
3MF Boolean Operations Extension

Specification & Reference Guide

Version **0.5.1**

Status Draft

Table of Contents

- [Preface](#)
 - [About this Specification](#)
 - [Document Conventions](#)
 - [Language Notes](#)
 - [Software Conformance](#)
- [Part I: 3MF Documents](#)
 - [Chapter 1. Overview of Additions](#)
 - [Chapter 2. Components](#)
- [Part II. Appendixes](#)
 - [Appendix A. Glossary](#)
 - [Appendix B. 3MF XSD Schema](#)
 - [Appendix C. Standard Namespace](#)
 - [Appendix D: Example file](#)
- [References](#)

Preface

About this Specification

This 3MF Boolean Operations Extension 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.x.

Document Conventions

See [the 3MF Core Specification conventions](#).

In this extension specification, as an example, the prefix "bo" maps to the xml-namespace "http://schemas.microsoft.com/3dmanufacturing/booleanoperations/2021/02". See [Appendix C. Standard Namespace](#).

Language Notes

See [the 3MF Core Specification language notes](#).

Software Conformance

See [the 3MF Core Specification software conformance](#).

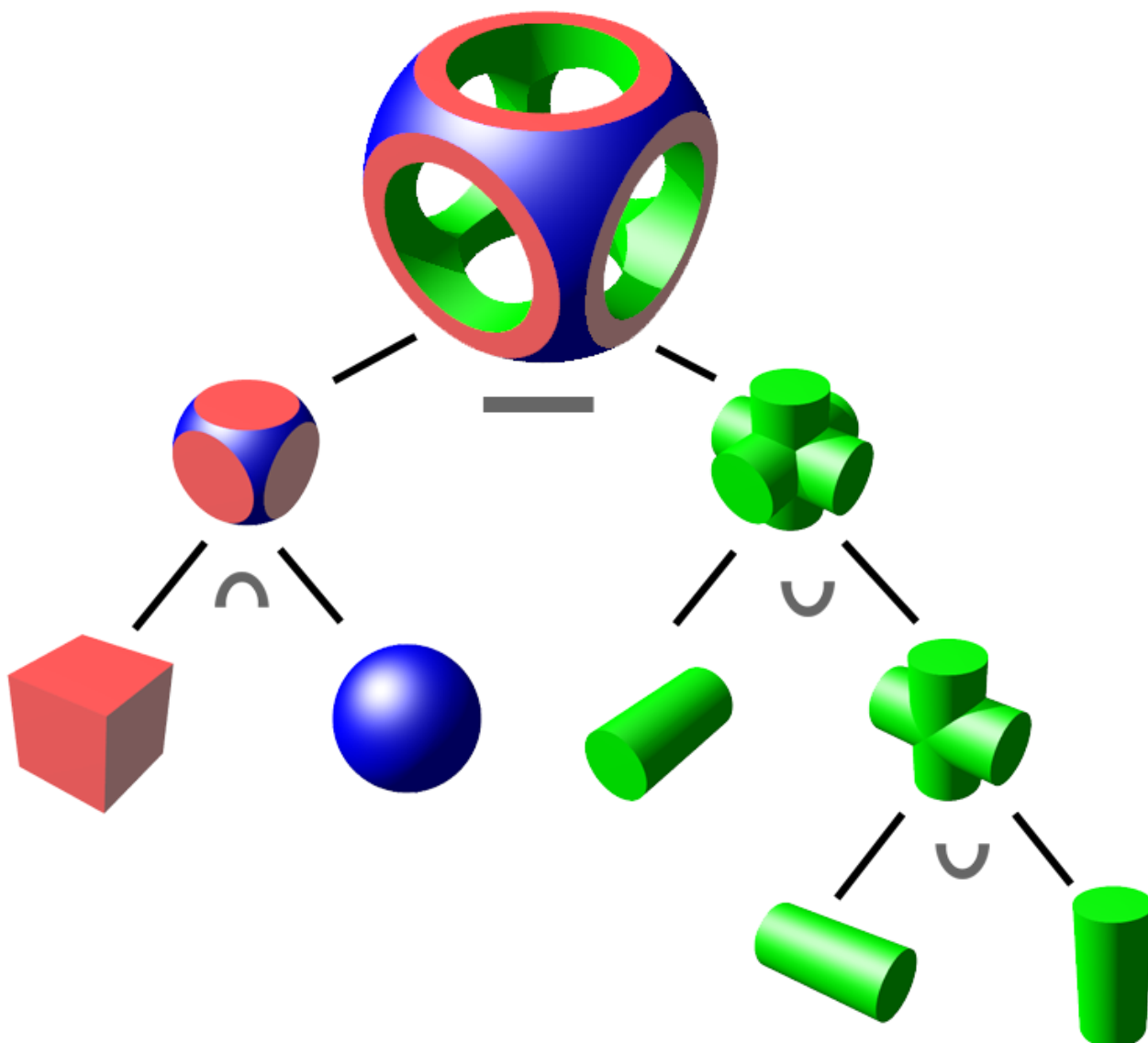
Part I: 3MF Documents

Chapter 1. Overview of Additions



The 3MF Core Specification defines the `<components>` element in the `<object>` resource as definition of a logical association of different objects to form an assembly, with the intent to allow reuse of model definitions for an efficient encoding. However, it allows that each individual object component to be tracked as a single part in the assembly. The resultant shape of a `<components>` element is the aggregation (union) of each `<component>` object element.

This extension is based in the Constructive Solid Geometry ([CSG](#)) which defines a binary tree of operations, and it extends this concept to non-binary tree.

