

## 3MF 核心规格



# 3D 制造格式

## 核心规格及参考指南

版本 1.3.0

地位 发布

## 免责声明

这些材料按“原样”提供。贡献者明确否认任何保证（明示、暗示或其他方式），包括与材料相关的适销性、非侵权、适合特定用途或所有权的暗示保证。实施或以其他方式使用这些材料的全部风险由实施者和用户承担。在任何情况下，任何成员均不对任何其他方因利润损失或任何形式的

对于与本交付物或其管辖协议有关的任何诉讼原因而造成的任何性质的间接、特殊、偶然或结果性损害，不论是基于违约、侵权（包括疏忽）还是其他原因，也不论其他成员是否已被告知有此类损害的可能性。

## 目录

- [前言](#)
  - [关于本规范](#)
  - [文档约定](#)
  - [语言说明](#)
  - [软件一致性](#)
- [第一部分 3MF 文件](#)
  - [第 1 章 3MF 文档格式 1.1. 包 第 2 章 零件和关系](#)
    - [2.1. 3D 有效载荷](#)
  - [2.2. 零件命名建议](#)
    - 
    - 
    - [2.3. 3MF 文档标记](#)
  - [第 3 章 3D 模型 3.1. 坐标空间](#)
    - 
    - [3.2. 相对方向和测量](#)
    - [3.3. 3D 矩阵](#)
    - [3.4. 模型](#)
  - [第 4 章 对象资源](#)
    - [4.1. 网格](#)
    - [4.2. 组件](#)
  - [第五章 材料资源 5.1. 基础材料](#)
    -
  - [第 6 章 3MF 文档包功能](#)
    - [6.1. 包缩略图和对象缩略图 6.2. 核心属性](#)
    - 
    - [6.3. 数字签名](#)
    - [6.4. 受保护的内容](#)

- 第二部分 附录 附录 A. 词
  - 汇总表 附录 B.1. 3MF XSD 架构
  - 
  - 附录 B.2. 3MF 元数据示例 附录 C. 标准命名空间和内
  - 容类型 C.1 内容类型 C.2 关系类型
    - 
    - 
    - C.3 命名空间

## 前言

### 关于本规范

3D 制造格式( 3MF )描述了使用 XML 和其他广泛使用的技术来描述一个或多个 3D 模型的内容和外观的一组约定。它是为构建系统来处理 3MF 内容的开发人员编写的。

本规范的主要目标是确保生产或使用 3MF 内容的独立创建的软件和硬件系统的互操作性。本规范定义了生产者和消费者为实现互操作性必须满足的正式要求。

本规范描述了一种称为 3MF 文档的 3D 模型和包含格式。格式要求是开放打包约定 (OPC) 规范中描述的打包要求的扩展。该规范描述了使用 XML、Unicode、ZIP 和其他技术和规范来组织构成任何模型的内容和资源的打包和物理格式约定。它们是 3MF 规范的组成部分。

理解此规范需要具备可扩展标记语言 (XML) 和 XML 命名空间规范的应用知识。全面理解可能还需要具备 3D 制造领域内常用术语和程序的领域知识,尽管我们已尽最大努力将这种依赖降到最低。

3MF 联盟提供[免费使用的开源实现](#)以便处理 3D 内容的应用程序能够轻松采用该格式。

第一部分“3MF 文档”介绍了主要基于 XML 的 3MF 文档格式的详细信息。本节介绍定义 3D 文档的组成和文档中每个模型的外观的 XML 标记。

第二部分“附录”包含额外的技术细节和图表,这些细节和图表过于繁多,无法包含在正文中,也包含方便的参考信息。

本规范中包含的信息可能会发生变化。我们已尽最大努力确保其在发布时的准确性。

此核心规范已扩展并添加了内容。例如,前缀“t”映射到 xml 命名空间“<http://schemas.microsoft.com/3dmanufacturing/trianglesets/2021/07>”前缀“mm”映射到 xml 命名空间“<http://schemas.microsoft.com/3dmanufacturing/mirroring/2021/07>”,均在 1.3 版中定义。请参阅[附录 C.3 命名空间](#)

### 文档约定

除非另有说明,语法描述均以 RFC 4234 中定义的 ABNF 格式表示。

词汇表术语的格式如下。

语法描述和代码被格式化为Markdown代码块。

可替换项（即旨在由值替换的项）采用等宽草书字体格式。

注释格式如下：

注:这是一条注释。

## 语言说明

在本规范中,用于定义每个要求的重要性的单词均以大写形式书写。这些单词的使用符合 RFC 2119 中的定义,其各自含义如下所示：

- 必须。该词或形容词“必需”表示该项目是规范的绝对要求。
- 应该。这个词,或者形容词“推荐”,意味着在特定情况下可能存在忽略此项目的正当理由,但在选择其他方案之前,应该充分理解其含义并仔细权衡情况。
- 可以。这个词或形容词“可选”表示此项确实是可选的。例如,一种实现方式可能选择包含此项,因为特定市场或场景需要它,或者因为它可以增强产品。另一种实现方式可能会忽略同一项。

## 软件一致性

大多数需求都表达为格式或包装要求而不是实现要求。

为了让消费者被视为符合规定,他们必须遵守以下规则：

- 在处理符合文档格式的实例时,它们绝不能报告错误,除非由于资源耗尽而被迫这样做。
- 当处理不符合要求的文档格式实例时,他们应该报告错误,只要这样做不会造成过度的处理或性能负担。

为了使生产者被视为符合要求,他们必须遵守以下规则：

- 他们不得生成任何新的、不符合文档格式的实例。
- 他们在修改文档格式实例时不得引入任何不符合规定的情况。

编辑申请必须遵守上述所有规则。

# 第一部分 3MF 文件

## 第 1 章 3MF 文档格式

本规范描述了 3MF 文档格式的内部组织方式以及在外以 3D 对象的形式实现的方式。它可以用作独立文件格式,也可以用作打印管道中的有效负载。它建立在开放包装约定规范中描述的原则之上。3MF 文档必须遵守该规范的所有要求和建议,除非本规范另有说明。此处提供的信息既适用于以 3MF 文档格式发布内容的生产者,也适用于访问 3MF 文档内容并将其转换为 3D 对象的消费者。编辑器是充当 3MF 文档格式内容的生产者和消费者的实体。制造设备是生产物理部件的消费者。

3MF 文档格式表示 3D 模型,或以标记格式表示一个或多个物理对象描述。采用此格式的文件包含消费者通过增材制造或基本减材制造技术生成物理对象所需的基本信息。这包括可能需要的资源,例如纹理,以重现颜色方面的精确所需外观或材料方面的内部结构。

此格式还包括可选组件,这些组件基于生成物理对象所需的最小组件集构建。其中包括指定打印作业控制指令的能力,描述以联锁或不相交方式同时生成的对象的组装等。

最后,3MF 文档格式实现了开放包装约定规范指定的支持数字签名和核心属性的通用包装功能。

## 1.1. 软件包

3MF 文档格式必须使用 ZIP 存档作为其物理模型。开放包装约定规范描述了包装模型,即包装在内部如何通过零件和关系来表示。

ZIP 存档必须遵循.ZIP [文件格式规范](#)由 PKWARE Inc. 提供。ZIP 档案中代表 3MF 文档的文件必须使用压缩方法Deflate ( “8 - 文件已压缩”)或以未压缩形式存储 ( “0 - 文件已存储 (无压缩)” ) ,符合 OPC 规范 ( “附件 C, (规范性)

ZIP Appnote.txt 说明” ) 。消费者必须同时支持 PKWARE ZIP64™ 扩展和流式扩展。但是,由于一些较旧的 3MF 消费者无法使用 ZIP64 文件,因此如果 3MF 数据合适,生产者应该生成纯 ZIP 文件,并且仅在必要时生成 ZIP64。为了在流式模式下实现这一点,生产者可以在 ZIP 本地文件头中预先分配一个自定义块,ID 为 (0x9999),并在完成压缩文件块的写入后,如果发现 ZIP64 扩展是必要的,则可以选择用 ZIP64 扩展块覆盖它。

3MF 文档格式包含一组定义明确的部分和关系,每个部分和关系在文档中都发挥着特定作用。该格式还扩展了软件包功能,包括数字签名和缩略图。

## 第 2 章 部件和关系

开放包装约定规范中描述的包装约定可用于携带任何有效载荷。

有效载荷是包内相互依赖的部分和关系的完整集合。本规范定义了一种包含 3D 对象定义及其支持文件的特定有效载荷:3D 有效载荷。

包含 3D 有效载荷并遵循本规范所述规则的 OPC 包称为 3MF 文档。3MF 文档的生产者和消费者可以根据本规范实现自己的解析器和制造设备。

### 2.1. 3D 载荷

具有 3D 模型根部分的有效载荷称为3D 有效载荷。3MF文档中可以有多于 3D 有效载荷,但只能有一个主 3D 有效载荷。

#### 2.1.1. 3D零件与有效载荷关系

定义了一种特定的关系类型来标识 3MF 文档中 3D 有效载荷的根: 3MF 文档 StartPart 关系。主要3D 有效载荷根是 3MF 文档 StartPart 关系引用的 3D 模型部分,用于在包中查找主要 3D 有效载荷。3MF 文档 StartPart 关系必须指向标识 3D 有效载荷根的 3D 模型部分。

有效载荷包括处理 3D 模型部件所需的全套部件。制造 3D 有效载荷中描述的对象的所有内容都必须包含在 3MF 文档中。表 2-1 列出了 3MF 文档中可以找到的部件。这些部件的关系和内容类型在附录 C “标准命名空间和内容类型”中定义。每个部件都必须使用附录 C 或 3MF 扩展规范中指定的适当内容类型 (请参阅[2.3.1.支持版本控制和可扩展性](#)) 。

3D 有效负载中包含的部分通过关系明确链接到 3D 有效负载根。除非扩展中另有规定,否则 3MF 文档不得引用 3MF 文档包外部的资源。有关关系的更多信息,请参阅开放打包约定规范。

3D 负载中的部件必须使用下面描述的适当关系之一来建立负载中两个部件之间的关系。从一个部件到第二个部件,给定关系类型的关系不得超过一个。关系类型在附录 C “标准命名空间和内容类型”中定义。

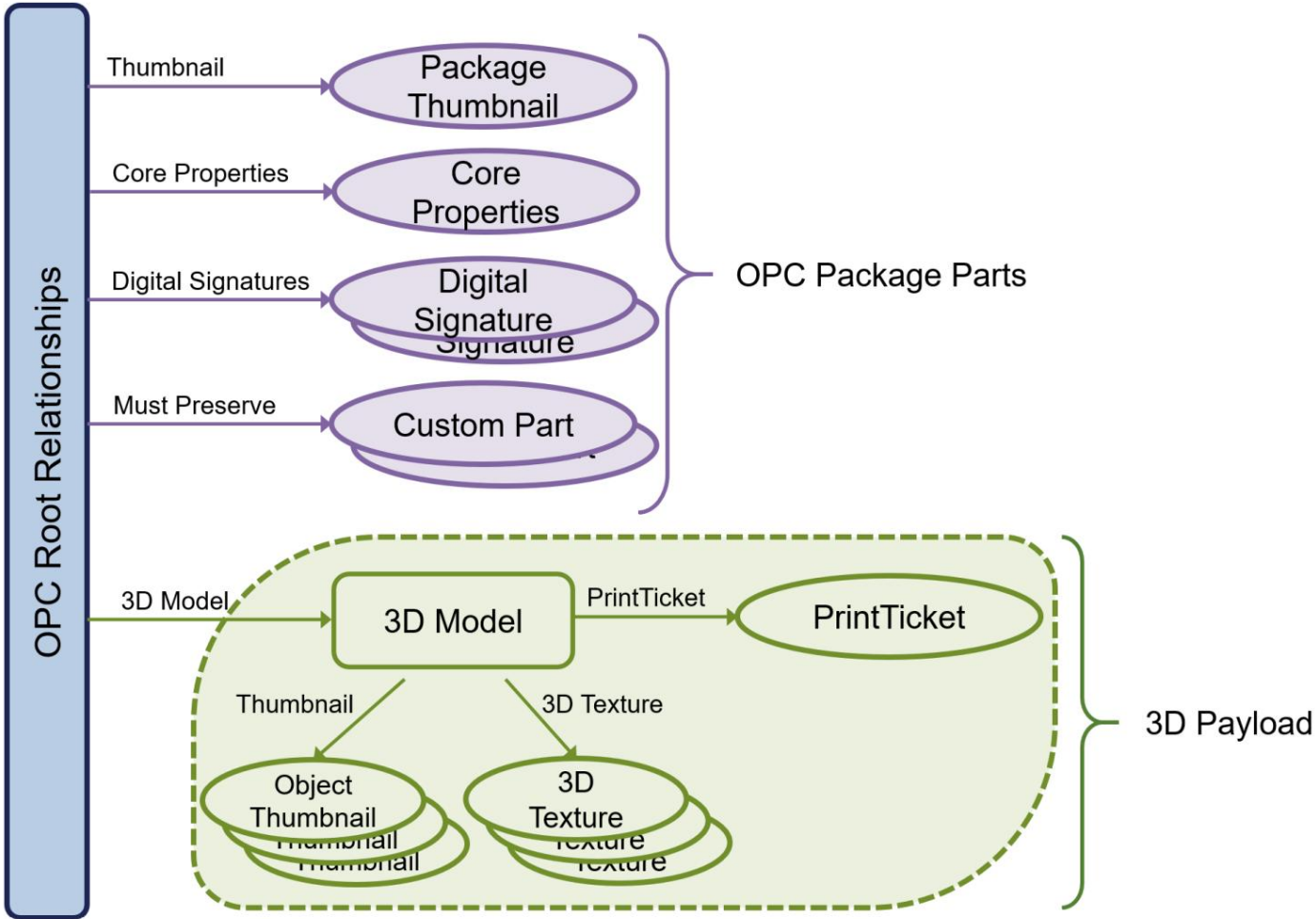
生成关系的生产者必须在 3MF 文档中包含以下任何一项的目标部分

关系类型:PrintTicket、StartPart 和 Thumbnail。如果 3MF 文档中未包含该部分,则访问具有这些关系类型之一的任何关系的目标部分的消费者必须生成错误。

