

# IDM-3D.Material

---

*Ekkehard Beier (Editor)*

## Version

**Specification:** IDM-3D.Material 1.0

**Document version:** Initial Edition (EN)

**Copyright:** The copyright for the material model on which this specification is based lies and remains with intelligentgraphics AG. From the point of view of intelligentgraphics AG, this does not result in any restrictions for the right of use of this specification by third parties.

## Specification

### IDM-3D.Material

IDM-3D.Material is part of the IDM-3D specification and describes the description or exchange of materials based on 2D textures and parameters, with a primary focus on the furnishing industry. Here, the focus is on the master-data level. The dynamic-data level (i.e. exchange or order data) may differ from this, for example due to the resolution of references to distributed data or the mapping of declarative parameters to direct material parameters.

Preferably, an IDM-3D.Material data record is a complete compilation of the materials of a project, whereby the assignment of a project to a manufacturer, a product series, a part-product series, several product series [at a commercial level] is not defined. Materials are usually organized cross-manufacturer, because they are relevant to several product lines.

IDM-3D.Material allows the following application scenarios, among others:

- Provision of the materials from the fabric supplier as the standard way of delivery or from a service provider who provides the hardware scans
- Provision of the materials by the user to a service provider or sales partner for further use (e.g. CGI generation, generation of a neutral material view)
- Creation of a central industry database for fabric suppliers, furniture manufacturers and the retail trade

At the geometry level, the equivalent of IDM-3D.Material is IDM-3D.Geometry. IDM-3D.Geometry and IDM-3D.Material will be the building blocks for further IDM-3D specifications.

## Material

In principle, a material is a comprehensive, geometry-independent<sup>1</sup> and context-neutral description of the way in which geometries interact with light or emit light themselves.

The material model described below represents a combination of three basic components and is primarily based on a physically-based description:

- Parameters: Phong reflection model<sup>2</sup> with extended parameters [PHONG]
- Textures: Textures that replace or are combined with the above listed parameters
- Taxonomy: Descriptive material properties, the interpretation of which allows certain degrees of freedom for each target platform

By definition, a material is a static entity and is used as such in IDM-3D. Explicit material modifications at the instantiation/assignment level are not intended. The management of dynamic materials, for example by overriding parameters to visually mark objects or geometry-dependent materials, is entirely the subject of the respective platform.

Furthermore, the term Meta Material is used. A meta material is an autonomous entity that can be referenced by N materials and replaces and/or extends their properties. Meta materials is an efficient form of description from the point of view of data creation and maintenance wherever a change or extension to a meta material can directly affect a large number of materials. From the master-data management perspective, there is also the advantage that the actual material model can be decoupled from the dynamics of additional requirements. In principle, meta materials can describe all of the components listed above and may also contain additional parameters.

Reference values for standard and reference materials are listed under [SAMPLES] and [REFERENCE].

The material model introduces the following entities:

- Material - combination of parameters, textures and the metamaterial
- Meta Material - extended material properties
- Mapping - a material or texture-related manipulation of geometry texture coordinates, see below

---

<sup>1</sup> However, there are scenarios where geometry and material are combined, for example when there are normal maps on both layers.

<sup>2</sup> The Phong reflection model can still be reasonably mapped to current APIs and exchange formats. At the same time, it represents a useful - if not completely consistent - basis for physically-based models. In this respect, it represents a kind of lowest common denominator, which is characterized by good inter-compatibility and applicability.

- Texture - one or more image files, which may be in different resolution versions

The characters A - z, 0 - 9, \_ and - are permitted for use in the names of these entities. A project may not include two entities that are identical in terms of a case-tolerant comparison.

### Material parameters

The following parameters are defined. These include some redundant parameters, whereby a scalar and a texture, for example, control the same property. In these cases, the relationship is described in the specific context.

Ambient: float [0, 1]<sup>3</sup>

This parameter is used to simulate the interaction with ambient light. It should normally result from a linear combination of the Diffuse parameter (0.3 x R + 0.59 x G + 0.11 x B).

