U3M - Unified 3D Material

Introduction & Technical Specification
Version 0.9 Beta

Changelog:

Creation: 7th August – Martin Semsch (Vizoo)
Update: 7th August – Martin Semsch (Vizoo)
Update: 8th August – Nolan West (Vizoo)
Update: 19th September – Martin Semsch (Vizoo)

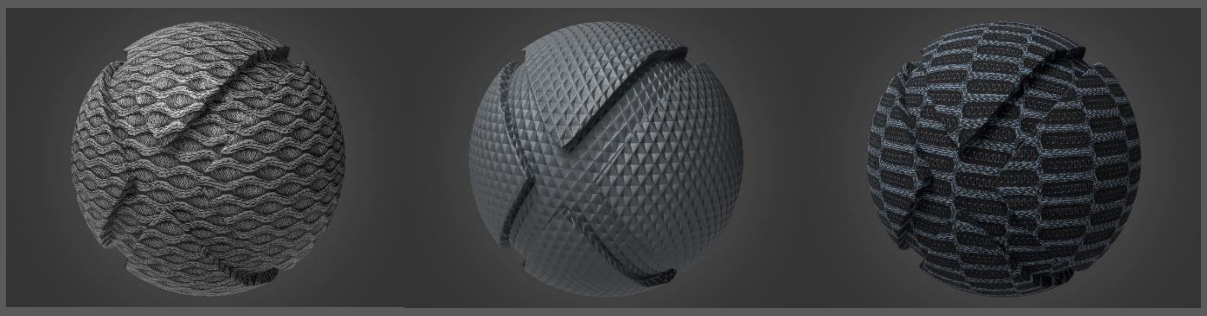
Table of Contents

1 Introduction

Right now, different vendors output their digital materials either in a proprietary format or embedded directly in their styles. And while various integrations already exist, there is a steady conversion that has to take place, different file types to manage and still a lot of incompatibility.

Therefore, the Unified 3D Material is being developed as a bridge between software vendors in the Apparel industry with the goal to replace proprietary material formats with a single open–source format.

1.1 Aligned Visualization



Loading texture maps into different 3d applications does not automatically result in identical visual representations, even if we assume that the environment and lighting situations are the same.
The reason for this is shaders. They are responsible for interpreting the textures, light, environment, geometry, and calculating the image that is being projected onto your screen. And every application uses its own slightly different shading models.

To overcome this, the U3M is now based on a fixed shading model, the **Principled BRDF***. This means that all texture maps of the U3M are based on the Principled shader, and will have the same, or very similar, appearance in applications supporting that shading model.

Even applications not supporting the Principled at this point in time now have a basis they can use to convert and translate the material into their proprietary shader engine.

* https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf

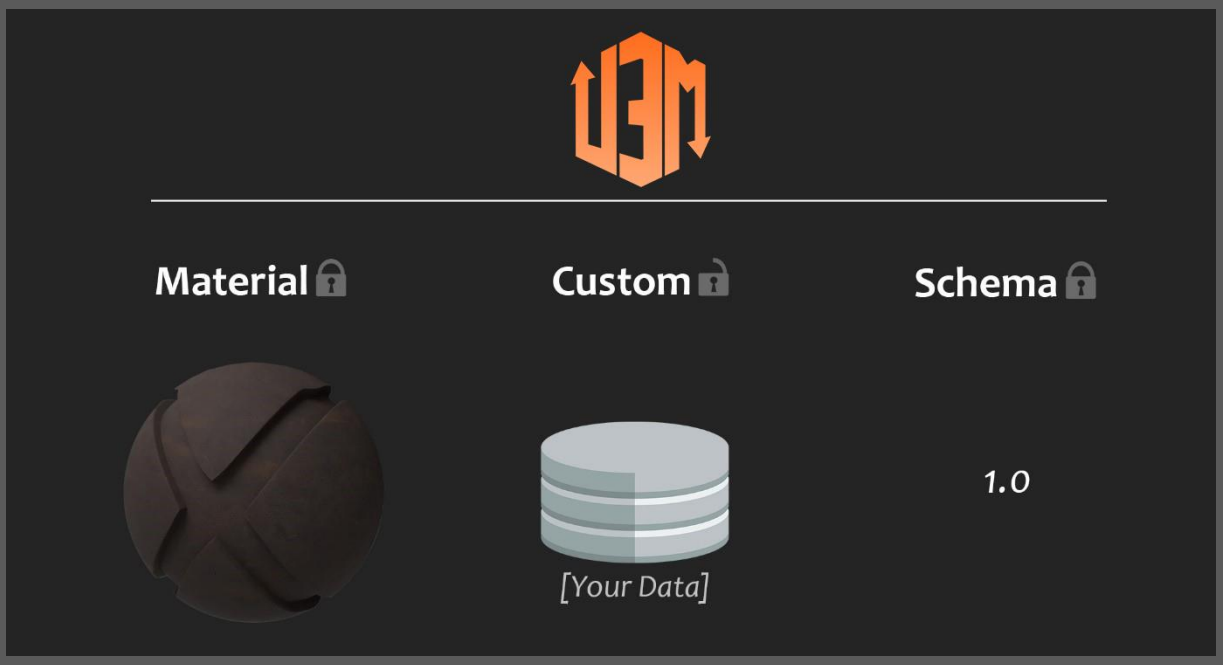
1.2 Combined Visual and Physical Information

