Interoperability and FBX Technology
White Paper

November 2007

Autodesk®

Overview

One of the most widely used 3D interchange formats on the market, Autodesk® FBX® technology is designed to enable workflows built around multiple 2D and 3D digital content creation applications. Such workflows are the lifeblood of large production houses and small boutique studios. In this white paper, you learn how FBX technology can help improve workflow efficiency and productivity and how to use it to enable interoperability among Autodesk® 3ds Max®, Autodesk® Maya®, Autodesk® MotionBuilder™ software, and other applications.

Being able to move data between applications in a production pipeline that employs multiple tools is crucial to the success of any digital content creation workflow, especially for companies using 3D tools to produce films and develop games. For example, such companies often need to access existing content created with earlier versions of the software because their pipelines are standardized on that product. To attract top-notch talent some boutique studios offer artists their choice of tools. When a film house needs to collaborate with a game studio, data interoperability enables team members to share assets built with one toolset and repurpose them with another. It also enables production facilities to have artists modeling with one tool and animators working with a different tool that is better suited to the task. With 3D interchange technology like FBX, a company that prefers to model and animate characters using one toolset is able to outsource architectural and virtual environments to a company that uses a different toolset.

Managing data in production pipelines is becoming more complex. FBX interoperability technology helps reduce the amount of time spent on human, technological, and economic resources repairing or rebuilding assets. The preceding examples show that having both a single file format and a robust data conversion technology that supports a plethora of data types is essential to streamlining a facility's workflow.

Autodesk FBX technology offers a solution to this complex interchange challenge. FBX technology makes it easier to transfer all manner of 3D data types across applications, including market-leading 3ds Max, Maya, MotionBuilder, Autodesk® Flame®, Autodesk® Flint®, and Autodesk® Smoke® software tools, as well as countless proprietary and third-party 2D and 3D tools and plug-ins. Most major 3D data elements, including motion, cameras, characters, non-polygonal surfaces, and skeletal hierarchies, as well as 2D, audio, and video media elements are supported by FBX technology. In addition, FBX technology supports all major NURBS and polygon surface types, keyframe and motion-capture animation, shapes and morph targets, materials and textures, lights, cameras, hierarchical information, and character animation data, including inverse kinematics, envelopes, and deformations.

Autodesk includes FBX plug-ins with both 3ds Max and Maya. MotionBuilder is built to support the FBX file format natively. For application and content vendors, Autodesk offers a C++ software development kit (SDK) and application programming interface (API) for accessing and converting 3D elements to the FBX file format. The SDK is free and runs under Windows®, IRIX®, Mac OS® X, and Linux® operating systems. There is a free utility for converting OBJ, DXF™, 3DS, COLLADA, and other file formats to the FBX file format. And an FBX viewer is available as a plug-in to the QuickTime® application.

The FBX plug-in and SDK deliver unprecedented levels of interoperability. But to take full advantage of this power, you need to be aware of the differences among your 3D applications and to understand how FBX technology helps you solve seemingly insurmountable challenges.

The following resources are available for download, free of charge, on: www.autodesk.com/FBX

FBX plug-ins for both 3ds Max and Maya. These plug-ins help you achieve high levels of interoperability between these applications, as well as MotionBuilder.

A C++ software development kit and application programming interface tools. The FBX SDK and API enable content and application developers to access and convert 3D elements to the FBX file format. The SDK is available on Windows, Linux, Mac OS X, and Linux operating systems.

A free utility for converting OBJ, DXF, 3DS, COLLADA, and other file formats to the FBX format.

An FBX viewer as a plug-in to QuickTime. The plug-in, installed alongside QuickTime, enables you to view FBX files without needing 3ds Max, Maya, MotionBuilder, or another application installed.

Understanding FBX

FBX is much more than a file format. The FBX technology family of tools—SDK, plug-ins, and viewer, as well as the FBX file format—offers a comprehensive data interchange solution. In the same way that PDF files let you share documents with users who do not own the word processor or page layout program those documents were created with, FBX technology lets you exchange data between 2D and 3D applications. The FBX file format, however, is more flexible than the PDF format, because data passed between applications via the FBX file format retains its ability to be manipulated.

When data recognized by FBX technology is exported to the FBX file format, it is stored in a lossless manner, retaining full fidelity and functionality. When importing an FBX file, FBX technology in the destination application examines the incoming data. If the data type is not recognized by the destination application, the FBX plug-in tries to convert the data to a form that produces a result functionally equivalent to the result produced in the source application.

