VTF - Binary

GLview 7

REFERENCE GUIDE

Introduction

This document describes briefly the binary version of the VTF file format. This document should be read in conjunction with the VTF Reference Guide.

General file structure

Format specifiers

Format	Size in bytes	Description
I4	4	Four byte signed integer.
R4	4	Four byte real
C80	80	Description. Characters

File Header

Byte start	Length	Format	Description
0	4	I4	Magic number 1: 231272
4	4	I4	Magic number 2: -160871
8	4	I4	Magic number 3: 251271
12	4	I4	File Version. Current version: 1

File structure

File Header
Block 1
Block End Marker
Block 2
Block End Marker
...
Block n
Block End Marker

Block End Marker

Byte start	Length	Format	Description
0	4	I4	Always -999

Block contents

Block without sub-blocks: (E.g. Nodes)

Block Type
Block ID
Header section: Header size Data size ..... (Block dependent header info)
Data section: Data 1 Data 2 Data 3 ...

Block with sub-blocks: (E.g. Elements)

Block Type
Block ID
Header section: Header size Data size ..... (Block dependent header info)
Data section: Block-dependent sub header Data 1 Data 2 ... Block-dependent sub header Data 1 Data 2 ... .

Item	Description
Block Type (I4)	An integer identifying the block. All blocks have unique types.
Block ID (I4)	A unique ID for the given Block Type. Used for referencing the block.
Block Header	A block-specific header. All block headers have the same first two members: - Header size (I4) in bytes. - Data size (I4) in bytes. - Block specific header information. - .....
Block Data	Block specific data section (Any format) Might contain block dependent sub-headers.

Note:
Total size of block is:
Header Size + Data size + 8 (Block type + Block ID)

Block Types

Block Type	VTF ASCII Token	Description
1001	*NODES	3D coordinates
1006	*INDEXEDFACESET	Block with an indexed face set.
1007	*ELEMENTS	Block with element specification
1008	*GLVIEWGEOMETRY	Block connecting element blocks into geometries.
1009	*RESULTS	Block with 1D or 3D results
1010	*GLVIEWSCALAR	A definition of a GLview scalar result.
1011	*GLVIEWVECTOR	A definition of a GLview vector result.
1013	*TRANSFORMATIONS	A set of transformations matrices.
1014	*VIEWPOINTS	A set of viewpoints.
1016	*2DPLOTSERIES	A 2D plot series.
1021	*USER	A user defined block. Not used by GLview.
1023	*POSITIONRESULTS	A block of results with position in space.
1024	*GLVIEWPOSITIONSCALAR	A definition of a sequence of scalars with pos.
1025	*GLVIEWPOSITIONVECTOR	A definition of a GLview separate vector.
1026	*TRANSFORMATIONRESULT	A transformation matrix
1027	*GLVIEWTRANSFORMATION	A definition of a GLview transformation result
1028	*CROSSECTION	Definition of cross sectional data
1029	*DIRECTIONS	Definition of cross section directions
1030	*PROPERTIES	Properties block used for exporting settings from GLview API
1031	*GLVIEWSTATEINFO	Block describing all the states present in the file
1032	*GLVIEWDISPLACEMENT	A definition of a GLview displacement result.
1033	*2DPLOTDATA	A definition of 2D plot data.

Node Block (Block type: 1001)

This block is used to specify 3D coordinates for elements.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 16)
iDataSize	I4	Size of following data section
iWithID	I4	1 if nodes have user defined IDs, else 0.
iNumNodes	I4	Number of nodes in block

```
struct VTFNodeHeader
{
	int iHeaderSize;
	int iDataSize;
	int iWithID;
	int iNumNodes;
}
```

Data section

Name	Type	Description
iUserID	I4	NB! Only present if iWithID is 1.
fXCoord	R4	X coordinate
fYCoord	R4	Y coordinate
fZCoord	R4	Z coordinate

One of these blocks for each node.

