



MACHSTUDIO PRO

Tutorial Series

Exporting Assets from 3ds Max using fastExport

Lesson 1 Exporting from 3ds Max

This tutorial explains how to tag and export your sets, animations, and cameras from 3ds Max using the fastExport scripts for use in MachStudio™ Pro 1.4 and above. The [Reference](#) section contains examples for exporting parent nodes, instancing, and using mesh flags.

NOTE: The fastExport scripts only work for 3DS Max version 2010 and 2011. They will not work on earlier versions.

Downloading and Installing the Exporter and fastExport Scripts

1. Click [Download Max fastExport](#) to download the most current version of the Max exporter and fastExport scripts.

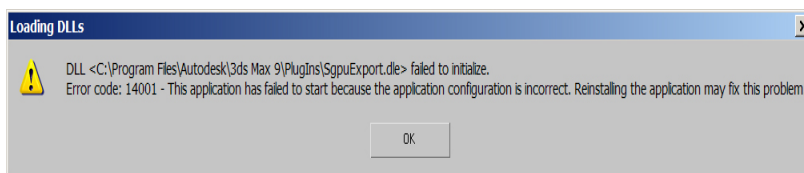
Note: MachStudio Pro does not need to be installed on the machine where you install the exporter.

2. After downloading, unzip the file.
3. Double-click the **vcredist** folder, choose between **32bit** and **64bit** folders, and double-click the corresponding application file. 32bit users should double-click **vcredist_x86.exe** and 64bit users should double-click **vcredist_x64.exe**.

Note: If you have not installed this before, follow all instructions to install. If you have installed this before, select the **Repair** option.

4. There are four folders: **Max 9**, **Max2008**, **Max2009**, **Max2010**, and **Max2011**. Open the folder that corresponds with the version of 3ds Max you are running.
5. Double-click the **Windows** folder and choose between the **32bit** or **64bit** folders, depending on the version of 3ds Max you are running.
6. Copy the **SgpuExport.dle** file to your Max Plug-ins directory.

Note: When you start 3ds Max and see the following error:



- a. Close down max, install the following software from Microsoft and re-open Max:
Microsoft Visual C++ 2008 SP1 Redistributable Package (x86)

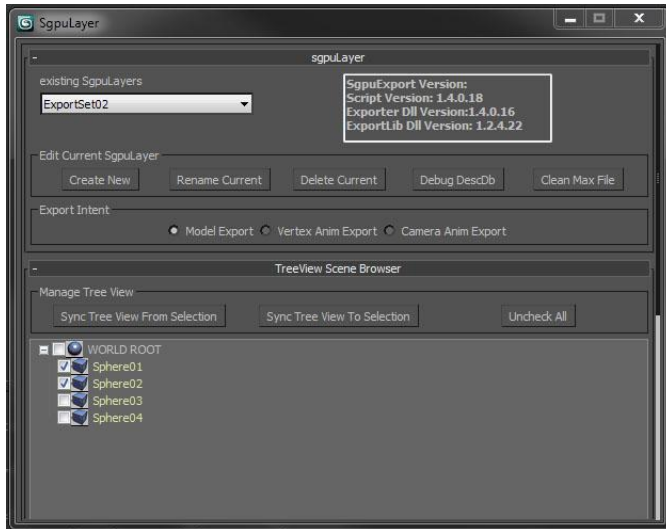
<http://www.microsoft.com/downloads/details.aspx?familyid=A5C84275-3B97-4AB7-A40D-3802B2AF5FC2&displaylang=en>

7. Double-click the **scripts** folder and copy the contents of the **Startup** subdirectory to **MaxInstallDir/Startup**.
8. Copy the **SgpuScripts** directory to **MaxInstallDir/Scripts**.

The fastExporter has been installed and can be accessed from the **MAXScript** menu.

fastExport Interface - Tagging

Sets, objects and animations can be tagged using the **SgpuLayer** interface.



