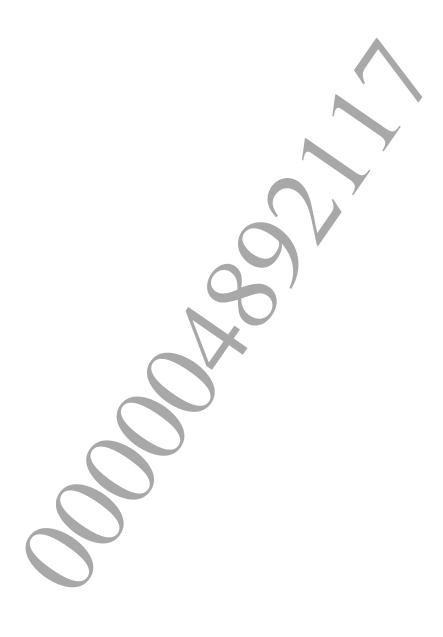


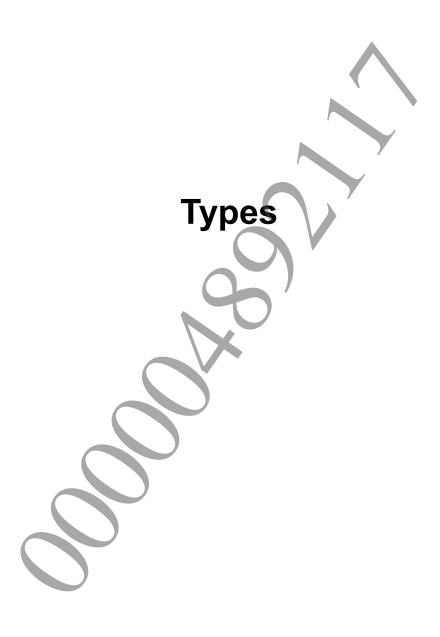
© 2011 Sony Computer Entertainment Inc. All Rights Reserved. SCE Confidential

# **Table of Contents**

Types		4
	colladaRenderUtil::InstanceVisualScene	5
	colladaRenderUtil::collada::VertexInput	6
colladaF	RenderUtil::SimpleRenderer	7
	colladaRenderUtil::SimpleRenderer	
	colladaRenderUtil::SimpleRenderer::SceneInfo	9
	colladaRenderUtil::SimpleRenderer::initialize	10
	colladaRenderUtil::SimpleRenderer::finalize	
	colladaRenderUtil::SimpleRenderer::loadCollada	12
	colladaRenderUtil::SimpleRenderer::disposeCollada	13
	colladaRenderUtil::SimpleRenderer::instantiateVisualScene	14
	colladaRenderUtil::SimpleRenderer::setTime	15
	colladaRenderUtil::SimpleRenderer::disposeVisualScene	
	colladaRenderUtil::SimpleRenderer::render	17
colladaF	RenderUtil::Collada	18
	colladaRenderUtil::Collada::getLibraryGeometries	19
	colladaRenderUtil::Collada::getLibraryImages	20
colladaF	RenderUtil::collada::geometry::LibraryGeometries	21
	colladaRenderUtil::collada::geometry::LibraryGeometries::getNumGeometries	
	colladaRenderUtil::collada::geometry::LibraryGeometries::getGeometry	
colladaF	RenderUtil::collada::geometry::Geometry	24
	colladaRenderUtil::collada::geometry::Geometry::getMesh	
collada	RenderUtil::collada::geometry::Mesh	
oonaaa	colladaRenderUtil::collada::geometry::Mesh::getNumTriangless	
	colladaRenderUtil::collada::geometry::Mesh::getTriangles	
collada	RenderUtil::collada::geometry::Triangles	
Colladar	colladaRenderUtil::collada::geometry::Triangles::getIndices	
	colladaRenderUtil::collada::geometry::Triangles::getNumIndices	
	colladaRenderUtil::collada::geometry::Triangles::getNumVertexInput	
	colladaRenderUtil::collada::geometry::Triangles::getVertexInput	
	colladaRenderUtil::collada::geometry::Triangles::getNumVertices	
	colladaRenderUtil::collada::geometry::Triangles::getStream	
	colladaRenderUtil::collada::geometry::Triangles::getStride	
	colladaRenderUtil::collada::geometry::Triangles::getSemanticOffset	
collada	RenderUtil::collada::Image::LibraryImages	
oonaaa	colladaRenderUtil::collada::Image::LibraryImages::getNumImages	
	colladaRenderUtil::collada::Image::LibraryImages::getImage	
collada	RenderUtil::ShaderParameterManager	
Conauar	colladaRenderUtil::ShaderParameterManager::getParameterId	
	colladaRenderUtil::ShaderParameterManager::getSource	
	colladaRenderUtil::ShaderParameterManager::setNodeToSceneRootMatrix	
	colladaRenderUtil::ShaderParameterManager::setSceneRootToWorldMatrix	
	colladaRenderUtil::ShaderParameterManager::setViewProjectionMatrix	