Indexed Face Set Block (Block type: 1006)

This block is used to specify polygons as an indexed face set.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 124)
iDataSize	I4	Size of following data section
iNodeBlockID	I4	ID of the block with nodes for the elements.
szDescription	C80	Block description
fRColor	R4	Red color component (0.0 -> 1.0)
fGColor	R4	Green color component (0.0 -> 1.0)
fBColor	R4	Blue color component (0.0 -> 1.0)
iWithID	I4	1 if polygons have user defined IDs, else 0.
iNumPolygons	I4	Number of polygons in indexed face set block
iNumConnects	I4	Number of connections (edges) in face set block.
iPartID	I4	The ID of the part produced from this block
iMapToNodeIDs	I4	If 1, the polygon nodes are Node IDs. If 0, the polygon nodes are one based indices.

```
struct VTFIndexedFaceSetHeader
{
	int iHeaderSize;
	int iDataSize;
	int iNodeBlockID;
	char szDescription[80];
	float fRColor;
	float fGColor;
	float fBColor;
	int iWithID;
	int iNumPolygons;
	int iPartID;
	int iMapToNodeIDs;
}
```

Data section

Name	Type	Description
iUserID	I4	NB! Only present if iWithID is 1.
iFirstEdgeIndex	I4	
iSecondEdgeIndex	I4	
...		
- iNthEdgeIndex	I4	Last edge in polygon. MUST be negative.

One of these blocks for each polygon.

Note that all indices are one based.

Element Block (Block type: 1007)

This block is used to define elements and element connectivity. Multiple element types within one block is allowed.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 112)
iDataSize	I4	Size of following data section
iNodeBlockID	I4	ID of the block with nodes for the elements.
szDescription	C80	Block description
fRColor	R4	Red color component (0.0 -> 1.0)
fGColor	R4	Green color component (0.0 -> 1.0)
fBColor	R4	Blue color component (0.0 -> 1.0)
iWithID	I4	1 if elements have user defined IDs, else 0.
iNumElementTypes	I4	Number of elements types in block
iSubHeaderSizes	I4	1 if the Element type headers have header sizes.
iPartID	I4	The Part ID of the part created from the block.
iMapToNodeIDs	I4	1 if the element nodes are IDs, 0 if it is 1 based indices.

```
struct VTFElementHeader
{
	int iHeaderSize;
	int iDataSize;
	int iNodeBlockID;
	char szDescription[80];
	float fRColor;
	float fGColor;
	float fBColor;
	int iWithID;
	int iNumElementTypes;
	int iSubHeaderSizes;
	int iPartID;
	int iMapToNodeIDs;
}
```

Data section

Element type header format: (NO HEADER SIZE (OLD FORMAT))

Name	Type	Description
iElementType	I4	Element type (See table below).
iNumElements	I4	Number of elements (of same type).

Element type header format: (WITH HEADER SIZE (NEW FORMAT))

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 20)
iElementType	I4	Element type (See table below).
iNumElements	I4	Number of elements (of same type).
iCrossectionsBlockID	I4	ID of the block with the cross section data for the elements in the group. –1 for none.
iDirectionsBlockID	I4	ID of the block with the direction of the cross section for the elements in the group. –1 for none.

`iNumElementTypes` number of this block in the Element Block.

Element data format:

Name	Type	Description
iUserID	I4	NB! Only present if iWithID is 1.
iNode1	I4	1st element node
iNode2	I4	2nd element node

...		
iNodeN	I4	n^{th} element node (n = number of nodes in element)

One of this block for each element. **Note that all indices are one based.**

Supported element types:

Element Type	VTF ASCII token	Description	Num. nodes
1	%BEAMS	Two node beam elements.	2
2	%BEAMS_3	Three node beam elements.	3
3	%TRIANGLES	Three node triangle elements.	3
4	%TRIANGLES_6	Six node triangle elements.	6
5	%QUADS	Four node quadrangle elements.	4
6	%QUADS_8	Eight node quadrangle elements.	8
7	%TETRAHEDRONS	Four node tetrahedron elements.	4
8	%TETRAHEDRONS_10	Ten node tetrahedron elements.	10
9	%HEXAHEDRONS	Eight node hexahedron elements.	8
10	%HEXAHEDRONS_20	Twenty node hexahedron elements.	20
11	%PENTAHEDRONS	Six node pentahedron elements.	6
12	%PENTAHEDRONS_15	Fifteen node pentahedron elements.	15
18	%POINTS	Single node point element	1
19	%QUADS_9	Center node quadrangle elements	9
20	%PYRAMIDS	Five node pyramid elements	5
21	%PYRAMIDS_13	Thirteen node pyramid elements	13

