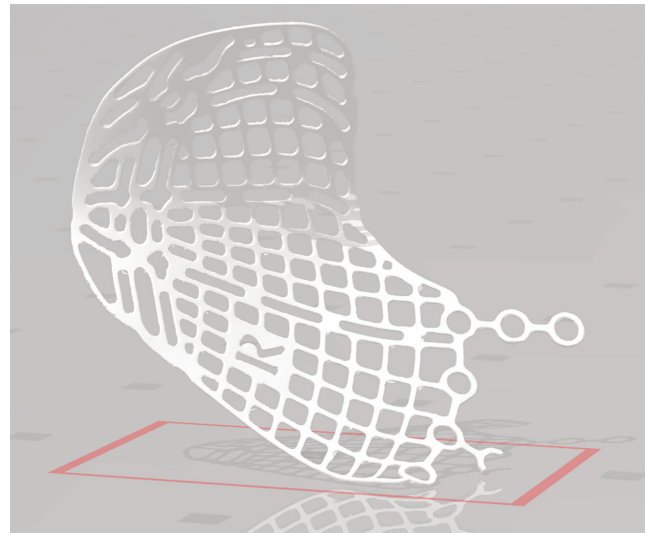
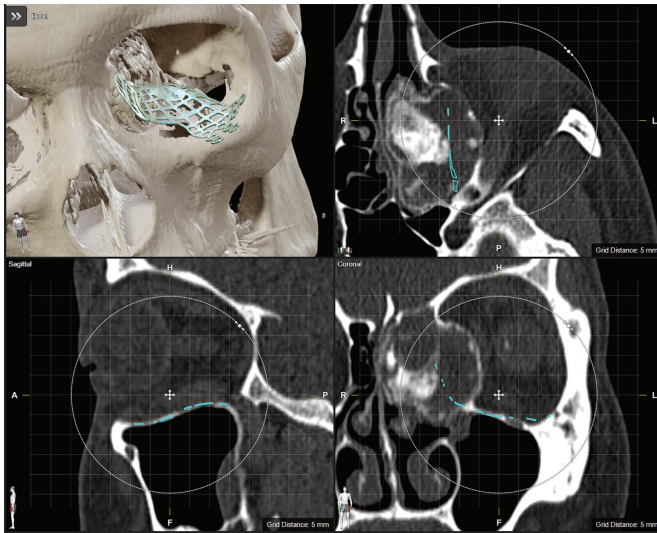


STL Conformance Statement – Brainlab Elements



1. Conformance Statement Overview

This is a conformance statement for the Brainlab Elements software including Elements **Contouring** and **STL Export Performer**. It briefly describes how files stored in the surface tessellation language (STL) are handled and in which format the data is read and written.

The STL import and export part of the application includes the following features:

- Import binary STL files (generic three-dimensional models) to view them in the application.
- Modify imported or generate new 3D objects in the application.
- Export of 3D objects (voxel objects internally stored in DICOM segmentation format and label points internally stored in DICOM surface segmentation) in binary STL format to further use the models in third-party applications.

2. Supported STL File Format

The software is able to import and export STL data in a binary STL format with the possibility to handle color information, stored in the **attribute byte count** as described in 2.3.3.

2.1 General Rules for Import and Export

- STL files selected for import are supposed to be built as surfaces of closed and connected triangles, where every triangle edge is part of exactly two adjacent triangles. This includes single objects as well as fragmented ones, whereas these rules are valid for each fragment.
- The file extension for STL import has to be ".stl". Exported STL files are stored with the same extension.
- The defined color of an object could change during import and export due to a down-sampling of the color information to 5 bit per color (see 2.3.3).
- Brainlab Elements is able to import or export files using the metric unit mm. No unit information is read from or written to the files. Only a plausibility check is performed internally during import regarding the object size.
- The file size for import is only limited by the import quality setting and therefore is connected with the memory consumption needed to reconstruct an object. As a rough estimation, the file size for import should not exceed 300 MB.

NOTE: If an STL object contains parts constructed out of large triangles (e.g., planes or cubic structures), the object import could produce distortions in the resulting voxel object. If this happens, the number of triangles should be increased for this special region.

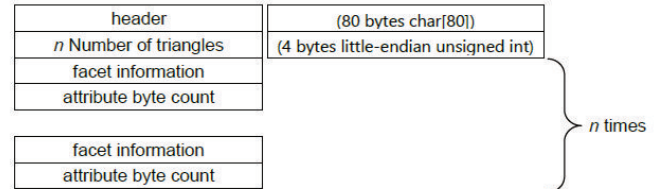
NOTE: Imported and exported objects always have to be inspected for their correct appearance, as the data structure and shape of a model change with the transformation from STL to voxel objects and vice versa.

2.2 Coordinate Systems

- The coordinate system used for STL export of an DICOM object is the DICOM coordinate system of the linked 3D image set.
- An STL object always gets imported into the currently active (i.e., visible) slice set of the Elements **Contouring** application.
- If the coordinate systems of the object and the active slice set do not match (i.e., the bounding box surrounding the object does not completely fit into the volume described by the slice set), the imported object is always shifted into the center of the current slice set in order to prevent partial truncation.
- Always review the position of an imported object for correctness.
- Adjust an incorrect position or orientation using the "Position" functionality.

2.3 Structure of a Binary File Containing "n" Facets (Triangles)

The general structure anticipated for the import and thus used for the export of binary STL formatted data is depicted in the following diagram:



```

UINT8 [80] - Header - 80 bytes
UINT32 - Number of triangles - 4 bytes

For each triangle - 50 bytes:
REAL32 [3] - Normal vector - 12 bytes
REAL32 [3] - Vertex 1 - 12 bytes
REAL32 [3] - Vertex 2 - 12 bytes
REAL32 [3] - Vertex 3 - 12 bytes
UINT16 - Attribute byte count - 2 bytes
  
```

2.3.1 Header

The header of the file contains the following information:

Size in Bytes	Data Type	Description
80	char	Must not begin with "solid" as it identifies the non-supported ASCII STL file type
4	unsigned int	Number of facets in file

2.3.2 Facet Information

The facet information contains the facet "normal" as a unit vector pointing outwards from the solid object and the coordinates of the triangle's vertices. The coordinate system is arbitrary.

Size in Bytes	Data Type	Description
4	float	normal x
4	float	normal y
4	float	normal z
4	float	vertex1 x
4	float	vertex1 y
4	float	vertex1 z
4	float	vertex2 x
4	float	vertex2 y
4	float	vertex2 z
4	float	vertex3 x
4	float	vertex3 y
4	float	vertex3 z

The floating point numbers within supported binary files are represented in little-endian format.

2.3.3 Attribute Byte Count

The attribute byte count is used to read and store additional color information for an object. Even if it would allow encoding multi-colored objects, only single-colored objects are supported by the software. If different colors are defined for the facets, the color of the last read triangle is used for the whole imported object. If no color is defined, a default color is taken for the import.

Size in Bytes	Data Type	Description
2	unsigned short	Used to store facet's color information

In detail, this structure could be sketched as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	0	0	1	1	0	0	1	1	1	0	1	1

Bit	Description
15	1 - if attribute byte count contains color information, 0 - otherwise
10-14	Red intensity (0-31)
5-9	Green intensity (0-31)
0-4	Blue intensity (0-31)