表 2-1. 3MF 文档部分

姓名	描述	关系来源	必选/可选
3D模型	包含一个或多个用于制造的 3D 对象的描述。	包裹	必需的
核特性	包含各种文档属性的OPC部分。	包裹	选修的
数字的签名起源	包中数字签名的根源是 OPC 部分。	包裹	选修的
数字的签名	每个 OPC 部分都包含一个数字签名。	数字的签名起源	选修的
数字的签名证书	包含数字签名证书的OPC部分。	数字的签名	选修的
打印票据	提供在 3D 模型部分中输出 3D 对象时要使用的设置。	3D模型	选修的
包裹缩略图	包含表示包中的 3D 对象或整个包的小型 JPEG 或 PNG 图像。	包裹	选修的
目的缩略图	包含代表 3D 模型中的 3D 对象的小型 JPEG 或 PNG 图像。	3D模型	选修的
3D 纹理	包含用于将颜色应用于 3D 对象的纹理 3D 模型部分（可供扩展）	3D模型	选修的
自定义部件	与元数据关联的 OPC 部件	包裹	选修的

图 2-1. 典型的 3MF 文档



2.1.2. 3D模型部分

3D 模型部件包含要通过 3D 制造工艺制造的一个或多个对象的定义。3D 模型部件是 3D 有效负载的唯一有效根。

3D 模型部件包含两个部分：一组包含对象和材料的资源定义，以及一组实际要构建的特定项目。3D 模型部件的内容类型在附录 C “标准命名空间和内容类型”中定义。

2.1.3. 包缩略图和对象缩略图部分

包缩略图和对象缩略图是表示整个 3MF 文档内容的小图像。缩略图使外部代理能够轻松查看 3MF 文档的内容。可以通过引用根模型关系文件中的缩略图来为整个包定义包缩略图。可以使用对象缩略图属性为单个对象定义对象缩略图。这些缩略图部分必须采用 JPEG 或 PNG 格式。

3MF 文档中的所有缩略图都必须通过缩略图关系进行引用。

有关缩略图部分关系类型的更多信息，请参阅第C.2节 “关系类型”。

2.1.4. 打印票证部分

PrintTicket 部件为打印消费者提供用户意图和设备配置信息。PrintTicket 部件只能附加到 3D 模型部件，并且每个 3D 模型部件必须附加不超过一个PrintTicket。PrintTicket格式由特定的消费者环境控制。例如，对于在 Microsoft Windows 上打印，有效的 PrintTicket 设置在 3D 制造规范的打印架构关键字中指定。

如果没有提供 PrintTicket，或者消费者不支持所提供的 PrintTicket，则消费者必须自行申请

它自己的默认值。

### 2.1.5. MustPreserve 关系

生产商可以将自定义 OPC 部件添加到 3MF 包中。例如,软件供应商可以包含有关由唯一 ID 引用的对象的注释。

MustPreserve 关系表明消费者在修改 3MF 文件时应该保存相关部分,即使他们不了解如何处理数据。

必须以这种方式保留的部分必须通过 MustPreserve 关系与包根相关联。

如果自定义 OPC 部分未被 MustPreserve 关系引用,则消费者在修改 3MF 时不应保留这些部分,因为它们不再有效。

以下示例说明如何添加 MustPreserve 关系:

```
<?xml version= 1.0 encoding= UTF-8 ?>
<Relationships xmlns= http://schemas.openxmlformats.org/package/2006/relationships ><Relationship Target= /3D/
3dmodel.model Id= rel-1
类型= http://schemas.microsoft.com/3dmanufacturing/2013/01/3dmodel /><Relationship Target= /
Metadata/thumbnail.png Id= rel0
类型= http://schemas.openxmlformats.org/package/2006/relationships/metadata/thumbnail /><Relationship Target= /
Metadata/MustPreservePart.txt Id= rel-2
类型= http://schemas.openxmlformats.org/package/2006/relationships/mustpreserve /></Relationships>
```

## 2.2. 零件命名建议

3MF 文档的生产者和消费者通过名称引用部件并使用关系名称来识别相关部件的用途。开放包装约定规范描述了部件名称的语法。但是,仅遵循这些规则可能会导致用户难以理解包装。例如,用户必须打开每个关系部件才能知道哪些部件是准确制造 3MF 文档所必需的。

通过根据明确定义、人性化易读的惯例选择部件名称,生成的包将更易于浏览,并且特定部件也更容易找到。部件名称仍必须符合开放打包惯例规范中指定的语法。

建议 3MF 文档制作者使用以下部分命名约定:

- 3D 模型部件名称应包含两段,第一段为 “/3D/”,第二段的最后一段带有扩展名 “.model”,例如 “/3D/3dModel.model”。
- PrintTicket 部件名称应通过与 3D 模型部件的关系进行关联,并包含三个段,使用 “/3D/Metadata/”作为前两个段,扩展名为 “.xml”。例如,“/3D/Metadata/Model\_PT.xml”。
- 3D 纹理部件名称应包含三个段,使用 “/3D/Textures/”作为前两个段,例如 “/3D/Textures/coloring.png”。3D 纹理部件必须通过适当的关系与 3D 模型部件相关联。
- 与 3D 有效载荷相关的任何非标准部件的名称都应包含 3 个段,使用 “/3D/Other/”作为前两个段。

部件名称必须使用绝对路径,即所有路径都以 “/”开头。部件名称不得为空或以句点开头(例如 “/3D/.png”或 “/3D/”)。

当部件名称以 Unicode 表示时,它应该用部件 URI 语法表示,如开放包装约定规范的第 9.1.1.1 节部件名称语法中所述。

## 2.3. 3MF 文档标记

3MF 文档标记旨在促进生成或使用 3MF 文档的兼容系统的独立开发。

3MF 文档标记的设计反映了两个有时相互竞争的目标之间的权衡：

1. 3MF 文档标记应简洁 ;也就是说,它应仅包含制造具有完整保真度的常见 3D 对象所需的最小原始操作和标记构造集。规范中的冗余增加了独立实现引入意外不兼容性的机会。冗余还会增加实现和测试的成本,并且通常还会增加所需的内存组件。
2. 3MF 文档标记应该紧凑 ;也就是说,最常见的原语应该紧凑表示。臃肿的表示会影响处理 3MF 文档的系统的性能。随着字节数的增加,通信时间也会增加。虽然可以使用压缩来改善通信时间,但它无法消除由臃肿表示引起的性能损失。

### 2.3.1. 支持版本控制和扩展性

3MF 文档标记的设计考虑到了该规范的发展。它还允许第三方扩展标记。

扩展是 3MF 的关键部分,因此,此核心规范尽可能地狭窄。高级功能以扩展的形式构建,使用点菜模型,生产者可以明确说明使用哪些扩展 (通过在 <model> 元素中声明匹配的 XML 命名空间),消费者可以明确说明他们支持哪些扩展,因此链中的其他工具知道哪些部分将被忽略。版本控制是同时完成的,因为命名空间将更新以反映版本更改。因此,版本控制对于核心规范和每个扩展都是独立进行的,并且可以通过检查其命名空间来确定每个扩展的版本。

扩展规范必须包含此核心规范的一个或多个目标版本,以限制可能的配置数量。生产者可以根据特定 3MF 文档的要求指定某些扩展,在这种情况下,不支持这些扩展的消费者必须无法编辑或制作该文档,而不是忽略扩展命名空间。

在此核心 XSD 架构中 (参见[附录 B.1. 3MF XSD 架构](#)),扩展点已以 <any> 元素和 <anyAttribute> 的形式明确输入 (在本规范后面的元素图中也可见)。这些扩展点必须来自其他命名空间,这些命名空间应指向一种查找适当规范和随附 XSD 架构的方法。

供应商可能会定义私有 3MF 扩展。私有命名空间 (即未经 3MF 联盟批准的命名空间)的规范需要由生态系统中的各方协商。

### 2.3.2. XML 用法

本规范中定义的所有 XML 内容必须符合以下验证规则：

1. XML 内容必须使用 UTF-8 进行编码。如果任何此类部分包含编码声明 (如第 4.3.3)中规定,声明不得使用除 UTF-8 之外的任何其他编码。
2. XML 1.0 规范允许使用数据类型定义 (DTD),这会导致拒绝服务攻击,通常是通过使用内部实体扩展技术。为了缓解这种潜在威胁,DTD 内容不得在本规范定义的 XML 标记中使用,并且消费者应将 DTD 内容的存在视为错误。
3. XML 内容必须符合本规范中定义的相应 XSD 架构。特别是,XML 内容不得包含未在相应 XSD 中明确定义的命名空间中提取的元素或属性,除非 XSD 允许从任何命名空间中提取的元素或属性出现在 XML 标记中的特定位置。
4. XML 内容不得包含来自 “xml”或 “xsi”命名空间的元素或属性,除非它们在 XSD 模式或规范中的其他方式中明确定义。
5. XML 内容必须使用 en-us 语言环境生成和解析,尤其是包含以下内容的值：十进制数据。



2.3.3. 标记模型

3MF 文档标记是一种基于 XML 的标记语言,使用元素、属性和命名空间。3MF 文档标记的架构仅包含元素及其属性、注释和空格。

2.3.3.1. XML 命名空间

3MF 文档核心XML 命名空间,即 3D 模型部件标记中元素和属性使用的主要命名空间,在附录 C “标准命名空间和内容类型”中给出。本规范中未定义的任何元素和属性都必须以它们所属的 3MF 扩展对应的命名空间作为前缀。

提醒一下,元素上的非默认 XML 命名空间会自动应用于该元素的任何属性 (除非有另一个命名空间作为前缀),但不适用于子元素,因此所有子元素都必须单独加上前缀。

任何属于 anyattribute 扩展点的属性都必须以其对应的命名空间作为前缀 (因为所有此类扩展点都在 XSD 模式中为所需命名空间指定了“其他”)。

消费者或编辑者必须忽略其不明确支持的命名空间中的所有 XML 节点和属性。

2.3.4. 空格

3MF 文档允许在标记中灵活使用空格。只要允许使用单个空格字符,就可以使用多个空格字符。3MF 文档标记不得使用 xml:space属性。此外,如果 3MF 文档架构指定允许空格折叠的属性类型,则属性值中的前导空格和尾随空格可以与依赖于 XML 架构规范中指定的空格折叠行为的其他空格一起使用。

注意:请查阅 3MF 模式以了解允许的确切空格。

2.3.5. 语言

可以使用xml:lang属性来标识 3MF 文档内容的语言 (通常用于元数据中提供的内容),该属性的值由子元素和后代元素继承。此属性在 W3C XML 规范中定义。当内容的语言未知时,必须使用值“und”(未确定)

用过的。

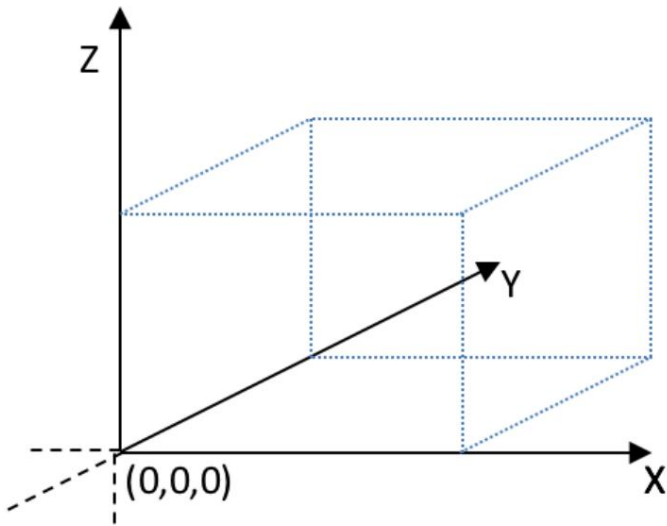
第 3 章 3D 模型

本规范中的模型是指通过 3D 制造流程作为单一操作创建的一个或多个对象。它可能包括单个对象、多个同质对象、多个异质对象、完全封闭在另一个对象中的对象或互锁且不可分离的组件中的多个对象。