Diffuse: { Red: float [0, 1], Green: float [0, 1], Blue: float [0, 1] }

This parameter describes the diffuse, i.e. the direction-independent, reflection. Normally, it is replaced by the diffuse texture<sup>4</sup>. Nevertheless, the parameter should always be configured meaningfully; for example, corresponding to the mean value of the diffuse texture.

Note: In principle, the diffuse value corresponds to the COLOR attribute of an IDM Living option definition.

Specular: { Red: float [0, 1], Green: float [0, 1], Blue: float [0, 1] }

Shininess: float [1, \*]

These parameters describe the specular, i.e. the direction-dependent, reflection.

The Specular value describes the color of the specular effect, while the Shininess value describes its spread.

Emission: { Red: float [0, 1], Green: float [0, 1], Blue: float [0, 1] }

This parameter describes an optional inherent color that can be used to a certain extent for mapping data light sources (floodlights, spots)<sup>5</sup>. Any materials that are

---

<sup>3</sup> Notation: parameter name: data type value range. The notation of the parameter name and the native data type are implementation-dependent.

<sup>4</sup> This means that there is no linking of color value and texture - as is the case in some other systems

<sup>5</sup> Using real active light sources in local lighting models is often problematic because of limitations and unwanted bloom effects. This should therefore be reserved for the scene environment.

not to represent an active light source, the Emission color should be black (0, 0, 0).

#### DiffuseMap: RGB texture [0, 1]

This parameter describes a diffuse texture and thus replaces the diffuse color. The latter should, however, always be specified, even if a diffuse texture exists.

Note: In principle, the diffuse texture can correspond to the TEXTURE attribute of an IDM Living option definition. However, the latter can also include a spatial representation of all aspects (wave).

#### RoughnessMap: Grayscale texture [0, 1]

This parameter describes the roughness of the surface in the value range from 0 (extremely smooth/not rough) to 1 (extremely rough). This influences the sharpness of potential reflections and the scattering of the incident light.

Physically based rendering uses this property in place of the specularity.

#### NormalMap: RGB texture

This parameter describes the alignment of the generic geometry surface using coded direction vectors.

If a material is not flat (e.g. coarsely woven fabric or brushed metal) this creates the impression of a modeled surface, independent of the underlying geometry, and thus remains material-neutral or planar in relation to the material structure. This ultimately means that any combination of geometries and material types is possible.

Tangent normal maps are used (base color: light blue), refer to the section "Texture structure".

#### Alpha: float [0, 1]

The Alpha parameter describes the transparency of the material. Here, 0 equates to full transparency and 1 equates to full opacity. The alpha value is therefore the inverse of the transparency parameters that are also sometimes used.

#### AlphaMap: Grayscale texture [0, 1]

This parameter describes the transparency of the material. The value range spreads from 0 (transparent) to 1 (opaque), meaning that black areas of an Alpha map are transparent, and white areas are opaque.

Note: Because real-time graphics cannot combine transparencies correctly in a large number of cases, the Alpha values and maps should only be used in exceptional cases. These maps are primarily intended for seam materials that

provide a strip-shaped partial mesh of an upholstery geometry with an exchangeable or configurable seam. For this, the material parameter "Overlay" (see below) must also be set.

#### DiffuseDeltaMap: RGB texture [0, 1]

This parameter describes the deviation from the above listed diffuse color or texture. Through a separate mapping process, large-area discolorations can be realized on small-tiled materials without the need for further independent texture coordinates.

The definition is as follows:

- 0.5 (medium gray) leaves the diffuse color unchanged.
- Values < 0.5 darken the color.
- Values > 0.5 brighten the color.

In principle, the RGB color channels are calculated individually - but a black and white diffuse delta map is used in most cases.

The separate mapping is based on the material texture coordinates of the geometry and usually uses a larger scaling to achieve large areas without (recognizable) repetitions.

#### RoughnessDeltaMap: Grayscale texture [0, 1]

This parameter describes a deviation from the roughness texture described below.

Through a separate mapping process, large-area brilliance differences can be realized on small-tiled materials without the need for further independent texture coordinates.

The definition is as follows:

- 0.5 (medium gray) leaves the roughness unchanged.
- Values < 0.5 reduce the roughness leading to increased brilliance.
- Values > 0.5 increase the roughness leading to reduced brilliance.

