3MF Materials and Properties Extension

Specification & Reference Guide

Version	1.1
Status	Published

THESE MATERIALS ARE PROVIDED "AS IS." The contributors expressly disclaim any warranties (express, implied, or otherwise), including implied warranties of merchantability, non-infringement, fitness for a particular purpose, or title, related to the materials. The entire risk as to implementing or otherwise using the materials is assumed by the implementer and user. IN NO EVENT WILL ANY MEMBER BE LIABLE TO ANY OTHER PARTY FOR LOST PROFITS OR ANY FORM OF INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS DELIVERABLE OR ITS GOVERNING AGREEMENT, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND WHETHER OR NOT THE OTHER MEMBER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 3MF Consortium 2017

Table of Contents

Preface 3
About this Specification 3
Document Conventions....3
Language Notes 3
Software Conformance 4
Part I: 3MF Documents 5
Chapter 1. Overview of Additions 6
1.1. Resources 7
Chapter 2. Color Groups 8
2.1. Color....8
Chapter 3. Texture 2D Groups 9
3.1. Texture 2D Coordinate 9
Chapter 4. Composite Materials 11
4.1. Composite 11
Chapter 5. Multi-Properties....13
5.1. Multi-Property 14
Chapter 6. 2D Texture....15
Part II. Appendixes 17
Appendix A. Glossary....18
Appendix B. 3MF XSD Schema....19
Appendix C. Standard Namespaces and Content Types 22
C.1 Content Types 22
C.2 Relationship Types 22
C.3 Namespaces 22
References....23

Preface

About this Specification

This 3MF materials and properties specification is an extension to the core 3MF specification. This document cannot stand alone and only applies as an addendum to the core 3MF specification. Usage of this and any other 3MF extensions follow an a la carte model, defined in the core 3MF specification.

Part I, "3MF Documents," presents the details of the primarily XML-based 3MF Document format. This section describes the XML markup that defines the composition of 3D documents and the appearance of each model within the document.

Part II, "Appendixes," contains additional technical details and schemas too extensive to include in the main body of the text as well as convenient reference information.

The information contained in this specification is subject to change. Every effort has been made to ensure its accuracy at the time of publication.

This extension MUST be used only with Core specification 1.0.

Document Conventions

Except where otherwise noted, syntax descriptions are expressed in the ABNF format as defined in RFC 4234.

Glossary terms are formatted like *this*.

Syntax descriptions and code are formatted in `monospace` type.

Replaceable items, that is, an item intended to be replaced by a value, are formatted in *`monospace cursive`* type.

Notes are formatted as follows:

> **Note:** This is a note.

Language Notes

In this specification, the words that are used to define the significance of each requirement are written in uppercase. These words are used in accordance with their definitions in RFC 2119, and their respective meanings are reproduced below:

Copyright 3MF Consortium 2017

- *MUST.* This word, or the adjective “REQUIRED,” means that the item is an absolute requirement of the specification.
- *SHOULD.* This word, or the adjective “RECOMMENDED,” means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
- *MAY.* This word, or the adjective “OPTIONAL,” means that this item is truly optional. For example, one implementation may choose to include the item because a particular marketplace or scenario requires it or because it enhances the product. Another implementation may omit the same item.

Software Conformance

Most requirements are expressed as format or package requirements rather than implementation requirements.

In order for consumers to be considered conformant, they must observe the following rules:

- They MUST NOT report errors when processing conforming instances of the document format except when forced to do so by resource exhaustion.
- They SHOULD report errors when processing non-conforming instances of the document format when doing so does not pose an undue processing or performance burden.

In order for producers to be considered conformant, they must observe the following rules:

- They MUST NOT generate any new, non-conforming instances of the document format.
- They MUST NOT introduce any non-conformance when modifying an instance of the document format.

