FBX Extensions SDK

FBX Extensions SDK Programmer's Guide 2011

April 2010

Autodesk® FBX® 2011 SDK

Contents

Welcome to FBX Extensions SDK

1

Welcome to the documentation for Autodesk FBX Extensions SDK. This document contains:

- Introduction on page 3.
  Introduces you to FBX Extensions SDK.
- Installing and Configuring on page 7
  Explains how to install and configure FBX Extensions SDK on Linux, Windows, and Mac OS computers.
- Extensions to file formats on page 11.
  Explains how to extend FBX and Collada file formats, and how to extend FBX SDK to support your own custom file format.
- Extensions to FBX Plug-ins on page 13.
  Explains how to extend FBX Plug-in for Autodesk 3ds Max and FBX Plug-in for Autodesk Maya.

NOTE In this document, we refer to the two plug-ins collectively as “FBX Plug-in for 3ds Max/Maya”.

For the most recent information and downloads, vist *http://www.autodesk.com/fbx*

Introduction 2

FBX Extensions SDK allows you to create three kinds of extensions to Autodesk FBX technology:

- Extensions to FBX Plug-in for 3ds Max/Maya.
- Extensions to two of the file formats supported by FBX: FBX file format (versions 5 and 6), and Collada file format.
- Extensions to FBX SDK to support your own custom file formats.

NOTE FBX Extensions SDK contains the source code for the modules used by FBX SDK to read and write files in two file formats: FBX and Collada.

What you must know

This document assumes that you are a developer who is familiar with FBX technology, particularly Autodesk FBX SDK.

For information about FBX SDK, see *FBX SDK Help*, which is distributed with both FBX Extensions SDK and FBX SDK.

What is FBX Extensions SDK

FBX Extensions SDK is a special version of FBX SDK aimed at developers who need to extend FBX technology in various ways.

As a developer, you can use FBX Extensions SDK to:

- Extend the functionality of the FBX plug-ins for 3ds Max/Maya in various ways, by coding extensions to the FBX plug-ins for 3ds Max/Maya. For example, you can create an extension that allows the FBX plug-in to read, process, and write custom data that you have added to a 3ds Max/Maya scene.
- Create a new file format by extending the FBX and/or Collada file format to support additional kinds of data.
- Extend FBX SDK to support a custom file format of your own design.
- Extend the functionality of the FBX plug-ins for 3ds Max/Maya to support the reading and writing of files that use your extended/custom file format.

- Develop applications that read and write files in your extended/custom file format, without passing through 3ds Max or Maya.

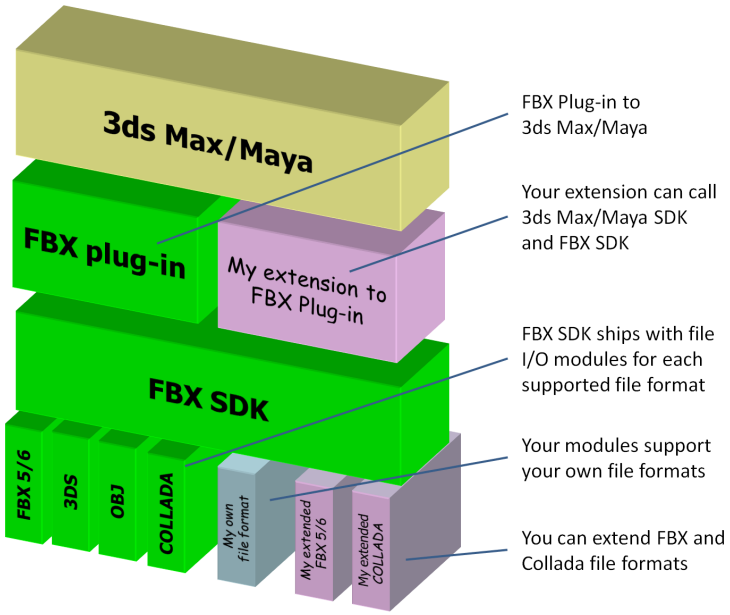
FBX Extensions SDK contains:

- The actual source code for the file I/O plug-ins that FBX SDK uses to read and write scenes to FBX and Collada files. You can extend the FBX or Collada file format by modifying these I/O plugins.
- Source code for a sample file I/O plug-in that reads and writes a custom file format.
- The source code for sample extensions to the FBX plug-ins for 3ds Max/Maya. You can use this code to help you write your own extensisons to the FBX plug-ins.
- Library files (Windows only) that link to the FBX plugins for 3ds Max/Maya.
- Header files, documentation, and sample programs that support the creation of extensions.

Where FBX extensions fits in

The diagram shows where FBX extensions fit in to FBX:

- We provide the source code for the modules that do that actual file I/O for FBX files and Collada files.
- We provide source code for a file I/O module that supports a trivial file format.
- We provide projects that show you how to link your file I/O modules to FBX SDK.
- We provide libraries and sample code to help you create extensions to FBX plug-ins.



FBX Extensions SDK architecture

FBX plug-in for 3ds Max/Maya

FBX plug-in for 3ds Max/Maya is our brief way of referring to FBX Plug-in for Autodesk 3ds Max, FBX Plug-in for Autodesk Maya, or both.

These two plug-ins are very similar. Both have a graphical user interface which allow users of 3ds Max/Maya to import and export scene data stored in FBX files, Collada files, 3DS files, etc. FBX Extensions SDK gives you the tools to create your own *extension to the FBX plug-in for 3ds Max/Maya*.

These plug-ins are shipped with 3ds Max/Maya. You can install the most recent releases of these plug-ins from *www.autodesk.com/fbx*. The plug-ins include full documentation for 3ds Max/Maya users.

File I/O modules

File I/O modules are components of FBX SDK. They do the work of converting, one scene object at a time, the data in an FBX scene to and from a particular file format. File I/O modules do not have a graphical user interface.

FBX SDK ships with several of these modules: each one reads and writes scene data from and to one particular file format. You can create your own file I/O module for your own file format.

FBX Extensions SDK includes the source code for the file I/O plug-ins that support FBX files (versions 5 and 6) and Collada files. You can modify this source code to create your own *extensions to file formats*, i.e., extensions to the FBX file format and/or the Collada file format.

You can also create your own file I/O plug-ins to support your own custom file formats. Custom file formats are *extensions to FBX SDK*, even when they are not extensions of an existing file format.

Installing and Configuring 3

Introduction

There are separate distributions of FBX Extensions SDK for Linux, Mac OS, and Windows. The structure of the FBX SDK distribution directory is, however, identical on all these platforms.

To download and install FBX Extensions SDK, follow the instructions for your development platform.

To obtain distributions for your development platform of FBX SDK, FBX Extensions SDK, FBX Plug-in for 3ds Max, and FBX Plug-in for Maya, visit *http://area.autodesk.com*.

Structure of FBX Extensions SDK

The structure of the FBX Extensions SDK distribution is very similar to the FBX SDK distribution, and contains many of the same folders. Here are the key differences:

- Instead of an `examples` directory, there is a `plugins` directory. This directory contains:
  - The actual source code and supporting files for the file I/O module that FBX SDK uses to support FBX files.
  - The actual source code and supporting files for the file I/O module that FBX SDK uses to support Collada files.
  - Source code and supporting files for MyOwnWriterReader, a sample file I/O module that supports CustomWriter files (a trivial file format used for demonstration purposes).
- For Windows: the `lib` folder contains small *.lib* files that link to the appropriate dynamically loaded library file of FBX Plug-in for 3ds Max/Maya (which contain the executable code for FBX SDK classes). Note: if you do not intend to use either of these plug-ins, link to the *.lib* files distributed with FBX SDK.
- For Maya on Mac OS or Linux: no library files are required to link with FBX Plug-in for Maya. Accordingly, there is no `lib` directory in distributions of 3ds Extensions SDK for Mac OS and Linux. Note: if you do not intend to use FBX Plug-in for Maya, link to the *.lib* files distributed with FBX SDK
- Note: 3ds Max does not run on Mac OS or Linux.
- The `doc` directory contains *FBX Extensions SDK Programmer's Guide* (the document you are now reading), *FBX SDK Programmer's Guide*, and *FBX SDK Help*.

