

# STL (file format)

From Wikipedia, the free encyclopedia

**STL (STereoLithography)** is a file format native to the stereolithography CAD software created by 3D Systems.<sup>[1][2][3]</sup> STL has several after-the-fact backronyms such as "Standard Triangle Language" and "Standard Tessellation Language".<sup>[4]</sup> This file format is supported by many other software packages; it is widely used for rapid prototyping, 3D printing and computer-aided manufacturing.<sup>[5]</sup> STL files describe only the surface geometry of a three-dimensional object without any representation of color, texture or other common CAD model attributes. The STL format specifies both ASCII and binary representations. Binary files are more common, since they are more compact.<sup>[6]</sup>

An STL file describes a raw unstructured triangulated surface by the unit normal and vertices (ordered by the right-hand rule) of the triangles using a three-dimensional Cartesian coordinate system. STL coordinates must be positive numbers, there is no scale information, and the units are arbitrary.<sup>[7]</sup>

## Contents

- 1 ASCII STL
- 2 Binary STL
- 3 Color in binary STL
- 4 The facet normal
- 5 History of use
- 6 Use in other fields
- 7 Software handling STL
- 8 See also
- 9 References
- 10 External links

## ASCII STL

An ASCII STL file begins with the line

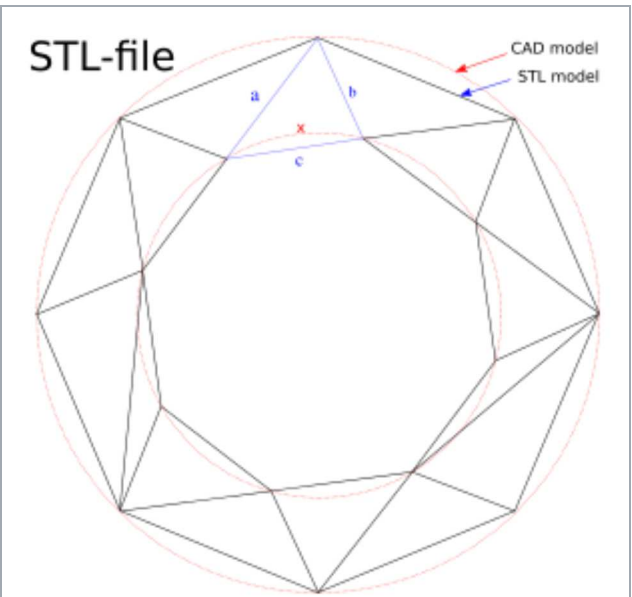
```
{solid name
```

where *name* is an optional string (though if *name* is omitted there must still be a space after solid). The file continues with any number of triangles, each represented as follows:

```
{facet normal n_i n_j n_k
```

STL

STL-file



Example of STL vs CAD format

Filename extension	.stl
Developed by	3D Systems
Type of format	Stereolithography

```

outer loop
    vertex v1_x v1_y v1_z
    vertex v2_x v2_y v2_z
    vertex v3_x v3_y v3_z
endloop
endfacet

```

where each  $n$  or  $v$  is a floating-point number in sign-mantissa-"e"-sign-exponent format, e.g., "2.648000e-002" (noting that each  $v$  must be non-negative). The file concludes with

```
endsolid name
```

The structure of the format suggests that other possibilities exist (e.g., facets with more than one "loop", or loops with more than three vertices). In practice, however, all facets are simple triangles.

White space (spaces, tabs, newlines) may be used anywhere in the file except within numbers or words. The spaces between "facet" and "normal" and between "outer" and "loop" are required.<sup>[6]</sup>

## Binary STL

Because ASCII STL files can become very large, a binary version of STL exists. A binary STL file has an 80-character header (which is generally ignored, but should never begin with "solid" because that will lead most software to assume that this is an ASCII STL file). Following the header is a 4-byte unsigned integer indicating the number of triangular facets in the file. Following that is data describing each triangle in turn. The file simply ends after the last triangle.

