchyInterface Class Reference	30
9.3	GtpVisibilityPreprocessor::BSPInterior Class Reference	34
9.4	GtpVisibilityPreprocessor::BSPLeaf Class Reference	36
9.5	GtpVisibilityPreprocessor::BSPNode Class Reference	38
9.6	GtpVisibilityPreprocessor::BSPTree Class Reference	40
9.7	GtpVisibility::CoherentHierarchicalCullingManager Class Reference	42
9.8	GtpVisibility::CullingManager Class Reference	45
9.9	GtpVisibility::DummyPreprocessingManager Class Reference	48
9.10	GtpVisibility::DummyQueryManager Class Reference	51
9.11	GtpVisibilityPreprocessor::ExactPreprocessor Class Reference	54
9.12	GtpVisibility::FrustumCullingManager Class Reference	57
9.13	GtpVisibility::GreaterDistance< T > Class Template Reference	60
9.14	GtpVisibility::HierarchyInterface Class Reference	62
9.15	GtpVisibilityPreprocessor::KdInterior Class Reference	68
9.16	GtpVisibilityPreprocessor::KdLeaf Class Reference	70
9.17	GtpVisibilityPreprocessor::KdNode Class Reference	72
9.18	GtpVisibilityPreprocessor::KdTree Class Reference	74
9.19	GtpVisibilityPreprocessor::Mesh Class Reference	76
9.20	GtpVisibility::MeshInfo Class Reference	78
9.21	GtpVisibility::NodeInfo Class Reference	79
9.22	GtpVisibility::OcclusionQuery Class Reference	80
9.23	Ogre::OctreeHierarchyInterface Class Reference	82
9.24	GtpVisibilityPreprocessor::Patch Class Reference	87
9.25	GtpVisibilityPreprocessor::Plane3 Class Reference	88
9.26	Ogre::PlatformHierarchyInterface Class Reference	89
9.27	Ogre::PlatformOcclusionQuery Class Reference	95
9.28	GtpVisibility::PreprocessingManager Class Reference	98

9.29	GtpVisibilityPreprocessor::Preprocessor Class Reference	103
9.30	GtpVisibility::QueryManager Class Reference	107
9.31	GtpVisibilityPreprocessor::SamplingPreprocessor Class Reference	110
9.32	GtpVisibilityPreprocessor::SceneGraph Class Reference	113
9.33	GtpVisibilityPreprocessor::SceneGraphNode Class Reference	114
9.34	Ogre::SceneNodeHierarchyInterface Class Reference	115
9.35	Ogre::SolidHalfBoundingBox Class Reference	119
9.36	GtpVisibility::StopAndWaitCullingManager Class Reference	121
9.37	GtpVisibilityPreprocessor::Vector3 Class Reference	124
9.38	Ogre::VisibilityBspSceneManager Class Reference	125
9.39	Ogre::VisibilityDotSceneManager Class Reference	128
9.40	GtpVisibility::VisibilityEnvironment Class Reference	131
9.41	GtpVisibility::VisibilityManager Class Reference	132
9.42	Ogre::VisibilityOctreeSceneManager Class Reference	134
9.43	Ogre::VisibilitySceneManager Class Reference	137
9.44	Ogre::VisibilityTerrainSceneManager Class Reference	140
10	GameTools Visibility Modules File Documentation	143
10.1	GtpVisibility/include/CoherentHierarchicalCullingManager.h File Reference	143
10.2	GtpVisibility/include/CullingManager.h File Reference	145
10.3	GtpVisibility/include/DistanceQueue.h File Reference	146
10.4	GtpVisibility/include/DummyPreprocessingManager.h File Reference	147
10.5	GtpVisibility/include/DummyQueryManager.h File Reference	148
10.6	GtpVisibility/include/FrustumCullingManager.h File Reference	149
10.7	GtpVisibility/include/HierarchyInterface.h File Reference	150
10.8	GtpVisibility/include/OcclusionQuery.h File Reference	151
10.9	GtpVisibility/include/PreprocessingManager.h File Reference	152
10.10	GtpVisibility/include/QueryManager.h File Reference	153
10.11	GtpVisibility/include/StopAndWaitCullingManager.h File Reference	154
10.12	GtpVisibility/include/VisibilityAxisAlignedBox.h File Reference	155
10.13	GtpVisibility/include/VisibilityCamera.h File Reference	156
10.14	GtpVisibility/include/VisibilityEnvironment.h File Reference	157
10.15	GtpVisibility/include/VisibilityInfo.h File Reference	158
10.16	GtpVisibility/include/VisibilityManager.h File Reference	159
10.17	GtpVisibility/include/VisibilityMesh.h File Reference	160
10.18	GtpVisibility/include/VisibilityRay.h File Reference	161
10.19	GtpVisibility/include/VisibilityVector3.h File Reference	162

10.20GtpVisibility/src/CoherentHierarchicalCullingManager.cpp File Reference	163
10.21GtpVisibility/src/CullingManager.cpp File Reference	164
10.22GtpVisibility/src/DummyPreprocessingManager.cpp File Reference	165
10.23GtpVisibility/src/DummyQueryManager.cpp File Reference	166
10.24GtpVisibility/src/FrustumCullingManager.cpp File Reference	167
10.25GtpVisibility/src/HierarchyInterface.cpp File Reference	168
10.26GtpVisibility/src/PreprocessingManager.cpp File Reference	169
10.27GtpVisibility/src/QueryManager.cpp File Reference	170
10.28GtpVisibility/src/StopAndWaitCullingManager.cpp File Reference	171
10.29GtpVisibility/src/VisibilityEnvironment.cpp File Reference	172
10.30GtpVisibility/src/VisibilityManager.cpp File Reference	173
10.31GtpVisibilityPreprocessor/include/AxisAlignedBox3.h File Reference	174
10.32GtpVisibilityPreprocessor/include/Containers.h File Reference	175
10.33GtpVisibilityPreprocessor/include/ExactPreprocessor.h File Reference	176
10.34GtpVisibilityPreprocessor/include/KdTree.h File Reference	177
10.35GtpVisibilityPreprocessor/include/Mesh.h File Reference	178
10.36GtpVisibilityPreprocessor/include/Plane3.h File Reference	179
10.37GtpVisibilityPreprocessor/include/Preprocessor.h File Reference	180
10.38GtpVisibilityPreprocessor/include/SamplingPreprocessor.h File Reference	181
10.39GtpVisibilityPreprocessor/include/SceneGraph.h File Reference	182
10.40GtpVisibilityPreprocessor/include/Vector3.h File Reference	183
10.41GtpVisibilityPreprocessor/include/ViewCellBSP.h File Reference	184
10.42GtpVisibilityPreprocessor/src/ExactPreprocessor.cpp File Reference	185
10.43GtpVisibilityPreprocessor/src/KdTree.cpp File Reference	186
10.44GtpVisibilityPreprocessor/src/Preprocessor.cpp File Reference	187
10.45GtpVisibilityPreprocessor/src/SamplingPreprocessor.cpp File Reference	188
10.46Ogre/include/OgreBspHierarchyInterface.h File Reference	189
10.47Ogre/include/OgreOctreeHierarchyInterface.h File Reference	190
10.48Ogre/include/OgrePlatformHierarchyInterface.h File Reference	191
10.49Ogre/include/OgrePlatformOcclusionQuery.h File Reference	192
10.50Ogre/include/OgreSceneNodeHierarchyInterface.h File Reference	193
10.51Ogre/include/OgreSolidHalfBoundingBox.h File Reference	194
10.52Ogre/include/OgreVisibilityBspSceneManager.h File Reference	195
10.53Ogre/include/OgreVisibilityDotSceneManager.h File Reference	196
10.54Ogre/include/OgreVisibilityOctreeSceneManager.h File Reference	197
10.55Ogre/include/OgreVisibilitySceneManager.h File Reference	198

10.56Ogre/include/OgreVisibilityTerrainSceneManager.h File Reference	199
10.57Ogre/src/OgreBspHierarchyInterface.cpp File Reference	200
10.58Ogre/src/OgreOctreeHierarchyInterface.cpp File Reference	201
10.59Ogre/src/OgrePlatformHierarchyInterface.cpp File Reference	202
10.60Ogre/src/OgrePlatformOcclusionQuery.cpp File Reference	203
10.61Ogre/src/OgreSceneNodeHierarchyInterface.cpp File Reference	204
10.62Ogre/src/OgreSolidHalfBoundingBox.cpp File Reference	205
10.63Ogre/src/OgreVisibilityBspSceneManager.cpp File Reference	206
10.64Ogre/src/OgreVisibilityDotSceneManager.cpp File Reference	207
10.65Ogre/src/OgreVisibilityOctreeSceneManager.cpp File Reference	208
10.66Ogre/src/OgreVisibilitySceneManager.cpp File Reference	209
10.67Ogre/src/OgreVisibilitySceneManagerDll.cpp File Reference	210
10.68Ogre/src/OgreVisibilityTerrainSceneManager.cpp File Reference	212

Chapter 1

Overview

The visibility workpackage aims to develop methods for making use of restricted visibility in large scenes. The major goal of these methods is to avoid wasting 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 parts 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 workpackage consists 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 easy communication between them.

1.2 Visibility Culling

Visibility culling seamlessly integrates into the rendering loop and eliminates most invisible objects from being sent 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.

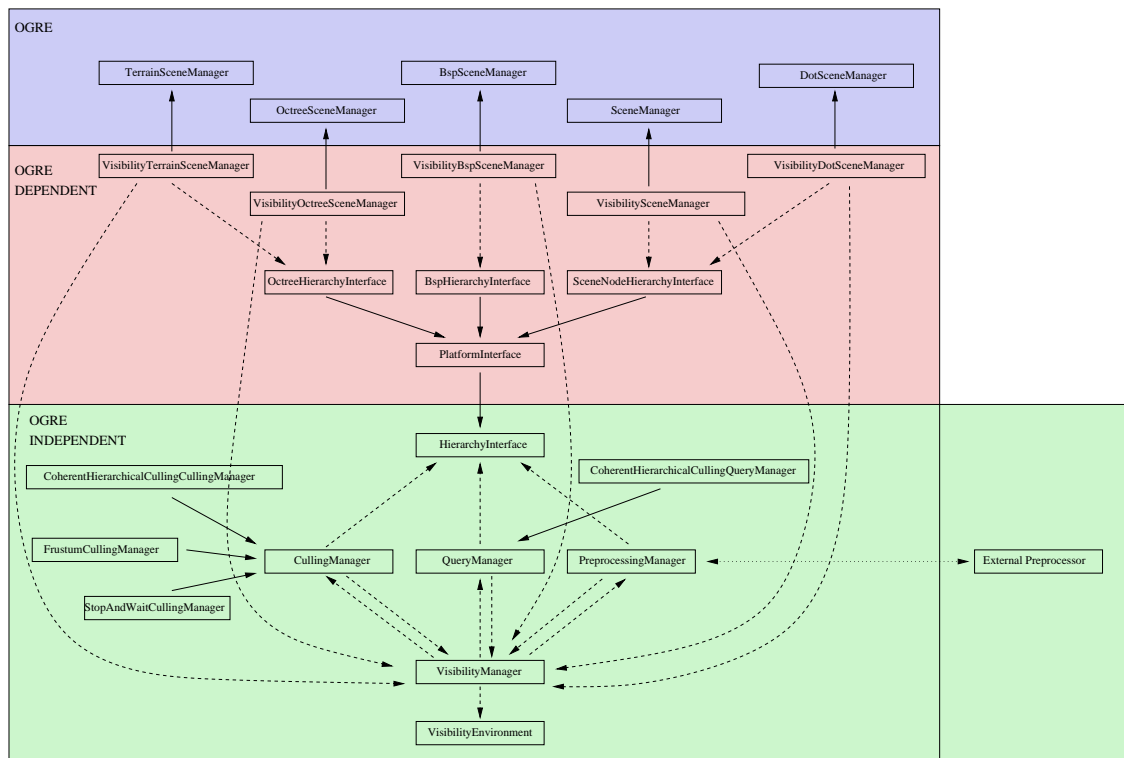


Figure 1.1: Overview of the visibility work packages classes and their integration into the Ogre game engine.

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 a set of requirements. Alternatively, the module can generate view cells by automatic view space partitioning.

Chapter 2

GameTools Visibility Modules Directory Hierarchy

2.1 GameTools Visibility Modules Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

GtpVisibility	13
include	17
src	21
GtpVisibilityPreprocessor	14
include	16
src	20
Ogre	18
include	15
src	19

Chapter 3

GameTools Visibility Modules Namespace Index

3.1 GameTools Visibility Modules Namespace List

Here is a list of all namespaces with brief descriptions:

GtpVisibility	23
GtpVisibilityPreprocessor	25
Ogre	26
std	27

Chapter 4

GameTools Visibility Modules Hierarchical Index

4.1 GameTools Visibility Modules Class Hierarchy

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

GtpVisibilityPreprocessor::AxisAlignedBox3	29
GtpVisibilityPreprocessor::BSPNode	38
GtpVisibilityPreprocessor::BSPInterior	34
GtpVisibilityPreprocessor::BSPLeaf	36
GtpVisibilityPreprocessor::BSPTree	40
GtpVisibility::CullingManager	45
GtpVisibility::CoherentHierarchicalCullingManager	42
GtpVisibility::FrustumCullingManager	57
GtpVisibility::StopAndWaitCullingManager	121
GtpVisibility::GreaterDistance< T >	60
GtpVisibility::HierarchyInterface	62
Ogre::PlatformHierarchyInterface	89
Ogre::BspHierarchyInterface	30
Ogre::OctreeHierarchyInterface	82
Ogre::SceneNodeHierarchyInterface	115
GtpVisibilityPreprocessor::KdNode	72
GtpVisibilityPreprocessor::KdInterior	68
GtpVisibilityPreprocessor::KdLeaf	70
GtpVisibilityPreprocessor::KdTree	74
GtpVisibilityPreprocessor::Mesh	76
GtpVisibility::MeshInfo	78
GtpVisibility::NodeInfo	79
GtpVisibility::OcclusionQuery	80
Ogre::PlatformOcclusionQuery	95
GtpVisibilityPreprocessor::Patch	87
GtpVisibilityPreprocessor::Plane3	88
GtpVisibility::PreprocessingManager	98
GtpVisibility::DummyPreprocessingManager	48
GtpVisibilityPreprocessor::Preprocessor	103

GtpVisibilityPreprocessor::ExactPreprocessor	54
GtpVisibilityPreprocessor::SamplingPreprocessor	110
GtpVisibility::QueryManager	107
GtpVisibility::DummyQueryManager	51
GtpVisibilityPreprocessor::SceneGraph	113
GtpVisibilityPreprocessor::SceneGraphNode	114
Ogre::SolidHalfBoundingBox	119
GtpVisibilityPreprocessor::Vector3	124
Ogre::VisibilityBspSceneManager	125
Ogre::VisibilityDotSceneManager	128
GtpVisibility::VisibilityEnvironment	131
GtpVisibility::VisibilityManager	132
Ogre::VisibilityOctreeSceneManager	134
Ogre::VisibilitySceneManager	137
Ogre::VisibilityTerrainSceneManager	140

Chapter 5

GameTools Visibility Modules Class Index

5.1 GameTools Visibility Modules Class List

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

GtpVisibilityPreprocessor::AxisAlignedBox3	29
Ogre::BspHierarchyInterface	30
GtpVisibilityPreprocessor::BSPInterior	34
GtpVisibilityPreprocessor::BSPLeaf	36
GtpVisibilityPreprocessor::BSPNode	38
GtpVisibilityPreprocessor::BSPTree	40
GtpVisibility::CoherentHierarchicalCullingManager	42
GtpVisibility::CullingManager	45
GtpVisibility::DummyPreprocessingManager	48
GtpVisibility::DummyQueryManager	51
GtpVisibilityPreprocessor::ExactPreprocessor	54
GtpVisibility::FrustumCullingManager	57
GtpVisibility::GreaterDistance< T >	60
GtpVisibility::HierarchyInterface	62
GtpVisibilityPreprocessor::KdInterior	68
GtpVisibilityPreprocessor::KdLeaf	70
GtpVisibilityPreprocessor::KdNode	72
GtpVisibilityPreprocessor::KdTree	74
GtpVisibilityPreprocessor::Mesh	76
GtpVisibility::MeshInfo	78
GtpVisibility::NodeInfo	79
GtpVisibility::OcclusionQuery	80
Ogre::OctreeHierarchyInterface	82
GtpVisibilityPreprocessor::Patch	87
GtpVisibilityPreprocessor::Plane3	88
Ogre::PlatformHierarchyInterface	89
Ogre::PlatformOcclusionQuery	95
GtpVisibility::PreprocessingManager	98
GtpVisibilityPreprocessor::Preprocessor	103
GtpVisibility::QueryManager	107
GtpVisibilityPreprocessor::SamplingPreprocessor	110

GtpVisibilityPreprocessor::SceneGraph	113
GtpVisibilityPreprocessor::SceneGraphNode	114
Ogre::SceneNodeHierarchyInterface	115
Ogre::SolidHalfBoundingBox	119
GtpVisibility::StopAndWaitCullingManager	121
GtpVisibilityPreprocessor::Vector3	124
Ogre::VisibilityBspSceneManager	125
Ogre::VisibilityDotSceneManager	128
GtpVisibility::VisibilityEnvironment	131
GtpVisibility::VisibilityManager	132
Ogre::VisibilityOctreeSceneManager	134
Ogre::VisibilitySceneManager	137
Ogre::VisibilityTerrainSceneManager	140

Chapter 6

GameTools Visibility Modules File Index

6.1 GameTools Visibility Modules File List

Here is a list of all files with brief descriptions:

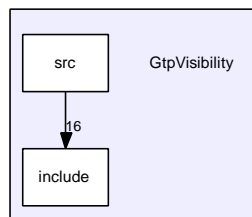
GtpVisibility/include/CoherentHierarchicalCullingManager.h	143
GtpVisibility/include/CullingManager.h	145
GtpVisibility/include/DistanceQueue.h	146
GtpVisibility/include/DummyPreprocessingManager.h	147
GtpVisibility/include/DummyQueryManager.h	148
GtpVisibility/include/FrustumCullingManager.h	149
GtpVisibility/include/HierarchyInterface.h	150
GtpVisibility/include/OcclusionQuery.h	151
GtpVisibility/include/PreprocessingManager.h	152
GtpVisibility/include/QueryManager.h	153
GtpVisibility/include/StopAndWaitCullingManager.h	154
GtpVisibility/include/VisibilityAxisAlignedBox.h	155
GtpVisibility/include/VisibilityCamera.h	156
GtpVisibility/include/VisibilityEnvironment.h	157
GtpVisibility/include/VisibilityInfo.h	158
GtpVisibility/include/VisibilityManager.h	159
GtpVisibility/include/VisibilityMesh.h	160
GtpVisibility/include/VisibilityRay.h	161
GtpVisibility/include/VisibilityVector3.h	162
GtpVisibility/src/CoherentHierarchicalCullingManager.cpp	163
GtpVisibility/src/CullingManager.cpp	164
GtpVisibility/src/DummyPreprocessingManager.cpp	165
GtpVisibility/src/DummyQueryManager.cpp	166
GtpVisibility/src/FrustumCullingManager.cpp	167
GtpVisibility/src/HierarchyInterface.cpp	168
GtpVisibility/src/PreprocessingManager.cpp	169
GtpVisibility/src/QueryManager.cpp	170
GtpVisibility/src/StopAndWaitCullingManager.cpp	171
GtpVisibility/src/VisibilityEnvironment.cpp	172
GtpVisibility/src/VisibilityManager.cpp	173
GtpVisibilityPreprocessor/include/AxisAlignedBox3.h	174

GtpVisibilityPreprocessor/include/Containers.h	175
GtpVisibilityPreprocessor/include/ExactPreprocessor.h	176
GtpVisibilityPreprocessor/include/KdTree.h	177
GtpVisibilityPreprocessor/include/Mesh.h	178
GtpVisibilityPreprocessor/include/Plane3.h	179
GtpVisibilityPreprocessor/include/Preprocessor.h	180
GtpVisibilityPreprocessor/include/SamplingPreprocessor.h	181
GtpVisibilityPreprocessor/include/SceneGraph.h	182
GtpVisibilityPreprocessor/include/Vector3.h	183
GtpVisibilityPreprocessor/include/ViewCellBSP.h	184
GtpVisibilityPreprocessor/src/ExactPreprocessor.cpp	185
GtpVisibilityPreprocessor/src/KdTree.cpp	186
GtpVisibilityPreprocessor/src/Preprocessor.cpp	187
GtpVisibilityPreprocessor/src/SamplingPreprocessor.cpp	188
Ogre/include/OgreBspHierarchyInterface.h	189
Ogre/include/OgreOctreeHierarchyInterface.h	190
Ogre/include/OgrePlatformHierarchyInterface.h	191
Ogre/include/OgrePlatformOcclusionQuery.h	192
Ogre/include/OgreSceneNodeHierarchyInterface.h	193
Ogre/include/OgreSolidHalfBoundingBox.h	194
Ogre/include/OgreVisibilityBspSceneManager.h	195
Ogre/include/OgreVisibilityDotSceneManager.h	196
Ogre/include/OgreVisibilityOctreeSceneManager.h	197
Ogre/include/OgreVisibilitySceneManager.h	198
Ogre/include/OgreVisibilityTerrainSceneManager.h	199
Ogre/src/OgreBspHierarchyInterface.cpp	200
Ogre/src/OgreOctreeHierarchyInterface.cpp	201
Ogre/src/OgrePlatformHierarchyInterface.cpp	202
Ogre/src/OgrePlatformOcclusionQuery.cpp	203
Ogre/src/OgreSceneNodeHierarchyInterface.cpp	204
Ogre/src/OgreSolidHalfBoundingBox.cpp	205
Ogre/src/OgreVisibilityBspSceneManager.cpp	206
Ogre/src/OgreVisibilityDotSceneManager.cpp	207
Ogre/src/OgreVisibilityOctreeSceneManager.cpp	208
Ogre/src/OgreVisibilitySceneManager.cpp	209
Ogre/src/OgreVisibilitySceneManagerDll.cpp	210
Ogre/src/OgreVisibilityTerrainSceneManager.cpp	212

Chapter 7

GameTools Visibility Modules Directory Documentation

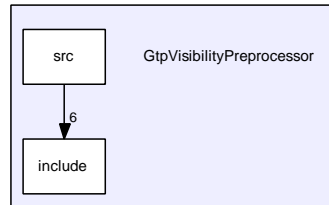
7.1 GtpVisibility/ Directory Reference



Directories

- directory [include](#)
- directory [src](#)

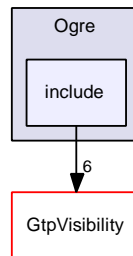
7.2 GtpVisibilityPreprocessor/ Directory Reference



Directories

- directory [include](#)
- directory [src](#)

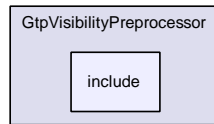
7.3 Ogre/include/ Directory Reference



Files

- file [OgreBspHierarchyInterface.h](#)
- file [OgreOctreeHierarchyInterface.h](#)
- file [OgrePlatformHierarchyInterface.h](#)
- file [OgrePlatformOcclusionQuery.h](#)
- file [OgreSceneNodeHierarchyInterface.h](#)
- file [OgreSolidHalfBoundingBox.h](#)
- file [OgreVisibilityBspSceneManager.h](#)
- file [OgreVisibilityDotSceneManager.h](#)
- file [OgreVisibilityOctreeSceneManager.h](#)
- file [OgreVisibilitySceneManager.h](#)
- file [OgreVisibilityTerrainSceneManager.h](#)

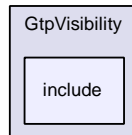
7.4 GtpVisibilityPreprocessor/include/ Directory Reference



Files

- file [AxisAlignedBox3.h](#)
- file [Containers.h](#)
- file [ExactPreprocessor.h](#)
- file [KdTree.h](#)
- file [Mesh.h](#)
- file [Plane3.h](#)
- file [Preprocessor.h](#)
- file [SamplingPreprocessor.h](#)
- file [SceneGraph.h](#)
- file [Vector3.h](#)
- file [ViewCellBSP.h](#)

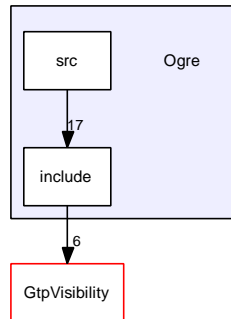
7.5 GtpVisibility/include/ Directory Reference



Files

- file [CoherentHierarchicalCullingManager.h](#)
- file [CullingManager.h](#)
- file [DistanceQueue.h](#)
- file [DummyPreprocessingManager.h](#)
- file [DummyQueryManager.h](#)
- file [FrustumCullingManager.h](#)
- file [HierarchyInterface.h](#)
- file [OcclusionQuery.h](#)
- file [PreprocessingManager.h](#)
- file [QueryManager.h](#)
- file [StopAndWaitCullingManager.h](#)
- file [VisibilityAxisAlignedBox.h](#)
- file [VisibilityCamera.h](#)
- file [VisibilityEnvironment.h](#)
- file [VisibilityInfo.h](#)
- file [VisibilityManager.h](#)
- file [VisibilityMesh.h](#)
- file [VisibilityRay.h](#)
- file [VisibilityVector3.h](#)

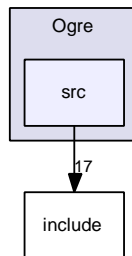
7.6 Ogre/ Directory Reference



Directories

- directory [include](#)
- directory [src](#)

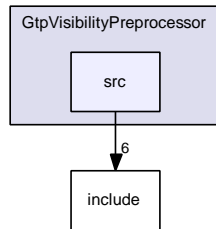
7.7 Ogre/src/ Directory Reference



Files

- file [OgreBspHierarchyInterface.cpp](#)
- file [OgreOctreeHierarchyInterface.cpp](#)
- file [OgrePlatformHierarchyInterface.cpp](#)
- file [OgrePlatformOcclusionQuery.cpp](#)
- file [OgreSceneNodeHierarchyInterface.cpp](#)
- file [OgreSolidHalfBoundingBox.cpp](#)
- file [OgreVisibilityBspSceneManager.cpp](#)
- file [OgreVisibilityDotSceneManager.cpp](#)
- file [OgreVisibilityOctreeSceneManager.cpp](#)
- file [OgreVisibilitySceneManager.cpp](#)
- file [OgreVisibilitySceneManagerDll.cpp](#)
- file [OgreVisibilityTerrainSceneManager.cpp](#)

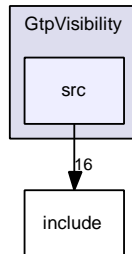
7.8 GtpVisibilityPreprocessor/src/ Directory Reference



Files

- file [ExactPreprocessor.cpp](#)
- file [KdTree.cpp](#)
- file [Preprocessor.cpp](#)
- file [SamplingPreprocessor.cpp](#)

7.9 GtpVisibility/src/ Directory Reference



Files

- file [CoherentHierarchicalCullingManager.cpp](#)
- file [CullingManager.cpp](#)
- file [DummyPreprocessingManager.cpp](#)
- file [DummyQueryManager.cpp](#)
- file [FrustumCullingManager.cpp](#)
- file [HierarchyInterface.cpp](#)
- file [PreprocessingManager.cpp](#)
- file [QueryManager.cpp](#)
- file [StopAndWaitCullingManager.cpp](#)
- file [VisibilityEnvironment.cpp](#)
- file [VisibilityManager.cpp](#)

Chapter 8

GameTools Visibility Modules Namespace Documentation

8.1 GtpVisibility Namespace Reference

Classes

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

Typedefs

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

8.1.1 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.

8.1.2 Typedef Documentation

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

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

8.1.2.3 `typedef void GtpVisibility::HierarchyNode`

8.1.2.4 `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.

8.1.2.5 `typedef Ogre::AxisAlignedBox GtpVisibility::AxisAlignedBox`

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

8.1.2.6 `typedef Ogre::Camera GtpVisibility::Camera`

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

8.1.2.7 `typedef Ogre::Mesh GtpVisibility::Mesh`

8.1.2.8 `typedef Ogre::Ray GtpVisibility::Ray`

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

8.1.2.9 `typedef Ogre::Vector3 GtpVisibility::Vector3`

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

8.2 GtpVisibilityPreprocessor Namespace Reference

Classes

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

Typedefs

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

8.2.1 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).

8.2.2 Typedef Documentation

8.2.2.1 typedef vector<Mesh *> GtpVisibilityPreprocessor::MeshContainer

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

8.2.2.2 typedef vector<ViewCell *> GtpVisibilityPreprocessor::ViewCellContainer

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

8.2.2.3 typedef vector<HierarchyNode *> GtpVisibilityPreprocessor::NodeContainer

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

8.3 Ogre Namespace Reference

Classes

- 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](#)

8.3.1 Detailed Description

This namespace contains the Ogre dependent classes.

8.3.2 Function Documentation

8.3.2.1 void [Ogre::dllStartPlugin](#) (void)

8.3.2.2 void [Ogre::dllStopPlugin](#) (void)

8.3.3 Variable Documentation

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

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

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

8.4 std Namespace Reference

Chapter 9

GameTools Visibility Modules Class Documentation

9.1 GtpVisibilityPreprocessor::AxisAlignedBox3 Class Reference

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

Protected Attributes

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

9.1.1 Detailed Description

Axis aligned box

9.1.2 Member Data Documentation

9.1.2.1 [Vector3 GtpVisibilityPreprocessor::AxisAlignedBox3::min](#) [protected]

9.1.2.2 [Vector3 GtpVisibilityPreprocessor::AxisAlignedBox3::max](#) [protected]

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

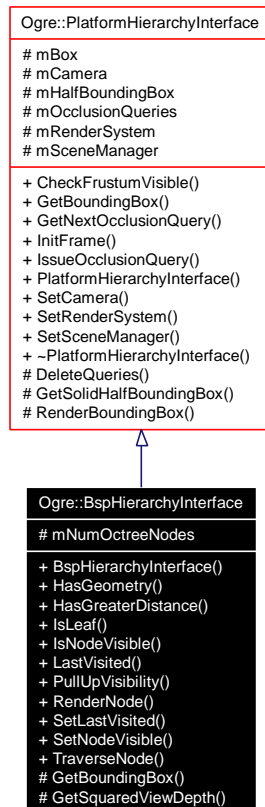
- GtpVisibilityPreprocessor/include/[AxisAlignedBox3.h](#)

9.2 Ogre::BspHierarchyInterface Class Reference

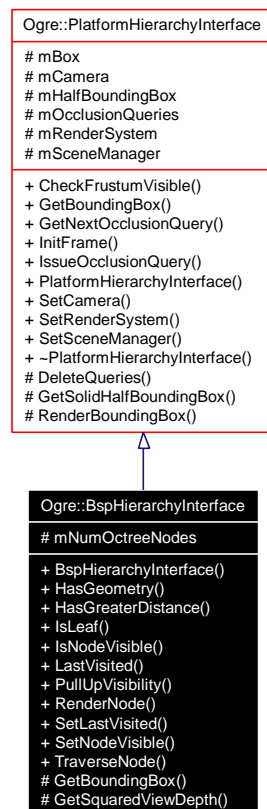
```
#include <OgreBspHierarchyInterface.h>
```

Inherits [Ogre::PlatformHierarchyInterface](#).