colladaRenderUtil::ShaderParameterManager::getTextureParameterId	.47
colladaRenderl Itil··ShaderParameterManager··getSourceTexture	48





# colladaRenderUtil::InstanceVisualScene

## COLLADA scene instance

#### **Definition**

## **Description**

This is a COLLADA scene instance.

It retains scene-specific data such as node information and animation time. Multiple scene instances can be created from one Collada.

Keep the source Collada until all created scene instances are disposed of.



# colladaRenderUtil::collada::VertexInput

### Vertex information in vertex stream

#### **Definition**

```
#include <colladaRenderUtil.h>
namespace colladaRenderUtil {
        namespace collada {
              struct VertexInput
                      VertexSemantic mSemantic;
                      int mInputSet;
                      int mOffsetInVertex;
               };
}
```

#### **Members**

mSemantic mInputSet

Semantic of the vertex data Input set of the vertex data mOffsetInVertex Offset in the vertex data

## **Description**

This indicates the data of one vertex attribute in the vertex data. One of the following enters into mSemantic.

Macro	Value
SEMANTIC_BINORMAL	0
SEMANTIC_COLOR	1
SEMANTIC_CONTINUITY	2
SEMANTIC_IMAGE	3
SEMANTIC_INPUT	4
SEMANTIC_IN_TANGENT	5
SEMANTIC_INTERPOLATION	6
SEMANTIC_INV_BIND_MATRIX	7
SEMANTIC_JOINT	8
SEMANTIC_LINEAR STEPS	9
SEMANTIC_MORPH_TARGETMORPH_WEIGHT	10
SEMANTIC_NORMAL	11
SEMANTIC_OUTPUT	12
SEMANTIC_OUT_TANGENT	13
SEMANTIC_POSITION	14
SEMANTIC_TANGENT	15
SEMANTIC_TEXBINORMAL	17
SEMANTIC_TEXCOORD	17
SEMANTIC_TEXTANGENT	18
SEMANTIC_UV	19
SEMANTIC_VERTEX	20
SEMANTIC_WEIGHT	21
SEMANTIC_INVALID	22



# colladaRenderUtil::SimpleRenderer

## COLLADA simple renderer class

#### **Definition**

### **Internal Type**

Type	Description
SceneInfo	Scene information during rendering

#### **Members Functions**

Function	Description
initialize	Initialize SimpleRenderer
finalize	Finalize SimpleRenderer
loadCollada	Initialize Collada class
disposeCollada	Dispose of Collada
instantiateVisualScene	Create a scene instance
disposeVisualScene	Dispose of a scene instance
setTime	Set the scene time
render	Render InstanceVisualScene

## Description

This is a COLLADA simple renderer class. Use this with the following steps.

- (1) Create this class instance and initialize it with the initialize () function.
- (2) Use the loadCollada () function to load a COLLADA file.
- (3) Use instantiateVisualScene () to create a scene instance.
- (4) Use the setTime() function to set the time of the scene instance.
- (5) Use the render () function to render the scene instance.