For example, say you have an animation produced using a specific type of fcurve:

- If the FBX processing technology recognizes the type of curve, it is stored in a lossless manner and retains full fidelity and functionality in the FBX file.
- If on import the destination application recognizes the type of curve, the result is lossless interoperability.
- If on import the destination application does not recognize the type of curve, the FBX plug-in attempts to interpolate the curve's data so its intended result can be reproduced as closely as possible in the destination application.
- If on export the FBX technology does not recognize the curve type, the curve is automatically resampled (interpolated) using a type of curve supported by the FBX plug-in. When the FBX file is imported into the destination application, the result is full visual fidelity, but your ability to manipulate and edit the data is limited.

Note: The FBX Warning Manager reports all conversions, warnings, and errors after export and import operations. Warnings keep you informed of what the plug-in needs to modify and in what manner. They do not mean there was a problem. Errors, on the other hand, require attention. An error message means the FBX plug-in encountered something unexpected or the operation may have failed.

Taking this example one step further, say your animation uses TCB (tension, continuity, bias) curves. FBX processing technology is TCB curve-aware, so when the animation is saved in an FBX file, the curve's data is retained. When you import that curve into another application that is TCB curve-aware, the curve appears exactly as it appeared in the source application. If the destination application does not recognize TCB curves, FBX technology attempts a conversion that reconciles the intended result of the TCB curve with existing functionality in the destination application.

FBX technology occasionally encounters data types that cannot be reconciled between two applications. These situations are usually caused by radically different design architectures in the source and destination applications. Autodesk 3ds Max and Autodesk Maya software are prime examples. At the source code level, Maya is based on a graph of nodes, whereas 3ds Max is based on Modifier Stacks. Another difference is that Maya uses two-sided objects and 3ds Max uses two-sided materials. This issue cannot be reconciled by the interchange technology; it must first be addressed by the source and destination applications.

And herein is the source of confusion. FBX technology does not create new functionality in applications but rather rebuilds scenes from the source application using available functionality in the destination application.

FBX technology supports a wide variety of 2D and 3D data types. If the application you are exporting data to recognizes and supports the same data types and shares algorithmic equivalence, your interchange experience should be seamless and lossless.

In many cases, however, different 3D authoring applications are based on structurally different architectures. They use different algorithms to produce similar effects. Certain features are proprietary and therefore not transferable to other applications.

To accommodate situations that are otherwise impossible to resolve, baking assets is an effective way to move the desired result to an application that would not be capable of producing them otherwise.

Interoperability & Emulation

Things FBX Technology Can and Cannot Do

FBX technology supports a wide variety of 2D and 3D data types. The list of data types that it supports is constantly growing, so the following should not be construed as an FBX technology compatibility chart. Rather it is a high-level road map to help you understand what FBX technology can and cannot do, and to help you navigate the differences in 3D content creation tools—in particular 3ds Max, Maya, and MotionBuilder—to achieve the best results.

Geometry

Meshes and NURBS are fully supported in FBX technology. Subdivision surfaces, however, are not yet supported. Maya and 3ds Max use different subdivision surface algorithms, so if you need to move a model built with subdivision surfaces from one application to the other, export it as a mesh with a specific resolution.

Deformers

Skinning and the Maya clusters, as well as smooth and rigid bodies, are supported. Vertex caching—the ability to bake vertex animation in both 3ds Max and Maya—is also supported. The FBX SDK and plug-ins can convert 3ds Max representations to Maya and vice versa. This capability is useful for modeling and animating in one application and rendering in another. For example, you have a complex character rig in Maya that you want to render in 3ds Max, but you know that the complex control rig involves constraints and expressions that FBX technology cannot currently transport with full fidelity and functionality. Vertex caching enables you to bake the animation of the vertices into a point cache file. That file can be played back in 3ds Max with exactly the same vertex animation it had in Maya, resulting in a final rendering that looks identical to how the animation looked in Maya.

You retain all the visual fidelity, but you lose the ability to modify the animation in the new application, because the animation consists of vertices that are moving around without any of the data associated with the character rig you used to build it.

Materials and Shaders

FBX technology supports materials and submaterials as the 3ds Max architecture defines them. The Maya architecture defines materials in a manner different from 3ds Max. In programming terms, a micrograph of a network box is used to express material relationships in Maya, and FBX technology approximates 3ds Max materials with a micrograph that behaves as it would within a Maya application.

Procedural shaders are not supported by FBX technology, which cannot reproduce the procedural shader algorithms found in the source application. To share the visual results of a procedural shader between applications, bake them into an image file before exporting your FBX file.