Unfortunately, there is no equivalent to the visual alignment solution for the physical measurement information (yet). However, the U3M makes it possible to store proprietary information in its *Custom* section. This allows vendors and

users to add their physical measurements directly to the format.

With visual and physical data combined, this makes the usage in your 3D application easier and simplifies the management of the data in general.

2 Overview



For a real example of the u3m, please visit: https://github.com/vizoogmbh/u3m/blob/master/Example_0.9.u3m

For the json schema, please visit: https://github.com/vizoogmbh/u3m/blob/master/u3m_schema.json

Split into three parts, the U3M format manages to be flexible enough to embrace new input as well as manage your personal company data, yet structured enough to provide a reliable visualization.

2.1 Schema / Version

This part simply holds the version number of the U3M file. Please make sure your version number is the same matched by the corresponding json schema.
This will help to provide downward- compatibility.

2.2 Material

The material section is the core of the U3M. It contains both meta data and the visual data. Most importantly, this section is strictly structured through the corresponding JSON schema and must not be changed, as most applications reading & writing U3M rely on it.

2.2.1 Material meta data

Unique ID: Every U3M is unique through its id.

Timestamps: Both the creation and last modification dates are stored.

Name: A human readable name.

Description: Optional, can be used to add fabric information.

2.2.2 Material front & back

Front and back of the material can both be stored inside U3M and are structurally identical. The references to the texture maps and the values of the Principled Shader are stored here. Both can also be empty (null).

Each parameter of the shader can either be initialized with a default value or filled with a custom value and texture map.
The texture maps are placed relatively to the material file, either in the same folder or within a subfolder.

2.3 Custom

The Custom section is the framework that makes interoperability possible. It enables the vendors to add the data they require for reading & writing of the files, this provides brands and suppliers the option to add meta and production information to it.
This part is not controlled by the JSON schema, it has to be read and written **as it is**, without changing the data. The exception is your own proprietary Custom section.

3 Implementation

Name	n/a
Type	object
Required Properties	"schema", "material", "custom"
Optional Properties	*forbidden*
JSON-Parent	n/a
Description	U3M file

3.1 Definitions

3.1.1 Color RGB

Name	"color_rgb"
Type	object
Required Properties	"r" (type: number, minimum: 0, maximum 1), "g" (type: number, minimum: 0, maximum 1), "b" (type: number, minimum: 0, maximum 1)
Optional Properties	*forbidden*
JSON-Parent	n/a
Description	Color object of U3M, usually initialized with 1.0

3.1.2 Image

Name	"image"
Type	object
Required Properties	"width" (type: number, minimum: 0), "height" (type: number, minimum: 0), "dpi" (type: number, minimum: 0), "path" (type: string, relative), "repeat"
Optional Properties	*forbidden*
JSON-Parent	n/a
Description	Image object, stores size and path to the texture map, as well as the "repeat" type. “path” should always be relative, not absolute. “width” and “height” are in centimeters.