Each triangle is described by twelve 32-bit floating-point numbers: three for the normal and then three for the X/Y/Z coordinate of each vertex – just as with the ASCII version of STL. After these follows a 2-byte ("short") unsigned integer that is the "attribute byte count" – in the standard format, this should be zero because most software does not understand anything else.<sup>[6]</sup>

Floating-point numbers are represented as IEEE floating-point numbers and are assumed to be little-endian, although this is not stated in documentation.

```

UINT8[80] - Header
UINT32 - Number of triangles
foreach triangle
    REAL32[3] - Normal vector
    REAL32[3] - Vertex 1
    REAL32[3] - Vertex 2
    REAL32[3] - Vertex 3
    UINT16 - Attribute byte count
end

```

## Color in binary STL

There are at least two non-standard variations on the binary STL format for adding color information:

- The VisCAM and SolidView software packages use the two "attribute byte count" bytes at the end of every triangle to store a 15-bit RGB color:
  - bits 0 to 4 are the intensity level for blue (0 to 31),
  - bits 5 to 9 are the intensity level for green (0 to 31),
  - bits 10 to 14 are the intensity level for red (0 to 31),
  - bit 15 is 1 if the color is valid, or 0 if the color is not valid (as with normal STL files).

- The Materialise Magics software uses the 80-byte header at the top of the file to represent the overall color of the entire part. If color is used, then somewhere in the header should be the ASCII string "COLOR=" followed by four bytes representing red, green, blue and alpha channel (transparency) in the range 0–255. This is the color of the entire object, unless overridden at each facet. Magics also recognizes a material description; a more detailed surface characteristic. Just after "COLOR=RGBA" specification should be another ASCII string ",MATERIAL=" followed by three colors (3×4 bytes): first is a color of diffuse reflection, second is a color of specular highlight, and third is an ambient light. Material settings are preferred over color. The per-facet color is represented in the two "attribute byte count" bytes as follows:
  - bits 0 to 4 are the intensity level for red (0 to 31),
  - bits 5 to 9 are the intensity level for green (0 to 31),
  - bits 10 to 14 are the intensity level for blue (0 to 31),
  - bit 15 is 0 if this facet has its own unique color, or 1 if the per-object color is to be used.

The red/green/blue ordering within those two bytes is reversed in these two approaches – so while these formats could easily have been compatible, the reversal of the order of the colors means that they are not – and worse still, a generic STL file reader cannot automatically distinguish between them. There is also no way to have facets be selectively transparent because there is no per-facet alpha value – although in the context of current rapid prototyping machinery, this is not important.

## The facet normal

In both ASCII and binary versions of STL, the **facet normal** should be a unit vector pointing outwards from the solid object. In most software this may be set to (0,0,0), and the software will automatically calculate a normal based on the order of the triangle vertices using the "right-hand rule". Some STL loaders (e.g. the STL plugin for Art of Illusion) check that the normal in the file agrees with the normal they calculate using the right-hand rule and warn the user when it does not. Other software may ignore the facet normal entirely and use only the right-hand rule. Although it is rare to specify a normal that cannot be calculated using the right-hand rule, in order to be entirely portable, a file should both provide the facet normal and order the vertices appropriately. A notable exception is SolidWorks, which uses the normal for shading effects.

## History of use

Stereolithography machines are 3D printers that can build any volume shape as a series of slices. Ultimately these machines require a series of closed 2D contours that are filled in with solidified material as the layers are fused together. A natural file format for such a machine would be a series of closed polygons corresponding to different Z-values. However, since it is possible to vary the layer thicknesses for a faster though less precise build, it was easier to define the model to be built as a closed polyhedron that can be sliced at the necessary horizontal levels.

The STL file format appears capable of defining a polyhedron with any polygonal facet, but in practice it is only ever used for triangles, which means that much of the syntax of the ASCII protocol is superfluous.

To properly form a 3D volume, the surface represented by any STL files must be closed and connected, where every edge is part of exactly two triangles, and not self-intersecting. Since the STL syntax does not enforce this property, it can be ignored for applications where the closedness does not matter. The closedness only matters insofar as the software that slices the triangles requires it to ensure that the resulting 2D polygons are closed. Sometimes such software can be written to clean up small discrepancies by moving vertices that are close together so that they coincide. The results are not predictable, but it is often sufficient.

## Use in other fields

STL file format is simple and easy to output. Consequently, many computer-aided design systems can output the STL file format. Although the output is simple to produce, some connectivity information is discarded.

Many computer-aided manufacturing systems require triangulated models. STL format is not the most memory- and computationally efficient method for transferring this data, but STL is often used to import the triangulated geometry into the CAM system. The format is commonly available, so the CAM system will use it. In order to use the data, the CAM system may have to reconstruct the connectivity.

STL can also be used for interchanging data between CAD/CAM systems and computational environments such as Mathematica.

