

Autodesk® FBX® SDK

Functionality basics



Video

<ftp://sparks:Sparks2012@87.106.97.50/>

or directly:

IP address: 87.106.97.50

- User name: *sparks*
- Password: *Sparks2012*
- *FBX_SDK_webcast* folder contains the video and handouts from yesterday



Exercise – Unit 1

- Install and configure SDK
- Make sure your compiler is working with the samples
 - Build the samples
- Review the online documentation



FBX SDK Webcast Agenda

- Day / Hour 1 – Welcome to FBX SDK
- **Day / Hour 2 – FBX Basics (Scenes/Import/Export)**
- Day / Hour 3 – SDK Object Model
- Day / Hour 4 – Geometry
- Day / Hour 5 – Animation



SDK functionality basics

- Object Lifetime
- Scene and node basics
- Import / Export



Object Lifetime



Object Lifetime

- Object lifetime refers to memory management
- Use FbxManager
 - The [FbxManager](#) class is responsible for creating, managing, and destroying FBX SDK objects
 - Only one instance is needed
 - Typically the first thing you do



Object Lifetime

- Creating Objects with the FBX Manager
 - Use the object's Create and Destroy methods
 - [FbxManager](#) instance is passed to Create
 - Call Destroy to properly free the object
 - [FbxManager](#) will automatically free the memory allocated for that object
 - updates all the internal connections



Object Lifetime

- Typically Scene elements are created within the scene, that was created by [FbxManager](#)
 - In this case the scene will control the lifetime management
- It is also possible to create elements using the [FbxManager](#) directly
 - In this case the [FbxManager](#) controls them directly



Scene and Node basics



What makes up a scene?

- Nodes
- The Nodes have “attributes”
 - Geometry
 - Deformations
 - Animation
 - Constraints
 - Material / Shaders
 - Camera / Lights



Creating a scene

- Invoke `FbxScene::Create(pManager, “name”);`
- contains
 - “root” node (`FbxNode`)
 - global settings (`FbxGlobalSettings`)
 - The scene's axis system, system units, ambient lighting, and time settings



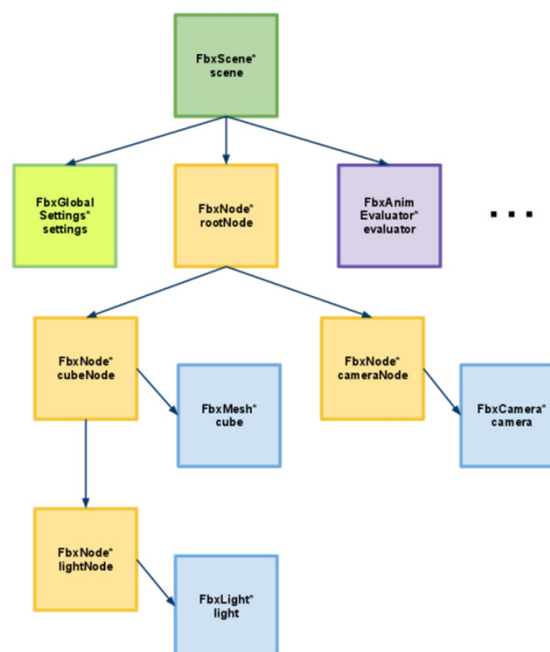
Other Scene Properties

- Animation Evaluation
 - scene's [FbxAnimEvaluator](#) is accessed via [FbxScene::GetEvaluator\(\)](#)
- Texture and Material Management
 - accessed and modified using member functions such as [FbxScene::GetMaterial\(\)](#) and [FbxScene::GetTexture\(\)](#)
- Characters and Character Pose Management
 - accessed using [FbxScene::GetCharacter\(\)](#) and [FbxScene::GetCharacterPose\(\)](#)