Inheritance diagram for `Ogre::BspHierarchyInterface`:



Collaboration diagram for `Ogre::BspHierarchyInterface`:



Public Member Functions

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

Protected Member Functions

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

Protected Attributes

- unsigned int [mNumOctreeNodes](#)

9.2.1 Detailed Description

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

9.2.2 Constructor & Destructor Documentation

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

9.2.3 Member Function Documentation

9.2.3.1 `void Ogre::BspHierarchyInterface::PullUpVisibility (GtpVisibility::HierarchyNode *
node)`

Gets the given option for the scene traverser.

Remarks:

See `setOption`

9.2.3.2 `void Ogre::BspHierarchyInterface::TraverseNode (GtpVisibility::HierarchyNode * node)`

Traverses given node.

Parameters:

node current node

Remarks:

pushes children on distance queue

- 9.2.3.3 void `Ogre::BspHierarchyInterface::RenderNode` (`GtpVisibility::HierarchyNode * node`)
- 9.2.3.4 bool `Ogre::BspHierarchyInterface::IsLeaf` (`GtpVisibility::HierarchyNode * node`)
- 9.2.3.5 bool `Ogre::BspHierarchyInterface::HasGeometry` (`GtpVisibility::HierarchyNode * node`)
- 9.2.3.6 bool `Ogre::BspHierarchyInterface::HasGreaterDistance` (`GtpVisibility::HierarchyNode * node1`, `GtpVisibility::HierarchyNode * node2`)
- 9.2.3.7 void `Ogre::BspHierarchyInterface::SetNodeVisible` (`GtpVisibility::HierarchyNode * node`, const bool *visible*)
- 9.2.3.8 void `Ogre::BspHierarchyInterface::SetLastVisited` (`GtpVisibility::HierarchyNode * node`, const int *frameId*)
- 9.2.3.9 bool `Ogre::BspHierarchyInterface::IsNodeVisible` (`GtpVisibility::HierarchyNode * node`)
- 9.2.3.10 int `Ogre::BspHierarchyInterface::LastVisited` (`GtpVisibility::HierarchyNode * node`)
- 9.2.3.11 `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](#).

- 9.2.3.12 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

9.2.4 Member Data Documentation

- 9.2.4.1 unsigned int `Ogre::BspHierarchyInterface::mNumOctreeNode`s [protected]

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

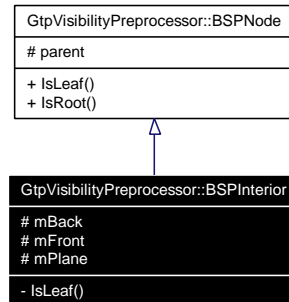
- [Ogre/include/OgreBspHierarchyInterface.h](#)
- [Ogre/src/OgreBspHierarchyInterface.cpp](#)

9.3 GtpVisibilityPreprocessor::BSPInterior Class Reference

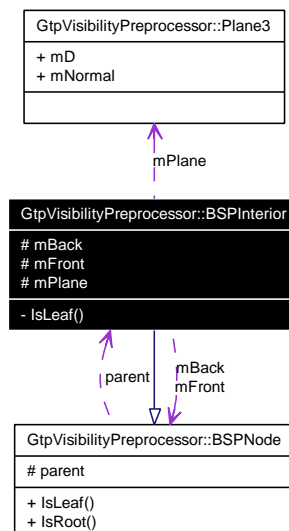
```
#include <ViewCellBSP.h>
```

Inherits [GtpVisibilityPreprocessor::BSPNode](#).

Inheritance diagram for GtpVisibilityPreprocessor::BSPInterior:



Collaboration diagram for GtpVisibilityPreprocessor::BSPInterior:



Protected Attributes

- `BSPNode * mBack`
back node
- `BSPNode * mFront`
front node
- `Plane3 mPlane`
Splitting plane corresponding to this node.

Private Member Functions

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

9.3.1 Detailed Description

BSP interior node implementation

9.3.2 Member Function Documentation

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

Returns:

false since it is an interior node

Implements [GtpVisibilityPreprocessor::BSPNode](#).

9.3.3 Member Data Documentation

9.3.3.1 Plane3 GtpVisibilityPreprocessor::BSPInterior::mPlane [protected]

Splitting plane corresponding to this node.

9.3.3.2 BSPNode* GtpVisibilityPreprocessor::BSPInterior::mBack [protected]

back node

9.3.3.3 BSPNode* GtpVisibilityPreprocessor::BSPInterior::mFront [protected]

front node

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

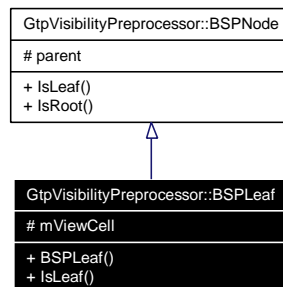
- GtpVisibilityPreprocessor/include/ViewCellBSP.h

9.4 GtpVisibilityPreprocessor::BSPLeaf Class Reference

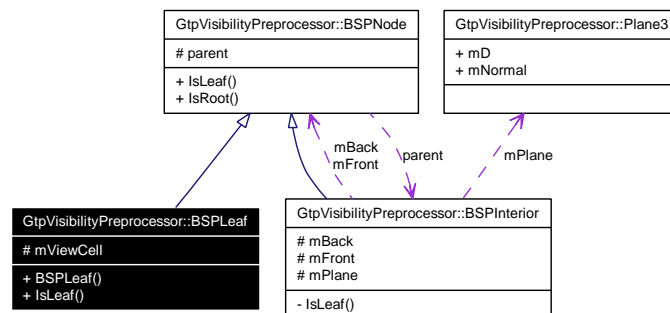
```
#include <ViewCellBSP.h>
```

Inherits [GtpVisibilityPreprocessor::BSPNode](#).

Inheritance diagram for GtpVisibilityPreprocessor::BSPLeaf:



Collaboration diagram for GtpVisibilityPreprocessor::BSPLeaf:



Public Member Functions

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

Protected Attributes

- ViewCell * [mViewCell](#)

9.4.1 Detailed Description

BSP leaf node implementation

9.4.2 Constructor & Destructor Documentation

9.4.2.1 `GtpVisibilityPreprocessor::BSPLeaf::BSPLeaf (ViewCell * viewCell = NULL)` [inline]

9.4.3 Member Function Documentation

9.4.3.1 `virtual bool GtpVisibilityPreprocessor::BSPLeaf::IsLeaf () const` [inline, virtual]

Returns:

true since it is an interior node

Implements [GtpVisibilityPreprocessor::BSPNode](#).

9.4.4 Member Data Documentation

9.4.4.1 `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:

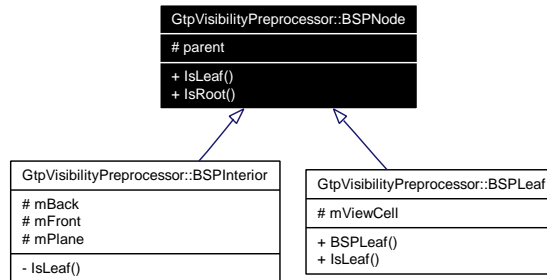
- [GtpVisibilityPreprocessor/include/ViewCellBSP.h](#)

9.5 GtpVisibilityPreprocessor::BSPNode Class Reference

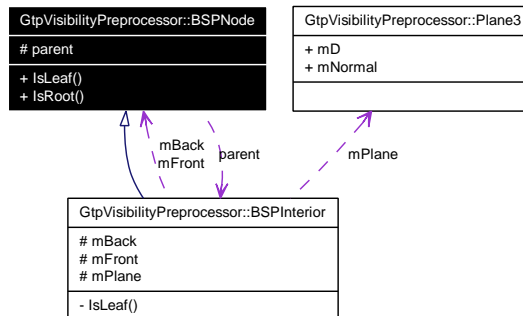
```
#include <ViewCellBSP.h>
```

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

Inheritance diagram for GtpVisibilityPreprocessor::BSPNode:



Collaboration diagram for GtpVisibilityPreprocessor::BSPNode:



Public Member Functions

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

Protected Attributes

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

9.5.1 Detailed Description

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

9.5.2 Member Function Documentation

9.5.2.1 virtual bool GtpVisibilityPreprocessor::BSPNode::IsLeaf () const [pure virtual]

Determines whether this node is a leaf or not

Returns:

true if leaf

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

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

Determines whether this node is a root

Returns:

true if root

9.5.3 Member Data Documentation

9.5.3.1 BSPInterior* GtpVisibilityPreprocessor::BSPNode::parent [protected]

parent of this node

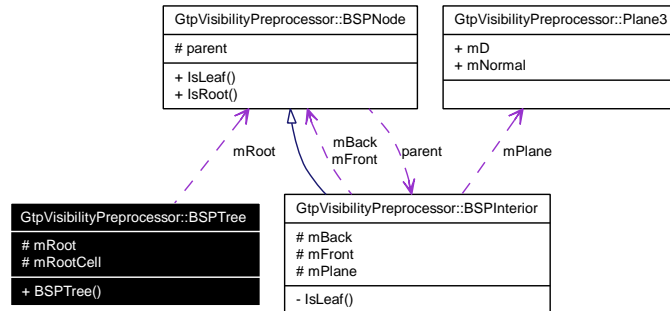
The documentation for this class was generated from the following file:

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

9.6 GtpVisibilityPreprocessor::BSPTree Class Reference

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

Collaboration diagram for GtpVisibilityPreprocessor::BSPTree:



Public Member Functions

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

Protected Attributes

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

9.6.1 Detailed Description

Implementation of the ViewCell BSP tree

9.6.2 Constructor & Destructor Documentation

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

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

9.6.3 Member Data Documentation

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

Pointer to the root of the tree.

9.6.3.2 ViewCell* [GtpVisibilityPreprocessor::BSPTree::mRootCell](#) [protected]

Pointer to the root cell of the viewspace */.

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

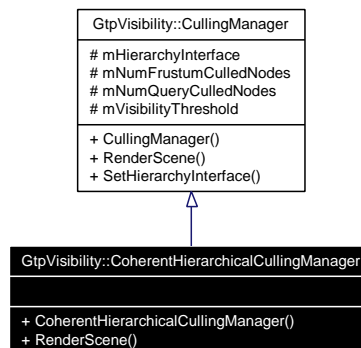
- [GtpVisibilityPreprocessor/include/ViewCellBSP.h](#)

9.7 GtpVisibility::CoherentHierarchicalCullingManager Class Reference

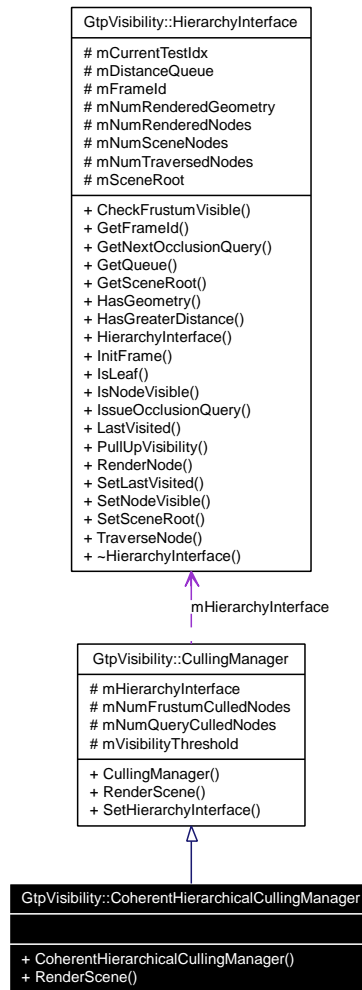
```
#include <CoherentHierarchicalCullingManager.h>
```

Inherits [GtpVisibility::CullingManager](#).

Inheritance diagram for GtpVisibility::CoherentHierarchicalCullingManager:



Collaboration diagram for GtpVisibility::CoherentHierarchicalCullingManager:



Public Member Functions

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

9.7.1 Detailed Description

Renders the scene with the coherent hierarchical culling algorithm.

9.7.2 Constructor & Destructor Documentation

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

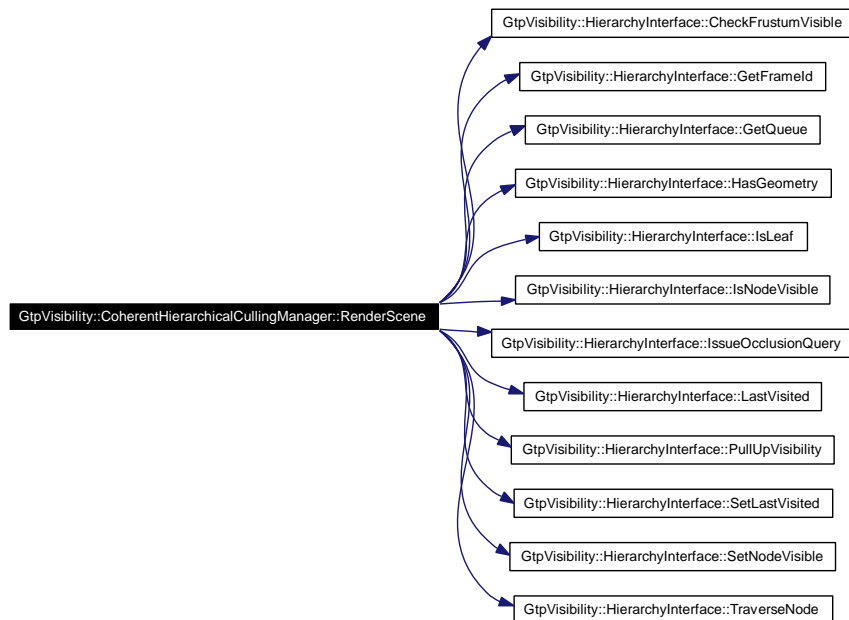
9.7.3 Member Function Documentation

9.7.3.1 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](#).

Here is the call graph for this function:



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

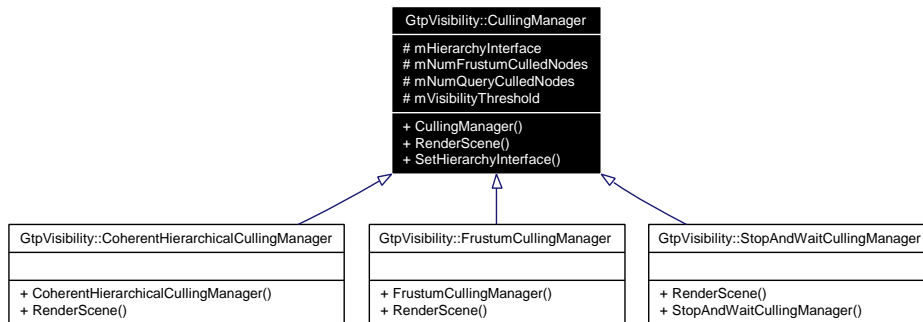
- [GtpVisibility/include/CoherentHierarchicalCullingManager.h](#)
- [GtpVisibility/src/CoherentHierarchicalCullingManager.cpp](#)

9.8 GtpVisibility::CullingManager Class Reference

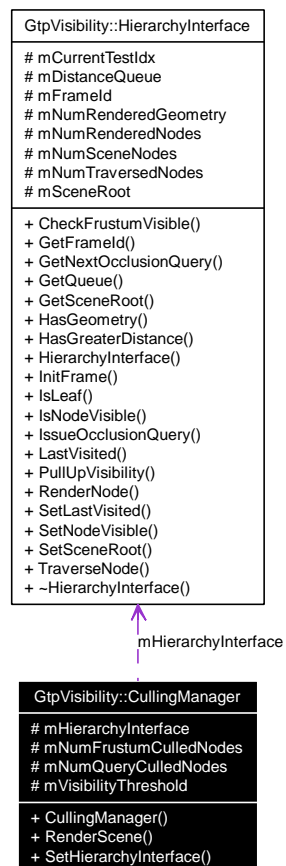
```
#include <CullingManager.h>
```

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

Inheritance diagram for GtpVisibility::CullingManager:



Collaboration diagram for GtpVisibility::CullingManager:



Public Member Functions

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

Protected Attributes

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

9.8.1 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.

9.8.2 Constructor & Destructor Documentation

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

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

9.8.3 Member Function Documentation

9.8.3.1 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](#).

9.8.3.2 void [GtpVisibility::CullingManager::SetHierarchyInterface](#) ([HierarchyInterface](#) * *hierarchyInterface*)

Sets the hierarchy interface.

Parameters:

hierarchyInterface

Remarks:

the hierarchy interface encapsulates the hierarchy we are working on

9.8.4 Member Data Documentation

9.8.4.1 unsigned int [GtpVisibility::CullingManager::mNumQueryCulledNodes](#) [protected]

9.8.4.2 unsigned int [GtpVisibility::CullingManager::mNumFrustumCulledNodes](#)
[protected]

9.8.4.3 unsigned int [GtpVisibility::CullingManager::mVisibilityThreshold](#) [protected]

9.8.4.4 [HierarchyInterface*](#) [GtpVisibility::CullingManager::mHierarchyInterface](#)
[protected]

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

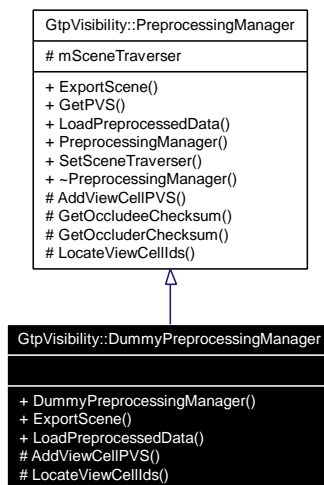
- [GtpVisibility/include/CullingManager.h](#)
- [GtpVisibility/src/CullingManager.cpp](#)

9.9 GtpVisibility::DummyPreprocessingManager Class Reference

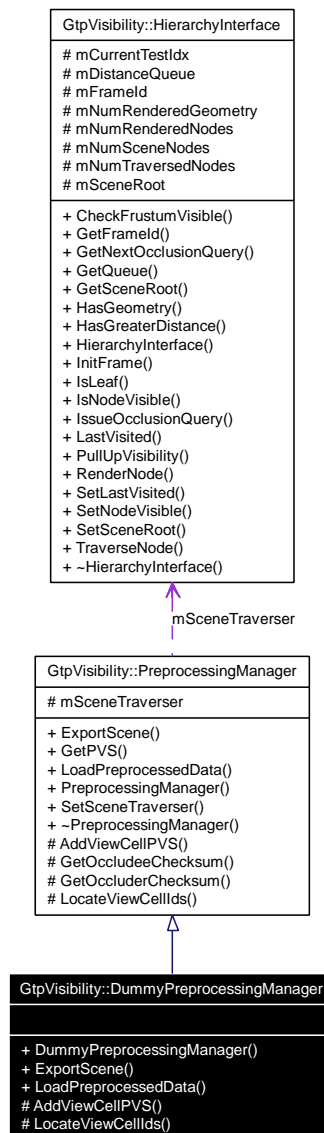
```
#include <DummyPreprocessingManager.h>
```

Inherits [GtpVisibility::PreprocessingManager](#).

Inheritance diagram for GtpVisibility::DummyPreprocessingManager:



Collaboration diagram for GtpVisibility::DummyPreprocessingManager:



Public Member Functions

- `DummyPreprocessingManager` (`HierarchyInterface *hierarchyInterface`)
- virtual bool `ExportScene` (const string filename)
- virtual bool `LoadPreprocessedData` (const string filename)

Protected Member Functions

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

9.9.1 Detailed Description

Implements a dummy interface to the external visibility preprocessing.

9.9.2 Constructor & Destructor Documentation

9.9.2.1 `GtpVisibility::DummyPreprocessingManager::DummyPreprocessingManager` ([HierarchyInterface](#) * *hierarchyInterface*)

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

9.9.3 Member Function Documentation

9.9.3.1 `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](#).

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

Load preprocessed visibility information

Implements [GtpVisibility::PreprocessingManager](#).

9.9.3.3 `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](#).

9.9.3.4 `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:

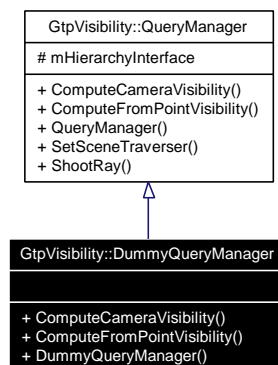
- [GtpVisibility/include/DummyPreprocessingManager.h](#)
- [GtpVisibility/src/DummyPreprocessingManager.cpp](#)

9.10 GtpVisibility::DummyQueryManager Class Reference

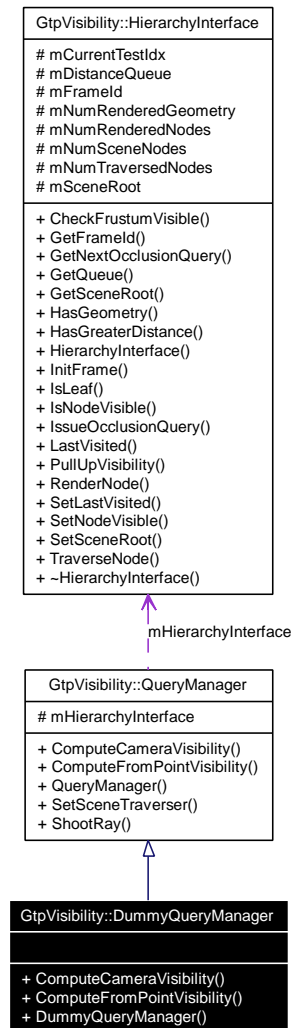
```
#include <DummyQueryManager.h>
```

Inherits [GtpVisibility::QueryManager](#).

Inheritance diagram for GtpVisibility::DummyQueryManager:



Collaboration diagram for GtpVisibility::DummyQueryManager:



Public Member Functions

- 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)
- [DummyQueryManager](#) ([HierarchyInterface](#) *hierarchyInterface)

9.10.1 Detailed Description

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

9.10.2 Constructor & Destructor Documentation

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

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

9.10.3 Member Function Documentation

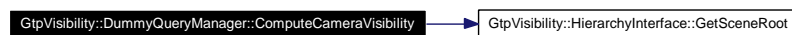
9.10.3.1 void GtpVisibility::DummyQueryManager::ComputeCameraVisibility (const [Camera](#) & *camera*, [InfoContainer](#)< [NodeInfo](#) > * *visibleNodes*, [InfoContainer](#)< [MeshInfo](#) > * *visibleGeometry*, bool *relativeVisibility* = false) [[virtual](#)]

See also:

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

Implements [GtpVisibility::QueryManager](#).

Here is the call graph for this function:



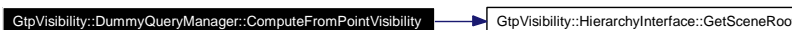
9.10.3.2 void GtpVisibility::DummyQueryManager::ComputeFromPointVisibility (const [Vector3](#) & *point*, [InfoContainer](#)< [NodeInfo](#) > * *visibleNodes*, [InfoContainer](#)< [MeshInfo](#) > * *visibleGeometry*, bool *relativeVisibility* = false) [[virtual](#)]

See also:

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

Implements [GtpVisibility::QueryManager](#).

Here is the call graph for this function:



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

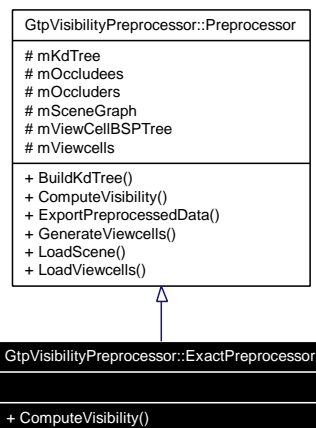
- [GtpVisibility/include/DummyQueryManager.h](#)
- [GtpVisibility/src/DummyQueryManager.cpp](#)

9.11 GtpVisibilityPreprocessor::ExactPreprocessor Class Reference

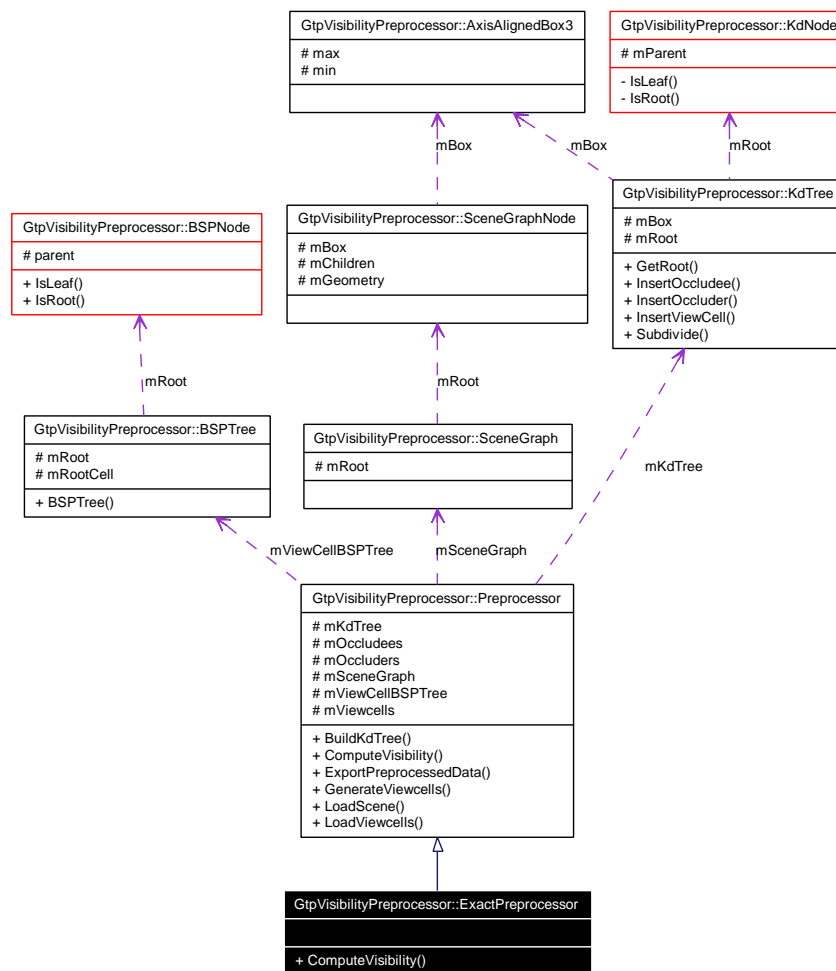
```
#include <ExactPreprocessor.h>
```

Inherits [GtpVisibilityPreprocessor::Preprocessor](#).

Inheritance diagram for GtpVisibilityPreprocessor::ExactPreprocessor:



Collaboration diagram for GtpVisibilityPreprocessor::ExactPreprocessor:



Public Member Functions

- virtual bool [ComputeVisibility](#) ()

9.11.1 Detailed Description

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

9.11.2 Member Function Documentation

9.11.2.1 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](#)

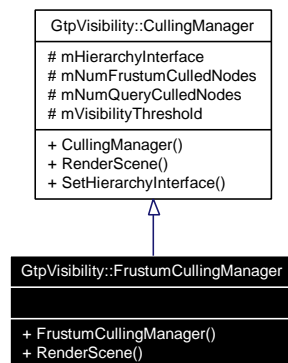
- [GtpVisibilityPreprocessor/src/ExactPreprocessor.cpp](#)

9.12 GtpVisibility::FrustumCullingManager Class Reference

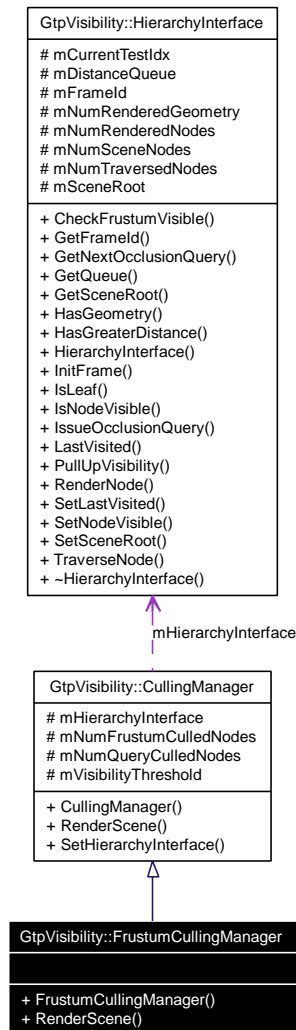
```
#include <FrustumCullingManager.h>
```

Inherits [GtpVisibility::CullingManager](#).

Inheritance diagram for GtpVisibility::FrustumCullingManager:



Collaboration diagram for GtpVisibility::FrustumCullingManager:



Public Member Functions

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

9.12.1 Detailed Description

Renders the scene, applies only view frustum culling.

9.12.2 Constructor & Destructor Documentation

9.12.2.1 GtpVisibility::FrustumCullingManager::FrustumCullingManager ([HierarchyInterface](#) * *hierarchyInterface*)

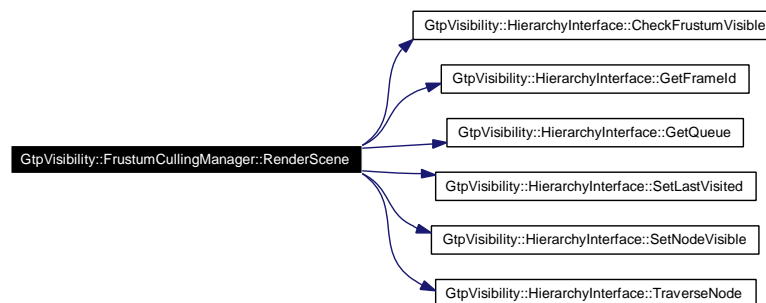
9.12.3 Member Function Documentation

9.12.3.1 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](#).

