

Using glTF Mesh in OpenSim

Posted on [December 1, 2025](#) by [bat](#)

OSCC-2025 Presentation

OpenSim has supported glTF mesh import since mid 2025. Since the internal mesh format is identical to that used for Collada mesh imports, glTF mesh import “just works” without OpenSim server side change.



This presentation describes some of the lessons learned and experience gained while testing glTF mesh import to Second Life and OpenSim and tips for ways to get the best out of using glTF mesh in OpenSim.



History

3D Mesh objects have been able to be used in Second Life using Collada as the import format since 2011 [1][2]. The meshes are converted to an internal mesh format known as llMesh. But Collada format is now old and is becoming unsupported in tools commonly used by creators. Graphics Language Transmission Format (glTF) mesh import to Second Life has been possible since mid 2024 with a public release of the Second Life Viewer in mid 2025 [3]. Initially the glTF meshes are converted internally to the very same mesh format used for Collada, i.e. llMesh. Restrictions on what can and cannot currently be imported are due to that common mesh format usage. The aim stated by Linden Lab for Second Life is that eventually a wider range of glTF models and extensions supported by the Khronos Group standards [4] will be supported. glTF often uses Physically Based Rendering (PBR) materials for texturing and those also are supported in Second Life. OpenSim has supported

glTF mesh import since a suitable viewer became available. Firestorm version 7.2.0 and onwards became available from mid 2025 allows for such import

glTF Mesh Import Workflow

1. Obtain or create 3D model.
2. Tidy the model up in a 3D modeller such as Blender.
3. Export to glTF (usually as .glb incorporating all textures and materials).
4. *Check the model in a glTF Model Viewer.*
5. *Transform (usually to simplify) the glTF model.*
6. *Check the model again in a glTF Model Viewer. Try to remain under the mesh uploader limitations where possible.*
7. *Import the transformed glTF mesh back into a 3D modeller such as Blender to make any changes you wish and if necessary to split the model into parts that can be separately imported to allow for scale 1.0 or as a single linkset in inventory after upload.*
8. *Export the final glTF mesh.*
9. Import to Second Life or OpenSim (include textures).
10. Import again into the PBR materials uploader if you want the PBR materials in the model. But currently they must be applied manually after upload. This will change when an integrated mechanism is provided as planned.

Of course, steps 4 to 8 are not necessary for simple models.

Current Limitations on glTF Mesh

The current mesh uploader has the following limits due to internal llMesh restrictions:

- Mesh parts with more than 65,534 vertices are split by the mesh uploader into multiple “faces”. If object goes over 8 faces, extra faces become ‘submodels’ and those have limits.
- Any mesh part cannot have more than 8 materials. Sub-meshes with more than 8 materials will be split into more meshes in a future build.
- The mesh uploader uses a viewer debug setting of ImporterModelLimit (default 768) as a limit of the number of mesh parts. There cannot be more than 4,000 mesh parts (called instances in the mesh loader stats) in a model.
- Models must be less than 52,428,800 bytes in total.
- There must be less than 256 mesh parts (instances) in a model or the model will upload without linking into one inventory item which can make it difficult to edit, move or delete when rezzed in world.
- Mesh part limits (in metres) – Min. <0.01, 0.01, 0.01> (parts will be set to this limit if smaller) Second Life Max. <64.0, 64.0, 64.0> OpenSim Max. <256.0, 256.0, 256.0>
- Only one glTF scene is loaded at present – the default scene or first scene if no default.
- Rigged meshes cannot have more than four joint weights per vertex. Rigged meshes must be rigged in accordance with with one of the Second Life skeletons[6].

Tips to successfully use glTF Models in OpenSim

- Obtain or create your model and make sure to keep the creation safely even if you make versions with simplifications or splitting into parts. Export original .glb model from there.
- Always check that the model is within the limitations specified. If there are too many mesh parts the item on import will be a potentially large number of separate coalesced objects in inventory and after rezzing can be difficult to manage or delete.
- Use glTF Viewers to make sure the glTF mesh appears as you wish and to look at basic statistics of number of meshes, number of triangles, etc.
- Use glTF Transformation tools to simplify the meshes, weld common material parts together where feasible, etc.
- Check the overall size of any glTF mesh and if larger than the limitation, consider splitting it into parts.

glTf Model Viewers

Khronos Group glTF Sample Viewer [7] can be used to view the glTF models and in an “Advanced Mode” tab to examine statistics such mesh parts count, triangle count and counts for materials. Also available is the ModelViewer.dev glTF/GLB Viewer [8] which uses “PBR Neutral” lighting with a simple white background which can be useful for photo capture of the models.

glTF Import Differences between Second Life and OpenSim

The maximum mesh extent in Second Life is 64m, whereas in OpenSim it is just under 256m. E.g. RGU Oil Rig (scale 1.0 possible in OpenSim) and 2001 Space Station (scaler 0.6 possible in OpenSim as it is still larger than 256m across).

