



AMXtex DirectX .X Exporter for Animation:Master  
Help File (v2006.04.29)

#### Legal Information

AMXtex © 2001-2006 Christopher J Roy, Obsidian Games

“Hash” and “Animation:Master” are trademarks of Hash, Inc.

“Microsoft”, “Windows”, “DirectX”, and “Direct3D” are trademarks of Microsoft Corporation

All other trademarks are owned by their respective owners

# Table of Contents

General AMXtex Information.....	3
Introduction to AMXtex.....	3
AMXtex Features.....	3
Exporting Models from Animation:Master.....	4
How to export a single model.....	4
How to export multiple models.....	5
Export Options.....	6
Mesh Export Method.....	6
Polygons Per Patch.....	7
Mesh Options.....	7
Export Material Attributes.....	7
Set Face Alphas to 1 (Opaque).....	7
Export Mesh Normals.....	7
Export Duplicate Vertices.....	7
Ignore Duplicate Normals.....	8
Texture Options.....	8
Export Texture Info.....	8
Set Textured Faces to White.....	8
Reverse Texture Sort (Bottom -> Top).....	8
Generate Model Maps.....	9
Bone Options.....	10
Export Bone Information.....	10
Export Weight Information.....	10
Model Orientation.....	11
Exporting Animations from Animation:Master.....	11
How to export animation.....	11
Export Options.....	12
Animation Export Method.....	12
Animation File Options.....	13
AnimationSet Name.....	13
Animation Timing Interval.....	13
Registering AMXtex.....	13
Entering Registration/License Information.....	13
AMXtex Support.....	14
Help and FAQ Files.....	14
Email Support.....	14
AMXtex Frequently Asked Questions (FAQ).....	14
What versions of A:M does AMXtex work with?.....	14
What's the difference between the unregistered and registered version of AMXtex?.....	15
What are the limitations of AMXtex?.....	15
Why does the percentage bar hang at 100 (or another) percent?.....	16
Why do the exported bitmaps sometimes display incorrectly? .....	16
Can I modify the exported .X file?.....	16

Why does my retained mode mode export incorrectly?.....	16
Why does my model look/animate wrong when using the 2005+ versions of the DirectX 9 SDK Mesh Viewer?.....	17

## General AMXtex Information

### ***Introduction to AMXtex***

AMXtex is an export plugin for Hash Inc's Animation:Master, which exports A:M .MDL files as DirectX .X files.

### ***AMXtex Features***

AMXtex currently exports the following features of a Hash Animation:Master model (MDL) file:

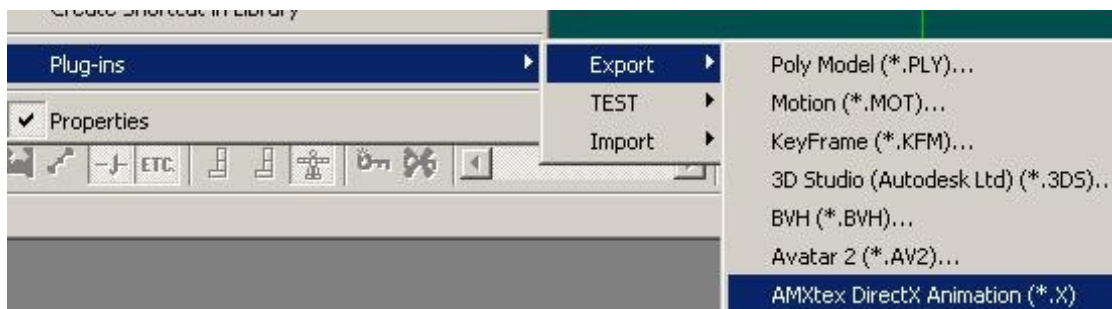
- Vertices
- Duplicate Vertices (for proper handling of textures and normals)
- Vertex Weights (supported on A:M v13 and higher)
- Polygons
- Face Colors
- Specular Highlights
- Alpha Transparency
- Texture Maps
- Normals
- Standard Bones
- Animation