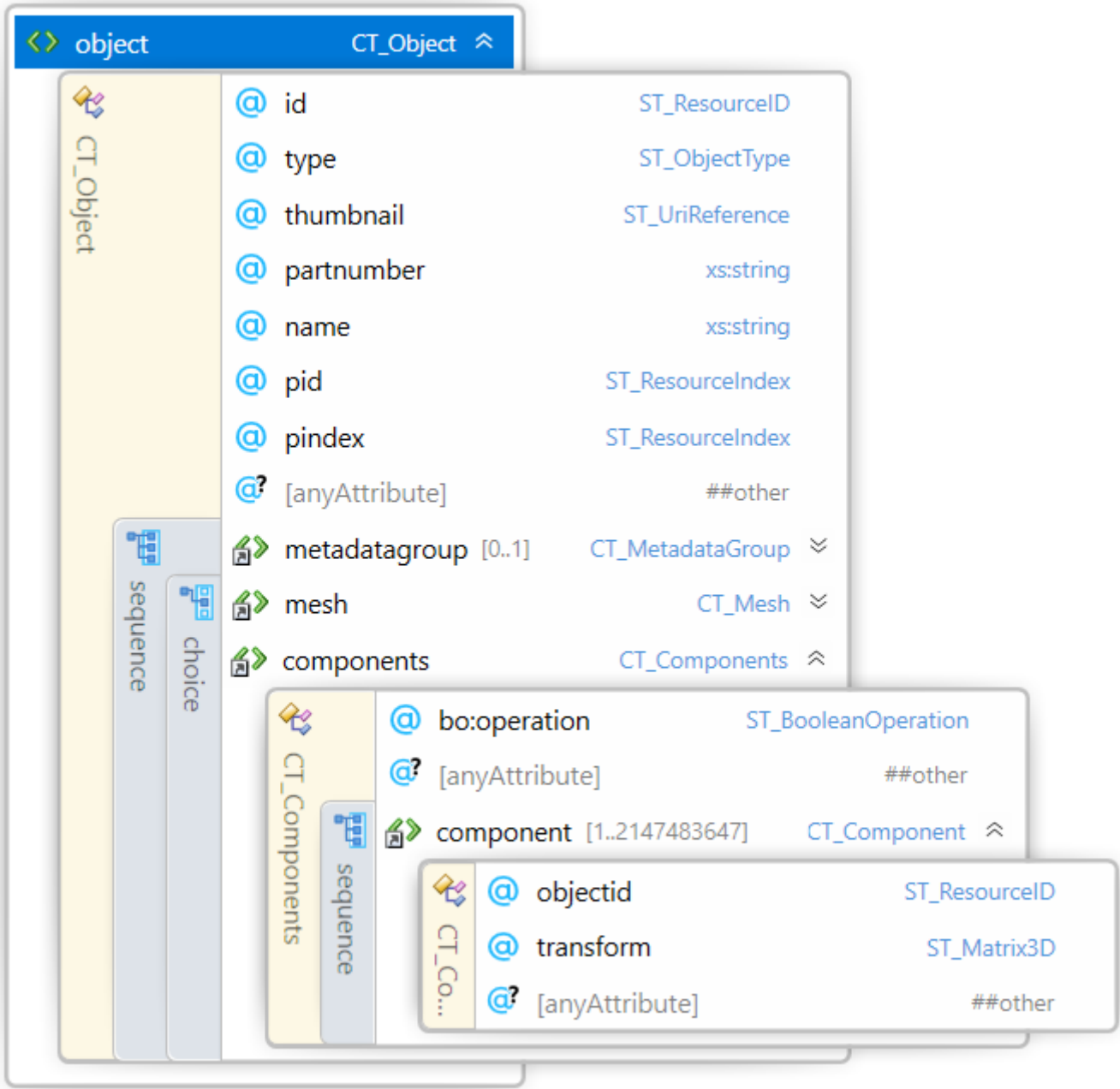


This document extends that definition by defining whether the components are a logical association (assembly) or define a boolean operation to merge the objects as a physical association (single part).

This document describes a new attribute in the <components> elements to describe the association of objects. This attribute is OPTIONAL for producers but MUST be supported by consumers that specify support for the 3MF Boolean Operations Extension.

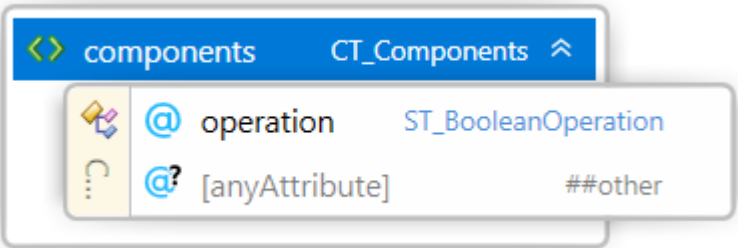
To avoid data loss while parsing, a 3MF package which uses referenced objects MUST enlist the 3MF Boolean Operations Extension as "required extension", as defined in the core specification. However if the 3MF Boolean Operations Extension is not enlisted a required, any consumer non-supporting the 3MF Boolean Operations Extension may be able to process the rest of the document.

Figure 1-1: Overview of 3MF Boolean Operations Extension XML structure



Chapter 2. Components

Element `<components>`



The `<components>` element in the [the 3MF Core Specification](#) is enhanced by new attributes to define the compenents association.

Name	Type	Use	Default	Annotation
------	------	-----	---------	------------

Name	Type	Use	Default	Annotation