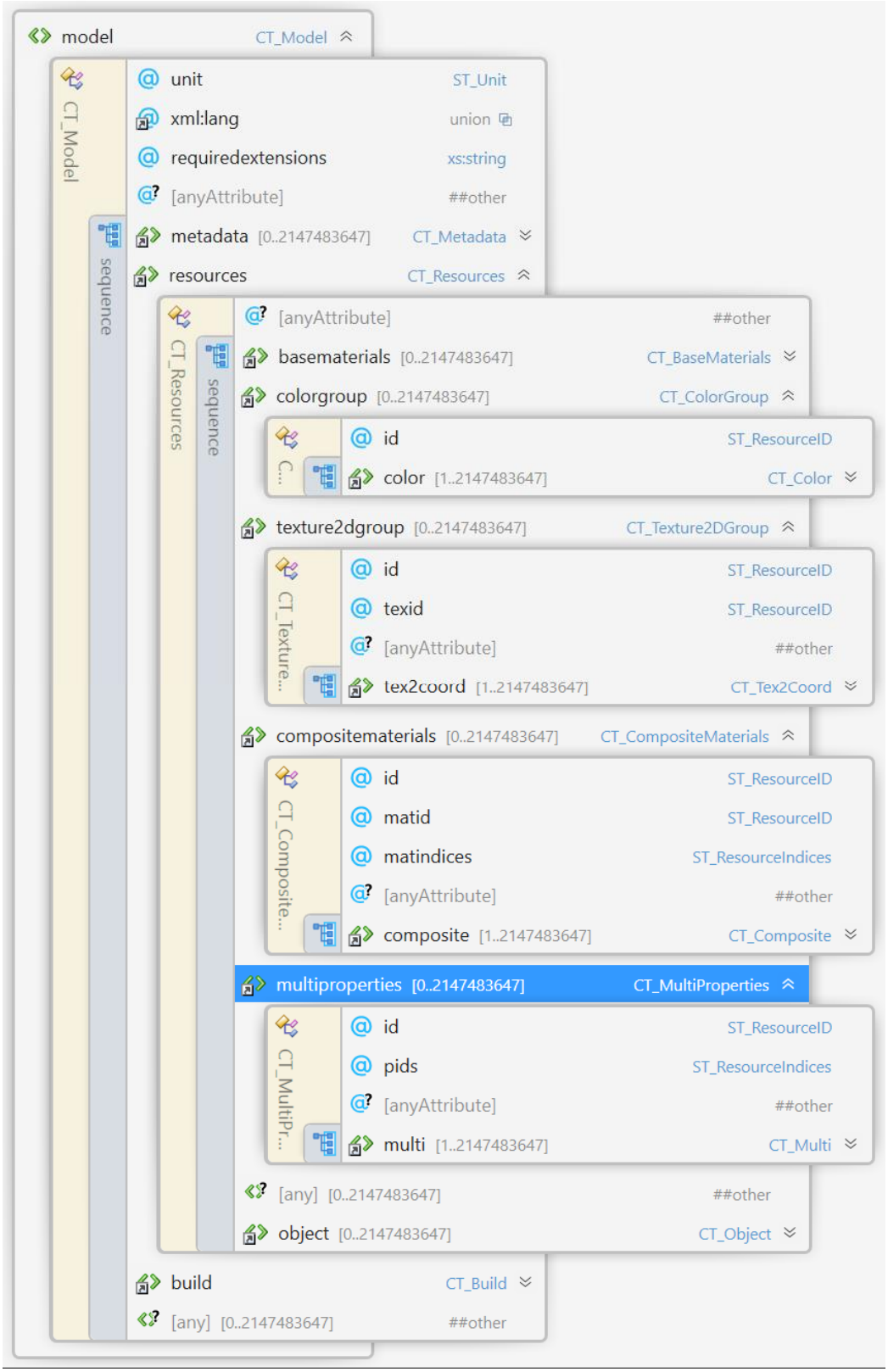
Editing applications are subject to all of the above rules.

Part I: 3MF Documents

Copyright 3MF Consortium 2017

Chapter 1. Overview of Additions

Figure 1-1: Overview of model XML structure of 3MF with material and property additions



This chapter describes new non-object resources. Each of these resources is OPTIONAL for producers, but MUST be supported by consumers that specify support for this materials extension of 3MF.

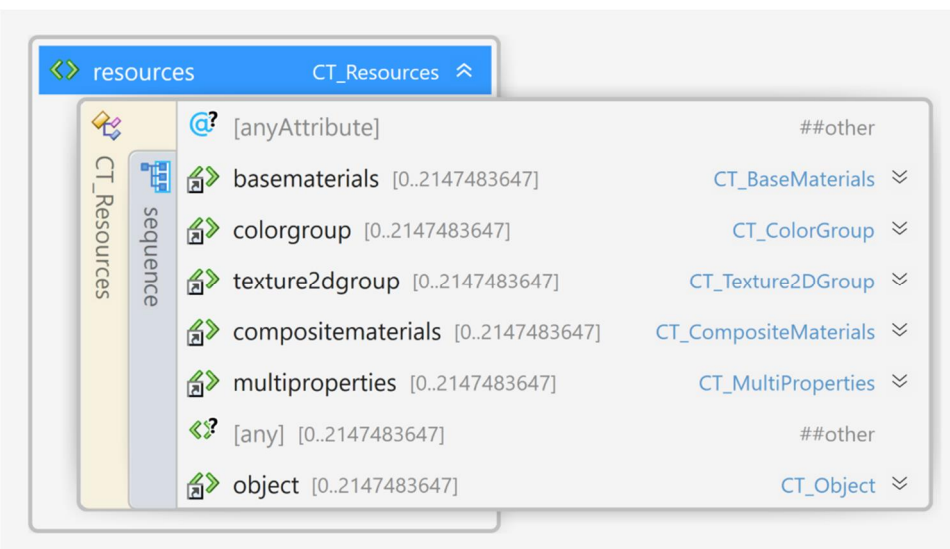
As a general idea, the following resource groups will determine different ways of representing material properties of a part. The corresponding resource IDs MAY be referenced by triangle attributes defined in the core specification.

