



MACHSTUDIO PRO

Tutorial Series

Exporting Assets from 3ds Max

Updated for exporter version 1.2.0.7

Lesson 1 Exporting Assets from 3ds Max

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

Downloading and Installing the Exporter

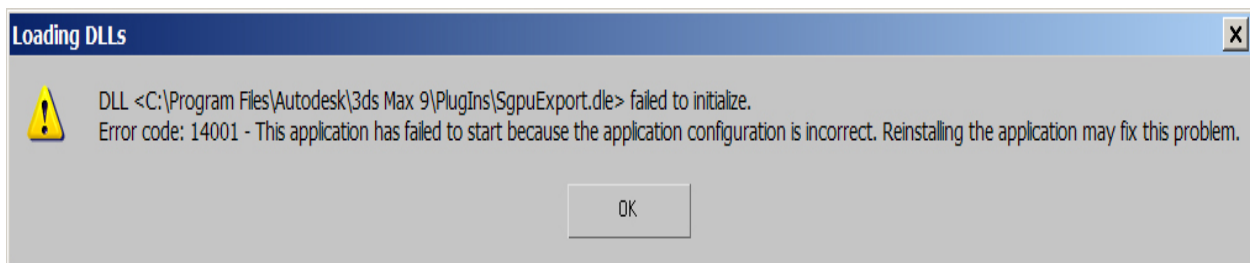
1. Click [Download Max Exporter](#) to download the most current version of the Max Exporter.

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

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

The Exporter has been installed and can be accessed from the File Menu.

Note: When you start 3ds Max, if you are getting an error dialog as below:



1. 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>

If you are working with a 64 bit version of 3ds max, use this instead:

<http://www.microsoft.com/downloads/details.aspx?familyid=BA9257CA-337F-4B40-8C14-157CFDFEE4E&displaylang=en>

Step 1: Exporting a Model using the exporter

Select this export intent to export a hierarchy of nodes which are not animating.

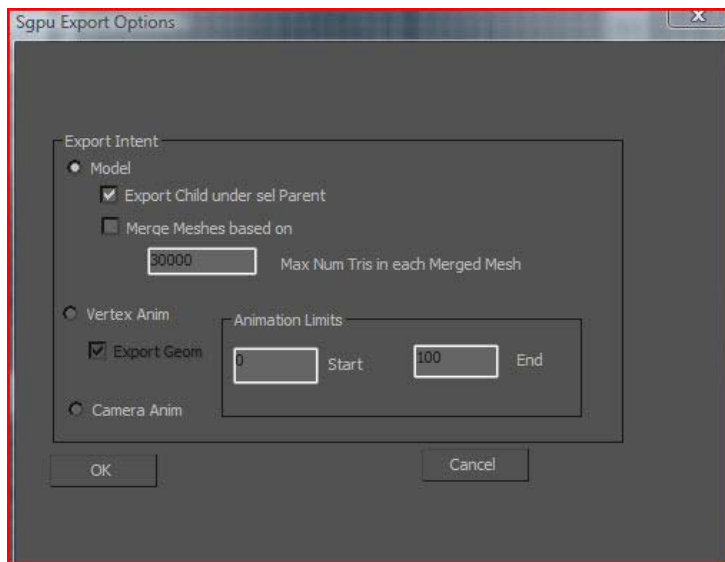
If you want to assign user definite mesh flags during the exporter process, see the [User Defined Mesh Flags](#) section.

1. Select the geometry of the set you'd like to export.
2. Select **File > Export Selected**.
3. Enter a name and select **Sgpu File (*.GXB, *.CAM, *.GAB)** from the pull down menu.

The specific extension (.gxb, .cam, .gab) will be automatically determined. You only have to supply the name of the file. If the export intent is 'model export', the .gxb extension is used.

4. Click **Save**.

This launches the SGPU Export Options dialog box:



Note: If you want all nodes under a selected parent node to be exported automatically, select **Export Child under sel Parent**. If this choice is unchecked, then children have to be explicitly selected for export. See the [Example – Export Child under sel Parent](#) section for more information.

5. Under **Export Intent**, select the **Model** option.
6. Keep **Export Child Under sel Parent** checked.
7. Hit **OK**.

Step 2: Exporting Animation

Use this export intent to export baked vertex animation of hierarchical nodes.

There are two parts to the exporting animation process. For both, Vertex animations and camera export animations, only the selected nodes are exported; and they are exported at world space. See the [Example – Export Child under sel Parent](#) section for more information.

Part 1

1. Select all the geometry of the animated object that you'd like to export.
2. Select **File > Export Selected**.
3. Enter a name and select **Sgpu File (*.GXB, *.CAM, *.GAB)** from the pull down menu.

The specific extension (.gxb, .cam, .gab) will be automatically determined. You only have to supply the name of the file. If the export intent is 'vertex animation export', and the export geom option is checked, the .gxb extension is used.

4. Click **Save**. This launches the SGPU Export Options dialog box.
5. Select **Vertex Anim**.
6. Check the **Export Geom** option.
7. Hit **OK**.

This produces a **.GXB** file, which contains the geometry for the animated object.