A separate mapping is based on the material texture coordinates of the geometry and usually uses a larger scaling to achieve large areas without (recognizable) repetitions.

It is logical to use the DiffuseDeltaMap and RoughnessDeltaMap synchronously, which means that they also use the same mapping.

Mapping: {  
    TranslationS: float,  
    TranslationT: float,  
    Rotation: float,

```
    ScaleS: float,  
    ScaleT: float  
}
```

This parameter describes an optional override of the geometry's (fixed) material texture coordinates, which usually (but not necessarily) correspond to an infinite grid (tiling). This can be necessary, for example, for the material-specific adjustment of incompatible tile sizes between material (i.e. the textures of a material or a texture of the material) and geometry.

This optional parameter can exist both at the material level and at the texture level. If it does not exist on the texture level, it is used from the material level.

The following properties are defined:

- TranslationS: float - shifts the texture in the horizontal direction. The unit is the normalized patch size. If the value is not specified, 0 has to be adopted.
- TranslationT: float - in contrast to TranslationS, this shifts the texture in the vertical direction.
- Rotation: float - rotates the texture clockwise. The unit is degrees. If the value is not specified, 0 has to be adopted.
- ScaleS: float - scales the texture in the horizontal direction by the specified factor. If the value is not specified, 1 has to be adopted.
- ScaleT: float - in contrast to ScaleS, this scales the texture in the vertical direction.

The sequence is defined as follows:

1. Application of the translation
2. Application of the scaling
3. Application of the rotation

## Metamaterial parameters

At the meta-material level, the following parameters can be configured and these then become effective for the respective materials. In this case, the respective materials are those that reference this meta material. If this results in overlaps, the material has a higher priority than the meta material in each case.

Here, textures and mappings are specified as local identifiers, i.e. they always refer implicitly to the respective material project. Material projects should be completed with regard to all entities. Referencing entities from other projects is not permitted.

### AlphaMap: string

This parameter is interpreted as the local identifier that refers to the transparency texture within a texture entity and sets the material parameter of the same name defined above.

### DeltaMapping: string

This parameter is interpreted as the local identifier that is to be mapped to an entity of the Mapping parameter type defined above. This mapping is then used for the DiffuseDeltaMap and RoughnessDeltaMap textures of the respective materials.

### DiffuseDeltaMap: string

This parameter is interpreted as the local identifier that refers to the diffuse delta texture within a texture entity and sets the material parameter of the same name defined above.

### DoNotRescale: bool

For psychological reasons, an explicit deviation from the true-to-scale representation of the material (i.e. the texturing) may be desirable. This can be prevented by setting this parameter to true.

### NormalMap: string

This parameter is interpreted as the local identifier that refers to the normal texture within a texture entity and sets the material parameter of the same name defined above.

### Overlay: bool

This parameter calls for an adaptive elevation of geometries for the purpose of assigning another material to the identical geometry. This is needed in the context of the alpha texture application described above. The elevation is intended to avoid so-called Z fighting.

### RoughnessDeltaMap: string

This parameter is interpreted as the local identifier that refers to the roughness delta texture within a texture entity and sets the material parameter of the same name defined above.

### RoughnessMap: string

This parameter is interpreted as the local identifier that refers to the roughness texture within a texture entity and sets the material parameter of the same name defined above.

## Taxonomy

Taxonomy is a special meta-material parameter. In contrast to the previously listed material and meta-material parameters, which have a computer graphics background, these are specifications that are primarily based on real materials and the respective manufacturing processes. Defining the taxonomy enables extended material properties to be specified in a practicable way in order to ultimately be able to achieve an arbitrarily realistic result. This can result in overriding or extending the properties already defined by the material and metamaterial specifications. In principle, the interpretation of the taxonomy depends on the respective representation system and the underlying representation method. Extreme implementations can, for example, provide for the use of procedural textures for wood materials.