As there are existing file formats and use cases which need multiple pieces of information per triangle, it is possible to define multiple properties per triangle (see chapter 5). Consumers MUST be strict in obeying the mixing rules as laid out in the corresponding paragraphs in order to avoid ambiguous interpretation of the design intent.

1.1. Resources

Element **<resources>**

diagram

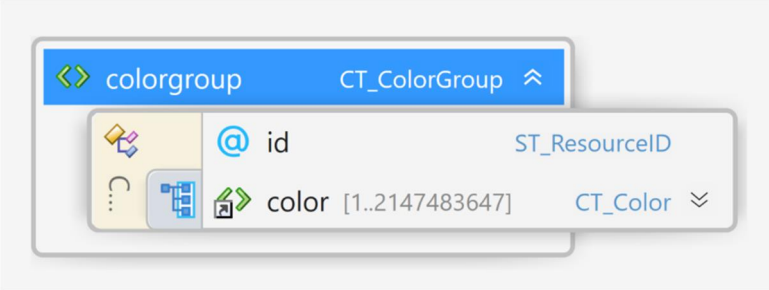


All of the new elements defined in this 3MF extension specification live under the <resources> element from the core 3MF specification. The <object> and <basematerials> elements are from the core spec, while the rest are defined in the following chapters. The ordering shown here is not enforced in the schema, as these extension elements all fall under the <any> element from the core spec.

Copyright 3MF Consortium 2017

Chapter 2. Color Groups

Element **<colorgroup>**

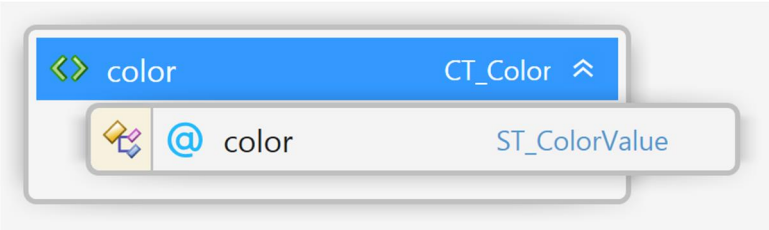
diagram	

attributes	Name	Type	Use	Default	Fixed	Annotation
	id	**ST_ResourceID**	required			Unique ID among all resources (which could include elements from extensions to the spec).

A <colorgroup> element acts as a container for color properties. The order of these elements forms an implicit 0-based index that is referenced by other elements, such as the <object> and <triangle> elements. A producer MAY define multiple <colorgroup> containers in order to help organize the file, for instance by grouping colors related to particular objects.

2.1. Color

Element **<color>**

diagram	

attributes	Name	Type	Use	Default	Fixed	Annotation
	color	**ST_ColorValue**	required			Specifies the sRGB color for rendering the material.

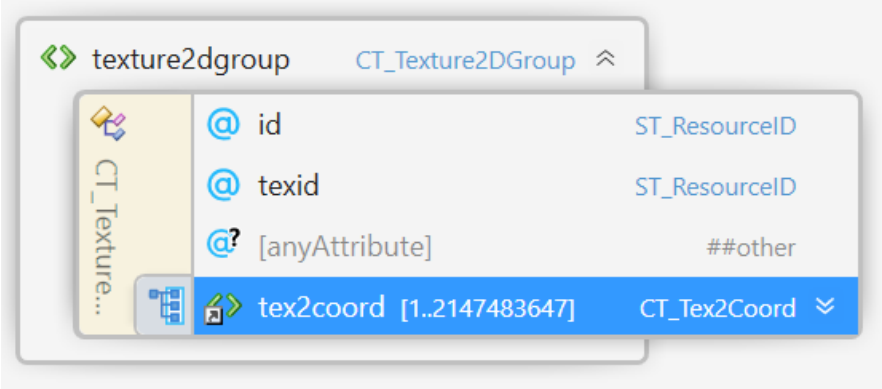
Colors are used to represent rich color, specifically what most 3D formats call “vertex colors”. These elements are used when color is the only property of interest for the material, and a large number will be needed. The format is the same sRGB color as defined in the core 3MF specification.

In order to avoid integer overflows, a color group MUST contain less than 2^31 colors.