3.1. 坐标空间

本规范中的坐标基于右手坐标空间。生产者和消费者必须定义并将坐标空间的原点映射到设备输出字段 (例如托盘、平台或床)的左下角,其中 x 轴沿输出字段的右侧增加,y 轴沿输出字段的后部增加,z 轴沿输出字段的顶部增加。生产者和消费者必须使用 <model> 元素中指定的坐标空间的单位分辨率。

图 3-1. 坐标空间



3.2. 相对方向和测量

本规范中的相对方向定义如下。术语“顶部”是指坐标空间中具有最大可打印 Z 值的 XY 平面。术语“底部”是指坐标空间中最小可打印 XY 平面,定义为 Z 值为 0 的 XY 平面。这通常与打印床表面重合。术语“左侧”是指坐标空间中最小可打印 YZ 平面,定义为 X 值为 0 的 YZ 平面。术语“右侧”是指坐标空间中具有最大可打印 X 值的 YZ 平面。术语“前”是指坐标空间中最小可打印 XZ 平面,定义为 Y 值为 0 的 XZ 平面。术语“后”是指坐标空间中具有最大可打印 Y 值的 XZ 平面。

这些术语也可能适用于模型,在这种情况下,它们是在转换到本规范中定义的坐标空间时相对于模型的边界框定义的。

生产者和消费者必须根据本规范中定义的坐标空间来解释坐标。

3.3. 3D 矩阵

当需要对对象进行旋转、缩放或平移变换时,可使用行主仿射3D 矩阵(4x4)。矩阵不应为奇异矩阵或近似奇异矩阵。

变换的形式为,其中仅指定前 3 列值。最后一列从未提供,并具有固定值 0.0、0.0、0.0、1.0。当指定为属性值时,矩阵的形式为“m00 m01 m02 m10 m11 m12 m20 m21 m22 m30 m31 m32”,其中每个值都是任意精度的十进制数。

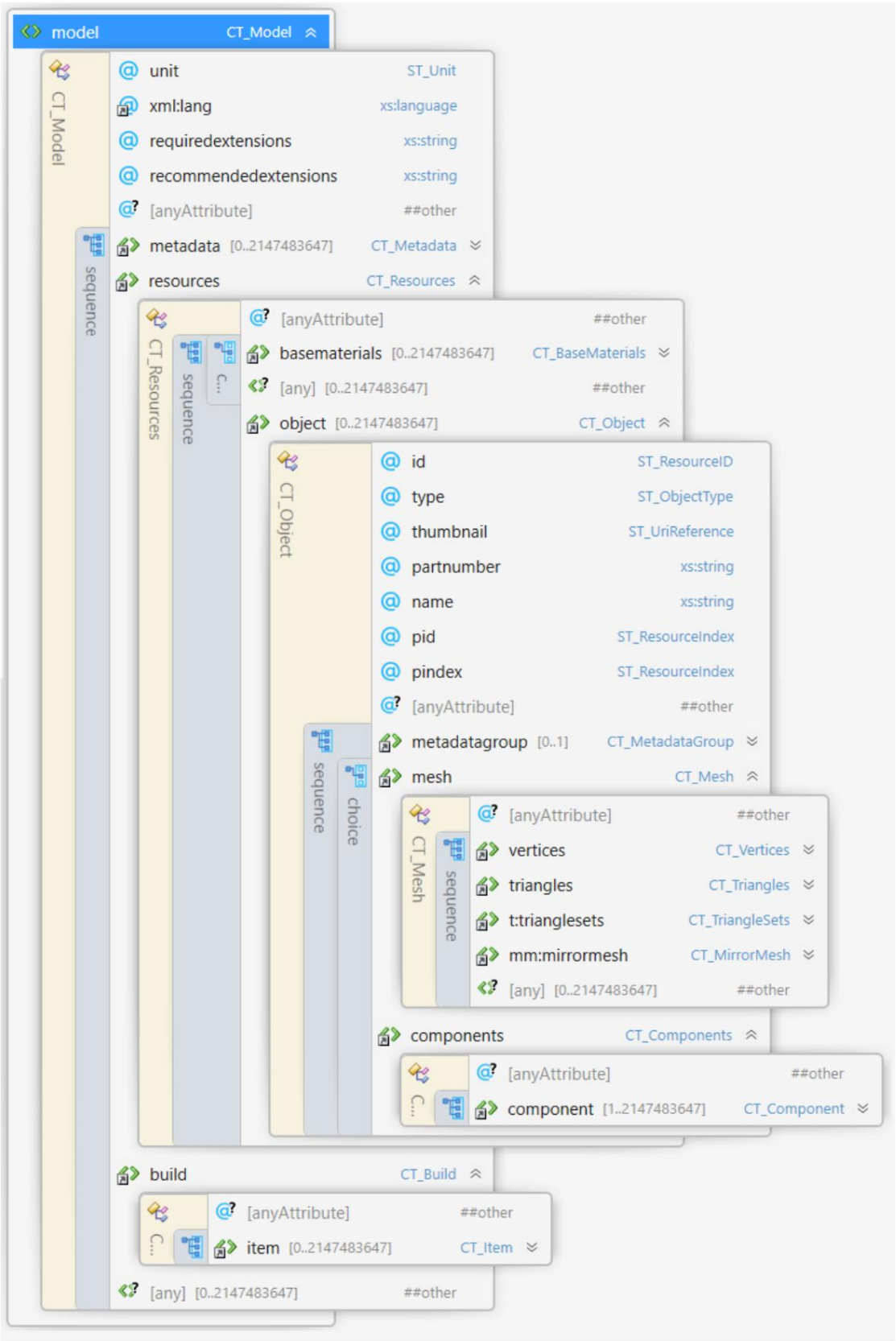
$$\begin{bmatrix} m00 & m01 & m02 & 0.0 \\ m10 & m11 & m12 & 0.0 \\ m20 & m21 & m22 & 0.0 \\ m30 & m31 & m32 & 1.0 \end{bmatrix}$$

变换应用于对象定义的形状。当应用具有负行列式的变换时,生成的形状不得改变其体积的符号。

将所有变换应用到对象后,模型应该具有正体积,并且应该位于坐标空间的正八分圆内。

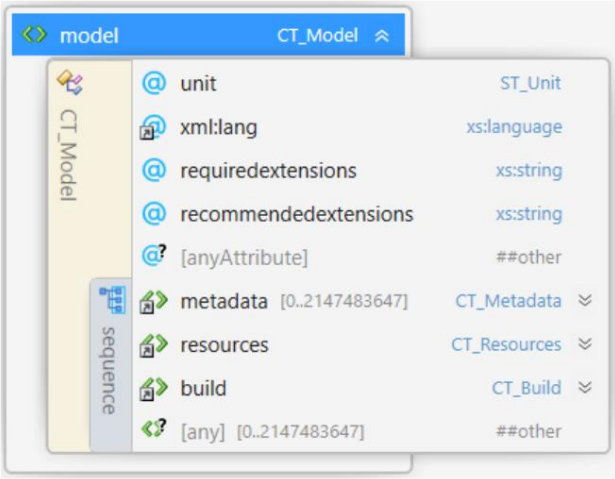
3.4. 模型

图3-2 :3MF模型XML结构概览



此 XML 规范旨在与简单的、仅向前的解析器一起使用,并且定义的元素排序支持这一点。除非扩展明确允许,否则生产者必须在文档中的其他地方引用每个元素之前对其进行定义。

元素<模型>



属性

姓名	类型	使用默认	注解
单元	ST_单位	毫米	指定用于解释模型中所有顶点、位置或测量值的单位。有效值为微米、毫米、厘米、英寸、英尺和
			仪表。
xml:语言	xs:语言		指定当前元素及其子元素使用的默认语言。语言根据 RFC 3066 指定。
必需扩展	xs:字符串		以空格分隔的命名空间前缀列表,表示处理文档所需的扩展集。如果编辑器和制造设备不支持所需的扩展,则不得处理该文档。
推荐扩展xs:string			以空格分隔的命名空间前缀列表,表示建议用于处理文档的扩展集,以达到文档的设计意图。编辑器和制造设备应警告并通知用户它们不支持推荐的扩展,并要求输入如何继续。必需的扩展必须
			不建议同时使用。

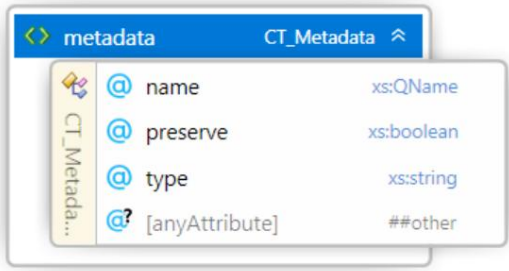
@anyAttribute

<model> 元素是 3D 模型部件的根元素。3D 模型部件中必须只有一个 <model> 元素。模型可以有零个或多个子元数据元素（有关更多信息,请参阅[3.4.1. 元数据](#)）。模型必须有两个附加子元素:<resources> 和 <build>。<resources> 元素提供了一组定义,可以从中得出 3D 对象。<build> 元素提供了一组应作为作业的一部分实际制造的项目。

生产商不应该要求扩展,除非文档在没有扩展数据的情况下会失去关键含义。允许消费者忽略不受支持的扩展可以实现更优雅的回退。必需的扩展可能会取代核心规范的要求。但是,与可选扩展一起使用时,必须完全支持核心规范。

3.4.1. 元数据

元素<metadata>



属性

姓名	类型	使用	默认	注解
姓名	xs:QName必填			包含由以下定义的元数据的知名名称 本规范（见下表 3-1）或供应商定义的 元数据,必须以有效的 XML 作为前缀 在 <model> 元素上声明的命名空间名称。
保存	xs:布尔值			非零值表示生产者希望消费者 保存此 3MF 的修改版本时保留此值
类型	xs:字符串			表示存储在中的数据的数据的 XML 类型的字符串 元数据值。

@anyAttribute

3MF 文档的制作者应在以下情况下以元数据的形式提供有关文档的其他信息：  
根 <model> 元素。

与 <model> 关联的元数据可能包含一组众所周知的值。3MF 文档中没有  
命名空间名称必须限制为本规范定义的名称和值。如果名称值未在本  
规范,它必须以 <model> 上的 XML 命名空间声明的命名空间名称作为前缀  
不是从默认命名空间中提取的元素。

本规范定义的众所周知的元数据名称和值包括：

表 3-1. 元数据值

语境	姓名	评论
模型	标题	3MF 文档的标题
	设计师	本文档设计者的姓名
	描述	文档描述
	版权	与本文件相关的版权
	许可条款	与此文档相关的许可信息
	等级	与本文件相关的行业评级
	创建日期	源应用程序创建此文档的日期
	ModificationDate	该文档最后修改的日期
应用	应用	最初创建此文档的源应用程序的名称

可选的 “type”属性允许将值设为任何数据.type 的默认值假定为 “xs:string” 。如果  
类型不存在,则 <metadata> 值被假定为任意字符串。但是,如果指定了类型,则  
必须包含表示值中包含的数据的内置简单 XML 类型的名称。

简单 XML 类型包括 “xs:anySimpleType”指定的任何内置原始 XML 类型或派生 XML 类型。

生产者不得创建具有相同名称的多个元数据元素。希望与其他消费者进行互操作的生产者应发布命名空间 URI 和一组定义明确的元数据名称和预期内容,以便消费者能够以预期的方式运行。

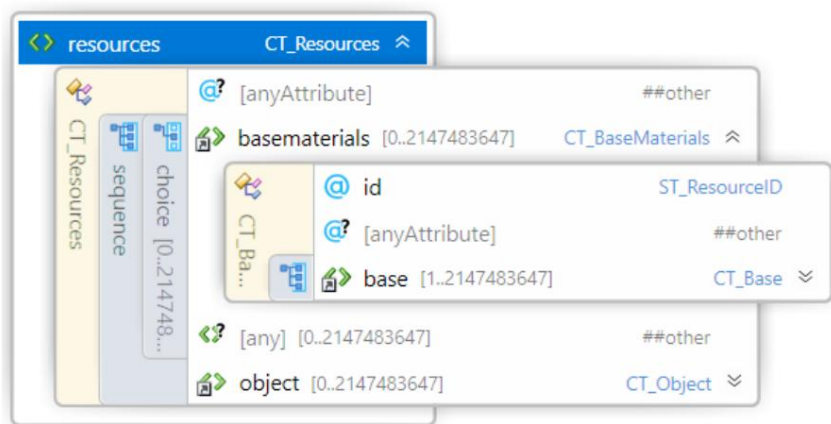
消费者应该忽略任何他们不认识的名称的元数据,通常来自该规范的未来版本或无法识别的生产者或目标消费者。

生产者可以使用保留属性指示应保留某些元数据值。默认值假定为 0 或 false。当保留属性为 1 或 true 时,修改 3MF 文件的消费者应保留原始元数据值,即使其引用的数据被修改。元数据应在其关联元素的整个生命周期内保留,例如,如果删除了 <Item>,则应同时删除关联的元数据。