Existing SgpuLayers: This pull-down menu lists all tagged sets, animations, and cameras.

Create New: Use this button to create a new tagged set, animation, or camera.

Rename Current: Use this to rename anything that has been tagged.

Delete Current: Use this to delete selected sets, animations, or cameras that have been tagged. Use the Existing SgpuLayers pull down menu to select the tagged item to be deleted and click **Delete Current**.

Clean Max File: Deletes all sets, animations, or cameras that have been tagged.

Model Export: Use this option for static objects

Vertex Anim Export: Select this for animated objects.

Camera Anim Export: Select this for cameras.

Sync Tree View From Selection: Use this option to sync the tree view with what you've manually selected.

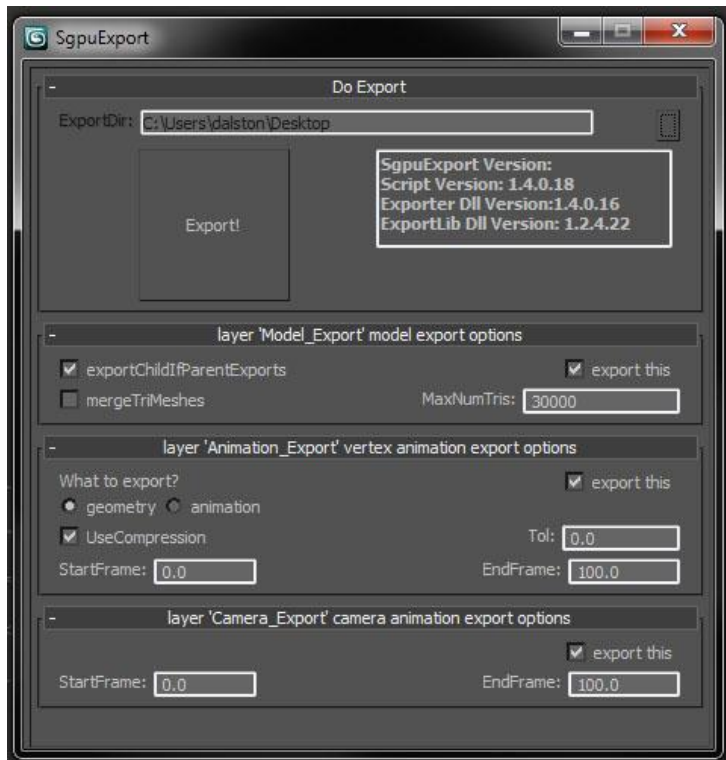
Sync Tree View to Selection: Use this to sync the tree view from what you've selected under the World Root.

Uncheck All: This will deselect everything from the tagged model.

Adding a subdivision flag: A subdivision flag can be added by right clicking on the world root in the Tree View Scene Browser and selecting Export as subdiv.

fastExport Interface - Exporting

Sets, animations, and cameras can be exported using the **Do Export** interface.



Export Dir: Allows for selection of export directory.

Model Export Options

exportChildifParentExports: All nodes under a selected parent node will be exported

MergeTriMeshes: Merges the geometry that shares the same shader together. This will improve performance in MachStudio Pro.

Vertex Animation Export Options

Exporting animation is a two step process as you will first need to export the geometry and then export the animation. Use the geometry and animation options for exporting. By default, the vertex animation is compressed in a lossless manner, which should result in a compression gain of 0.25. (i.e. the final file produced would be only 0.25 of the size of the uncompressed version).

With Vertex Animation compression turned on, you can also suggest a lossy compression by giving an appropriate tolerance value. If the tolerance value is greater than 0.0, then a lossy compression method is used. An acceptable lossy compression depends on the scale of the object.

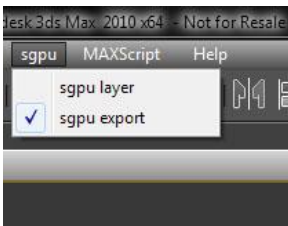
Camera Animation Export Options

Use the **start frame** and **end frame** options to enter a specific time frame.