Development environments

FBX Extensions SDK uses the same development environments as FBX SDK: see *FBX SDK Help* for more information.

Runtime environment

To run any software that you develop with FBX Extensions SDK 2010.2, you need one of the following installed:

Any FBX Plug-in for Maya, provided the version number of FBX Plug-in is 2010.2.	Your code will link to the library file for the Maya plug-in (in Windows, a DLL). You do not need to install FBX SDK.
Any FBX Plug-in for 3ds Max, provided the version number of FBX Plug-in is 2010.2.	Your code will link to the library file for the 3ds Max plug-in (in Windows, a DLL). You do not need to install FBX SDK.
Any other application that you develop with FBX Extensions SDK 2010.2.	Your code will link directly to the library file for FBX SDK. See *FBX SDK Help*.

You can, however, **install** FBX Extensions SDK without having any of the above installed.

To allow Maya users to use your extension to FBX Plug-in or your file I/O plug-in:

- Place your *.dll* file in: `MAYA_INSTALL_FOLDER/bin/plug-ins/fbx/`

To allow 3ds Max users to use your extension to FBX Plug-in or your file I/O plug-in:

- Place your *.dll* file in `MAX_INSTALL_FOLDER/stdplugs/fbx/`

To allow your own applications to use your file I/O plug-in:

- Place your*.dll* file in the location required by your application.
  Your applications can specify the location where it expects to find your scene I/O plug-in in two ways:

  ```
  FbxSdkManager->LoadPluginsDirectory(path, extension);
  ```

  Or:

  ```
  FbxSdkManager->LoadPlugin(filepath);
  ```

  See class KfbxSDKManager in *FBX SDK Help*.

NOTE The `plugins` directory contains sample extensions to FBX Plug-in for 3ds Max and FBX Plug-in for Maya. You cannot run these extensions in Debug mode, because the FBX Plug-ins are built in Release mode.

Downloading and installing

You can install FBX Extensions SDK in any location on your computer.

Windows

To download and install FBX Extensions SDK on your Windows computer:

1. Go to http://www.autodesk.com/fbx.

2 Navigate to the Downloads page and follow any instructions.

3 Find the Windows distribution file for FBX Extensions SDK 2010.2.

4 Download the distribution file to your computer. The distribution file is a Setup program, i.e., an executable.

5 Run the Setup program and follow the instructions.

6 The Setup program will let you specify a destination folder. Specify a folder which is new or empty.

7 Read `readme.txt`, located in the destination folder.

Notes:

- The Setup program does not modify the Windows Registry or the Windows Start menu.
- You can have more than one version of FBX Extensions SDK installed on your computer, providing you install each version in a separate folder.

To remove FBX Extensions SDK from your computer:

- Run `uninstall.exe`, located in the FBX Extensions SDK distribution folder.

Mac OS X

The procedure is the same as for FBX SDK for Mac OS X.

Linux

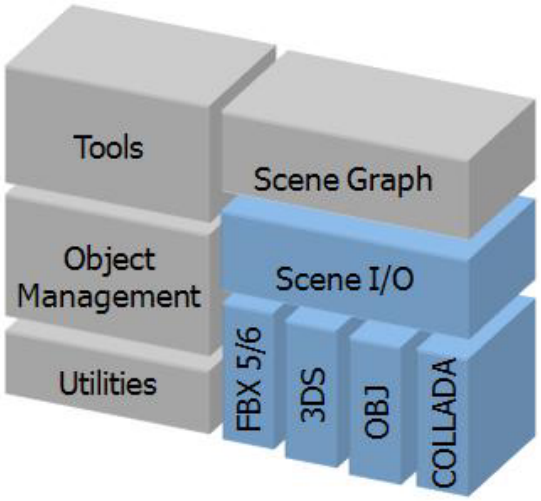
The procedure is the same as for FBX SDK for Linux.