Here is the call graph for this function:



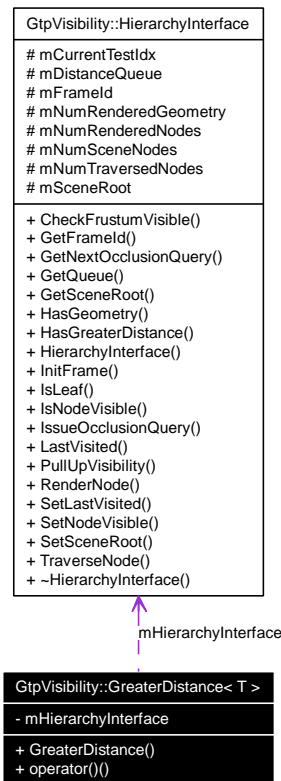
The documentation for this class was generated from the following files:

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

9.13 GtpVisibility::GreaterDistance< T > Class Template Reference

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

Collaboration diagram for GtpVisibility::GreaterDistance< T >:



Public Member Functions

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

Private Attributes

- `HierarchyInterface * mHierarchyInterface`

9.13.1 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.

9.13.2 Constructor & Destructor Documentation

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

9.13.3 Member Function Documentation

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

Here is the call graph for this function:



9.13.4 Member Data Documentation

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

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

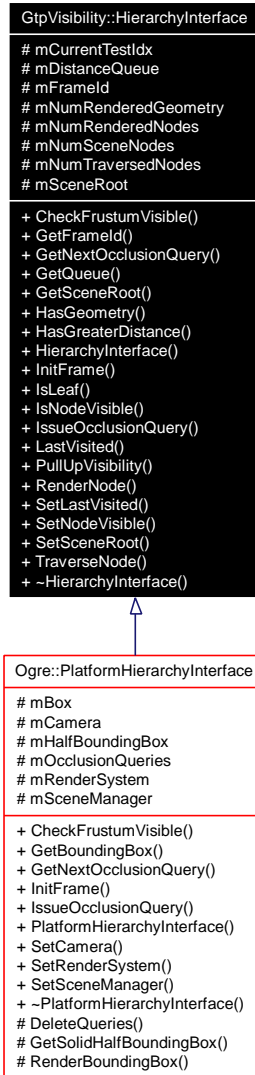
- [GtpVisibility/include/DistanceQueue.h](#)

9.14 GtpVisibility::HierarchyInterface Class Reference

```
#include <HierarchyInterface.h>
```

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

Inheritance diagram for GtpVisibility::HierarchyInterface:



Public Member Functions

- virtual bool [CheckFrustumVisible](#) ([HierarchyNode](#) *node, bool &intersects)=0
- int [GetFrameId](#) ()
- virtual [OcclusionQuery](#) * [GetNextOcclusionQuery](#) ()=0
- [DistanceQueue](#) * [GetQueue](#) ()
- [HierarchyNode](#) * [GetSceneRoot](#) () const
- virtual bool [HasGeometry](#) ([HierarchyNode](#) *node)=0
- virtual bool [HasGreaterDistance](#) ([HierarchyNode](#) *node1, [HierarchyNode](#) *node2)=0

- [HierarchyInterface](#) ()
- void [InitFrame](#) ([HierarchyNode](#) *root)
- virtual bool [IsLeaf](#) ([HierarchyNode](#) *node)=0
- virtual bool [IsNodeVisible](#) ([HierarchyNode](#) *node)=0
- virtual [OcclusionQuery](#) * [IssueOcclusionQuery](#) ([HierarchyNode](#) *node)=0
- virtual int [LastVisited](#) ([HierarchyNode](#) *node)=0
- virtual void [PullUpVisibility](#) ([HierarchyNode](#) *node)=0
- virtual void [RenderNode](#) ([HierarchyNode](#) *node)=0
- virtual void [SetLastVisited](#) ([HierarchyNode](#) *node, const int frameId)=0
- virtual void [SetNodeVisible](#) ([HierarchyNode](#) *node, const bool visible)=0
- void [SetSceneRoot](#) ([HierarchyNode](#) *root)
- virtual void [TraverseNode](#) ([HierarchyNode](#) *node)=0
- virtual [~HierarchyInterface](#) ()

Protected Attributes

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

9.14.1 Detailed Description

Class which implements a hierarchy interface for a scene hierarchy.

9.14.2 Constructor & Destructor Documentation

9.14.2.1 GtpVisibility::HierarchyInterface::HierarchyInterface ()

Default constructor.

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

9.14.3 Member Function Documentation

9.14.3.1 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

9.14.3.2 `virtual void GtpVisibility::HierarchyInterface::TraverseNode (HierarchyNode * node)`
[pure virtual]

Traverses the given node.

Parameters:

node the hierarchy node

9.14.3.3 `virtual void GtpVisibility::HierarchyInterface::RenderNode (HierarchyNode * node)`
[pure virtual]

Renders current scene node .

Parameters:

node current scene node to be rendered

9.14.3.4 `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

9.14.3.5 `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

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

Sets the root of the scene hierarchy.

Parameters:

root the hierarchy root

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

Get the root of the scene hierarchy.

Returns:

the hierarchy root

9.14.3.8 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.

Here is the call graph for this function:

**9.14.3.9 int GtpVisibility::HierarchyInterface::GetFrameId ()**

Returns current frame id.

Returns:

frame id

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

Returns the current distance queue.

Returns:

current distance queue

9.14.3.11 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

9.14.3.12 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

9.14.3.13 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](#).

9.14.3.14 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

9.14.3.15 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

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

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

9.14.3.17 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

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

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

9.14.4 Member Data Documentation

- 9.14.4.1 **unsigned int** [GtpVisibility::HierarchyInterface::mFrameId](#) [protected]
- 9.14.4.2 **int** [GtpVisibility::HierarchyInterface::mCurrentTestIdx](#) [protected]
- 9.14.4.3 **unsigned int** [GtpVisibility::HierarchyInterface::mNumSceneNodes](#) [protected]
- 9.14.4.4 **unsigned int** [GtpVisibility::HierarchyInterface::mNumTraversedNodes](#) [protected]
- 9.14.4.5 **unsigned int** [GtpVisibility::HierarchyInterface::mNumRenderedGeometry](#) [protected]
- 9.14.4.6 **unsigned int** [GtpVisibility::HierarchyInterface::mNumRenderedNodes](#) [protected]
- 9.14.4.7 **DistanceQueue*** [GtpVisibility::HierarchyInterface::mDistanceQueue](#) [protected]
- 9.14.4.8 **HierarchyNode*** [GtpVisibility::HierarchyInterface::mSceneRoot](#) [protected]

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

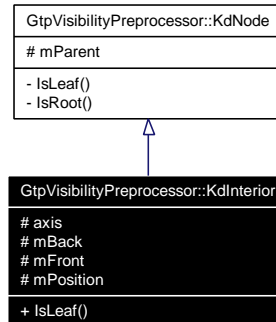
- [GtpVisibility/include/HierarchyInterface.h](#)
- [GtpVisibility/src/HierarchyInterface.cpp](#)

9.15 GtpVisibilityPreprocessor::KdInterior Class Reference

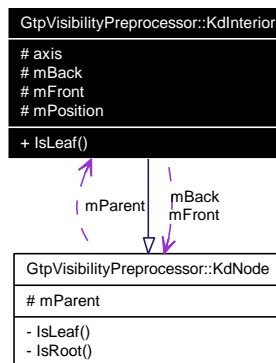
```
#include <KdTree.h>
```

Inherits [GtpVisibilityPreprocessor::KdNode](#).

Inheritance diagram for GtpVisibilityPreprocessor::KdInterior:



Collaboration diagram for GtpVisibilityPreprocessor::KdInterior:



Public Member Functions

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

Protected Attributes

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

9.15.1 Detailed Description

Implementation of the kd-tree interior node

9.15.2 Member Function Documentation

9.15.2.1 `virtual bool GtpVisibilityPreprocessor::KdInterior::IsLeaf () const` [inline, virtual]

See also:

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

Implements [GtpVisibilityPreprocessor::KdNode](#).

9.15.3 Member Data Documentation

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

splitting axis

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

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

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

back node

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

front node

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

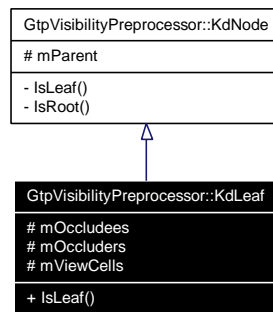
- [GtpVisibilityPreprocessor/include/KdTree.h](#)

9.16 GtpVisibilityPreprocessor::KdLeaf Class Reference

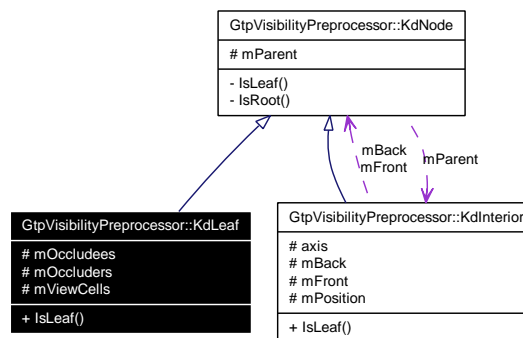
```
#include <KdTree.h>
```

Inherits [GtpVisibilityPreprocessor::KdNode](#).

Inheritance diagram for GtpVisibilityPreprocessor::KdLeaf:



Collaboration diagram for GtpVisibilityPreprocessor::KdLeaf:



Public Member Functions

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

Protected Attributes

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

9.16.1 Detailed Description

Implementation of the kd-tree leaf node

9.16.2 Member Function Documentation

9.16.2.1 `virtual bool GtpVisibilityPreprocessor::KdLeaf::IsLeaf () const` [inline, virtual]

See also:

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

Implements [GtpVisibilityPreprocessor::KdNode](#).

9.16.3 Member Data Documentation

9.16.3.1 `MeshContainer GtpVisibilityPreprocessor::KdLeaf::mOccluders` [protected]

pointers to occluders contained in this node

9.16.3.2 `MeshContainer GtpVisibilityPreprocessor::KdLeaf::mOccludees` [protected]

pointers to occludees contained in this node

9.16.3.3 `ViewCellContainer GtpVisibilityPreprocessor::KdLeaf::mViewCells` [protected]

pointers to viewcells contained in this node

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

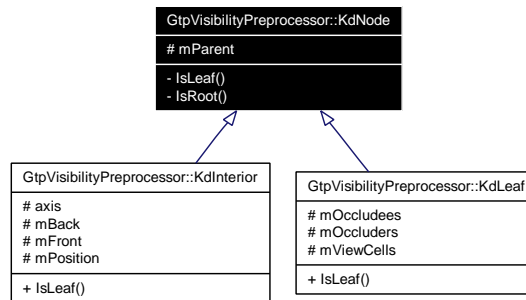
- [GtpVisibilityPreprocessor/include/KdTree.h](#)

9.17 GtpVisibilityPreprocessor::KdNode Class Reference

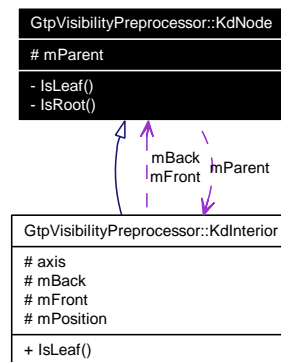
```
#include <KdTree.h>
```

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

Inheritance diagram for GtpVisibilityPreprocessor::KdNode:



Collaboration diagram for GtpVisibilityPreprocessor::KdNode:



Protected Attributes

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

Private Member Functions

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

9.17.1 Detailed Description

Abstract class for kd-tree node

9.17.2 Member Function Documentation

9.17.2.1 `virtual bool GtpVisibilityPreprocessor::KdNode::IsLeaf () const` [private, pure virtual]

Determines whether this node is a leaf or interior node

Returns:

true if leaf

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

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

Determines whether this node is the root of the tree

Returns:

true if root

9.17.3 Member Data Documentation

9.17.3.1 `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

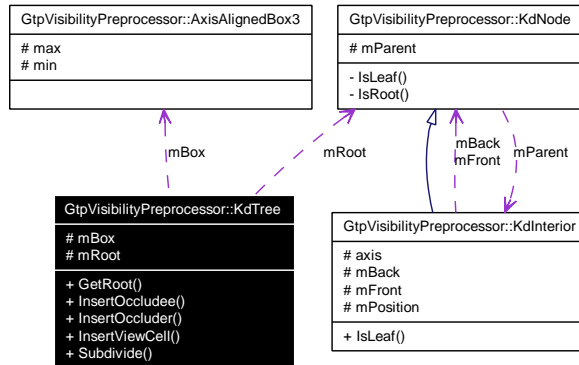
The documentation for this class was generated from the following file:

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

9.18 GtpVisibilityPreprocessor::KdTree Class Reference

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

Collaboration diagram for GtpVisibilityPreprocessor::KdTree:



Public Member Functions

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

Protected Attributes

- `AxisAlignedBox3 mBox`
bounding box of the tree root
- `KdNode * mRoot`
root of the tree

9.18.1 Detailed Description

`KdTree` for indexing scene entities - occluders/occludees/viewcells

9.18.2 Member Function Documentation

- 9.18.2.1** virtual void `GtpVisibilityPreprocessor::KdTree::InsertOccluder (Mesh * occluder)`
[inline, virtual]

Insert occluder into the tree

9.18.2.2 `virtual void GtpVisibilityPreprocessor::KdTree::InsertOccludee (Mesh * occludee)`
[inline, virtual]

Insert occludee into the tree

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

Insert view cell into the tree

9.18.2.4 `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

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

Get the root of the tree

9.18.3 Member Data Documentation

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

root of the tree

9.18.3.2 `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](#)

9.19 GtpVisibilityPreprocessor::Mesh Class Reference

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

Public Types

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

Public Member Functions

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

Protected Attributes

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

9.19.1 Detailed Description

[Mesh](#) containing polygonal patches

9.19.2 Member Typedef Documentation

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

default vertex container for [Mesh](#)

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

default patch container for [Mesh](#)

9.19.3 Constructor & Destructor Documentation

9.19.3.1 [GtpVisibilityPreprocessor::Mesh::Mesh](#) () [inline]

Default constructor.

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

Constructor with container preallocation.

9.19.4 Member Data Documentation

9.19.4.1 [VertexContainer GtpVisibilityPreprocessor::Mesh::mVertices](#) [protected]

Vertices forming the mesh

9.19.4.2 [PatchContainer GtpVisibilityPreprocessor::Mesh::mPatches](#) [protected]

Patches forming the mesh

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

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

9.20 GtpVisibility::MeshInfo Class Reference

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

Public Member Functions

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

Protected Attributes

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

9.20.1 Detailed Description

Class storing the visibility information of a mesh.

9.20.2 Constructor & Destructor Documentation

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

9.20.3 Member Data Documentation

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

Pointer to the mesh.

9.20.3.2 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](#)

9.21 GtpVisibility::NodeInfo Class Reference

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

Public Member Functions

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

Protected Attributes

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

9.21.1 Detailed Description

Class storing the visibility information of a scene node.

9.21.2 Constructor & Destructor Documentation

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

9.21.3 Member Data Documentation

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

pointer to the scene node

9.21.3.2 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:

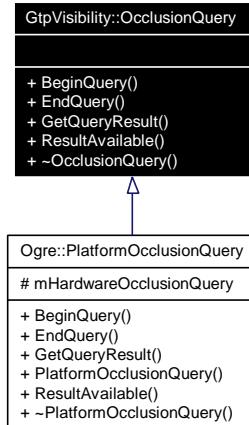
- [GtpVisibility/include/VisibilityInfo.h](#)

9.22 GtpVisibility::OcclusionQuery Class Reference

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

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

Inheritance diagram for GtpVisibility::OcclusionQuery:



Public Member Functions

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

9.22.1 Detailed Description

This class is an abstract interface for occlusion queries.

9.22.2 Constructor & Destructor Documentation

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

9.22.3 Member Function Documentation

9.22.3.1 virtual unsigned int [GtpVisibility::OcclusionQuery::GetQueryResult](#) () const [`pure virtual`]

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

Returns:

number of visible pixels

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

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

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

Returns:

if result is available

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

9.22.3.3 `virtual void GtpVisibility::OcclusionQuery::BeginQuery () const` [pure virtual]

Begins occlusion query.

Remarks:

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

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

9.22.3.4 `virtual void GtpVisibility::OcclusionQuery::EndQuery () const` [pure virtual]

Ends occlusion query.

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

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

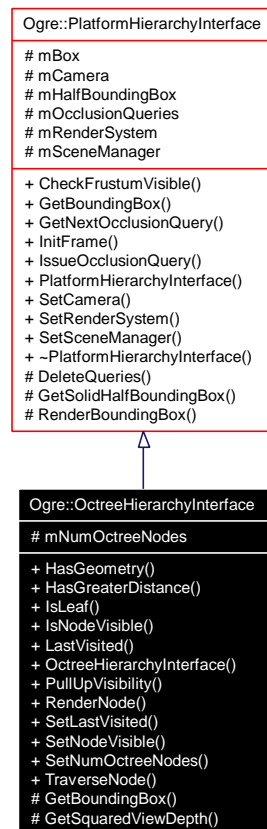
- [GtpVisibility/include/OcclusionQuery.h](#)

9.23 Ogre::OctreeHierarchyInterface Class Reference

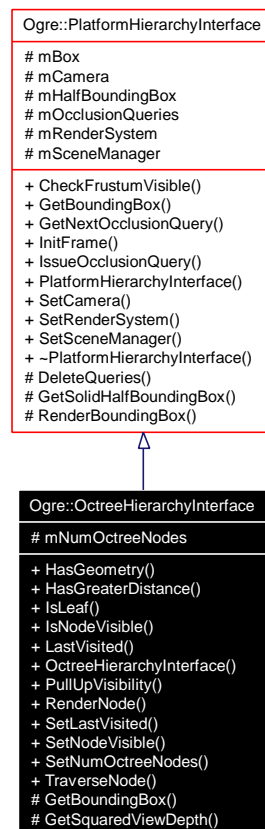
```
#include <OgreOctreeHierarchyInterface.h>
```

Inherits [Ogre::PlatformHierarchyInterface](#).

Inheritance diagram for `Ogre::OctreeHierarchyInterface`:



Collaboration diagram for `Ogre::OctreeHierarchyInterface`:



Public Member Functions

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

Protected Member Functions

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

Protected Attributes

- unsigned int [mNumOctreeNodes](#)

9.23.1 Detailed Description

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

9.23.2 Constructor & Destructor Documentation

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

9.23.3 Member Function Documentation

9.23.3.1 `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

9.23.3.2 `void Ogre::OctreeHierarchyInterface::PullUpVisibility` (`GtpVisibility::HierarchyNode * node`)

Gets the given option for the scene traverser.

Remarks:

See `setOption`

9.23.3.3 `void Ogre::OctreeHierarchyInterface::TraverseNode` (`GtpVisibility::HierarchyNode * node`)

Traverses given node.

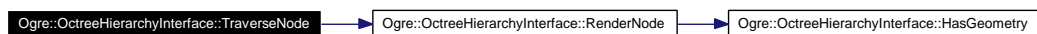
Parameters:

node current node

Remarks:

pushes children on distance queue

Here is the call graph for this function:



9.23.3.4 void Ogre::OctreeHierarchyInterface::RenderNode (GtpVisibility::HierarchyNode * node)

Here is the call graph for this function:



9.23.3.5 bool Ogre::OctreeHierarchyInterface::IsLeaf (GtpVisibility::HierarchyNode * node)

9.23.3.6 bool Ogre::OctreeHierarchyInterface::HasGeometry (GtpVisibility::HierarchyNode * node)

9.23.3.7 bool Ogre::OctreeHierarchyInterface::HasGreaterDistance (GtpVisibility::HierarchyNode * node1, GtpVisibility::HierarchyNode * node2)

Here is the call graph for this function:



9.23.3.8 void Ogre::OctreeHierarchyInterface::SetNodeVisible (GtpVisibility::HierarchyNode * node, const bool visible)

9.23.3.9 void Ogre::OctreeHierarchyInterface::SetLastVisited (GtpVisibility::HierarchyNode * node, const int frameId)

9.23.3.10 bool Ogre::OctreeHierarchyInterface::IsNodeVisible (GtpVisibility::HierarchyNode * node)

9.23.3.11 int Ogre::OctreeHierarchyInterface::LastVisited (GtpVisibility::HierarchyNode * node)

9.23.3.12 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](#).

9.23.3.13 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

9.23.4 Member Data Documentation**9.23.4.1 unsigned int [Ogre::OctreeHierarchyInterface::mNumOctreeNode](#)s** [protected]

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

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

9.24 GtpVisibilityPreprocessor::Patch Class Reference

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

Public Member Functions

- [Patch \(\)](#)

Protected Attributes

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

9.24.1 Detailed Description

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

9.24.2 Constructor & Destructor Documentation

9.24.2.1 [GtpVisibilityPreprocessor::Patch::Patch \(\)](#) [inline]

9.24.3 Member Data Documentation

9.24.3.1 `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](#)

9.25 GtpVisibilityPreprocessor::Plane3 Class Reference

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

Public Attributes

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

9.25.1 Detailed Description

3D Plane

9.25.2 Member Data Documentation

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

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

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

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

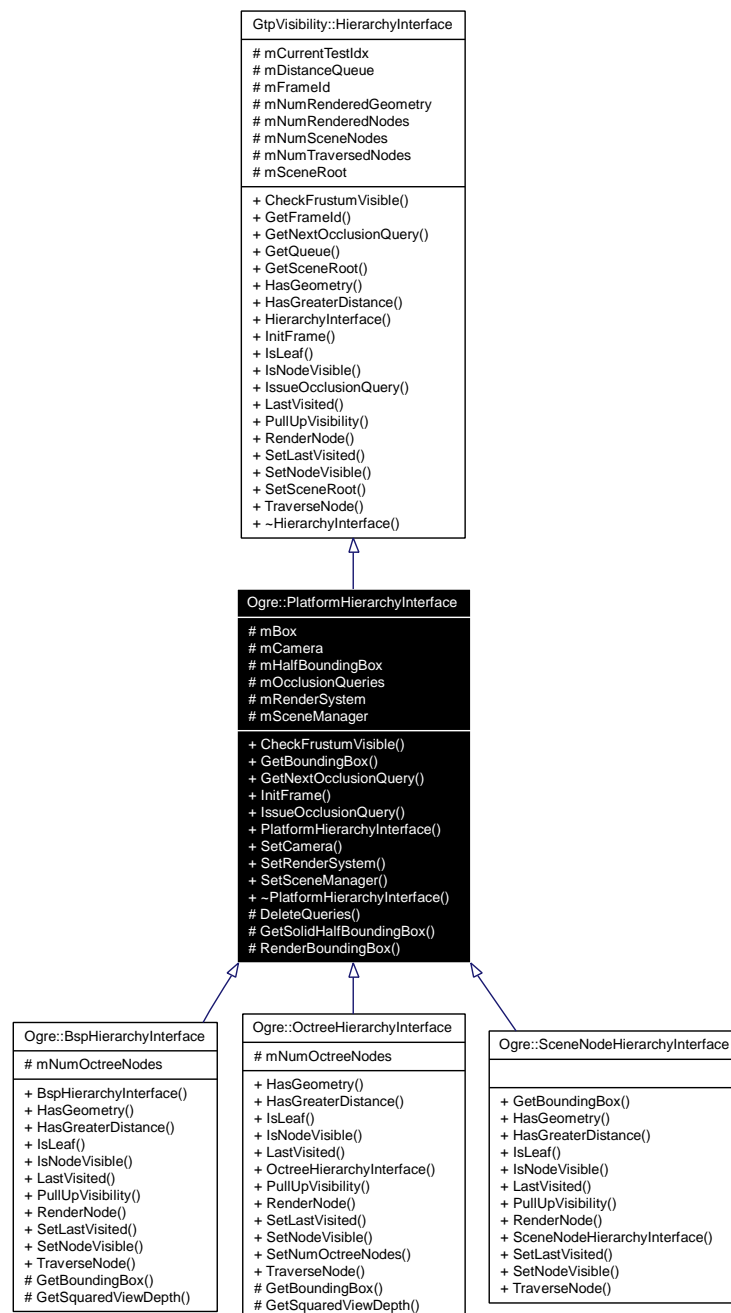
9.26 Ogre::PlatformHierarchyInterface Class Reference

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

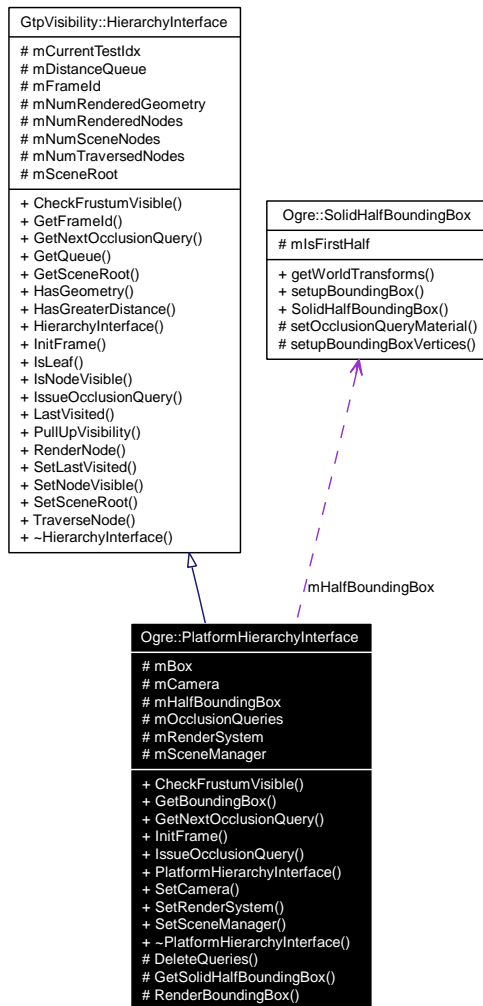
Inherits [GtpVisibility::HierarchyInterface](#).

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

Inheritance diagram for [Ogre::PlatformHierarchyInterface](#):



Collaboration diagram for Ogre::PlatformHierarchyInterface:



Public Member Functions

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

Protected Member Functions

- void [DeleteQueries](#) ()

- [SolidHalfBoundingBox](#) * [GetSolidHalfBoundingBox](#) (int half)
- void [RenderBoundingBox](#) ([AxisAlignedBox](#) *box)

Protected Attributes

- [AxisAlignedBox](#) mBox
- [Camera](#) * mCamera
- [SolidHalfBoundingBox](#) * mHalfBoundingBox [2]
- std::vector< [PlatformOcclusionQuery](#) * > mOcclusionQueries
- [RenderSystem](#) * mRenderSystem
- [SceneManager](#) * mSceneManager

9.26.1 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

9.26.2 Constructor & Destructor Documentation

9.26.2.1 [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

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

Here is the call graph for this function:



9.26.3 Member Function Documentation

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

Returns next available occlusion query or creates new one.

Returns:

the next occlusion query

Implements [GtpVisibility::HierarchyInterface](#).

9.26.3.2 void Ogre::PlatformHierarchyInterface::SetCamera (Camera * cam)

Sets the current camera.

Parameters:

cam the current camera

9.26.3.3 void Ogre::PlatformHierarchyInterface::InitFrame (GtpVisibility::HierarchyNode * root, Ogre::Camera * cam)

Initialises this scene traverser for the current frame.

Parameters:

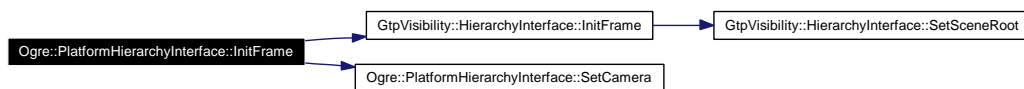
root root of the hierarchy

cam current camera

Remarks:

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

Here is the call graph for this function:



9.26.3.4 bool Ogre::PlatformHierarchyInterface::CheckFrustumVisible (GtpVisibility::HierarchyNode * node, bool & intersects)

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

Here is the call graph for this function:



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

Sets pointer to the current scene manager.

Parameters:

sm the scene manager

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

Sets pointer to the current render system

Parameters:

rsys the rendersystem

9.26.3.7 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](#).

9.26.3.8 GtpVisibility::OcclusionQuery * Ogre::PlatformHierarchyInterface::IssueOcclusionQuery (GtpVisibility::HierarchyNode * node)

Issue a occlusion query for this node.

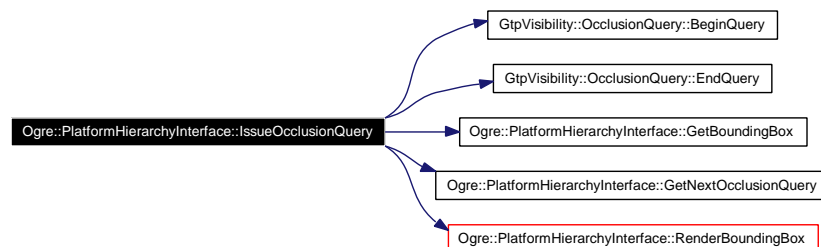
Parameters:

node the current hierarchy node

Returns:

occlusion query for this node

Here is the call graph for this function:



9.26.3.9 void Ogre::PlatformHierarchyInterface::DeleteQueries () [protected]

Deletes all occlusion queries.

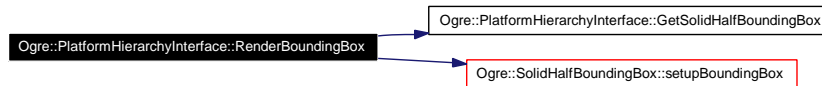
9.26.3.10 void `Ogre::PlatformHierarchyInterface::RenderBoundingBox` (`AxisAlignedBox * box`) [protected]

Renders bounding box of specified node.

Parameters:

box the bounding box of the scene node to be rendered

Here is the call graph for this function:



9.26.3.11 `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)

9.26.4 Member Data Documentation

9.26.4.1 `SolidHalfBoundingBox*` `Ogre::PlatformHierarchyInterface::mHalfBoundingBox`[2] [protected]

two halves of an aabb.