Geometry Block (Block type: 1008)

The GLview geometry block is used for defining complete geometries from element blocks, and also define a new geometry for each time step.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 96)
iDataSize	I4	Size of following data section
szDescription	C80	Geometry description
iNumSteps	I4	Number of time steps defined
iWithStateID	I4	If 1, the ID for the state must be specified in the sub headers
iWithGeometryIDs	I4	If 1, the ID for the geometry must be specified in the sub headers

```
struct VTFGeometryHeader
{
	int iHeaderSize;
	int iDataSize;
	char szDescription[80];
	int iNumSteps;
	int iStateIDs;
	int iWithGeometryIDs;
}
```

Data section

Header for each step:

Name	Type	Description
iStepNumber	I4	Step number
szStepName	C80	Step description.
fStepTime	R4	The time stamp of the step (-1.0 if not applicable)
iNumElementBlocks	I4	Number of element blocks in step.
iNumIFSBlocks	I4	Number of indexed face sets in step
RESERVED	I4	Reserved for future use. **Must be -1.**
RESERVED	I4	Reserved for future use. **Must be -1.**
iStateID	I4	ID of the state (optional, see iWithStateID)
iGeometryID	I4	ID of the geometry (optional, see iWithGeometryIDs)

One of this block for each step. (Size: 104 or 108 bytes).

Data for each step:

Name	Type	Description
iElementBlockID1	I4	1st element block in step
iElementBlockID2	I4	2nd element block in step
...		
iElementBlockIDN	I4	nth element block (n = number of element blocks in step)
iIndexedFaceSetBlock1	I4	1st indexed face set block in step
iIndexedFaceSetBlock2	I4	2nd indexed face set block in step
...		
iIndexedFaceSetBlockN	I4	nth indexed face set block (n = number of IFS blocks in step)

One of this block for each step.

Result Block (Block type: 1009)

This block is used to specify 1D or 3D results.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 28)
iDataSize	I4	Size of following data section
iDimension	I4	1 for scalar (1D) and 3 for vector (3D) results.
iMapToBlockID	I4	ID of the node or element block that these results should be mapped to.
iMappingType	I4	Mapping type: 0: Results per node. 1: Results per element 2: Results per face (IFS) 3: Results per element node 4: Results per element face 5: Results per element face node
iWithID	I4	1 if explicit results mapping is used, else 0.
iNumResults	I4	Number of values (scalar or vector) in block

```
struct VTFResultsHeader
{
	int iHeaderSize;
	int iDataSize;
	int iDimension;
	int iMapToBlockID;
	int iMappingType;
	int iWithID;
	int iNumResults;
}
```

Data section

Name	Type	Description
iUserID	I4	NB! Only present if iWithID is 1 If node block has user IDs, this is the node ID that the result belongs to. If no IDs are present in the node block, this iUserID is the one based index in the node block.
fScalarValue	R4	Scalar (1D) value
or		
fXVectorComponent	R4	Vector (3D) x component
fYVectorComponent	R4	Vector (3D) y component
fZVectorComponent	R4	Vector (3D) z component

One of these blocks for each result.

GLview Scalar Block (Block type: 1010)

This block is used for grouping the 1D result blocks that belongs to the same step and to define the sequence of the result blocks.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 100)
iDataSize	I4	Size of following data section
szDescription	C80	Result description (name)
iNumSteps	I4	Number of time steps defined
iResultID	I4	ID of the result. If –1, the block ID is used.
iSectionID	I4	ID of the result section. –1 for none.
iWithStateID	I4	If 1, the IDs for the states must be specified in the sub headers. Else 0.

```
struct VTFScalarHeader
{
	int iHeaderSize;
	int iDataSize;
	char szDescription[80];
	int iNumSteps;
	int iResultID;
	int iSectionID;
	int iWithStateID;
}
```

Data section

Header for each step:

Name	Type	Description
iStepNumber	I4	Step number
szStepName	C80	Step description.
fStepTime	R4	The time stamp of the step (-1.0 if not applicable)
iNumResultsBlocks	I4	Number of results blocks in step.
iStateID	I4	ID of the state (optional, see iWithStateID)

One of this block for each step. (Size: 92 or 96 bytes).