Extensions to file formats

4

The part of FBX SDK that deals with importing scene data and exporting scene data is called *scene I/O* or *file I/O*.

FBX SDK imports from and exports to several file formats, namely FBX, COLLADA, OBJ, 3DS, and DXF. Two versions of the FBX file format are supported: Version 5 and Version 6.



A *file I/O module* is a set of classes that read scene data from a file, and write scene data to a file, using a particular file format. Each format supported by FBX SDK has its own *writer* class (subclassed from class KFbxWriter) and *reader* class (subclassed from class KFbxReader). FBX Extensions SDK includes source code for FBX file I/O module and the Collada file I/O module.

FBX SDK implements its file I/O using a plug-in architecture. This makes it relatively easy to extend FBX SDK with your own reader and writer classes so that it supports:

- Additional *custom file formats* of your own design.
- Extensions of the FBX and Collada file formats.

Accordingly, the projects for FBX, Collada, and MyOwnReaderWriter all include code to register the file I/O modules with FBX SDK.

NOTE The file I/O modules are registered as plug-ins to FBX SDK. Do not confuse them with FBX Plug-in for 3ds Max/Maya

---

Creating a custom file format

Sample program MyOwnReaderWriter (in the `plugin` directory) shows how to use FBX SDK to write and read "CustomWriter" files, which use file extension *.abc*. In general, to support your own custom file format, you must:

- Implement your own writer class (see `MyOwnWriter.cpp` and `MyOwnWriter.h`).
- Implement your own reader class (see `MyOwnReader.cpp` and `MyOwnReader.h`).

You must also use class `KFbxIOPluginRegistry` (see `MyOwnWriterReaderPlugin.cpp` and `MyOwnWriterReaderPlugin.h`) to:

- Register your reader and writer as a file I/O plug-in.
- Register your own file extension (which need not be unique).
- Register your own file description (which must be unique).

NOTE Do not confuse file I/O plugins with the plug-ins to 3ds Max and Maya. See "Plug-ins and Extensions".

The `MyOwnWriterReader` folder contains the source code and supporting files for a file I/O plug-in that imports and exports .ABC files, a trvial custom file format.

See also Extending FBX or Collada file formats on page 12.

Extending FBX or Collada file formats

FBX Extensions SDK allows you to extend FBX file format (versions 5 and 6) and the Collada file format. You can:

- Extend the FBX and/or Collada file format to support additional kinds of data.
- Extend the functionality of the FBX plug-ins for 3ds Max/Maya to support the reading and writing of files that use your extended file format.
- Use FBX SDK directly to read and write files that use your extended file format, without passing through 3ds Max or Maya.

The `plugins` directory contains:

- Source code and project files for the I/O plug-ins that read and write scenes to FBX files.
- Source code and project files for the I/O plug-ins that read and write scenes to Collada files.

See also Creating a custom file format on page 12.

Extensions to FBX Plug-ins

5

Users of 3ds Max can export all or part of a 3ds Max scene to an FBX file. Simililarly, they can import scene data from another application into their 3ds Max scene (providing that the scene data has been stored in a file format supported by FBX).