9.26.4.2 `SceneManager*` `Ogre::PlatformHierarchyInterface::mSceneManager` [protected]

9.26.4.3 `RenderSystem*` `Ogre::PlatformHierarchyInterface::mRenderSystem` [protected]

9.26.4.4 `Camera*` `Ogre::PlatformHierarchyInterface::mCamera` [protected]

9.26.4.5 `AxisAlignedBox` `Ogre::PlatformHierarchyInterface::mBox` [protected]

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

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

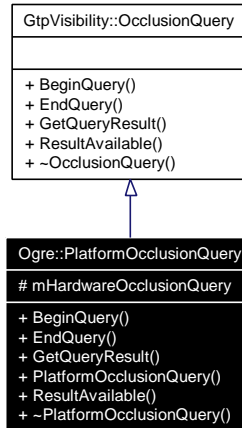
- [Ogre/include/OgrePlatformHierarchyInterface.h](#)
- [Ogre/src/OgrePlatformHierarchyInterface.cpp](#)

9.27 Ogre::PlatformOcclusionQuery Class Reference

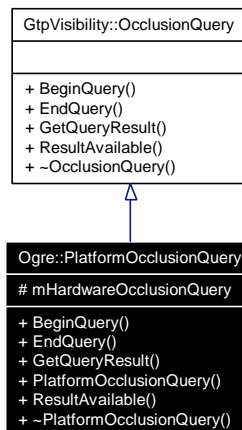
```
#include <OgrePlatformOcclusionQuery.h>
```

Inherits [GtpVisibility::OcclusionQuery](#).

Inheritance diagram for `Ogre::PlatformOcclusionQuery`:



Collaboration diagram for `Ogre::PlatformOcclusionQuery`:



Public Member Functions

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

Protected Attributes

- HardwareOcclusionQuery * [mHardwareOcclusionQuery](#)

9.27.1 Detailed Description

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

Remarks:

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

9.27.2 Constructor & Destructor Documentation

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

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

9.27.3 Member Function Documentation

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

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

Returns:

number of visible pixels

Implements [GtpVisibility::OcclusionQuery](#).

9.27.3.2 `bool Ogre::PlatformOcclusionQuery::ResultAvailable () const` [virtual]

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

Returns:

if result is available

Implements [GtpVisibility::OcclusionQuery](#).

9.27.3.3 `void Ogre::PlatformOcclusionQuery::BeginQuery () const` [virtual]

Begins occlusion query.

Remarks:

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

Implements [GtpVisibility::OcclusionQuery](#).

9.27.3.4 `void Ogre::PlatformOcclusionQuery::EndQuery () const` [virtual]

Ends occlusion query.

Implements [GtpVisibility::OcclusionQuery](#).

9.27.4 Member Data Documentation

9.27.4.1 HardwareOcclusionQuery* [Ogre::PlatformOcclusionQuery::mHardwareOcclusionQuery](#) [protected]

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

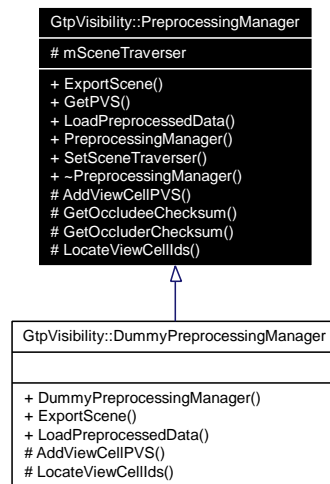
- [Ogre/include/OgrePlatformOcclusionQuery.h](#)
- [Ogre/src/OgrePlatformOcclusionQuery.cpp](#)

9.28 GtpVisibility::PreprocessingManager Class Reference

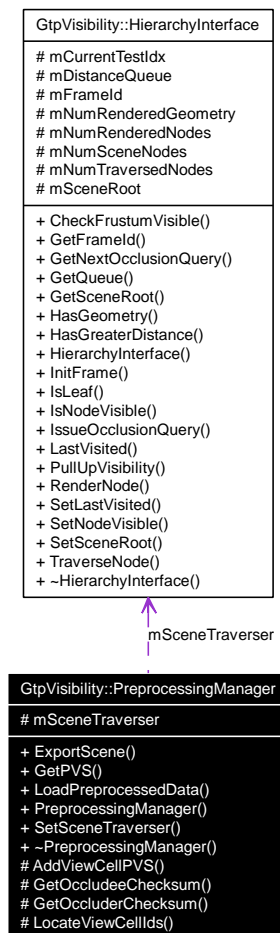
```
#include <PreprocessingManager.h>
```

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

Inheritance diagram for GtpVisibility::PreprocessingManager:



Collaboration diagram for GtpVisibility::PreprocessingManager:



Public Member Functions

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

Protected Member Functions

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

Protected Attributes

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

9.28.1 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).

9.28.2 Constructor & Destructor Documentation

9.28.2.1 [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.

9.28.2.2 [GtpVisibility::PreprocessingManager::~~PreprocessingManager](#) () [virtual]

Destructor which deletes all data describing static scene visibility.

9.28.3 Member Function Documentation

9.28.3.1 `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](#).

9.28.3.2 `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 successful 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 successful.

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

9.28.3.3 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:

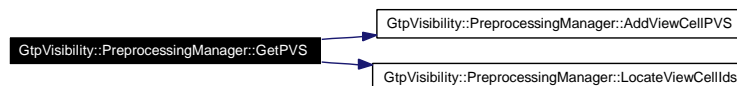
point the center point of the spherical neighborhood

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 leafs 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.

Here is the call graph for this function:



9.28.3.4 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.

9.28.3.5 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 pre-processed data matches the current scene graph -> \$\$ could be moved to VisibilitySceneTraverser

9.28.3.6 `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 pre-processed data matches the current scene graph -> \$\$ could be moved to VisibilitySceneTraverser

9.28.3.7 `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](#).

9.28.3.8 `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](#).

9.28.4 Member Data Documentation

9.28.4.1 `HierarchyInterface* GtpVisibility::PreprocessingManager::mSceneTraverser` [protected]

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

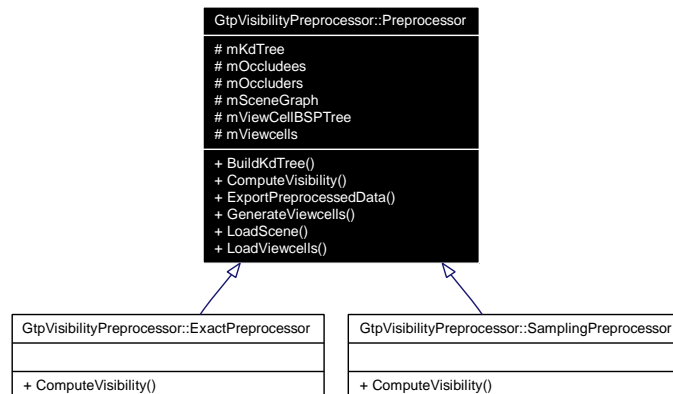
- [GtpVisibility/include/PreprocessingManager.h](#)
- [GtpVisibility/src/PreprocessingManager.cpp](#)

9.29 GtpVisibilityPreprocessor::Preprocessor Class Reference

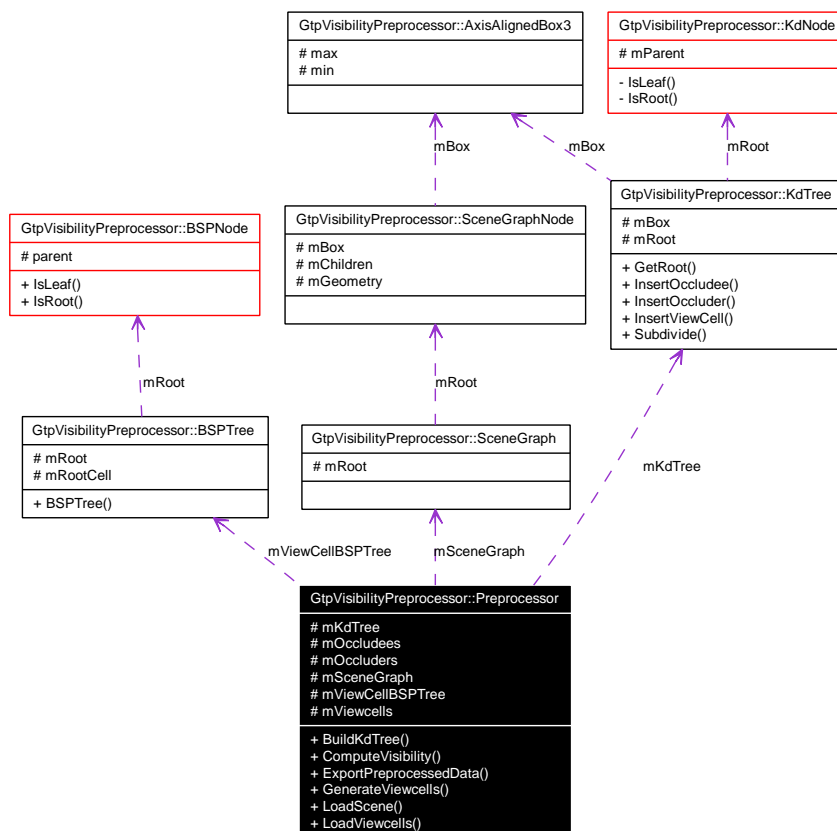
```
#include <Preprocessor.h>
```

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

Inheritance diagram for GtpVisibilityPreprocessor::Preprocessor:



Collaboration diagram for GtpVisibilityPreprocessor::Preprocessor:



Public Member Functions

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

Protected Attributes

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

9.29.1 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.

9.29.2 Member Function Documentation

9.29.2.1 bool [GtpVisibilityPreprocessor::Preprocessor::LoadScene](#) (const string *filename*) [virtual]

Load the input scene.

Parameters:

filename file to load

Returns:

true on success

9.29.2.2 `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 assume 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

9.29.2.3 `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.

9.29.2.4 `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

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

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

Here is the call graph for this function:



9.29.2.6 `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::Sampling-Preprocessor](#).

9.29.3 Member Data Documentation

9.29.3.1 [SceneGraph*](#) [GtpVisibilityPreprocessor::Preprocessor::mSceneGraph](#) [protected]

scene graph loaded from file

9.29.3.2 [BSPTree*](#) [GtpVisibilityPreprocessor::Preprocessor::mViewCellBSPTree](#) [protected]

BSP tree representing the viewcells.

9.29.3.3 [KdTree*](#) [GtpVisibilityPreprocessor::Preprocessor::mKdTree](#) [protected]

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

9.29.3.4 [MeshContainer](#) [GtpVisibilityPreprocessor::Preprocessor::mOccluders](#) [protected]

list of all loaded occluders

9.29.3.5 [MeshContainer](#) [GtpVisibilityPreprocessor::Preprocessor::mOccludees](#) [protected]

list of all loaded occludees

9.29.3.6 [ViewCellContainer](#) [GtpVisibilityPreprocessor::Preprocessor::mViewcells](#) [protected]

list of all loaded/generated viewcells

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

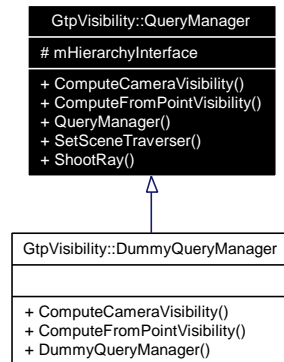
- [GtpVisibilityPreprocessor/include/Preprocessor.h](#)
- [GtpVisibilityPreprocessor/src/Preprocessor.cpp](#)

9.30 GtpVisibility::QueryManager Class Reference

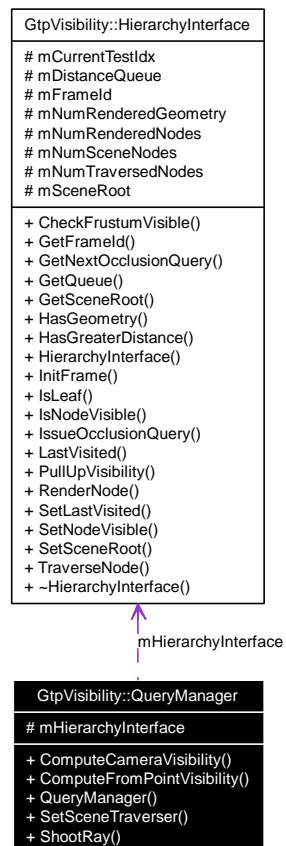
```
#include <QueryManager.h>
```

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

Inheritance diagram for GtpVisibility::QueryManager:



Collaboration diagram for GtpVisibility::QueryManager:



Public Member Functions

- 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
- [QueryManager](#) ([HierarchyInterface](#) *hierarchyInterface)
- void [SetSceneTraverser](#) ([HierarchyInterface](#) *hierarchyInterface)
- virtual bool [ShootRay](#) (const [Ray](#) &ray, std::vector< [Mesh](#) * > *visibleMeshes, bool isGlobal-Line=false)

Protected Attributes

- [HierarchyInterface](#) * mHierarchyInterface

9.30.1 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 buffering or based purely on occlusion queries. Note that the actual implementation can also exploit the output of the visibility preprocessor.

9.30.2 Constructor & Destructor Documentation

9.30.2.1 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.

9.30.3 Member Function Documentation

9.30.3.1 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](#).

9.30.3.2 `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](#).

9.30.3.3 `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.

9.30.3.4 `void GtpVisibility::QueryManager::SetSceneTraverser (HierarchyInterface * hierarchyInterface)`

Sets the scene traverser.

Remarks:

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

9.30.4 Member Data Documentation

9.30.4.1 `HierarchyInterface* GtpVisibility::QueryManager::mHierarchyInterface [protected]`

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

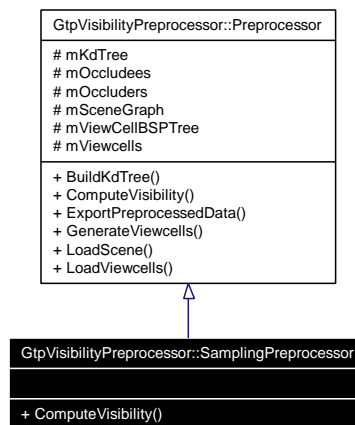
- [GtpVisibility/include/QueryManager.h](#)
- [GtpVisibility/src/QueryManager.cpp](#)

9.31 GtpVisibilityPreprocessor::SamplingPreprocessor Class Reference

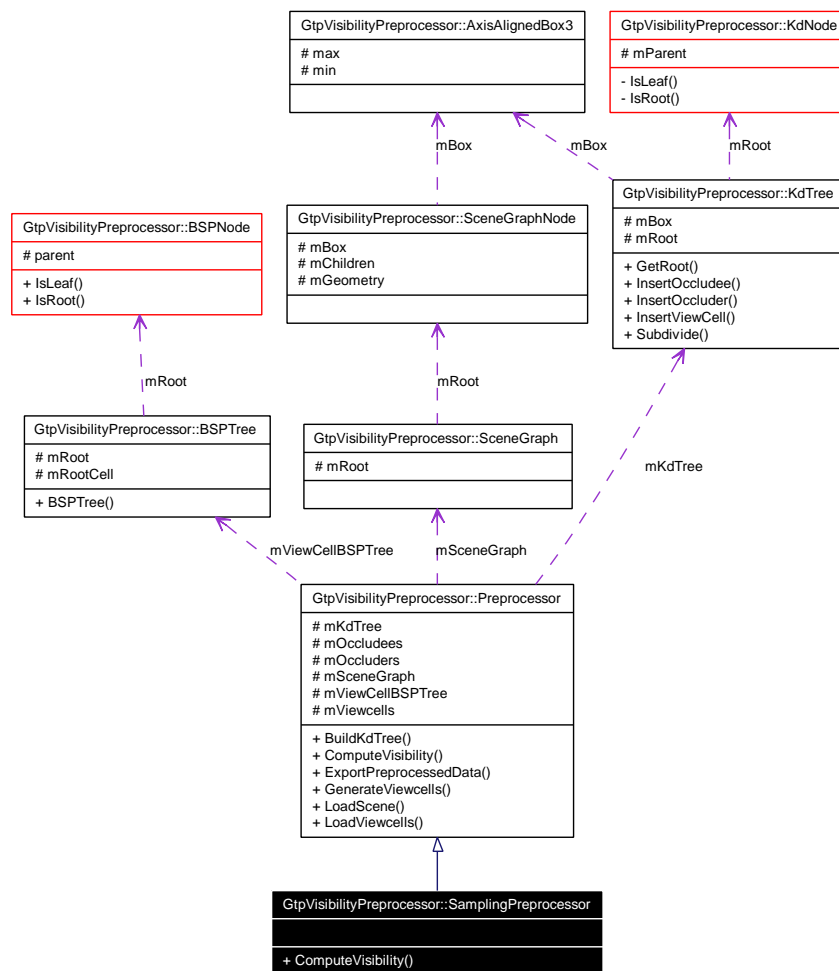
```
#include <SamplingPreprocessor.h>
```

Inherits [GtpVisibilityPreprocessor::Preprocessor](#).

Inheritance diagram for GtpVisibilityPreprocessor::SamplingPreprocessor:



Collaboration diagram for GtpVisibilityPreprocessor::SamplingPreprocessor:



Public Member Functions

- virtual bool [ComputeVisibility](#) ()

9.31.1 Detailed Description

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

9.31.2 Member Function Documentation

9.31.2.1 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](#).

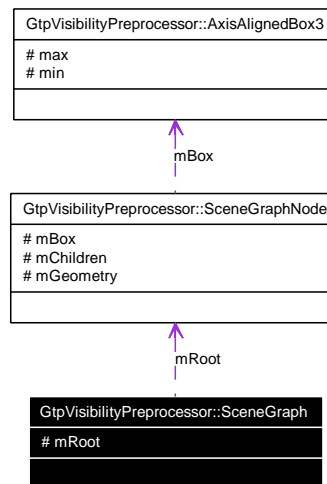
The documentation for this class was generated from the following files:

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

9.32 GtpVisibilityPreprocessor::SceneGraph Class Reference

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

Collaboration diagram for GtpVisibilityPreprocessor::SceneGraph:



Protected Attributes

- `SceneGraphNode * mRoot`

9.32.1 Detailed Description

Scene graph class

9.32.2 Member Data Documentation

9.32.2.1 `SceneGraphNode* GtpVisibilityPreprocessor::SceneGraph::mRoot` [protected]

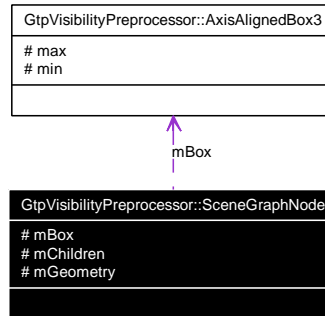
The documentation for this class was generated from the following file:

- `GtpVisibilityPreprocessor/include/SceneGraph.h`

9.33 GtpVisibilityPreprocessor::SceneGraphNode Class Reference

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

Collaboration diagram for GtpVisibilityPreprocessor::SceneGraphNode:



Protected Attributes

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

9.33.1 Detailed Description

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

9.33.2 Member Data Documentation

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

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

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

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

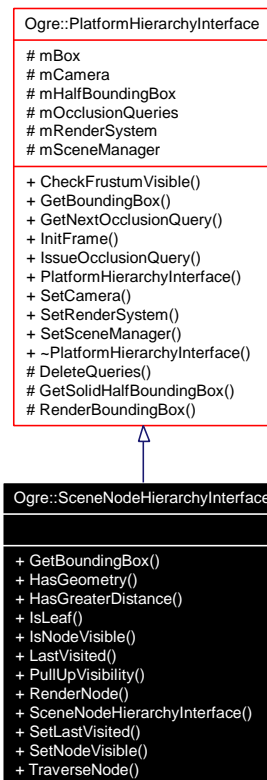
- GtpVisibilityPreprocessor/include/SceneGraph.h

9.34 Ogre::SceneNodeHierarchyInterface Class Reference

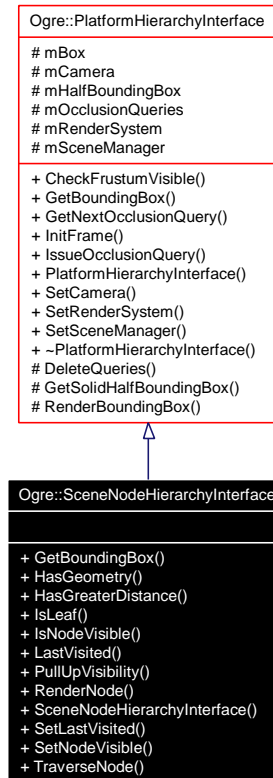
```
#include <OgreSceneNodeHierarchyInterface.h>
```

Inherits [Ogre::PlatformHierarchyInterface](#).

Inheritance diagram for Ogre::SceneNodeHierarchyInterface:



Collaboration diagram for Ogre::SceneNodeHierarchyInterface:



Public Member Functions

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

9.34.1 Detailed Description

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

9.34.2 Constructor & Destructor Documentation

9.34.2.1 Ogre::SceneNodeHierarchyInterface::SceneNodeHierarchyInterface (SceneManager * *sm*, RenderSystem * *rsys*)

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

Parameters:

sm current scene manager

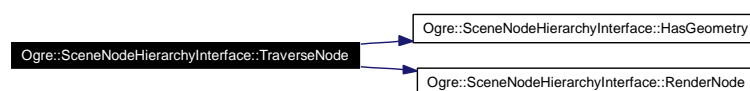
rsys current render system

9.34.3 Member Function Documentation

9.34.3.1 bool Ogre::SceneNodeHierarchyInterface::IsLeaf (GtpVisibility::HierarchyNode * *node*)

9.34.3.2 void Ogre::SceneNodeHierarchyInterface::TraverseNode (GtpVisibility::HierarchyNode * *node*)

Here is the call graph for this function:



- 9.34.3.3 `void Ogre::SceneNodeHierarchyInterface::RenderNode (GtpVisibility::HierarchyNode * node)`
- 9.34.3.4 `void Ogre::SceneNodeHierarchyInterface::PullUpVisibility (GtpVisibility::HierarchyNode * node)`
- 9.34.3.5 `bool Ogre::SceneNodeHierarchyInterface::HasGeometry (GtpVisibility::HierarchyNode * node)`
- 9.34.3.6 `bool Ogre::SceneNodeHierarchyInterface::HasGreaterDistance (GtpVisibility::HierarchyNode * node1, GtpVisibility::HierarchyNode * node2)`
- 9.34.3.7 `void Ogre::SceneNodeHierarchyInterface::SetNodeVisible (GtpVisibility::HierarchyNode * node, const bool visible)`
- 9.34.3.8 `void Ogre::SceneNodeHierarchyInterface::SetLastVisited (GtpVisibility::HierarchyNode * node, const int frameId)`
- 9.34.3.9 `bool Ogre::SceneNodeHierarchyInterface::IsNodeVisible (GtpVisibility::HierarchyNode * node)`
- 9.34.3.10 `int Ogre::SceneNodeHierarchyInterface::LastVisited (GtpVisibility::HierarchyNode * node)`
- 9.34.3.11 `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](#)

9.35 Ogre::SolidHalfBoundingBox Class Reference

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

Public Member Functions

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

Protected Member Functions

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

Protected Attributes

- bool [mIsFirstHalf](#)

9.35.1 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.

9.35.2 Constructor & Destructor Documentation

9.35.2.1 Ogre::SolidHalfBoundingBox::SolidHalfBoundingBox (bool *isFirstHalf*)

Here is the call graph for this function:



9.35.3 Member Function Documentation

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

Here is the call graph for this function:



9.35.3.2 void Ogre::SolidHalfBoundingBox::getWorldTransforms (Matrix4 * *xform*) const

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

9.35.3.3 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.

9.35.3.4 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

9.35.4 Member Data Documentation**9.35.4.1 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:

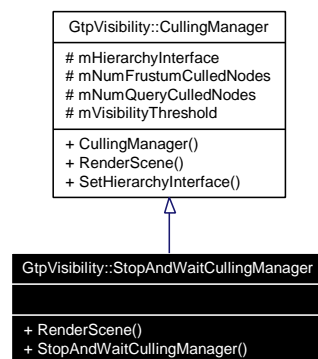
- [Ogre/include/OgreSolidHalfBoundingBox.h](#)
- [Ogre/src/OgreSolidHalfBoundingBox.cpp](#)

9.36 GtpVisibility::StopAndWaitCullingManager Class Reference

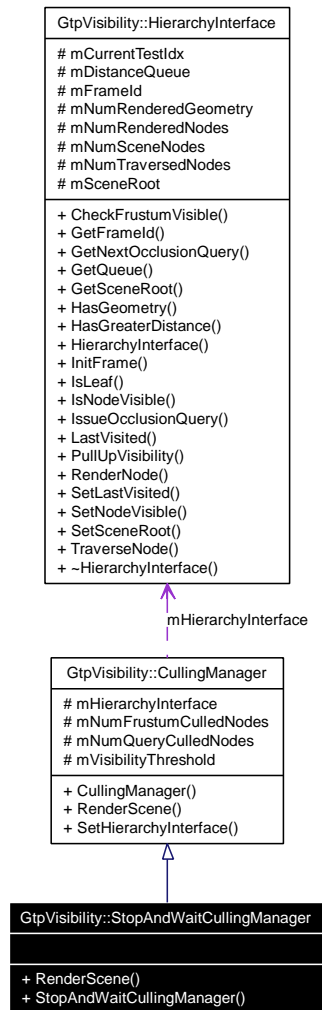
```
#include <StopAndWaitCullingManager.h>
```

Inherits [GtpVisibility::CullingManager](#).

Inheritance diagram for GtpVisibility::StopAndWaitCullingManager:



Collaboration diagram for GtpVisibility::StopAndWaitCullingManager:



Public Member Functions

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

9.36.1 Detailed Description

Renders the scene with the hierarchical stop and wait algorithm.

9.36.2 Constructor & Destructor Documentation

9.36.2.1 GtpVisibility::StopAndWaitCullingManager::StopAndWaitCullingManager (HierarchyInterface * hierarchyInterface)

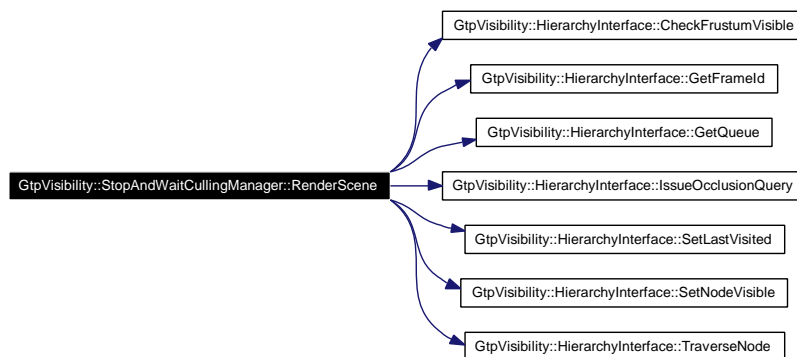
9.36.3 Member Function Documentation

9.36.3.1 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](#).

Here is the call graph for this function:



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

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

9.37 GtpVisibilityPreprocessor::Vector3 Class Reference

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

Public Attributes

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

9.37.1 Detailed Description

3D vector

9.37.2 Member Data Documentation

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

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

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

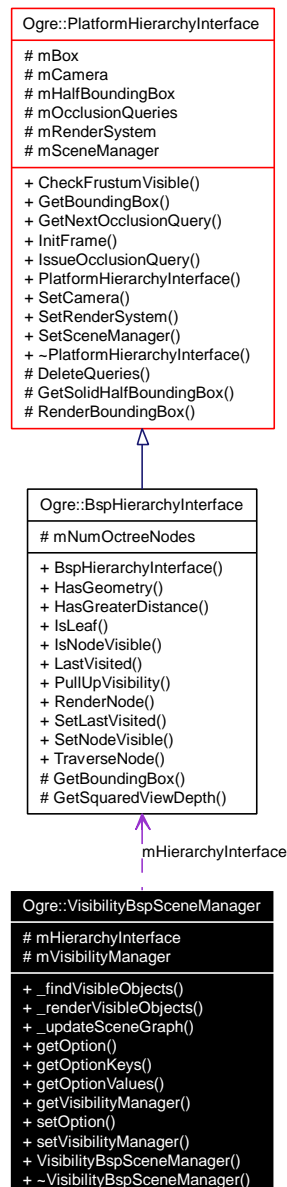
The documentation for this class was generated from the following file:

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

9.38 Ogre::VisibilityBspSceneManager Class Reference

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

Collaboration diagram for Ogre::VisibilityBspSceneManager:



Public Member Functions

- void [_findVisibleObjects](#) (Camera *cam, bool onlyShadowCasters)
- void [_renderVisibleObjects](#) ()
- void [_updateSceneGraph](#) (Camera *cam)
- virtual bool [getOption](#) (const String &, void *)
- bool [getOptionKeys](#) (StringVector &refKeys)

- bool [getOptionValues](#) (const String &key, StringVector &refValueList)
- GtpVisibility::Manager * [getVisibilityManager](#) (void)
- virtual bool [setOption](#) (const String &, const void *)
- void [setVisibilityManager](#) (GtpVisibility::Manager *visManager)
- [VisibilityBspSceneManager](#) (GtpVisibility::Manager *visManager)
- [~VisibilityBspSceneManager](#) ()

Protected Attributes

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

9.38.1 Detailed Description

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

9.38.2 Constructor & Destructor Documentation

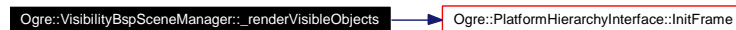
9.38.2.1 [Ogre::VisibilityBspSceneManager::VisibilityBspSceneManager](#) (GtpVisibility::Manager * visManager)

9.38.2.2 [Ogre::VisibilityBspSceneManager::~~VisibilityBspSceneManager](#) ()

9.38.3 Member Function Documentation

9.38.3.1 void [Ogre::VisibilityBspSceneManager::_renderVisibleObjects](#) ()

Here is the call graph for this function:



9.38.3.2 void [Ogre::VisibilityBspSceneManager::_findVisibleObjects](#) ([Camera](#) * cam, bool onlyShadowCasters)

9.38.3.3 void [Ogre::VisibilityBspSceneManager::_updateSceneGraph](#) ([Camera](#) * cam)

Here is the call graph for this function:



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

Sets the given option for the SceneManager

Remarks:

Options are: "Algorithm", int *;

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

Gets the given option for the Scene Manager.