## Exporting Models from Animation:Master

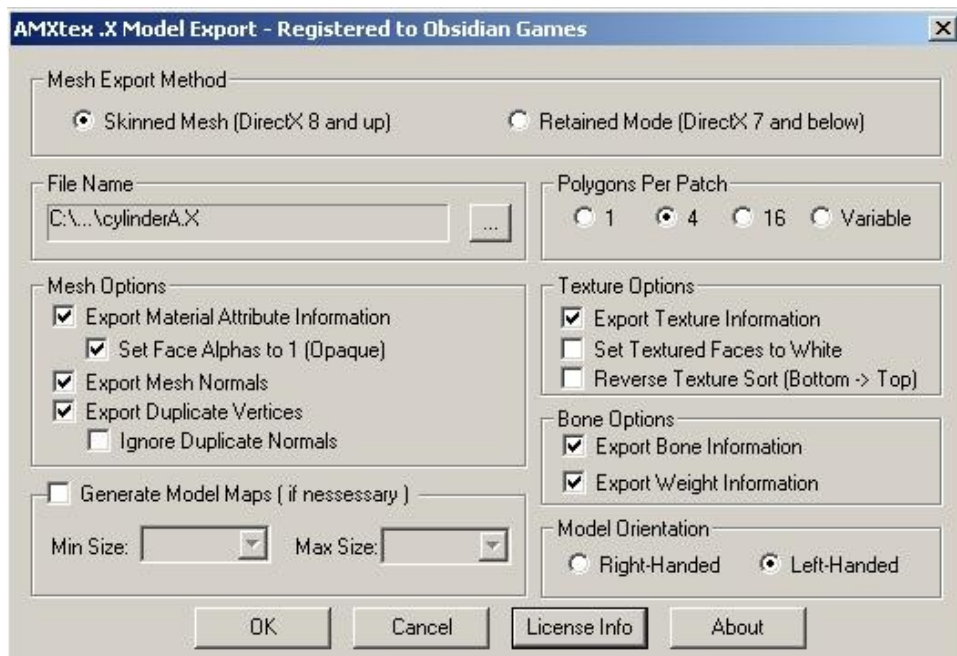
A model can be exported from Animation:Master in a number of ways.

### *How to export a single model*

To export a single model, right click the model object, and select "Plug-ins -> Export -> AMXtex DirectX Model".

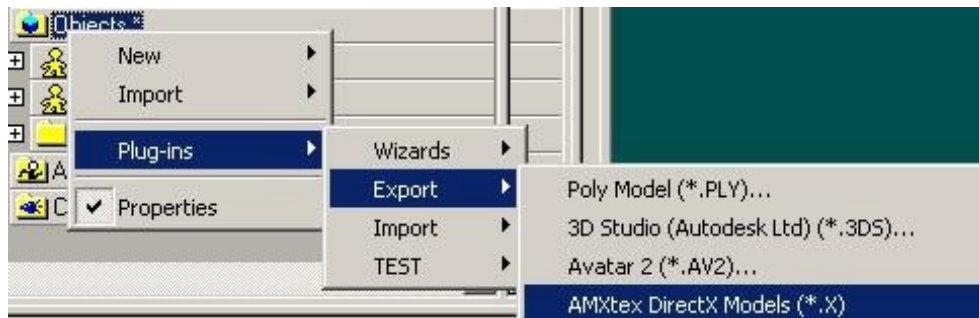


The Single Model Export Dialog (see below) should then appear. The default selections are the most common options, though you can change them as you see fit. When you're ready to start the export, select the OK button. When the status bar at the bottom of the Animation:Master screen reads 100%, the model should be exported.