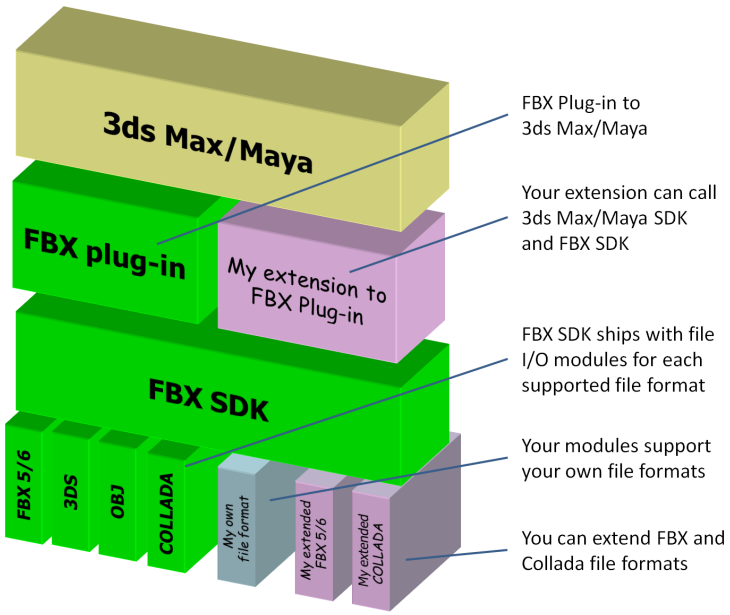
They do this from within the 3ds Max user interface, by using the FBX Export and FBX Import dialog boxes. These two dialog boxes are the user interface to FBX Plug-in for 3ds Max.

Similarly, Maya users use FBX Plug-in for Maya to import and export files containing scene data.

3ds Max/Maya users can import and export FBX files, Collada files, and files in other file formats. For details, see *FBX SDK Help*.

If, however, you have created *custom object types* for your 3ds Max/Maya scenes, your users may be creating scenes that they cannot completely import and export using FBX Plug-in.

The solution may be to create your own extensions to FBX Plug-in for 3ds Max/Maya.



Your extension is called by FBX Plug-in at key points in the import/export process. In turn, your extension can call functions in the 3ds Max/Maya SDK as well as the FBX SDK.

As the FBX Plug-in traverses the 3ds Max/Maya scene, it calls your extension to FBX Plug-in at key points, passing it the appropriate scene data. Your extension can call functions in 3ds Max/Maya SDK and in FBX SDK to process the scene and its custom data.

Sample extensions and templates

The `plugins` directory contains Visual Studio projects for:

- Fully functional sample extensions for 3ds Max and Maya.
- Template programs for 3ds Max and Maya that contain skeleton code and detailed comments on the function calls that you can implement.

Directory	Description
`maxcustomdata`	Fully functional sample extension for 3ds Max
`maxextensiontemplate`	Skeleton code and comments for 3ds Max
`mayacustomdata`	Fully functional sample extension for Maya
`mayaextensiontemplate`	Skeleton code and comments for Maya

The header files in the Visual Studio projects contain declarations for the functions that your extension can implement.

Summary of functions

Maya

Your extension to FBX Plug-in for Maya can implement the following functions:

```
EXPORT_DLL bool MayaExt_IsExtension(){return true;}

EXPORT_DLL bool MayaExt_ExportHandled(MObject& pMayaObject);
EXPORT_DLL void MayaExt_ExportBegin(KFbxScene* pFbxScene);
EXPORT_DLL void MayaExt_ExportTranslated(KFbxObject* pFbxObject, MObject& pMayaObject);
EXPORT_DLL void MayaExt_ExportEnd(KFbxScene* pFbxScene);

EXPORT_DLL bool MayaExt_ImportHandled(KFbxObject* pFbxObject);
EXPORT_DLL void MayaExt_ImportBegin(KFbxScene* pFbxScene);
EXPORT_DLL void MayaExt_ImportTranslated(KFbxObject* pFbxObject, MObject& pMayaObject);
EXPORT_DLL void MayaExt_ImportEnd(KFbxScene* pFbxScene);
```