	“dpi” should always be kept updated in case changes have been applied to image size and resolution.

3.1.3 Repeat

Name	"repeat"
Type	object
Required Properties	"rotation" (type: number), "mode" (type: Enum ["normal", "mirror_x", "mirror_y", "mirror_xy"])
Optional Properties	*forbidden*
JSON-Parent	n/a
Description	“mode” defines how the material is supposed to repeat on the 3D model. “rotation” can be used to align the material correctly, if the thread running is not vertical.

3.1.4 Texture and Number

Name	"texture_and_number"
Type	object
Required Properties	“constant” (type: “number”), "factor" (type: "number"), "mode" (type: Enum ["add", "subtract", "multiply", "divide", "max", "min", "overlay"]), "image"
Optional Properties	*forbidden*
JSON-Parent	n/a
Description	Greyscale/data texture object of u3m. It holds an image and its adjustments, and a constant value if no texture is available or it is not supported. For this case, every texture **must** have a constant default value. “mode” defines which mathematical operator is used to apply “factor” to “image”. “add” -> “image” + “factor” “subtract” -> “image” – “factor “multiply” -> “image” x “factor” “divide” -> “image” / “factor” “max” -> if (“factor” > “image”) = “factor” “min” -> if (“factor” < “image”) = “factor” “overlay” -> see https://en.wikipedia.org/wiki/Blend_modes

3.1.5 Texture and Color

Name	"texture_and_color"
Type	object
Required Properties	“constant” (type: “color_rgb”), "factor" (type: "color_rgb"),

	"mode" (type: Enum ["add", "subtract", "multiply", "divide", "max", "min", "overlay"]), "image"
Optional Properties	*forbidden*
JSON-Parent	n/a
Description	Colored texture object of u3m. It holds an image and its adjustments, and a constant color value if no texture is available or it is not supported. For this case, every texture **must** have a constant default value. “mode” defines which mathematical operator is used to apply “factor” to “image”. “add” -> “iamge” + “factor” “subtract” -> “image” – “factor “multiply” -> “image” x “factor” “divide” -> “image” / “factor” “max” -> if (“factor” > “image”) = “factor” “min” -> if (“factor” < “image”) = “factor” “overlay” -> see https://en.wikipedia.org/wiki/Blend_modes

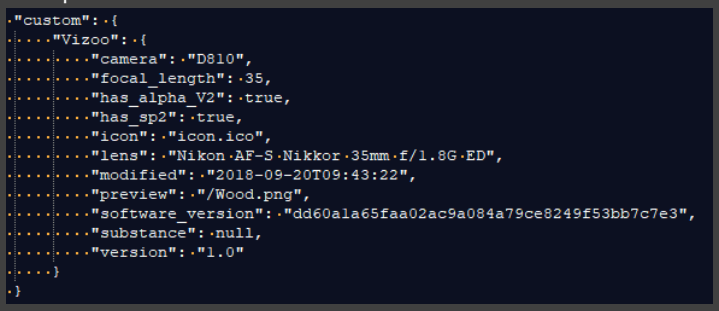
3.1.6 Visualization

Name	visualization
Type	object, null
Required Properties	"alpha", "anisotropy_rotation", "anisotropy_value", "basecolor", "clearcoat_normal", "clearcoat_roughness", "clearcoat_value", "displacement", "ior", "metalness", "normal", "roughness", "shader", "sheen_tint", "sheen_value", "specular_tint", "specular_value", "subsurface_color", "subsurface_radius", "subsurface_value", "transmission"
Optional Properties	forbidden
JSON-Parent	n/a
Description	The object contains the visual representation of the material. This can either be shader values, textures, or both.

3.2 Schema

Name	schema
Type	Enum ["1.0"]
Required Properties	*forbidden*
Optional Properties	*forbidden*
JSON-Parent	n/a
Description	Contains the version identifier of the U3M file format. **Do not change this** when editing a file. Only use an official version number, otherwise compatibility with other applications cannot be assured.