Data for each step:

Name	Type	Description
iResultsBlockID1	I4	1st results block in step
iResultsBlockID2	I4	2nd results block in step
...		
iResultsBlockIDN	I4	nth results block (n = number of results blocks in step)

One of this block for each step.

GLview Vector Block (Block type: 1011)

This block is used for grouping the 3D result blocks that belongs to the same step and to define the sequence of the result blocks.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 92)
iDataSize	I4	Size of following data section
szDescription	C80	Result description (name)
iNumSteps	I4	Number of time steps defined
iResultID	I4	ID of the result. If –1, the block ID is used.
iSectionID	I4	ID of the result section. –1 for none.
iWithStateID	I4	If 1, the IDs for the states must be specified in the sub headers. Else 0.

```
struct VTFVectorHeader
{
	int iHeaderSize;
	int iDataSize;
	char szDescription[80];
	int iNumSteps;
	int iResultID;
	int iSectionID;
	int iWithStateID;
}
```

Data section

Header for each step:

Name	Type	Description
iStepNumber	I4	Step number
szStepName	C80	Step description.
fStepTime	R4	The time stamp of the step (-1.0 if not applicable)
iNumResultsBlocks	I4	Number of results blocks in step.
iStateID	I4	ID of the state (optional, see iWithStateID)

One of this block for each step. (Size: 92 or 96 bytes).

Data for each step:

Name	Type	Description
iResultsBlockID1	I4	1st results block in step
iResultsBlockID2	I4	2nd results block in step
...		
iResultsBlockIDN	I4	nth results block (n = number of results blocks in step)

One of this block for each step.

Transformations Block (Block type: 1013)

This block contains transformation matrices for element- and indexed face set blocks. Different transformation matrices may be defined for each step.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 96)
iDataSize	I4	Size of follo wing data section
szDescription	C80	Description
iWithID	I4	1 if a block ID is given for each matrix
iNumSteps	I4	Number of time steps defined

```
struct VTFTransformationHeader
{
	int iHeaderSize;
	int iDataSize;
	char szDescription[80];
	int iWithID;
	int iNumSteps;
}
```

Data section

Header for each step:

Name	Type	Description
iStepNumber	I4	Step number
szStepName	C80	Step description.
fStepTime	R4	The time stamp of the step (-1.0 if not applicable)
iNumElementBlocks	I4	Number of element blocks in step.
iNumIFSBlocks	I4	Number of indexed face sets in step

One of this block for each step. (Size: 96 bytes).

Data for each step:

Name	Type	Description
iElementBlockID1	I4	ID of 1st element block. Only present if iWithID is 1
pfTransformationMatrix1[]	12*R4	Transformation matrix for 1st element block.
iElementBlockID2	I4	ID of 2nd element block. Only present if iWithID is 1
pfTransformationMatrix2[]	12*R4	Transformation matrix for 2nd element block.
...		
iElementBlockIDN	I4	ID of nth element block. Only present if iWithID is 1
pfTransformationMatrixN[]	12*R4	Transformation matrix for nth element block.
iIndexedFaceSetID1	I4	ID of 1st IFS. Only present if iWithID is 1
pfTransformationMatrix1[]	12*R4	Transformation matrix for 1st IFS.
iIndexedFaceSetID2	I4	ID of 2nd IFS. Only present if iWithID is 1
pfTransformationMatrix2[]	12*R4	Transformation matrix for 2nd IFS.
...		
iIndexedFaceSetIDN	I4	ID of nth IFS. Only present if iWithID is 1
pfTransformationMatrixN[]	12*R4	Transformation matrix for nth IFS.

One of this block for each step.

Viewpoint Block (Block type: 1014)

This block contains transformation matrices for element- and indexed face set blocks. Different transformation matrices may be defined for each step.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 92)
iDataSize	I4	Size of following data section
szDescription	C80	Description of the viewpoint series
iNumSteps	I4	Number of time steps defined

```
struct VTFViewpointHeader
{
	int iHeaderSize;
	int iDataSize;
	char szDescription[80];
	int iNumSteps;
}
```

Data section

Header for each step:

Name	Type	Description
iStepNumber	I4	Step number
szStepName	C80	Step description.
fStepTime	R4	The time stamp of the step (-1.0 if not applicable)

One of this block for each step. (Size: 88 bytes).

