INTERNATIONAL STANDARD

ISO/ASTM 52915

Second edition
2016-02-15

Specification for Additive Manufacturing File Format (AMF) Version 1.2

Spécification normalisée pour le format de fichier pour la fabrication additive (AMF) Version 1.2

Reference number
ISO/ASTM 52915:2016(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

ASTM International
100 Barr Harbor Drive, PO Box C700
West Conshohocken, PA 19428-2959, USA
Tel. +610 832 9634
Fax +610 832 9635
khooper@astm.org
www.astm.org

Contents

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/TC 261, *Additive manufacturing*, in cooperation with ASTM F 42.91, *Terminology*, on the basis of a partnership agreement between ISO and ASTM International with the aim to create a common set of ISO/ASTM standards on Additive Manufacturing.

This second edition cancels and replaces the first edition (ISO/ASTM 52915:2013), which has been technically revised. This revision contains changes to normative language and details of a minimum implementation, as well as corrections and clarifications.

Introduction

This International Standard describes an interchange format to address the current and future needs of additive manufacturing technology. For the last three decades, the stereolithography (STL) file format has been the industry standard for transferring information between design programs and additive manufacturing equipment. An STL file defines only a surface mesh and has no provisions for representing colour, texture, material, substructure and other properties of the fabricated object. As additive manufacturing technology is evolving quickly from producing primarily single-material, homogeneous objects to producing geometries in full colour with functionally-defined gradations of materials and microstructures, there is a growing need for a standard interchange file format that can support these features.

The Additive Manufacturing File Format (AMF) has many benefits. It describes an object in such a general way that any machine can build it to the best of its ability, and as such is technology independent. It is easy to implement and understand, scalable and has good performance. Crucially, it is both backwards compatible, allowing any existing STL file to be converted, and future compatible, allowing new features to be added as advances in technology warrant.

Specification for Additive Manufacturing File Format (AMF) Version 1.2

1 Scope

This International Standard provides the specification for the Additive Manufacturing File Format (AMF), an interchange format to address the current and future needs of additive manufacturing technology.

The AMF may be prepared, displayed and transmitted provided the requirements of this specification are met. When prepared in a structured electronic format, strict adherence to an extensible markup language (XML)[1] schema is required to support standards-compliant interoperability.

A W3C XML schema definition (XSD) for the AMF is available from ISO from http://standards.iso.org/iso/52915 and from ASTM from www.astm.org/MEETINGS/images/amf.xsd. An implementation guide for such an XML schema is provided in Annex A.

It is recognized that there is additional information relevant to the final part that is not covered by the current version of this International Standard. Suggested future features are listed in Annex B.

This International Standard does not specify any explicit mechanisms for ensuring data integrity, electronic signatures and encryptions.

2 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

2.1
AMF consumer
software reading (parsing) the Additive Manufacturing File Format (AMF) file for fabrication, visualization or analysis

Note 1 to entry: AMF files are typically imported by additive manufacturing equipment, as well as viewing, analysis and verification software.

2.2
AMF editor
software reading and rewriting the Additive Manufacturing File Format (AMF) file for conversion

Note 1 to entry: AMF editor applications are used to convert an AMF from one form to another, for example, convert all curved triangles to flat triangles or convert porous material specification into an explicit mesh surface.

2.3
AMF producer
software writing (generating) the Additive Manufacturing File Format (AMF) file from original geometric data

Note 1 to entry: AMF files are typically exported by computer-aided design (CAD) software, scanning software or directly from computational geometry algorithms.

2.4
attribute
characteristic of data, representing one or more aspects or descriptors of the data in an element

Note 1 to entry: In the XML framework, attributes are characteristics of elements.

2.5
comments
all text elements associated with any data within the Additive Manufacturing File Format (AMF) to be ignored by import software

Note 1 to entry: Comments are used for enhancing human readability of the file and for debugging purposes.

2.6
element
information unit within an XML document consisting of a start tag, an end tag, the content between the tags and any attributes

Note 1 to entry: In the XML framework, an element can contain data, attributes and other elements.

2.7
extensible markup language
XML
standard from the WorldWideWeb Consortium (W3C) that provides for tagging of information content within documents offering a means for representation of content in a format that is both human and machine readable

Note 1 to entry: Through the use of customizable style sheets and schemas, information can be represented in a uniform way, allowing for interchange of both content (data) and format (metadata).

[SOURCE: ISO/ASTM 52900:2015, 2.4.7]

2.8
STL
stereolithography
file format for model data describing the surface geometry of an object as a tessellation of triangles used to communicate 3D geometries to machines in order to build physical parts

Note 1 to entry: The STL file format was originally developed as part of the CAD package for the early stereolithography apparatus, thus referring to that process. It is sometimes also described as “Standard Triangulation Language” or “Standard Tessalation Language”, though it has never been recognized as an official standard by any standardization organization.

