

# FLEXPORTER

Pierre Terdiman

Last revision : June, 18, 2001

<http://www.codercorner.com/Flexporter.htm>

*Flexporter* is a free utility plug-in for *3D Studio MAX*, designed to easily export a lot of information out of it.

Flexporter is built on top of *ICE*, which is the latest incarnation of my own development framework. That is, Flexporter needs some ICE dlls to run. On the other hand, the plug-in heavily takes advantage of the existing framework and then provides additional features such as automatic convex hull computation, texture downsampling, etc.

Current features :

- Cameras
- Lights
- Materials and submaterials
- All textures (diffuse, opacity, bump, whatever)
- Texture downsampling and filtering if needed
- Cropping values
- Meshes
- Dummy objects
- Gizmo objects
- Character Studio BIPED and PHYSIQUE
- Hierarchy
- Mesh instances
- User defined properties
- Groups
- Environment
- Time related information
- Sampled animation controllers (PRS, some camera & light parameters)
- Visibility track
- Automatic [convex hull](#) computation
- Automatic [bounding sphere](#) computation
- Automatic inertia tensor computation
- Mesh consolidation
- User-defined export format
- Morphing
- Shapes
- Texture quantization

Supported versions : MAX 3 and MAX 4.

MAX 2 is not supported anymore.

## Flexporter : a flexible exporter

Flexporter can be customized with plug-ins. Yes, I have plug-ins for my plug-in ! Flexporter comes with the *Flexporter SDK*, which allows you to build your own export format, in case the default ones don't match your needs.

The two default export formats are ASCII and ZCB. Each export format has its own dll, whose extension has been renamed into *flx*.

The ASCII format is a standard ASE-like text file.

The ZCB format is a very simple chunk-based binary format.

Those Flexporter plug-ins, as well as your own ones, are MAX3 *and* MAX4 compatible.

## Installation

- 1) Copy Flexporter.dlu in MAX's Plugins directory.
- 2) Copy IceASCIIExporter.flx and IceZCBExporter.flx in MAX's root directory. (for example in C:\3dsmax3\_1\ ) All your own exporter plug-ins must be copied in the root directory as well.
- 3) Copy ICE dlls in a reachable place. A convenient one is the System directory from Windows.

## Usage

### Export

Just klik here to export the current scene. Static information is extracted from current MAX frame.

### Options

Click here to display the Options panel. This one allows you to choose your export format (i.e. the exporter plug-in you want to use), and change the export settings.

### Copy textures

I worked with artists whose scenes used a little subset of a giant texture directory. This option allows you to copy the textures used by current scene to another place, so that you don't have to look for them in the big directory, which often is a big mess.

## **Export BIPED motion**

The animation for BIPED characters can be exported in independant *motion files*. That way, you can export a particular skin once, then multiple independent motion files to use with that skin. This option calls the exporter plug-in only to export the motion data.

A special window will be displayed before exporting a motion, so that you can customize the motion with a name (possibly different than the filename) and a type. This is useful in motion-related state machines, to use generic scripts on different characters. (say for example to perform motion blending from type « Idle » to « Walk »)

## **Unload exporters**

This option unloads the Flexporter plug-ins (.flx files). What is it useful for ? Usually, when building a MAX plug-in, you must close and restart MAX each time you recompile your plug after a modification. That's because MAX loads all plug-ins at start of the day, once and for all. Now, the same limitation doesn't exist for Flexporter plug-ins. You can rebuild your Flexporter plug-in and test the new version *without leaving MAX*. The recipe is :

- unload the exporters
- recompile your Flexporter plug-in
- now you can export your scene again, the new versions will be automatically loaded !

## **Flexporter Options Panel**

Here is a brief description of Flexporter's options :

### **Selected exporter**

No options there, just the name of currently selected exporter, the extension string, and a button (« Next plug-in ») to jump to the next available format.

There you can also find two extra-buttons :

- « Load settings » loads all the saved settings from the registry.
- « Save settings » saves all the settings to the registry.

Different settings can be saved for each export plug-in.

## General settings

### **Compress data**

This option is used to compress ZCB files.

- No compression : files are saved uncompressed.
- Compress with Zlib : vertices are quantized, faces are delta-encoded, and the resulting file is compressed with Zlib
- Compress with Bzip2 : vertices are quantized, faces are delta-encoded, and the resulting file is compressed with Bzip2.

You can choose the number of bits used to quantize any floating-point values thanks to the slider.

The Flexporter SDK contains code to dequantize the vertices and get them back.

### **Export complete scene**

You can export the whole current scene, or just selected nodes. Before exporting, a selection panel is displayed in which you can choose the nodes from MAX you want to export. Exporter plug-ins aren't called for unselected nodes.

### **Export visibility track**

This option is used to export the possible visibility track for each node. Mainly useful for rides.

### **Export hidden nodes**

Hidden nodes are not exported, unless you explicitly force the export by using that option.

## Character Studio

### **Export PHYSIQUE**

Exports PHYSIQUE skins as actual skinned characters, else export them as standard meshes. Exporting them as standard meshes allows one to use a MORPH controller on skins.