# colladaRenderUtil::SimpleRenderer::SceneInfo

## Scene information during rendering

#### **Definition**

```
#include <colladarenderutil.h>
namespace colladaRenderUtil{
        class SimpleRenderer
            struct SceneInfo
               colladaRenderUtil::InstanceVisualScene *instanceVisualScene;
               Vectormath::Aos::Matrix4 localToWorld;
        };
```

#### **Members Variables**

Variable	Description	
instanceVisualScene	Pointer to colladaRenderUtil::InstanceVisualScene to be rendered	
localToWorld	Matrix for converting from InstanceVisualScene route coordinate system to world coordinate system	

# Description

This type is used with colladaRenderUtil: SimpleRenderer::render().

Specify InstanceVisualScene to be rendered and the matrix for converting from the scene route coordinate system to world coordinate system.



# colladaRenderUtil::SimpleRenderer::initialize

Initialize SimpleRenderer

#### **Definition**

#### **Arguments**

gpuHeapSize Size of memory area to be allocated within SimpleRenderer and be

accessible from GPU

uniformParamManager ShaderParameterManager used in a scene drawn with this

SimpleRenderer

verbose Flag that enables the redundant TTY output generated at the time of load

processing

patcher Shader patcher used when SimpleRenderer uses the shader. Shader

patcher is generated automatically and internally in the case that NULL is

specified.

#### **Return Values**

Returns SCE OK(0) as the value of the function for success.

Returns a negative value for errors.

#### **Description**

Initializes SimpleRenderer

# colladaRenderUtil::SimpleRenderer::finalize

Finalize SimpleRenderer

#### **Definition**

#include <colladarenderutil.h>
int colladaRenderUtil::SimpleRenderer::finalize(void)

### **Arguments**

None

#### **Return Values**

Returns SCE OK(0) as the value of the function for success.

Returns a negative value for errors.

### **Description**

Finalizes SimpleRenderer.

To perform proper finalization, this function must be executed after disposing of all instances of Collada and InstanceVisualScene created using SimpleRenderer.



# colladaRenderUtil::SimpleRenderer::loadCollada

Initialize Collada class

#### **Definition**

```
#include <colladarenderutil.h>
int colladaRenderUtil::SimpleRenderer::loadCollada(
        colladaRenderUtil::Collada &collada,
        const char* daePath
)
```

## **Arguments**

collada Instance of Collada class for storing collada data COLLADA file path

#### **Return Values**

Returns SCE OK (0) as the value of the function for success.

Returns a negative value for errors.

## **Description**

Loads the COLLADA file and stores the information in the Collada class.



# colladaRenderUtil::SimpleRenderer::disposeCollada

Dispose of Collada

#### **Definition**

```
#include <colladarenderutil.h>
int colladaRenderUtil::SimpleRenderer::disposeCollada(
        colladaRenderUtil::Collada &collada
)
```

### **Arguments**

collada Collada class instance

#### **Return Values**

Returns SCE OK(0) as the value of the function for success.

Returns a negative value for errors.

## **Description**

Disposes of all data within the Collada class instance.

To dispose of the data properly, all instances of InstanceVisualScene created with this Collada must be disposed of.



# colladaRenderUtil::SimpleRenderer::instantiateVisualScene

Create a scene instance

#### **Definition**

## **Arguments**

scene Instance of Instance Visual Scene class where created

scenes are stored

collada used to create a scene

instanceShaderParameterManager ShaderParameterManager used by this instance. This

precedes the ShaderParameterManager registered in

SimpleRenderer as for this instance

### **Return Values**

Returns SCE OK(0) as the value of the function for success.

Returns a negative value for errors.

#### **Description**

Creates an instance of a scene within collada and stores it in scene.

Keep the source Collada until this scene instance is disposed of.

# colladaRenderUtil::SimpleRenderer::setTime

Set the scene time

#### **Definition**

```
#include <colladarenderutil.h>
int colladaRenderUtil::SimpleRenderer::setTime(
        colladaRenderUtil::InstanceVisualScene &scene,
        float time
)
```

## **Arguments**

scene Scene instance time Animation time

#### **Return Values**

Returns SCE OK(0) as the value of the function for success.

Returns a negative value for errors.

### **Description**

Specify the animation time of a scene. Specifying the time reproduces the scene at that time.