[SOURCE: ISO/ASTM 52900:2015, 2.4.16]

3 Key considerations

3.1 General

3.1.1 There is a natural trade-off between the generality of a file format and its usefulness for a specific purpose. Thus, features designed to meet the needs of one community may hinder the usefulness of a file format for other uses. To be successful across the field of additive manufacturing, the file format described in this International Standard, the AMF, is designed to address the concerns listed in 3.1.2 to 3.1.7.

3.1.2 **Technology independence.** The AMF describes an object in such a general way that any machine can build it to the best of its ability. It is resolution and layer-thickness independent and does not contain information specific to any one manufacturing process or technique. This does not negate the inclusion of features that describe capabilities that only certain advanced machines support (for example, colour, multiple materials), but these are defined in such a way as to avoid exclusivity.

3.1.3 **Simplicity.** The AMF is easy to implement and understand. The format can be read and debugged in a simple text viewer to encourage comprehension and adoption. Identical information is not stored in multiple places.

3.1.4 Scalability. The file size and processing time scales well with the increase in part complexity and with the improving resolution and accuracy of manufacturing equipment. This includes being able to handle large arrays of identical objects, complex periodic internal features (for example, meshes and lattices) and smooth curved surfaces when fabricated with very high resolution.

3.1.5 Performance. The AMF enables reasonable duration (interactive time) for read-and-write operations and reasonable file sizes for a typical large object. Detailed performance data are provided in Annex B.

3.1.6 Backwards compatibility. Any existing STL file can be converted directly into a valid AMF file without any loss of information and without requiring any additional information. AMF files are also easily converted back to STL for use on legacy systems, although advanced features will be lost. This format maintains the triangle-mesh geometry representation to take advantage of existing optimized slicing algorithms and code infrastructure already in existence.

3.1.7 Future compatibility. To remain useful in a rapidly changing industry, this file format is easily extensible while remaining compatible with earlier versions and technologies. This allows new features to be added as advances in technology warrant, while still working flawlessly for simple homogeneous geometries on the oldest hardware.

3.2 Guidelines for the inclusion of future new elements

3.2.1 Any new element proposed shall be applicable across all hardware platforms and technologies that could conceivably be used to generate the desired outcome.

3.2.2 In support of the consideration above, new elements proposed for this International Standard shall describe the final object, not how to build it. For instance, a hypothetical future element `<hollow>` might be allowed to tell an additive manufacturing system to leave the volume empty if possible. However, an element `<objectLayerFillPath>` that describes how to build a hollow volume shall not be included since it assumes a particular fabrication process.

4 Structure of this specification

4.1 Format. Information specified throughout this specification is stored in XML 1.0 format. XML is a text file comprising a list of elements and attributes. Using this widely accepted data format allows for the use of many tools for creating, viewing, manipulating, parsing and storing AMF files. XML is human-readable, which makes debugging errors in the file possible. XML can be compressed or encrypted or both if desired in a post-processing step using highly optimized standardized routines.

4.2 Flexibility. Another significant advantage of XML is its inherent flexibility. Missing or additional parameters do not present a problem for a parser as long as the document conforms to the XML standard. Practically, the use of XML namespaces allows new features to be added without breaking old versions of the parser, such as in legacy software.

4.3 Precision. This file format is agnostic as to the precision of the representation of numeric values. It is the responsibility of the generating program to write as many or as few digits as are necessary for proper representation of the target object. However, an AMF consumer should read and process real numbers in double precision (64 bits).

4.4 Future amendments and additions. While additional XML elements can be added provisionally to any AMF file for internal purpose, such additions shall not be considered part of this specification. An unofficial AMF element may be ignored by any AMF consumer and may not be stored or reproduced by an editor application. An element becomes official only when it is formally accepted into this specification.

5 General structure

5.1 The AMF file shall begin with the XML declaration line specifying the XML version and encoding, for example:

<?xml version="1.0" encoding="UTF-8"?>

The XML version shall be 1.0. Only UTF-8 and UTF-16 should be specified. Unrecognized encodings should cause the file to fail to load.

5.2 Whitespace characters and standard XML comments may be interspersed in the file and shall be ignored by any interpreter, for example:

<!-- ignore this comment -->

5.3 The remainder of the file shall be enclosed between start `</amf>` and end `</amf>` element tags. This element denotes the file type and fulfils the requirement that all XML files have a single root element. A version attribute denoting the version of the AMF standard the file is compliant with should be used. Standard XML namespace attributes may also be used, such as the lang attribute designed to identify the natural human language used. The unit system may also be specified (millimetre, inch, foot, metre or micron). In the absence of a unit specification, the attribute value millimetres is assumed, for example:

<amf unit="millimeter" version="1.0" xml:lang="en"

xmlns:amf="www.astm.org/Standards/F2915-14">

5.4 Enclosed within the `<amf/>` element start- and end-tags, there are five top level elements, as described in 5.4.1 to 5.4.5.