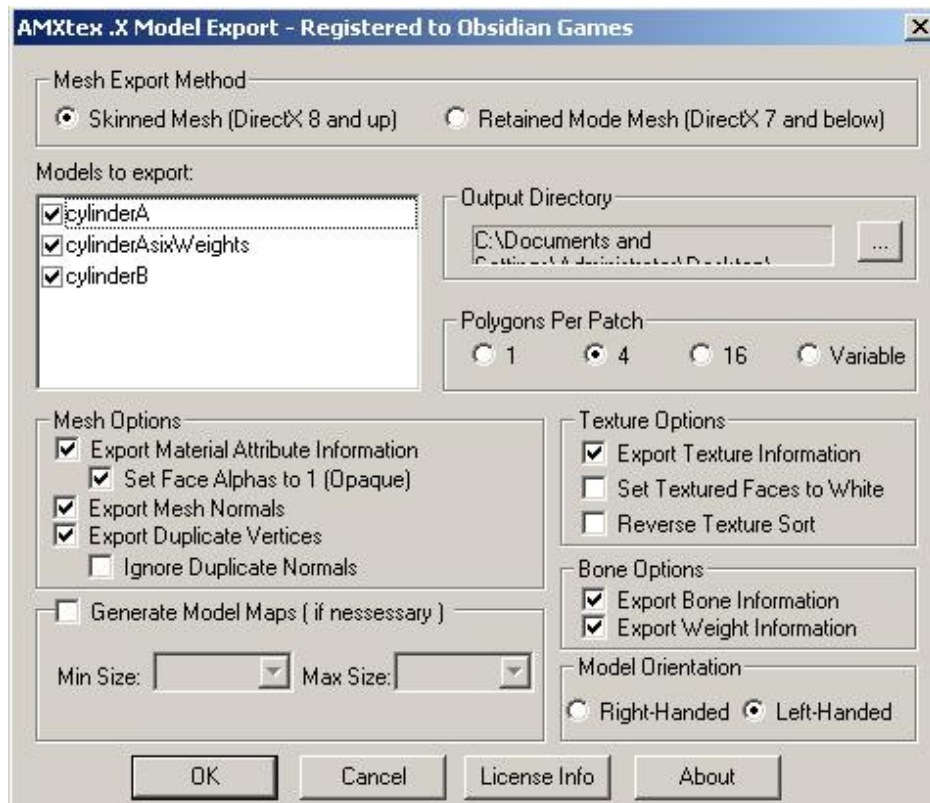


## How to export multiple models

To export multiple models at once, right click the Objects tab and select Plug-ins -> Export -> "AMXtex DirectX Models"



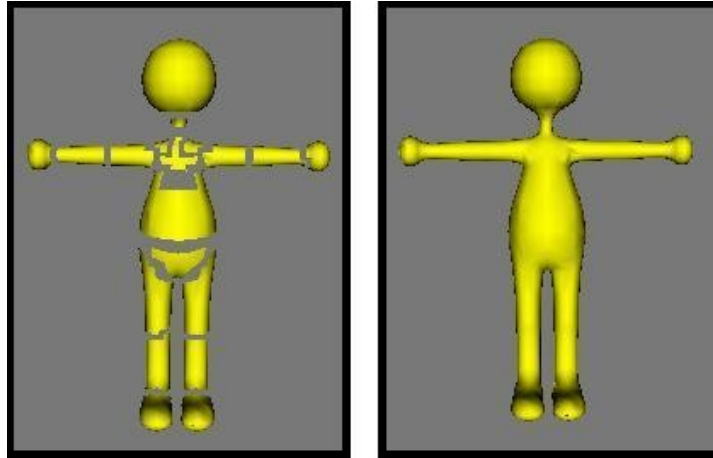
The Multiple Model Export Dialog (see below) should then appear. All open models will be listed, and checked off by default. If there are any models you don't want exported, deselect them before continuing. The default option selections are the most common options, though you can change them as you see fit. When you're ready to start the export, select the OK button. When the status bar at the bottom of the Animation:Master screen reads 100%, the model should be exported.



## ***Export Options***

### **Mesh Export Method**

This option allows you to export to the two types of DirectX .X models.



Prior to version 8, DirectX models did not support bone skinning and deformation. Instead, each bone represented a specific set of polygons, which meant models could only be composed of segments. Individual polygons could not belong to more than one bone. This type of model (seen above on the left) is called a Retained Mode model.

This made it difficult to get truly realistic looking characters, as each body joint had to be separate from the others. Modelers could get around this by overlapping joints, but even then, creases in the models were apparent.

With DirectX 8, .X files were updated to allow for non-segmented models. Model bones could connect with each other, allowing for mesh deformation. This type of model (seen above on the right) is called a Skinned Mesh model.

Overall, Skinned Meshes allow for far greater flexibility. If you have a program that allows for their use, it's probably the better way to go. However, Many 3D programming packages only use the Retained Mode format, such as WildTangent, DarkBASIC, and Blitz3D.

You'll need to keep the target format in mind when building your models. If your program has to use retained mode models, you'll need to plan on that fact when designing your model.

## Polygons Per Patch

This feature controls the level of detail in the exported model.

Hash models are stored in a patch format, while DirectX .X files are stored in polygon format. The polygon format is much faster in real-time display, but provides less detail than the patch model.

To make up for this lack of detail, you can choose to export multiple polygons per patch. The exporter will subdivide each patch into the number of polygons specified by this option. The result is higher detail, better curves, etc.