UVs are similar to procedural textures in that many tools in 3ds Max and Maya generate them using algorithms. FBX technology does not transport those algorithms. It transports UVs that have been baked onto a model. If you used a modifier in 3ds Max to unwrap the UVs or projected UVs onto your model, FBX technology takes a snapshot of those vertices. That snapshot captures the UVs at that moment in time and is transported in the FBX file.

Weights transfer among 3ds Max, Maya, and MotionBuilder well.

One layer of vertex colors is supported, as are normals. But there are a lot of misconceptions about normals. FBX technology transports geometry normals information. Both 3ds Max and Maya usually compute normals on the fly. The 3ds Max algorithm for computing normals, however, is different from that in Maya. When transferring a model between 3ds Max and Maya, you may find that the normals are not always where they were in the original application. When this happens, users sometimes assume the normals did not transfer properly. In the case where default geometry normals do not transfer properly, you need to display and verify the normals. You can explicitly define these normals in Maya using the Set Normals Angle function to correct the issue.

FBX technology also transports the smoothing group information from 3ds Max. But there is no equivalent to smoothing groups in Maya, so FBX technology converts 3ds Max smoothing groups into soft edge/hard edge information that affects the geometry normals in Maya.

Animation and Constraints

FBX technology supports all the function curves used by 3ds Max and Maya software. Note that 3ds Max and Maya do not themselves support the same types of fcurves. For example, Maya does not recognize TCB curves, whereas

3ds Max does. If an animation created with TCB curves is exported to Maya, the animation is saved to an FBX file. The FBX file is TCB-curve aware, so it stores the curve in a lossless manner. When the Maya FBX plug-in encounters the TCB curve during import, FBX technology resamples the curve to approximate its intended result. The FBX technology converts the data to reproduce the animation in the destination application. Without resampling the curve, the animation could not have been transported from 3ds Max to Maya.

The Maya Full Body IK (FBIK) constraint and the MotionBuilder Character constraint are compatible and supported by FBX technology. Solver input data is transported between these two applications. However, there is no equivalent in 3ds Max. Biped is as close as 3ds Max gets, but there is currently no conversion between the FBIK or the Character constraint and Biped. You can, however, export Biped data to Maya or MotionBuilder, which can recognize the data and build a corresponding solver. To move FBIK animations from Maya to 3ds Max, bake its results using the bake complex animation feature in the FBX export plug-in. To move character animations from MotionBuilder to 3ds Max, bake its results using the plot character function in MotionBuilder, before exporting your FBX file. To bring animation back onto a 3ds Max Biped, you must first open the original Biped file in 3ds Max and then import your FBX file onto this Biped (Bipeds must match for results to be identical).

FBX technology supports the basic constraints in Maya and MotionBuilder: position, scaling, point, aim, and orient. You can exchange assets between Maya and MotionBuilder using those constraints.

Note: Maya constraints are not currently supported in 3ds Max but can be baked or plotted (using the bake complex animation feature in the Maya FBX export plug-in) to transfer the resulting animation. Similarly, MotionBuilder constraints can be plotted using the animation/plot properties functionality before exporting the FBX file.

MotionBuilder is able to layer different animation curves. The result of those layers is computed only at runtime. FBX technology transports these layers. The Maya FBX import plug-in provides the option to either choose the base layer of animation or bake all the layers into a single fcurve (merge animation layers). 3ds Max currently bakes all layers into a single fcurve on import by default. This feature is useful if you have received animation built with multiple curves in MotionBuilder that have not already been baked to a single fcurve. The 3ds Max and Maya FBX plug-ins enable you to do so when importing the FBX file containing the multiple-fcurve animation from MotionBuilder.

The FBX SDK supports all of the motion-capture data capability in MotionBuilder. And if you are exporting motion-capture data in the FBX file format, it transfers with complete fidelity.

Maya and 3ds Max handle local animation differently. Maya provides six matrices and vectors (including rotation and scaling pivots) that handle global transformations of objects, and pivot points can be part of a parent/child relationship. For example, when modeling a door, you start with a cube and scale on one side of it so the cube resembles a door. The door has a knob on it. When you modify a pivot point to open the door, the knob moves with the door, because the knob is set up as a child of the door.

That scenario is quite different in 3ds Max, which uses a single rotation/translation/scale pivot that modifies geometry. Child nodes do not inherit pivot point modifications. In the door example and using 3ds Max, the only thing modified by the pivot point is the vertices of the door. Visually, the result looks identical—the door and its knob swing open.

