



# 3MF Boolean Operations Extension

## Specification & Reference Guide

**Version** 1.1.0

**Status** Published

## Disclaimer

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.

## Table of Contents

- [Preface](#)
  - [About this Specification](#)
  - [Document Conventions](#)
  - [Language Notes](#)
  - [Software Conformance](#)
- [Part I: 3MF Documents](#)
  - [Chapter 1. Overview of Additions](#)
  - [Chapter 2. Object Resources](#)
    - [2.1. Boolean Shape](#)
- [Part II. Appendices](#)
  - [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, "Appendices," 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.3mf.io/3dmanufacturing/booleanoperations/2023/07>". 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 tree of different objects to form an assembly, with the intent to allow the reuse of model definitions for an efficient encoding. The resultant shape of a <components> element is the aggregation of each <component> object element.

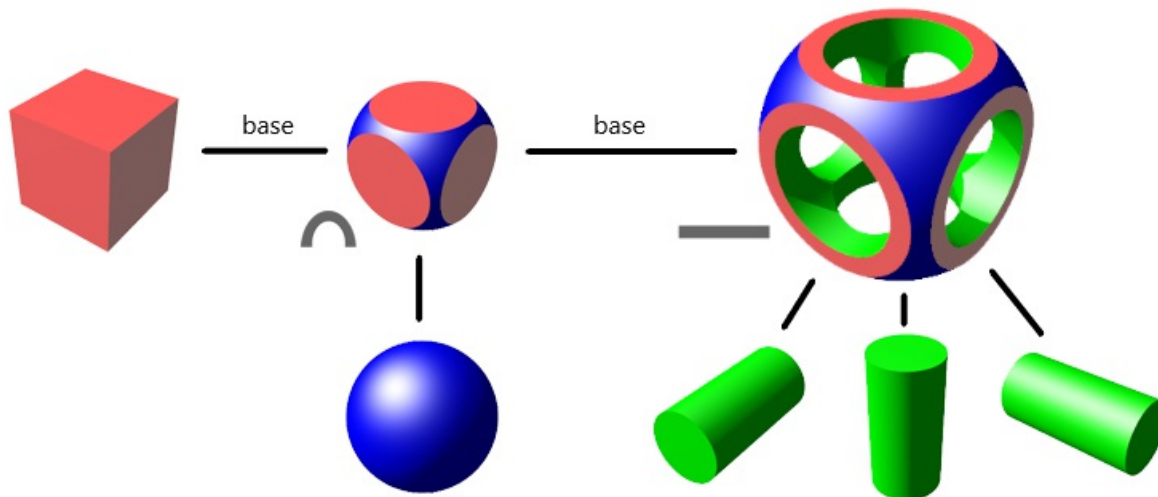
The [section 4.1 Meshes in the core specification](#). defines a <mesh> element as a basic object shape which is defined by triangles.

This extension defines how to combine different objects into a new type of shape defined as

a *booleanshape* object. It is based in Constructive Solid Geometry (CSG).

However, to limit complexity in the consumer, this spec reduces the CSG scope to an ordered concatenation of boolean shapes (left to right in figure 1.1 below).