operation	ST_BooleanOperation		none	Boolean operation: union, difference or intersection operations, or none when no operation is specified

@anyAttribute

The <components> element is enhanced the the following attributes:

operation - The boolean operation to perform, of "none" for no operacion.

It starts by performing the boolean operation to the first components with the second component, if available. If more components are available, it iteratively performs the boolean operation from previous result and next component.

The options for the operations are the following:

1. *none*. No operation is specified. The <components> element specifies an assembly of objects.
2. *union*. The new object shape is defined as the merger of the shapes. The new object surface property is defined by the property of the surface property defining the outer surface. If material and the volumetric property, if available, in the overlapped volume is defined by the added object.

$$\text{union}(a,b,c,d) = ((a \cup b) \cup c) \cup d$$

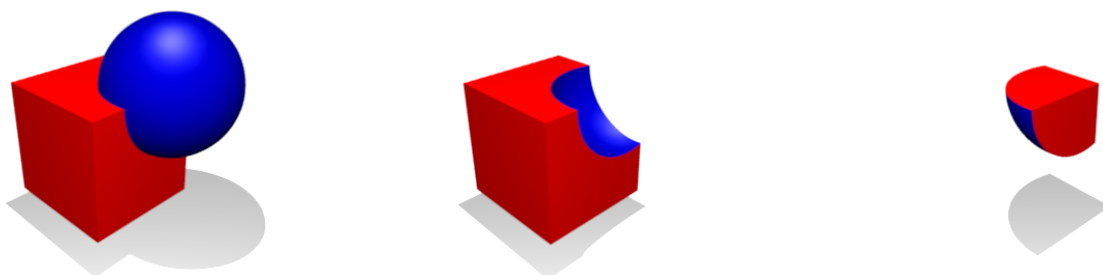
3. *difference*. The new object shape is defined by the shape in the first object shape that is not in any other object shape. The new object surface property, where overlaps, is defined by the object surface property of the substracting object(s).