Mesh Optimisation Prior to Import – glTF Transformation Tools

glb.babylonpress.org [9] and gltf.report [10] are examples of “glTF Transformation” tools which can simplify the glTF prior to upload to make it more suitable and come under some of the limitations to avoid things like model splitting failing to import as a single linkset which would mean many separate objects when an item is rezzed from inventory..

These are essential for complex models. Keep the original model in the 3D modeller of choice, e.g. Blender, but export to glTF and then transform that prior to upload into Second Life or OpenSimulator.

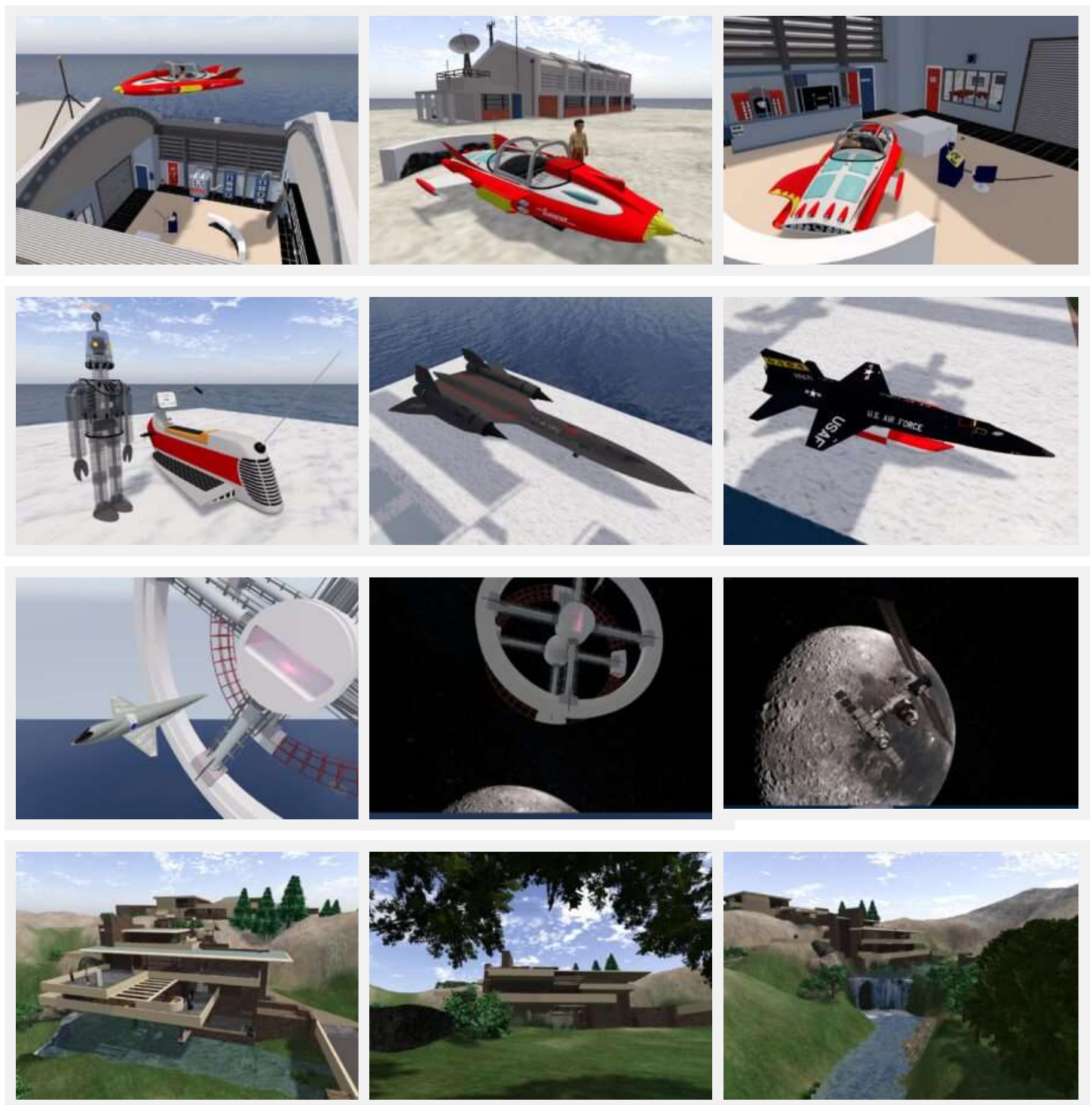
Keep texture format as original as the tools usually convert to *webp* image format otherwise.