Data for each step:

Name	Type	Description
eye1	3*R4	The 1st eye position in world coordinates
vrp1	3*R4	The 1st view reference point in world coordinates
vup1	3*R4	The 1st up vector in world coordinates
eye2	3*R4	The 2nd eye position in world coordinates
vrp2	3*R4	The 2nd view reference point in world coordinates
vup2	3*R4	The 2nd up vector in world coordinates
...		

One of this block for each step.

2D Plot Series Block (Block type: 1016)

This block is used to specify x, y coordinates for 2D plotting.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 276)
iDataSize	I4	Size of following data section
szName	C80	The name of the plot
szXAxisName	C80	The name of the X axis
szYAxis Name	C80	The name of the Y axis
fXMin	R4	Minimum value on X axis. Undefined (3e38) for auto scaling.
fXMax	R4	Maximum value on X axis. Undefined (3e38) for auto scaling.
fXUnit	R4	Unit on X axis. Undefined (3e38) for auto scaling.
fYMin	R4	Minimum value on Y axis. Undefined (3e38) for auto scaling.
fYMax	R4	Maximum value on Y axis. Undefined (3e38) for auto scaling.
fYUnit	R4	Unit on Y axis. Undefined (3e38) for auto scaling.
iNumSeries	I4	Number of series in this block

```
struct VTF2DPlotSeriesHeader
{
	int iHeaderSize;
	int iDataSize;
	char szName[80];
	char szXAxisName[80];
	char szYAxisName[80];
	float fXMin;
	float fXMax;
	float fXUnit;
	float fYMin;
	float fYMax;
	float fYUnit;
	int iNumSeries;
}
```

Data section

Header for each step:

Name	Type	Description
szSeriesName	C80	Series description.
fRColor	R4	Red color component (0.0 -> 1.0)
fGColor	R4	Green color component (0.0 -> 1.0)
fBColor	R4	Blue color component (0.0 -> 1.0)
iNumPoints	I4	Number of points (x,y) in this series

One of this block for each series. (Size: 96 bytes).

Data for each step:

Name	Type	Description
1st X value	R4	The 1st X value
1st Y value	R4	The 1st Y value
2nd X value	R4	The 2nd X value
2nd Y value	R4	The 2nd Y value
....		

User Block (Block type: 1021)

This block is a user-defined block that is ignored by GLview. The block can contain any data within the data section.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 8)
iDataSize	I4	Size of following data section

```
struct VTFUserHeader
{
	int iHeaderSize;
	int iDataSize;
}
```

Data section

Any data with the size specified in the header.

Position Results (Block type: 1023)

This block is used to specify 1D or 3D results with a local or global position.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 36)
iDataSize	I4	Size of following data section
iDimension	I4	1 for scalar (1D) and 3 for vector (3D) results.
iMapToBlockType	I4	1: Element Block
iMapToBlockID	I4	ID of the block that these results should be mapped to.
iMappingType	I4	Mapping type: 0 : No mapping info. N/A or implicit mapping 1 : Map to item IDs 2 : Map to item indices.
iGlobalPositions	I4	1 if the positions are given in global coordinates. Else local coordinates are used.
iGlobalResults	I4	1 if the vector results are given in global coordinates. Else local coordinates are used.
iNumResults	I4	Number of values (scalar or vector) in block

```
struct VTFPositionResultsHeader
{
	int iHeaderSize;
	int iDataSize;
	int iDimension;
	int iMapToBlockType;
	int iMapToBlockID;
	int iMappingType;
	int iGlobalPositions;
	int iGlobalResults;
	int iNumResults;
}
```

Data section

Name	Type	Description
iMap	I4	NB! Only present if iMappingType != 0. Specifies the ID or index (according to the iMappingType) of the item to map results to.
fXPosition	R4	Postion x component (local or global)
fYPosition	R4	Postion y component (local or global)
fZPosition	R4	Postion z component (local or global)
fScalarValue	R4	Scalar (1D) value
or		
fXVectorComponent	R4	Vector (3D) x component
fYVectorComponent	R4	Vector (3D) y component
fZVectorComponent	R4	Vector (3D) z component

One of these blocks for each result.

