VTF - ASCII

GLview 7

REFERENCE GUIDE

1. General notes

- ASCII 8-bit text format. Both PC style and UNIX style text files are accepted.
- Default extension : "*.vtf"
- All block identifiers and directives must begin on the first character of the line.
- Comment lines are defined by starting with: "#", "!", ";". The file parser ignores these lines.
- A block starts with "*BLOCK_KEYWORD".
- The order of the blocks in a file is insignificant.
- Lines longer than 256 characters will be truncated.

2. Structure of file

The file must start with a predefined header. The ASCII file header is "*VTF-1.00". The rest of the file consists of one or many blocks, defined by the top-level identifiers (see chapter 3). All top-level identifiers start on the first column of a line and begins with a "*". Each block may contain various directives identified by a "%" character, and data. Each block is (optionally) given an ID that is used for cross-reference between blocks.

The logical structure of a block is shown below.

```
*BLOCK_KEYWORD
%DIRECTIVE 1
%DIRECTIVE 2
....
DATA
DATA
....
```

One directive can only appear once in a block (unless explicitly specified). The block IDs must be unique for one block type.

The data section must contain one item (node, element, polygon, etc.) per line. If user defined element or polygon IDs are used, these IDs must be placed on the beginning of each line.

In the data section the values are given with space separation (spaces or tabs allowed). One exception from this rule is when listing block ID references. They are given as a comma separated list. See the example below.

```
%ELEMENTS
1, 3, 15
```

3. Top-level identifiers

The table below contains a list and a short description of the legal block keywords allowed in the ViewTech file format. These identifiers must start on the first character of a line, and one block lasts to the definition of another block or the end of the file.

Identifier	Description
*VTF-1.00	ASCII file header.
*NODES [<ID>]	A block of nodes (3D coordinates).
*INDEXEDFACESET [<ID>]	A block with polygons as indexed face sets.
*ELEMENTS [<ID>]	A block of elements.
*GLVIEWGEOMETRY [<ID>]	A definition of a GLview geometry.
*RESULTS [<ID>]	A block of results
*GLVIEWSCALAR [<ID>]	A definition of a GLview scalar result.
*GLVIEWVECTOR [<ID>]	A definition of a GLview vector result.
*TRANSFORMATIONS [<ID>]	A set of transformations matrices.
*VIEWPOINTS [<ID>]	A set of viewpoints.
*2DPLOTSERIES [<ID>]	A 2D plot series.
*USER [<ID>]	A user defined block. Not used by GLview.
*POSITIONRESULTS [<ID>]	A block of results with position in space.
*GLVIEWPOSITIONSCALAR [<ID>]	A definition of a sequence of scalars with pos.
*GLVIEWPOSITIONVECTOR [<ID>]	A definition of a GLview separate vector.
*TRANSFORMATIONRESULT [<ID>]	One transformation matrix
*GLVIEWTRANSFORMATION [<ID>]	A definition of a GLview transformation result.
*CROSSECTIONS [<ID>]	Definition of cross sectional data
*DIRECTIONS [<ID>]	Definition of cross section directions
*GLVIEWDISPLACEMENT [<ID>]	A definition of a GLview displacement result.
*SET [<ID>]	An element set.
*2DPLOTDATA [<ID>]	A definition of 2D plot data.

4. Description of each block

4.1. *VTF-1.00

This identifier defines the header of an ASCII VTF-file. It must exist on the first line of the file and be exactly as specified. The four last characters define the version of the file specification. This should be 1.00. This block has no directives and no data.

4.2. *NODES [<ID>]

4.2.1. Description

This keyword identifies a block of 3D node coordinates. The node block may be assigned a block ID that is used to refer to the block. If the `%NO_ID` keyword is used, the nodes are assigned a sequential node ID starting from 1. If node IDs are specified, they must be integers greater or equal to 1, and be unique for the block. The node IDs will be used when the nodes are referred to from another block. The data section must contain one node (3 or 4 values, depending on if IDs are used) per line.