# colladaRenderUtil::SimpleRenderer::disposeVisualS cene

Dispose of a scene instance

### **Definition**

### **Arguments**

scene Scene instance

### **Return Values**

Returns SCE OK(0) as the value of the function for success.

Returns a negative value for errors.

## **Description**

Disposes of the scene instance.



# colladaRenderUtil::SimpleRenderer::render

Render InstanceVisualScene

#### **Definition**

### **Arguments**

context Context of GXM

projectionMatrix Projection matrix

viewMatrix View conversion matrix

lightPosition Point light color (X=red, Y=green, Z=blue) value range is 0.0 - 1.0

scenes Array of scenes to be rendered numScenes Length of array of scenes

#### **Return Values**

Returns SCE OK(0) as the value of the function for success.

Returns a negative value for errors.

## **Description**

 $Renders \ {\tt InstanceVisualScene} \ specified \ with \ scenes.$ 





# colladaRenderUtil::Collada::getLibraryGeometries

# Get library geometries

#### **Definition**

#include <colladarenderutil.h>

colladaRenderUtil::collada::geometry::LibraryGeometries
colladaRenderUtil::Collada::getLibraryGeometries() const

## **Arguments**

None

#### **Return Values**

Pointer to the library geometry structure

## **Description**

Returns LibraryGeometries within collada.



# colladaRenderUtil::Collada::getLibraryImages

Get the library images

#### **Definition**

#include <colladarenderutil.h>

colladaRenderUtil::collada::geometry::LibraryImages
colladaRenderUtil::Collada::getLibraryImages() const

## **Arguments**

None

#### **Return Values**

Pointer to the library image structure

## **Description**

Returns Library Images within collada.





# colladaRenderUtil::collada::geometry::LibraryGeometries::getNumGeometries

Get the number of geometries

### **Definition**

#include <colladarenderutil.h>
const uint32 t

colladaRenderUtil::collada::geometry::LibraryGeometries::getNumGeometries()

### **Arguments**

None

### **Return Values**

The number of geometries

### **Description**

Returns the number of geometies within LibraryGeometries



# colladaRenderUtil::collada::geometry::LibraryGeom etries::getGeometry

Get geometry with index specification

### **Definition**

```
#include <colladarenderutil.h>
const Geometry*
colladaRenderUtil::collada::geometry::LibraryGeometries::getGeometry(
```

#### **Arguments**

index Index of geometry

### **Return Values**

Pointer to the Geometry structure

### **Description**

Returns the geometry of the *index* number in LibraryGeometries.



# colladaRenderUtil::collada::geometry::Geometry::g etMesh

Get mesh

### **Definition**

#include <colladarenderutil.h>

Mesh\*

colladaRenderUtil::collada::geometry::Geometry::getMesh()

**Arguments** 

None

**Return Values** 

Pointer to the Mesh structure

**Description** 

Returns the mesh within the geometry.





# colladaRenderUtil::collada::geometry::Mesh::getNumTriangless

Get the number of Triangles

### **Definition**

#include <colladarenderutil.h>
uint32\_t
colladaRenderUtil::collada::geometry::Mesh::getNumTriangless()

### **Arguments**

None

### **Return Values**

The number of Triangles

### **Description**

Returns the number of triangular arrays in the mesh.



# colladaRenderUtil::collada::geometry::Mesh::getTri angles

Get Triangles

### **Definition**

```
#include <colladarenderutil.h>
const Triangles*
colladaRenderUtil::collada::geometry::Mesh::getTriangles(
```

# **Arguments**

index Index of Triangles

#### **Return Values**

Triangular array

### **Description**

Returns the triangular array with an index specification.





# colladaRenderUtil::collada::geometry::Triangles::ge tIndices

Get vertex index array

### **Definition**

#include <colladarenderutil.h>
unsigned short\*
colladarenderUtil::collada::geometry::Triangles::getIndices()

**Arguments** 

None

**Return Values** 

Start pointer of the vertex index array

**Description** 

Returns the vertex index array.