Chapter 3. Texture 2D Groups

Element **<texture2dgroup>**

diagram



attributes

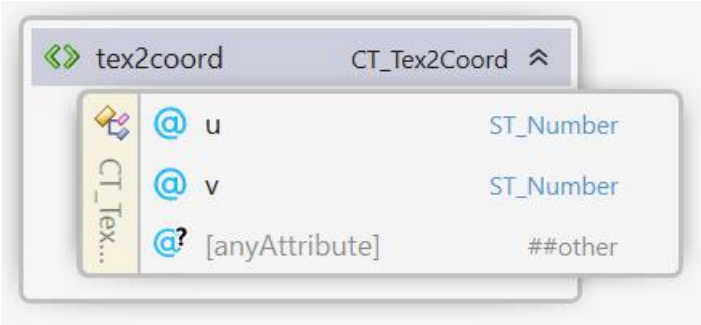
Name	Type	Use	Default	Fixed	Annotation
id	**ST_ResourceID**	required			Unique ID among all resources (which could include elements from extensions to the spec).
texid	**ST_ResourceID**	required			Reference to the <texture2d> element with the matching id attribute value.

A <texture2dgroup> element acts as a container for texture coordinate properties. The order of these elements forms an implicit 0-based index that is referenced by other elements, such as the <object> and <triangle> elements. It also specifies which image to use, via texid. The referenced <texture2d> elements are described below in Chapter 6.

3.1. Texture 2D Coordinate

Element **<tex2coord>**

diagram



attributes

Name	Type	Use	Default	Fixed	Annotation
u	**ST_Number**	required			The u-coordinate within the texture, horizontally right from the origin in the lower left of the texture.
v	**ST_Number**	required			The v-coordinate within the texture, vertically up from the origin in the lower left of the texture.

Copyright 3MF Consortium 2017

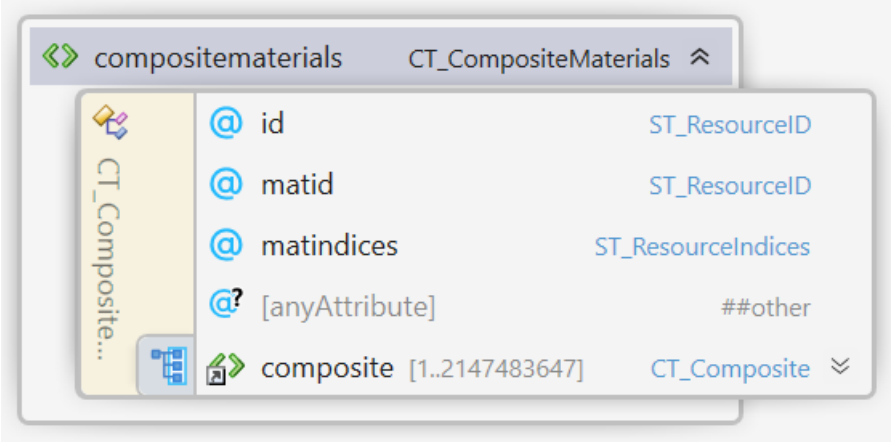
Texture coordinates map a vertex of a triangle to a position in image space (U, V coordinates). Texture mapping allows high-resolution color bitmaps to be applied to any surface. The primary advantage of texture mapping over the vertex colors of the previous section is that the textures allow color at a much finer detail level than the underlying mesh, while vertex colors are always at the same resolution as the mesh.

In order to avoid integer overflows, a texture coordinate group MUST contain less than 2^31 tex2coords.

Chapter 4. Composite Materials

Element **<compositematerials>**

diagram



attributes

Name	Type	Use	Default	Fixed	Annotation
id	**ST_ResourceID**	required			Unique ID among all resources (which could include elements from extensions to the spec).
matid	**ST_ResourceID**	required			Reference to the base material group element with the matching id attribute value (e.g. <basematerials>).
matindices	**ST_ResourceIndices**	required			A space-delimited list of ST_ResourceIndex values of the material constituents

A <compositematerials> element acts as a container for composite materials. The order of these elements forms an implicit 0-based index that is referenced by other elements, such as the <object> and <triangle> elements. A producer MAY define multiple <compositematerials> containers, for instance by grouping mixtures of different materials.

The <compositematerials> element defines materials derived by mixing 2 or more base materials in defined ratios. This collective mixture is referred to as a *composite material*. The matid attribute specifies the material group that all constituents are from, which MUST be a <basematerials> group. The matindices attribute specifies the indices of the materials to mix.

4.1. Composite

Element **<composite>**

diagram

Copyright 3MF Consortium 2017

attributes	Name	Type	Use	Default	Fixed	Annotation
	values	**ST_Numbers**	required			A space-delimited list of ST_Number values between 0 and 1, inclusive representing the fraction of the material constituents, respectively.