4.2.2. Directives

`%NO_ID`	Default	The node data is just three coordinates for each node, no ID is assigned for each node. Format of data section is `x y z <NewLine> x y z ...`
`%WITH_ID`		Each node has an ID that is used to reference the node. Format of data section `ID x y z <NewLine> ID x y z.`

4.2.3. Example

```
*NODES 1
%WITH_ID
1     0.0 0.0 0.0
2     1.0 0.0 0.0
10    1.0 1.0 0.0
11    0.0 1.0 0.0
3     0.0 1.0 1.0
4     1.0 1.0 1.0
```


4.3. *INDEXEDFACESET [<ID>]

4.3.1. Description

The indexed face set is used to define (parts of) a geometry. The indexed face set specifies polygons by connecting a series of nodes. The node coordinates are not contained in this block, indices are used to refer to nodes in a node block.

4.3.2. Directives

%NAME "Name"		Assign a name to the indexed face set.
%DESCRIPTION "Desc"		Assign a description to the block.
%NODES #ID		Specifies the block that contains the nodes for the elements.
%COLORS <R G B>		Specifies the color for all the faces. Color components (R,G,B) are in the range <0..1>
%NO_ID	Default	The polygons are specified without an ID and are assigned a sequential ID starting on 1. Format of connectivity data section is `n1 n2 ... -nX <NewLine> n1 n2 ... -nX ...`
%WITH_ID		Each polygon has an ID that is used to refer the polygon. Format of connectivity data section is `ID n1 n2 ... -nX <NewLine> ID n1 n2 ... -nX ...`
%MAP_NODE_IDS		Specifies that the polygon nodes are node IDs.
%MAP_NODE_INDICES		Specifies that the polygon nodes are one based indices.

4.3.3. Example

```
*INDEXEDFACESET 1
%NAME "Part 1"
%DESCRIPTION "Test of Indexed face set"
%NODES #1
%WITH_ID
#Two polygons are specified, one with vertices 1,2,10,11 (ID 1001)
#and one with 11,10,3,4 (ID 1002)
1001 1 2 10 -11
1002 11 10 3 -4
```


4.4. *ELEMENTS [<ID>]

4.4.1. Description

This block contains a collection of elements. The legal element types are listed in the next section. The data section must specify one element per line.

4.4.2. Directives

%NAME "Name"		Assign a name to the elements.
%DESCRIPTION "Desc"		Assign a description of the element block.
%NODES #ID		Specifies the block that contains the nodes for the elements.
%NO_ID	Default	The elements are specified without an ID and are assigned a sequencial ID starting on 1. Format of data section is n1 n2 ... nX <NewLine> n1 n2 ...
%WITH_ID		Each element has an ID that is used to reference the element. Format of data section ID n1 n2 ... nX <NewLine> ID n1 n2 ...
%COLORS <R G B>		Specifies the color for all the faces. Color components (R,G,B) are in the range <0..1>
%PART_ID <ID>		Specifies the ID of the part produced from this element block.
%MAP_NODE_IDS		Specifies that the element nodes are node IDs.
%MAP_NODE_INDICES		Specifies that the element nodes are one based node indices.
%CROSSECTIONS		Specifies constant cross section for beam elements. This token must be placed right after the element type. A new cross section can be specified for each element group.
%DIRECTIONS		Specifies directional data for beam elements. This token must be placed right after the element type. A new direction can be specified for each element group.
%POINTS		One node point elements.
%BEAMS		Two node beam elements.
%BEAMS_3		Three node beam elements.
%TRIANGLES		Three node triangle elements.
%TRIANGLES_6		Six node triangle elements.
%QUADS		Four node quadrangle elements.
%QUADS_8		Eight node quadrangle elements.
%QUADS_9		Center node quadrangle elements
%TETRAHEDRONS		Four node tetrahedron elements.
%TETRAHEDRONS_10		Ten node tetrahedron elements.
%HEXAHEDRONS	Default	Eight node hexahedron elements.
%HEXAHEDRONS_20		Twenty node hexahedron elements.
%PENTAHEDRONS		Six node pentahedron elements.

%PENTAHEDRONS_15		Fifteen node pentahedron elements.
%PYRAMIDS		Five node pyramid elements
%PYRAMIDS_13		Thirteen node pyramid elements

4.4.3. Example

```
*ELEMENTS 1
%NAME "Hex elements"
%DESCRIPTION "The hexahedron elements"
%NODES #3
%HEXAHEDRONS
1 2 3 4 5 6 7 8
5 6 7 8 9 10 11 12
```


4.5. *GLVIEWGEOMETRY [<ID>]

4.5.1. Description

The GLview geometry block is used for grouping various geometry blocks into a complete geometry. A GLview geometry can contain many element blocks. The data part of this block refers to block ids (element blocks), and are comma separated.