Step 1: Tagging Sets, Animations, and Cameras using the fastExporter

Sets, animations, and cameras are tagged through the sgpu layer interface.

1. Select the item intended for tagging (static objects, animations, or cameras)
2. Select **MAX Script > sgpu layer**.



3. Click **Create New**.



4. Enter a **name**.
5. Select an **Export Intent**: Model Export (static objects), Vertex Anim Export (objects that animation), or Camera Anim Export (Cameras).

Note: For models, a subdivision flag can be added by right clicking on the world root in the Tree View Scene Browser and selecting Export as subdiv.

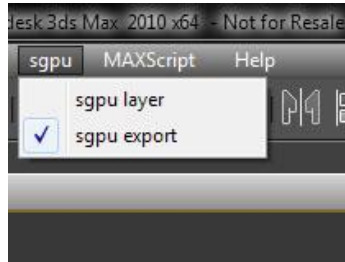
6. Click **OK**.

Continue tagging until you are ready to export.

Step 2: Exporting Sets, Animations, and Cameras

use the sgpu export interface to export Sets, Animation, and Cameras.

1. Select **MAX Script > Sgpu Export**.

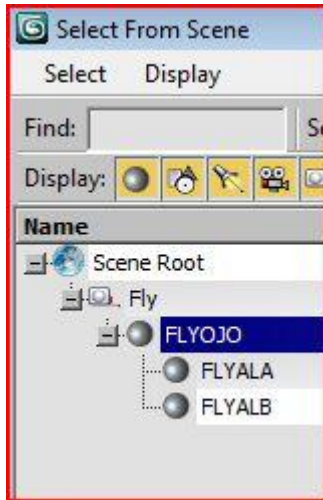


2. In the **ExportDir** field, select a directory where all tagged selections will be exported.
3. Select options for sets, animations, and cameras.
 - a. For models, choose either **exportChildifParentExports** or **mergeTrimeshes**.
 - b. For animation, exporting is a two step process – export the geometry for animation and then export the animation. Compression options can also be selected. See the [fastExport Interface - Exporting](#) section above for more details regarding compression options.
 - c. For cameras, frame options can be selected.
4. Click the **Export!** button.

The tagged options will be exported to the selected export directory. Sets (static and those with animation) will be exported as GXBs. Animation will be exported as GABs and cameras as CAMs.

Reference

Example – Export Child under sel Parent



Exporting Models

If you choose to export the parent node, all non-hidden children are automatically exported.

In this example, the **FLYOJO** node is selected for export. If the **Export Child Under sel parent** is checked in the exporter then both **FLYALA** and **FLYALB** will be automatically exported.

If the **Export Child under Parent** is unchecked, then only the parent node **FLYOJO** will be exported.

In the case that you wanted to export **FLYOJO** and **FLYALB** but not **FLYALA**, you would need to uncheck **Export Child under Parent**.

Exporting Vertex Animation and Cameras

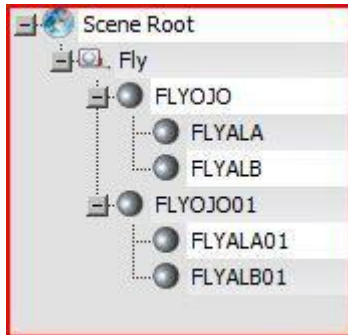
If you select **FLYOJO** and export, only **FLYOJO** would export and since it's exported in world space, **FLYOJO** would inherit the transforms of its parents.

Example – Instancing

The SGPU exporter recognizes instances in the max file and incorporates that information into the .GXB file, so that most of the geometry data can be shared among the common instances.

Note: The instancing feature is implemented only if a scene is exported with the “export model” intent, (i.e. for static non- animated meshes).

The following example explains this process.