GLview Position Scalar Block (Block type: 1024)

This block is used for grouping the 1D position result blocks that belongs to the same step and to define the sequence of the position result blocks.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 92)
iDataSize	I4	Size of following data section
szDescription	C80	Result description
iNumSteps	I4	Number of time steps defined

```
struct VTFPositionScalarHeader
{
	int iHeaderSize;
	int iDataSize;
	char szDescription[80];
	int iNumSteps;
}
```

Data section

Header for each step:

Name	Type	Description
iStepNumber	I4	Step number
szStepName	C80	Step description.
fStepTime	R4	The time stamp of the step (-1.0 if not applicable)
iNumResultsBlocks	I4	Number of position results blocks in step.

One of this block for each step. (Size: 92 bytes).

Data for each step:

Name	Type	Description
iPosResultsBlockID1	I4	1st position results block in step
iPosResultsBlockID2	I4	2nd position results block in step
...		
iPosResultsBlockIDN	I4	nth position results block (n = number of results blocks in step)

One of this block for each step.

GLview Position Vector Block (Block type: 1025)

This block is used for grouping the 3D position result blocks that belongs to the same step and to define the sequence of the position result blocks.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 92)
iDataSize	I4	Size of following data section
szDescription	C80	Result description (name)
iNumSteps	I4	Number of time steps defined

```
struct VTFPositionVectorHeader
{
	int iHeaderSize;
	int iDataSize;
	char szDescription[80];
	int iNumSteps;
}
```

Data section

Header for each step:

Name	Type	Description
iStepNumber	I4	Step number
szStepName	C80	Step description.
fStepTime	R4	The time stamp of the step (-1.0 if not applicable)
iNumResultsBlocks	I4	Number of position results blocks in step.

One of this block for each step. (Size: 92 bytes).

Data for each step:

Name	Type	Description
iPosResultsBlockID1	I4	1st results block in step
iPosResultsBlockID2	I4	2nd results block in step
...		
iPosResultsBlockIDN	I4	nth results block (n = number of position results blocks in step)

One of this block for each step.

Transformation Result Block (Block type: 1026)

This block is used to specify a transformation matrix for one or all blocks within one time step. See the VTF Reference Guide for a specification of the matrix used.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 16)
iDataSize	I4	Size of following data section
iIFSBlockID	I4	ID of the IFS block for this matrix
iElementBlockID	I4	ID of the Element block for this matrix

If both iIFSBlockID and iElementBlockID are -1 then this block applies for all parts in the step it is used.

```
struct VTFTransformationResultsHeader
{
	int iHeaderSize;
	int iDataSize;
	int iIFSBlockID;
	int iElementBlockID;
}
```

Data section

Name	Type	Description
pfTransformationMatrix[ ]	12*R4	Transformation matrix for the block.

Data size is always 48 bytes.

GLview Transformation Block (Block type: 1027)

This block is used for grouping the transformation matrices that belongs to the same step and to define the sequence of the position result blocks.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 96)
iDataSize	I4	Size of following data section
szDescription	C80	Transformation result description
iNumSteps	I4	Number of time steps defined
iWithStateID	I4	If 1, the IDs for the states must be specified in the sub headers. Else 0.

```
struct VTFGLviewTransformationHeader
{
	int iHeaderSize;
	int iDataSize;
	char szDescription[80];
	int iNumSteps;
	int iWithStateID;
}
```

Data section

Header for each step:

Name	Type	Description
iStepNumber	I4	Step number
szStepName	C80	Step description.
fStepTime	R4	The time stamp of the step (-1.0 if not applicable)
iNumTransformationBlocks	I4	Number of blocks in step.
iStateID	I4	ID of the state (optional, see iWithStateID)

One of this block for each step. (Size: 96 bytes).

Data for each step:

Name	Type	Description
iTransfResultsBlockID1	I4	1st transformation results block in step
iTransfResultsBlockID2	I4	2nd transformation results block in step
...		
iTransfResultsBlockIDN	I4	nth transformation results block (n = number of blocks in step)

One of this block for each step.

Cross Section Block (Block type: 1028)

This block contains cross section definition for one element group.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 12)
IDataSize	I4	Size of following data section
iNumCrossections	I4	Number of cross sections defined in the block.

```
struct VTFGLviewTransformationHeader
{
	int iHeaderSize;
	int iDataSize;
	int iNumCrossections;
}
```

Data section

Header for each cross section:

Name	Type	Description
IHeaderSize	I4	Size of cross section sub header
IType	I4	Cross section type
iNumValues	I4	Number of cross section parameters.