Exporting the door-opening sequence from Maya to 3ds Max requires that FBX conversion technology recompute the local coordinates of all the objects and the function curves related to them. But the result is still lossless—you see the door and its knob swing open. Animators looking at the keyframe data, however, would notice that although they started with three keyframes, the exported sequence has a keyframe for every frame of the animation sequence. So they might wonder what FBX technology did to their animation: FBX technology compensated for the architectural differences between 3ds Max and Maya.

Cameras and Lights

FBX technology supports all data related to cameras in both 3ds Max and Maya, but there are major differences in each application's camera scheme. Maya animates the camera's focal length and recomputes the angle of view (also known as field of view or FOV). 3ds Max does the opposite. It animates the FOV and computes the focal length. When transferring camera data between 3ds Max and Maya, FBX technology recomputes the camera frame by frame. On playback, the results look identical whether you started in 3ds Max and are playing back in Maya or vice versa.

For lights, FBX technology currently supports a subset common to MotionBuilder, 3ds Max, and Maya: spotlights, point lights, and area lights. FBX technology converts unsupported light types to their closest equivalent. For example, a 3ds Max photometric target area light is converted to a standard point light when the scene containing the photometric light is exported to an FBX file. Some environmental lights, such as mrSky, are replaced by a null. It is important to note that when the FBX plug-in does a conversion of this sort, the FBX Warning Manager always notifies you. In this example, the messages displayed would read as follows:

- The following lights are not supported by the plug-in and will be exported as Point light: Area01.
- The following lights are not supported by the plug-in and will not be exported: Sky01.

Rendering attributes—render frame, environment, and mental ray® software settings—are not part of the asset exchange in FBX technology.

Visibility is supported by FBX technology, but the visibility scheme in 3ds Max is different from that of Maya. This means the interoperability functionality is different when the source application is 3ds Max and the destination application is Maya than it is the other way around. In Maya, if you turn off the visibility of a parent node, all of its children inherit its invisibility. So, for example, if you wanted to hide your entire character in Maya, you would simply turn off the visibility of the hips, making the entire character hierarchy—legs, arms, head, and so on (all children of the hips)—invisible. In 3ds Max, visibility is not passed from parent to children. Turning visibility off on the hips would only make the hips invisible in 3ds Max. In other words, although invisible in Maya, the character would be visible (without hips) in 3ds Max.

Embedded Files

Other file formats can be embedded in FBX files. FBX files can carry audio, video, 2D texture files, and so on. Virtually any file type can be embedded. MotionBuilder takes full advantage of this capability. Maya and 3ds Max take advantage of texture file embedding but do not handle embedded audio in the FBX file. With a single exception, files embedded in an FBX file are direct binary copies—the data in embedded files is unchanged.

The exception is texture files, which can be converted to TIFF files in two color spaces: RGB/RGBa and YUV. The 3ds Max and Maya FBX plug-ins give you the option to uncompress TIFF files for improved interoperability between applications. The FBX processing technology retains the original file format but embeds the TIFF version of the texture. When the file is reopened, it gets converted back to the original format if there is a function in the destination application to do so. If there is no such converter, the file remains in TIFF format. If you start with a JPEG file, on export the FBX technology creates a TIFF file in the YUV color space. If you start with a BMP file, on export FBX technology creates a TIFF file in the RGBa color space.

The current FBX SDK supports 12 file formats, including FBX ASCII/binary, OBJ, DXF™, 3DS, COLLADA, and custom properties. An example of the latter would be Custom Attributes created using the 3ds Max parameter editor.

Summary

FBX technology and tools are constantly evolving. The engineers at Autodesk are constantly adding to its functionality and intelligence. User feedback is of vital importance in helping to ensure that the technology is moving in the right direction.