In the example above, **FLYOJO01** is an instance of **FLYOJO**. The SGPU exporter will export these instances only if the sub hierarchies share the same tree-structure and corresponding nodes in the instanced sub hierarchy have the same material, sub-material face mapping. If the SGPU exporter cannot export the instances, they will be de-instanced or exported as individual disconnected objects, losing the instancing information.

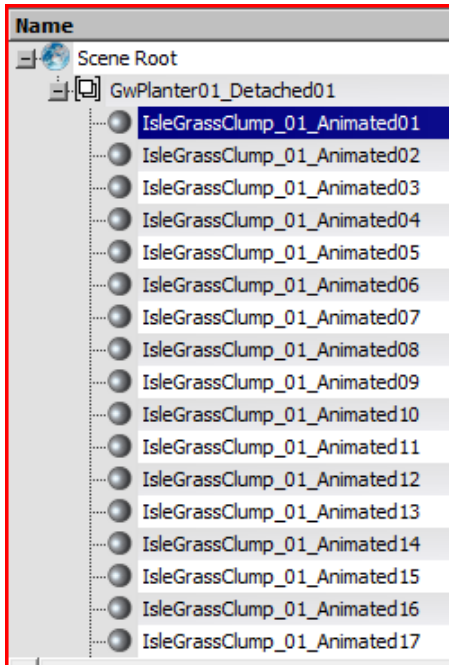
If you altered the sub material-face mapping of mesh **FLYOJO**, assigned another material to **FLYOJO**, or added another child to **FLYOJO**, then the exporter cannot export the instancing of the nodes **FLYOJO** and **FLYOJO01**. They will be de-instanced in the .GXB. However, the children nodes **FLYALA** and **FLYALA01** will be instances and **FLYALB** and **FLYALB01** will be instances.

Among a collection of instances, the exporter will try to find subsets where the material face mapping is consistent, and export the subsets as separate instances.

For example, in the hierarchy below, all the objects named **IsleGrassClump_01_Animatedxx** are instanced. Assuming the material face mapping of objects named from **IsleGrassClump_01_Animated01** to **IsleGrassClump_01_Animated10** are of one kind and that of objects from **IsleGrassClump_01_Animated11** to **IsleGrassClump_01_Animated17** are of another kind. When exported, this will result in two sets of instances.

Mesh flags are discussed in the next section. Mesh flags for a set of instances will be based on the leading instance. A leading instance is the instance that comes first, in the “select by name” browser in max.

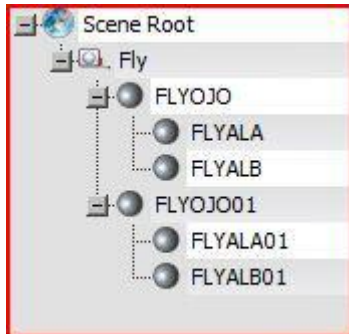
In the example below, the leading instances are **IsleGrassClump_01_Animated01** and **IsleGrassClump_01_Animated11**



The above example shows part of the scene hierarchy, where the max node **IsleGrassClump_01_Animated01** was instanced into 1800 objects. This method of instancing can cause significant reduction of memory footprint in MSP.

Example – Material Merging

When exporting a large scene, if meshes are merged with the same material into a single mesh, the resulting gxb file will behave faster when loaded in MSP.



Consider exporting the scene above, let's make the following assumptions.

--FLYALA, FLYALB, FLYOJO all contain valid meshes.

--FLYOJO is assigned a multimaterial with two sub materials "yellow" and "green".

--FLYALA is assigned a standard material called "pink"

--FLYALB is assigned a standard material called "skyBlue".

--The sub-hierarchies rooted at FLYOJO and FLYOJO01 are instances.

If exported with the material merge option, all parts of meshes or instances of meshes with the same material will be merged into one mesh.

The exported file will contain 4 meshes.

-mesh_1 will contain the "yellow" part of FLYOJO and FLYOJO01, exported in world space.