Remarks:

See setOption

9.38.3.6 bool Ogre::VisibilityBspSceneManager::getOptionValues (const String & key, StringVector & refValueList)**9.38.3.7 bool Ogre::VisibilityBspSceneManager::getOptionKeys (StringVector & refKeys)****9.38.3.8 void Ogre::VisibilityBspSceneManager::setVisibilityManager (GtpVisibility::Manager * visManager)**

Sets the visibility manager.

Parameters:

visManager the visibility manager

9.38.3.9 GtpVisibility::Manager * Ogre::VisibilityBspSceneManager::getVisibilityManager (void)

See set.

9.38.4 Member Data Documentation**9.38.4.1 BspHierarchyInterface* Ogre::VisibilityBspSceneManager::mHierarchyInterface**
[protected]**9.38.4.2 GtpVisibility::Manager* Ogre::VisibilityBspSceneManager::mVisibilityManager**
[protected]

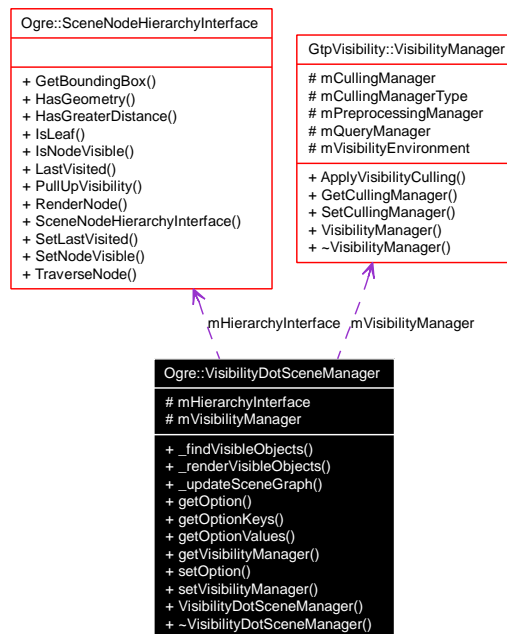
The documentation for this class was generated from the following files:

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

9.39 Ogre::VisibilityDotSceneManager Class Reference

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

Collaboration diagram for Ogre::VisibilityDotSceneManager:



Public Member Functions

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

Protected Attributes

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

9.39.1 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

9.39.2 Constructor & Destructor Documentation

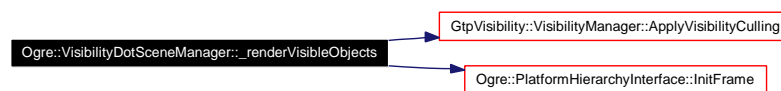
9.39.2.1 `Ogre::VisibilityDotSceneManager::VisibilityDotSceneManager` ([GtpVisibility::VisibilityManager](#) * *visManager*)

9.39.2.2 `Ogre::VisibilityDotSceneManager::~~VisibilityDotSceneManager` ()

9.39.3 Member Function Documentation

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

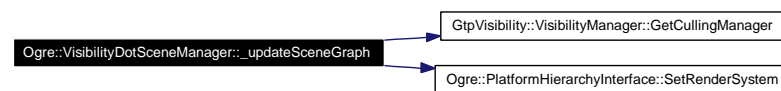
Here is the call graph for this function:



9.39.3.2 `void Ogre::VisibilityDotSceneManager::_findVisibleObjects` ([Camera](#) * *cam*, `bool` *onlyShadowCasters*)

9.39.3.3 `void Ogre::VisibilityDotSceneManager::_updateSceneGraph` ([Camera](#) * *cam*)

Here is the call graph for this function:



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

Sets the given option for the SceneManager

Remarks:

Options are: "Algorithm", int *;

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

Gets the given option for the Scene VisibilityManager.

Remarks:

See setOption

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

9.39.3.7 `bool Ogre::VisibilityDotSceneManager::getOptionKeys (StringVector & refKeys)`

9.39.3.8 `void Ogre::VisibilityDotSceneManager::setVisibilityManager (GtpVisibility::VisibilityManager * visManager)`

Sets the visibility manager.

Parameters:

visManager the visibility manager

9.39.3.9 `GtpVisibility::VisibilityManager * Ogre::VisibilityDotSceneManager::getVisibilityManager ()`

See set.

9.39.4 Member Data Documentation

9.39.4.1 `Ogre::SceneNodeHierarchyInterface* Ogre::VisibilityDotSceneManager::mHierarchyInterface` [protected]

9.39.4.2 `GtpVisibility::VisibilityManager* Ogre::VisibilityDotSceneManager::mVisibilityManager` [protected]

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

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

9.40 GtpVisibility::VisibilityEnvironment Class Reference

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

Public Types

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

Public Member Functions

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

9.40.1 Detailed Description

This class provides different parameters for the visibility manager.

9.40.2 Member Enumeration Documentation

9.40.2.1 enum [GtpVisibility::VisibilityEnvironment::CullingManagerType](#)

Different types of occlusion culling algorithms

Enumeration values:

FRUSTUM_CULLING

STOP_AND_WAIT

COHERENT_HIERARCHICAL_CULLING

9.40.3 Constructor & Destructor Documentation

9.40.3.1 [GtpVisibility::VisibilityEnvironment::VisibilityEnvironment](#) ()

9.40.4 Member Function Documentation

9.40.4.1 void [GtpVisibility::VisibilityEnvironment::LoadEnvironment](#) ()

Loads an environment from disk.

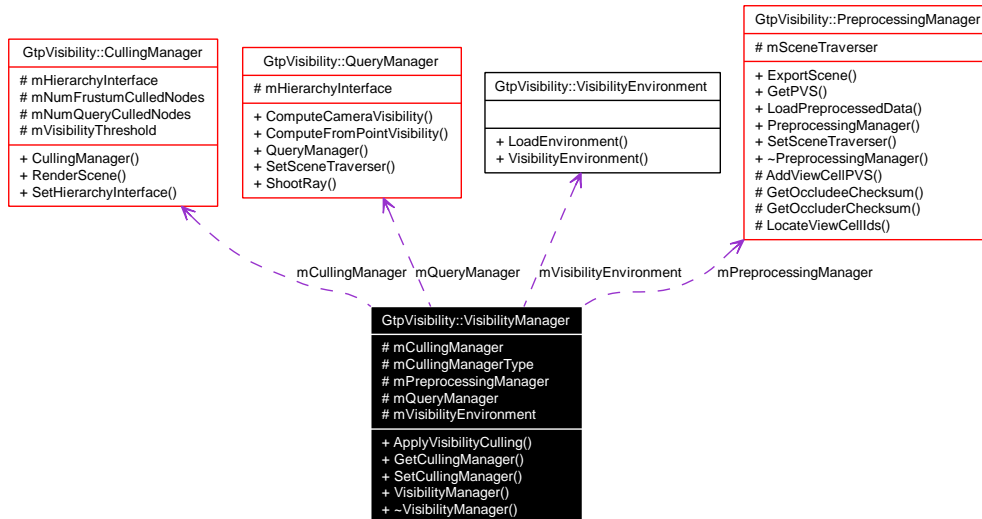
The documentation for this class was generated from the following files:

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

9.41 GtpVisibility::VisibilityManager Class Reference

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

Collaboration diagram for GtpVisibility::VisibilityManager:



Public Member Functions

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

Protected Attributes

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

9.41.1 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.

9.41.2 Constructor & Destructor Documentation

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

Constructor taking the visibility environment object as parameter

Parameters:

visEnvironment the visibility environment

9.41.2.2 GtpVisibility::VisibilityManager::~~VisibilityManager ()

9.41.3 Member Function Documentation

9.41.3.1 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

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

Returns the current online occlusion culling manager. See set

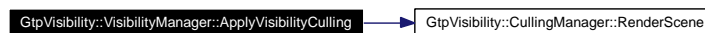
9.41.3.3 void GtpVisibility::VisibilityManager::ApplyVisibilityCulling ()

Applies the online visibility culling algorithm on a scene.

Remarks:

the algorithm depends on the current culling manager.

Here is the call graph for this function:



9.41.4 Member Data Documentation

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

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

9.41.4.3 PreprocessingManager* GtpVisibility::VisibilityManager::mPreprocessingManager
[protected]9.41.4.4 VisibilityEnvironment* GtpVisibility::VisibilityManager::mVisibilityEnvironment
[protected]9.41.4.5 VisibilityEnvironment::CullingManagerType GtpVisibility::VisibilityManager::m-
CullingManagerType [protected]

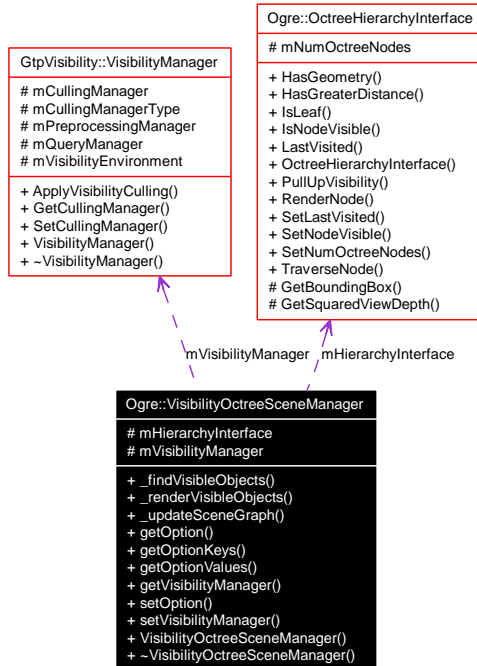
The documentation for this class was generated from the following files:

- GtpVisibility/include/VisibilityManager.h
- GtpVisibility/src/VisibilityManager.cpp

9.42 Ogre::VisibilityOctreeSceneManager Class Reference

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

Collaboration diagram for Ogre::VisibilityOctreeSceneManager:



Public Member Functions

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

Protected Attributes

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

9.42.1 Detailed Description

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

9.42.2 Constructor & Destructor Documentation

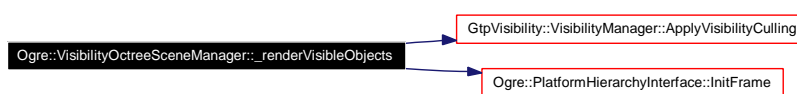
9.42.2.1 **Ogre::VisibilityOctreeSceneManager::VisibilityOctreeSceneManager** (**GtpVisibility::VisibilityManager** * *visManager*)

9.42.2.2 **Ogre::VisibilityOctreeSceneManager::~~VisibilityOctreeSceneManager** ()

9.42.3 Member Function Documentation

9.42.3.1 **void Ogre::VisibilityOctreeSceneManager::_renderVisibleObjects** ()

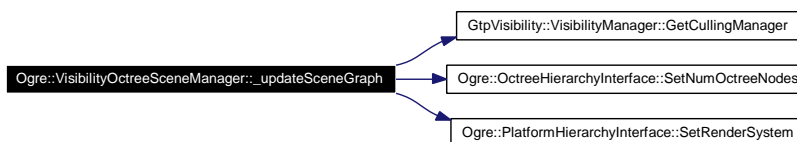
Here is the call graph for this function:



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

9.42.3.3 **void Ogre::VisibilityOctreeSceneManager::_updateSceneGraph** (**Camera** * *cam*)

Here is the call graph for this function:



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

Sets the given option for the SceneManager

Remarks:

Options are: "Algorithm", int *;

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

Gets the given option for the Scene VisibilityManager.

Remarks:

See setOption

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

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

9.42.3.8 **void** `Ogre::VisibilityOctreeSceneManager::setVisibilityManager` (`GtpVisibility::VisibilityManager * visManager`)

Sets the visibility manager.

Parameters:

visManager the visibility manager

9.42.3.9 `GtpVisibility::VisibilityManager * Ogre::VisibilityOctreeSceneManager::getVisibilityManager` (`void`)

See set.

9.42.4 Member Data Documentation

9.42.4.1 `OctreeHierarchyInterface* Ogre::VisibilityOctreeSceneManager::mHierarchyInterface` [protected]

9.42.4.2 `GtpVisibility::VisibilityManager* Ogre::VisibilityOctreeSceneManager::mVisibilityManager` [protected]

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

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

9.43 Ogre::VisibilitySceneManager Class Reference

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

Collaboration diagram for Ogre::VisibilitySceneManager:



Public Member Functions

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

Protected Attributes

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

9.43.1 Detailed Description

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

9.43.2 Constructor & Destructor Documentation

9.43.2.1 **Ogre::VisibilitySceneManager::VisibilitySceneManager** (**GtpVisibility::VisibilityManager** * *visManager*)

9.43.2.2 **Ogre::VisibilitySceneManager::~~VisibilitySceneManager** ()

9.43.3 Member Function Documentation

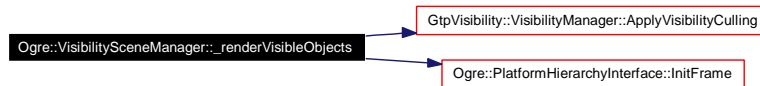
9.43.3.1 **void Ogre::VisibilitySceneManager::_renderVisibleObjects** ()

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.

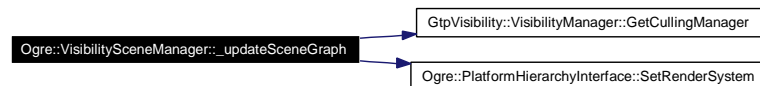
Here is the call graph for this function:



9.43.3.2 **void Ogre::VisibilitySceneManager::_findVisibleObjects** (**Camera** * *cam*, **bool** *onlyShadowCasters*)

9.43.3.3 **void Ogre::VisibilitySceneManager::_updateSceneGraph** (**Camera** * *cam*)

Here is the call graph for this function:



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

Sets the given option for the SceneManager

Remarks:

Options are: "Algorithm", int *;

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

Gets the given option for the Scene VisibilityManager.

Remarks:

See setOption

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

9.43.3.7 **bool** `Ogre::VisibilitySceneManager::getOptionKeys` (`StringVector & refKeys`)

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

Sets the visibility manager.

Parameters:

visManager the visibility manager

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

See set.

9.43.4 Member Data Documentation

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

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

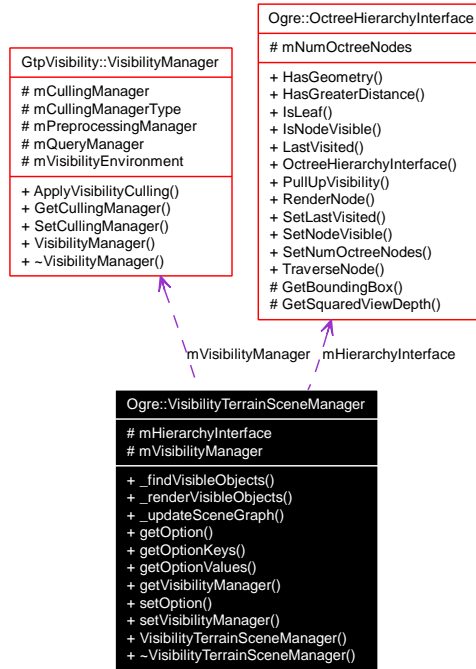
The documentation for this class was generated from the following files:

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

9.44 Ogre::VisibilityTerrainSceneManager Class Reference

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

Collaboration diagram for `Ogre::VisibilityTerrainSceneManager`:



Public Member Functions

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

Protected Attributes

- `OctreeHierarchyInterface * mHierarchyInterface`
- `GtpVisibility::VisibilityManager * mVisibilityManager`

9.44.1 Detailed Description

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

9.44.2 Constructor & Destructor Documentation

9.44.2.1 Ogre::VisibilityTerrainSceneManager::VisibilityTerrainSceneManager (GtpVisibility::VisibilityManager * visManager)

Here is the call graph for this function:

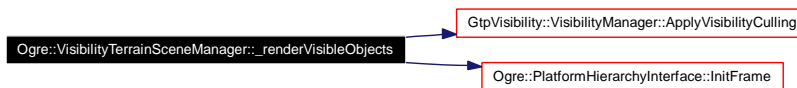


9.44.2.2 Ogre::VisibilityTerrainSceneManager::~~VisibilityTerrainSceneManager ()

9.44.3 Member Function Documentation

9.44.3.1 void Ogre::VisibilityTerrainSceneManager::_renderVisibleObjects (void)

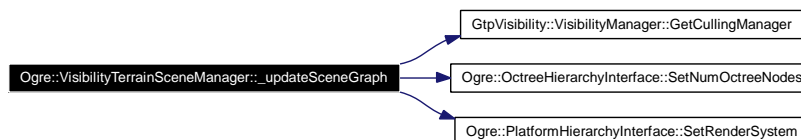
Here is the call graph for this function:



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

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

Here is the call graph for this function:



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

Sets the given option for the SceneManager

Remarks:

Options are: "Algorithm", int *;

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

Gets the given option for the Scene VisibilityManager.

Remarks:

See `setOption`

9.44.3.6 `bool Ogre::VisibilityTerrainSceneManager::getOptionValues (const String & key, StringVector & refValueList)`

9.44.3.7 `bool Ogre::VisibilityTerrainSceneManager::getOptionKeys (StringVector & refKeys)`

9.44.3.8 `void Ogre::VisibilityTerrainSceneManager::setVisibilityManager (GtpVisibility::VisibilityManager * visManager)`

Sets the visibility manager.

Parameters:

visManager the visibility manager

9.44.3.9 `GtpVisibility::VisibilityManager * Ogre::VisibilityTerrainSceneManager::getVisibilityManager ()`

See `set`.

9.44.4 Member Data Documentation

9.44.4.1 `OctreeHierarchyInterface* Ogre::VisibilityTerrainSceneManager::mHierarchyInterface` [protected]

9.44.4.2 `GtpVisibility::VisibilityManager* Ogre::VisibilityTerrainSceneManager::mVisibilityManager` [protected]

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

- `Ogre/include/OgreVisibilityTerrainSceneManager.h`
- `Ogre/src/OgreVisibilityTerrainSceneManager.cpp`

Chapter 10

GameTools Visibility Modules File Documentation

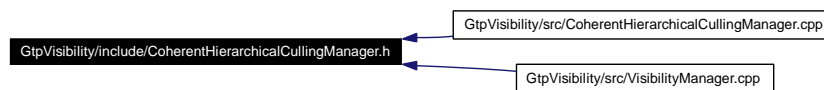
10.1 GtpVisibility/include/CoherentHierarchicalCullingManager.h File Reference

```
#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](#)

Typedefs

- typedef std::pair< [HierarchyNode](#) *, OcclusionQuery * > [QueryPair](#)
- typedef std::queue< [QueryPair](#) > [QueryQueue](#)

10.1.1 Typedef Documentation

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

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

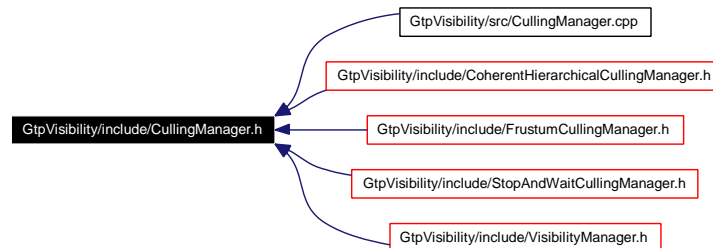
10.2 GtpVisibility/include/CullingManager.h File Reference

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

Include dependency graph for CullingManager.h:



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



Namespaces

- namespace [GtpVisibility](#)

10.3 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](#)

Typedefs

- typedef `std::priority_queue< HierarchyNode *, std::vector< HierarchyNode * >, GreaterDistance< std::vector< HierarchyNode * >::value_type > >` [DistanceQueue](#)
- typedef void [HierarchyNode](#)

10.3.1 Typedef Documentation

10.3.1.1 typedef void [GtpVisibility::HierarchyNode](#)

10.3.1.2 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.

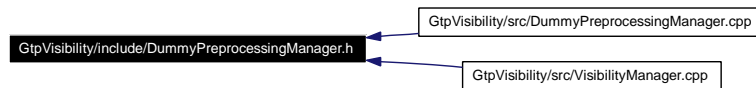
10.4 GtpVisibility/include/DummyPreprocessingManager.h File Reference

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

Include dependency graph for DummyPreprocessingManager.h:



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



Namespaces

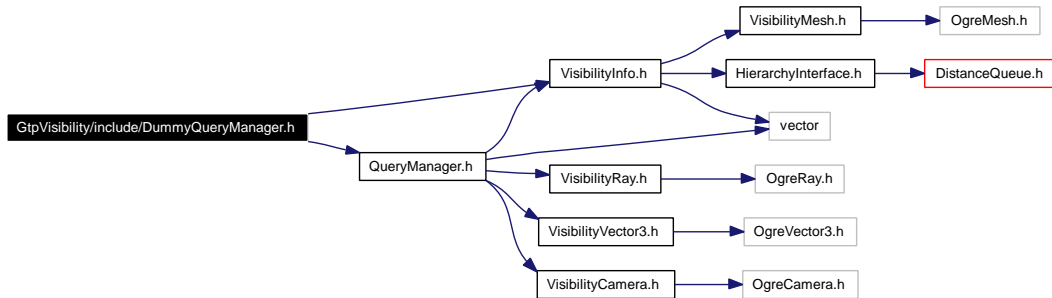
- namespace [GtpVisibility](#)

10.5 GtpVisibility/include/DummyQueryManager.h File Reference

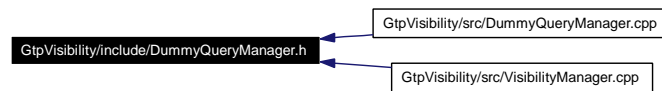
```
#include "VisibilityInfo.h"
```

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

Include dependency graph for DummyQueryManager.h:



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



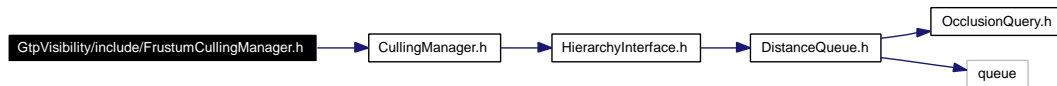
Namespaces

- namespace [GtpVisibility](#)

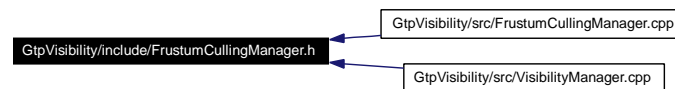
10.6 GtpVisibility/include/FrustumCullingManager.h File Reference

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

Include dependency graph for FrustumCullingManager.h:



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



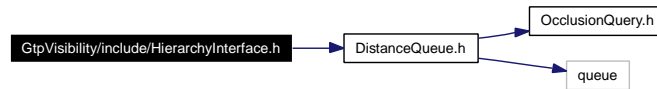
Namespaces

- namespace [GtpVisibility](#)

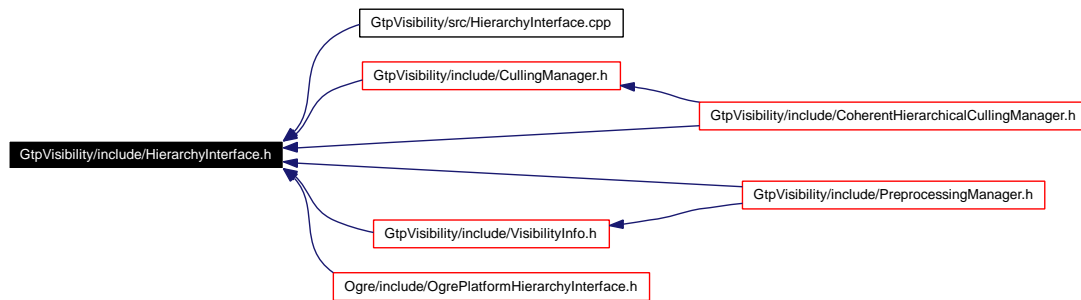
10.7 GtpVisibility/include/HierarchyInterface.h File Reference

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

Include dependency graph for HierarchyInterface.h:



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

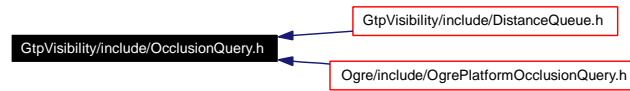


Namespaces

- namespace `GtpVisibility`

10.8 GtpVisibility/include/OcclusionQuery.h File Reference

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



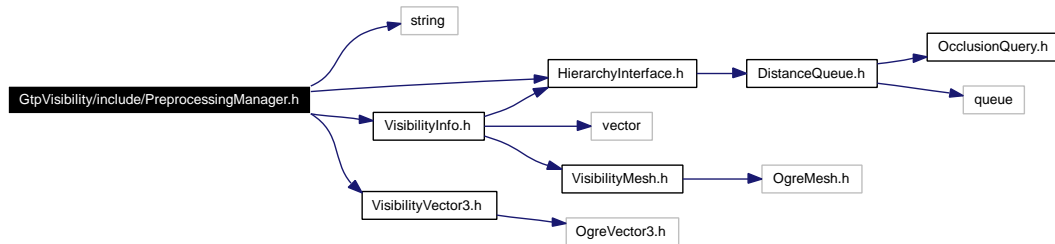
Namespaces

- namespace [GtpVisibility](#)

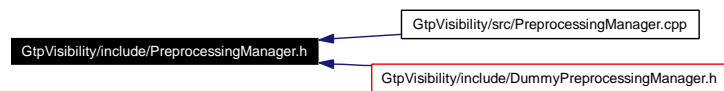
10.9 GtpVisibility/include/PreprocessingManager.h File Reference

```
#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:



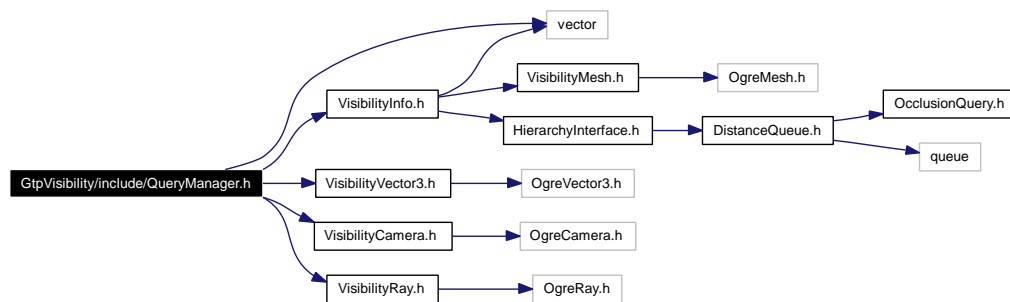
Namespaces

- namespace `GtpVisibility`
- namespace `std`

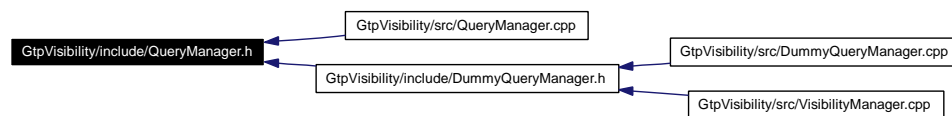
10.10 GtpVisibility/include/QueryManager.h File Reference

```
#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:



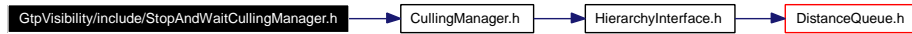
Namespaces

- namespace [GtpVisibility](#)

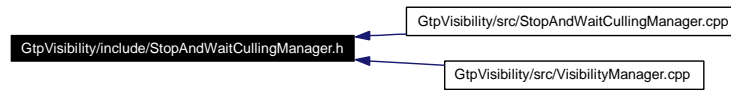
10.11 GtpVisibility/include/StopAndWaitCullingManager.h File Reference

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

Include dependency graph for StopAndWaitCullingManager.h:



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



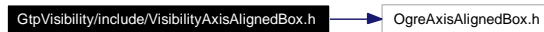
Namespaces

- namespace [GtpVisibility](#)

10.12 GtpVisibility/include/VisibilityAxisAlignedBox.h File Reference

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

Include dependency graph for VisibilityAxisAlignedBox.h:



Namespaces

- namespace [GtpVisibility](#)

Typedefs

- typedef [Ogre::AxisAlignedBox](#) [AxisAlignedBox](#)

10.12.1 Typedef Documentation

10.12.1.1 typedef [Ogre::AxisAlignedBox](#) [GtpVisibility::AxisAlignedBox](#)

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

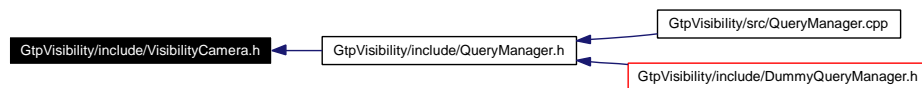
10.13 GtpVisibility/include/VisibilityCamera.h File Reference

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

Include dependency graph for VisibilityCamera.h:



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



Namespaces

- namespace [GtpVisibility](#)

Typedefs

- typedef [Ogre::Camera](#) [GtpVisibility::Camera](#)

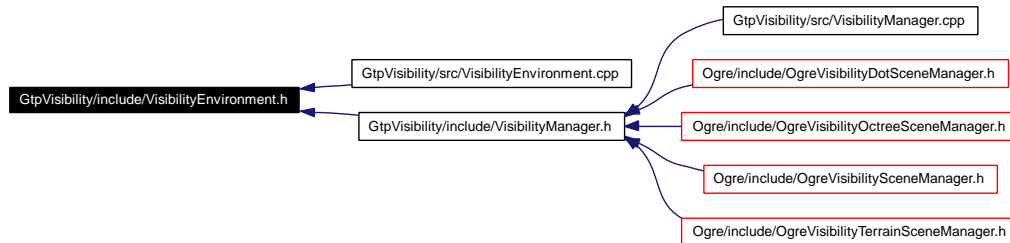
10.13.1 Typedef Documentation

10.13.1.1 typedef [Ogre::Camera](#) [GtpVisibility::Camera](#)

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

10.14 GtpVisibility/include/VisibilityEnvironment.h File Reference

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



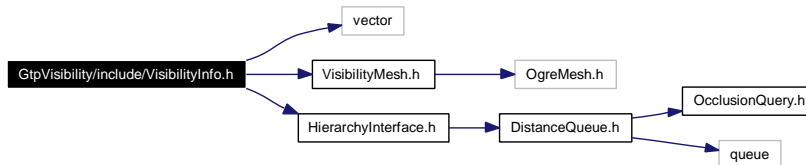
Namespaces

- namespace [GtpVisibility](#)

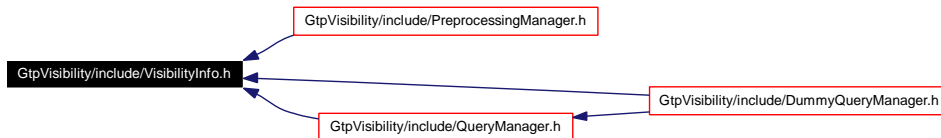
10.15 GtpVisibility/include/VisibilityInfo.h File Reference

