

How to export to glTF for WebGL

Most of the time, WebGL platforms are a “plug & play” type of system. You load a single webGL model and it loads everything linked together, all finished. Textures, materials, animation, etc. There is little to no tweaking afterwards. How you export your glTF/gIB from your 3D tool, is how it will show up in the webGL platform.

When exporting to **glTF**. Your exporter should produce a folder containing the following:

ObjectName.bin	TextureFiles_C.png	TextureFiles_ARM.png
ObjectName.glTF	TextureFiles_N.png	TextureFile_AO.png

This folder is then loaded into the platform and should load your model with everything linked together. Textures are pre-assigned to materials, and materials are pre-assigned to their triangles and meshes.

When exporting or packing to **glB**. You will only have a single file: ObjectName.glB

Both the glTF folder, or the single glB file. Can be loaded into <https://sandbox.babylonjs.com/> or <https://gltf-viewer.donmccurdy.com/> for testing. (Not sure how safe these are regarding NDA's)

UV channels

Two UV channels are needed:

- 1 - Color/texture map. You can scale the UV islands as you wish to get the desired material tiling.
- 2 - AO map. This one cannot have overlapping UV's. and must be within 0 to 1 UV space.

Tiling materials

Most often we do not have access to the materials or shaders. Which means we cannot put in tiling values. Instead you need to do tiling in the UV map. Make UV islands bigger or smaller.

Textures

To reduce file sizes and so improve end-user experience, textures should be in JPG file format. Except for transparent textures, which use transparent PNG's.

For the model's AO. Create a single AO file in linear color space:

T_AO_01.jpg

For each (tile-able) material. Create the following texture sets:

T_Map_C.jpg = Albedo (.png is it needs transparency)

T_Map_N.jpg = Normal

T_Map_ARM.jpg = AO, Rough, Metal. (This AO is for the AO of the material, not the AO of the model)

Materials

Have all your textures assigned to their respected slots in your materials. And have all materials correctly assigned to your mesh. Make sure you name your materials neatly. Depending on your 3D application, you

might have to choose a specific glTF material/shader for it to export correctly into the glTF/gIB format. Make sure you assign your textures to the correct UV channel. Eg, the AO map should use the second UV channel.

Transparency is supported using default PNG transparency in the _C (color) map.

Test your glTF and/or gIB exports to see if the textures are correctly exported. For example in Modo, if the shaders/materials used are not the glTF shaders. The exporter will not see the textures and so will not export them.

Animation

- Set your scene or time rate to 30 frames per seconds.
- Use animation tracks/clips/actions for all animations with easy understandable names. In code, these animation names are “called” to play and build interaction.
- Make sure the first frame in your scene is always the “default” pose or position. Call it “DefaultPose”.
- If animations are added. Add additional animation tracks for static states. For example, if you have a car door that needs to open and close. Add a “close” and “open” animation. But also add a “static Closed” and “static Opened” animation.



Links & Resources

Babylon WebGL online previewer and exporter, exporter plugins and knowledge:

<https://www.babylonjs.com/>

<https://sandbox.babylonjs.com/>

<https://github.com/BabylonJS/Exporters/>

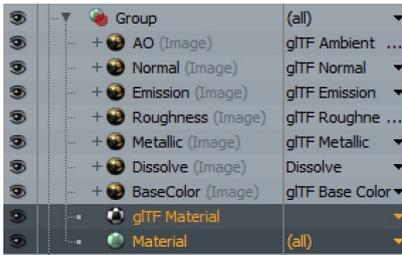
Online previewer:

<https://gltf-viewer.donmccurdy.com/>

Modo glTF information:

https://learn.foundry.com/modo/content/help/pages/shading_lighting/shader_items/glTF.html

<https://community.foundry.com/discuss/post/1131857>



Debugging & Solutions

Textures or UV's not working correctly

Add a temporary texture to your materials before you export. So the glTF and WebGL platform is “forced” to understand that a given material slot uses a texture. Then in the final webGL platform, on the developer side, replace it with the texture they need.

Transparency not working

- Confirm the material and shader setup within your authoring tool is correct. Eg, in Modo all textures need to be loaded into the glTF slots. Except for the transparent texture, which needs to be loaded into the Dissolve slot.
- Use the [Babylon online tool](#) to export to glB. You can set up materials in their tool.

AO issues

Make sure the AO uses the second UV channel.
And that the AO is disabled on the decal textures