Adaptive geometries are supported by using the `%STEP <number>` directive. It is not necessary to specify one geometry per step, only when the geometry is changing.

4.5.2. Directives

`%NAME "Name"`		Assign a name to the geometry.
`%DESCRIPTION "Desc"`		Assign a description of the geometry.
`%STEP <StepNo>`		This directive describes that the following geometry belongs to step `<StepNo>`. This directive may appear many times within this block.
`%STEPNAME <"Name">`		Assign a description to the current step. This directive may appear many times within this block.
`%STEPTIME <time>`		Assign a time stamp to the current step. This directive may appear many times within this block.
`%INDEXEDFACESET`	Default	Defines that the subsequent data section is IDs to indexed face sets. This directive may appear many times within this block.
`%ELEMENTS`		Defines that the subsequent data section is IDs to element blocks. This directive may appear many times within this block.
`%GEOMETRY_ID`		Defines the Geometry ID for the given step.

4.5.3. Example

```
*GLVIEWGEOMETRY
%NAME "Sample model"
%DESCRIPTION "A sample model from the vtf file format"
%ELEMENTS
1,3
```


4.6. *RESULTS [<ID>]

4.6.1. Description

This block contains a collection of results for nodes, polygons or elements. The results might be scalar results (1D) or vector results (3D). The block specifies which node, elements, or indexed face set the results belong to and how the results should be bound. The result block contains only results for one block for one time step. Multiple time steps are not allowed within one result block. The data section must only contain one result per line (i.e. one float per line if scalar, three floats per line if vector result).

4.6.2. Directives

%DIMENSION <Dim>	Def : 1	Specifies the dimension of the results. 1 for scalar results and 3 for vector results.
%PER_NODE #ID		The results are bound to the nodes in the node block ID.
%PER_ELEMENT #ID		The results are bound to the elements in the element block ID.
%PER_ELEMENT_NODE #ID		One result for each node in each element.
%PER_ELEMENT_FACE #ID		One result for each element face
%PER_ELEMENT_FACE_NODE #ID		One result for each node in each element face
%PER_FACE #ID		The results are bound to the polygons in the indexed face set block ID.
%NO_ID	Default	The results do not have IDs. The format of the data sectoion is `x1 (y1 z1) <NewLine> x2 (y2 z2) ...`
%WITH_ID		Each result has assigned a unique ID. This ID refers to the ID in the node block. The format of the data sectoion is `ID x1 (y1 z1) <NewLine> ID x2 (y2 z2) ...` If there are no IDs in the node block, the IDs in this block is one based indices into the node block.

4.6.3. Example

```
*RESULTS 2
%DIMENSION 1
%WITH_ID
%PER_NODE #1
1 2.3
2 3.2
3 4.3
4 3.2
10 4.5
11 3.2
```


4.7. *GLVIEWSCALAR [<ID>]

4.7.1. Description

This block is used for grouping the result blocks that belongs to the same step and to define the sequence of the result blocks. The results will be available in as scalar results. If one or more of the result blocks contain vector (3D) data, the length of these vectors will be used as scalar values.

4.7.2. Directives

%NAME "Name"		Assign a name to the scalar result.
%DESCRIPTION "Desc"		Assign a description of the scalar result.
%STEP <StepNo>	Step 1	This directive describes that the following result blocks belong to step <StepNo>. This directive may appear many times within this block.
%STEPNAME <"Name">	Def "Step n"	Assign a description to the current step. This directive may appear many times within this block.
%STEPTIME <time>		Assign a time stamp to the current step. This directive may appear many times within this block.
%RESULT_ID <ID>	Block ID	The ID of the result. If –1 or not present, the block ID is used.
%SECTION_ID <ID>	-1	The ID of the result section. Default –1.

4.7.3. Example

```
*GLVIEWSCALAR 1
%NAME "von Mises"
%DESCRIPTION "von Mises results"
%STEP 1
%STEPNAME "Time: 0.0"
2,3
%STEP 2
%STEPNAME "Time: 1.0"
12, 13
```


4.8. *GLVIEWVECTOR [<ID>]

4.8.1. Description

This block is used for grouping the result blocks that belong to the same frame and to define the sequence of the result blocks. These results will be available in GLview as displacements or vector results.

