

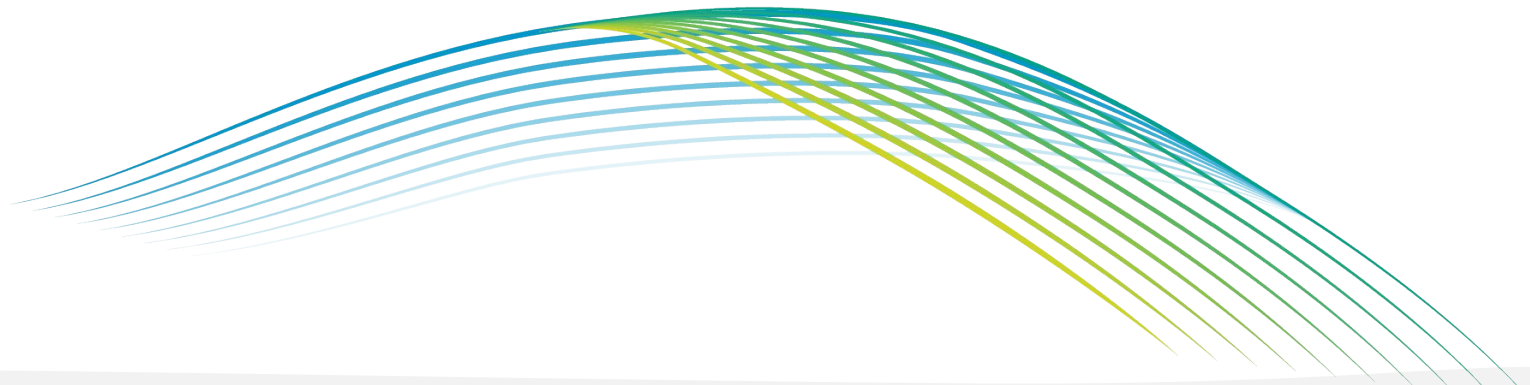


Metaverse
STANDARDS FORUM™



SIGGRAPH 2026
Los Angeles 19–23 JUL

3D Assets: New features, Tooling, and What Comes Next



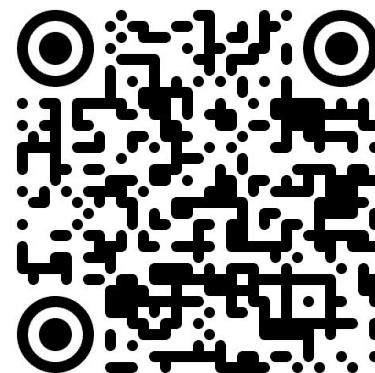


glTF - Character and Avatar Extensions

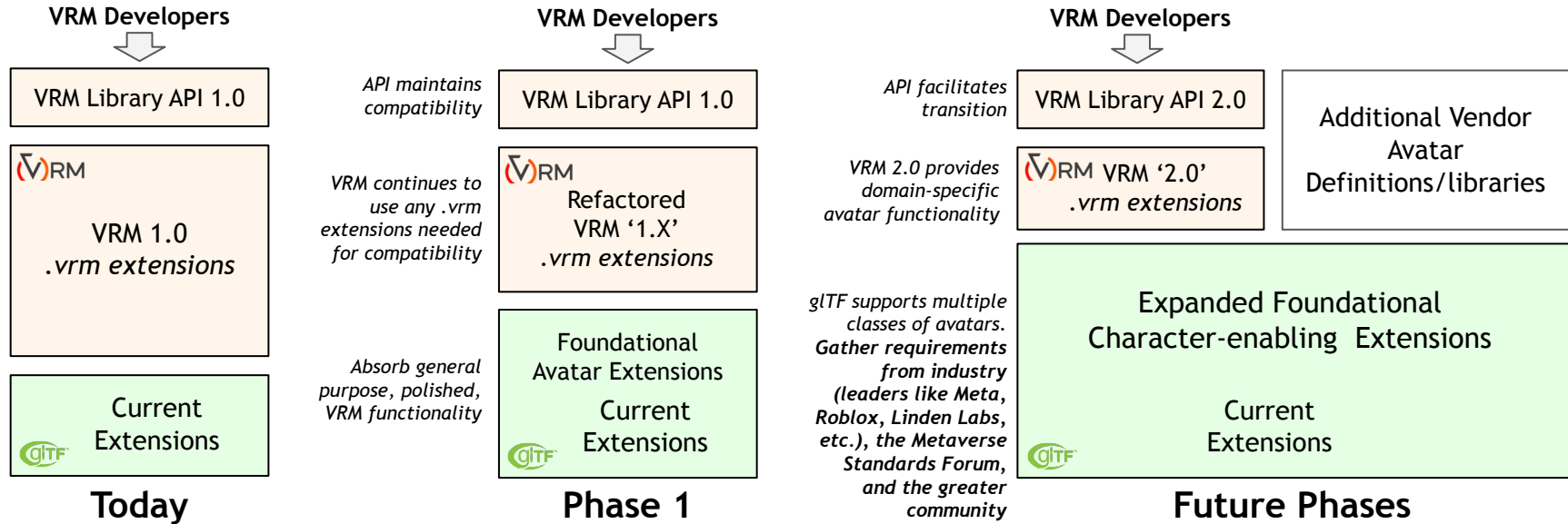
Siggraph
July 2026

Character and Avatar Extensions: A Khronos glTF and VRM Consortium Effort

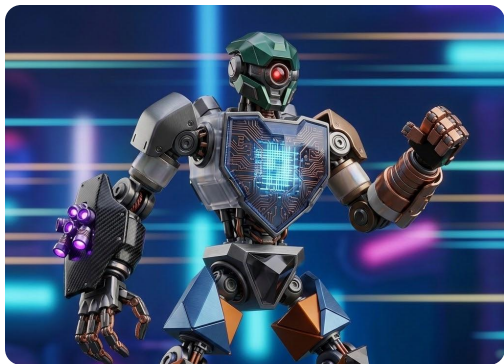
- **October 2024** - Khronos and the VRM Consortium (VRMC) announced a collaboration to evolve both glTF and VRM as global open standards to support interoperable 3D avatars
- **December 2024-October 2025** - Khronos and VRMC work together on the initial phase of general-purpose character and avatar extensions (which you can view on github)
- **October 2025** - The Khronos 3D Format Working Group established the Character and Avatar TSG (CATSG) to focus on the problem space of characters and avatars
- **Today** - Both groups have continued to collaborate, and are making steady progress on the extension set and example tooling



glTF VRM - Character Extension Convergence

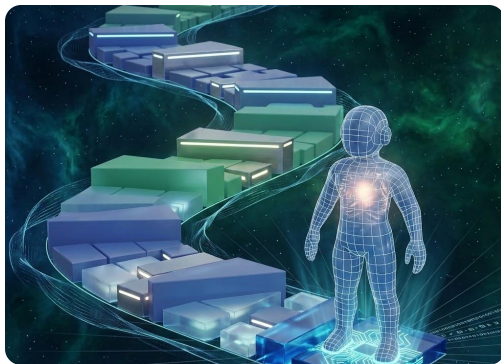


Goals and Philosophy



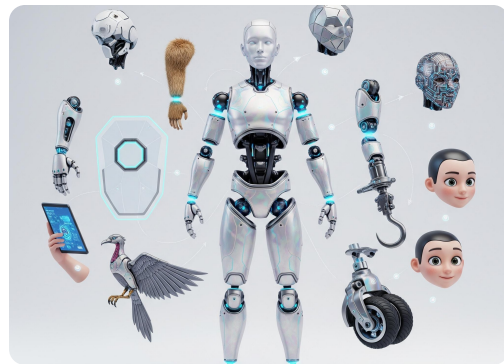
Building Blocks

Create a series of building-block standard extensions for characters and avatars.



Extension Path

Provide a path for self-describing characters and avatars through glTF extension composition.



Flexible Basis

Create a flexible basis, ratify what we know won't change, and don't prevent use of heavily opinionated properties.

Balancing Rigid Structure and Flexibility

To foster wider adoption, we are prioritizing flexibility. Establishing rigid skeletal and expression standards too early creates barriers for organizations looking to integrate these extensions.



For CATSG, this translates into us contributing to open source work on translation as well as push conversations around standardization for common character/avatar concepts



At the same time, achieving interoperability necessitates that we either standardize data composition or establish effective methods for translating shared concepts.