The <composite> element defines a values attribute, which specifies the proportion of the overall mixture for each material. If the sum of the values is greater than zero, consumers MUST divide each value by the sum of the values of all constituent value attributes to apply the correct proportion for each material. If the sum of all constituent value attributes is zero, each value MUST be treated as 1.0 divided by the number of constituent elements.

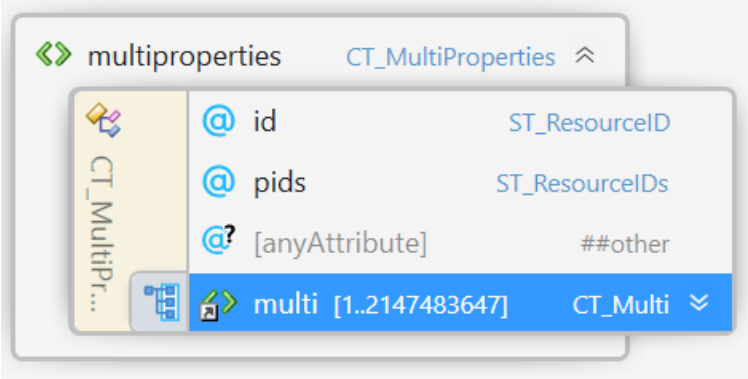
If the values list is shorter than the matindices list, consumers MUST use a default value of zero for unspecified values. Extra values MUST be ignored.

In order to avoid integer overflows, a composite group MUST contain less than 2^31 composites.

Chapter 5. Multi-Properties

Element **<multiproperties>**

diagram



attributes

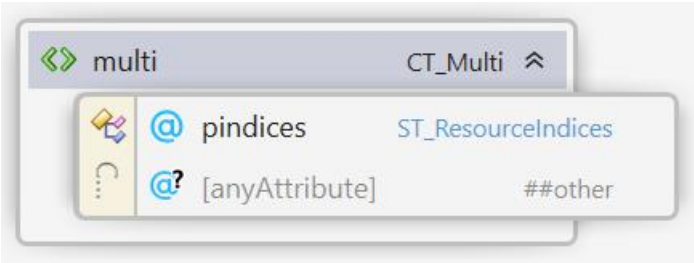
Name	Type	Use	Default	Fixed	Annotation
id	**ST_ResourceID**	required			Unique ID among all resources (which could include elements from extensions to the spec).
pids	**ST_ResourceIDs**	required			A space-delimited list of ST_ResourceID values representing the property group of each constituent.

A <multiproperties> element acts as a container for multi-properties. The order of these elements forms an implicit 0-based index that is referenced by other elements, such as the <object> and <triangle> elements. The pids list MUST NOT contain more than one reference to a material (base or composite). Likewise, the pids list MUST NOT contain more than one reference to a colorgroup (for performance reasons). Finally, the pids list MUST NOT contain any references to a multiproperties group. A producer MAY define multiple <multiproperties> containers, for instance to layer textures in a different order or to specify a different base material.

Copyright 3MF Consortium 2017

Multi-Property

Element **<multi>**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	pindices	**ST_ResourceIndices**	required			A space-delimited list of ST_ResourceIndex values of the constituents

The <multi> element combines the constituent materials and properties. This can be used for layering 2D textures and colors, or adding independent material properties defined in other extensions.

In the case of layering colored materials, colorgroup and texture2dgroup are handled the same way. First they are independently sampled and interpolated on a triangle, then they are layered, starting from the first one in the list, using their alpha values to control transparency to the colors below, according to the following equation:

$$\text{[illegible]} = \text{[illegible]} * \text{[illegible]} + \text{[illegible]} * (1 - \text{[illegible]})$$

The first layer is considered fully opaque (its alpha channel MUST be ignored) and sRGB values are mixed without performing a gamma correction.

Display color of the base material is overridden by any other color properties specified.

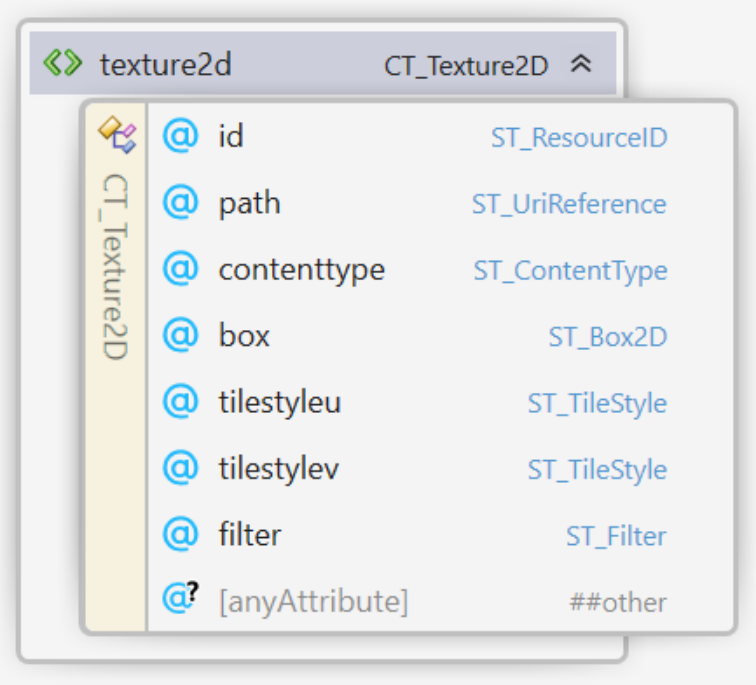
If the pindices list is shorter than the pids list, consumers MUST use a default index of zero for any unspecified pindices. Extra pindices MUST be ignored.