# colladaRenderUtil::collada::geometry::Triangles::ge tNumIndices

Get the number of vertex indices

### **Definition**

#include <colladarenderutil.h>
unsigned short

colladaRenderUtil::collada::geometry::Triangles::getNumIndices()

### **Arguments**

None

### **Return Values**

Number of vertex indices

### **Description**

Returns the number of vertex indices.



# colladaRenderUtil::collada::geometry::Triangles::ge tNumVertexInput

Get the number of attributes included in the vertex data

#### **Definition**

#include <colladarenderutil.h>
int

int

colladaRenderUtil::collada::geometry::Triangles::getNumVertexInput()

### **Arguments**

None

### **Return Values**

Number of attributes included in the vertex data

### **Description**

Returns the number of attributes included in the vertex data



# colladaRenderUtil::collada::geometry::Triangles::ge **tVertexInput**

Get attribute information included in the vertex data

#### **Definition**

```
#include <colladarenderutil.h>
VertexInput
colladaRenderUtil::collada::geometry::Triangles::getVertexInput(
```

## **Arguments**

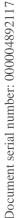
index Index of VertexInput

#### **Return Values**

Attribute information included in the vertex data

# **Description**

Returns the attribute information included in the vertex data



# colladaRenderUtil::collada::geometry::Triangles::ge tNumVertices

Get the number of vertices

### **Definition**

#include <colladarenderutil.h>
unsigned short

colladaRenderUtil::collada::geometry::Triangles::getNumVertices()

### **Arguments**

None

### **Return Values**

Number of vertices

### **Description**

Returns the number of vertices in the vertex data.



# colladaRenderUtil::collada::geometry::Triangles::ge tStream

Get vertex data array

### **Definition**

#include <colladarenderutil.h>
const float\*

colladaRenderUtil::collada::geometry::Triangles::getStream()

### **Arguments**

None

### **Return Values**

Start pointer of the vertex data array

### **Description**

Returns the vertex data array.



# colladaRenderUtil::collada::geometry::Triangles::ge tStride

Get stride of the vertex data

### **Definition**

#include <colladarenderutil.h>
unsigned int

colladaRenderUtil::collada::geometry::Triangles::getStride()

### **Arguments**

None

### **Return Values**

Stride of the vertex data

### **Description**

Returns the stride of the vertex data.



### colladaRenderUtil::collada::geometry::Triangles::ge tSemanticOffset

Get offset of semantic

#### **Definition**

#### **Arguments**

semantic Vertex semantic
indexset Index set of the semantic

#### **Return Values**

Offset of semantic

#### **Description**

Returns the offset in the vertex stream of the specified semantic.

For semantic, specify one of the following.

Macro	Value
SEMANTIC_BINORMAL	0
SEMANTIC_COLOR	1
SEMANTIC_CONTINUITY	2
SEMANTIC_IMAGE	3
SEMANTIC_INPUT	4
SEMANTIC_IN_TANGENT	5
SEMANTIC_INTERPOLATION	6
SEMANTIC_INV_BIND_MATRIX	7
SEMANTIC_JOINT	8
SEMANTIC_LINEAR_STEPS	9
SEMANTIC_MORPH_TARGETMORPH_WEIGHT	10
SEMANTIC_NORMAL	11
SEMANTIC_OUTPUT	12
SEMANTIC_OUT_TANGENT	13
SEMANTIC_POSITION	14
SEMANTIC_TANGENT	15
SEMANTIC_TEXBINORMAL	17
SEMANTIC_TEXCOORD	17
SEMANTIC_TEXTANGENT	18
SEMANTIC_UV	19
SEMANTIC_VERTEX	20
SEMANTIC_WEIGHT	21
SEMANTIC_INVALID	22



# colladaRenderUtil::collada::lmage::Librarylmages:: getNumlmages

Get the number of images

#### **Definition**

#include <colladarenderutil.h>
uint32\_t
colladaRenderUtil::collada::Image::LibraryImages::getNumImages()

**Arguments** 

None

**Return Values** 

The number of images

**Description** 

Returns the number of images.



# colladaRenderUtil::collada::lmage::Librarylmages:: getlmage

Get colladaRenderUtil::collada::Image

#### **Definition**

#### **Arguments**

Index Index

#### **Return Values**

Image

#### **Description**

Returns the image of the specified index.