Interoperability Challenges

Visual & Data Structure

- Rigging & Animation Differences
- Expression & Behavior Systems
- Material & Shader Inconsistency

Dynamic Behavior

- Physics Simulation Variance
- Real-time Tracking Compatibility

Performance & Opt.

- Performance Optimization
- Level of Detail (LOD) Management

Standards Fragmentation

- Platform-Specific Features
- Standard Format Compatibility

Identity & Asset Rights

- Identity & Ownership Verification
- Asset Rights Management

Open-Source Neighbors

As part of all of this, we're actively working with the Metaverse Standards Forum and AOUSD.

- We have representation in AOUSD's Character Interest Group to ensure our ongoing standard and their upcoming work is interoperable/compatible as much as possible.
- We'll be working with the Metaverse Standards Forum with their upcoming work on Character/Avatar translation frameworks.



Example Implementations (Work in Progress)

Runtime Implementations*

Leveraged/referenced by VRM and Khronos example scenes and tools

- [Three.js](#)-based sample viewer for reference
- Unity Gltf branch & Testbed Project

Shared Libraries*

For KHR base implementations

[Three.js](#) and Unity/C#

* Prone to change in the next few months as we move towards Release Candidate status.



Three.js Testbed

Live viewer sandbox, VRM converter, and reference library



Unity Testbed

Reference runtime scene and tools



UnityGltf Plugin Branch

KHR_character plugin code

Base Extension

KHR_character

Serves as the mandatory root-level metadata extension for identifying glTF models as characters.

It enables runtime discovery, allowing engines to recognize character assets and apply relevant behaviors for all dependent character extensions.



Expressions

KHR_character_expression*

defines expressions, weights, overrides, and masks via morph targets, joints, and textures, creating a unified interface.

KHR_character_expression_mask

assists in preventing expression conflicts (e.g., speech/mouth overrides).

KHR_character_expression_mapping

enables mapping expressions to standard or internal vocabularies (e.g., ARKit, VRM).