3ds Max

Your extension to FBX Plug-in for 3ds Max can implement the following functions:

```
EXPORT_DLL bool MaxExt_IsExtension(){return true;}

EXPORT_DLL bool MaxExt_ExportHandled(INode* pMaxObject);
EXPORT_DLL void MaxExt_ExportBegin(KFbxScene* pFbxScene, INode* pMaxRootNode);
EXPORT_DLL void MaxExt_ExportTranslated(KFbxObject* pFbxObject, INode* pMaxObject);
EXPORT_DLL void MaxExt_ExportEnd(KFbxScene* pFbxScene, INode* pMaxRootNode);

EXPORT_DLL bool MaxExt_ImportHandled(KFbxObject* pFbxObject);
EXPORT_DLL void MaxExt_ImportBegin(KFbxScene* pFbxScene, INode* pMaxRootNode);
EXPORT_DLL void MaxExt_ImportTranslated(KFbxObject* pFbxObject, INode* pMaxObject);
EXPORT_DLL void MaxExt_ImportEnd(KFbxScene* pFbxScene, INode* pMaxRootNode);
```

Export functions

Below are the functions called by the FBX Plug-ins for 3ds Max/Maya while it is exporting data from a 3ds Max/Maya scene. All these functions are called before the file I/O to the user-selected file format begins.

Here is pseudocode that shows how FBX Plug-in calls these functions:

```
Before translating any scene objects to FBX:
    Call ExportBegin();

For each object in the 3ds Max/Maya scene:
    Call ExportHandled()
    If ExportHandled() returns true,
        Translate the object into an FBX object
    Else,
        Do nothing (i.e., the extension has translated the object);

For each object translated by FBX Plug-in:
    Call ExportTranslated(), passing the object and its translation;

The translation of the scene is now complete:
    Call ExportEnd()
```

ExportBegin()

```
EXPORT_DLL void MayaExt_ExportBegin(KFbxScene* pFbxScene);
EXPORT_DLL void MaxExt_ExportBegin(KFbxScene* pFbxScene, INode* pMaxRootNode);
```

FBX Plug-in calls this function before it translates any data from the 3ds Max/Maya scene into the corresponding FBX scene. FBX Plug-in passes a pointer to the empty FBX scene that it has created. The extension can access the 3ds Max/Maya scene by calling functions in the 3ds/Maya SDK.

Your extension can create your custom data now, or wait at the end of the export depending if that custom data replaces objects in the FBX scene hierarchy.

ExportHandled()

```
EXPORT_DLL bool MayaExt_ExportHandled(MObject& pMayaObject);
EXPORT_DLL bool MaxExt_ExportHandled(INode* pMaxObject);
```

Called for each 3ds Max/Maya object in the 3ds Max/Maya scene. Called *before* that object is translated into an FBX object. May be called more than once for the same object.

Your extension must return:

- true, if the extension is responsible for creating the corresponding FBX object for this 3ds Max/Maya object: FBX Plug-in will not translate the object.
- false: FBX Plug-in will translate the object (providing that it knows how to translate objects of that type).

ExportTranslated()

```
EXPORT_DLL void MayaExt_ExportTranslated(KFbxObject* pFbxObject, MObject& pMayaObject);
EXPORT_DLL void MaxExt_ExportTranslated(KFbxObject* pFbxObject, INode* pMaxObject);
```

After all objects in the 3ds Max/Maya scene have been translated into FBX objects, FBX Plug-in calls this function once for each 3ds Max/Maya object that has been translated.

The purpose of this function is to tell the extension: for this 3ds Max/Maya object, this is the corresponding FBX object.

ExportEnd()

```
EXPORT_DLL void MayaExt_ExportEnd(KFbxScene* pFbxScene);
EXPORT_DLL void MaxExt_ExportEnd(KFbxScene* pFbxScene, INode* pMaxRootNode);
```

FBX Plug-in calls this function after it has completely translated the scene into FBX (and before the file I/O begins). This is the last chance for your extension to modify the FBX scene.

Index

C