附录 B.2 包括支持自定义元数据的 3MF 示例。

### 3.4.2. 资源

元素<resources>



`<resources>` 元素是整个 3D 对象定义组成部分库的根元素。

本规范中将对象、属性和材料统称为资源。

每个资源可能依赖于其他资源来实现其完整定义。例如,对象资源可能引用材料资源,甚至引用其他对象资源来完整描述 3D 对象。

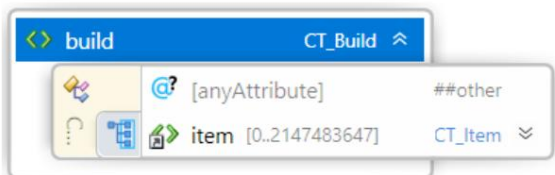
对象资源表示可以制造但不一定能制造的单个 3D 对象。

实际要制造的对象是从 <build> 元素的 <item> 子元素引用的。对象被定义为资源,主要是为了帮助模块化设计和组件的重复使用,从而压缩整体标记大小。

资源 ID 在模型内必须是唯一的。

### 3.4.3. 构建说明

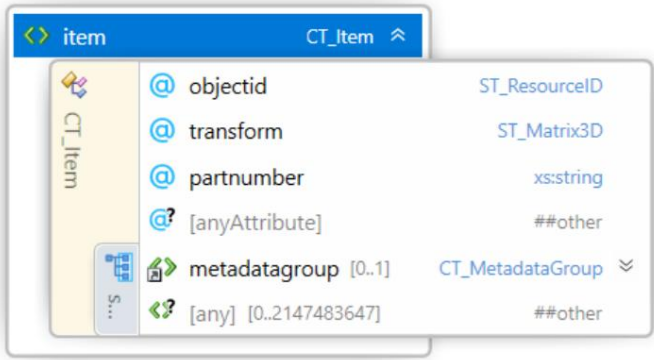
元素<build>



<build> 元素包含一个或多个要在处理作业过程中制造的项目。消费者不得输出任何未由 <item> 元素引用的 3D 对象。

#### 3.4.3.1. 项目元素

元素<item>



属性

姓名	类型	使用	默认	注解
对象标识	需要ST_ResourceID			引用具有匹配 id 的 <object> 元素属性值
转换	ST_Matrix3D			矩阵变换（参见 <a href="#">3.3. 3D 矩阵</a> ）应用于要输出的项目。
零件号	xs:字符串			该项目的唯一标识符。应由如果仅改变转换,则编辑器。
@anyAttribute				

元素

姓名	类型	使用默认	注解
元数据组CT_MetadataGroup			指定的可选 CT_Metadata 元素组在模型的元数据部分。

<item> 元素标识 3D 制造设备要输出的一个对象资源。消费者必须申请输出对象之前的变换。

<item> 元素可能包含一个 <metadatagroup> 元素,该元素包含一个或多个供应商提供的 <metadata> 元素。

3MF 文档可能同时包含多个要制造的对象。这些项目的排列方式构建被视为默认;消费者可以重新排列制造物品,以便更好地包装构建体积。有时物体被排列在坐标空间中,以便以互锁的方式制造;这些物体的生产者应该把它们作为组件（见[4.2.组件](#)）收集起来,作为3D制造设备一定不能将对象的各个组件相对彼此进行变换。

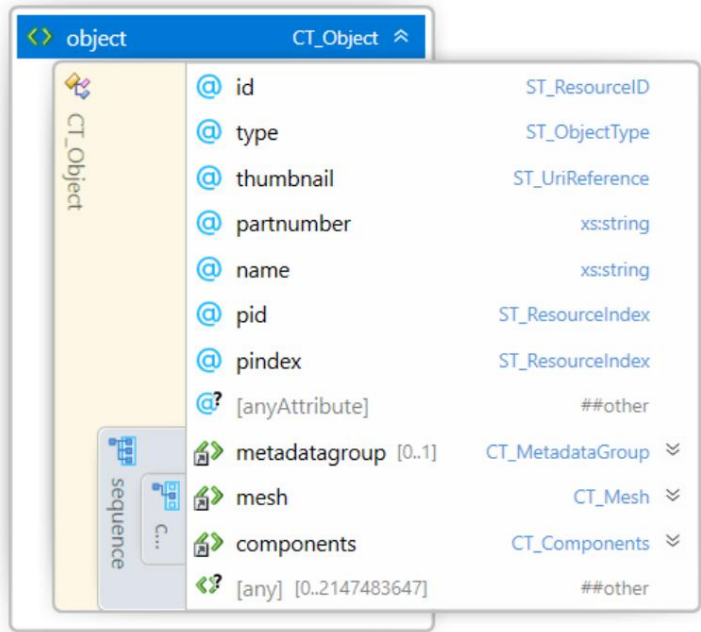
如果项目重叠,3D 制造设备必须使用正填充规则（如第 4.1.1 节所述）来解决最终几何图形上的歧义。如果任何重叠项具有定义的属性,则最终几何图形上的属性重叠体积取自最后一个重叠项的属性。如果最后一个项没有在重叠体积,属性不得应用。

注意 :项目不得直接或递归引用 “其他”类型的对象。

## 第 4 章 对象资源

对象资源描述可重复使用的对象,这些对象可以输出（通过引用）或组合成更复杂的对象或程序集。

元素<object>



属性

姓名	类型	使用	默认	注解
ID	ST_资源ID	必需的		定义此对象的唯一标识符。
类型	ST_对象类型		模型	指定模型中对象的功能。有效值为“model”、“solidsupport”、“support”、“surface”，或“其他”。如果它们是不需要的。消费者可以生成其拥有支持对象来代替提供的支持对象,但是不推荐这样做。
缩略图	ST_Uri引用			JPEG 或 PNG 类型的对象缩略图路径表示对象的渲染图像。
零件号	xs:字符串			部件编号,编辑人员应在修改和派生对象的过程。
姓名	xs:字符串			对象的名称以提高可读性。
进程号	ST_资源ID			引用属性组元素 匹配 id 属性值（例如 <basematerials>）。 如果指定了 pindex,则为必需。
品目索引	ST_资源索引			引用属性中的从零开始的索引 pid 指定的组。此属性用于构建目的。

@anyAttribute

元素

姓名	类型	使用默认	注解
元数据组CT_MetadataGroup			指定的可选 CT_Metadata 元素组在模型的元数据部分。

对象资源由 <object> 元素定义。<object> 元素具有属性组的属性和应用于整个对象的特定属性成员,除非被后代元素覆盖,



例如 <triangle> 元素或组件引用的 <object> 元素。如果此对象包含任何已分配属性的三角形,则该对象必须指定 pid 和 pindex,以作为任何未指定属性的三角形的默认值。如果未分配任何属性,则对象属性的选择权将留给消费者。

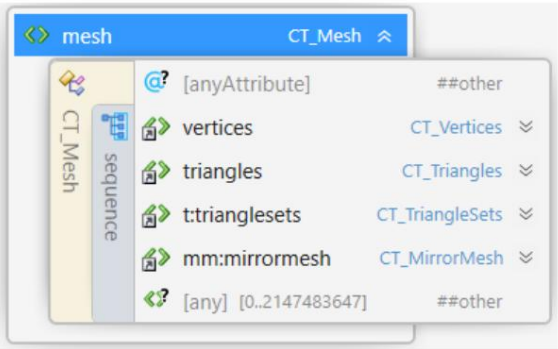
包含组件的对象会忽略对象类型,因为类型总是被后代对象覆盖。  
生产者不得为包含组件的对象分配 pid 或 pindex 属性。这可确保没有材质的对象不会因在多个对象中被引用为组件而被拆分为具有不同材质的两种表示形式。

对象缩略图必须与模型部分具有适当的缩略图关系,如第 2.1.1 节所述。  
为了实现向后兼容,消费者可以通过 3D 纹理关系找到与对象关联的其他缩略图。制作者必须将具有缩略图关系的缩略图存储在关联对象缩略图部分中。

部件编号旨在用于跟踪工具链中可能已修改的对象。在编辑或处理 3MF 文档时,应尽可能保留这些部件编号,为拆分成块的对象复制这些部件编号,从组合的对象中删除这些部件编号,并为已修改的对象保留这些部件编号。

4.1. 网格

元素<网格>

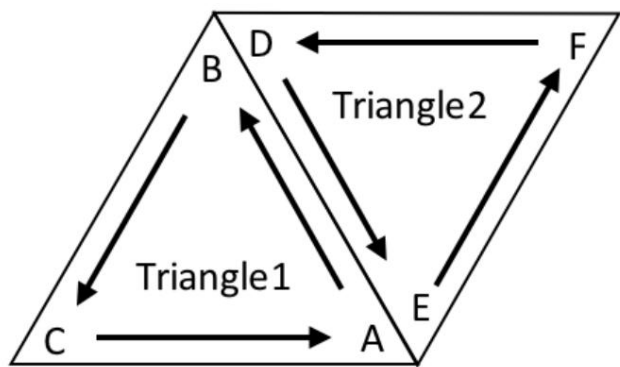


<mesh> 元素是物体体积的三角网格表示的根。它包含一组顶点和一组三角形。

如果网格位于“模型”或“solidsupport”类型的对象下,则它必须具有：

- 流形边:网格中的每个三角形边都与另一个三角形的边共享公共顶点端点。边的顶点端点通过顶点的索引定义。
- 一致的三角形方向:网格中每对相邻三角形的面法线方向都必须相同,朝向网格外部,这意味着共享边上的顶点声明顺序必须相反。例如,如果 Triangle1 有顶点 ABC, Triangle2 有顶点 DEF,并且 Triangle1 和 Triangle2 共享 AB/DE 边,则必须满足顶点 A=E 和顶点 B=D 的情况（参见下图 4-1）。在本规范中,三角形面法线（对于三角形 ABC,按此顺序）定义为向量叉积 (B - A) x (C - A) 方向上的单位向量。例如,图 4-1 中显示的三角形的法线指向网格外。
- 朝外的法线:所有三角形的法线必须指向远离物体内部的方向。  
根据下一节中定义的正填充规则,具有负体积的网格将不会被打印（或将成为空隙）。因此,结合前两个规则,网格是一个连续的表面,没有孔、间隙、开放边缘或不可定向的表面（例如克莱因瓶）。

图 4-1. 一致的三角形方向



允许“支撑”、“表面”和“其他”类型的对象打破这些条件。“其他”类型的对象不得在构建部分中被引用,无论是递归引用还是直接引用。

所有网格都应该具有:

- 最小自相交:生产者不应该生产自相交或重叠物体;消费者必须接受自相交网格和重叠物体,但将“模型”和“固体支持”类型视为用正填充规则填充(参见下一部分)。
- 非退化:所有三角形都应具有非零面积,因为这会在计算三角形法线时产生数值不稳定性。

制造设备可以使用实体材料或内部结构填充“模型”或“实体支撑”类型的对象,具体情况可能在打印单中指定。如果对象类型为“支撑”,制造设备可以决定(可能考虑到打印单参数)支撑的实际表示(例如厚度)。无论如何,制造设备不得填充此网格。

“support”或“solidsupport”类型的对象应该只在包含一个或多个其支持的“model”类型的组件的对象中引用。

支撑结构(“solidsupport”和“support”类型)可能会被忽略或被自动生成的支撑所取代,但不推荐这样做。

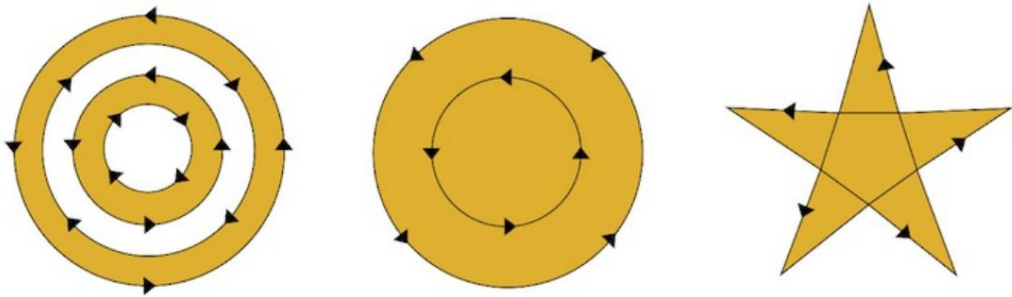
4.1.1. 填充规则

由于网格可能会重叠或自相交,因此必须指定填充规则来指示哪些封闭区域位于体积的“内部”,哪些位于体积的“外部”。一组网格的封闭体积是通过应用填充算法来定义的。填充算法决定几何形状相交区域如何组合形成一个区域。填充规则在 2D 和 3D 中的定义相同,因此此处的示例将显示 2D 等效规则以简化图形。

消费者可以利用这一事实,在切片后应用填充规则,从而简化算法,只需在 2D 中操作即可。按照惯例,2D 图形逆时针旋转时面积为正,因此沿边看时,局部“外侧”在右侧,相当于三角形的局部“外侧”是法线方向。