Caution: The more polygons per patch you export, the slower the resulting X file will display. For best results, test your model with varying levels of polygons per patch to see what gives the best balance between looks and speed.

## Mesh Options

### ***Export Material Attributes***

This option exports the face color, alpha transparency, and specular highlights information for each polygon.

*Note:* The material attributes must be exported if textures are to be exported, as textures are a subset of materials in a .X file.

*DirectX .X template:* Material

### ***Set Face Alphas to 1 (Opaque)***

This option will set the alpha transparency component of the Material colors to 1 (which is opaque).

### ***Export Mesh Normals***

This option exports the normals for a mesh.

*DirectX .X template:* MeshNormals

### ***Export Duplicate Vertices***

This option allows AMXtex to calculate and implement duplicate vertices. Duplicate vertices are needed when two textured polygons share a vertex.

Each vertex can only store the texture (U,V) coordinates of one texture. A duplicate vertex is used by DirectX to point to multiple instances of the same (x,y,z) vertex, but with different U,V and/or Normal information.

Using duplicate vertices also allows DirectX to optimize somewhat (since the same physical vertex is being used).

*DirectX .X template:* VertexDuplicationIndices

## **Ignore Duplicate Normals**

This option tells AMXtex to ignore duplicate normals when performing calculations on the export of duplicate vertices.

## **Texture Options**

### ***Export Texture Info***

This option exports all texture information for the model.

*Note:* Since .X files can only hold one texture per polygon by default, AMXtex will attempt to pick the correct texture on polygons with multiple texture maps.

*Note:* When viewing your exported .X file in a viewer, all textures must be in the same directory as the exported .X file. This can be changed later on by changing the path name in the TextureFilename directive of the .X file.

*DirectX .X template:* Material, TextureFilename, MeshTextureCoords

### ***Set Textured Faces to White***

When a texture is implemented, it's attached to whatever color material a polygon has. When displaying a textured polygon, DirectX will blend the color of a texture map with the underlying polygon face color.

This can sometimes lead to unwanted coloring/darkening of a texture. By selecting this option, the underlying face color will always be set to white, so the texture will always appear the correct color.

### ***Reverse Texture Sort (Bottom -> Top)***

This option reverses the algorithm that AMXtex uses to find the best texture for a multi-textured polygon.

Normally, AMXtex searches from the top texture on a polygon to the bottom texture. This option will reverse that search from the bottom texture to the top texture.



## Generate Model Maps

This feature is used to create a single model bitmap, for use with the exported .X file. A:M combines all the textures used in the model into a single image, which helps with complexity and will speed up .X model rendering. It does this by creating a portion of the bitmap for each patch in the model.

To use this option, you'll have to also have texture export selected. You'll also have to change the settings in your model's decal attributes. Make sure each Decals->Images->Properties->Map Export Method is set to "Include In Patch Maps". (See Figure 1).

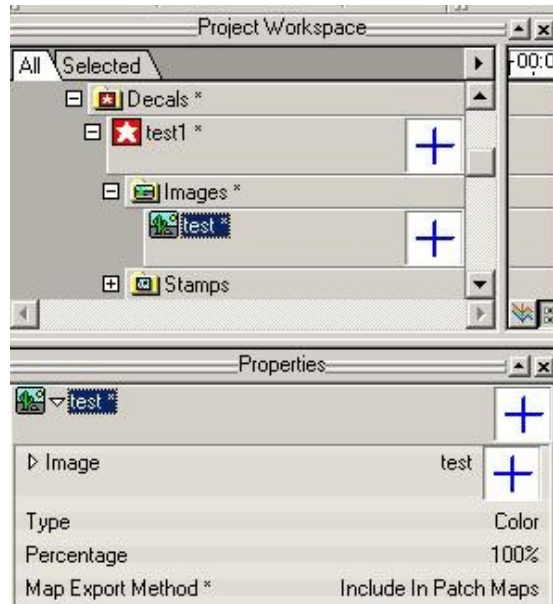


Figure 1: Setting a decal to export into single model bitmap

If you do not see the "Map Export Method" option, you may need to turn on "Show advanced properties". You can do this by selecting the "Options" option from the main menu's "Tools" menu. See Figure 2 for details.