Technically, the taxonomy is a straight list of properties, defined as follows:

### Wood materials

#### *Material property ("mat")*

Possible values:

- Oak ("oak")
- Beech ("beech")
- Spruce ("spruc")
- Pine ("pine")
- Maple ("maple")
- Cherry ("chry")
- Teak ("teak")

#### *Coating property ("coa")*

Possible values:

- Not coated ("nc")
- Waxed ("wa")



- Shellac ("sh")
- Lacquer ("fi")
- Oil ("oi")
- Polyurethane ("pu")
- Epoxy resin ("ep")

*Roughness property ("rou")*

Possible values:

- Value range as per Roughness - see above
- Default: 0.5

*Brightness property ("bri")*

Possible values:

- Value range: 0 ... 1
- Default: 0.5

*Contrast property ("con")*

Possible values:

- Value range: 0 ... 1
- Default: 0.5

Metal materials

*Material property ("mat")*

Possible values:

- Aluminum ("alu")
- Stainless steel ("hqust")
- Iron ("iron")
- Copper ("coppr")
- Brass ("brass")
- Gold ("gold")
- Silver ("silvr")
- Chrome ("chro")

*Coating property ("coa")*

Possible values:

- Not coated ("nc")
- Lacquer ("fi")

- Polyurethane ("pu")
- Epoxy resin ("ep")
- Galvanized ("el")
- Anodized ("an")
- Powder coated ("pc")

*Roughness property ("rou")*

Possible values:

- Value range as per Roughness - see above
- Default: 0.5

*Brightness property ("bri")*

Possible values:

- Value range: 0 ... 1
- Default: 0.5

*Base color property ("col")*

Possible values:

- Value range: "0, 0, 0" ... "1.0, 1.0, 1.0"
- Default: "0.5, 0.5, 0.6"

Plastic materials

Material property ("mat")

Possible values:

- PVC ("pvc")
- Polyester ("polye")

*Roughness property ("rou")*

Possible values:

- Value range as per Roughness - see above
- Default: 0.5

Inorganic/mineral materials

Material property ("mat")

Possible values:

- Ceramic ("ceram")
- Marble ("marbl")
- Concrete ("concr")

#### Cover materials

##### *Material property ("mat")*

Possible values:

- Leather ("leath")
- Fabric ("fabri")
- Velvet ("velv")

##### *Roughness property ("rou")*

Possible values:

- Value range as per Roughness - see above
- Default: 0.5

##### *Fresnel property ("fre")*

Possible values:

- Value range: 0 ... 2
- Default: 1

#### Transparent materials

##### *Material property ("mat")*

Possible values:

- Glass ("glass")

##### *Roughness property ("rou")*

Possible values:

- Value range as per Roughness - see above
- Default: 0.5

##### *Refraction property ("ref")*

Possible values:

- Value range: 0.01 ... 3
- Default: 1

### Other materials

#### *Material property ("mat")*

Possible values:

- Paper ("paper")

#### *Coating property ("coa")*

Possible values:

- Not coated ("nc")
- Oil ("oi")
- Lacquer ("fi")
- Shellac ("sh")

## Textures

### Texture structure

The textures should be images as square if possible, with sizes in powers of two (256, 512, 1024, 2048). This means there will be a distortion in the case of non-square patterns, which has to be compensated for by the mappings described above.

The textures must be available in the JPEG or PNG format. The file extension is *jpeg/jpg* (JPEG) or *png* (PNG).

In contrast to geometric normal textures, where PNG is the preferred format, the JPEG format is preferred for material-related normal textures.

The textures must be available as a linear RGB and may not include color profiles. It is recommended to apply an sRGB color model with a gamma value of 2.2 and then save the file without a color model.

In principle, the dpi values are ignored.

The images/textures of an entity texture are stored in a directory whose name is derived from the name of the entity plus the extension '.tex'.

In parallel to the standard (*std*) resolution, a low (*low*) and/or a high (*high*) resolution can be stored. A guide value for the standard resolutions is 1k pixels/dimension. Higher values (2k) may be supported for larger fabric

patterns (e.g. 40 x 50 cm) . The low resolution is then normally 0.5k, the higher 2k or 4k respectively.

Optional textures are declared in the material index (see below). The real patch size can then also be specified in this context, so that mappings for adapting deviating texture and geometry patch sizes can be generated automatically if necessary.