Part 2

1. Repeat Steps **1-5**.

The specific extension (.gxb, .cam, .gab) will be automatically determined. You only have to supply the name of the file. If the export intent is 'vertex animation export', and the export geom option is unchecked, the .gab extension is used.

2. Uncheck the **Export Geom** option.
3. Fill in the animation start and end frames that you need to export.
4. Hit **OK**.

This produces a **.GAB** file, which contains the animation.

Step 3: Exporting a Camera

1. Select the camera to be exported.
Note: You can only export one camera at a time.
2. Select **File > Export Selected**.
3. Select **Sgpu File (*.GXB, *.CAM, *.GAB)** from the pull down menu.

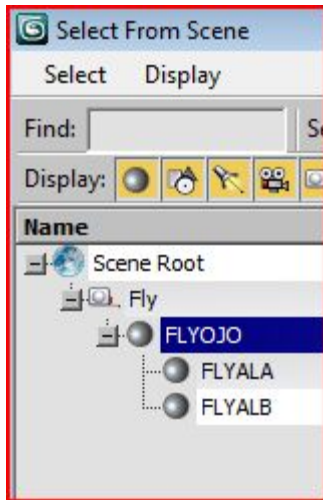
The specific extension (.gxb, .cam, .gab) will be automatically determined. You only have to supply the name of the file. If the export intent is 'camera animation export', and the export geom option is unchecked, the .cam extension is used.

4. Click **Save**.
5. In the export options dialog, select **Camera Anim**.
6. Fill in the animation start and end frames that you need to export.
7. Hit **OK**.

This will produce a **.CAM** file for the camera's animation.

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

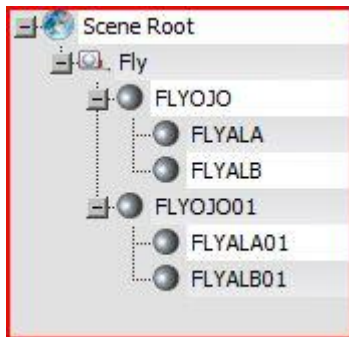
If you select **FLYOJO** and export, only **FLYOJO** would export and since it's exported in world space, **FLYOJO** would inherit the transforms of its parents.

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.

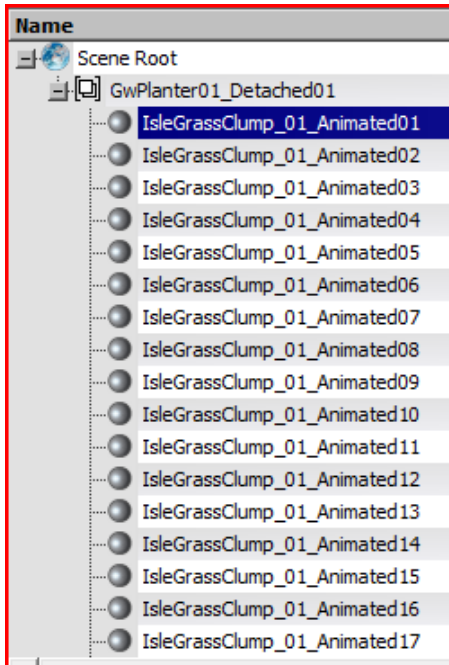
If you altered the sub material-face mapping of mesh **FLYOJO**, assigned another material to **FLYOJO**, or added another child to **FLYOJO**, then the exporter cannot export the instancing of the nodes **FLYOJO** and **FLYOJO01**. They will be de-instanced in the .GXB. However, the children nodes **FLYALA**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

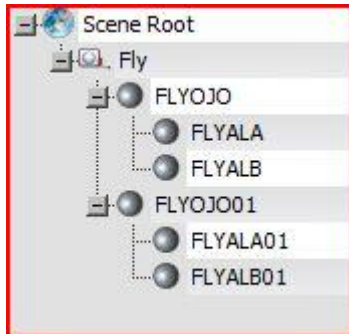
In the example below, the leading instances are **IsleGrassClump_01_Animated01** and **IsleGrassClump_01_Animated11**



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

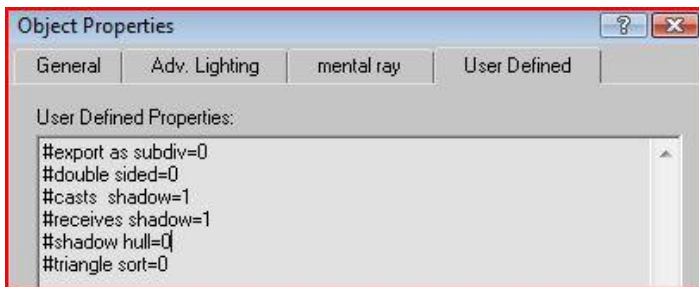
The **Max Num Tris For each Merged Mesh** field allows you to enter a restriction on the maximum number of triangles a merged mesh can contain. If that limit is reached, more than one merged meshes are used for a single material.

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.

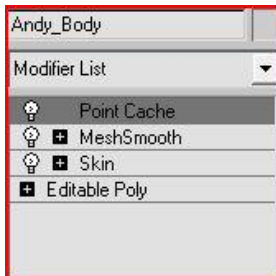
Example

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.

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.