Autodesk
FBX

FbxScene

- Nodes live within a scene
 - Scene provides a Parent/Child hierarchy
- The scene graph is abstracted by the [FbxScene](#) class
- A scene element is defined by combining a [FbxNode](#) with a subclass of [FbxNodeAttribute](#)

Autodesk
FBX

Other Scene functionality

- [FbxScene](#) is derived from [FbxDocument](#)
- Both contain lots of helpful “global” things

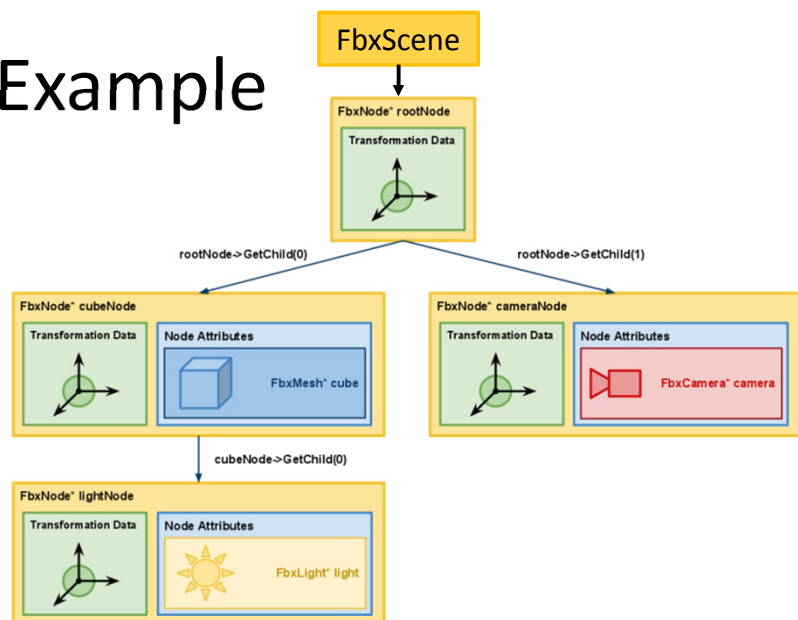
Autodesk
FBX

demo



Scene/Node Example

- Nodes live within a scene
- Nodes contain transform data



Autodesk
FBX



FbxNode

- Provides transformation
 - Translation, Rotation, and Scaling Vectors are properties
 - LclTranslation
 - LclRotation
 - LclScaling
 - Global and Local Transformation Matrices
 - EvaluateGlobalTransform
 - EvaluateLocalTransform
- Transformation Matrices
 - Note: Can be handled differently per application



FbxNode – Accessing from Scene

```
// Get the root node of the scene.
FbxNode* currentNode = lScene->GetRootNode();

// Recurse on the node children
int numKids = currentNode->GetChildCount();
for (int k = 0; k < numKids; k++)
{
    FbxNode *childNode = currentNode->GetChild(k);
    traverseScene(childNode, level+1);
}
```



FbxNode – Adding to Scene

```
// Get the root node of the scene.  
FbxNode* lRootNode = lScene->GetRootNode();  
  
// Create a child node.  
FbxNode* lNode = FbxNode::Create(lScene, "child");  
  
// Add the child to the root node.  
lRootNode->AddChild(lNode);
```