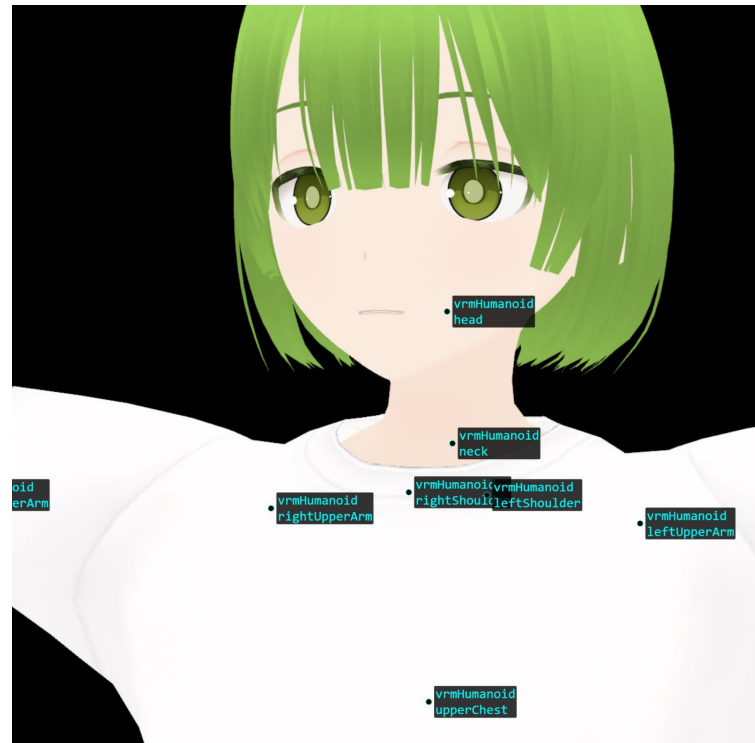
Skeleton Mapping and Reference Poses

KHR_character_skeleton_mapping

maps character rigs to standard vocabularies;
supports multiple mappings for characters with
multiple surfaces.

KHR_character_reference_pose

establishes canonical reference poses for animation
retargeting, deformation validation, and toolchain
consistency by tagging glTF animations with semantic
pose data. Stored as single-frame animations.



Node and Mesh Primitive Utilities

KHR_node_lookat

Look-at behaviors are ubiquitous in 3D applications, yet lack a standard method for identifying targets.

This extension standardizes target identification across all engines.

KHR_node_camera_hint

Provides runtime camera framing hints (e.g., first-person, third-person, selfie, custom, etc.).

Leverages glTF Camera structures to ensure consistent metadata and framing across platforms and Uls.

KHR_node_visibility_hint & KHR_mesh_primitive_visibility_hint

Provides a way to indicate what meshes and nodes are visible for a subset of the camera_hints vocabulary (first-person, third-person, always, custom).



Next Focus Areas

ACTIVE FOCUS AREAS

Animation

Types, Layers, Events, and integrating into Interactivity.

Secondary Animation & Physics

Springbone Extensions based on VRM's implementations, and skeletal geometry explorations.

FUTURE TOPICS UNDER DISCUSSION

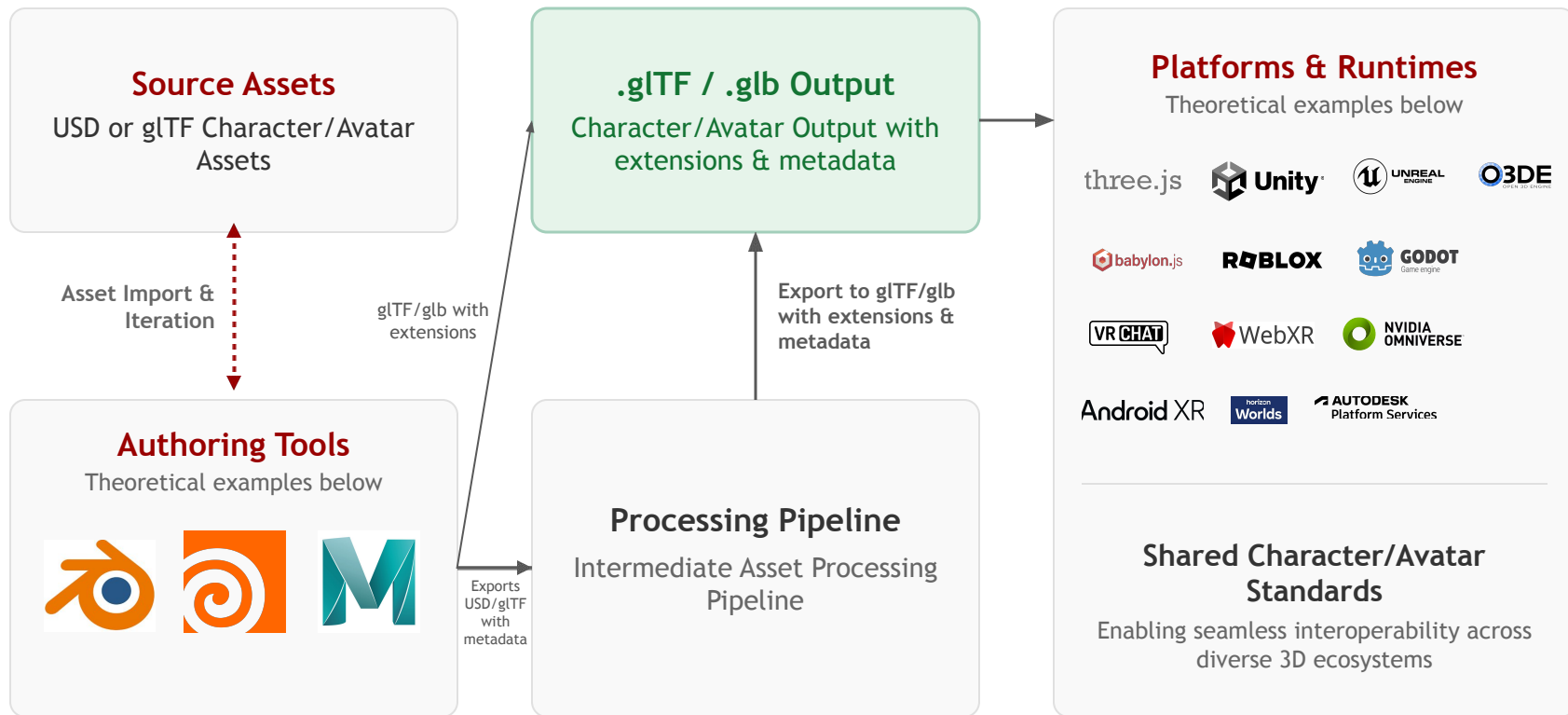
Metadata & Runtime

- **Metadata Ontologies:** Standardize asset attribution, licensing, and usage rights.
- **Mesh Annotation & LODs:** Manage runtime labels and levels of detail.
- **Skeleton Biped Standards:** Collaborate with AOUSD and MSF on common humanoid bone vocabularies.