-mesh_2 will contain the "green" part of FLYOJO and FLYOJO01 exported in world space.

-mesh_3 will contain FLYALA and FLYALA01

-mesh_4 will contain FLYALB and FLYALB01

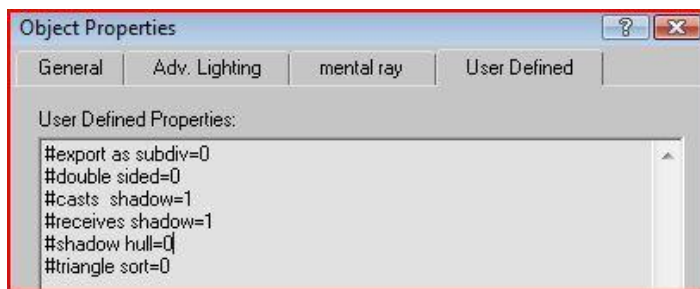
The **Max Num Tris for each Merged Mesh** field allows you to enter a restriction on the maximum number of triangles a merged mesh can contain. If that limit is reached, more than one merged meshes are used for a single material.

Currently, the merging will only affect meshes which are not tagged with "export as subdiv". If any of the meshes are tagged "export as subdiv" they are left undisturbed.

User Defined Mesh Flags

The MachStudio™ Pro pipeline accepts the following mesh flags through user defined node properties. The default values are shown along with each supported user property. Please note the # sign in front of each property.

- double sided = By default, this value is set to false if this property is absent.
- casts shadow = By default, this value is set to true if this property is absent.
- receives shadow = By default, this value is set to true if this property is absent.
- shadow hull = By default, this value is set to false if this property is absent.
- triangle sort = By default, this value is set to false if the property is absent.
- cloth = By default, this value is set to false if the property is absent.
- export as subdiv = This flags the model for catmull-clarke subdivision in MSP. You do not need to add turbo smooth to your models before exporting with this. Currently, you get one level of subdiv. By default, this value is set to false if the property is absent.
- visible anim = By default, this value is set to false if the property is absent.
- low res = This flags your model as low res or high res. If you enter #lowres=1, the model will only display in low resolution. If you enter #lowres=0 the model only displays in high resolution. By default, this value is set to 0 if the property is absent.



Modifiers for Vertex Animation

The following example explains the benefits of using a point cache modifier during the exporting process.

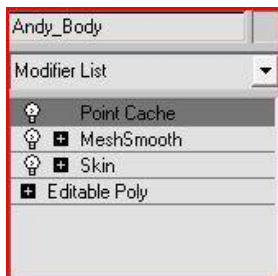
Example

The following modifier stack for an animated object has a skin modifier added to the vertex skinning:



Currently, the exporter ignores any modifier on top of the skin or physique modifier. In this case the “MeshSmooth” modifier is ignored.

In order to have the exporter take all modifiers add a point cache modifier at the top and bake the point cache before exporting, as shown below.



Instead of using smoothing modifiers, which results in geometry explosion, the user is encouraged to use the export flag “export as subdiv”, which makes use of hardware accelerated subdivision in MSP.

Using Vertex Animation Compression

As discussed in the steps above, exporting animation involves two steps, exporting the geometry for animation and exporting the animation itself. By default, the vertex animation is compressed in a lossless manner, which should result in a compression gain of 0.25. (i.e. the final file produced would be only 0.25 of the size of the uncompressed version).

With Vertex Animation compression turned on, you can also suggest a lossy compression by giving an appropriate tolerance value. If the tolerance value is greater than 0.0, then a lossy compression method is used. An acceptable lossy compression depends on the scale of an object.

Compression can be turned off by deselecting the **Use Compression for Vertex Anim** option.

Legal Notice

All asset files including, but not limited to 3D models, animations, textures and materials, are the property of studio|gpu and provided solely for the purpose of user training. Any commercial use of these assets is strictly prohibited.