```
#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`

10.15.1 Define Documentation

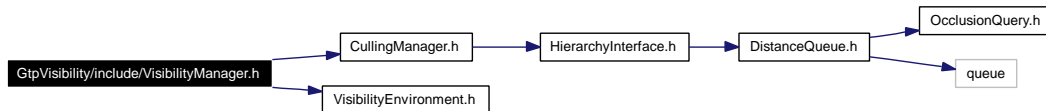
10.15.1.1 #define InfoContainer std::vector

10.16 GtpVisibility/include/VisibilityManager.h File Reference

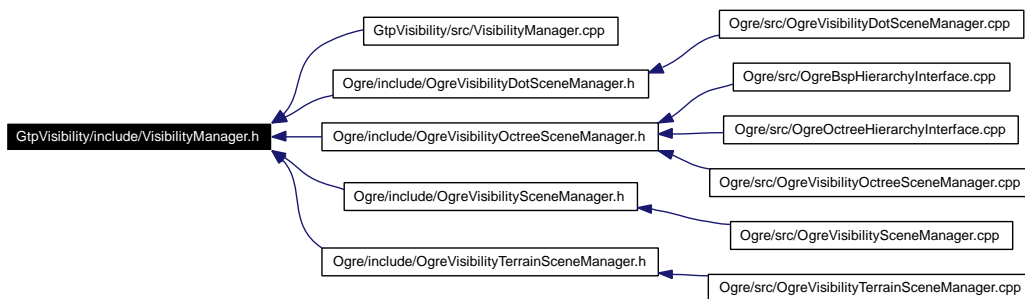
```
#include "CullingManager.h"
```

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

Include dependency graph for VisibilityManager.h:



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



Namespaces

- namespace [GtpVisibility](#)

10.17 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](#)

Typedefs

- typedef [Ogre::Mesh](#) [GtpVisibility::Mesh](#)

10.17.1 Typedef Documentation

10.17.1.1 typedef [Ogre::Mesh](#) [GtpVisibility::Mesh](#)

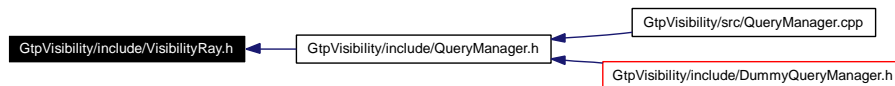
10.18 GtpVisibility/include/VisibilityRay.h File Reference

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

Include dependency graph for VisibilityRay.h:



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



Namespaces

- namespace [GtpVisibility](#)

Typedefs

- typedef [Ogre::Ray](#) [Ray](#)

10.18.1 Typedef Documentation

10.18.1.1 typedef [Ogre::Ray](#) [GtpVisibility::Ray](#)

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

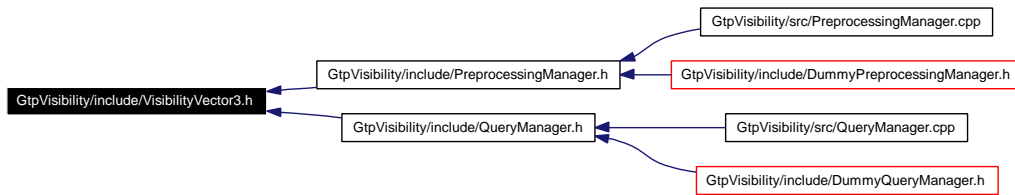
10.19 GtpVisibility/include/VisibilityVector3.h File Reference

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

Include dependency graph for VisibilityVector3.h:



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



Namespaces

- namespace [GtpVisibility](#)

Typedefs

- typedef [Ogre::Vector3](#) [Vector3](#)

10.19.1 Typedef Documentation

10.19.1.1 typedef [Ogre::Vector3](#) [GtpVisibility::Vector3](#)

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

10.20 GtpVisibility/src/CoherentHierarchicalCullingManager.cpp File Reference

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

Include dependency graph for CoherentHierarchicalCullingManager.cpp:



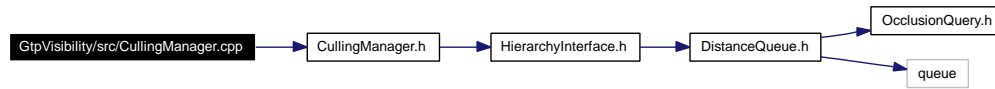
Namespaces

- namespace [GtpVisibility](#)

10.21 GtpVisibility/src/CullingManager.cpp File Reference

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

Include dependency graph for CullingManager.cpp:



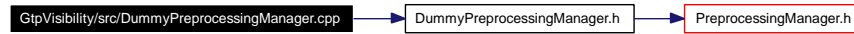
Namespaces

- namespace [GtpVisibility](#)

10.22 GtpVisibility/src/DummyPreprocessingManager.cpp File Reference

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

Include dependency graph for DummyPreprocessingManager.cpp:



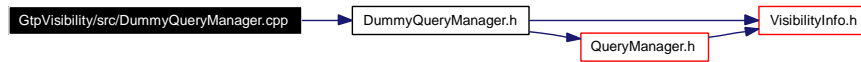
Namespaces

- namespace [GtpVisibility](#)

10.23 GtpVisibility/src/DummyQueryManager.cpp File Reference

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

Include dependency graph for DummyQueryManager.cpp:



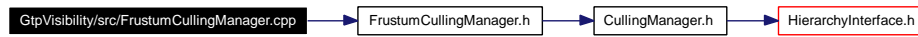
Namespaces

- namespace [GtpVisibility](#)

10.24 GtpVisibility/src/FrustumCullingManager.cpp File Reference

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

Include dependency graph for FrustumCullingManager.cpp:



Namespaces

- namespace [GtpVisibility](#)

10.25 GtpVisibility/src/HierarchyInterface.cpp File Reference

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

Include dependency graph for HierarchyInterface.cpp:



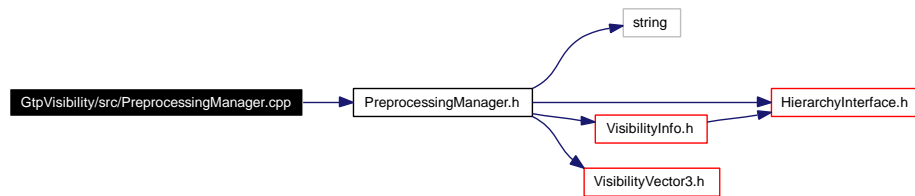
Namespaces

- namespace [GtpVisibility](#)

10.26 GtpVisibility/src/PreprocessingManager.cpp File Reference

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

Include dependency graph for PreprocessingManager.cpp:



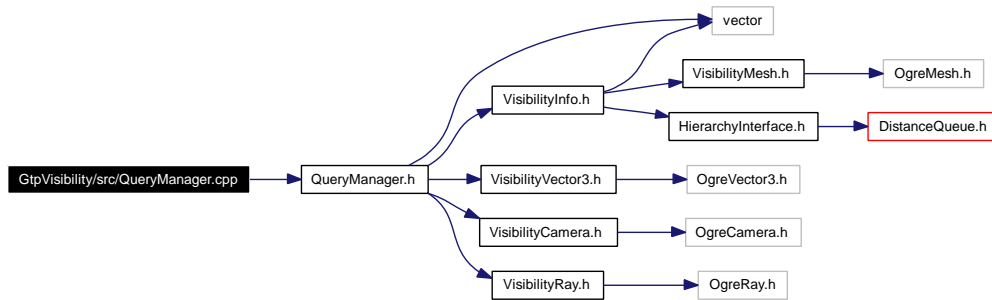
Namespaces

- namespace [GtpVisibility](#)

10.27 GtpVisibility/src/QueryManager.cpp File Reference

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

Include dependency graph for QueryManager.cpp:



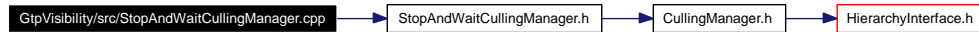
Namespaces

- namespace [GtpVisibility](#)

10.28 GtpVisibility/src/StopAndWaitCullingManager.cpp File Reference

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

Include dependency graph for StopAndWaitCullingManager.cpp:



Namespaces

- namespace [GtpVisibility](#)

10.29 GtpVisibility/src/VisibilityEnvironment.cpp File Reference

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

Include dependency graph for VisibilityEnvironment.cpp:



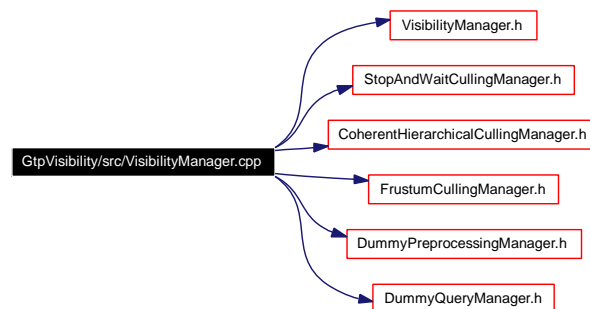
Namespaces

- namespace [GtpVisibility](#)

10.30 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](#)

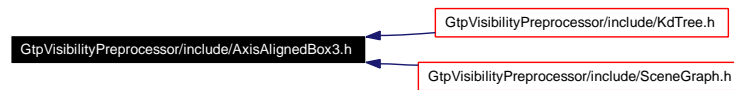
10.31 GtpVisibilityPreprocessor/include/AxisAlignedBox3.h File Reference

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

Include dependency graph for AxisAlignedBox3.h:



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



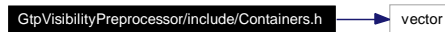
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

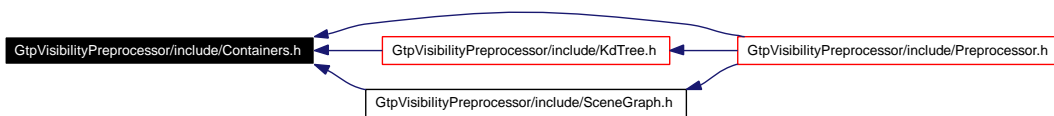
10.32 GtpVisibilityPreprocessor/include/Containers.h File Reference

```
#include <vector>
```

Include dependency graph for Containers.h:



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



Namespaces

- namespace [GtpVisibilityPreprocessor](#)

Typedefs

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

10.32.1 Typedef Documentation

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

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

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

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

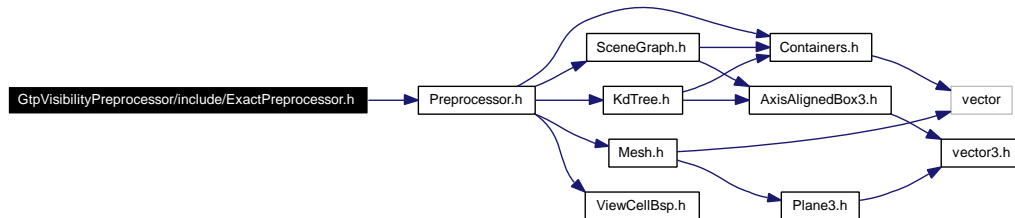
10.32.1.3 typedef vector<[HierarchyNode *](#)> [GtpVisibilityPreprocessor::NodeContainer](#)

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

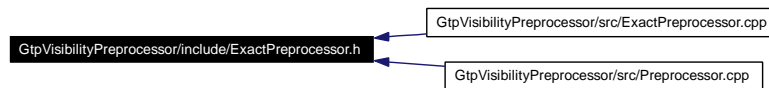
10.33 GtpVisibilityPreprocessor/include/ExactPreprocessor.h File Reference

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

Include dependency graph for ExactPreprocessor.h:



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



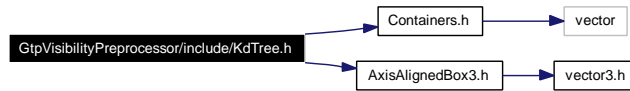
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

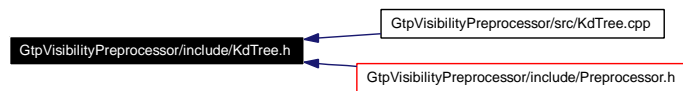
10.34 GtpVisibilityPreprocessor/include/KdTree.h File Reference

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

Include dependency graph for KdTree.h:



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



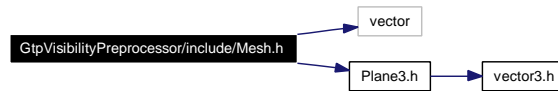
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

10.35 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](#)

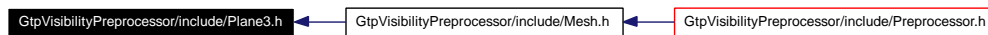
10.36 GtpVisibilityPreprocessor/include/Plane3.h File Reference

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

Include dependency graph for Plane3.h:



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



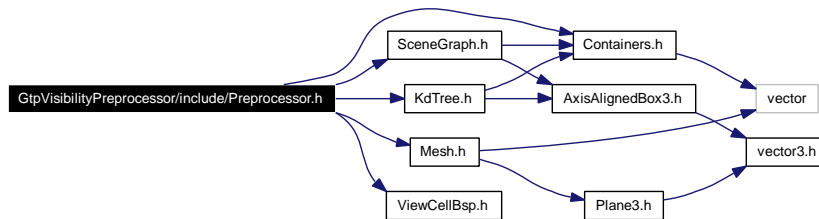
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

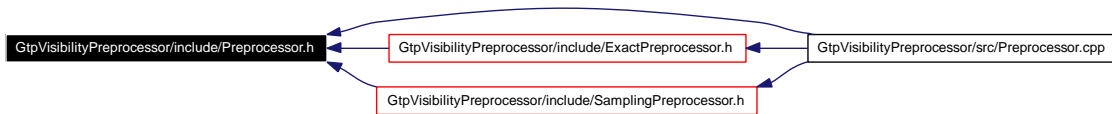
10.37 GtpVisibilityPreprocessor/include/Preprocessor.h File Reference

```
#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:



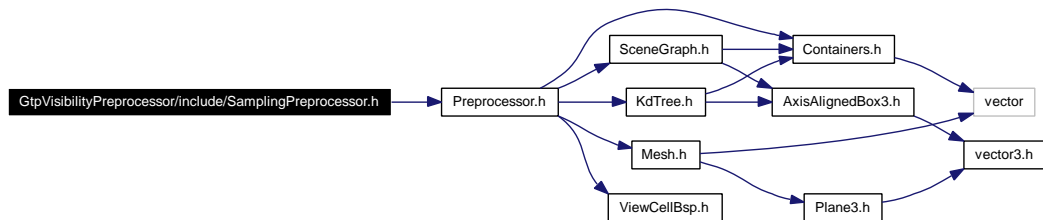
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

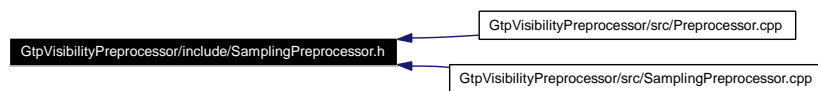
10.38 GtpVisibilityPreprocessor/include/SamplingPreprocessor.h File Reference

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

Include dependency graph for SamplingPreprocessor.h:



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



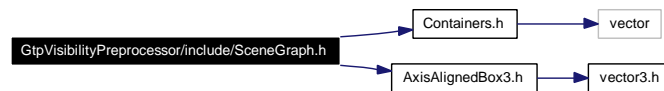
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

10.39 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:

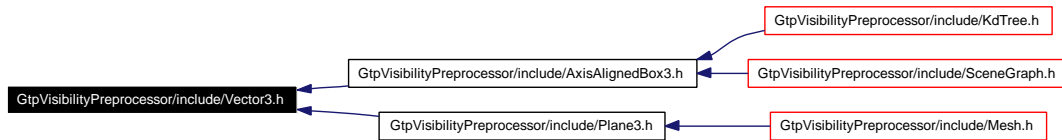


Namespaces

- namespace [GtpVisibilityPreprocessor](#)

10.40 GtpVisibilityPreprocessor/include/Vector3.h File Reference

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



Namespaces

- namespace [GtpVisibilityPreprocessor](#)

10.41 GtpVisibilityPreprocessor/include/ViewCellBSP.h File Reference

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



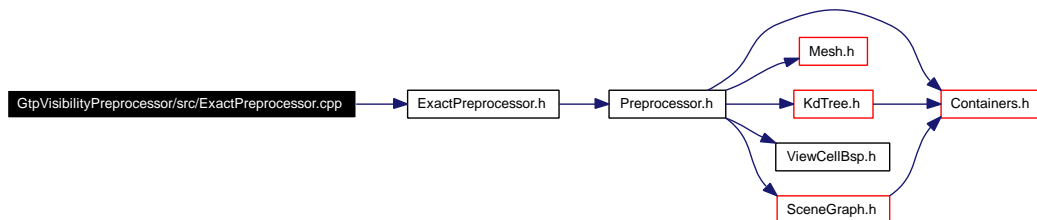
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

10.42 GtpVisibilityPreprocessor/src/ExactPreprocessor.cpp File Reference

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

Include dependency graph for ExactPreprocessor.cpp:



Namespaces

- namespace [GtpVisibilityPreprocessor](#)

10.43 GtpVisibilityPreprocessor/src/KdTree.cpp File Reference

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

Include dependency graph for KdTree.cpp:



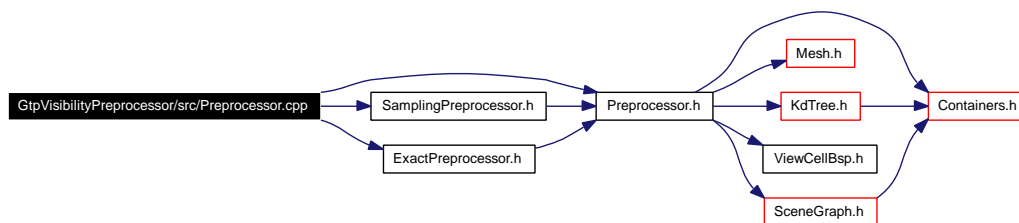
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

10.44 GtpVisibilityPreprocessor/src/Preprocessor.cpp File Reference

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

Include dependency graph for Preprocessor.cpp:



Namespaces

- namespace [GtpVisibilityPreprocessor](#)

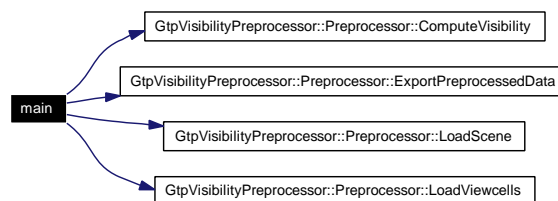
Functions

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

10.44.1 Function Documentation

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

Here is the call graph for this function:



10.45 GtpVisibilityPreprocessor/src/SamplingPreprocessor.cpp File Reference

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

Include dependency graph for SamplingPreprocessor.cpp:



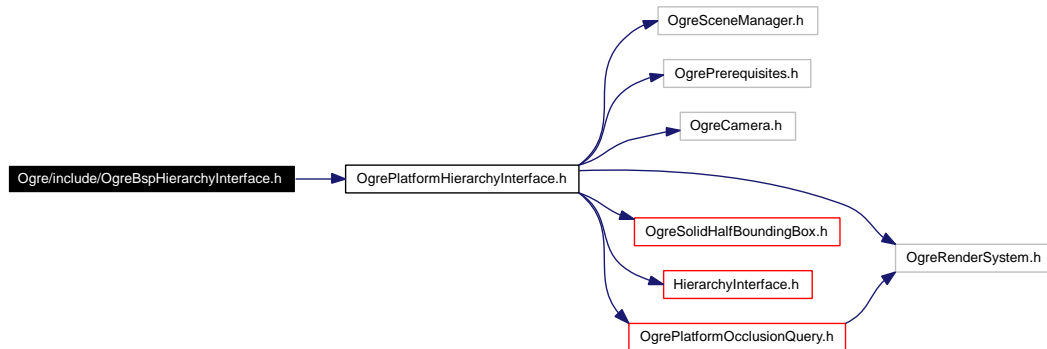
Namespaces

- namespace [GtpVisibilityPreprocessor](#)

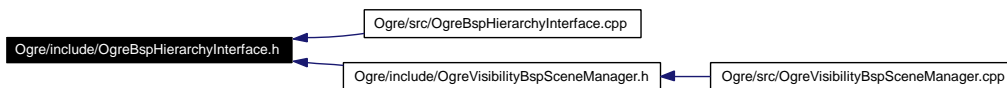
10.46 Ogre/include/OgreBspHierarchyInterface.h File Reference

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

Include dependency graph for OgreBspHierarchyInterface.h:



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



Namespaces

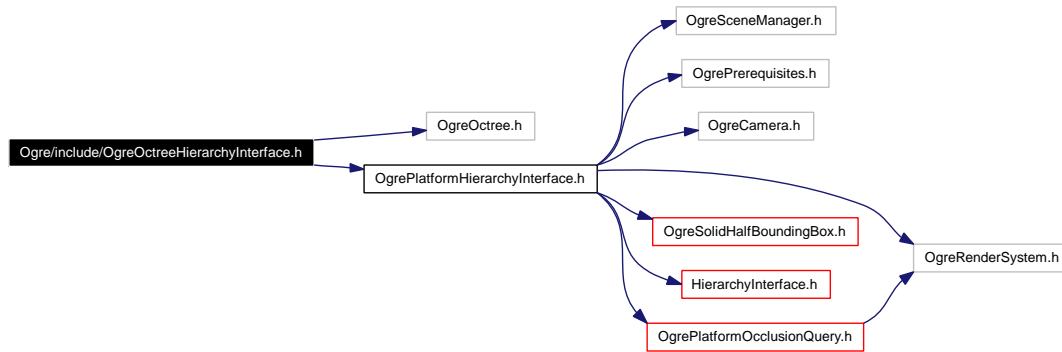
- namespace [Ogre](#)

10.47 Ogre/include/OgreOctreeHierarchyInterface.h File Reference

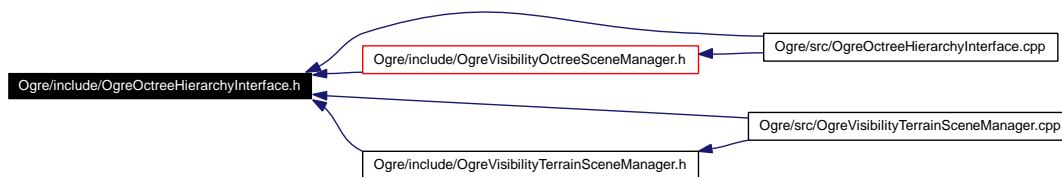
```
#include <OgreOctree.h>
```

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

Include dependency graph for OgreOctreeHierarchyInterface.h:



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



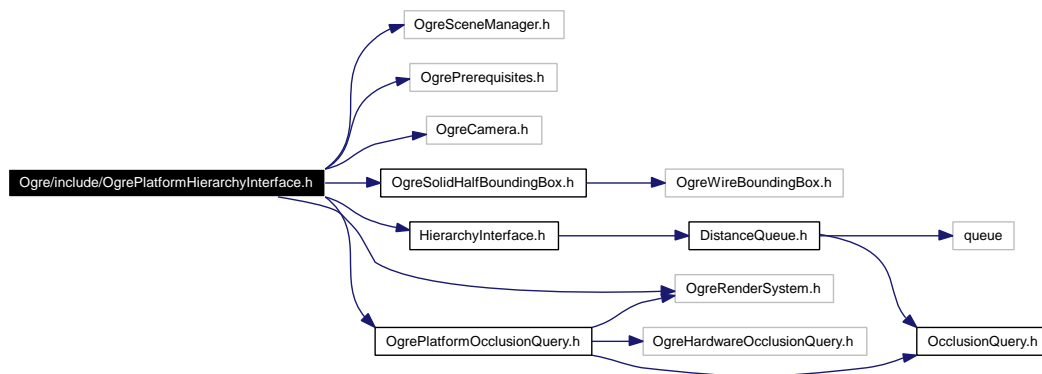
Namespaces

- namespace [Ogre](#)

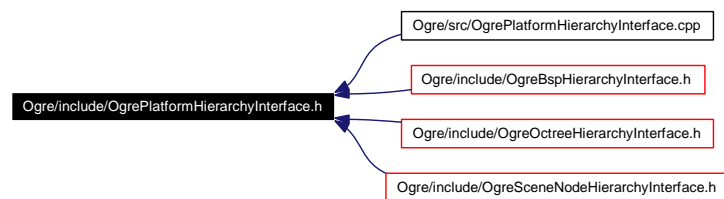
10.48 Ogre/include/OgrePlatformHierarchyInterface.h File Reference

```
#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:



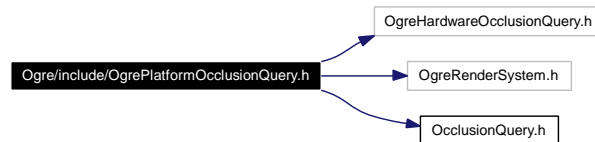
Namespaces

- namespace [Ogre](#)

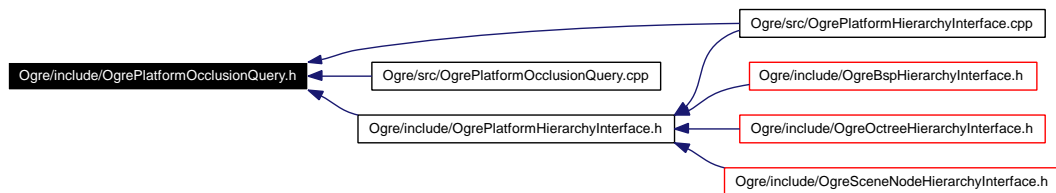
10.49 Ogre/include/OgrePlatformOcclusionQuery.h File Reference

```
#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](#)

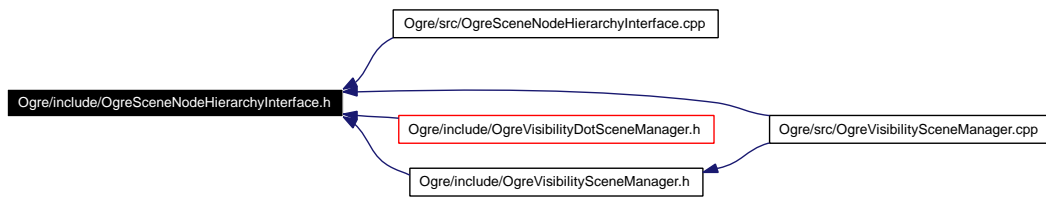
10.50 Ogre/include/OgreSceneNodeHierarchyInterface.h File Reference

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

Include dependency graph for OgreSceneNodeHierarchyInterface.h:



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



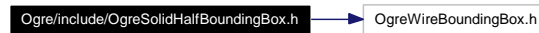
Namespaces

- namespace [Ogre](#)

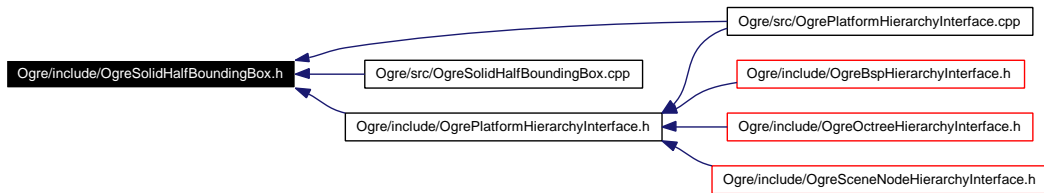
10.51 Ogre/include/OgreSolidHalfBoundingBox.h File Reference

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

Include dependency graph for OgreSolidHalfBoundingBox.h:



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



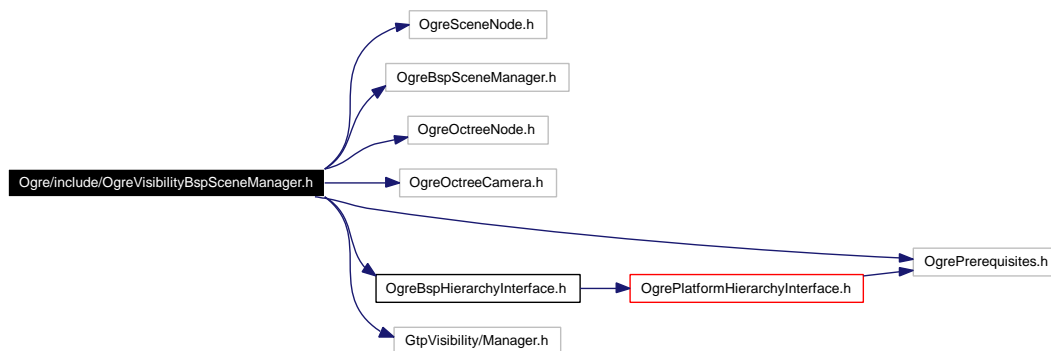
Namespaces

- namespace [Ogre](#)

10.52 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:



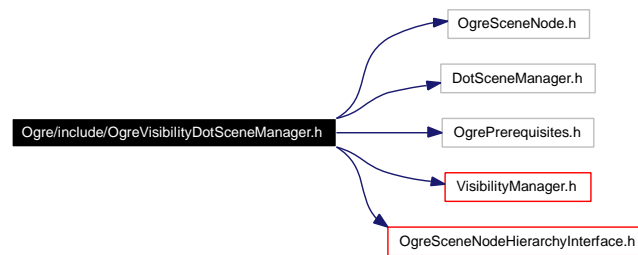
Namespaces

- namespace `Ogre`

10.53 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:



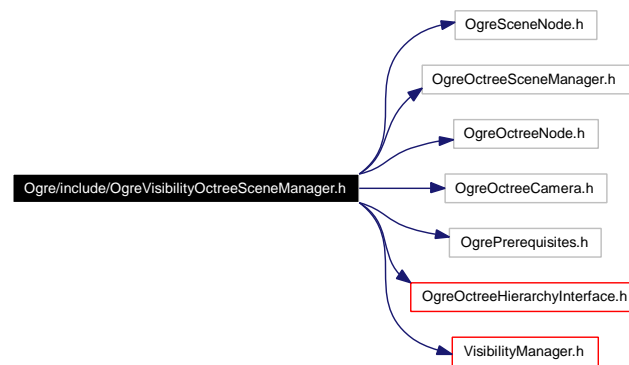
Namespaces

- namespace [Ogre](#)

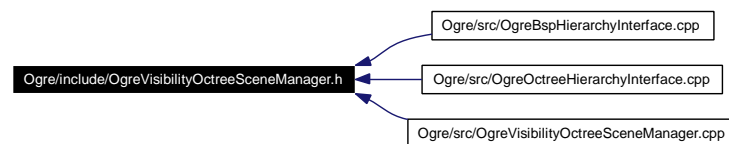
10.54 Ogre/include/OgreVisibilityOctreeSceneManager.h File Reference