## colladaRenderUtil::ShaderParameterManager::getP arameterId

Get the parameter ID

#### **Definition**

```
#include <colladarenderutil.h>
colladaRenderUtil::ShaderParameterManager::getParameterId(
        ShaderStage stage,
        const char *parameterName,
        uint32 t componentCount
)
```

#### **Arguments**

stage Shader stage parameterName Parameter name componentCount Number of elements

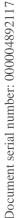
#### **Return Values**

Parameter ID

#### **Description**

Set the ID to the parameter specified with the argument, and return that ID. The returned ID is used with getSource().

If a negative value is returned, the default value is used for that parameter.



## colladaRenderUtil::ShaderParameterManager::getS ource

Get the value of the parameter

#### **Definition**

```
#include <colladarenderutil.h>
const float*
colladaRenderUtil::ShaderParameterManager::getSource(
        ShaderStage stage,
        int id
)
```

#### **Arguments**

stage Shader stage id Parameter ID

#### **Return Values**

Value of the parameter

#### **Description**

Return the value to be set to the parameter corresponding to the parameter ID.



## colladaRenderUtil::ShaderParameterManager::setNodeToSceneRootMatrix

Send notification of conversion matrix from node coordinate system to scene coordinate system

#### **Definition**

#### **Arguments**

nodeToSceneRootMatrix Conversion matrix from node coordinate system to scene coordinate system

#### **Return Values**

If the value is set successfully, SCE OK is returned. Otherwise, a negative value is returned.

#### **Description**

This is called when a renderer sends notification of a matrix for conversion from a node coordinate system to a scene coordinate system while rendering a node within a scene graph.

### colladaRenderUtil::ShaderParameterManager::setSc eneRootToWorldMatrix

Send notification of conversion matrix from scene coordinate system to world coordinate system

#### **Definition**

#### **Arguments**

sceneRootToWorldMatrix Conversion matrix from scene coordinate system to world coordinate system

#### **Return Values**

If the value is set successfully, SCE OK is returned. Otherwise, a negative value is returned.

#### **Description**

This is called when a renderer sends notification of a matrix for conversion from a scene coordinate system to a world coordinate system.

## colladaRenderUtil::ShaderParameterManager::setVi **ewProjectionMatrix**

Send notification of view matrix and projection matrix

#### **Definition**

```
#include <colladarenderutil.h>
colladaRenderUtil::ShaderParameterManager::setViewProjectionMatrix(
        sce::Vectormath::Simd::Aos::Matrix4 arg viewMatrix,
        sce::Vectormath::Simd::Aos::Matrix4 arg projMatrix
)
```

#### **Arguments**

```
viewMatrix View matrix
projMatrix Projection matrix
```

#### **Return Values**

If the value is set successfully, SCE OK is returned. Otherwise, a negative value is returned.

#### **Description**

This is called when a renderer sends notification of a view matrix and projection matrix.

## colladaRenderUtil::ShaderParameterManager::getTe xtureParameterId

Get the texture parameter ID

#### **Definition**

#### **Arguments**

stage Shader stage parameterName Parameter name

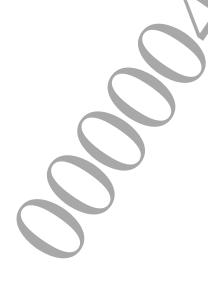
#### **Return Values**

Texture ID

#### **Description**

Set the ID to the texture parameter specified with the argument, and return that ID. The returned ID is used with getSourceTexture().

If a negative value is returned, the default value is used for that parameter.



## colladaRenderUtil::ShaderParameterManager::getS ourceTexture

Get the texture

#### **Definition**

```
#include <colladarenderutil.h>
const SceGxmTexture*
colladaRenderUtil::ShaderParameterManager::getSourceTexture(
        ShaderStage stage,
        int id
)
```

#### **Arguments**

Shader stage stage Texture parameter ID

#### **Return Values**

Texture

#### **Description**

Return the value to be set to the texture corresponding to the texture parameter ID.