3.3 Custom

Name	custom
Type	object, null
Required Properties	n/a
Optional Properties	n/a
JSON-Parent	n/a
Description	Contains the vendor specific information. Each vendor can add a new object to it. To avoid conflict, please make sure that your object's name is proprietary to your company (e.g. "company_XYZ" instead of "metadata"). Has to be read and written **as is**, without changes, with the exception of your proprietary section. Example: 


```
"custom": {
    "Vizoo": {
        "camera": "D810",
        "focal_length": 35,
        "has_alpha_V2": true,
        "has_sp2": true,
        "icon": "icon.ico",
        "lens": "Nikon AF-S Nikkor 35mm f/1.8G ED",
        "modified": "2018-09-20T09:43:22",
        "preview": "/Wood.png",
        "software_version": "dd60a1a65faa02ac9a084a79ce8249f53bb7c7e3",
        "substance": null,
        "version": "1.0"
    }
}
```


3.4 Material

Name	material
Type	object
Required Properties	"id", "name", "description", "created", "modified", "front", "back"
Optional Properties	n/a
JSON-Parent	n/a
Description	Contains the visual and meta data regarding the material. This section must not be changed structurally. All custom data has to be written to the Custom sections.

3.4.1 ID

Name	id
Type	string

Required Properties	n/a
Optional Properties	n/a
JSON-Parent	"material"
Description	Unique ID, represented as string. Example: “{45d231aa-96a0-4d94-9aae-4a99fae0e32a}”

3.4.2 Name

Name	name
Type	string
Required Properties	n/a
Optional Properties	n/a
JSON-Parent	"material"
Description	Name of the material, minimal name length is 1.

3.4.3 Description

Name	description
Type	string
Required Properties	n/a
Optional Properties	n/a
JSON-Parent	"material"
Description	Description of the material. You can use this to describe your type of material, e.g. “twill”, “jersey”, “leather”, etc.. Can be an empty string.

3.4.4 Created

Name	created
Type	string (format: "date-time")
Required Properties	n/a
Optional Properties	n/a

JSON-Parent	"material"
Description	The date and time on which the material was created. **Must not** be edited after creation.

3.4.5 Modified

Name	modified
Type	string (format: "date-time")
Required Properties	True
Optional Properties	n/a
JSON-Parent	"material"
Description	The date and time on which the material was last modified. Has to updated by the last application editing or writing the the U3M file.

3.4.6 Front

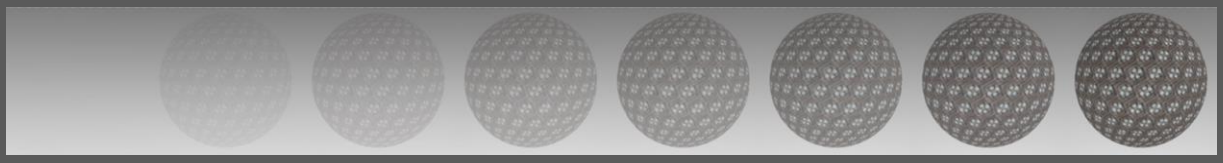
Name	front
Type	visualization
Required Properties	n/a
Optional Properties	n/a
JSON-Parent	"material"
Description	The object contains the visual representation of the material's frontside. This can either be shader values, textures, or both.

3.4.7 Back

Name	back
Type	visualization
Required Properties	n/a
Optional Properties	n/a
JSON-Parent	"material"
Description	The object contains the visual representation of the material's backside. This can either be shader values, textures, or both. In case of back being null, front is used for back instead.

3.4.8 Alpha

Name	alpha
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Alpha defines the opacity of a material. The smaller the alpha value, the bigger the transparency. The constant or factor can be used for a global transparency, an alpha texture can be used to create *holes* in the material. Default "constant": 1.0 Default "factor": 1.0 Default "mode": "multiply"



3.4.9 Anisotropy Value

Name	anisotropy_value
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Describes the intensity of the anisotropy. Default "constant": 1.0 Default "factor": 1.0 Default "mode": "multiply"

3.4.10 Anistropy Rotation

Name	anisotropy_rotation
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden

JSON-Parent	"visualization"
Description	Describes the rotation of the anisotropy in degree (0 – 360.0) Default "constant": 0 Default "factor": 1.0 Default "mode": "multiply"

3.4.11 Clearcoat Value

Name	clearcoat_value
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Describes the intensity, or visibility, of the clearcoat. By default, clearcoat is set to zero. Default "constant": 0 Default "factor": 1.0 Default "mode": "multiply"

3.4.12 Clearcoat Normal

Name	clearcoat_normal
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Describes the normals or structure of the clearcoat. The *regular* Normals only apply to the material, not the clearcoat. Constant, factor and mode are not used in this case.