```
#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:



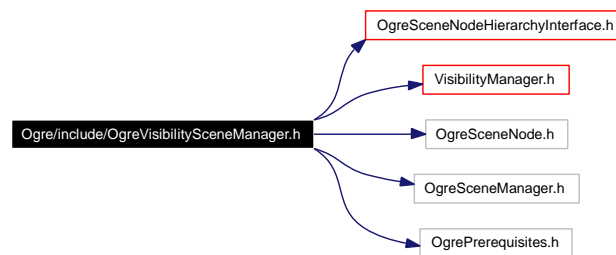
Namespaces

- namespace [Ogre](#)

10.55 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:



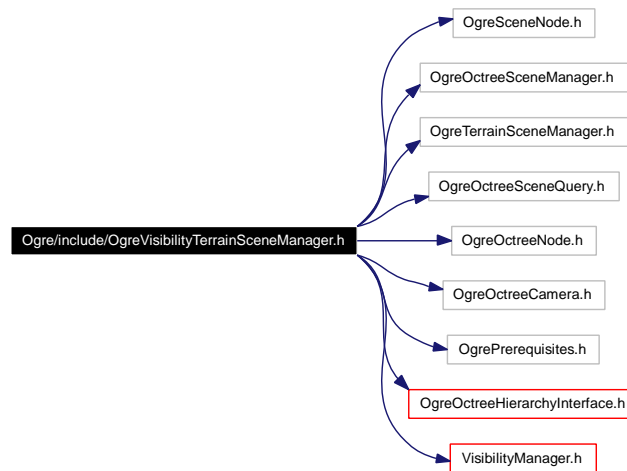
Namespaces

- namespace [Ogre](#)

10.56 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:



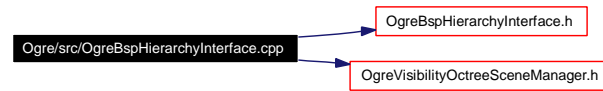
Namespaces

- namespace [Ogre](#)

10.57 Ogre/src/OgreBspHierarchyInterface.cpp File Reference

```
#include "OgreBspHierarchyInterface.h"  
#include "OgreVisibilityOctreeSceneManager.h"
```

Include dependency graph for OgreBspHierarchyInterface.cpp:



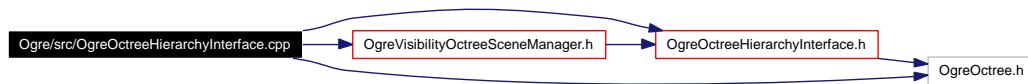
Namespaces

- namespace [Ogre](#)

10.58 Ogre/src/OgreOctreeHierarchyInterface.cpp File Reference

```
#include "OgreOctreeHierarchyInterface.h"  
#include "OgreVisibilityOctreeSceneManager.h"  
#include <OgreOctree.h>
```

Include dependency graph for OgreOctreeHierarchyInterface.cpp:



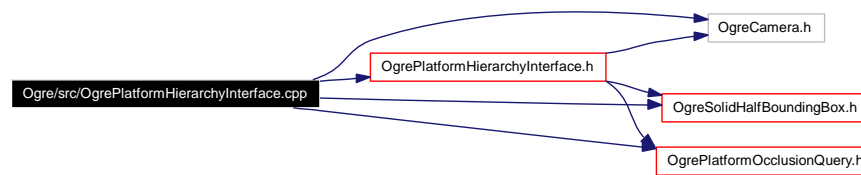
Namespaces

- namespace [Ogre](#)

10.59 Ogre/src/OgrePlatformHierarchyInterface.cpp File Reference

```
#include <OgreCamera.h>
#include "OgreSolidHalfBoundingBox.h"
#include "OgrePlatformHierarchyInterface.h"
#include "OgrePlatformOcclusionQuery.h"
```

Include dependency graph for OgrePlatformHierarchyInterface.cpp:



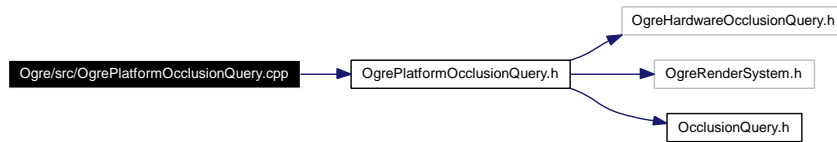
Namespaces

- namespace [Ogre](#)

10.60 Ogre/src/OgrePlatformOcclusionQuery.cpp File Reference

```
#include "OgrePlatformOcclusionQuery.h"
```

Include dependency graph for OgrePlatformOcclusionQuery.cpp:



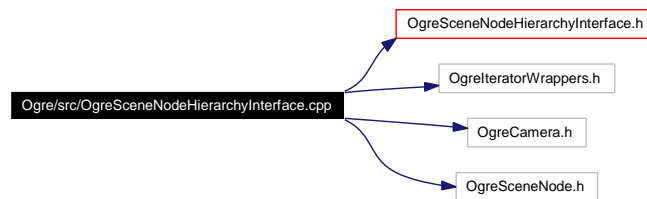
Namespaces

- namespace [Ogre](#)

10.61 Ogre/src/OgreSceneNodeHierarchyInterface.cpp File Reference

```
#include "OgreSceneNodeHierarchyInterface.h"  
#include <OgreIteratorWrappers.h>  
#include <OgreCamera.h>  
#include <OgreSceneNode.h>
```

Include dependency graph for OgreSceneNodeHierarchyInterface.cpp:



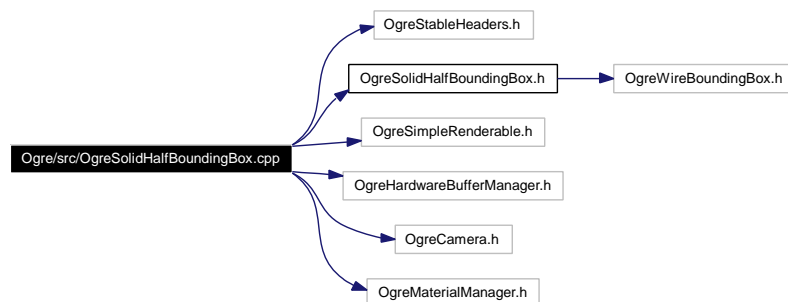
Namespaces

- namespace [Ogre](#)

10.62 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

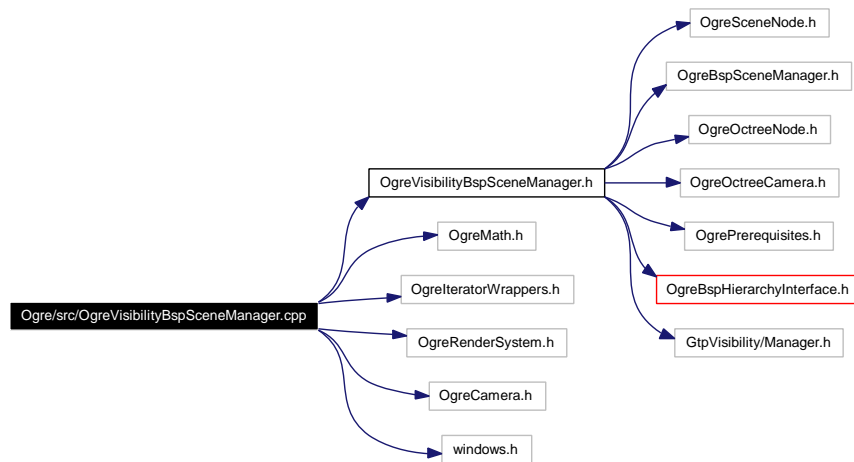
10.62.1 Define Documentation

10.62.1.1 `#define` [POSITION_BINDING](#) 0

10.63 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`:



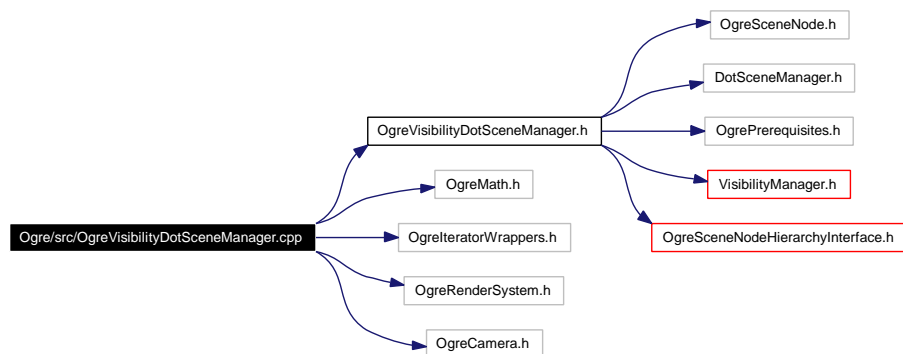
Namespaces

- namespace [Ogre](#)

10.64 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:



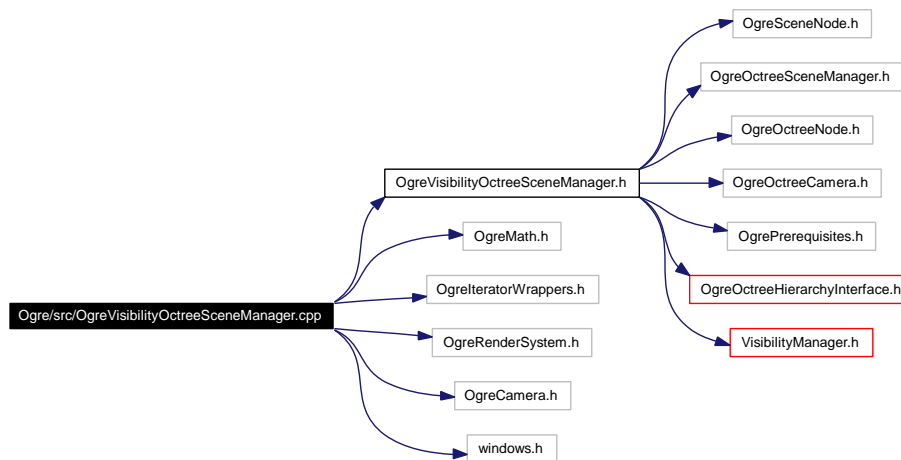
Namespaces

- namespace [Ogre](#)

10.65 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:



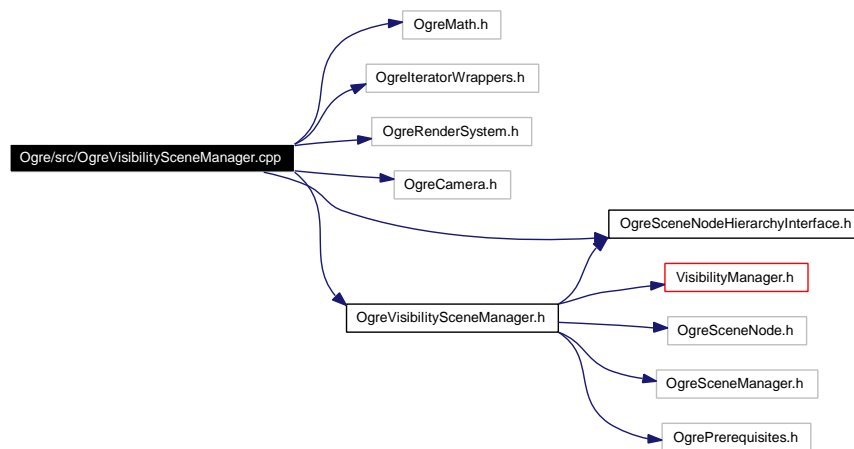
Namespaces

- namespace [Ogre](#)

10.66 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:



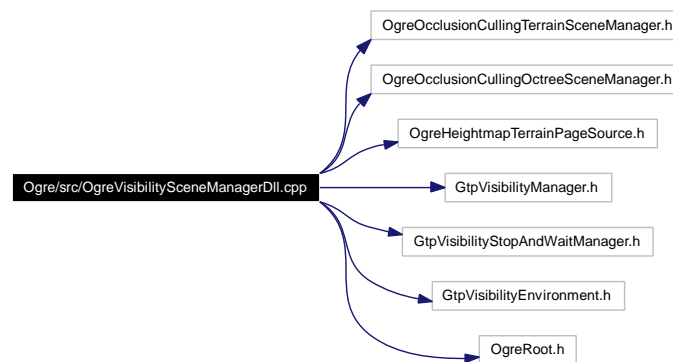
Namespaces

- namespace [Ogre](#)

10.67 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](#)

Functions

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

Variables

- OcclusionCullingOctreeSceneManager * [cullingOctreePlugin](#)
- OcclusionCullingTerrainSceneManager * [cullingTerrainPlugin](#)
- HeightmapTerrainPageSource * [heightmapTerrainPageSource](#)
- GtpVisibility::Environment * [visEnv](#)
- GtpVisibility::Manager * [visManager](#)

10.67.1 Function Documentation

10.67.1.1 void `Ogre::dllStartPlugin` (void)

10.67.1.2 void `Ogre::dllStopPlugin` (void)

10.67.2 Variable Documentation

10.67.2.1 `GtpVisibility::Environment*` [visEnv](#)

10.67.2.2 `GtpVisibility::Manager*` [visManager](#)

10.67.2.3 `OcclusionCullingOctreeSceneManager*` [Ogre::cullingOctreePlugin](#)

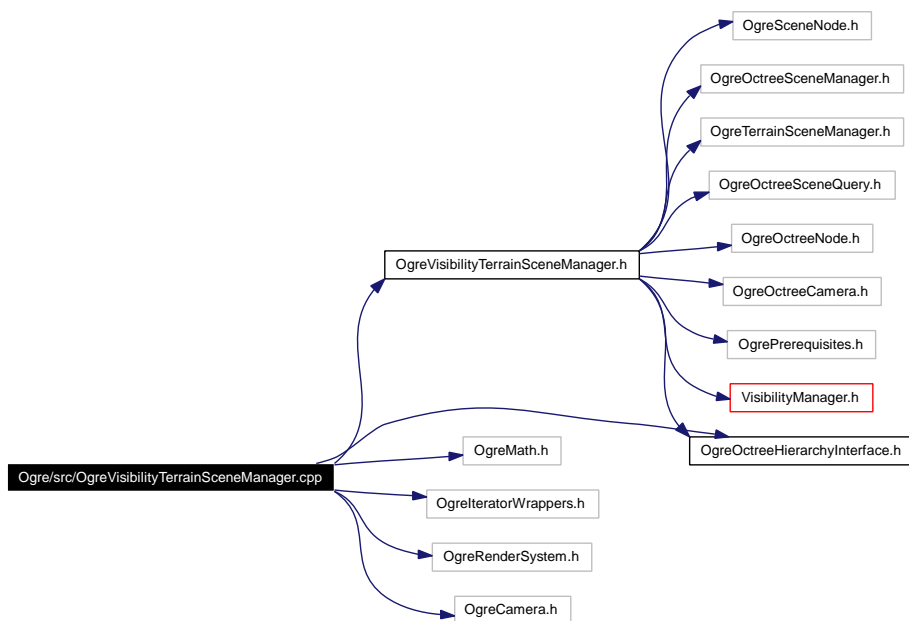
10.67.2.4 `OcclusionCullingTerrainSceneManager*` [Ogre::cullingTerrainPlugin](#)

10.67.2.5 `HeightmapTerrainPageSource*` [Ogre::heightmapTerrainPageSource](#)

10.68 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, 63
- ~OcclusionQuery
 - GtpVisibility::OcclusionQuery, 80
- ~PlatformHierarchyInterface
 - Ogre::PlatformHierarchyInterface, 91
- ~PlatformOcclusionQuery
 - Ogre::PlatformOcclusionQuery, 96
- ~PreprocessingManager
 - GtpVisibility::PreprocessingManager, 100
- ~VisibilityBspSceneManager
 - Ogre::VisibilityBspSceneManager, 126
- ~VisibilityDotSceneManager
 - Ogre::VisibilityDotSceneManager, 129
- ~VisibilityManager
 - GtpVisibility::VisibilityManager, 133
- ~VisibilityOctreeSceneManager
 - Ogre::VisibilityOctreeSceneManager, 135
- ~VisibilitySceneManager
 - Ogre::VisibilitySceneManager, 138
- ~VisibilityTerrainSceneManager
 - Ogre::VisibilityTerrainSceneManager, 141
- _findVisibleObjects
 - Ogre::VisibilityBspSceneManager, 126
 - Ogre::VisibilityDotSceneManager, 129
 - Ogre::VisibilityOctreeSceneManager, 135
 - Ogre::VisibilitySceneManager, 138
 - Ogre::VisibilityTerrainSceneManager, 141
- _renderVisibleObjects
 - Ogre::VisibilityBspSceneManager, 126
 - Ogre::VisibilityDotSceneManager, 129
 - Ogre::VisibilityOctreeSceneManager, 135
 - Ogre::VisibilitySceneManager, 138
 - Ogre::VisibilityTerrainSceneManager, 141
- _updateSceneGraph
 - Ogre::VisibilityBspSceneManager, 126
 - Ogre::VisibilityDotSceneManager, 129
 - Ogre::VisibilityOctreeSceneManager, 135
 - Ogre::VisibilitySceneManager, 138
 - Ogre::VisibilityTerrainSceneManager, 141
- AddViewCellPVS
 - GtpVisibility::DummyPreprocessingManager, 50
 - GtpVisibility::PreprocessingManager, 102
- ApplyVisibilityCulling
 - GtpVisibility::VisibilityManager, 133
- axis
 - GtpVisibilityPreprocessor::KdInterior, 69
- AxisAlignedBox
 - GtpVisibility, 24
 - VisibilityAxisAlignedBox.h, 155
- BeginQuery
 - GtpVisibility::OcclusionQuery, 81
 - Ogre::PlatformOcclusionQuery, 96
- BspHierarchyInterface
 - Ogre::BspHierarchyInterface, 32
- BSPLeaf
 - GtpVisibilityPreprocessor::BSPLeaf, 37
- BSPTree
 - GtpVisibilityPreprocessor::BSPTree, 40
- BuildKdTree
 - GtpVisibilityPreprocessor::Preprocessor, 105
- Camera
 - GtpVisibility, 24
 - VisibilityCamera.h, 156
- CheckFrustumVisible
 - GtpVisibility::HierarchyInterface, 65
 - Ogre::PlatformHierarchyInterface, 92
- COHERENT_HIERARCHICAL_CULLING
 - GtpVisibility::VisibilityEnvironment, 131
- CoherentHierarchicalCullingManager
 - GtpVisibility::CoherentHierarchicalCullingManager, 44
- CoherentHierarchicalCullingManager.h
 - QueryPair, 144
 - QueryQueue, 144
- ComputeCameraVisibility
 - GtpVisibility::DummyQueryManager, 53
 - GtpVisibility::QueryManager, 108
- ComputeFromPointVisibility
 - GtpVisibility::DummyQueryManager, 53
 - GtpVisibility::QueryManager, 108
- ComputeVisibility
 - GtpVisibilityPreprocessor::ExactPreprocessor, 55
 - GtpVisibilityPreprocessor::Preprocessor, 105

- GtpVisibilityPreprocessor::Sampling-Preprocessor, 111
- Containers.h
 - MeshContainer, 175
 - NodeContainer, 175
 - ViewCellContainer, 175
- CullingManager
 - GtpVisibility::CullingManager, 46
- CullingManagerType
 - GtpVisibility::VisibilityEnvironment, 131
- cullingOctreePlugin
 - Ogre, 26
 - OgreVisibilitySceneManagerDll.cpp, 211
- cullingTerrainPlugin
 - Ogre, 26
 - OgreVisibilitySceneManagerDll.cpp, 211
- DeleteQueries
 - Ogre::PlatformHierarchyInterface, 93
- DistanceQueue
 - DistanceQueue.h, 146
 - GtpVisibility, 24
- DistanceQueue.h
 - DistanceQueue, 146
 - HierarchyNode, 146
- dllStartPlugin
 - Ogre, 26
 - OgreVisibilitySceneManagerDll.cpp, 211
- dllStopPlugin
 - Ogre, 26
 - OgreVisibilitySceneManagerDll.cpp, 211
- DummyPreprocessingManager
 - GtpVisibility::DummyPreprocessingManager, 50
- DummyQueryManager
 - GtpVisibility::DummyQueryManager, 53
- EndQuery
 - GtpVisibility::OcclusionQuery, 81
 - Ogre::PlatformOcclusionQuery, 96
- ExportPreprocessedData
 - GtpVisibilityPreprocessor::Preprocessor, 105
- ExportScene
 - GtpVisibility::DummyPreprocessingManager, 50
 - GtpVisibility::PreprocessingManager, 100
- FRUSTUM_CULLING
 - GtpVisibility::VisibilityEnvironment, 131
- FrustumCullingManager
 - GtpVisibility::FrustumCullingManager, 59
- GenerateViewcells
 - GtpVisibilityPreprocessor::Preprocessor, 105
- GetBoundingBox
 - Ogre::BspHierarchyInterface, 33
 - Ogre::OctreeHierarchyInterface, 85
 - Ogre::PlatformHierarchyInterface, 93
 - Ogre::SceneNodeHierarchyInterface, 118
- GetCullingManager
 - GtpVisibility::VisibilityManager, 133
- GetFrameId
 - GtpVisibility::HierarchyInterface, 65
- GetNextOcclusionQuery
 - GtpVisibility::HierarchyInterface, 65
 - Ogre::PlatformHierarchyInterface, 91
- GetOccludeeChecksum
 - GtpVisibility::PreprocessingManager, 101
- GetOccluderChecksum
 - GtpVisibility::PreprocessingManager, 101
- getOption
 - Ogre::VisibilityBspSceneManager, 127
 - Ogre::VisibilityDotSceneManager, 129
 - Ogre::VisibilityOctreeSceneManager, 135
 - Ogre::VisibilitySceneManager, 138
 - Ogre::VisibilityTerrainSceneManager, 141
- getOptionKeys
 - Ogre::VisibilityBspSceneManager, 127
 - Ogre::VisibilityDotSceneManager, 130
 - Ogre::VisibilityOctreeSceneManager, 136
 - Ogre::VisibilitySceneManager, 139
 - Ogre::VisibilityTerrainSceneManager, 142
- getOptionValues
 - Ogre::VisibilityBspSceneManager, 127
 - Ogre::VisibilityDotSceneManager, 129
 - Ogre::VisibilityOctreeSceneManager, 135
 - Ogre::VisibilitySceneManager, 138
 - Ogre::VisibilityTerrainSceneManager, 142
- GetPVS
 - GtpVisibility::PreprocessingManager, 101
- GetQueryResult
 - GtpVisibility::OcclusionQuery, 80
 - Ogre::PlatformOcclusionQuery, 96
- GetQueue
 - GtpVisibility::HierarchyInterface, 65
- GetRoot
 - GtpVisibilityPreprocessor::KdTree, 75
- GetSceneRoot
 - GtpVisibility::HierarchyInterface, 64
- GetSolidHalfBoundingBox
 - Ogre::PlatformHierarchyInterface, 94
- GetSquaredViewDepth
 - Ogre::BspHierarchyInterface, 33
 - Ogre::OctreeHierarchyInterface, 85
- getVisibilityManager
 - Ogre::VisibilityBspSceneManager, 127
 - Ogre::VisibilityDotSceneManager, 130
 - Ogre::VisibilityOctreeSceneManager, 136

- Ogre::VisibilitySceneManager, 139
- Ogre::VisibilityTerrainSceneManager, 142
- getWorldTransforms
 - Ogre::SolidHalfBoundingBox, 119
- GreaterDistance
 - GtpVisibility::GreaterDistance, 61
- GtpVisibility, 23
- GtpVisibility
 - AxisAlignedBox, 24
 - Camera, 24
 - DistanceQueue, 24
 - HierarchyNode, 24
 - Mesh, 24
 - QueryPair, 24
 - QueryQueue, 24
 - Ray, 24
 - Vector3, 24
- GtpVisibility/ Directory Reference, 13
- GtpVisibility/include/ Directory Reference, 17
- GtpVisibility/include/CoherentHierarchicalCullingManager.h, 143
- GtpVisibility/include/CullingManager.h, 145
- GtpVisibility/include/DistanceQueue.h, 146
- GtpVisibility/include/DummyPreprocessingManager.h, 147
- GtpVisibility/include/DummyQueryManager.h, 148
- GtpVisibility/include/FrustumCullingManager.h, 149
- GtpVisibility/include/HierarchyInterface.h, 150
- GtpVisibility/include/OcclusionQuery.h, 151
- GtpVisibility/include/PreprocessingManager.h, 152
- GtpVisibility/include/QueryManager.h, 153
- GtpVisibility/include/StopAndWaitCullingManager.h, 154
- GtpVisibility/include/VisibilityAxisAlignedBox.h, 155
- GtpVisibility/include/VisibilityCamera.h, 156
- GtpVisibility/include/VisibilityEnvironment.h, 157
- GtpVisibility/include/VisibilityInfo.h, 158
- GtpVisibility/include/VisibilityManager.h, 159
- GtpVisibility/include/VisibilityMesh.h, 160
- GtpVisibility/include/VisibilityRay.h, 161
- GtpVisibility/include/VisibilityVector3.h, 162
- GtpVisibility/src/ Directory Reference, 21
- GtpVisibility/src/CoherentHierarchicalCullingManager.cpp, 163
- GtpVisibility/src/CullingManager.cpp, 164
- GtpVisibility/src/DummyPreprocessingManager.cpp, 165
- GtpVisibility/src/DummyQueryManager.cpp, 166
- GtpVisibility/src/FrustumCullingManager.cpp, 167
- GtpVisibility/src/HierarchyInterface.cpp, 168
- GtpVisibility/src/PreprocessingManager.cpp, 169
- GtpVisibility/src/QueryManager.cpp, 170
- GtpVisibility/src/StopAndWaitCullingManager.cpp, 171
- GtpVisibility/src/VisibilityEnvironment.cpp, 172
- GtpVisibility/src/VisibilityManager.cpp, 173
- GtpVisibility::CoherentHierarchicalCullingManager, 42
- GtpVisibility::CoherentHierarchicalCullingManager
 - CoherentHierarchicalCullingManager, 44
 - RenderScene, 44
- GtpVisibility::CullingManager, 45
- GtpVisibility::CullingManager
 - CullingManager, 46
 - mHierarchyInterface, 47
 - mNumFrustumCulledNodes, 47
 - mNumQueryCulledNodes, 47
 - mVisibilityThreshold, 47
 - RenderScene, 46
 - SetHierarchyInterface, 46
- GtpVisibility::DummyPreprocessingManager, 48
- GtpVisibility::DummyPreprocessingManager
 - AddViewCellPVS, 50
 - DummyPreprocessingManager, 50
 - ExportScene, 50
 - LoadPreprocessedData, 50
 - LocateViewCellIds, 50
- GtpVisibility::DummyQueryManager, 51
- GtpVisibility::DummyQueryManager
 - ComputeCameraVisibility, 53
 - ComputeFromPointVisibility, 53
 - DummyQueryManager, 53
- GtpVisibility::FrustumCullingManager, 57
- GtpVisibility::FrustumCullingManager
 - FrustumCullingManager, 59
 - RenderScene, 59
- GtpVisibility::GreaterDistance, 60
- GtpVisibility::GreaterDistance
 - GreaterDistance, 61
 - mHierarchyInterface, 61
 - operator(), 61
- GtpVisibility::HierarchyInterface, 62
- GtpVisibility::HierarchyInterface
 - ~HierarchyInterface, 63
 - CheckFrustumVisible, 65
- GetFrameId, 65
- GetNextOcclusionQuery, 65
- GetQueue, 65
- GetSceneRoot, 64
- HasGeometry, 66
- HasGreaterDistance, 65
- HierarchyInterface, 63
- InitFrame, 64
- IsLeaf, 63
- IsNodeVisible, 66