4.8.2. Directives

%NAME <"Name">		Assign a name to the vector result.
%DESCRIPTION "Desc"|#ID		Assign a description of the vector result.
%STEP <StepNo>	Step 1	This directive describes that the following result blocks belong to step <StepNo>. This directive may appear many times within this block.
%STEPNAME <"Name">	Def "Step n"	Assign a description to the current step. This directive may appear many times within this block.
%STEPTIME <time>		Assign a time stamp to the current step. This directive may appear many times within this block.
%RESULT_ID <ID>	Block ID	The ID of the result. If –1 or not present, the block ID is used.
%SECTION_ID <ID>	-1	The ID of the result section. Default –1.

4.8.3. Example

```
*GLVIEWVECTOR 1
%NAME "Displacements"
%DESCRIPTION "Displacements results"
%STEP 1
%STEPNAME "Time: 0.0"
1, 3
%STEP 2
%STEPNAME "Time: 1.0"
11, 18
```

4.9. *TRANSFORMATIONS [<ID>]

4.9.1. Description

One transformation matrix can be applied to each element or indexed face set block per step. The transformations are specified as 3 by 4 matrices, one for each step. The transformation matrix is in the following format:

$$M = \begin{bmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \\ 41 & 42 & 43 \end{bmatrix}$$

This is equivalent to the standard four by four transformation matrices used with homogeneous coordinates, where the fourth column is taken to be $[0\ 0\ 0\ 1]^T$. The coordinates (x_0, y_0, z_0) are transformed into (x,y,z) as specified in the following equation:

$$[x \quad y \quad z] = [x_0 \quad y_0 \quad z_0 \quad 1] \bullet M$$

The data section should contain three values per line. If Block IDs are used (%WITH_ID), the ID must be specified on the first line alone.

4.9.2. Directives

%NAME <"Name">		Assign a name to the transformations.
%NO_ID	Default	The transformation matrices are assigned to the element blocks, then to the indexed face set blocks.
%WITH_ID		For the element and indexed face set blocks, a Block ID is specified before each matrix, assigning that matrix to the specified block.
%STEP <number>		Specifies that the following matrix belongs to step <number>. This directive may appear many times within this block.
%STEPNAME <"Name">	Def "Step n"	Assign a description to the current step. This directive may appear many times within this block.
%STEPTIME <time>		Assign a time stamp to the current step. This directive may appear many times within this block.
%INDEXEDFACESET		Specifies that the following matrices and (optionally) Block IDs refers to Indexed face set blocks.
%ELEMENTS	(Default)	Specifies that the following matrices and (optionally) Block IDs refers to Element blocks.

4.9.3. Example

```
*TRANSFORMATIONS 1
%NAME "Transformation number 1"
%WITH_ID

%STEP 1
%INDEXEDFACESET
#Identity matrix, the original geometry is shown in step 1
101
1 0 0
0 1 0
0 0 1
0 0 0
102
1 0 0
0 1 0
0 0 1
```



```
0 0 0

%STEP 2
%INDEXEDFACESET
101
1 0 0
0 1 0
0 0 1
2 1 0
102
1 0 0
0 1 0
0 0 1
2 2 0
```


4.10. *VIEWPOINTS [<ID>]

4.10.1. Description

A viewpoint can be specified for a given number of steps. The eye position (eye), view reference point (vrp) and an up vector (vup) specify the viewpoint.

The data section should contain three values per line.

4.10.2. Directives

%NAME <"Name">		Assign a name to the transformations.
%STEP <number>		Specifies that the following viewpoint belongs to step <number>. This directive may appear many times within this block.
%STEPNAME <"Name">	Def "Step n"	Assign a description to the current step. This directive may appear many times within this block.
%STEPTIME <time>		Assign a time stamp to the current step. This directive may appear many times within this block.

4.10.3. Example

```
*VIEWPOINTS 1
%STEP 1
# Eye:
10.0 0.0 0.0
# View direction:
0.0 0.0 0.0
# View Up Vector:
0.0 0.0 1.0
%STEP 2
# Eye:
8.0 0.0 0.0
# View direction:
0.0 0.0 0.0
# View Up Vector:
0.0 0.0 1.0
```


4.11. *2DPLOTSERIES [<ID>]

4.11.1. Description

This block is used to specify x, y coordinates for 2D plotting. The data section should contain one (x,y) data point per line.