In order to avoid integer overflows, a multi properties group MUST contain less than 2^31 elements.

Chapter 6. 2D Texture

Element **<texture2d>**

diagram



attributes

Name	Type	Use	Default	Fixed	Annotation
id	**ST_ResourceID**	required			Specifies a unique identifier for this texture resource.
path	**ST_UriReference**	required			Specifies the part name of the texture data.
contenttype	**ST_ContentType**	required			Specifies the content type of the 2D Texture part referenced by the path attribute. Valid values are image/jpeg and image/png.
box	**ST_Box2D**		ñ0 0 1 1ò		A bounding box (ñu v width heightò) defining the boundary of the texture swatch to tile.
tilestyleu	**ST_TileStyle**		wrap		Specifies how tiling should occur in the u axis in order to fill the overall requested area. Valid values are wrap, mirror, clamp.
tilestylev	**ST_TileStyle**		wrap		Specifies how tiling should occur in the v axis in order to fill the overall requested area. Valid values are wrap, mirror, clamp.
filter	**ST_Filter**		auto		Specifies the texture filter to apply when scaling the source texture. Allowed values are ñautoò, ñlinearò, ñnearestò

A *2D texture resource* provides information about *texture* image data, found via the provided path reference, which MUST also be the target of a 3D Texture relationship from the 3D Model part. The box attribute allows specifying a sub-portion of the image as a texture swatch that can be tiled as necessary.

Copyright 3MF Consortium 2017

The four numbers specifying the box are horizontal offset, vertical offset, width and height, all in image-relative measurements, so 0,0 is the lower-left corner of the image and 1,1 is the upper-right corner. Texture 2D Coordinates are specified relative to the bounding box, so 0,0 is the lower-left corner of the box and 1,1 is the upper-right corner of the box. If the bounding box exceeds the image (beyond the zero to one range), area outside of the image is assigned the default material color specified at the object level.

The tile style of wrap essentially means that the same texture should be repeated in the specified axis (both in the positive and negative directions), for the axis value modulo the specified width or height of the bounding box. The tile style of mirror means that each time the texture width or height is exceeded, the next repetition of the texture should be reflected across a plane perpendicular to the axis in question. The tile style of clamp means all Texture 2D Coordinates outside of the range zero to one will be assigned the color of the nearest edge pixel.

The only supported content types are JPEG and PNG, as more specifically specified in the 3MF core spec under the Thumbnails section. Alpha transparency values are ignored, except in the case of multi-materials, as described in chapter 5.

The producer MAY require the use of a specific filter type by specifying either “linear” for bilinear interpolation or “nearest” for nearest neighbor interpolation. The producer SHOULD use “auto” to indicate to the consumer to use the highest quality filter available. If source texture is scaled with the model, the specified filter type MUST be applied to the scaling operation. The default value is “auto”.

Part II. Appendixes

Copyright 3MF Consortium 2017

Appendix A. Glossary

3D model. The markup that defines a model for output.

3D Model part. The OPC part that contains a 3D model.

3D Texture part. A file used to apply complex information to a 3D object in the 3D Model part. In this extension spec, it specifically a JPEG of PNG image used to apply color data to the surface of an object.

3MF. The 3D Manufacturing Format described by this specification, defining one or more 3D objects intended for output to a physical form.

3MF Document. The digital manifestation of an OPC package that contains a 3D payload that conforms with the 3MF specification.

Composite material. A material that is comprised of a ratio of other materials.

Consumer. A software, service, or device that reads in a 3MF Document.

Editor. A software, service, or device that both reads in and writes out 3MF Documents, possibly changing the content in between.

Material. The description of a physical substance that can be used to output an object.

Material resource. A potential resource that might be referenced by an object to describe what the object will be made of.

Producer. A software, service, or device that writes out a 3MF Document.

Resource. A texture, color, material, action, or object that could be used by another resource or might be necessary to build a physical 3D object according to build instructions.

Texture resource. A resource in an object that describes the basic size and location of the texture image data.

XML namespace. A namespace declared on the <model> element, in accordance with the XML Namespaces specification.