### Diffuse textures

The image content is divided by 255 per channel and used as the diffuse color on the affected part of the geometry.

Image: (127, 244, 124) → red: 0.498 , green: 0.957, blue: 0.486

The name of the texture[s] is(are) *diffuse\_{std, low, high}.{png, jpg, jpeg}*.

### Diffuse Delta textures

The image content is divided by 255 per channel and doubled.

Image: (127, 244, 124) → red delta: 0.498 , green delta: 0.957, blue delta: 0.486  
→ doubled to: red factor: 0.996, green factor: 1.914, blue factor: 0.972

To determine the overall diffuse color at a position in the geometry, the diffuse color at that position (from diffuse color or diffuse texture) is determined and multiplied by the diffuse factors at that geometry position.

Examples:

- Image: (0, 0, 0) “black” → red factor: 0.0, green factor: 0.0, blue factor: 0.0  
→ multiplied by diffuse color → darkens to black
- Image: (64, 64, 64) “dark gray” → red factor: 0.5, green factor: 0.5, blue factor: 0.5 → multiplied by diffuse color → makes it half as bright
- Image: (128, 128, 128) “medium gray” → red factor: 1.004, green factor: 1.004, blue factor: 1.004 → multiplied by diffuse color → preserves the diffuse color practically unchanged
- Image: (192, 192, 192) “light gray” → red factor: 1.5, green factor: 1.5, blue factor: 1.5 → multiplied by diffuse color → increases this by 50% i.e. brightens it
- Image: (255, 255, 255) “white” → red factor: 2, green factor: 2, blue factor: 2 → multiplied by diffuse color → doubles the values, i.e. brightens it extremely

As described above, a different mapping method can be used to simulate large-scale veils over several tiles

The name of the texture[s] is(are) *diffuse\_delta\_{std, low, high}.{png, jpg, jpeg}*.

## Normal textures

A normal texture describes the manipulation of the normal vector resulting from the geometry at a specific point (determined via the respective texture coordinates). This is performed via an image file, the dimensions of which are powers of two.

The textures are specified as tangent normal maps (light blue is vertical).

The name of the texture[s] is(are) *normals\_{std, low, high}.{png, jpg, jpeg}*.

## Roughness textures

The textures only contain grayscales. The image content is divided by 255 and used as the roughness on the affected part of the geometry.

Examples:

- Value: 0 → roughness 0 → black → not very rough/brilliant
- Value: 128 → roughness 0.5 → medium gray → medium rough
- Value: 255 → roughness 1 → black → extremely rough/matt

The name of the texture[s] is(are) *roughness\_{std, low, high}.{png, jpg, jpeg}*.

## Roughness Delta textures

The textures only contain grayscales. The image content is divided by 255, doubled, and multiplied by the roughness value on the affected part of the geometry. The image content is divided by 255 per channel and doubled.

Example:

- Value: 244 → delta roughness: 0.957 → doubled by the roughness factor: 1.914

To determine the overall roughness at a position in the geometry, the roughness at that position (from the roughness texture) is determined and multiplied by the roughness factor at that geometry position.

Examples:

- Value: 0 → “black” → roughness factor: 0 → multiplied by roughness: reduces the roughness to 0 → total brilliance
- Value: 0 → “medium gray” → roughness factor: 1 → multiplied by roughness: leaves the roughness unchanged
- Value: 255 → “white” → roughness factor: 2 → multiplied by roughness: doubles the roughness → matt

The name of the texture[s] is(are) *roughness\_delta\_{std, low, high}.{png, jpg, jpeg}*.

## Material index

An index file is a necessary component part of an IDM-3D.Material data set. This file must be valid in terms of the JSON schema [SCHEMA].

## References

[PHONG] Illumination for Computer Generated Pictures, June 1975

[REFERENCE] IG.Reference Workspace

[SAMPLES] <http://www.it.hiof.no/~borres/j3d/explain/light/p-materials.html>

[SCHEMA] <https://archive.intelligentgraphics.biz/schemas/idm-3d/material.json>

## History

**IDM-3D.Material 1.0 (EN)** - 2022-02-17