**Figure 1-1: Concatenating booleans operations.**

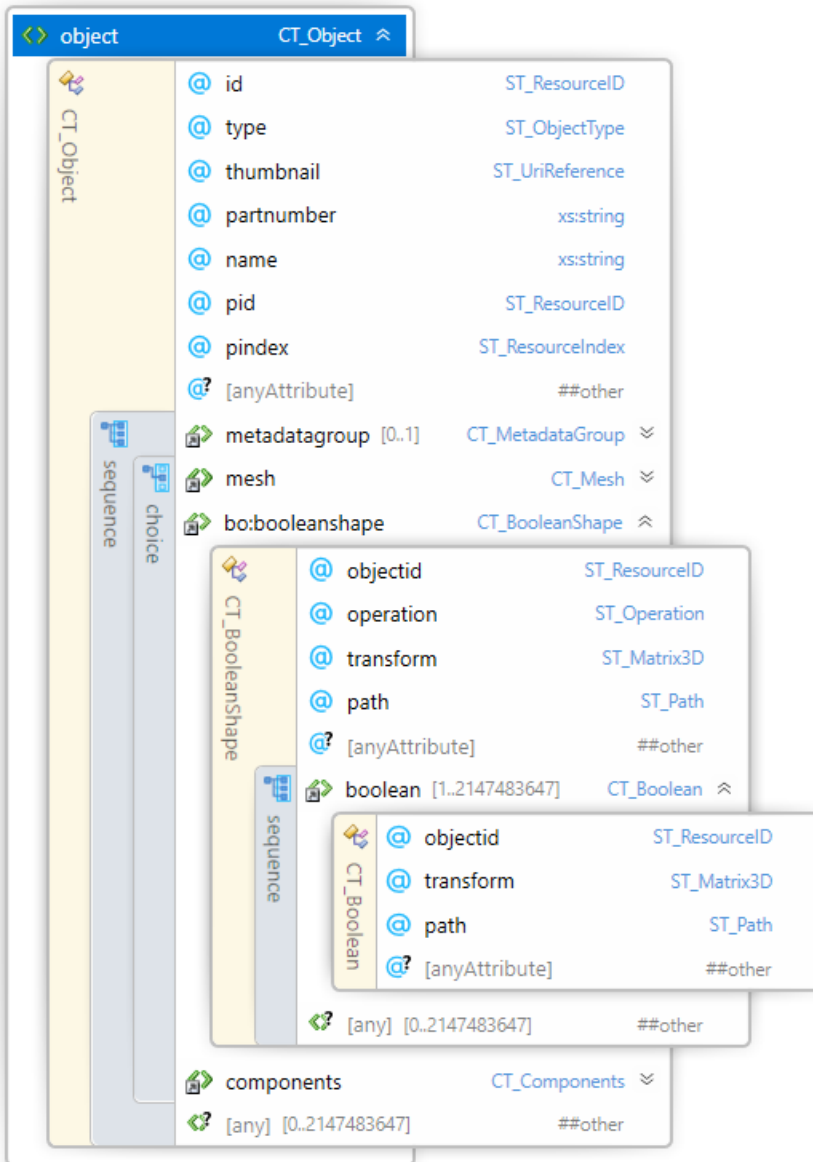


This document describes a new element `<booleanshape>` in the `<object>` elements choice that specifies a new object type, other than a mesh shape or components. This element is OPTIONAL for producers but MUST be supported by consumers that specify support for the 3MF Boolean Operations Extension.

The `<booleanshape>` element defines a new object shape referencing a base object to perform boolean operations by the meshes defined in the `<boolean>` elements.

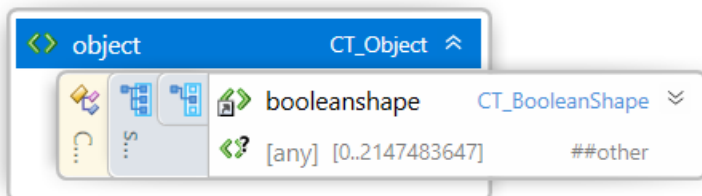
This is a non-backwards compatible change since it declares a different type of object. Therefore, a 3MF package which uses *booleanshape* objects MUST enlist the 3MF Boolean Operations Extension as “required extension”, as defined in the core specification.

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



## Chapter 2. Object Resources

Element <object>



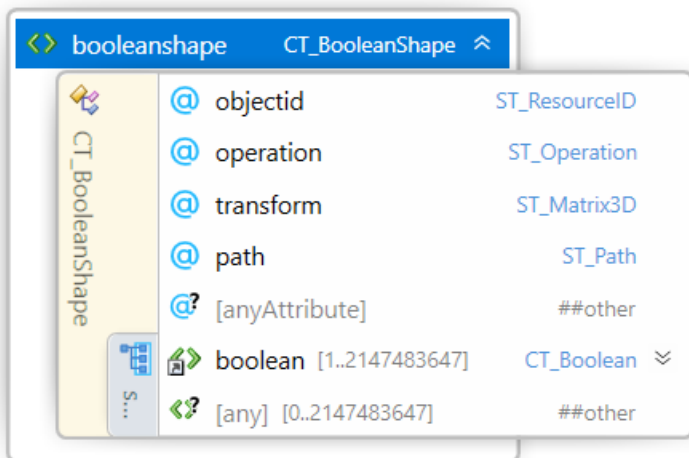
The <object> element is enhanced with an additional element <booleanshape> in the object choice, declaring that the object represents a *boolean shape* defining boolean operations, instead of a *mesh shape* or *components* defining assemblies, This extends [the 3MF Core Specification object resources](#)

Similarly as defined in [the 3MF Core Specification object resources](#), producers MUST NOT assign pid or pindex attributes to objects that contain *booleanshape*. This ensures that an object with no material will not be split into two representations with different materials due

to being referenced as a boolean in multiple objects.