One of this block for each cross section. (Size: 12 bytes).

Data for each step:

Name	Type	Description
fCSParam1	I4	1st cross section parameter
fCSParam2	I4	2nd cross section parameter
...		
fCSParamN	I4	nth cross section parameter (n = number of cross sections parameters)

One of this block for each cross section.

Note: See the ASCII documentation for information about the available cross sections and the parameters.

Direction Section Block (Block type: 1029)

This block contains directions of the cross sections for one element group.

Header

Name	Type	Description
IHeaderSize	I4	Size of this header structure (Always: 12)
IDataSize	I4	Size of following data section
iNumDirections	I4	Number of directions defined in the block.

```
struct VTFGLviewTransformationHeader
{
	int iHeaderSize;
	int iDataSize;
	int iNumDirections;
}
```

Data section

Name	Type	Description
FXCoord	R4	Vector X coordinate
FYCoord	R4	Vector Y coordinate
FZCoord	R4	Vector Z coordinate

The vector specified is the global vector that defined the local Z-axis of the cross section.

One of these vectors for each direction vector specified.

Properties Block (Block type: 1030)

This block is used for exporting settings from GLview API. The properties block is a general purpose block for exporting token based settings to a VTF file. The settings from GLview API can be Frame Set attributes, part attributes, view settings, animation settings, etc.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 24)
iDataSize	I4	Size of following data section
iGroupID	I4	ID of the group the properties block belongs to
iClass	I4	Property Class (see below)
iNumProperties	I4	Number of properties in the block
iContextSize	I4	Size of the context specification (including iNumSpecifications) in bytes.

```
struct VTFPropertiesHeader
{
	int iHeaderSize;
	int iDataSize;
	int iGroupID;
	int iClass;
	int iNumProperties;
	int iContextSize;
}
```

Property Classes:

ClassID	Name	Description	Num. IDs
101	View Settings	Settings from VTView	1
102	Background	Settings from VTBackground	1
103	Animation Control	Settings from VTAnimationControl	1
104	Navigation Setup	Settings from VTNavigationHandler	1
105	Lighting	Settings from VTLighting	1
106	Frame Set Attributes	Settings from VTFrameSetAttributes	1
107	Scalar Settings	Settings from VTScalarSettings	2
108	Vector Settings	Settings from VTVectorSettings	2
109	Color Legend Settings	Settings from VTColorLegend	1
110	Clipping Settings	Settings from VTClipPlaneManager	1
111	Set Attributes	Settings from VTFrameSetSetAttributes	1
112	Frame Generator Settings	Settings from VTFrameGenerator	1
113	Cutplane	Settings from VTCutPlaneParameters	1
114	Iso Surface	Settings from VTIsoSurfaceParameters	1
115	Particle Trace	Settings from VTParticleTraceParameters and VTParticleTraceAttributes	1

The Num. IDs field defines the number of IDs used per context specification in the start of the data section.

Data section
First part of the data section defines the context of the settings block. The size is specified in the header. The number of IDs per context is defined by the Class ID of the block.

Context specification:

Name	Type	Description
iNumSpecifications (N)	I4	Number of context defined. Always 1
ID1-1	I4	First ID of the first context
ID1-2	I4	Second ID of the first context
...	I4	
ID N-M	I4	Last ID of the N'th context

Only one context specification per Properties Block (Size: iContextSize bytes from header).

Sub-header for each property:

Name	Type	Description
iHeaderSize	I4	Size of the propery header (Always: 16).
iDataSize	I4	Length of data in bytes
iPropertyID	I4	ID of the property. Property IDs are defined in GLview API header files.
iDataType	I4	Data type of the item (see below)

One of this block for property in the block (Number of items specified in the header).

Data Types:

TypeID	Name	Description
1	VTF_DT_INTEGER	Four byte signed integers
2	VTF_DT_FLOAT	Four byte floats
3	VTF_DT_BOOL	Boolean value (Size: 4 bytes (int), 0 FALSE, 1 TRUE)
4	VTF_DT_STRING	String array
5	VTF_DT_VECTOR	Vector (3*floats)
6	VTF_DT_BYTE_COLOR	Byte color (Size: 12 bytes, each color as an integer)
7	VTF_DT_RAW	Raw data bytes. Size (iDataSize bytes)

The propery data is written directly after the sub-header.