Appendix B. 3MF XSD Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://schemas.microsoft.com/3dmanufacturing/material/2015/02"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xml="http://www.w3.org/XML/1998/namespace"
targetNamespace="http://schemas.microsoft.com/3dmanufacturing/material/2015/02"
elementFormDefault="unqualified" attributeFormDefault="unqualified" blockDefault="#all">
   <xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd"/>
   <xs:annotation>
      <xs:documentation><![CDATA[
      Schema notes:

      Items within this schema follow a simple naming convention of appending a prefix indicating the
type of element for references:

      Unprefixed: Element names
      CT_: Complex types
      ST_: Simple types

      ]]></xs:documentation>
   </xs:annotation>
   <!-- Complex Types -->
   <xs:complexType name="CT_Texture2D">
      <xs:attribute name="id" type="ST_ResourceID" use="required"/>
      <xs:attribute name="path" type="ST_UriReference" use="required"/>
      <xs:attribute name="contenttype" type="ST_ContentType" use="required"/>
      <xs:attribute name="box" type="ST_Box2D"/>
      <xs:attribute name="tilestyleu" type="ST_TileStyle" default="wrap"/>
      <xs:attribute name="tilestylev" type="ST_TileStyle" default="wrap"/>
      <xs:attribute name="filter" type="ST_Filter" default="auto"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
   </xs:complexType>
   <xs:complexType name="CT_ColorGroup">
      <xs:sequence>
         <xs:element ref="color" maxOccurs="2147483647"/>
      </xs:sequence>
      <xs:attribute name="id" type="ST_ResourceID" use="required"/>
   </xs:complexType>
   <xs:complexType name="CT_Color">
      <xs:attribute name="color" type="ST_ColorValue" use="required"/>
   </xs:complexType>
   <xs:complexType name="CT_Texture2DGroup">
      <xs:sequence>
         <xs:element ref="tex2coord" maxOccurs="2147483647"/>
      </xs:sequence>
      <xs:attribute name="id" type="ST_ResourceID" use="required"/>
      <xs:attribute name="texid" type="ST_ResourceID" use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
   </xs:complexType>
   <xs:complexType name="CT_Tex2Coord">
      <xs:attribute name="u" type="ST_Number" use="required"/>
      <xs:attribute name="v" type="ST_Number" use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
   </xs:complexType>
   <xs:complexType name="CT_CompositeMaterials">
      <xs:sequence>
         <xs:element ref="composite" maxOccurs="2147483647"/>
      </xs:sequence>
      <xs:attribute name="id" type="ST_ResourceID" use="required"/>
      <xs:attribute name="matid" type="ST_ResourceID" use="required"/>
      <xs:attribute name="matindices" type="ST_ResourceIndices" use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
   </xs:complexType>
   <xs:complexType name="CT_Composite">
      <xs:attribute name="values" type=" ST_Numbers" use="required"/>
   </xs:complexType>
```

Copyright 3MF Consortium 2017

```
   <xs:complexType name="CT_MultiProperties">
      <xs:sequence>
         <xs:element ref="multi" maxOccurs="2147483647"/>
      </xs:sequence>
      <xs:attribute name="id" type="ST_ResourceID" use="required"/>
      <xs:attribute name="pids" type="ST_ResourceIDs" use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
   </xs:complexType>
   <xs:complexType name="CT_Multi">
      <xs:attribute name="pindices" type="ST_ResourceIndices" use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
   </xs:complexType>
   <!-- Simple Types -->
   <xs:simpleType name="ST_ContentType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="image/jpeg"/>
         <xs:enumeration value="image/png"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="ST_TileStyle">
      <xs:restriction base="xs:string">
         <xs:enumeration value="clamp"/>
         <xs:enumeration value="wrap"/>
         <xs:enumeration value="mirror"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="ST_Filter">
      <xs:restriction base="xs:string">
         <xs:enumeration value="auto"/>
         <xs:enumeration value="linear"/>
         <xs:enumeration value="nearest"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="ST_ColorValue">
      <xs:restriction base="xs:string">
         <xs:pattern value="#[0-9|A-F|a-f][0-9|A-F|a-f][0-9|A-F|a-f][0-9|A-F|a-f][0-9|A-F|a-f][0-9|A-F|a-f][0-9|A-F|a-
f]([0-9|A-F|a-f][0-9|A-F|a-f])?"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="ST_UriReference">
      <xs:restriction base="xs:anyURI">
         <xs:pattern value="/.*"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="ST_Box2D">
      <xs:restriction base="xs:string">
         <xs:whiteSpace value="collapse"/>
         <xs:pattern value="((\-|\+)?(([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?) ((\-
|\+)?(([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?) ((([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-
|\+)?[0-9]+)?) ((([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?)"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="ST_Number">
      <xs:restriction base="xs:double">
         <xs:whiteSpace value="collapse"/>
         <xs:pattern value="((\-|\+)?(([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?)"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="ST_ZeroToOne">
      <xs:restriction base="ST_Number">
         <xs:minInclusive value="0.0"/>
         <xs:maxInclusive value="1.0"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="ST_Numbers">
      <xs:restriction base="xs:string">
         <xs:whiteSpace value="collapse"/>
         <xs:pattern value="(((\-|\+)?(([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?( )?)+)"/>
      </xs:restriction>
```