在 3MF 中,应用的填充规则始终是正填充规则,该规则通过从点向任意方向绘制一条射线到无穷远,然后检查它与形状相交的位置来确定点的“内部性”。从零开始计数,每次射线从局部“内部”穿过局部“外部”时加一,反之亦然。计数交叉点后,如果结果为一或更大,则该点全局位于体积内部;否则,它位于外部。

图 4-2. 使用 Positive 算法填充



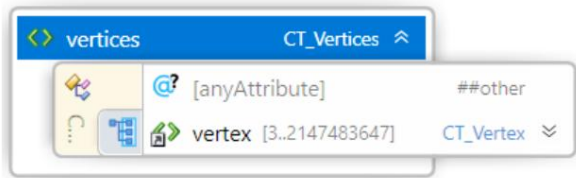
4.1.2. 重叠顺序

因为网格可能会与自身重叠,所以必须指定重叠顺序来指示网格的哪些属性重叠表面应用。

当网格表面上的两个三角形相互重叠时,重叠区域的属性取自最后一个重叠三角形的属性。如果最后一个重叠三角形中未定义任何属性,则属性应该不适用于三角形的重叠区域。为避免这种情况,3MF 生产商不应创建重叠三角形。

4.1.3. 顶点

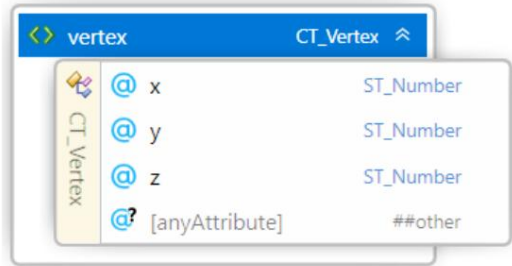
元素<vertices>



vertices 元素包含此对象的所有 <vertex> 元素。vertices 表示每个三角形的角在网格中。这些元素的顺序定义了一个隐式的基于 0 的索引,该索引由其他元素引用,例如 <triangle> 元素。生产者不应该包含重复的顶点,除非合并重复项会创建非流形边。此外,生产者应该折叠与单个顶点非常接近的顶点只要适当。为了避免整数溢出,顶点数组必须包含少于 2^31 个顶点。

4.1.3.1. 顶点

元素<顶点>



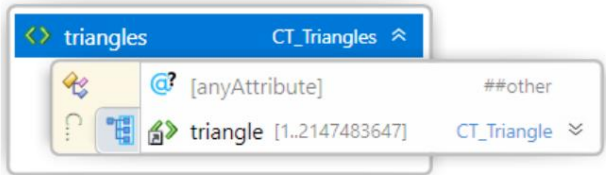
属性

姓名	类型	使用	默认	注解
+	需要ST_Number			顶点沿 X 轴的位置。
和	需要ST_Number			顶点沿 Y 轴的位置。
和	需要ST_Number			顶点沿 Z 轴的位置。
@anyAttribute				

<vertex> 元素表示三维空间中的点,由网格中的三角形引用。表示坐标的十进制值可以记录为任意精度。生产者使用的精度不应超过其计算中产生的误差或其消费者的预期分辨率。ASCII 编码的可变精度特性与固定宽度二进制格式相比具有显著优势,有助于弥补存储效率的差异。

4.1.4. 三角形

元素<三角形>

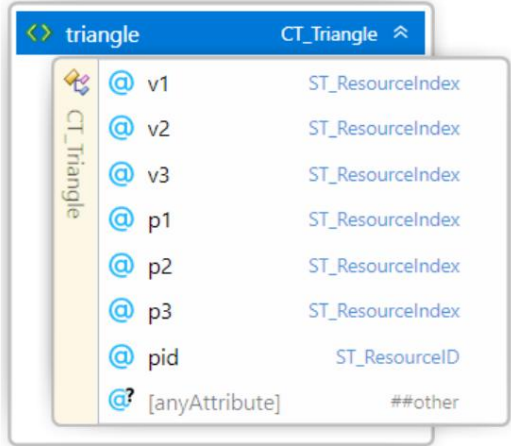


<triangles> 元素包含一组 1 个或多个 <triangle> 元素,用于描述完整的 3D 对象网格。如果对象类型为“模型”,则网格必须包含至少 4 个三角形才能形成实体。为了避免整数溢出,三角形数组必须包含少于 2^31 个三角形。

这些限制可以通过另一个必需的 3MF 扩展来取代。

4.1.4.1. 三角形

元素<三角形>



属性

姓名	类型	使用	默认	注解
v1	需要ST_ResourceIndex			引用此顶点的从零开始的索引 网格。定义三角形的第一个顶点。
v2	需要ST_ResourceIndex			引用此顶点的从零开始的索引 网格。定义三角形的第二个顶点。
v3	需要ST_ResourceIndex			引用此顶点的从零开始的索引 网格。定义三角形的第三个顶点。
p1	ST_资源索引			覆盖第一个顶点的对象级 pindex 三角形。
p2	ST_资源索引			覆盖第二个顶点的对象级 pindex 三角形。
p3	ST_资源索引			覆盖第三个顶点的对象级 pindex 三角形。
进程号	ST_资源ID			覆盖三角形的对象级 pid。
@anyAttribute				

<triangle> 元素表示网格的一个面。顶点 (v1,v2,v3) 的顺序必须在  
逆时针方向,这样三角形的面法线指向物体的外部。索引  
v1,v2 和 v3 必须不同。

应用于每个顶点 (p1,p2,p3) 的属性允许在整个三角形中定义属性梯度,其中  
属性的插值定义为线性凸组合。无法创建属性梯度的消费者  
必须将 p1 属性应用于整个三角形。无法按三角形使用属性的消费者必须  
忽略三角形属性,改用 <object> 级别属性。如果未指定 p1,则默认对象  
如果指定了属性,则将其分配给三角形。如果未指定 p2 或 p3,则将 p1 用于整个三角形。

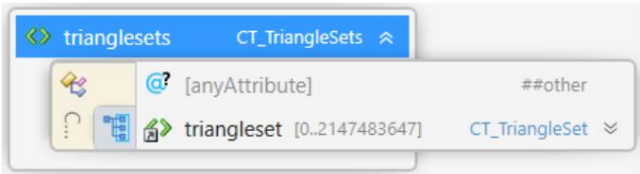
如果属性组与在对象级别定义的属性组不同,则属性组由 pid 属性指定。由于此  
应用于整个三角形,它隐式地强制这三个属性来自同一组,这意味着它们属于  
相同类型,如本规范的可能扩展所定义。

如果三角形上定义的属性来自<basematerials>组 (参见第 5 章),则它们不得形成  
梯度,因为本核心规范未定义基础材料的插值。因此 p1,p2 和 p3 必须是  
相等或未指定。材料梯度和插值方法在扩展规范中定义。

注意:三角形方向受变换行列式符号的影响,如第节所述  
4.1.

### 4.1.5. 三角形集

元素<t:trianglesets>

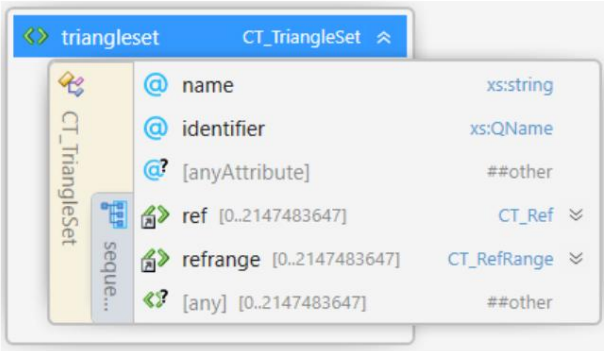


网格节点可能包含三角形集节点,该节点包含有关三角形如何分组和组织的信息。  
三角形集及其子节点必须位于三角形集的核心扩展命名空间下  
(<http://schemas.microsoft.com/3dmanufacturing/trianglesets/2021/07>)。

<t:trianglesets> 元素充当 <t:triangleset> 节点的容器。这些元素的顺序形成一个隐式的 0-  
基于索引的索引可以通过其标识符在外部引用。

4.1.5.1 三角集元素

元素<t:triangleset>



姓名	类型	使用默认	注解
姓名	xs:字符串	必需,三角形集合的可读名称。不得为空。	
标识符xs:QName		必需的	可能用于三角形集合数据的外部识别。 标识符属性在网格内必须是唯一的,并且不得空的。

三角形集包含对三角形子集的引用列表,用于应用分组操作并将属性分配给三角形。编辑应用程序可能会将此信息用于内部目的,例如颜色显示和选择工作流程。

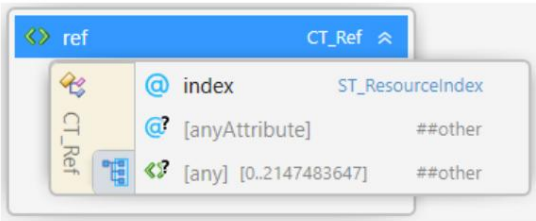
三角形集是三角形引用的集合。由于三角形集对几何形状,消费者可能会忽略该信息。

消费者必须忽略同一集中对同一三角形的重复引用。生产者应避免创建一个集合中重复引用同一个三角形。

注意:根据用途,单个三角形可以被多个三角形集引用。有许多应用程序期望网格的分割不相交。如果生产者的应用程序不期望分割不相交,则生产者应该尝试输出不相交的集合否则,需求就会发生变化。如果消费者需要将通用三角形集合转换为不相交的分割,则优先级应给予出现三角形的最后一组。

4.1.5.2 三角形集引用

元素<t:ref>

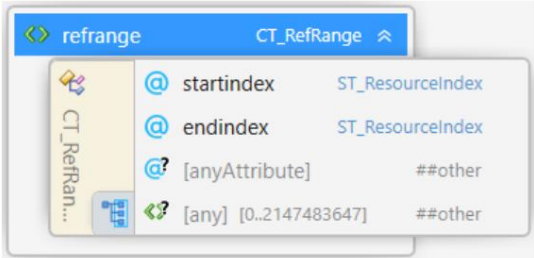


姓名	类型	使用	默认	注解
索引ST_ResourceIndex		必需		引用网格三角形列表中的索引。

三角形集中的 <ref> 元素指的是三角形中包含的从零开始索引的 <triangle> 元素节点。

4.1.5.3 三角组参考范围

元素<t:refrange>



姓名	类型	使用	默认	注解
startindex	ST_ResourceIndex必需			引用引用范围的起始索引 网格三角形列表。
endindex	ST_ResourceIndex需要			引用引用范围的结束索引 网格三角形列表。

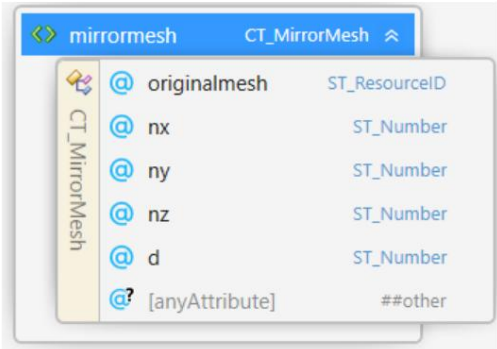
三角形集合中的 <refrange> 元素指的是包含在  
三角形节点。 <refrange> 元素将索引为三角形节点的所有三角形插入到父三角形集合中  
介于起始索引和结束索引之间。

4.1.6. 网格镜像变换

生产者不得使用具有负决定因素的变换来解决镜像问题,因为这会颠覆正常的  
三角形的方向。为了以镜像形式存储网格,生产者必须存储  
原始网格的镜像形式,这意味着具有绝对变换的顶点坐标以及校正的  
正三角取向。

但是,如果生产者想要从转换后的副本中引用原始网格,则生产者可以使用镜像  
命名空间 (<http://schemas.microsoft.com/3dmanufacturing/mirroring/2021/07>)存储对原始网格 ID 的引用  
以及所使用的镜像平面:

元素<mm:mirrormesh>



属性				
姓名	类型	使用	默认	注解
originalmesh	ST_ResourceID必需			原始网格对象的资源ID
nx	ST_ 编号	必需的		镜面法向方程的 X 坐标
这	ST_ 编号	必需的		镜面法向方程的 Y 坐标
新西兰	ST_ 编号	必需的		镜面法向方程的 Z 坐标
d	ST_ 编号	必需的		镜面法方程的距离值

<mirror>元素是可选的。originalmesh属性必须引用当前 3MF 中的前一个网格对象  
模型。originalmesh-attribute 不能引用包含 <mirrormesh> 元素的网格。

镜像变换应通过给定的镜像平面来定义,该镜像平面由方程定义

$$n_x x + n_y y + n_z z + d = 0$$

在局部网格坐标系中。如果给出了镜像网格变换,则必须对顶点和面应用以下规则:

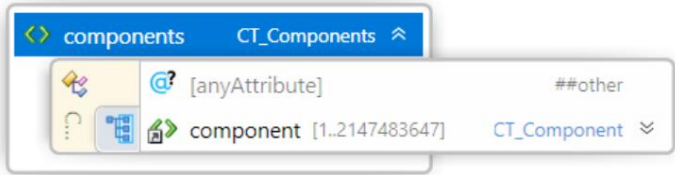
- 网格的顶点数必须等于原始网格的顶点数。
- 对于网格中的每个顶点,顶点列表中必须有一个与原始网格具有相同索引的对应顶点。 相应的顶点坐标必须通过在本地坐标系中的镜像平面上镜像来定义。
- 网格的三角形数量必须等于原始网格的数量。
- 对于网格中的每个三角形,必须有一个与转换后的原始网格相对应的三角形,其在三角形列表中的索引相同。对应的三角形必须具有相同的属性,但顶点 v1 和 v3 必须交换 (以及 p1 和 p3)。
- 网格的三角形集必须与原始网格的三角形集相同。

如果任何给定的要求无效,则实际网格坐标必须优先,并且生产者必须忽略镜像变换。

如果镜像命名空间是必需的扩展,则生产者可以选择存储重复信息。如果生产者选择这样做,则必须为此网格指定一个空三角形元素、一个空顶点元素和一个空三角形集元素。在这种情况下,任何消费者都必须通过给定的规则从原始网格隐式重建网格数据。为了向后兼容,生产者不应将镜像命名空间标记为必需的扩展。

## 4.2. 组件

元素<components>



<components> 元素充当所有要组合到当前对象中的组件的容器。组件是在另一个对象定义的上下文中使用的对象资源。通过使用组件,制作者可以减小 3MF 文档的整体大小。例如,如果 3MF 文档描述了一个汽车 3D 对象,则可能有一个汽车对象资源使用 1 个汽车车身组件实例和 4 个轮胎对象资源的组件实例,在应用必要的旋转和平移 (通过变换)后将车轮放置在汽车的车身对象上。

3D 制造设备必须尊重组件对象的相对位置;除非文件中另有规定,否则它不得将它们相对于彼此进行转换。

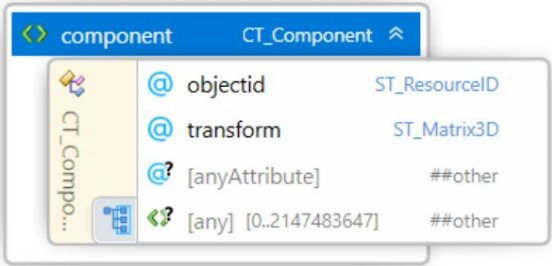
如果组件重叠,3D 制造设备必须使用正填充规则 (如第 4.1.1 节所述)来解决最终几何图形上的歧义。如果任何重叠组件已定义属性,则重叠体积上的结果属性将取自最后一个重叠组件的属性。如果最后一个组件在重叠体积中未定义任何属性,则不得应用属性。

为了避免整数溢出,组件元素必须包含少于 2^31 个组件。

### 4.2.1. 组件

元素<component>





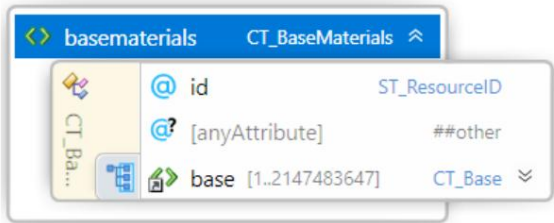
属性

姓名	类型	使用	默认	注解
对象标识	需要ST_ResourceID			引用具有匹配 id 属性的对象资源价值。
转换	ST_Matrix3D			矩阵变换（参见3.3. 3D 矩阵）应用于要输出的项目。
@anyAttribute				

组件选择预定义的对象资源,并在应用提供的  
矩阵变换。由多个原始组件组成的对象定义可以提供非常紧凑的  
文件大小对于一个相当复杂的模型来说。为了与简单解析器的使用保持一致,生产者必须在之前定义对象  
将它们引用为组件。

## 第五章 物质资源

元素<basematerials>



属性

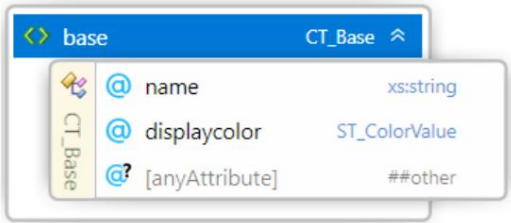
姓名	类型	使用	默认	注解
ID	需要ST_ResourceID			所有材料组中的唯一ID（可能包括来自规范扩展的元素）。
@anyAttribute				

<basematerials> 元素是充当基础材料容器的材料组。这些元素的顺序  
形成一个隐式的基于 0 的索引,供其他元素（例如 <object> 和 <triangle> 元素）引用。其他  
由于资源下有 <any> 元素,因此可以将属性组元素的类型添加为此规范的扩展。  
每个如此定义的属性组必须包含一个 ID,并且该 ID 在模型下的所有资源中必须是唯一的。  
考虑到许多可能的扩展,这些组允许分离和组织不同类型的属性。

为了避免整数溢出,材料组必须包含少于 2^31 种材料。

### 5.1. 基材

元素<base>



属性

姓名	类型	使用	默认	注解
姓名	xs:字符串	必需的		指定材料名称,旨在传达设计意图,以帮助用户进行打印映射材料。
显示颜色	需要ST_ColorValue			指定用于渲染材质的 sRGB 颜色。
@anyAttribute				

基础材料用于定义用于制造模型中某些物体的特定材料。具体而言,支撑物体通常由与模型非牺牲部分不同的材料制成。

由于这些材质既可以应用于物体层面,也可以应用于三角形层面,因此从技术上讲,它们只是指定了物体表面的材质。只要表面具有指定的材质,消费者就可以选择材质在整个体积中的分布方式。

基础材料名称旨在传达设计意图,生产商应避免使用特定于机器的命名,而应使用更易于移植的描述。特定于打印机的信息应存在于打印票证中,其中包括从这些基础材料到实际打印机材料的映射。基础材料名称在整个 3MF 包中应是唯一的。

注意: displaycolor属性仅用于渲染目的,不用于定义对象的实际材质颜色。生产商不得假设displaycolor对打印部件有任何影响。

为了指定印刷部件的颜色设计意图,生产者必须通过材料和属性扩展指定颜色属性或使用其他相应的扩展。

5.1.1. sRGB 颜色

必须使用 6 位或 8 位十六进制数的值来指定 sRGB 颜色,例如 “#RRGGBB”或 “#RRGGBBAA”,其中 RR 是红色通道,其值从 00 到 FF,GG 是绿色通道,其值从 00 到 FF, BB 是蓝色通道,其值从 00 到 FF,AA 是 alpha 通道,其值从 00 (完全透明)到 FF (完全不透明,如果未指定则为默认值)。

注意: 3MF 使用万维网联盟指定的 sRGB (<http://www.w3.org/Graphics/Color/sRGB>)

第 6 章 3MF 文档包功能

3MF 文档利用开放包装约定定义的部分来提供有关包装内容的附加信息。

6.1. 包缩略图和对象缩略图

3MF 文档的制作者可以添加 2D缩略图,以表示 3D 有效载荷的内容或 3D 有效载荷内的特定对象。此图像的内容类型可以是 image/jpeg 或 image/png,并遵循以下小节中描述的内部文件格式要求。

包缩略图具有从包根到缩略图的关系,对象缩略图具有

从 3D 模型部分到缩略图的关系。关系类型必须是：

<http://schemas.openxmlformats.org/package/2006/relationships/metadata/thumbnail>

外部系统（例如操作系统的文件资源管理器、模型浏览器或打印队列预览实用程序）可以访问和显示缩略图。

6.1.1. JPEG 图像

JPEG 图像部分必须包含符合 JPEG 规范的图像。消费者应支持包含 APP0、APP2、APP13 和 APP14 标记的 JPEG 图像。消费者必须支持包含 APP1 标记的 JPEG 图像并正确解释 EXIF 颜色空间。

表 6-1. 支持的 JPEG APPn 标记

应用标记	起源
APP0	JFIF 规范
应用程序1	JEITA 定义的 EXIF 扩展
APP2	ICC 规范定义的 ICC 配置文件标记
APP13	Photoshop 3.0 扩展
APP14	PostScript Level 2 扩展中的 Adobe DCT 滤镜

不得使用 CMYK JPEG 图像。

注意:如果同时指定了 APP2 和 APP13 标记,则 APP2 标记优先。

6.1.2. PNG 图像

PNG 图像部分必须包含符合 PNG 规范的图像。

表 6-2. 对辅助 PNG 块的支持

区块支持级别
tRNS 必须支持
iCCP 必须支持
sRGB 必须忽略
cHRM 必须忽略
gAMA 必须忽略
sBIT 必须忽略

6.2. 核心属性

3MF 文档格式依赖于开放包装约定规范中定义的核心属性部分,以提供有关创建时间、修改时间、作者、搜索关键字和其他以文档为中心的元数据的元数据。有关更多详细信息,请参阅开放包装约定规范。

6.3. 数字签名

3MF 文档可能包含一个或多个数字签名。数字签名提供了一种机制来验证 3MF 文档自签名以来未被篡改。没有数字签名无法证明 3MF 文档从未被签名。

开放封装约定规范提供了如何在 OPC 包中应用数字签名的完整细节。

数字签名来源部分充当 3MF 文档中数字签名有效负载的根。可以通过与数字签名来源部分的关系发现单个数字签名部分。每个数字签名部分都可以具有内联数字证书,也可以具有对包中外部数字签名证书部分的引用。

应用于 3D 模型部件的数字签名应仅包含 3D 模型部件及其引用的任何其他部件以及相关关系。它可能包括包缩略图、数字签名来源、核心属性部件和相关关系部件。

#### 6.3.1. 规范化

在对 3D 模型部件的内容应用或验证数字签名之前,必须首先对其中的 XML 进行规范化,这相当于根据 XML C14N 要求进行 XML 规范化。

### 6.4. 受保护的内容

本规范不包含任何自身的内容保护机制。但是,人们认识到,对于某些工作流程(例如,将 3MF 文档存储为独立文件的工作流程),重要的是防止拦截方在 3D 对象从生产点或分发点传输到授权设备上或授权应用程序或用户的预期消费点时访问 3D 对象详细信息。

开放包装约定规范提供了有关如何保护 OPC 包中内容的完整详细信息。

通过逆转上述步骤获得授权取消内容保护的消费者不得在未经保护机构(可能是也可能不是制作者)批准(书面或程序)的情况下重新保存内容或允许用户以不受保护的方式保存内容(无论文件格式如何)。

## 第二部分 附录

### 附录 A 词汇表

3D 矩阵。4x4 仿射矩阵,用于在三维空间中旋转、缩放和平移对象。

3D 模型。定义输出模型的标记。

3D 模型部件。包含 3D 模型的 OPC 部件。

3D 纹理部分。用于将复杂信息应用于 3D 模型部分中的 3D 对象的文件(本规范中未定义,但可用于本规范的扩展)。

3MF。本规范描述的 3D 制造格式,定义一个或多个用于输出为物理形式的 3D 对象。

3MF 文档。包含符合 3MF 规范的 3D 有效载荷的 OPC 包的数字表现形式。

3MF 文档 StartPart 关系。从包根到 3D 模型部分的 OPC 关系。

组件。包含两个或多个独立定义的对象模型,这些对象在 3D 制造过程期间或之后连接或互锁。组件可能能够逆转,或者各个零件可能不可分离地互锁。

返回。一旦转换到输出坐标空间,打印区域的最大可打印 XZ 平面或模型边界框的对应最大平面。

底部。一旦转换到输出坐标空间,打印区域的最小可打印 XY 平面或模型边界框的对应最小平面。

组件。以完整形状添加到另一个对象的整体定义中的对象。

消费者。读取 3MF 文档的软件、服务或设备。

核心属性。定义明确的包集合 (OPC) 属性,用于定义有关整个包的元数据,例如作者、上次修改时间等。

数字签名。一种机制,如果存在,可用于验证自应用数字签名以来部分或有效载荷未被篡改或更改。

编辑器。一种软件、服务或设备,用于读取和写出 3MF 文档,并可能在中间更改内容。

填充规则。用于确定任何特定点是否被视为位于几何图形“内部”的算法。

正面。一旦转换到输出坐标空间,打印区域的最小可打印 XZ 平面或模型边界框的对应最小平面。

左。一旦转换到输出坐标空间,打印区域的最小可打印 YZ 平面或模型边界框的对应最小平面。

材质。可用于输出对象的物理物质的描述。

材料资源。物体可能引用的潜在资源,用于描述物体的材质。

网格。物体表面的三角形镶嵌。

元数据。关于模型某些部分的辅助信息,为知识丰富的生产者或消费者提供更精细的处理。

模型。作为单个作业的一部分要制造的一组对象。这可能包括单个对象、同一对象的多个实例、多个不同的对象或组件中的多个对象。

必须保留。生产者在重写或保存 MustPreserve 关系类型指定的此 3MF 文件的更改时,应保留一组 OPC 部分。

对象资源。可由 3D 制造设备输出或用作其他对象资源组件的单个 3D 形状。

有效负载。包内相互依赖的部分和关系的完整集合。

PrintTicket。PrintTicket部分的内容。符合打印模式规范。建议尽可能使用 3D 制造的标准打印模式关键字。

PrintTicket 部分。PrintTicket部分提供打印 3MF 文档时使用的设置。

主要 3D 有效负载根。由 3MF 文档 StartPart 关系发现的 3D 有效负载。

制作者。编写 3MF 文档的软件、服务或设备。

保护机构。保护类型所描述的保护机制的所有者。这可能是最初保护文件的制作人、定义将接受哪种保护机制的消费者或第三方保护服务,例如数字版权管理服务。

保护命名空间。保护类型和相关元数据所取自的 XML 命名空间。它是在 <model> 元素上声明。

保护类型。已应用于 3MF 文档的资源和纹理的保护机制的限定名称。这可以像 XOR 混淆一样简单,也可以像完整的数字版权管理解决方案一样复杂。有效的保护类型未在本规范中定义,并且依赖于实现。