Materials & Rendering

Toon shading for anime-style avatars and VRM compatibility.

Example Asset Flow



Thank You!

Your contributions shape the future of 3D asset standards. Here is how you can actively participate and get involved:

1. Contribute to PRs

Directly drive the specification forward by providing feedback on active pull requests and submitting your proposals.

- Review the open extension PR
- Submit specification updates
- Share implementation findings

2. Join Khronos

Become an official member to secure voting rights and collaborate on tooling, standards, and ecosystem projects.

- Collaborate on key standards
- Gain voting opportunities
- Access member-only resources

3. Join 3DF Meetings

Participate in weekly 3D Formats Working Group meetings to synchronize with the community and track progress.

- Weekly Character/Avatar TSG meetings
- Align on character features
- Network with industry experts



AOUSD

Alliance for OpenUSD

Alliance for OpenUSD

Characters Motion and Interactivity Interest Group

Interest Group

- Film
- Games
- Robotics

Goals

- Rich representation of interoperable character components
- Rich representation for simulation/interactive systems related to characters
- Foundational OpenUSD components shared across Robotics, Film and Games

Existing Tech

- UsdSkel
- OpenExec

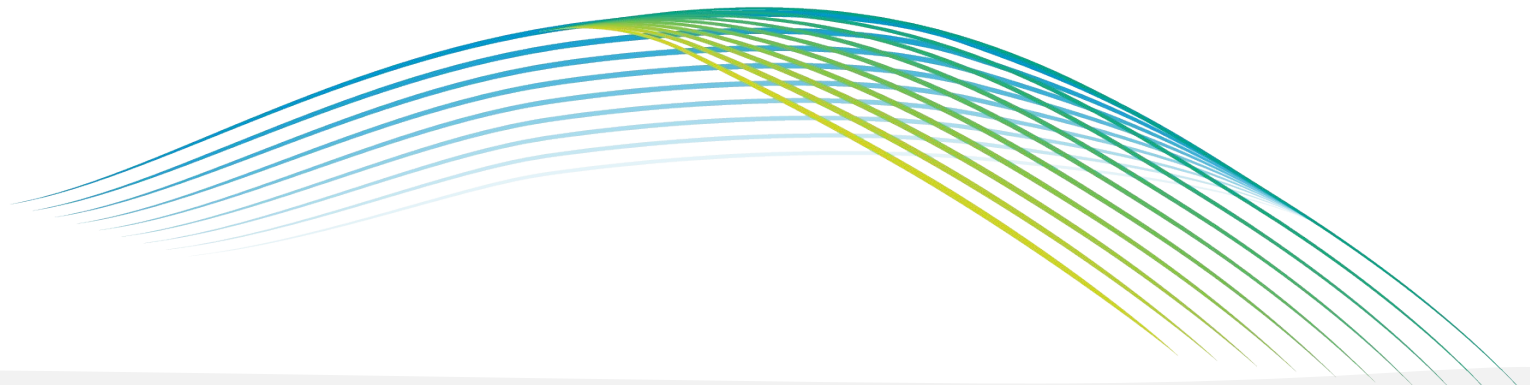
Output

- Mapping the current landscape of related efforts and ensuring coordination with ongoing initiatives
- Working Groups



3D Assets: glTF, OpenUSD, 3D Tiles, OpenPBR, MaterialX – Recent Developments & Tooling

Henrik Edstrom, Senior Principal Software Engineer, Cesium / Bentley Systems



3D Assets Working Group

Standardization is a cooperative endeavor

Cooperation between glTF and USD ecosystems is a significant industry benefit
Enabling discussions over technologies, requirements and potential roadmap alignments



KHRONOS
GROUP

Asset format to enable 3D content to be pervasively delivered and displayed on a wide diversity of native and web viewers, applications and engines



Metaverse
STANDARDS FORUM™

Cooperation between glTF and USD ecosystems is a significant industry benefit



ASWF
ACADEMY SOFTWARE FOUNDATION



Open-source projects e.g.,
OpenPBR and MaterialX



OpenUSD



AOUSD
Alliance for OpenUSD

Extensible framework and ecosystem for describing, composing, simulating, and collaboratively navigating and constructing 3D scenes