5.4.1 `<object>` The object element defines a volume or volumes of material, each of which might also reference a material identifier (ID) for AM processing. The object element shall also declare an object ID, which shall be unique. At least one object element shall be present in the file. Additional objects are optional.

5.4.2 `<material>` The optional material element defines one material for fabrication, each of which declares an associated material ID. The material ID declared shall be unique and shall not be 0. If no material element is included, a single default material is assumed.

5.4.3 `<texture>` The optional texture element defines one image or texture for colour or texture mapping, each of which declares an associated texture ID. The texture ID declared shall be unique.

5.4.4 `<constellation>` The optional constellation element hierarchically combines objects and other constellations into a relative pattern for printing. The constellation element may also declare an object ID, which shall be unique. If no constellation elements are specified, each object element shall be imported with no relative position data. The consumer software may determine the relative positioning of the objects if more than one object is specified in the file.

5.4.5 `<metadata>` The optional metadata element specifies additional information about the object(s) and elements contained in the file.

5.5 Only a single object element is required for a fully functional AMF file.

6 Geometry specification

6.1 General

6.1.1 The top level <object> element declares a unique ID and shall contain one child <mesh> element. The <mesh> element shall contain two child elements: <vertices> and <volume>. The <object> element may optionally reference a material.

6.1.2 The required <vertices> element shall contain all vertices that are used in this object. Each vertex is implicitly assigned an identifying integer in the order in which it is declared, starting at zero and increasing monotonically. The required child element <coordinates> gives the position of the vertex in three-dimensional (3D) space using the <x>, <y> and <z> child elements.

6.1.3 After the vertex information, at least one <volume> element shall be included. Each volume encapsulates a closed volume of the object. Multiple volumes may be included in a single object. Volumes may share vertices at interfaces but shall not have any overlapping volume.

6.1.4 Within each volume, multiple child <triangle> elements shall be used to define the triangles that tessellate the surface of the volume. Each <triangle> element shall reference three vertices from the set of indices of the previously defined vertices. The indices of the three vertices of the triangles shall be specified using the <v1>, <v2> and <v3> child elements. The vertices shall be ordered according to the right-hand rule such that vertices are listed in counter-clockwise order as viewed from the outside of the volume. Each triangle is implicitly assigned an identifying integer in the order in which it was declared starting at zero and increasing monotonically (see Figure 1).

6.1.5 The geometry shall not be used to describe support structure. Only the final target structure shall be described.

```
<?xml version="1.0" encoding="UTF-8"?>
<amf unit="millimeter">
  <object id="0">
    <mesh>
      <vertices>
        <vertex>
          <coordinates>
            <x>0</x>
            <y>1.32</y>
            <z>3.715</z>
          </coordinates>
        </vertex>
        <vertex>
          <coordinates>
            <x>0</x>
            <y>1.269</y>
            <z>2.45354</z>
          </coordinates>
        </vertex>
        ...
      </vertices>

      <volume>
        <triangle>
          <v1>0</v1>
          <v2>1</v2>
          <v3>3</v3>
        </triangle>
        <triangle>
          <v1>1</v1>
          <v2>0</v2>
          <v3>4</v3>
        </triangle>
        ...
      </volume>
    </mesh>
  </object>
</amf>
```


NOTE The figure shows a basic AMF file containing only a list of vertices and triangles. This structure is compatible with the STL standard and can be readable by a minimal implementation of an AMF consumer.

Figure 1 — Basic AMF file

6.2 Smooth geometry

6.2.1 By default, all triangles shall be assumed to be flat and all triangle edges shall be assumed to be straight lines connecting their two vertices. However, curved triangles and curved edges may optionally be specified to reduce the number of mesh elements required to describe a curved surface. Minimal AMF consumer software (see Clause 13) may ignore curvature information associated with triangles.

6.2.2 During import, a curved triangle patch shall be recursively subdivided into four triangles to generate a final temporary set of flat triangles. The depth of recursion shall be exactly five (that is, a single curved triangle will be converted into 1 024 flat triangles).

6.2.3 During production, the producing software that generates curved triangles shall determine automatically the number of curved triangles required to specify the target geometry to the desired tolerance, knowing that the consuming software will perform five levels of subdivision for any curved triangle.

6.2.4 To specify curvature, a vertex may contain a child element `<normal>` to specify the desired surface normal at the vertex. The normal should be unit length and pointing outwards. If this normal is specified, all triangle edges meeting at that vertex shall be curved so that they are perpendicular to that normal and in the plane defined by the normal and the original straight edge.

6.2.5 If a vertex is referenced by two volumes, the normal is considered identically for each volume, but its direction should be interpreted as consistent with the volume in consideration (so that it is pointing