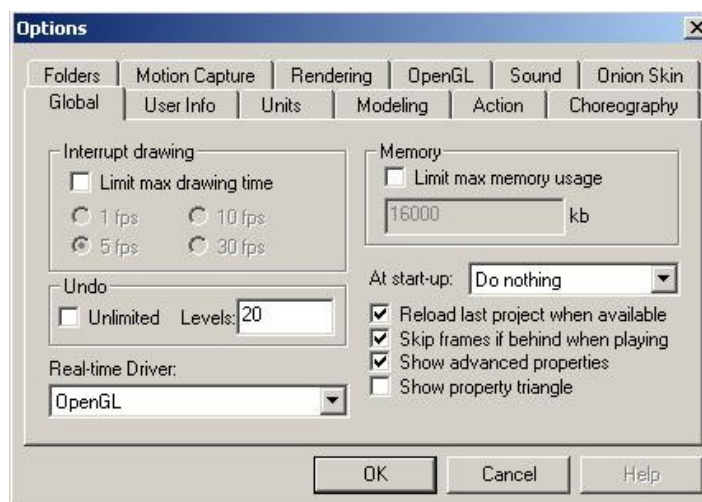


Figure 2: Turning on "Show advanced properties"

Note: The min/max width/height fields are used to set the desired size of each bitmap patch chunk inside the bitmap file. It is NOT used to set the bitmap dimensions. The bitmap dimensions always have a width of 1024, and a height determined by the number of textured patches in your model and the min/max width/height field in the export dialog.

## Bone Options

### ***Export Bone Information***

Select this option to export the Animation:Master bone hierarchy to the x file, in the form of Frames.

Please note that retained mode models must have bones exported, as they're used to position the various model segments.

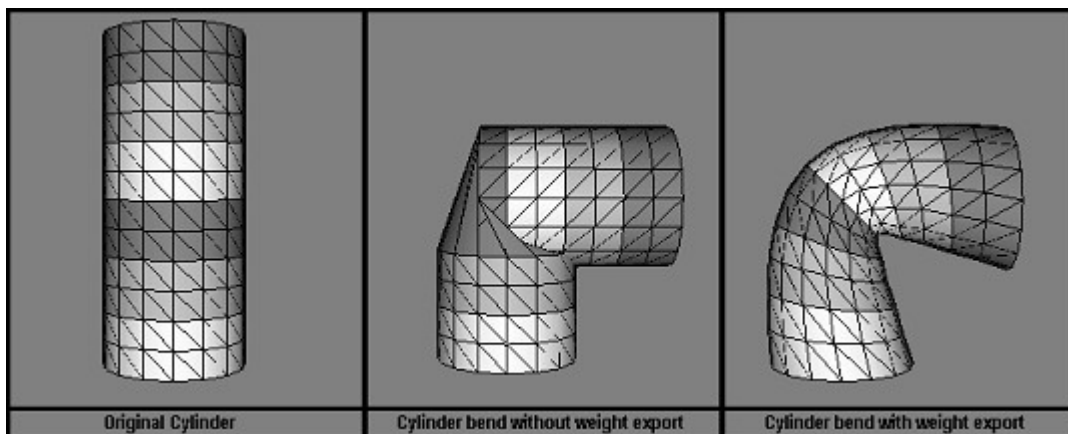
Additionally, DirectX bones are more limited in scope than their A:M counterparts. Where as A:M uses normal bones, target bones, constraints, etc, DirectX bones are simple matrices with associated vertices. Some advanced A:M bone features may still export correctly depending on your model and animation setup, but AMXtex is intended only for use with normal bones.

### ***Export Weight Information***

Select this option to export vertex weights (known in A:M as “CP Weights”). Vertex weights allow multiple bones to influence the animation of vertices, allowing for smoother joint bending. This option is only available on AMXtex for A:M v13 and higher, and is only available for use with Skinned Mesh export.

The DirectX format allows for vertices to be controlled by up to 4 bones. If a cp in A:M is controlled by more than 4 bones, the export process will remove the additional bone weights, and will re-weight the CP using the remaining 4 bones.

The following image helps demonstrate weights in action. It shows a two-boned cylinder (exported to the .X format using AMXtex) in various stages. The first image is the model itself, the second shows a bend action **without** weight export, and the third image shows the bend action **with** weight export.



## Model Orientation

This option sets the mesh orientation. 3D games and programs use one of two coordinate systems, right-handed and left-handed.

Typically, the coordinate systems are set up the with the following characteristics. Both coordinate systems have the positive X and Y axis point in the same direction. However, right-handed coordinate systems have the positive Z axis coming *out* of the screen, where as left-handed systems have the positive Z axis going *into* the screen.