资源。扩展中定义的对象、材料或其他内容,可供其他资源使用,或可能是根据构建说明构建物理 3D 对象所必需的。

右图。打印区域的最大可打印 YZ 平面或模型边界框的对应最大平面,

一旦转换到输出坐标空间。

支撑。许多 3D 打印机需要为模型的悬垂区域提供支架,以防止其倒塌或弯曲;这种支架称为支撑。

缩略图。帮助最终用户识别包裹内容的图像,以 JPEG 或 PNG 图像表示。

缩略图部分。包含包缩略图或对象缩略图的 OPC 部分。

顶部。一旦转换到输出坐标空间,打印区域的最大可打印 XY 平面或模型边界框的对应最大平面。

XML 命名空间。根据 XML 命名空间规范,在 <model> 元素上声明的命名空间。

## 附录 B.1. 3MF XSD 架构

### B.1.1 核心规范模式

```
<?xml version= 1.0 encoding= UTF-8 ?>
<xs:schema xmlns= http://schemas.microsoft.com/3dmanufacturing/core/2015/02
  xmlns:xs= http://www.w3.org/2001/XMLSchema
  xmlns:xm= http://www.w3.org/XML/1998/namespace
  targetNamespace= http://schemas.microsoft.com/3dmanufacturing/core/2015/02
  elementFormDefault= unqualified attributeFormDefault= unqualified blockDefault= #all ><!-- 导入 xml:命名空间 -->
  <xs:import namespace= http://www.w3.org/
    XML/1998/namespace schemaLocation= http://www.w3.org/2001/xml.xsd />

  <xs:注释>

    <xs:documentation><![CDATA[架构说明:

      此模式中的项目遵循一个简单的命名约定,即附加一个前缀来指示引用元素的类型:

      无前缀:元素名称
      CT_:复杂类型
      ST_:简单类型

    ]]></xs:documentation>
  </xs:注释>

  <!-- 复杂类型 --> <xs:complexType
    name= CT_Model >
    <xs:序列>
      <xs:element ref= 元数据 minOccurs= 0 maxOccurs= 2147483647 />
      <xs:element ref= 资源 />
      <xs:element ref= 构建 />
      <xs:any namespace= ##other processContents= lax minOccurs= 0 maxOccurs= 2147483647 />
    </xs:sequence>
    <xs:attribute名称= unit 类型= ST_Unit 默认值= 毫米 /> <xs:attribute ref= xml:lang /> <xs:attribute
      名称= requiredextensions 类型= xs:string />
    <xs:attribute名称= recommendedextensions 类型= xs:string /> <xs:anyAttribute命名空间
      = ##other processContents= lax /> </xs:complexType>
```

```

<xs:complexType名称= CT_Resources >
  <xs:序列>
    <xs:choice minOccurs= 0 maxOccurs= 2147483647 >
      <xs:element ref= basematerials minOccurs= 0 maxOccurs= 2147483647 />
      <xs:any namespace= ##other processContents= lax minOccurs= 0 maxOccurs= 2147483647 />
    </xs:选择>
    <xs:element ref= 对象 minOccurs= 0 maxOccurs= 2147483647 />
  </xs:序列>
  <xs:anyAttribute namespace= ##other processContents= lax /></xs:complexType> <xs:complexType
name= CT_Build >

  <xs:序列>
    <xs:element ref= 项目 minOccurs= 0 maxOccurs= 2147483647 />
  </xs:sequence>
  <xs:anyAttribute namespace= ##other processContents= lax /></xs:complexType> <xs:complexType
name= CT_BaseMaterials >

  <xs:序列>
    <xs:element ref= base maxOccurs= 2147483647 />
  </xs:sequence>
  <xs:attribute名称= id 类型= ST_ResourceID 使用= required /> <xs:anyAttribute命名空间= ##other
processContents= lax /></xs:complexType> <xs:complexType名称= CT_Base > <xs:attribute名称
= name 类型= xs:string 使
用= required /> <xs:attribute名称= displaycolor 类
型= ST_ColorValue 使用= required /> <xs:anyAttribute命名空间= ##other processContents= lax /
></xs:complexType> <xs:complexType名称= CT_MetadataGroup >

  <xs:序列>
    <xs:element ref= 元数据 maxOccurs= 2147483647 />
  </xs:sequence>
  <xs:anyAttribute namespace= ##other processContents= lax />
</xs:complexType>
<xs:complexType名称= CT_Object >
  <xs:序列>
    <xs:element ref= 元数据组 minOccurs= 0 maxOccurs= 1 /> <xs:choice>

      <xs:element ref= 网格 />

      <xs:element ref= 组件 /> </xs:choice>

    <xs:any namespace= ##other processContents= lax minOccurs= 0 maxOccurs= 2147483647 />
  </xs:序列>
  <xs:attribute名称= id 类型= ST_ResourceID 使用= 必需 /> <xs:attribute名称= 类型 类型
= ST_ObjectType 默认= 模型 /> <xs:attribute名称= 缩略图 类型= ST_UriReference /> <xs:attribute名称
= 零件号 类型= xs:string /> <xs:attribute名称= 名称 类型= xs:string /> <xs:attribute名称
= pid 类型= ST_ResourceIndex /> <xs:attribute名称= pindex 类型
= ST_ResourceIndex /> <xs:anyAttribute命名空间= ##other
processContents= lax /></xs:complexType> <xs:complexType名称= CT_Mesh >

```

```

<xs:序列>
  <xs:element ref= 顶点 />

  <xs:element ref= 三角形 /> <xs:any
    namespace= ##other processContents= lax minOccurs= 0 maxOccurs= 2147483647 />
</xs:sequence>

<xs:anyAttribute namespace= ##other processContents= lax /> </xs:complexType> <xs:complexType
name= CT_Vertices >

<xs:序列>
  <xs:element ref= vertex minOccurs= 3 maxOccurs= 2147483647 />

</xs:sequence>

<xs:anyAttribute namespace= ##other processContents= lax />
</xs:complexType>
<xs:complexType名称= CT_Vertex > <xs:attribute名称
= x 类型= ST_Number 用途= required /> <xs:attribute名称= y 类型= ST_Number 用
途= required /> <xs:attribute名称= z 类型= ST_Number 用途= required />
<xs:anyAttribute命名空间= ##other processContents= lax />

</xs:complexType>
<xs:complexType名称= CT_Triangles >

<xs:序列>
  <xs:element ref= 三角形 minOccurs= 1 maxOccurs= 2147483647 />
</xs:序列>

<xs:anyAttribute命名空间= ##other processContents= lax /> </xs:complexType> <xs:complexType名
称= CT_Triangle >

<xs:attribute名称= v1 类型= ST_ResourceIndex 使用
= required /> <xs:attribute名称= v2 类型= ST_ResourceIndex 使用= required /> <xs:attribute名称
= v3 类型= ST_ResourceIndex 使用= required /> <xs:attribute名称= p1 类型= ST_ResourceIndex /
> <xs:attribute名称= p2 类型= ST_ResourceIndex /> <xs:attribute名称= p3 类型= ST_ResourceIndex /
> <xs:attribute名称= pid 类型= ST_ResourceID /> <xs:anyAttribute命名空间
= ##other processContents= lax /> </xs:complexType> <xs:complexType名称
= CT_Components >

<xs:序列>
  <xs:element ref= 组件 maxOccurs= 2147483647 />
</xs:sequence>

<xs:anyAttribute namespace= ##other processContents= lax /> </xs:complexType> <xs:complexType
name= CT_Component >

<xs:序列>
  <xs:any namespace= ##other processContents= lax minOccurs= 0 maxOccurs= 2147483647 />
</xs:sequence>

<xs:attribute名称= objectid 类型= ST_ResourceID 使用= required /> <xs:attribute名称= transform 类型
= ST_Matrix3D /> <xs:anyAttribute命名空间= ##other processContents= lax />

</xs:complexType>
<xs:complexType名称= CT_Metadata 混合= true > <xs:attribute名称= 名称 类
型= xs:QName 使用= 必需 /> <xs:attribute名称= 保留 类型= xs:boolean 使用= 可选 />
<xs:attribute名称= 类型 类型= xs:string 使用= 可选 />

```



```
<xs:anyAttribute namespace= ##other processContents= lax /></
xs:complexType>
<xs:complexType name= CT_Item >
  <xs:序列>
    <xs:element ref= metadatagroup minOccurs= 0 maxOccurs= 1 /><xs:any
      namespace= ##other processContents= lax minOccurs= 0 maxOccurs= 2147483647 />
  </xs:序列>
  <xs:attribute name= objectid type= ST_ResourceID use= required /><xs:attribute name= transform
    type= ST_Matrix3D /><xs:attribute name= partnumber type= xs:string />
  <xs:anyAttribute namespace= ##other processContents= lax /></xs:complexType>
  <!-- 简单类型 --><xs:simpleType name= ST_Unit ><xs:restriction base= xs:string ><xs:enumeration
    value= micron />
    <xs:enumeration值= 毫米 />
    <xs:enumeration值= 厘米 />
    <xs:enumeration值= 英寸 />
    <xs:enumeration值= 脚 />
    <xs:enumeration值= 米 />
  </xs:限制>
</xs:simpleType>
<xs:simpleType name= ST_ColorValue ><xs:restriction
  base= xs:string ><xs:pattern value= #[0-9|AF|af][0-9| AF|af]
  [0-9|AF|af][0-9|AF|af][0-9|AF|af]
[0-9|AF|af]((0-9|AF|af)[0-9|AF|af])? />
</xs:限制>
</xs:simpleType>
<xs:simpleType名称= ST_UriReference ><xs:restriction
  base= xs:anyURI ><xs:pattern值= /.*/>
  </xs:限制>
</xs:simpleType>
<xs:simpleType名称= ST_Matrix3D ><xs:restriction
  base= xs:string ><xs:whiteSpace值= collapse /><xs:pattern
  值= ((\-\-)?(((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]+?) ((\-\-)?(((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]+?)
((\-\-)?(((0-9)+(\.[0-9]+)?)(\.[0-9]+))((e|E)\-\-)?[0-9]+?) ((\-\-)?(((0-9)+(\.[0-9]+)?)(\.[0-9]+))((e|E)\-\-)?[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名称= ST_Number >
  <xs:restriction base= xs:double >
    <xs:whiteSpace值= collapse /><xs:pattern值= ((\-\-)?
(((0-9)+(\.[0-9]+)?)(\.[0-9]+))((e|E)\-\-)?[0-9]+?) />
```

```
</xs:限制>
</xs:simpleType>
<xs:simpleType名称= ST_ResourceID >
  <xs:restriction base= xs:positiveInteger > <xs:maxExclusive值
    = 2147483648 />
</xs:限制>
</xs:simpleType>
<xs:simpleType名称= ST_ResourceIndex >
  <xs:restriction base= xs:nonNegativeInteger >
    <xs:maxExclusive值= 2147483648 />
</xs:限制>
</xs:simpleType>
<xs:simpleType名称= ST_ObjectType ><xs:restriction基础
  = xs:string > <xs:enumeration值= 模型 />

  <xs:enumeration值= 固体支撑 /> <xs:enumeration值= 支撑 />
  <xs:enumeration值= 表面 />

  <xs:enumeration值= 其他 />
</xs:限制>
</xs:simpleType><!-- 元素 -->

<xs:元素名称= metadatagroup 类型= CT_MetadataGroup /> <xs:元素名称= model 类型= CT_Model /
> <xs:元素名称= resources 类型= CT_Resources /> <xs:元素名称= build 类
型= CT_Build /> <xs:元素名称= basematerials 类型= CT_BaseMaterials /> <xs:元素名称
= base 类型= CT_Base /> <xs:元素名称= object 类型= CT_Object />
<xs:元素名称= mesh 类型= CT_Mesh /> <xs:元素名称= vertices 类型= CT_Vertices /> <xs:元素名称
= vertex 类型= CT_Vertex /> <xs:元素名称= triangles 类型
= CT_Triangles /> <xs:元素名称= triangle 类型= CT_Triangle /> <xs:元素名
称= components 类型= CT_Components /> <xs:element名称
= component 类型= CT_Component /> <xs:element名称= metadata 类型
= CT_Metadata /> <xs:element名称= item 类型= CT_Item />

</xs:架构>
```