```
   </xs:simpleType>
   <xs:simpleType name="ST_ResourceID">
      <xs:restriction base="xs:positiveInteger">
         <xs:maxExclusive value="2147483648"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="ST_ResourceIndex">
      <xs:restriction base="xs:nonNegativeInteger">
         <xs:maxExclusive value="2147483648"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="ST_ResourceIndices">
      <xs:restriction base="xs:string">
         <xs:whiteSpace value="collapse"/>
         <xs:pattern value="(([0-9]+)( )?)+"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="ST_ResourceIDs">
      <xs:restriction base="xs:string">
         <xs:whiteSpace value="collapse"/>
         <xs:pattern value="(([0-9]+)( )?)+"/>
      </xs:restriction>
   </xs:simpleType>
   <!-- Elements -->
   <xs:element name="texture2d" type="CT_Texture2D"/>
   <xs:element name="colorgroup" type="CT_ColorGroup"/>
   <xs:element name="color" type="CT_Color"/>
   <xs:element name="texture2dgroup" type="CT_Texture2DGroup"/>
   <xs:element name="tex2coord" type="CT_Tex2Coord"/>
   <xs:element name="compositematerials" type="CT_CompositeMaterials"/>
   <xs:element name="composite" type="CT_Composite"/>
   <xs:element name="multiproperties" type="CT_MultiProperties"/>
   <xs:element name="multi" type="CT_Multi"/>
</xs:schema>
```

Copyright 3MF Consortium 2017

Appendix C. Standard Namespaces and Content Types

C.1 Content Types

3D Texture	application/vnd.ms-package.3dmanufacturing-3dmodeltexture

C.2 Relationship Types

3D Texture	http://schemas.microsoft.com/3dmanufacturing/2013/01/3dtexture

C.3 Namespaces

Advanced Materials	http://schemas.microsoft.com/3dmanufacturing/material/2015/02

References

BNF of Generic URI Syntax

"BNF of Generic URI Syntax." World Wide Web Consortium.
http://www.w3.org/Addressing/URL/5_URI_BNF.html

JPEG

Hamilton, Eric. "JPEG File Interchange Format, Version 1.02." World Wide Web Consortium. 1992.
http://www.w3.org/Graphics/JPEG/jfif3.pdf

Open Packaging Conventions

Ecma International. "Office Open XML Part 2: Open Packaging Conventions." 2006. http://www.ecma-international.org

PNG

Duce, David (editor). "Portable Network Graphics (PNG) Specification," Second Edition. World Wide Web Consortium. 2003. http://www.w3.org/TR/2003/REC-PNG-20031110

sRGB

Anderson, Matthew, Srinivasan Chandrasekar, Ricardo Motta, and Michael Stokes. "A Standard Default Color Space for the Internet-sRGB, Version 1.10." World Wide Web Consortium. 1996.
http://www.w3.org/Graphics/Color/sRGB

Unicode

The Unicode Consortium. The Unicode Standard, Version 4.0.0, defined by: *The Unicode Standard, Version 4.0*. Boston, MA: Addison-Wesley, 2003.

XML

Bray, Tim, Eve Maler, Jean Paoli, C. M. Sperlberg-McQueen, and François Yergeau (editors). "Extensible Markup Language (XML) 1.0 (Fourth Edition)." World Wide Web Consortium. 2006.
http://www.w3.org/TR/2006/REC-xml-20060816/

XML C14N

Boyer, John. "Canonical XML Version 1.0." World Wide Web Consortium. 2001.
http://www.w3.org/TR/xml-c14n.

XML Namespaces

Copyright 3MF Consortium 2017

Bray, Tim, Dave Hollander, Andrew Layman, and Richard Tobin (editors). “Namespaces in XML 1.0 (Second Edition).” World Wide Web Consortium. 2006. http://www.w3.org/TR/2006/REC-xml-names-20060816/

XML Schema

Beech, David, Murray Maloney, Noah Mendelsohn, and Henry S. Thompson (editors). “XML Schema Part 1: Structures,” Second Edition. World Wide Web Consortium. 2004. http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/

Biron, Paul V. and Ashok Malhotra (editors). “XML Schema Part 2: Datatypes,” Second Edition. World Wide Web Consortium. 2004. http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/