Other Node functionality

- [FbxNode](#) is derived from [FbxObject](#) and other base classes



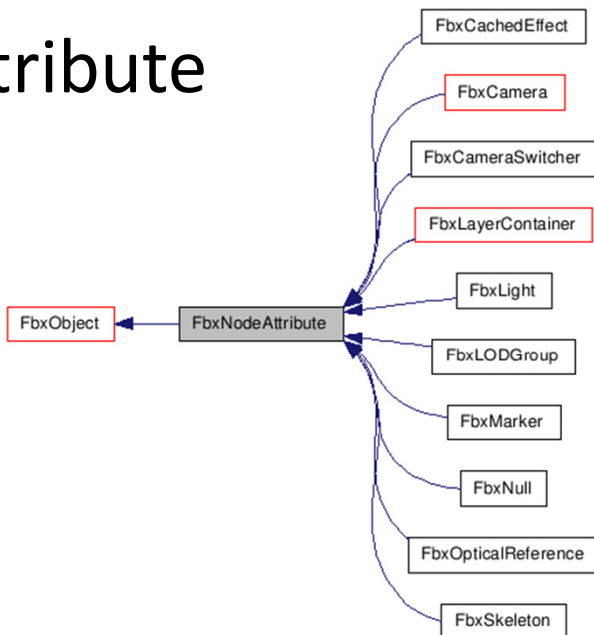
FbxNodeAttribute

- Node Attributes
 - Content of a node
 - A mesh, a light, a camera, or any other object present in a scene is abstracted by a subclass of [FbxNodeAttribute](#)



FbxNodeAttribute

- Hierarchy



FbxLayerContainer

- Contains a collection of [FbxLayer](#) objects
- [FbxLayer](#) class provides a base for the layering mechanism.

Normals
Binormals
Tangents
Materials
Polygon
Groups
UVs
Vertex

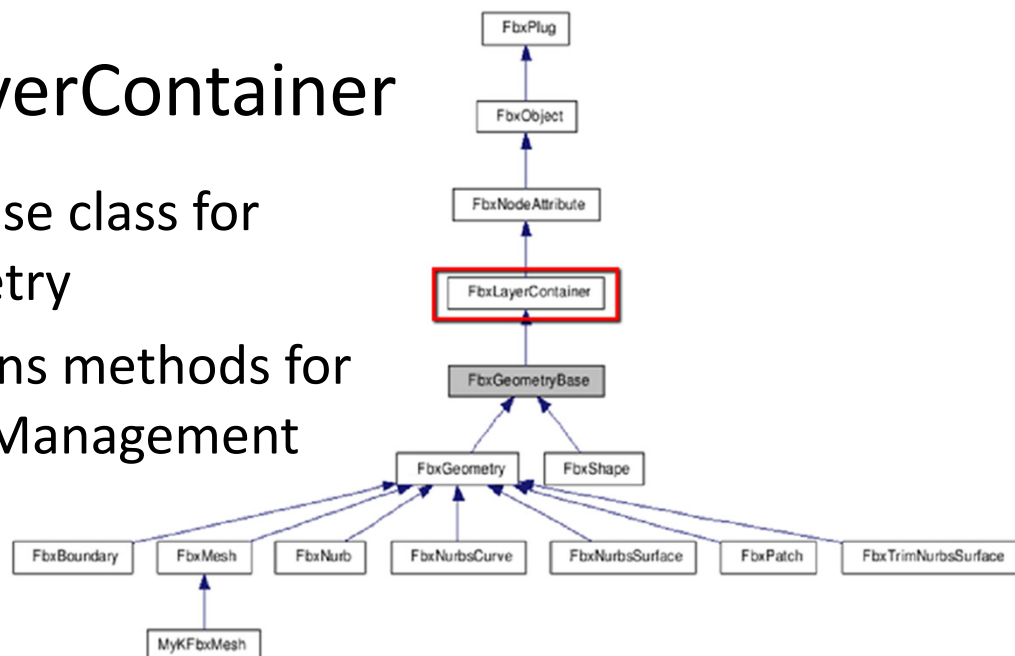
Colors
Smoothing
information
Vertex Creases
Edge Creases
Custom User Data
Visibilities

Autodesk
FBX



FbxLayerContainer

- The base class for geometry
- Contains methods for Layer Management



Autodesk
FBX



Summary

- The FbxManager is needed to enter the SDK
 - It helps manage object lifetime (handles memory)
 - Use it to access the scene
- Scene contains nodes
 - Add / Access nodes
- Nodes contain Node Attributes
 - FbxNodeAttribute is base for content
 - FbxLayerContainer more specific for Geometry



Import and Export



Scene Import and Export

- Primary function of the FBX SDK
- Scene import and export classes are the bridge between scene description classes and the .fbx file format
- Class [FbxIOBase](#) is the base class of FbxImporter and FbxExporter.