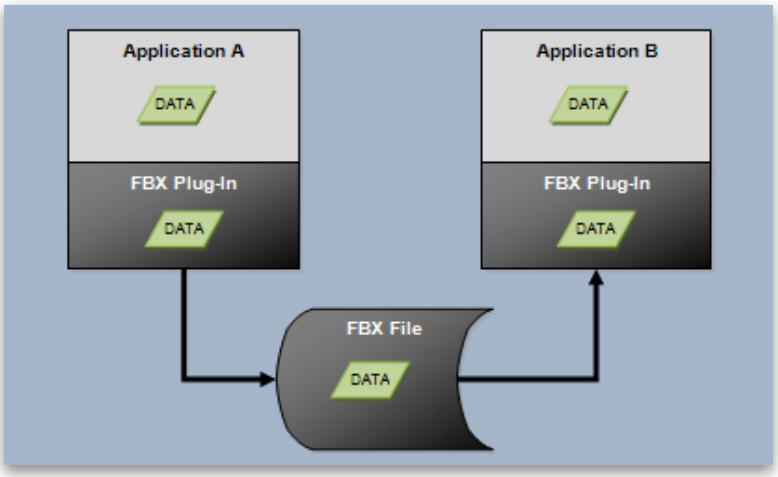
Each new version of FBX technology supports the most recent and the last two major releases of 3ds Max, Maya, and MotionBuilder software products. This is meant to accommodate pipelines that are locked down while engaged in two- or three-year production cycles. In addition, newer versions of the FBX plug-ins can read and save earlier versions of the FBX file format.

The FBX technology family of tools provides universal 3D asset interchange, helping to break through data compatibility barriers. FBX is a platform-independent file format that enables you to access legacy content after upgrading your pipeline to take advantage of the latest 3D tools; exchange digital assets with other studios whose pipelines are standardized on tools different from your own; and use multiple 3D tools to realize your creative vision in the most efficient manner possible.

Appendix

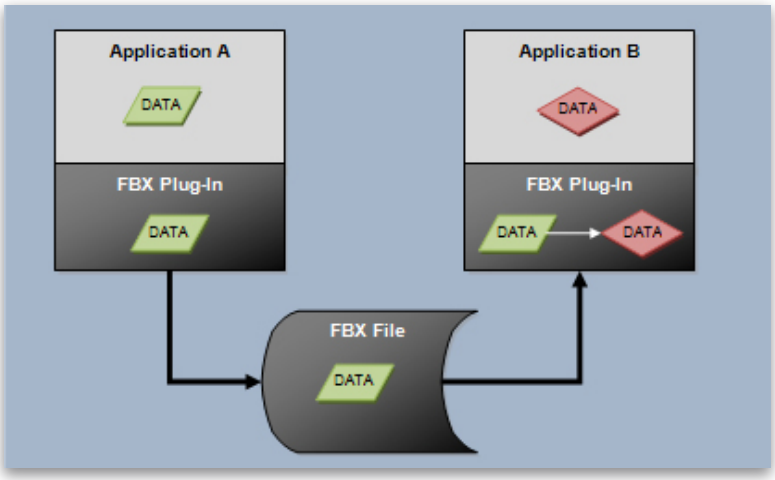
The Three Levels of Interoperability

The most common misconception about FBX technology is that it enables you to exchange all data seamlessly between any applications. However, seamless data interchange is possible only if functionality and algorithms are identical in both the source and destination application. It is important to understand that the FBX plug-in rebuilds scenes from the source application using available functionality in the destination application.



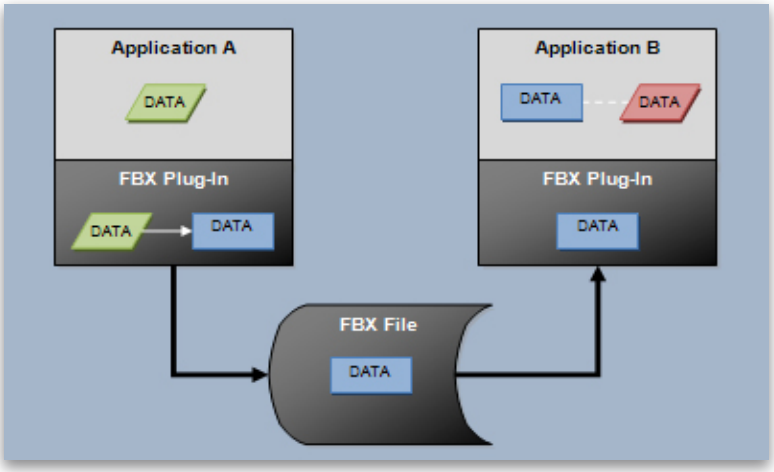
Perfect compatibility:

Data passed from a source application is recognized by the destination application, yielding identical results.



Data (interpolated) compatibility:

Two applications do not use identical algorithms to achieve certain functionality. Data passed between the applications is converted or interpolated to yield results that are functionally equivalent. The converted data, to some extent, can still be manipulated and edited.



Emulated compatibility:

Two applications have completely different capabilities, so transferring data between them requires that data be baked on export to an FBX file using the bake complex animation feature of the FBX plug-in. Baked data transfers visual fidelity. The ability to manipulate and edit baked data is limited.

Autodesk®