3.4.13 Clearcoat Roughness

Name	clearcoat_roughness
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"

Description	Describes the roughness of the clearcoat reflection. Default “constant”: 0 Default “factor”: 1.0 Default “mode”: “multiply”

3.4.14 Index of Refraction

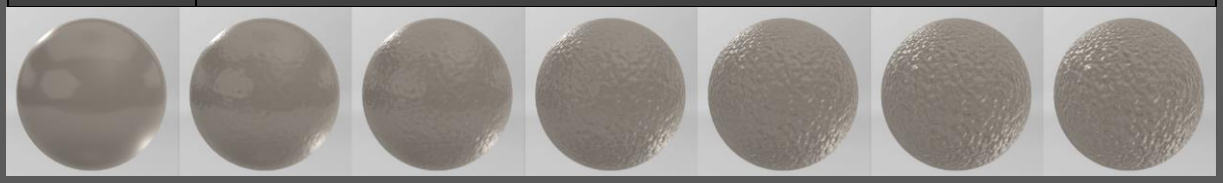
Name	ior
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Index of Refraction defines how much the path of the light is bent when it hits the material surface, in other words enters the material. In the case of the Principled shader, it affects the reflection intensity. Default “constant”: 1.4 Default “factor”: 1.0 Default “mode”: “multiply”

3.4.15 Metalness

Name	metalness
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	At the center of the shader is the *metallic value.* It defines how metallic, or dielectric, a material is. Non-metals have a maximum reflection intensity at a zero-degree observing angle of 4%. Limiting this creates much more realistic and reliable visual results. By default, materials are considered non-metallic. Default “constant”: 0 Default “factor”: 1.0 Default “mode”: “multiply”

3.4.16 Normal

Name	normal
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Describes the material surface normals, or structure of the material. This influences how the light is being reflected when hitting the surface. Normal maps do not influence the geometry's silhouette. Constant, factor and mode are not used in this case.



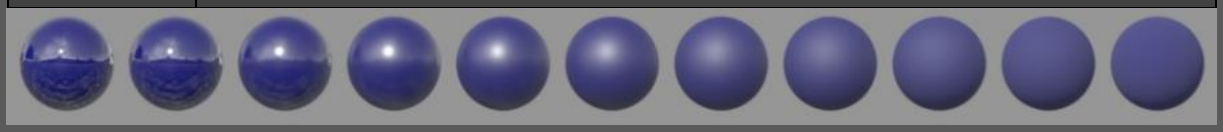
3.4.17 Displacement

Name	displacement
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Describes structure of the material. This influences how the light is being reflected when hitting the surface. Displacement maps do influence the geometry's silhouette, and are therefore *heavier* computational-wise. Default "constant": 0 Default "factor": 1.0 Default "mode": "multiply"

3.4.18 Roughness

Name	roughness
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden

JSON-Parent	"visualization"
Description	Roughness describes how much light scatters when it hits the surface, causing a highlight to either be small and bright or big and dull. Default "constant": 0.7 Default "factor": 1.0 Default "mode": "multiply"

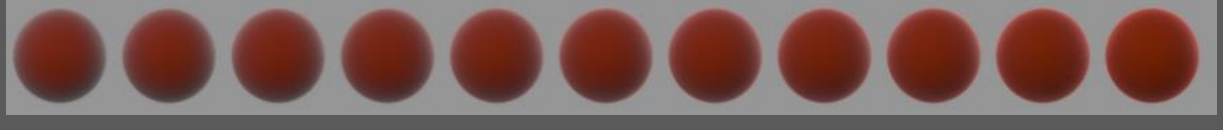


3.4.19 Sheen Value

Name	sheen_value
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Sheen increases the reflection at a grazing angle, simulation how most fabrics behave. Reacts more or less the same as *Fresnel*. Default "constant": 0 Default "factor": 1.0 Default "mode": "multiply"