## Software handling STL

- Clara.io, a free online 3D editor that can import, edit, and export STL files
- CloudCompare, an open-source application for handling STL files
- Maple (software) and Mathematica, technical computing systems that can work with STL files
- MeshLab, a free and open-source cross-platform application for visualizing, processing, and converting three-dimensional meshes to or from the STL file format
- FreeCAD, an open-source CAD program, can import and export STL files
- OpenSCAD, a constructive solid geometry modeler and domain-specific language can both generate and import STL files
- MountainsMap, a micro-topography software exporting profilometer and microscope 3D surfaces in STL
- SketchUp,<sup>[8]</sup> a 3D modeling computer program for a wide range of drawing applications such as architectural, interior design, civil and mechanical engineering
- Blender, a 3D computer graphics software used for creating animated films, visual effects, art, 3D printed models, interactive 3D applications and video games
- CATIA, a multi-platform computer-aided design (CAD)/computer-aided manufacturing (CAM)/computer-aided engineering (CAE) software suite
- SolidWorks, a solid modeling computer-aided design (CAD) and computer-aided engineering (CAE) software
- Solid Edge, a 3D CAD, parametric feature and synchronous technology solid modeling software
- PTC Creo Elements/Pro, a 3D CAD/CAM/CAE feature-based, associative solid modeling software
- Adobe Acrobat 3D, a 3D exchange solution that inserts 3D content in PDF file format.
- Autodesk ReMake, [www.remake.autodesk.com](http://www.remake.autodesk.com) Create 3D models from reality captured with photos or scans. Remake can edit and prepare huge meshes, generated or Imported - its scalable mesh streaming engine can visualize and edit billion polygon mesh files. Reads and writes .STL files
- Paraview, an open-source, multi-platform data analysis and visualization application based on VTK.
- Rhinoceros 3D, a free form surface modeler that utilizes the NURBS mathematical model.

## See also

- Additive Manufacturing File Format (AMF), a newer standard with native support for color, multiple materials, and constellations
- Autodesk 3ds Max
- Autodesk Inventor
- PLY (file format), an alternative file format offering more flexibility than most stereolithography applications
- Sketchup
- SolidWorks

- Wavefront .obj file, a 3D geometry definition file format with **.obj** file extension
- X3D, a royalty-free ISO standard for 3D computer graphics

## References

1. *StereoLithography Interface Specification*, 3D Systems, Inc., July 1988
2. *StereoLithography Interface Specification*, 3D Systems, Inc., October 1989
3. *SLC File Specification*, 3D Systems, Inc., 1994
4. Grimm, Todd (2004), *User's Guide to Rapid Prototyping*, Society of Manufacturing Engineers, p. 55, ISBN 0-87263-697-6. Many names are used for the format: for example, "standard triangle language", "stereolithography language", and "stereolithography tessellation language". Page 55 states, "Chuck Hull, the inventor of stereolithography and 3D Systems' founder, reports that the file extension is for stereolithography."
5. Chua, C. K; Leong, K. F.; Lim, C. S. (2003), *Rapid Prototyping: Principles and Applications* (2nd ed.), World Scientific Publishing Co, ISBN 981-238-117-1 Chapter 6, Rapid Prototyping Formats. Page 237, "The STL (STeroLithography) file, as the de facto standard, has been used in many, if not all, rapid prototyping systems." Section 6.2 STL File Problems. Section 6.4 STL File Repair.
6. Burns, Marshall (1993). *Automated Fabrication*. Prentice Hall. ISBN 978-0-13-119462-5.
7. Fabbers.com, The StL Format: Standard Data Format for Fabbers, reprinted from Marshall Burns, Automated Fabrication, <http://www.ennex.com/~fabbers/StL.asp> stating, "The object represented must be located in the all-positive octant. In other words, all vertex coordinates must be positive-definite (nonnegative and nonzero) numbers. The StL file does not contain any scale information; the coordinates are in arbitrary units."
8. <http://solidutopia.com/export-to-stl-format>

## External links

- The StL Format ([http://www.fabbers.com/tech/STL\\_Format](http://www.fabbers.com/tech/STL_Format)): Standard Data Format for Fabbers
- File Extension STL (<http://fileextension.info/file/stl>): List of software to work with STL file

Retrieved from "https://en.wikipedia.org/w/index.php?title=STL\_(file\_format)&oldid=756198211"

Categories: CAD file formats | 3D graphics file formats

- 
- This page was last modified on 22 December 2016, at 17:08.
  - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.