Authoring vs. Delivery Workflows



Authoring

- "Source of truth" data
- Everything is editable
- Cross-tool/vendor workflows

Delivery

- Derived data
- "Baked" – limited editing
- Optimized for transmission

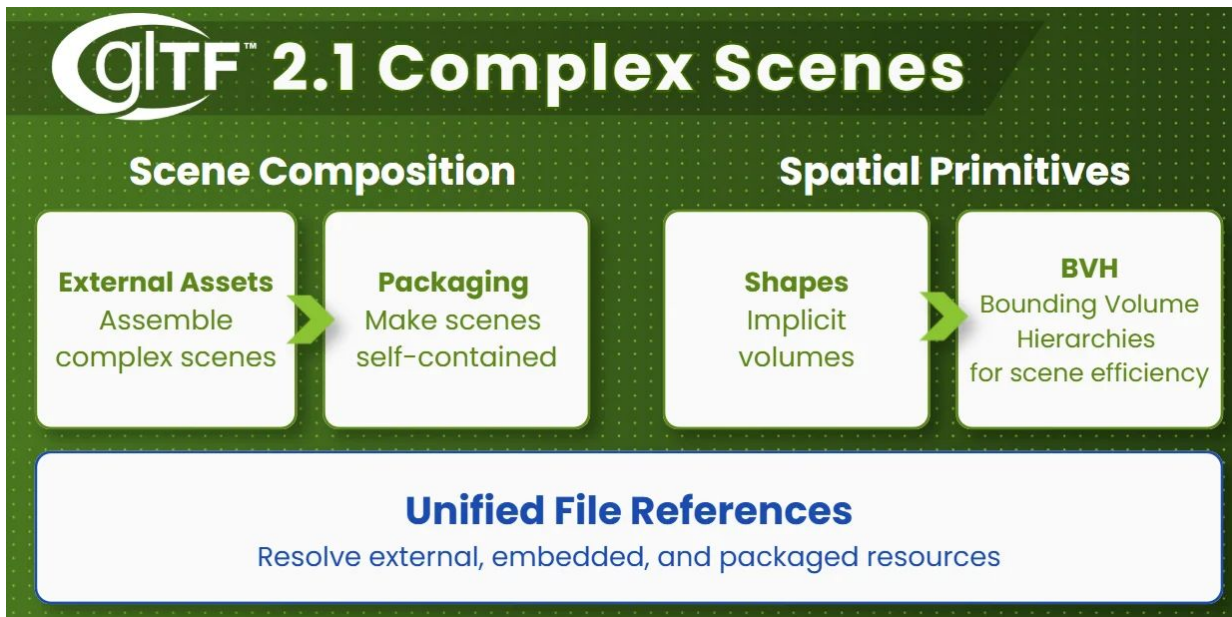


Last Mile Formats & Interchange Formats

CONTENTS	3D File Formats: Last Mile and Interchange Formats
Contents	Nick Porcino Aug 6, 2025 FOUNDATION first
Appendices	file formats glTF FBX OpenUSD last mile interchange
1. Introduction	Contributors
Foundational Taxonomy	• Primary Author: Nick Porcino; taxonomy development, rubric design • Thanks to Community contributors: Domain expertise, format-specific insights, analytical framework validation • Special Thanks: Félix Herbst, Patrick Cozzi, Guido Quaroni, Aaron Luk, Michael McCune, Rémi Arnaud, Neil Trevitt, Marc Pétit • Metaverse Standards Forum: Cross-domain interoperability requirements, emerging use case analysis • ASWF USD Working Group: USD ecosystem perspective • Alliance for OpenUSD Core Specification Working Group: USD architectural insights, interchange format analysis • Khronos Group: glTF history, evolution, and insights
Contemporary Usage	Contents
Document Scope and Audience	1. Introduction ◦ Foundational Taxonomy ◦ Contemporary Usage ◦ Analytical Framework ◦ Document Scope and Audience 2. Analytical Framework 3. Format Analysis ◦ 3.1 glTF (GL Transmission Format) ◦ 3.2 OpenUSD (Universal Scene Description) ◦ 3.3 FBX (Filmbox)
2. Analytical Framework	
2.2 Evolution Methodology	
3. Format Analysis	
3.1 glTF (GL Transmission Format)	
3.2 OpenUSD (Universal Scene Description)	
3.3 FBX (Filmbox)	
3.4 Alembic	
3.5 Additional Formats	
3.6 Collada (Collaborative Design Activity)	
3.7 Re-emergent Legacy Formats	
3.7.1 Architectural Principles of Hyper-specialized Persistence	
4. Comparative Insights	
5. Conclusion	
Recommendations for Standards Organisations	
Architectural Insights and Design Principles	
Call to Action for Community Feedback and Collaboration	



glTF 2.1



BOF: glTF 2.1 and Beyond: Building Interactive and Spatial 3D Experiences

Wednesday, July 22 | 10:00 am - 11:30 am PDT | JW Marriott Los Angeles L.A. LIVE, Diamond 1-2



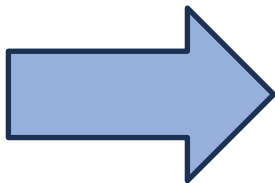
Blog: Introducing glTF 2.1 with Complex Scenes