GLview State Info Block (Block type: 1031)

This block is describes all the states present in the file.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 12)
iDataSize	I4	Size of following data section
iNumStates	I4	Number of states defined in this block

```
struct VTFGLviewStateInfoBlock
{
	int iHeaderSize;
	int iDataSize;
	int iNumStates;
}
```

Data section

Name	Type	Description
iStateID	I4	State ID
szStateName	C80	Step description.
fReferenceValue	R4	Load factor/Time/frequency
iRefType	I4	Type of reference value: 0: Time, 1: Frequency, 2: Load case, 3: Other
iGroup	I4	1 if state is a group state, else 0.
iParentID	I4	The ID of the parent of this state. –1 for no parent.

One of this block for each state. (Size: 100 bytes).

GLview Displacement Block (Block type: 1032)

This block is used for grouping the 3D result blocks that belongs to the same step and to define the sequence of the result blocks.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 104)
iDataSize	I4	Size of following data section
szDescription	C80	Result description (name)
iNumSteps	I4	Number of time steps defined
iResultID	I4	ID of the result. If –1, the block ID is used.
iWithStateID	I4	If 1, the IDs for the states must be specified in the sub headers. Else 0.
fDefaultScaleFactor	R4	Scale factor to set when loading the result. Default 1.0.
iRelativeDisplacementResults	I4	If 1, the displacement results are relative to the original node. If 0, the displacement results are new global coordinates (default).

```
struct VTFDisplacementHeader
{
	int iHeaderSize;
	int iDataSize;
	char szDescription[80];
	int iNumSteps;
	int iResultID;
	int iWithStateID;
	float fDefaultScaleFactor;
	int iRelativeDisplacementResults;
}
```

Data section

Header for each step:

Name	Type	Description
iStepNumber	I4	Step number
szStepName	C80	Step description.
fStepTime	R4	The time stamp of the step (-1.0 if not applicable)
iNumResultsBlocks	I4	Number of results blocks in step.
iStateID	I4	ID of the state (optional, see iWithStateID)

One of this block for each step. (Size: 92 or 96 bytes).

Data for each step:

Name	Type	Description
iResultsBlockID1	I4	1st results block in step
iResultsBlockID2	I4	2nd results block in step
...		
iResultsBlockIDN	I4	nth results block (n = number of results blocks in step)

One of this block for each step.

2D Plot data block (Block type: 1033)

This block is used to define a number of data series that can be plotted against each other. The block defines the number of series, and then data for each series is written for each step/state/frequency/etc.

Header

Name	Type	Description
iHeaderSize	I4	Size of this header structure (Always: 96)
iDataSize	I4	Size of following data section
szName	C80	Name of the plot
iNumVariables	I4	Number of variables, i.e. number of values per row.
iRows	I4	Number of time steps/states/frequencies/etc. defined.

```
struct VTF2DPlotDataHeader
{
	int iHeaderSize;
	int iDataSize;
	char szName[80];
	int iNumSeries;
	int iRows;
}
```

Data section

Series names first in the data section

Name	Type	Description
szVariableName1	C80	Name of the first variable
szVariableName2	C80	Name of the second variable
...		
szVariableNameN	C80	Name of the N'th variable (last variable)
iVariableType1	I4	Type of the first variable
iVariableType2	I4	Type of the second variable
...		
iVariableTypeN	I4	Type of the N'th variable (last variable)

The legal values for the variable types are:
0: NONE, 1: TIME, 2: STEP

Then the plot data as follows:

Name	Type	Description
fVal1Variable1	I4	1^{st} value of the 1^{st} varible
fVal1Variable2	I4	1^{st} value of the 2^{nd} varible
...		
fVal1VariableN	I4	1^{st} value of the N^{th} varible
fVal2Variable1	I4	2^{nd} value of the 1^{st} varible
fVal2Variable2	I4	2^{nd} value of the 2^{nd} varible
...		
fVal2VariableN	I4	2^{nd} value of the N^{th} varible
...		