### **Export bones and BIPED parts**

Exports bones and BIPED parts as actual meshes, else discard them.

### **Always export motion**

Exports Character Studio motions within the main exported file. The other way is to export the motion as a separate file.

## Meshes

### **Export UVW**

Exports the UVW mapping coordinates, else discard them.

### **Discard W in UVW**

Exports the UV's only and discard W, which saves some bytes. For most apps you don't need W, but Flexporter can export it anyway for the days we'll need it. (3D textures, someone ?)

### **Export vertex-colors**

Exports the vertex colors, else discard them. Some PSX games abundantly use vertex colors, because that's cheaper than dynamic lighting.

### **Use smoothing groups**

You can keep the smoothing groups, or just force each face to belong to the same group. Keeping the smoothing groups usually produce more replicated vertices within a vertex buffer, so pay attention to what you do and what you need.

### **Remove scaling**

This option removes all non-uniform scaling from meshes. That way you don't even have to save the scale, and the DX7 lighting doesn't blow up.

NB : whatever happens, non-uniform scaling are always removed from BIPED parts.

### **Convert to D3D frame**

MAX frame is right-handed, Direct3D frame is left-handed. This option converts data from MAX to Direct3D, which involves vector and quaternion transforms.

### **Export absolute PRS**

PRS stands for Position-Rotation-Scale. Position and Scale are exported as 3D vectors. Rotation is exported as a quaternion. Usually you want to export PRS data relative to the parent node, but sometimes you need to export absolute values (for example to test your new renderer which doesn't support hierarchy yet.)

### **Compute convex hull**

Flexporter can compute and export the convex hull of a mesh thanks to the underlying ICE framework. The convex hull is computed in an object's local frame, i.e. you'll need to transform it with the world matrix as well. The hull is exported as an independent mesh, with its own vertices and faces.

## Compute bounding sphere

Flexporter can compute and export the bounding sphere of a mesh thanks to the underlying ICE framework. The bounding sphere is computed in an object's local frame, i.e. you'll need to transform it with the world matrix as well. The sphere is exported as a center and a radius.

## Compute inertia tensor

Flexporter can compute and export the inertia tensor of a mesh thanks to the underlying ICE framework. Flexporter computes the inertia tensor relative to the origin, and relative to the center of mass.

## Export edge visibility

Flexporter can export visibility codes for each edge. Since this option is not often useful, you can discard it to save some bytes in the final file.

## Make manifold

Flexporter can fix non-manifold meshes by removing all of its singular edges. An edge of a triangular mesh is a boundary edge if it belongs to exactly one triangle, an internal edge if it is shared by exactly two triangles, and a singular edge if it is shared by three or more triangles (Note that boundary edges and internal edges are also called regular edges.) A triangular mesh is a manifold if it has no singular edges.

To remove singular edges, Flexporter replicates some vertices, and that's why fixed meshes have more vertices than the original ones.

## Export target nodes

Exports cameras & lights target nodes, or discard them. (this setting has been moved with the mesh ones, since targets are meshes within MAX)

## Consolidation settings

### Consolidate mesh

MAX native meshes are painful to deal with : basically you have 3 geometries and 3 topologies for the vertices, the UV's and the vertex colors. Thanks to the underlying ICE framework, Flexporter can rebuild a clean mesh for you – a process called *consolidation*. (see Real-Time Rendering, by Möller & Haines). The complete work requires :

- Packing multiple independent topologies in a single hardware-friendly one.
- Tracking and deleting redundant vertices, mapping coordinates or vertex colors.
- Killing zero-area or redundant faces.

- Computing normals according to the smoothing groups.
- Outputting normal information used to recompute them easily in realtime.
- Grouping output faces in submeshes, according to their rendering properties.
- Duplicating as few vertices as necessary.
- Building triangle strips if needed.
- Building progressive meshes

Support for strips and progressive meshes have been temporarily removed, even if my old MAX exporter computed and exported them as well.

### **Compute face normals**

Flexporter can compute and export face normals for you.

### **Compute vertex normals**

Flexporter can compute and export vertex normals for you. This is useful since you don't have to deal with the smoothing groups, Flexporter does the job.

### **Export normal info**

Flexporter can export extra normal information, used to recompute the vertex normals in realtime according to the smoothing groups. This may be used to recompute the vertex normals of Character Studio skins, each frame.

### **Weight normals by area**

Use this option to activate a second normal-computation mode, where face areas are taken into account.

## **Textures**

### **Max. texture size**

You can select the maximal allowed texture size there. Before exporting, Flexporter downsamples any needed texture so that it fits that allowed size. If you need arbitrary sized textures, select *Unlimited* there.

### **Export filenames only**

Some export formats can't deal with direct texture data (e.g. the ASCII format). This option allows you to export a texture's pathname only, so that your renderer can further do the right job itself. If you don't want to deal with texture loading (which suppose you can read TGAs, BMPs, JPGs, TIFFs, whatever), Flexporter can export any used texture under the form of an array of pixels. Each pixel is a 32 bits value (R,G,B,A) for an optimal texture quality.