“Baselining” 3D Tiles 2.0 on glTF 2.1



Version 2.0
Hierarchical LOD
CRS
New Content Types



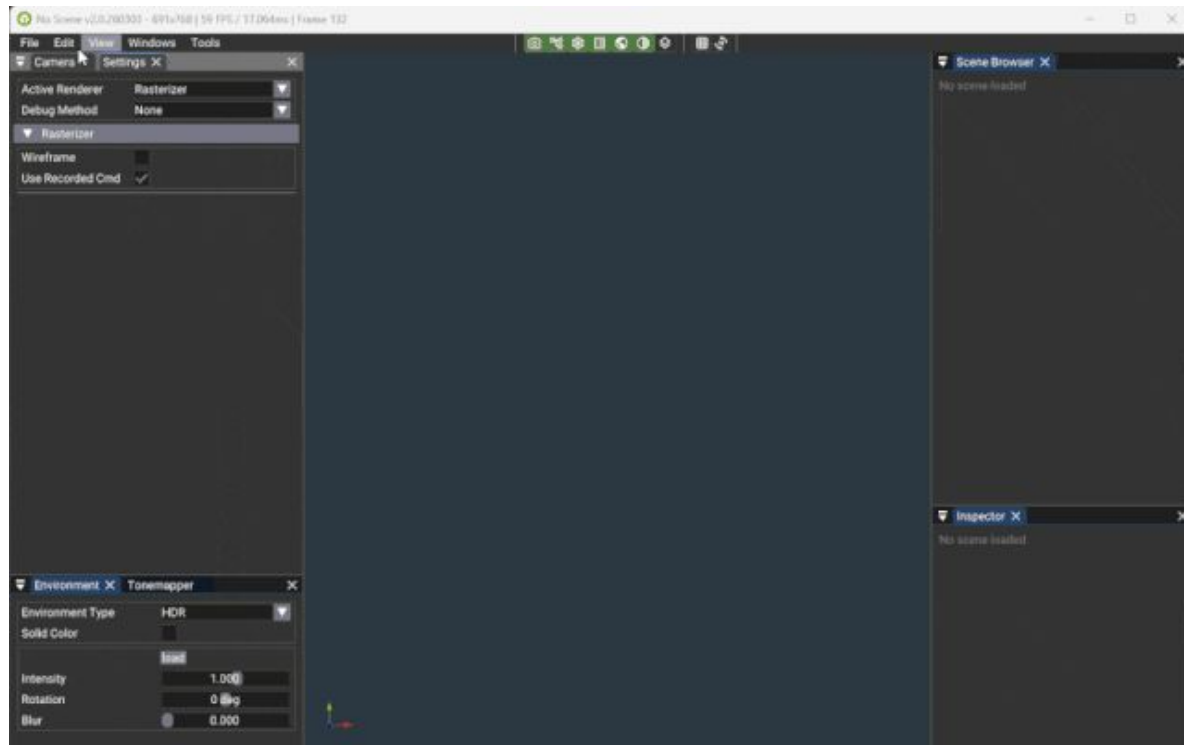
Version 2.1
External References
Bounding Volume Hierarchy (BVH)
External glTF References

Meshes
Materials



vk_gltf_renderer - glTF 2.1 preview implementation

NVIDIA Pro Samples / Martin-Karl Lefrancois



OpenUSD

USD Core Specification

v1.0.1 (b7bc21a)