## 2.1. Boolean Shape

Element <booleanshape>



Name	Type	Use	Default	Annotation
objectid	<b>ST_ResourceID</b>	required		It references the base object id to apply the boolean operation.
operation	<b>ST_Operation</b>		union	The boolean operation: union, difference and intersection.
transform	<b>ST_Matrix3D</b>			A matrix transform (see <a href="#">3.3. 3D Matrices</a> ) applied to the base object.
path	<b>ST_Path</b>			A file path to the base object file being referenced. The path is an absolute path from the root of the 3MF container.

@anyAttribute

The optional <booleanshape> element contains one or more <boolean> elements to perform an ordered sequence of boolean operations onto the referenced base object.

**objectid** - Selects the base object to apply the boolean operation. The object MUST be an object of type "model" defining a shape: mesh, booleanshape, or shapes defined in other 3MF extensions. It MUST NOT reference a components object. When used in combination with [the 3MF Production extension](#), it MUST NOT reference any object containing Alternatives.

**operation** - The boolean operation to perform. The options for the boolean shapes are the following:

1. *union*. The resulting object shape is defined as the merger of the shapes. The resulting object surface property is defined by the property of the surface property defining the outer surface, as defined by [the 3MF Core Specification overlapping order](#)

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

2. *difference*. The resulting object shape is defined by the shape in the base object shape that is not in any other object shape. The resulting object surface property, where overlaps, is defined by the object surface property of the subtracting object(s), or no-property when the subtracting object has no property defined in that surface.

$$\text{difference}(\text{base},a,b,c) = \text{base} - (a \cup b \cup c) = ((\text{base} - a) - b) - c$$

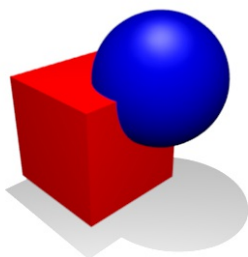
3. *intersection*. The resulting object shape is defined as the common (clipping) shape in all objects. The resulting object surface property is defined as the object surface property of the object defining the new surface, or no-property when that object has no property defined in the new surface.

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

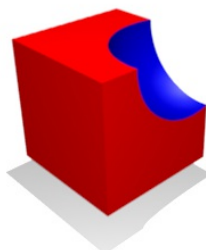
**transform** - The transform to apply to the selected base object.

**path** - When used in conjunction with [the 3MF Production extension](#), the "path" attribute references objects in non-root model files. Path is an absolute path to the target model file inside the 3MF container that contains the target object. The use of the path attribute in a <booleanshape> element is ONLY valid in the root model file.

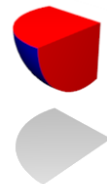
The following diagrams, from the **CSG** Wikipedia, show the three operations:



**union:** Merger of two objects into one



**difference:** Subtraction of object from another one



**intersection:** Portion common to objects

### 2.1.1. Boolean

Element <boolean>

<> boolean
CT\_Boolean ^

CT_Boolean	@ objectid	ST_ResourceID
@ transform	ST_Matrix3D	
@ path	ST_Path	
@? [anyAttribute]	##other	

Name	Type	Use	Default	Annotation
objectid	<b>ST_ResourceID</b>	required		It references the mesh object id performing the boolean operation.
transform	<b>ST_Matrix3D</b>			A matrix transform (see <a href="#">3.3. 3D Matrices</a> ) applied to the referenced object.
path	<b>ST_Path</b>			A file path to the model file being referenced. The path is an absolute path from the root of the 3MF container.
@anyAttribute				

The <boolean> element selects a pre-defined object resource to perform a boolean operation to the base object referenced in the enclosing <booleanshape> element. The boolean operation is applied in the sequence order of the <boolean> element.

**objectid** - Selects the object with the mesh to apply the boolean operation. The object MUST be only a triangle mesh object of type "model", and MUST NOT contain shapes defined in any other extension. When used in combination with [the 3MF Production extension](#), it MUST NOT reference any object containing Alternatives.