### **Opacity map in diffuse alpha**

This option allows you to pack an opacity map in the alpha channel of the diffuse texture, as used in most rendering APIs. If you don't select this option, the opacity map is exported as a standard texture.

### **Flip horizontal**

Flip the texture horizontally before exporting it.

### **Flip vertical**

Flip the texture vertically before exporting it.

### **Quantize**

Quantize the texture and export it as a 256-colors paletted picture.

### **Keep ambient texture**

Exports possible ambient textures, or discard them. This seems not to exist anymore in MAX 3.

### **Keep diffuse texture**

Exports possible diffuse textures, or discard them.

### **Keep specular texture**

Exports possible specular textures, or discard them.

### **Keep shininess texture**

Exports possible shininess textures, or discard them.

### **Keep shining strength texture**

Exports possible shining-strength textures, or discard them.

### **Keep self-illum texture**

Exports possible self-illumination textures, or discard them.

### **Keep opacity texture**

Exports possible opacity textures, or discard them.

### **Keep filter texture**

Exports possible filter textures, or discard them.

### **Keep bump texture**

Exports possible bump textures, or discard them.

### **Keep reflexion texture**

Exports possible reflexion textures, or discard them.

### **Keep refraction texture**

Exports possible refraction textures, or discard them.

### **Keep displacement texture**

Exports possible displacement-mapping textures, or discard them.

## Helpers

### **Export helpers**

Export helpers, or discard them.

## Lights

### **Export lights**

Exports lights, or discard them. By default, all lighting parameters are exported.

### **Export attributes controllers**

Export supported light attributes controllers, or discard them. Currently supported controllers are :

- Color
- Multiplier
- Contrast
- Diffuse Soften
- Attenuation Near Start
- Attenuation Near End
- Attenuation Far Start
- Attenuation Far End
- Shadow Color
- Shadow Density

### **Compute lighting data**

Lighting is precomputed for each vertex and exported as well.

### **Compute shadows**

Shadows are taken into account in the previous computation.

### **Color smoothing**

An extra smoothing filter is applied on the results of the previous computation.

### **Compute lightmaps**

Not working yet.

## **Cameras**

### **Export cameras**

Exports cameras, or discard them. By default, all cameras parameters are exported.

### **Export FOV controller**

Export the field-of-view controller, if it exists.

### **Export attributes controllers**

Export supported camera attributes controllers, or discard them. Currently supported controllers are :

- Near Clip
- Far Clip
- Target Distance
- Near Env Range
- Far Env Range

## **Materials**

### **Override ambient color**

This option allows you to override a material's ambient color, and set it as pure white.

### **Override diffuse color**

This option allows you to override a material's diffuse color, and set it as pure white.

## **Animation**

### **Export single frame**

Use this option to export data for a single frame. Data is taken from the current frame in MAX.

## Sampling

You can export samples or keyframes. If you choose the sampling method, you can use the slider to select the sampling rate.

Sampling rate = 1 => one sample is exported for one frame

Sampling rate = 2 => one frame is exported, one is skipped, and so on.

## Shapes

### Export shapes

Exports shapes, or discard them.

## User-defined properties

You can override some global settings for a particular node thanks to the user-defined properties. Here are the available commands :

FORCEEXPORT	force the node to be exported
FORCECONSOLIDATION	force the node to be consolidated
FORCEMANIFOLD	force the node to be cleaned / fixed
FORCENORMALS	force the normals to be computed
EXPORTPRS	force the PRS to be exported
EXPORTVIS	force the visibility track to be exported
EXPORTFOVCTRL	force the FOV controller to be exported
EXPORTCAMCTRL	force camera attributes controllers to be exported
EXPORTLITCTRL	force light attributes controllers to be exported
EXPORTUVW	force the UVWs to be exported
EXPORTVTXCOL	force the vertex colors to be exported
EXPORTSMG	force the smoothing groups to be used
DISCARDEXPORT	force the node to be discarded
DISCARDCONSOLIDATION	disable consolidation for the node
DISCARDMANIFOLD	disable manifold cleaning for the node
DISCARDNORMALS	disable normal computation for the node
DISCARDPRS	force the PRS to be discarded
DISCARDVIS	force the visibility track to be discarded
DISCARDFOVCTRL	force the FOV controller to be discarded
DISCARDCAMCTRL	force camera attributes controllers to be discarded
DISCARDLITCTRL	force light attributes controllers to be discarded
DISCARDUVW	force the UVWs to be discarded
DISCARDVTXCOL	force the vertex colors to be discarded
DISCARDSMG	force the smoothing groups to be discarded
MORPH	exports a Morph Controller for the tagged mesh

## Known bugs (a.k.a. *features*)

- Inertia tensors computation often fails for complex meshes. Even the standard MAX teapot seems to make the code crash. Since the original code was written by Brian Mirtich (i.e. that's not mine), I don't know whether it's normal or not, and fixing it might take some time. Now, you still can play with basic objects in your rigid body simulator.
- A rather severe limitation exists regarding Character Studio export. Please see the FAQ for more details.
- The texture quantization code sometimes crashes on some textures. (?)
- Beware of non-collapsed shapes (see the FAQ)