AOUSD Core Spec Working Group Authors

2025-12-12



OpenUSD Core Spec 1.0



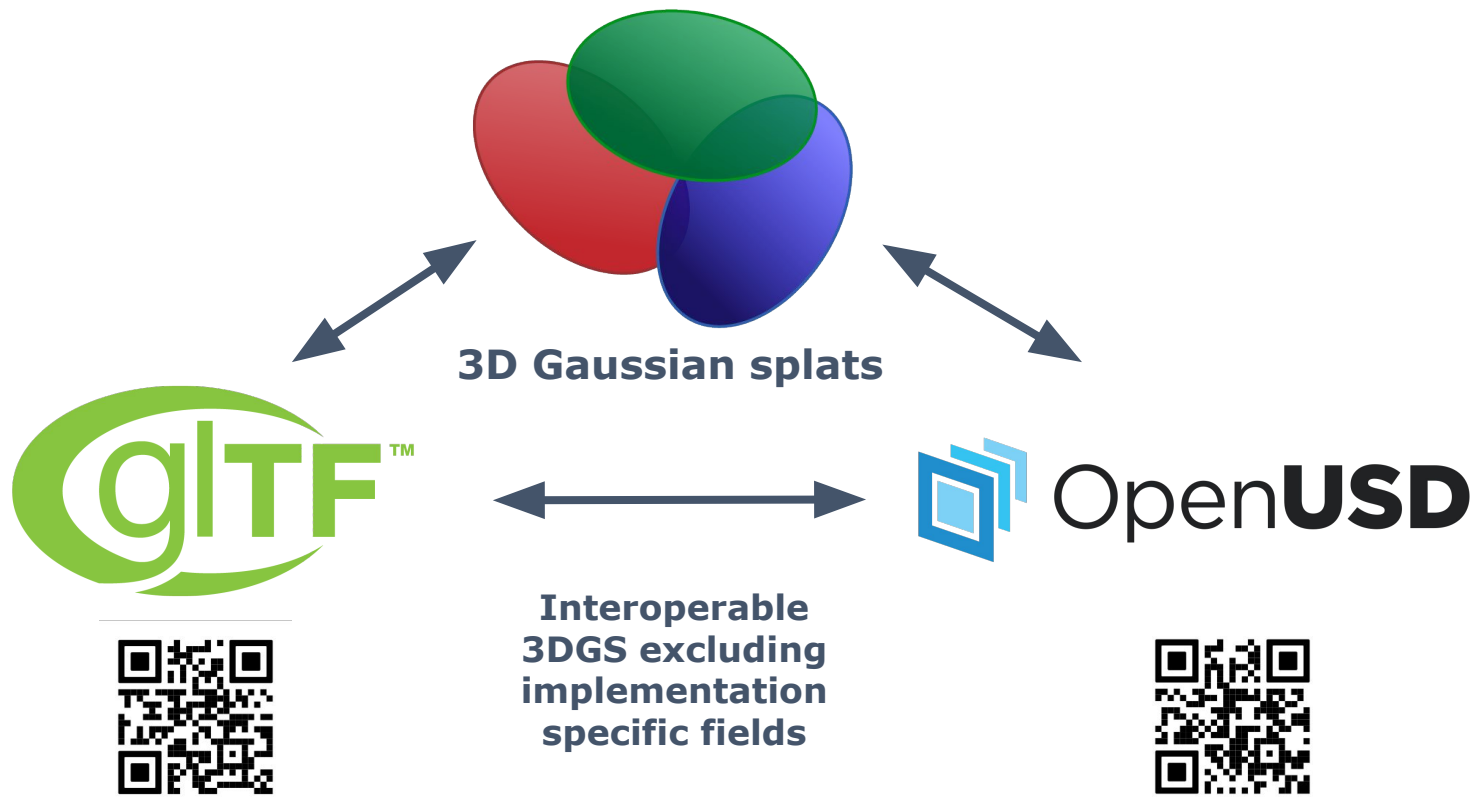
**Level of Detail (LOD) API
for USD**



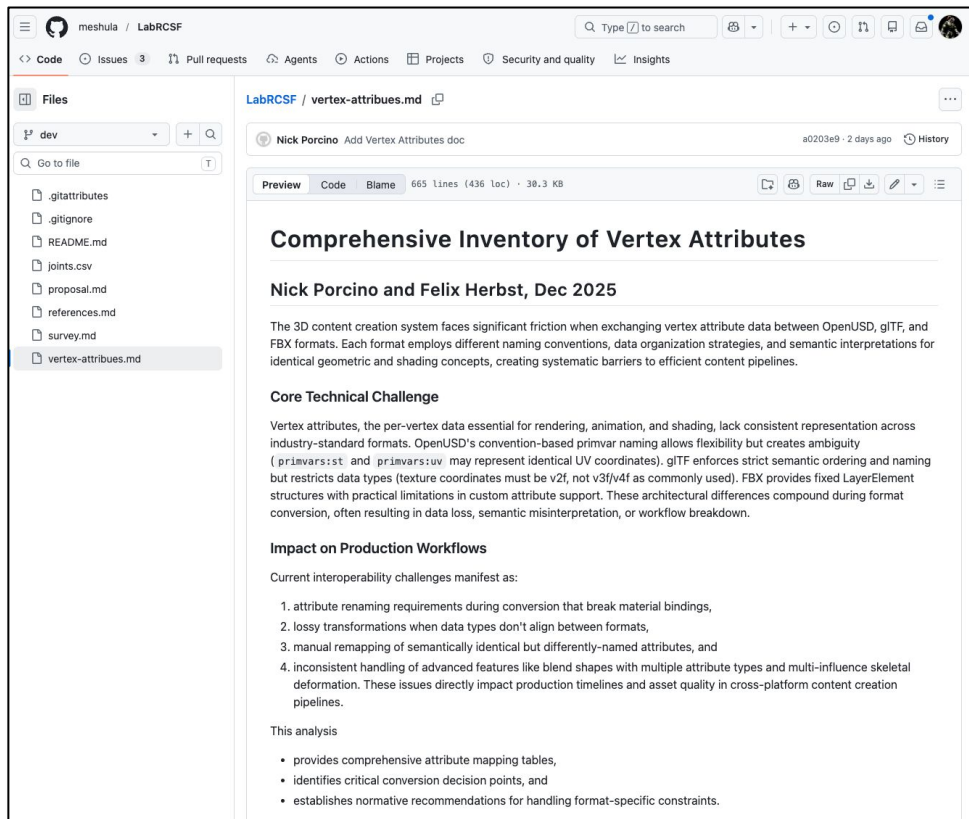
**USD Profiles and
Capabilities**



3D Gaussian Splats in OpenUSD and glTF



Vertex Attributes in OpenUSD and glTF (and FBX)



The screenshot shows a GitHub repository page for 'LabRCSF / vertex-attributes.md'. The left sidebar displays the file structure with 'vertex-attributes.md' selected. The main content area shows the document's title, author 'Nick Porcino', and a 'History' link. Below this, the document content is visible, starting with a title 'Comprehensive Inventory of Vertex Attributes' and a subtitle 'Nick Porcino and Felix Herbst, Dec 2025'. The text discusses the challenges of exchanging vertex attribute data between OpenUSD, glTF, and FBX formats, highlighting a 'Core Technical Challenge' and its 'Impact on Production Workflows'.

meshula / LabRCSF

Code Issues Pull requests Agents Actions Projects Security and quality Insights

Files

dev

Go to file

.gitattributes
.gitignore
README.md
joints.csv
proposal.md
references.md
survey.md
vertex-attributes.md

LabRCSF / vertex-attributes.md

Nick Porcino Add Vertex Attributes doc a0203e9 · 2 days ago History

Preview Code Blame 665 lines (436 loc) · 30.3 KB

Comprehensive Inventory of Vertex Attributes

Nick Porcino and Felix Herbst, Dec 2025

The 3D content creation system faces significant friction when exchanging vertex attribute data between OpenUSD, glTF, and FBX formats. Each format employs different naming conventions, data organization strategies, and semantic interpretations for identical geometric and shading concepts, creating systematic barriers to efficient content pipelines.

Core Technical Challenge

Vertex attributes, the per-vertex data essential for rendering, animation, and shading, lack consistent representation across industry-standard formats. OpenUSD's convention-based primvar naming allows flexibility but creates ambiguity (`primvars:st` and `primvars:uv` may represent identical UV coordinates). glTF enforces strict semantic ordering and naming but restricts data types (texture coordinates must be `v2f`, not `v3f/v4f` as commonly used). FBX provides fixed `LayerElement` structures with practical limitations in custom attribute support. These architectural differences compound during format conversion, often resulting in data loss, semantic misinterpretation, or workflow breakdown.