**transform** - The transform to apply to the selected object before the boolean operation.

**path** - When used in conjunction with [the 3MF Production extension](#), the "path" attribute references objects in non-root model files. Path is an absolute path to the target model file inside the 3MF container that contains the target object. The use of the path attribute in a <boolean> element is ONLY valid in the root model file.

The boolean operations are sequentially applied in the order defined by the <boolean> sequence, and they follow the fill rule conversion defined by [the 3MF Core Specification fill rule](#).

## Part II. Appendices

### 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.3mf.io/3dmanufacturing/booleanoperations/2023/07"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://schemas.3mf.io/3dmanufacturing/booleanoperations/2023/07"
  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: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_Object">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="booleanshape"/>
    </xs:choice>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="2147483647"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="CT_BooleanShape">
  <xs:sequence>
    <xs:element ref="boolean" maxOccurs="2147483647"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="2147483647"/>
  </xs:sequence>
  <xs:attribute name="objectid" type="ST_ResourceID" use="required"/>
  <xs:attribute name="operation" type="ST_Operation" default="union"/>
  <xs:attribute name="transform" type="ST_Matrix3D"/>
  <xs:attribute name="path" type="ST_Path"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:complexType name="CT_Boolean">
  <xs:attribute name="objectid" type="ST_ResourceID" use="required"/>
  <xs:attribute name="transform" type="ST_Matrix3D"/>
  <xs:attribute name="path" type="ST_Path"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<!-- Simple Types -->

```





model.

## 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.3mf.io/3dmanufacturing/booleanoperations/2023/07"
requiredextensions="bo" unit="millimeter" xml:lang="en-US">
  <resources>
    <basematerials id="2">
      <base name="Red" displaycolor="#FF0000" />
      <base name="Green" displaycolor="#00FF00" />
      <base name="Blue" displaycolor="#0000FF" />
    </basematerials>
    <object id="3" type="model" name="Cube" pid="2" pindex="0">
      <mesh>
        <vertices>...</vertices>
        <triangles>...</triangles>
      </mesh>
    </object>
    <object id="4" type="model" name="Sphere" pid="2" pindex="2">
      <mesh>
        <vertices>...</vertices>
        <triangles>...</triangles>
      </mesh>
    </object>
    <object id="5" type="model" name="Cylinder" pid="2" pindex="1">
      <mesh>
        <vertices>...</vertices>
        <triangles>...</triangles>
      </mesh>
    </object>
    <object id="6" type="model" name="Intersected">
      <bo:booleanshape objectid="3" operation="intersection" transform="0.0741111 0 0 0
0.0741111 0 0 0 0.0741111 2.91124 -0.400453 1.60607">
        <bo:boolean objectid="4" transform="0.0741111 0 0 0 0.0741111 0 0 0 0.0741111
2.91124 -0.400453 1.60607"/>
      </bo:booleanshape>
    </object>
    <object id="10" type="model" name="Full part">
      <bo:booleanshape objectid="6" operation="difference">
        <bo:boolean objectid="5" transform="0.0271726 0 0 0 0 0.0271726 0 -0.0680034 0
4.15442 3.58836 5.23705" />
        <bo:boolean objectid="5" transform="0.0272014 0 0 0 0.0272012 0 0 0 0.0680035
4.05357 6.33412 3.71548" />
        <bo:boolean objectid="5" transform="0 0 -0.0272013 0 0.0272013 0 0.0680032 0 0
5.05103 6.32914 3.35287" />
      </bo:booleanshape>
    </object>
  </resources>
</model>
```

```
</bo:booleanshape>
</object>
</resources>
<build>
  <item objectid="10" transform="25.4 0 0 0 25.4 0 0 0 25.4 0 0 0" />
</build>
</model>
```

## References

### CSG

Wikipedia, the free encyclopedia: Constructive solid geometry

[https://en.wikipedia.org/wiki/Constructive\\_solid\\_geometry](https://en.wikipedia.org/wiki/Constructive_solid_geometry)

### 3MF Core Specification references

See the 3MF Core Specification references

[https://github.com/3MFConsortium/spec\\_core/blob/1.2.3/3MF%20Core%20Specification.md#references](https://github.com/3MFConsortium/spec_core/blob/1.2.3/3MF%20Core%20Specification.md#references).