3.4.20 Sheen Tint

Name	sheen_tint
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Colors the sheen with the basecolor of the material. Can be used to create a velvet-like effect. Default "constant": 0 Default "factor": 1.0 Default "mode": "multiply"



3.4.21 Specular Value

Name	specular_value
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Slightly increases or decreases the reflection intensity of non-metallic materials, between 0 – 8 % at an observer angle of zero-degrees. 0.0 equals 0 %, and 1.0 equals 8%. Default "constant": 0.5 Default "factor": 1.0 Default "mode": "multiply"

3.4.22 Specular Tint

Name	specular_tint
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Applies the basecolor to the specular reflection, creating a metallic-looking effect. Default "constant": 0 Default "factor": 1.0 Default "mode": "multiply"

3.4.23 Subsurface Color

Name	subsurface_color
Type	texture_and_color
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"

Description	Subsurface, or Subsurface Scattering (SSS) describes the reflection of the light inside the material, after penetrating the surface. This usually causes a sort of *glow*, as in jade, wax, rubber or skin. This value sets the color of the SSS effect. Default "constant": 0 Default "factor": 1.0 Default "mode": "multiply"

3.4.24 Subsurface Radius

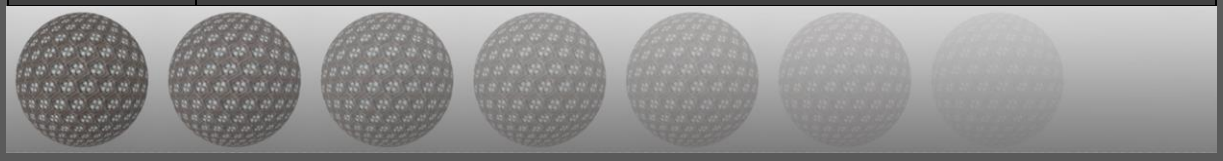
Name	subsurface_radius
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Subsurface, or Subsurface Scattering (SSS) describes the reflection of the light inside the material, after penetrating the surface. This usually causes a sort of *glow*, as in jade, wax, rubber or skin. This value defines how much the lights *spreads* inside the material. Default "constant": 0 Default "factor": 1.0 Default "mode": "multiply"

3.4.25 Subsurface Value

Name	subsurface_value
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Subsurface, or Subsurface Scattering (SSS) describes the reflection of the light inside the material, after penetrating the surface. This usually causes a sort of *glow*, as in jade, wax, rubber or skin. This value increases or decreases the intensity of the effect. Default "constant": 0 Default "factor": 1.0 Default "mode": "multiply"

3.4.26 Transmission

Name	transmission
Type	texture_and_number
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Describes how transparent a material is. In contrast to "alpha", this attribute does not affect reflection. Can be used to visualize transparent plastics or glass. Default "constant": 0 Default "factor": 1.0 Default "mode": "multiply"



3.4.27 Basecolor

Name	basecolor
Type	texture_and_color
Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	For non-metallic materials, basecolor describes the color, diffuse, or albedo of the material, either as a constant color value or texture. For metallic materials, basecolor describes the specular or reflection color of the material, and the diffuse reflection is automatically black. Default "constant": RGB (1.0, 1.0, 1.0) Default "factor": 1.0 Default "mode": "multiply"

3.4.28 Shader

Name	shader
Type	Enum ["principled"]

Required Properties	n/a
Optional Properties	forbidden
JSON-Parent	"visualization"
Description	Describes the shader type used in the u3m. Right now, only the *principled shader* is supported.

4 Be a part

Vendors, brands, suppliers – the more people participate, the more successful the format will be. In the end, everybody profits from the interoperability U3M offers.

If you want to integrate, bring in your own ideas or test the format, please reach out on:

Github: https://github.com/vizoogmbh/u3m

Slack: https://u3m.slack.com/