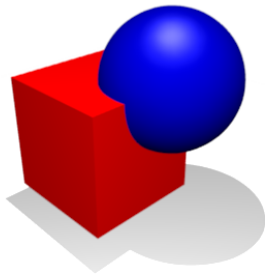
$$\text{difference}(a,b,c,d) = ((a - b) - c) - d = a - \text{union}(b,c,d)$$

4. *intersection*. The new object shape is defined as the common (clipping) shape in all objects. The new object surface property is defined as the object surface property of the object clipping that surface.

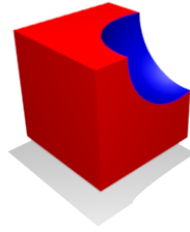
$$\text{intersection}(a,b,c,d) = ((a \cap b) \cap c) \cap d$$

Note: Since a component may refers to another object component as a tree of object components, once an object components is defined with a boolean operation (ie. merging models), all components in the subtree below **MUST** also be merged with a boolean operation. If not specified, the default boolean operation "none" **MUST** be treated as a "union".





union: merger of objects into one



difference: subtraction of object from another one



intersection: portion common to objects

The boolean operations follow the fill rule conversion defined by See [the 3MF Core Specification fill rule](#).

Part II. Appendixes

Appendix A. Glossary

See [the 3MF Core Specification glossary](#).

Appendix B. 3MF XSD Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://schemas.microsoft.com/3dmanufacturing/booleanoperations/2021/02"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"

  targetNamespace="http://schemas.microsoft.com/3dmanufacturing/booleanoperations/20
21/02"
  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_Components">
    <xs:attribute name="operation" type="ST_BooleanOperation" default="none"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
```

```

<xs:simpleType name="ST_BooleanOperation">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none"/>
    <xs:enumeration value="union"/>
    <xs:enumeration value="difference"/>
    <xs:enumeration value="intersection"/>
  </xs:restriction>
</xs:simpleType>

<!-- Elements -->
<xs:element name="components" type="CT_Components"/>
</xs:schema>

```

Appendix C. Standard Namespace

BooleanOperation <http://schemas.microsoft.com/3dmanufacturing/booleanoperations/2021/02>

Appendix D: Example file

3D model

```

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<model xmlns="http://schemas.microsoft.com/3dmanufacturing/core/2015/02"
  xmlns:bo="http://schemas.microsoft.com/3dmanufacturing/booleanoperations/2021/
02"
  requiredextensions="bo" unit="millimeter" xml:lang="en-US">
  <resources>
    <object id="11" type="model" name="shuttle">
      <mesh>
        <vertices>...</vertices>
        <triangles>...</triangles>
      </mesh>
    </object>
    <object id="21" type="model" name="OV-104">
      <mesh>
        <vertices>...</vertices>
        <triangles>...</triangles>
      </mesh>
    </object>
    <object id="22" type="model" name="Atlantis">
      <mesh>
        <vertices>...</vertices>
        <triangles>...</triangles>
      </mesh>
    </object>
    <object id="101" type="model" name="model with labels">

```

```

        <components bo:operation="union">
            <component objectid="2" transform="1.0000 0.0000 0.0000 0.0000
1.0000 0.0000 0.0000 0.0000 1.0000 34.1020 35.1070 5.1000"/>
            <component objectid="21" transform="1.00000 0.0000 0.0000 0.0000
1.0000 0.0000 0.0000 0.0000 1.000 35.7020 35.7070 5.7000"/>
        </components>
    </object>
    <object id="102" type="model" name="model with engraved label">
        <components bo:operation="difference">
            <component objectid="2" transform="1.0000 0.0000 0.0000 0.0000
1.0000 0.0000 0.0000 0.0000 1.0000 74.1020 35.1070 5.1000"/>
            <component objectid="22" transform="1.00000 0.0000 0.0000 0.0000
1.0000 0.0000 0.0000 0.0000 1.000 75.7020 35.7070 5.7000"/>
        </components>
    </object>
</resources>
<build>
    <item objectid="101"/>
    <item objectid="102"/>
</build>
</model>

```

References

CSG

From Wikipedia, the free encyclopedia. "Constructive solid geometry".

https://en.wikipedia.org/wiki/Constructive_solid_geometry

Cornelia Haslinger, Universität Salzburg. "Constructive Solid Geometry in Education". https://www.uni-salzburg.at/fileadmin/multimedia/Mathematik/images/EMMA/Workshop_Turkey/education_days/CSG_Haslinger_low_quality.pdf.

See the [3MF Core Specification references](#) for additional references.