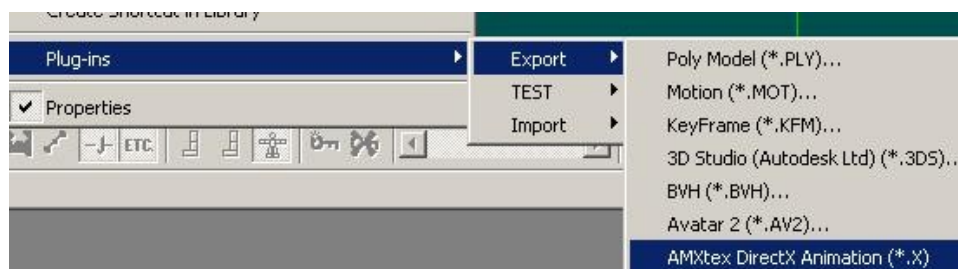
If you view a right-handed model in a left-handed viewer, the model will look like a mirror-image of the original. Any textures will be reversed, etc.

Animation:Master uses a right-handed system. DirectX can use either coordinate system, depending on how the program is set up. You should be aware of the system that your program will use, and export using the appropriate option.

## Exporting Animations from Animation:Master

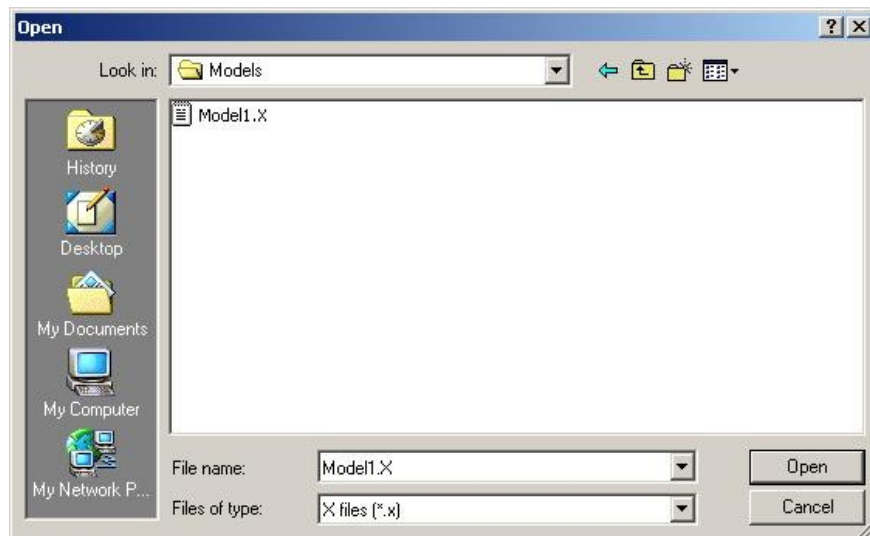
### *How to export animation*

To export animation, you must first export a boned model from Animation:Master to the .X format using AMXtex. Once you've done that, select the appropriate animation, right click, and select Plug-ins -> Export->AMXtex DirectX Animation (\*.X) (See Figure 1).



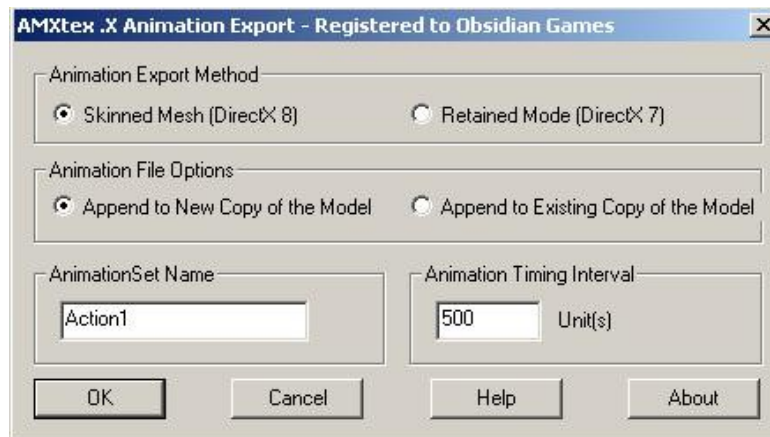
*Figure 1: Select Animation export.*

A dialog (Figure 2) will appear, prompting you to load the X file that the animation will be used with. This should be the .X file that you just exported.