Import / Export - I/O Settings

- I/O Settings
 - The [FbxIOSettings](#) class is responsible for specifying the elements that would be imported/exported
 - Configures embedded media behavior
 - Password protection
 - Custom I/O



FbxIOSettings - Creation

- By default, settings are “true”; supports everything

```
#include <fbxsdk.h>
#include <fbxfilesdk/fbxio/fbxiosettings.h>

// Create the FBX SDK manager
FbxManager* pManager = FbxManager::Create();

// Create an IOSettings object.
// IOSROOT is defined in FbxioSettingspath.h.
FbxIOSettings * ios = FbxIOSettings::Create(pManager, IOSROOT );
pManager->SetIOSettings(ios);
```

Autodesk
FBX



FbxIOSettings – Changing defaults

```
// Import options determine what kind of data is to be imported.
(*(lSdkManager->GetIOSettings())).SetBoolProp(IMP_FBX_MATERIAL, true);
(*(lSdkManager->GetIOSettings())).SetBoolProp(IMP_FBX_TEXTURE, true);
(*(lSdkManager->GetIOSettings())).SetBoolProp(IMP_FBX_LINK, false);
(*(lSdkManager->GetIOSettings())).SetBoolProp(IMP_FBX_SHAPE, false);
(*(lSdkManager->GetIOSettings())).SetBoolProp(IMP_FBX_GOBO, false);
(*(lSdkManager->GetIOSettings())).SetBoolProp(IMP_FBX_ANIMATION, true);
(*(lSdkManager->GetIOSettings())).SetBoolProp(IMP_FBX_GLOBAL_SETTINGS, true);

// Set the export states.
(*(lSdkManager->GetIOSettings())).SetBoolProp(EXP_FBX_MATERIAL, true);
(*(lSdkManager->GetIOSettings())).SetBoolProp(EXP_FBX_TEXTURE, true);
(*(lSdkManager->GetIOSettings())).SetBoolProp(EXP_FBX_EMBEDDED, true);
(*(lSdkManager->GetIOSettings())).SetBoolProp(EXP_FBX_SHAPE, false);
(*(lSdkManager->GetIOSettings())).SetBoolProp(EXP_FBX_GOBO, false);
(*(lSdkManager->GetIOSettings())).SetBoolProp(EXP_FBX_ANIMATION, true);
(*(lSdkManager->GetIOSettings())).SetBoolProp(EXP_FBX_GLOBAL_SETTINGS, true);
```

Autodesk
FBX



Importing a Scene (read Fbx)

- abstracted by the [FbxImporter](#) class
- must have its [FbxImporter::Initialize\(\)](#) method called with three parameters:
 - The path and filename of the file containing the scene to import.
 - The numeric file format identifier (-1)
 - The [FbxIOSettings](#) object containing the import configuration options



Exporting Scene (write Fbx)

- abstracted by the [FbxExporter](#) class
- must have its [FbxExporter::Initialize\(\)](#) method called with three parameters:
 - The path and filename of the file containing the scene to import.
 - The numeric file format identifier (-1)
 - The [FbxIOSettings](#) object containing the import configuration options



Things within the Import/Exporter

Autodesk®
FBX



Some Helper Utilities

- [FbxString](#)
 - Utility class to manipulate strings
- [FbxFileUtils](#)
 - File system utilities
- [FbxArray< Type >](#)
 - For example, use `Scene::FillAnimStackNameArray` to get the animation stack names as an array of strings
- [FbxIOPluginRegistry](#)
 - Accessed by `FbxManager::GetIOPluginRegistry()`

Autodesk®
FBX

demo



Exercise

- Create an importer and list the contents of an FBX file by traversing the scene nodes
 - Follow the Directions in the Exercise 2 document