B.1.2 三角集扩展模式

```
<?xml version= 1.0 encoding= UTF-8 ?> <xs:schema
xmlns= http://schemas.microsoft.com/3dmanufacturing/trianglesets/2021/07
xmlns:xs= http://www.w3.org/2001/XMLSchema
xmlns:xml= http://www.w3.org/XML/1998/namespace
targetNamespace= http://schemas.microsoft.com/3dmanufacturing/trianglesets/2021/07 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:注释>
<xs:documentation>
  <![CDATA[ 架构说
  明:
```

此架构中的项目遵循添加前缀的简单命名约定

指示参考元素的类型：

- 无前缀:元素名称
- CT\_:复杂类型
- ST\_:简单类型

```
]]>
</xs:文档>
</xs:注释>
<!-- 复杂类型 --> <xs:complexType
name=  CT_Mesh  >
  <xs:序列>
    <xs:element ref=  trianglesets  minOccurs=  0  maxOccurs=  1  /> <xs:any
      namespace=  ##other  processContents=  lax  minOccurs=  0
最大发生次数=  2147483647  />
    </xs:序列>
    <xs:anyAttribute namespace=  ##other  processContents=  lax  /> </
xs:complexType>
<xs:complexType name=  CT_TriangleSets  >
  <xs:序列>
    <xs:element ref=  triangleset  minOccurs=  0  maxOccurs=  2147483647  />
  </xs:sequence>
  <xs:anyAttribute namespace=  ##other  processContents=  lax  /> </xs:complexType>
<xs:complexType
name=  CT_TriangleSet  >
  <xs:序列>
    <xs:element ref=  ref  minOccurs=  0  maxOccurs=  2147483647  />
    <xs:element ref=  refrange  minOccurs=  0  maxOccurs=  2147483647  /> <xs:any
      namespace=  ##other  processContents=  lax  minOccurs=  0
最大发生次数=  2147483647  />
    </xs:序列>
    <xs:attribute名称=  名称  类型=  xs:string  默认值=  无  /> <xs:attribute名称=  标识
符  类型=  xs:QName  /> <xs:anyAttribute命名空间=  ##other
processContents=  lax  /> </xs:complexType> <xs:complexType名称=  CT_Ref  >

  <xs:序列>
    <xs:any namespace=  ##other  processContents=  lax  minOccurs=  0
最大发生次数=  2147483647  />
  </xs:sequence>
  <xs:attribute名称=  index  类型=  ST_ResourceIndex  使用=  required  /> <xs:anyAttribute命名空间=  ##other
processContents=  lax  /> </xs:complexType> <xs:complexType名称=  CT_RefRange  >

  <xs:序列>
    <xs:any namespace=  ##other  processContents=  lax  minOccurs=  0
最大发生次数=  2147483647  />
  </xs:sequence>
  <xs:attribute名称=  startindex  类型=  ST_ResourceIndex  使用=  required  /> <xs:attribute名称=  endindex  类型
=  ST_ResourceIndex  使用=  required  /> <xs:anyAttribute命名空间=  ##other  processContents=  lax  />
```

```

</xs:complexType> <!--
简单类型 --> <xs:simpleType
name= ST_ResourceIndex >
  <xs:restriction base= xs:nonNegativeInteger > <xs:maxExclusive值= 2147483648 /
  >
</xs:限制>
</xs:简单类型>

<!-- 元素 -->
<xs:元素名称= 网格 类型= CT_Mesh /> <xs:元素名称= 三
角形集 类型= CT_TriangleSets /> <xs:元素名称= 三角形集 类型
= CT_TriangleSet /> <xs:元素名称= ref 类型= CT_Ref /> <xs:元素名称
= refrange 类型= CT_RefRange />

</xs:架构>

```

### B.1.3 镜像扩展模式

```

<?xml version= 1.0 encoding= UTF-8 ?>
<xs:schema xmlns= http://schemas.microsoft.com/3dmanufacturing/mirroring/2021/07
  xmlns:xs= http://www.w3.org/2001/XMLSchema
  xmlns:xml= http://www.w3.org/XML/1998/namespace
  targetNamespace= http://schemas.microsoft.com/3dmanufacturing/mirroring/2021/07
  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:注释>
    <xs:documentation>
      <![CDATA[ 架构
      说明:

      此架构中的项目遵循添加前缀的简单命名约定

      指示参考元素的类型:

      无前缀:元素名称
      CT_:复杂类型
      ST_:简单类型

      ]]>
    </xs:文档>
  </xs:注释>

  <!-- 复杂类型 -->
  <xs:complexType名称= CT_Mesh >
    <xs:序列>
      <xs:element ref= 镜像 />
      <xs:any namespace= ##other processContents= lax minOccurs= 0 maxOccurs= 1 />
    </xs:sequence>
    <xs:anyAttribute namespace= ##other processContents= lax /> </xs:complexType>

  <xs:complexType
  name= CT_MirrorMesh >
    <xs:attribute名称= originalmesh 类型= ST_ResourceID /> <xs:attribute名称
    = nx 类型= ST_Number /> <xs:attribute名称= ny 类型
    = ST_Number /> <xs:attribute名称= nz 类型
    = ST_Number />

```

```
<xs:attribute name= d type= ST_Number /> <xs:anyAttribute
namespace= ##other processContents= lax /> </xs:complexType> <!-- 简单类型 --> <xs:simpleType
name= ST_ResourceID >

<xs:restriction base= xs:positiveInteger > <xs:maxExclusive值
= 2147483648 />
</xs:限制>
</xs:simpleType>
<xs:simpleType名称= ST_Number >
<xs:restriction base= xs:double >
<xs:whiteSpace值= collapse /> <xs:pattern值= ((\-\|+)?
([0-9]+\.\.[0-9]+)?)(\.[0-9]+))((e|E)(\-\|+)?[0-
9]+) ? ) ” />
</xs:限制>
</xs:simpleType> <!-- 元素 -->

<xs:element名称= mesh 类型= CT_Mesh /> <xs:element名称
= mirromesh 类型= CT_MirrorMesh />
</xs:架构>
```

附录 B.2. 3MF 元数据示例

```
<?xml version= 1.0 encoding= UTF-8 ?> <模型单位= 毫米
xml:lang= en-us xmlns:m= http://schemas.microsoft.com/
3dmanufacturing/material/2015/02 xmlns= http://schemas.microsoft.com/3dmanufacturing/core/2015/02 xmlns:vendor1= http://
www.vendorwebsite.com/3mf/vendor13mfextension/2017/01 >

<metadata name= 版权 >© Microsoft Corporation 2013</metadata> <metadata name= 应用程序 >Microsoft
3D Builder</metadata> <metadata name= LicenseTerms >保留所有权利</metadata> <metadata
name= Title >立方体</metadata>

<metadata name= 设计师 >微软公司</metadata>
<metadata name= CreationDate >2013-10-07</metadata>
<metadata name= ModificationDate >2014-03-27</metadata>
<metadata name= Description >多维数据集</metadata>
<metadata name= vendor1:CustomMetadata1 retain= 1 >CE8A91FB-C44E-4F00-B634-
BAA411465F6A</metadata>
<资源>
<basematerials id= 1 >
<base name= 绿色 displaycolor= #21BB4CFF />
</basematerials>
<object id= 2 type= 模型 pid= 1 pindex= 0 > <元数据组>

<metadata name= vendor1:CustomMetadata2 retain= true type= xs:string >03DAE6E4-24FF-4B20-
97A1-7487AB9C1CB0</元数据>
</元数据组> <网格>

<顶点>
<顶点x= 0 y= 42.998 z= 39.998 /> <顶点x= 39.998
y= 42.998 z= 39.998 /> <顶点x= 0 y= 82.998 z= 39.998 /> <顶点
x= 39.998 y= 82.998 z= 0 />
```

```
<顶点x= 0 y= 42.998 z= 0 /><顶点x= 0
y= 82.998 z= 0 /><顶点x= 39.998 y= 42.998
z= 0 /><顶点x= 39.998 y= 82.998 z= 39.998 />

</顶点>

<三角形><三角形
v1= 0 v2= 1 v3= 2 /><三角形v1= 3 v2= 4
v3= 5 /><三角形v1= 4 v2= 3 v3= 6 /><三角形
v1= 7 v2= 2 v3= 1 /><三角形v1= 4 v2= 6
v3= 1 /><三角形v1= “4” v2= 2 v3= 5 /><三角形
v1= 7 v2= 1 v3= 6 /><三角形v1= 5 v2= 2
v3= 7 /><三角形v1= 4 v2= 0 v3= 2 /><三角形
v1= 6 v2= 3 v3= 7 /><三角形v1= 1 v2= 0
v3= 4 /><三角形v1= “7” v2= “3” v3= “5” />

</三角形></网格>

</对象><对象ID
= “3”类型= “模型”><组件>

<component objectid= 2 />
</组件></对象>

</资源>
<构建>

<item objectid= 3 变换= 1 0 0 1 0 0 0 1 -19.999 -62.998 0 >
<元数据组><元数据名称
= vendor1:CustomMetadata3 类型= xs:boolean >1</元数据>
</metadatagroup></item>

</build>
</模型>
```

附录 C. 标准命名空间和内容类型

C.1 内容类型

3D 模型应用程序/vnd.ms-package.3dmanufacturing-3dmodel+xml

PrintTicket 应用程序/vnd.ms-printing.printticket+xml

C.2 关系类型

StartPart <http://schemas.microsoft.com/3dmanufacturing/2013/01/3dmodel>

缩略图<http://schemas.openxmlformats.org/package/2006/relationships/metadata/thumbnail>

PrintTicket <http://schemas.microsoft.com/3dmanufacturing/2013/01/printticket>

必须保留<http://schemas.openxmlformats.org/package/2006/relationships/mustpreserve>

C.3 命名空间

3D 模型<http://schemas.microsoft.com/3dmanufacturing/core/2015/02>

三角形集<http://schemas.microsoft.com/3dmanufacturing/trianglesets/2021/07>

镜像<http://schemas.microsoft.com/3dmanufacturing/mirroring/2021/07>

## 参考

通用 URI 语法的 BNF

“通用 URI 语法的 BNF。”万维网联盟。[http://www.w3.org/Addressing/URL/5\\_URI\\_BNF.html](http://www.w3.org/Addressing/URL/5_URI_BNF.html)

JPEG

Hamilton, Eric. “JPEG 文件交换格式,版本 1.02。”万维网联盟。1992 年。<http://www.w3.org/Graphics/JPEG/jfif3.pdf>

开放包装惯例

Ecma International. “Office Open XML 第 2 部分:开放打包约定。”2006 年。<http://www.ecma-international.org>

已弃用前几内容

Duce,David (编辑)。“便携式网络图形 (PNG) 规范”,第二版。万维网联盟。2003 年。<http://www.w3.org/TR/2003/REC-PNG-20031110>

打印架构

微软公司。“打印模式。”2006 年。<http://windowssdk.msdn.microsoft.com/en-us/library/default.aspx>

RFC 2119

Bradner,S. “用于在 RFC 中指示要求级别的关键词。”互联网协会。1997 年。<http://www.rfc-editor.org>

RFC 3066

Alvestrand, H. “语言识别标签。”互联网协会。2001 年。<http://www.rfc-editor.org>

sRGB

Anderson, Matthew,Srinivasan Chandrasekar,Ricardo Motta 和 Michael Stokes. “互联网的标准默认色彩空间 - sRGB,版本 1.10。”万维网联盟。1996 年。<http://www.w3.org/Graphics/Color/sRGB>

统一码

Unicode 联盟。Unicode 标准,版本 4.0.0,定义者: Unicode 标准,版本 4.0。马萨诸塞州波士顿:Addison-Wesley,2003 年。

XML

Bray, Tim,Eve Maler,Jean Paoli,CM Sperlberg-McQueen 和 François Yergeau (编辑)。“可扩展标记语言 (XML) 1.0 (第四版)” 。万维网联盟。2006 年。<http://www.w3.org/TR/2006/REC-xml-20060816/>

XML C14N

Boyer, John. “规范 XML 版本 1.0。”万维网联盟。2001 年。<http://www.w3.org/TR/xml-c14n>。

XML 命名空间

Bray, Tim,Dave Hollander,Andrew Layman 和 Richard Tobin (编辑)。“XML 1.0 中的命名空间 (第二版)” 。万维网联盟。2006 年。<http://www.w3.org/TR/2006/REC-xml-names-20060816/>

## XML 模式

Beech, David, Murray Maloney, Noah Mendelsohn 和 Henry S. Thompson (编辑)。“XML Schema 第 1 部分:结构”, 第二版。万维网联盟。2004 年。<http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>

Biron, Paul V. 和 Ashok Malhotra (编辑)。“XML Schema 第 2 部分:数据类型”, 第二版。万维网联盟。2004 年。<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>

版权所有 3MF Consortium 2018。