*Figure 2: Select appropriate .X file.*

Next, you'll see the Animation Export dialog (Figure 3).



*Figure 3: Animation Export Dialog*

Select the appropriate options and then select the OK button. Your animation should be exported, available for viewing in the Microsoft MeshViewer utility or other animated .X viewer.

## **Export Options**

### **Animation Export Method**

This option should correspond to the Mesh Export Method used. See the help section on Mesh Export Method for more information.

## **Animation File Options**

This option allows you to place the exported animation where you want it.

Selecting the "Append to new copy of the model" button will make a copy of the .X model, and append the animation to that file. The new copy of the model will have a filename in the format "MODELNAME - ANIMATIONNAME.x". If the target file exists, it will be overwritten.

Selecting the "Append to existing copy of the model" button will append the animation to the existing copy of the .X model.

Keep in mind that if you append an animation to a model that already has an animation attached, that some model viewers will only display the first animation in a .X file. However, newer versions of the Microsoft MeshViewer utility allow you to select which animation you wish to display.

## **AnimationSet Name**

This option allows you to set the AnimationSet name inside the .X file. This is useful if you have an X file that has multiple animations.

## **Animation Timing Interval**

This option allows you to specify the interval between animation keyframes. It is an arbitrary number, not based on any specific unit of time.

Each program which loads X files can use a different value. The Skinned Mesh viewer uses a value of 1 unit = 1 millisecond, where as others may use a value of 1 unit= 1/30 of a second. It's up to the programmer to decide.

## **Registering AMXtex**

### ***Entering Registration/License Information***

Once you have purchased an AMXtex license, you must enter that information into the product before AMXtex will enable all export features.

To do this, launch A:M, open the AMXtex export dialog, and select the "License Info" button.

You will see a window pop up which looks like this:

A screenshot of a Windows-style dialog box titled "AMXtex Registration Information". The dialog has a close button (X) in the top right corner. Inside, it says "Enter your AMXtex registration information below." and "Note: Username is case sensitive." There are three input fields: "Username" with the text "Unregistered" and a format hint "Format: 32 Character Max"; "License Number" with the text "0000-0000" and a format hint "Format: ####-####"; and "License Key" with the text "12345678" and a format hint "Format: #####". Below the License Key field is a checkbox labeled "Show License Key" which is checked. At the bottom are "OK" and "Cancel" buttons.

Enter your Username, License Number, and License Key, exactly as printed in your purchase email, and select “OK”. Your AMXtex export window should now show as being registered to you, and all of the export options will be available.

## AMXtex Support

### *Help and FAQ Files*

First and foremost, a user should always check the Help and FAQ files. The help file that comes with AMXtex (this file) should have answers to many questions. The most recent version can always be found on the web at <http://www.obsidiangames.com/>.

If the question cannot be answered there, please contact support by email.

### *Email Support*

Please email [support@obsidiangames.com](mailto:support@obsidiangames.com) with any support requests, questions, or bug reports.

Be sure to include your name, license number, version of A:M, OS version, and a brief description of the problem. If reporting a bug, please provide a detailed description of the bug, along with files (zipped models, jpg images, etc) to help illustrate the problem.

## AMXtex Frequently Asked Questions (FAQ)

### *What versions of A:M does AMXtex work with?*

AMXtex currently works with the following versions of Animation:Master. While it is typically only

supported on the most recent two versions of A:M, older versions may be provided, but they will be unsupported.

- v13.0 (supported)
- v12.0 (supported)
- v11.1 (not supported)
- v11.0 (not supported)
- v10.5 (not supported)
- v10.0 (not supported)
- v9.51e (not supported)
- v8.5 (not supported)

### ***What's the difference between the unregistered and registered version of AMXtex?***

There are three important differences between the registered and unregistered versions.

- The unregistered version cannot export textures to the model.
- The unregistered version can only export the first 5 frames of any animation.
- The unregistered version cannot export vertex/cp weights.

Other than that, both versions work the same. You're free to use the unregistered version for as long as you wish. There's no automatic expiration, and there are no limits on its use outside of the basic license agreement.

By purchasing AMXtex, you gain the option to export textures, full animations, and weight export. You also support Animation:Master 3rd party programmers, who work to make A:M a more powerful and useful application. You can purchase AMXtex by visiting the [Obsidian Games website](#) and selecting purchase.