- IssueOcclusionQuery, 64
- LastVisited, 66
- mCurrentTestIdx, 67
- mDistanceQueue, 67
- mFrameId, 67
- mNumRenderedGeometry, 67
- mNumRenderedNodes, 67
- mNumSceneNodes, 67
- mNumTraversedNodes, 67
- mSceneRoot, 67
- PullUpVisibility, 64
- RenderNode, 64
- SetLastVisited, 66
- SetNodeVisible, 66
- SetSceneRoot, 64
- TraverseNode, 63
- GtpVisibility::MeshInfo, 78
- GtpVisibility::MeshInfo
 - MeshInfo, 78
 - mMesh, 78
 - mVisibility, 78
- GtpVisibility::NodeInfo, 79
- GtpVisibility::NodeInfo
 - mNode, 79
 - mVisibility, 79
 - NodeInfo, 79
- GtpVisibility::OcclusionQuery, 80
- GtpVisibility::OcclusionQuery
 - ~OcclusionQuery, 80
 - BeginQuery, 81
 - EndQuery, 81
 - GetQueryResult, 80
 - ResultAvailable, 80
- GtpVisibility::PreprocessingManager, 98
- GtpVisibility::PreprocessingManager
 - ~PreprocessingManager, 100
 - AddViewCellPVS, 102
 - ExportScene, 100
 - GetOccludeeChecksum, 101
 - GetOccluderChecksum, 101
 - GetPVS, 101
 - LoadPreprocessedData, 100
 - LocateViewCellIds, 102
 - mSceneTraverser, 102
 - PreprocessingManager, 100
 - SetSceneTraverser, 101
- GtpVisibility::QueryManager, 107
- GtpVisibility::QueryManager
 - ComputeCameraVisibility, 108
 - ComputeFromPointVisibility, 108
 - mHierarchyInterface, 109
 - QueryManager, 108
 - SetSceneTraverser, 109
 - ShootRay, 109
- GtpVisibility::StopAndWaitCullingManager, 121
- GtpVisibility::StopAndWaitCullingManager
 - RenderScene, 123
 - StopAndWaitCullingManager, 123
- GtpVisibility::VisibilityEnvironment, 131
 - COHERENT_HIERARCHICAL_CULLING, 131
 - FRUSTUM_CULLING, 131
 - STOP_AND_WAIT, 131
- GtpVisibility::VisibilityEnvironment
 - CullingManagerType, 131
 - LoadEnvironment, 131
 - VisibilityEnvironment, 131
- GtpVisibility::VisibilityManager, 132
- GtpVisibility::VisibilityManager
 - ~VisibilityManager, 133
 - ApplyVisibilityCulling, 133
 - GetCullingManager, 133
 - mCullingManager, 133
 - mCullingManagerType, 133
 - mPreprocessingManager, 133
 - mQueryManager, 133
 - mVisibilityEnvironment, 133
 - SetCullingManager, 133
 - VisibilityManager, 132
- GtpVisibilityPreprocessor, 25
- GtpVisibilityPreprocessor
 - MeshContainer, 25
 - NodeContainer, 25
 - ViewCellContainer, 25
- GtpVisibilityPreprocessor/ Directory Reference, 14
- GtpVisibilityPreprocessor/include/ Directory Reference, 16
- GtpVisibilityPreprocessor/include/AxisAlignedBox3.h, 174
- GtpVisibilityPreprocessor/include/Containers.h, 175
- GtpVisibilityPreprocessor/include/ExactPreprocessor.h, 176
- GtpVisibilityPreprocessor/include/KdTree.h, 177
- GtpVisibilityPreprocessor/include/Mesh.h, 178
- GtpVisibilityPreprocessor/include/Plane3.h, 179
- GtpVisibilityPreprocessor/include/Preprocessor.h, 180
- GtpVisibilityPreprocessor/include/SamplingPreprocessor.h, 181
- GtpVisibilityPreprocessor/include/SceneGraph.h, 182
- GtpVisibilityPreprocessor/include/Vector3.h, 183
- GtpVisibilityPreprocessor/include/ViewCellBSP.h, 184
- GtpVisibilityPreprocessor/src/ Directory Reference, 20

- GtpVisibilityPreprocessor/src/ExactPreprocessor.cpp, GtpVisibilityPreprocessor::KdTree
185
- GtpVisibilityPreprocessor/src/KdTree.cpp, 186
- GtpVisibilityPreprocessor/src/Preprocessor.cpp,
187
- GtpVisibilityPreprocessor/src/SamplingPreprocessor.cpp,
188
- GtpVisibilityPreprocessor::AxisAlignedBox3, 29
- GtpVisibilityPreprocessor::AxisAlignedBox3
max, 29
min, 29
- GtpVisibilityPreprocessor::BSPInterior, 34
- GtpVisibilityPreprocessor::BSPInterior
IsLeaf, 35
mBack, 35
mFront, 35
mPlane, 35
- GtpVisibilityPreprocessor::BSPLeaf, 36
- GtpVisibilityPreprocessor::BSPLeaf
BSPLeaf, 37
IsLeaf, 37
mViewCell, 37
- GtpVisibilityPreprocessor::BSPNode, 38
- GtpVisibilityPreprocessor::BSPNode
IsLeaf, 39
IsRoot, 39
parent, 39
- GtpVisibilityPreprocessor::BSPTree, 40
- GtpVisibilityPreprocessor::BSPTree
BSPTree, 40
mRoot, 40
mRootCell, 40
- GtpVisibilityPreprocessor::ExactPreprocessor, 54
- GtpVisibilityPreprocessor::ExactPreprocessor
ComputeVisibility, 55
- GtpVisibilityPreprocessor::KdInterior, 68
- GtpVisibilityPreprocessor::KdInterior
axis, 69
IsLeaf, 69
mBack, 69
mFront, 69
mPosition, 69
- GtpVisibilityPreprocessor::KdLeaf, 70
- GtpVisibilityPreprocessor::KdLeaf
IsLeaf, 71
mOccludees, 71
mOccluders, 71
mViewCells, 71
- GtpVisibilityPreprocessor::KdNode, 72
- GtpVisibilityPreprocessor::KdNode
IsLeaf, 73
IsRoot, 73
mParent, 73
- GtpVisibilityPreprocessor::KdTree, 74
- GetRoot, 75
- InsertOccluee, 74
- InsertOccluder, 74
- InsertViewCell, 75
- mBox, 75
- mRoot, 75
- Subdivide, 75
- GtpVisibilityPreprocessor::Mesh, 76
- GtpVisibilityPreprocessor::Mesh
Mesh, 76
mPatches, 77
mVertices, 77
PatchContainer, 76
VertexContainer, 76
- GtpVisibilityPreprocessor::Patch, 87
- GtpVisibilityPreprocessor::Patch
mVertices, 87
Patch, 87
- GtpVisibilityPreprocessor::Plane3, 88
- GtpVisibilityPreprocessor::Plane3
mD, 88
mNormal, 88
- GtpVisibilityPreprocessor::Preprocessor, 103
- GtpVisibilityPreprocessor::Preprocessor
BuildKdTree, 105
ComputeVisibility, 105
ExportPreprocessedData, 105
GenerateViewcells, 105
LoadScene, 104
LoadViewcells, 104
mKdTree, 106
mOccludees, 106
mOccluders, 106
mSceneGraph, 106
mViewCellBSPTree, 106
mViewcells, 106
- GtpVisibilityPreprocessor::SamplingPreprocessor,
110
- GtpVisibilityPreprocessor::SamplingPreprocessor
ComputeVisibility, 111
- GtpVisibilityPreprocessor::SceneGraph, 113
- GtpVisibilityPreprocessor::SceneGraph
mRoot, 113
- GtpVisibilityPreprocessor::SceneGraphNode, 114
- GtpVisibilityPreprocessor::SceneGraphNode
mBox, 114
mChildren, 114
mGeometry, 114
- GtpVisibilityPreprocessor::Vector3, 124
- GtpVisibilityPreprocessor::Vector3
x, 124
y, 124
z, 124

- HasGeometry
 - GtpVisibility::HierarchyInterface, 66
 - Ogre::BspHierarchyInterface, 33
 - Ogre::OctreeHierarchyInterface, 85
 - Ogre::SceneNodeHierarchyInterface, 118
- HasGreaterDistance
 - GtpVisibility::HierarchyInterface, 65
 - Ogre::BspHierarchyInterface, 33
 - Ogre::OctreeHierarchyInterface, 85
 - Ogre::SceneNodeHierarchyInterface, 118
- heightmapTerrainPageSource
 - Ogre, 26
 - OgreVisibilitySceneManagerDll.cpp, 211
- HierarchyInterface
 - GtpVisibility::HierarchyInterface, 63
- HierarchyNode
 - DistanceQueue.h, 146
 - GtpVisibility, 24
- InfoContainer
 - VisibilityInfo.h, 158
- InitFrame
 - GtpVisibility::HierarchyInterface, 64
 - Ogre::PlatformHierarchyInterface, 92
- InsertOccludee
 - GtpVisibilityPreprocessor::KdTree, 74
- InsertOccluder
 - GtpVisibilityPreprocessor::KdTree, 74
- InsertViewCell
 - GtpVisibilityPreprocessor::KdTree, 75
- IsLeaf
 - GtpVisibility::HierarchyInterface, 63
 - GtpVisibilityPreprocessor::BSPInterior, 35
 - GtpVisibilityPreprocessor::BSPLeaf, 37
 - GtpVisibilityPreprocessor::BSPNode, 39
 - GtpVisibilityPreprocessor::KdInterior, 69
 - GtpVisibilityPreprocessor::KdLeaf, 71
 - GtpVisibilityPreprocessor::KdNode, 73
 - Ogre::BspHierarchyInterface, 33
 - Ogre::OctreeHierarchyInterface, 85
 - Ogre::SceneNodeHierarchyInterface, 117
- IsNodeVisible
 - GtpVisibility::HierarchyInterface, 66
 - Ogre::BspHierarchyInterface, 33
 - Ogre::OctreeHierarchyInterface, 85
 - Ogre::SceneNodeHierarchyInterface, 118
- IsRoot
 - GtpVisibilityPreprocessor::BSPNode, 39
 - GtpVisibilityPreprocessor::KdNode, 73
- IssueOcclusionQuery
 - GtpVisibility::HierarchyInterface, 64
 - Ogre::PlatformHierarchyInterface, 93
- LastVisited
 - GtpVisibility::HierarchyInterface, 66
 - Ogre::BspHierarchyInterface, 33
 - Ogre::OctreeHierarchyInterface, 85
 - Ogre::SceneNodeHierarchyInterface, 118
- LoadEnvironment
 - GtpVisibility::VisibilityEnvironment, 131
- LoadPreprocessedData
 - GtpVisibility::DummyPreprocessingManager, 50
 - GtpVisibility::PreprocessingManager, 100
- LoadScene
 - GtpVisibilityPreprocessor::Preprocessor, 104
- LoadViewcells
 - GtpVisibilityPreprocessor::Preprocessor, 104
- LocateViewCellIds
 - GtpVisibility::DummyPreprocessingManager, 50
 - GtpVisibility::PreprocessingManager, 102
- main
 - Preprocessor.cpp, 187
- max
 - GtpVisibilityPreprocessor::AxisAlignedBox3, 29
- mBack
 - GtpVisibilityPreprocessor::BSPInterior, 35
 - GtpVisibilityPreprocessor::KdInterior, 69
- mBox
 - GtpVisibilityPreprocessor::KdTree, 75
 - GtpVisibilityPreprocessor::SceneGraphNode, 114
 - Ogre::PlatformHierarchyInterface, 94
- mCamera
 - Ogre::PlatformHierarchyInterface, 94
- mChildren
 - GtpVisibilityPreprocessor::SceneGraphNode, 114
- mCullingManager
 - GtpVisibility::VisibilityManager, 133
- mCullingManagerType
 - GtpVisibility::VisibilityManager, 133
- mCurrentTestIdx
 - GtpVisibility::HierarchyInterface, 67
- mD
 - GtpVisibilityPreprocessor::Plane3, 88
- mDistanceQueue
 - GtpVisibility::HierarchyInterface, 67
- Mesh
 - GtpVisibility, 24
 - GtpVisibilityPreprocessor::Mesh, 76
 - VisibilityMesh.h, 160
- MeshContainer
 - Containers.h, 175
 - GtpVisibilityPreprocessor, 25

- MeshInfo
 - GtpVisibility::MeshInfo, 78
- mFrameId
 - GtpVisibility::HierarchyInterface, 67
- mFront
 - GtpVisibilityPreprocessor::BSPInterior, 35
 - GtpVisibilityPreprocessor::KdInterior, 69
- mGeometry
 - GtpVisibilityPreprocessor::SceneGraphNode, 114
- mHalfBoundingBox
 - Ogre::PlatformHierarchyInterface, 94
- mHardwareOcclusionQuery
 - Ogre::PlatformOcclusionQuery, 97
- mHierarchyInterface
 - GtpVisibility::CullingManager, 47
 - GtpVisibility::GreaterDistance, 61
 - GtpVisibility::QueryManager, 109
 - Ogre::VisibilityBspSceneManager, 127
 - Ogre::VisibilityDotSceneManager, 130
 - Ogre::VisibilityOctreeSceneManager, 136
 - Ogre::VisibilitySceneManager, 139
 - Ogre::VisibilityTerrainSceneManager, 142
- min
 - GtpVisibilityPreprocessor::AxisAlignedBox3, 29
- mIsFirstHalf
 - Ogre::SolidHalfBoundingBox, 120
- mKdTree
 - GtpVisibilityPreprocessor::Preprocessor, 106
- mMesh
 - GtpVisibility::MeshInfo, 78
- mNode
 - GtpVisibility::NodeInfo, 79
- mNormal
 - GtpVisibilityPreprocessor::Plane3, 88
- mNumFrustumCulledNodes
 - GtpVisibility::CullingManager, 47
- mNumOctreeNodeNodes
 - Ogre::BspHierarchyInterface, 33
 - Ogre::OctreeHierarchyInterface, 86
- mNumQueryCulledNodes
 - GtpVisibility::CullingManager, 47
- mNumRenderedGeometry
 - GtpVisibility::HierarchyInterface, 67
- mNumRenderedNodes
 - GtpVisibility::HierarchyInterface, 67
- mNumSceneNodes
 - GtpVisibility::HierarchyInterface, 67
- mNumTraversedNodes
 - GtpVisibility::HierarchyInterface, 67
- mOccludees
 - GtpVisibilityPreprocessor::KdLeaf, 71
 - GtpVisibilityPreprocessor::Preprocessor, 106
- mOccluders
 - GtpVisibilityPreprocessor::KdLeaf, 71
 - GtpVisibilityPreprocessor::Preprocessor, 106
- mOcclusionQueries
 - Ogre::PlatformHierarchyInterface, 94
- mParent
 - GtpVisibilityPreprocessor::KdNode, 73
- mPatches
 - GtpVisibilityPreprocessor::Mesh, 77
- mPlane
 - GtpVisibilityPreprocessor::BSPInterior, 35
- mPosition
 - GtpVisibilityPreprocessor::KdInterior, 69
- mPreprocessingManager
 - GtpVisibility::VisibilityManager, 133
- mQueryManager
 - GtpVisibility::VisibilityManager, 133
- mRenderSystem
 - Ogre::PlatformHierarchyInterface, 94
- mRoot
 - GtpVisibilityPreprocessor::BSPTree, 40
 - GtpVisibilityPreprocessor::KdTree, 75
 - GtpVisibilityPreprocessor::SceneGraph, 113
- mRootCell
 - GtpVisibilityPreprocessor::BSPTree, 40
- mSceneGraph
 - GtpVisibilityPreprocessor::Preprocessor, 106
- mSceneManager
 - Ogre::PlatformHierarchyInterface, 94
- mSceneRoot
 - GtpVisibility::HierarchyInterface, 67
- mSceneTraverser
 - GtpVisibility::PreprocessingManager, 102
- mVertices
 - GtpVisibilityPreprocessor::Mesh, 77
 - GtpVisibilityPreprocessor::Patch, 87
- mViewCell
 - GtpVisibilityPreprocessor::BSPLeaf, 37
- mViewCellBSPTree
 - GtpVisibilityPreprocessor::Preprocessor, 106
- mViewCells
 - GtpVisibilityPreprocessor::KdLeaf, 71
- mViewcells
 - GtpVisibilityPreprocessor::Preprocessor, 106
- mVisibility
 - GtpVisibility::MeshInfo, 78
 - GtpVisibility::NodeInfo, 79
- mVisibilityEnvironment
 - GtpVisibility::VisibilityManager, 133
- mVisibilityManager
 - Ogre::VisibilityBspSceneManager, 127
 - Ogre::VisibilityDotSceneManager, 130
 - Ogre::VisibilityOctreeSceneManager, 136
 - Ogre::VisibilitySceneManager, 139

- Ogre::VisibilityTerrainSceneManager, 142
- mVisibilityThreshold
 - GtpVisibility::CullingManager, 47
- NodeContainer
 - Containers.h, 175
 - GtpVisibilityPreprocessor, 25
- NodeInfo
 - GtpVisibility::NodeInfo, 79
- OctreeHierarchyInterface
 - Ogre::OctreeHierarchyInterface, 84
- Ogre, 26
 - cullingOctreePlugin, 26
 - cullingTerrainPlugin, 26
 - dllStartPlugin, 26
 - dllStopPlugin, 26
 - heightmapTerrainPageSource, 26
- Ogre/ Directory Reference, 18
- Ogre/include/ Directory Reference, 15
- Ogre/include/OgreBspHierarchyInterface.h, 189
- Ogre/include/OgreOctreeHierarchyInterface.h, 190
- Ogre/include/OgrePlatformHierarchyInterface.h, 191
- Ogre/include/OgrePlatformOcclusionQuery.h, 192
- Ogre/include/OgreSceneNodeHierarchyInterface.h, 193
- Ogre/include/OgreSolidHalfBoundingBox.h, 194
- Ogre/include/OgreVisibilityBspSceneManager.h, 195
- Ogre/include/OgreVisibilityDotSceneManager.h, 196
- Ogre/include/OgreVisibilityOctreeSceneManager.h, 197
- Ogre/include/OgreVisibilitySceneManager.h, 198
- Ogre/include/OgreVisibilityTerrainSceneManager.h, 199
- Ogre/src/ Directory Reference, 19
- Ogre/src/OgreBspHierarchyInterface.cpp, 200
- Ogre/src/OgreOctreeHierarchyInterface.cpp, 201
- Ogre/src/OgrePlatformHierarchyInterface.cpp, 202
- Ogre/src/OgrePlatformOcclusionQuery.cpp, 203
- Ogre/src/OgreSceneNodeHierarchyInterface.cpp, 204
- Ogre/src/OgreSolidHalfBoundingBox.cpp, 205
- Ogre/src/OgreVisibilityBspSceneManager.cpp, 206
- Ogre/src/OgreVisibilityDotSceneManager.cpp, 207
- Ogre/src/OgreVisibilityOctreeSceneManager.cpp, 208
- Ogre/src/OgreVisibilitySceneManager.cpp, 209
- Ogre/src/OgreVisibilitySceneManagerDll.cpp, 210
- Ogre/src/OgreVisibilityTerrainSceneManager.cpp, 212
- Ogre::BspHierarchyInterface, 30
- Ogre::BspHierarchyInterface
 - BspHierarchyInterface, 32
 - GetBoundingBox, 33
 - GetSquaredViewDepth, 33
 - HasGeometry, 33
 - HasGreaterDistance, 33
 - IsLeaf, 33
 - IsNodeVisible, 33
 - LastVisited, 33
 - mNumOctreeNode, 33
 - PullUpVisibility, 32
 - RenderNode, 32
 - SetLastVisited, 33
 - SetNodeVisible, 33
 - TraverseNode, 32
- Ogre::OctreeHierarchyInterface, 82
- Ogre::OctreeHierarchyInterface
 - GetBoundingBox, 85
 - GetSquaredViewDepth, 85
 - HasGeometry, 85
 - HasGreaterDistance, 85
 - IsLeaf, 85
 - IsNodeVisible, 85
 - LastVisited, 85
 - mNumOctreeNode, 86
 - OctreeHierarchyInterface, 84
 - PullUpVisibility, 84
 - RenderNode, 84
 - SetLastVisited, 85
 - SetNodeVisible, 85
 - SetNumOctreeNode, 84
 - TraverseNode, 84
- Ogre::PlatformHierarchyInterface, 89
- Ogre::PlatformHierarchyInterface
 - ~PlatformHierarchyInterface, 91
 - CheckFrustumVisible, 92
 - DeleteQueries, 93
 - GetBoundingBox, 93
 - GetNextOcclusionQuery, 91
 - GetSolidHalfBoundingBox, 94
 - InitFrame, 92
 - IssueOcclusionQuery, 93
 - mBox, 94
 - mCamera, 94
 - mHalfBoundingBox, 94
 - mOcclusionQueries, 94
 - mRenderSystem, 94
 - mSceneManager, 94
 - PlatformHierarchyInterface, 91
 - RenderBoundingBox, 93
 - SetCamera, 91
 - SetRenderSystem, 92
 - SetSceneManager, 92
- Ogre::PlatformOcclusionQuery, 95

- Ogre::PlatformOcclusionQuery
 - ~PlatformOcclusionQuery, 96
 - BeginQuery, 96
 - EndQuery, 96
 - GetQueryResult, 96
 - mHardwareOcclusionQuery, 97
 - PlatformOcclusionQuery, 96
 - ResultAvailable, 96
- Ogre::SceneNodeHierarchyInterface, 115
- Ogre::SceneNodeHierarchyInterface
 - GetBoundingBox, 118
 - HasGeometry, 118
 - HasGreaterDistance, 118
 - IsLeaf, 117
 - IsNodeVisible, 118
 - LastVisited, 118
 - PullUpVisibility, 118
 - RenderNode, 117
 - SceneNodeHierarchyInterface, 117
 - SetLastVisited, 118
 - SetNodeVisible, 118
 - TraverseNode, 117
- Ogre::SolidHalfBoundingBox, 119
- Ogre::SolidHalfBoundingBox
 - getWorldTransforms, 119
 - mIsFirstHalf, 120
 - setOcclusionQueryMaterial, 120
 - setupBoundingBox, 119
 - setupBoundingBoxVertices, 120
 - SolidHalfBoundingBox, 119
- Ogre::VisibilityBspSceneManager, 125
- Ogre::VisibilityBspSceneManager
 - ~VisibilityBspSceneManager, 126
 - _findVisibleObjects, 126
 - _renderVisibleObjects, 126
 - _updateSceneGraph, 126
 - getOption, 127
 - getOptionKeys, 127
 - getOptionValues, 127
 - getVisibilityManager, 127
 - mHierarchyInterface, 127
 - mVisibilityManager, 127
 - setOption, 126
 - setVisibilityManager, 127
 - VisibilityBspSceneManager, 126
- Ogre::VisibilityDotSceneManager, 128
- Ogre::VisibilityDotSceneManager
 - ~VisibilityDotSceneManager, 129
 - _findVisibleObjects, 129
 - _renderVisibleObjects, 129
 - _updateSceneGraph, 129
 - getOption, 129
 - getOptionKeys, 130
 - getOptionValues, 129
 - getVisibilityManager, 130
 - mHierarchyInterface, 130
 - mVisibilityManager, 130
 - setOption, 129
 - setVisibilityManager, 130
 - VisibilityDotSceneManager, 129
- Ogre::VisibilityOctreeSceneManager, 134
- Ogre::VisibilityOctreeSceneManager
 - ~VisibilityOctreeSceneManager, 135
 - _findVisibleObjects, 135
 - _renderVisibleObjects, 135
 - _updateSceneGraph, 135
 - getOption, 135
 - getOptionKeys, 136
 - getOptionValues, 135
 - getVisibilityManager, 136
 - mHierarchyInterface, 136
 - mVisibilityManager, 136
 - setOption, 135
 - setVisibilityManager, 136
 - VisibilityOctreeSceneManager, 135
- Ogre::VisibilitySceneManager, 137
- Ogre::VisibilitySceneManager
 - ~VisibilitySceneManager, 138
 - _findVisibleObjects, 138
 - _renderVisibleObjects, 138
 - _updateSceneGraph, 138
 - getOption, 138
 - getOptionKeys, 139
 - getOptionValues, 138
 - getVisibilityManager, 139
 - mHierarchyInterface, 139
 - mVisibilityManager, 139
 - setOption, 138
 - setVisibilityManager, 139
 - VisibilitySceneManager, 138
- Ogre::VisibilityTerrainSceneManager, 140
- Ogre::VisibilityTerrainSceneManager
 - ~VisibilityTerrainSceneManager, 141
 - _findVisibleObjects, 141
 - _renderVisibleObjects, 141
 - _updateSceneGraph, 141
 - getOption, 141
 - getOptionKeys, 142
 - getOptionValues, 142
 - getVisibilityManager, 142
 - mHierarchyInterface, 142
 - mVisibilityManager, 142
 - setOption, 141
 - setVisibilityManager, 142
 - VisibilityTerrainSceneManager, 141
- OgreSolidHalfBoundingBox.cpp
 - POSITION_BINDING, 205
- OgreVisibilitySceneManagerDll.cpp

- cullingOctreePlugin, 211
- cullingTerrainPlugin, 211
- dllStartPlugin, 211
- dllStopPlugin, 211
- heightmapTerrainPageSource, 211
- visEnv, 211
- visManager, 211
- operator()
 - GtpVisibility::GreaterDistance, 61
- parent
 - GtpVisibilityPreprocessor::BSPNode, 39
- Patch
 - GtpVisibilityPreprocessor::Patch, 87
- PatchContainer
 - GtpVisibilityPreprocessor::Mesh, 76
- PlatformHierarchyInterface
 - Ogre::PlatformHierarchyInterface, 91
- PlatformOcclusionQuery
 - Ogre::PlatformOcclusionQuery, 96
- POSITION_BINDING
 - OgreSolidHalfBoundingBox.cpp, 205
- PreprocessingManager
 - GtpVisibility::PreprocessingManager, 100
- Preprocessor.cpp
 - main, 187
- PullUpVisibility
 - GtpVisibility::HierarchyInterface, 64
 - Ogre::BspHierarchyInterface, 32
 - Ogre::OctreeHierarchyInterface, 84
 - Ogre::SceneNodeHierarchyInterface, 118
- QueryManager
 - GtpVisibility::QueryManager, 108
- QueryPair
 - CoherentHierarchicalCullingManager.h, 144
 - GtpVisibility, 24
- QueryQueue
 - CoherentHierarchicalCullingManager.h, 144
 - GtpVisibility, 24
- Ray
 - GtpVisibility, 24
 - VisibilityRay.h, 161
- RenderBoundingBox
 - Ogre::PlatformHierarchyInterface, 93
- RenderNode
 - GtpVisibility::HierarchyInterface, 64
 - Ogre::BspHierarchyInterface, 32
 - Ogre::OctreeHierarchyInterface, 84
 - Ogre::SceneNodeHierarchyInterface, 117
- RenderScene
 - GtpVisibility::CoherentHierarchicalCulling-Manager, 44
 - GtpVisibility::CullingManager, 46
 - GtpVisibility::FrustumCullingManager, 59
 - GtpVisibility::StopAndWaitCullingManager, 123
- ResultAvailable
 - GtpVisibility::OcclusionQuery, 80
 - Ogre::PlatformOcclusionQuery, 96
- SceneNodeHierarchyInterface
 - Ogre::SceneNodeHierarchyInterface, 117
- SetCamera
 - Ogre::PlatformHierarchyInterface, 91
- SetCullingManager
 - GtpVisibility::VisibilityManager, 133
- SetHierarchyInterface
 - GtpVisibility::CullingManager, 46
- SetLastVisited
 - GtpVisibility::HierarchyInterface, 66
 - Ogre::BspHierarchyInterface, 33
 - Ogre::OctreeHierarchyInterface, 85
 - Ogre::SceneNodeHierarchyInterface, 118
- SetNodeVisible
 - GtpVisibility::HierarchyInterface, 66
 - Ogre::BspHierarchyInterface, 33
 - Ogre::OctreeHierarchyInterface, 85
 - Ogre::SceneNodeHierarchyInterface, 118
- SetNumOctreeNode
 - Ogre::OctreeHierarchyInterface, 84
- setOcclusionQueryMaterial
 - Ogre::SolidHalfBoundingBox, 120
- setOption
 - Ogre::VisibilityBspSceneManager, 126
 - Ogre::VisibilityDotSceneManager, 129
 - Ogre::VisibilityOctreeSceneManager, 135
 - Ogre::VisibilitySceneManager, 138
 - Ogre::VisibilityTerrainSceneManager, 141
- SetRenderSystem
 - Ogre::PlatformHierarchyInterface, 92
- SetSceneManager
 - Ogre::PlatformHierarchyInterface, 92
- SetSceneRoot
 - GtpVisibility::HierarchyInterface, 64
- SetSceneTraverser
 - GtpVisibility::PreprocessingManager, 101
 - GtpVisibility::QueryManager, 109
- setupBoundingBox
 - Ogre::SolidHalfBoundingBox, 119
- setupBoundingBoxVertices
 - Ogre::SolidHalfBoundingBox, 120
- setVisibilityManager
 - Ogre::VisibilityBspSceneManager, 127
 - Ogre::VisibilityDotSceneManager, 130
 - Ogre::VisibilityOctreeSceneManager, 136
 - Ogre::VisibilitySceneManager, 139

- Ogre::VisibilityTerrainSceneManager, [142](#)
- ShootRay
 - GtpVisibility::QueryManager, [109](#)
- SolidHalfBoundingBox
 - Ogre::SolidHalfBoundingBox, [119](#)
- std, [27](#)
- STOP_AND_WAIT
 - GtpVisibility::VisibilityEnvironment, [131](#)
- StopAndWaitCullingManager
 - GtpVisibility::StopAndWaitCullingManager, [123](#)
- Subdivide
 - GtpVisibilityPreprocessor::KdTree, [75](#)
- TraverseNode
 - GtpVisibility::HierarchyInterface, [63](#)
 - Ogre::BspHierarchyInterface, [32](#)
 - Ogre::OctreeHierarchyInterface, [84](#)
 - Ogre::SceneNodeHierarchyInterface, [117](#)
- Vector3
 - GtpVisibility, [24](#)
 - VisibilityVector3.h, [162](#)
- VertexContainer
 - GtpVisibilityPreprocessor::Mesh, [76](#)
- ViewCellContainer
 - Containers.h, [175](#)
 - GtpVisibilityPreprocessor, [25](#)
- visEnv
 - OgreVisibilitySceneManagerDll.cpp, [211](#)
- VisibilityAxisAlignedBox.h
 - AxisAlignedBox, [155](#)
- VisibilityBspSceneManager
 - Ogre::VisibilityBspSceneManager, [126](#)
- VisibilityCamera.h
 - Camera, [156](#)
- VisibilityDotSceneManager
 - Ogre::VisibilityDotSceneManager, [129](#)
- VisibilityEnvironment
 - GtpVisibility::VisibilityEnvironment, [131](#)
- VisibilityInfo.h
 - InfoContainer, [158](#)
- VisibilityManager
 - GtpVisibility::VisibilityManager, [132](#)
- VisibilityMesh.h
 - Mesh, [160](#)
- VisibilityOctreeSceneManager
 - Ogre::VisibilityOctreeSceneManager, [135](#)
- VisibilityRay.h
 - Ray, [161](#)
- VisibilitySceneManager
 - Ogre::VisibilitySceneManager, [138](#)
- VisibilityTerrainSceneManager
 - Ogre::VisibilityTerrainSceneManager, [141](#)
- VisibilityVector3.h
 - Vector3, [162](#)
- visManager
 - OgreVisibilitySceneManagerDll.cpp, [211](#)
- x
 - GtpVisibilityPreprocessor::Vector3, [124](#)
- y
 - GtpVisibilityPreprocessor::Vector3, [124](#)
- z
 - GtpVisibilityPreprocessor::Vector3, [124](#)