You can adjust other settings, but most defaults will work fine. But even though some of the tools are based on the same underlying code library [11] they do have different defaults which can be worth exploring to achieve your purpose when the mesh is uploaded.

glTF Mesh Testing in OpenSim

Test meshes were as used in Second Life mentioned in the blog post above, and described again below. They are usually originated in a range of 3D creation tools such as Studio Max, Cinema4D, etc or downloaded from sites such as Khronos glTF Sample Assets [12], NASA mesh models [13], Sketchup 3D Warehouse [14], etc. and then converted to Blender. From Blender they are exported to glTF (.glb) and then cleaned up and number of mesh parts reduced to less than 256 parts so they import as one linked inventory object) using glTF transformation tools such as <https://glb.babylonpress.org> and <https://gltf.report>.





glTF Rigged Mesh Avatar Testing in OpenSim

Rigged mesh avatars can also be uploaded in glTF format in the same way that Collada rigged mesh works. A version of the Silent Running Drone#1 – Dewey – rigged by Fred Beckhusen and saved as glTF rigged mesh was also tested. Ready Player Me avatars cannot be uploaded in their native .glb format since the skeleton/armature differs between that used by Ready Player Me and SecondLife/OpenSim. Conversion first is needed via a tool such as Onigiri [15].



OAR Converter

OAR Converter [16] is a tool to take OpenSim Archive (OAR) Files and convert them to Collada or OBJ 3d mesh formats for use in external tools and platforms such as 3D modellers like Blender, or game and app development platforms like Unity [17] and Unreal Engine [18], etc. It is now also able to convert to glTF which opens up interesting opportunities including the possibility of reimporting modified content back into OpenSim [18].

References

- [1] Khronos Group (2010) “Second Life chooses COLLADA as the official Mesh import format.” The Khronos Group Inc. September 21, 2010.
<https://www.khronos.org/news/permalink/second-life-chooses-collada-as-the-official-mesh-import-format>
- [2] Search Engine Watch (2011) “Mesh Gets a Release Timeline for Second Life.” Search Engine Watch, June 18, 2011.
<https://www.searchenginewatch.com/2011/06/18/mesh-gets-a-release-timeline-for-second-life>
- [3] Second Life (2025) “Second Life Release Notes for Viewer version 7.2.0.16356654553 – glTF Mesh Import”, July 23, 2025.
<https://releasenotes.secondlife.com/viewer/7.2.0.16356654553.html>
- [4] Khronos Group (2025) “GitHub – glTF”. (Accessed 3-Oct-2025).
<https://github.com/KhronosGroup/glTF>
- [5] Firestorm Viewer (2025) “Firestorm Beta 7.2.0 and some stats”, August 10, 2025. (Accessed 3-Oct-2025).
<https://www.firestormviewer.org/firestorm-beta-7-2-0-and-some-stats/>
- [6] Second Life (2025) “Second Life – Creator Resources”. (Accessed 3-Oct-2025).
<https://github.com/secondlife/creator-resources>
- [7] Khronos Group (2025) “glTF Sample Viewer”. (Accessed 3-Oct-2025).
<https://github.khronos.org/glTF-Sample-Viewer-Release/>
- [8] Modelviewer.dev (2025) “ModelViewer.dev glTF/GLB Viewer”. (Accessed 3-Oct-2025).
<https://modelviewer.dev/editor/>

[9] glb.babylonpress.org – see Help page for details. (Accessed 3-Oct-2025). <https://glb.babylonpress.org>

[10] gltf.report (2025). gltf.report. (Accessed 3-Oct-2025).
<https://gltf.report>

[11] GLTF-Transform – glTF 2.0 SDK for JavaScript and TypeScript, on Web and Node.js.
<https://gltf-transform.dev/>

[12] Khronos Group (2025) glTF Sample Assets. (Accessed 3-Oct-2025).
<https://github.com/KhronosGroup/glTF-Sample-Assets>

[13] NASA (2025) 3D Resources: A repository of 3D models, 3D printable models, and textures. (Accessed 3-Oct-2025).
<https://science.nasa.gov/3d-resources/>

[14] Sketchup (2025) Sketchup 3D Warehouse. (Accessed 3-Oct-2025).
<https://3dwarehouse.sketchup.com/>

[15] Onigiri [12] “GitHub – Onigiri 4”. (Accessed 3-Oct-2025).
<https://github.com/aiaustin/Onigiri>

See also <https://blog.inf.ed.ac.uk/atate/2023/03/21/ready-player-me-avatars-in-second-life-and-opensim/>

[16] Unity (2025) “Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine”. (Accessed 3-Oct-2025).
<https://unity.com/>

[17] Epic Games (2025) “Unreal Engine”. (Accessed 3-Oct-2025).
<https://www.unrealengine.com/>

[18] Iseki, F and Tate, A. (2025) “OAR Converter with glTF”. (Accessed 3-Oct-2025).
<https://blog.inf.ed.ac.uk/atate/2024/09/23/oar-converter-with-gltf/>



This entry was posted in [3D](#), [OpenSim](#) and tagged [glTF](#), [OpenSim](#). Bookmark the [permalink](#).

Austin Tate's Blog

Proudly powered by [WordPress](#).