Impact on Production Workflows

Current interoperability challenges manifest as:

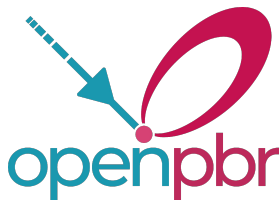
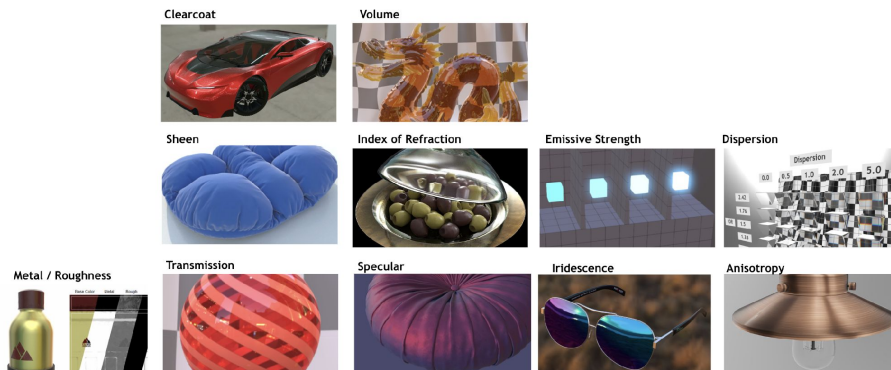
1. attribute renaming requirements during conversion that break material bindings,
2. lossy transformations when data types don't align between formats,
3. manual remapping of semantically identical but differently-named attributes, and
4. inconsistent handling of advanced features like blend shapes with multiple attribute types and multi-influence skeletal deformation. These issues directly impact production timelines and asset quality in cross-platform content creation pipelines.

This analysis

- provides comprehensive attribute mapping tables,
- identifies critical conversion decision points, and
- establishes normative recommendations for handling format-specific constraints.



MaterialX, OpenPBR and glTF PBR Alignment



USD File Format Plugins

github.com/adobe/USD-Fileformat-plugins

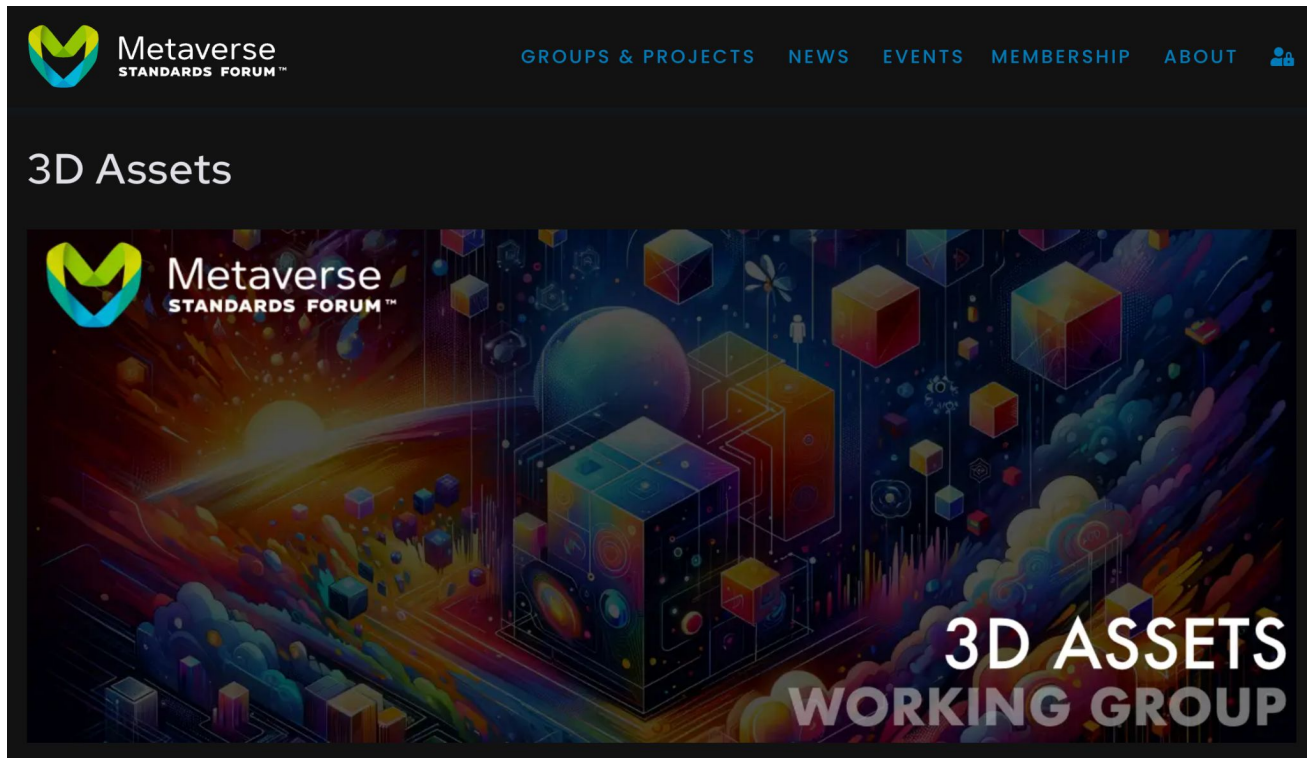
- Built with the ecosystem in mind
 - Adobe is an AOUSD founding member
 - Participating in Khronos glTF standards work
- Open-source USD plugins — 7 file formats, cross-platform
 - glTF 2.0, FBX, OBJ, PLY, SBSAR, SPZ, STL
 - Windows / macOS / Linux
- 2026.07 release: transitioning to the OpenPBR standard
 - OpenPBR-native material processing authored as MaterialX on USD
 - Expanded glTF/USD material translation



glTF plugin release contributors: Mike Bond, Filip Baba, Angus Davis, Florian Hecht, Paul Edmondson (Lead)



MSF 3D Assets Working Group





Metaverse

STANDARDS FORUM™

Standards Cooperation for an Open
and Inclusive Metaverse

metaverse-standards.org | [@Metaverse_Forum](https://twitter.com/Metaverse_Forum)