### ***What are the limitations of AMXtex?***

There are some limitations with exporting models from Animation:Master to the DirectX .X format. The DirectX .X format is intended to be a fast, real-time format, and as such, does not support all of the features of Animation:Master models. Please keep these issues in mind when modeling and/or animating your models inside A:M. With these limitations in mind, it's best to test your model and animations by exporting them on a regular basis. This can ensure that your final .X model will look as much like the Hash equivalent as possible.

\* AMXtex cannot export multiple textures per polygon. The information is available from Animation:Master, but the .X format is limited to a single texture per polygon.

\* AMXtex cannot export smart skin, or muscle, animations. The DirectX .X format only allows for bone-based animation.

\* Some .X mesh viewers have limits to what they can display. Older versions of DirectX could not

display objects with more than 65535 ( $2^{16}-1$ ) vertices or faces. Newer versions can handle up to  $2^{32}-1$  indices, however, as of this writing, the Microsoft MeshViewer utility appears to have a limit of 65535 ( $2^{16}-1$ ) triangulated polygons when a model includes bones. Any time you export a model with more than 33,768 4-point Hash patches, it's possible to hit this limit, and your model will not load in that utility.

\* DirectX has a strict naming convention for some objects, such as bone names. Acceptable characters are 0-9, A-Z, a-z, and `_`, and names cannot begin with a number. AMXtex will convert all bone names to this format.

\* Bone names are limited to 256 characters, which is the limit set by the Hash SDK.

\* AMXtex animations can only be exported to models that were created by AMXtex.

### ***Why does the percentage bar hang at 100 (or another) percent?***

Some models, especially medium to large textured models, will sometimes cause the percentage complete bar to "hang" at 100 %. This is normal. The initial percentage is calculated using the number of vertices in the model. Textured models will usually require lots of duplicate vertices to be created, which causes the percentage to go beyond the 100 mark.

This is normal, and can be safely ignored. The export will still complete correctly. Don't kill the application, even if it appears to hang.

### ***Why do the exported bitmaps sometimes display incorrectly?***

Some legacy programs use reversed texture coordinates. This causes the bitmaps in an exported .X file to display incorrectly. WildTangent, for instance, reverses the vertical texture coordinate. You can fix this by flipping your texture(s) horizontally and/or vertically.

### ***Can I modify the exported .X file?***

Certainly. AMXtex exports .X models in text format. Anyone with knowledge of the format can open the file in a text editor and change whatever they like.

For a detailed look at the .X model format, please download the Microsoft DirectX Software Development Kit (SDK).

### ***Why does my retained mode mode export incorrectly?***

Some users have experienced issues with exporting models to the DirectX 7 "Retained Mode" model format. Due to the fact that the retained mode format was Microsoft's first attempt at a standard model format, there are some constraints that users should be aware of when exporting.



- **Missing polygons/Holes in the exported .X model** - The retained mode model is a segmented format which ties polygons to bones. Polygons can only be owned by a single bone, so any polygons which have CPs/vertices owned by two different bones will not be exported. You can offset this by increasing the "Patches per Polygon" export setting, which will limit the space between bones, and lessen the holes in the model.

### ***Why does my model look/animate wrong when using the 2005+ versions of the DirectX 9 SDK Mesh Viewer?***

Starting in 2005, the DirectX SDK started shipping with a new Mesh Viewer, called "DirectX Model Viewer". This viewer does not properly display models exported using AMXtex. There is a workaround, but it currently involves using the "Mesh Viewer" utility from older versions of the DirectX 9 SDK. To get your models to work in the new viewer, you will need to obtain a copy of this older viewer, which is available for download from Microsoft's website as part of the October 2004 DirectX 9 SDK.

Once you have that Mesh Viewer utility, do the following:

- 1) Run the "MeshView" program.
- 2) Select File->Open Mesh File and open your file.
- 3) Select something on your model (vertex, polygon, etc)
- 4) Select MeshOps->Validate Mesh. You'll get an Information dialog.
- 5) Select OK from the Information dialog. You may get a dialog that reads "The selected mesh failed D3DXValidMesh, do you want to call D3DXCleanMesh?"
- 6) Select Yes from that dialog if it appears.
- 7) Select File-> Save Mesh As, and pick a new filename.

You're done, the mesh should now look (and animate) correctly in the newer "DirectX Model Viewer".