4.11.2. Directives

%NAME <"Name">		Assigns the name "name" to the plot. The name is used as a title in the 2D plot window.
%XAXIS_NAME <"name">		Specifies the title of the x-axis.
%YAXIS_NAME <"name">		Specifies the title of the y-axis.
%XAXIS_MIN <min>	Def. Auto	Specifies the minimum value on the x-axis.
%XAXIS_MAX <max>	Def. Auto	Specifies the maximum value on the x-axis.
%XAXIS_UNIT <unit>	Def. Auto	Specifies the unit of the x-axis
%YAXIS_MIN <min>	Def. Auto	Specifies the minimum value on the y-axis.
%YAXIS_MAX <max>	Def. Auto	Specifies the maximum value on the y-axis.
%YAXIS_UNIT <unit>	Def. Auto	Specifies the unit of the y-axis
%SERIES <number>	Def. 1	Specifies that the subsequent data section belongs to series <number>. This directive may appear many times within this block.
%SERIES_NAME <"name">	Def Auto	Specifies the name of the current series. This directive may appear many times within this block.
%SERIES_COLOR <R G B>	Def Auto	Specifies the color of the current series. This directive may appear many times within this block.

4.11.3. Example

```
*2DPLOTSERIES
%NAME "Modal plot of bicubic stresses"
%XAXIS_NAME "Time"
%YAXIS_NAME "Bicubic Stress"
%SERIES 1
%SERIES_NAME "Test-run 1"
1 2
2 5
3 6
4 3.3
5 45.4
%SERIES 2
%SERIES_NAME "Test-run 2"
1 32.1
2 2.1
3 10.1
4 23.1
5 43.2
```


4.12. *USER [<ID>]

4.12.1. Description

This block is a user-defined block that is ignored by GLview. The block can contain any data except for an asterisk (*) as the first character of the line.

4.12.2. Directives

Not available.

4.12.3. Example

```
*USER 1
jfkdl fjsd
fjkldsjfds jfklds fsd
jfkdsjflksdj
1.0 2.0 3.0

*USER 2
dka dklk
12
2
34
```

4.13. *POSITIONRESULTS [<ID>]

4.13.1. Description

This block contains a collection of results with a given position. The results can be bound to an element, and both position and value can be specified in either local or global coordinate systesm. It is possible to have the results independent of elements, but only for global coordinate systems. The result block contains only results for one block for one time step. The data section must only contain one result per line.

4.13.2. Directives

%DIMENSION <Dim>	Def : 1	Specifies the dimension of the results. 1 for scalar results and 3 for vector results.
%MAPTO_ELEMENTS #ID		The results are bound to the elements in the element block ID.

%MAP_NONE	Default	No mapping is specified or implicit mapping is used. If a MAPTO_* is specified, the first results maps to the first item in the MAPTO block etc. Format of data section:
		`px py pz rx1 (ry1 rz1) <NewLine>`
		`px py pz rx2 (ry2 rz2) ...`

%MAP_ID		Each line starts with the ID of the item to map to. Format of the data section:
		`ID px py pz rx (ry rz) <NewLine>`
		`ID px py pz rx (ry rz) ...`

%MAP_INDEX		Each line starts with the ID of the item to map to. Format of the data section:
		`Idx px py pz rx (ry rz) <NewLine>`
		`Idx px py pz rx (ry rz) ...`

%GLOBAL_POSITIONS	Default	All positions are given in global coordinates.

%LOCAL_POSITIONS		All positions are given in local coordinates. Require MAPTO_ELEMENTS specified.
%GLOBAL_RESULTS	Default	All 3D results are given in global coordinates.
%LOCAL_RESULTS		All 3D results are given in local coordinates. This requires 3 dimensions and MAPTO_ELEMENTS specified.

4.13.3. Example

```
*POSITIONRESULTS 1
%DIMENSION 3
%MAPTO_ELEMENTS #1
%MAP_ID
1 1.0 1.0 1.0 12.3 123.1 0.21
1 1.0 2.0 1.0 123.3 23.3 3.21
1 1.0 3.0 1.0 122.3 23.2 3.1
2 3.0 2.0 3.0 112.3 3.2 23.21
2 2.0 2.0 2.0 122.3 3.2 13.21
10 1.0 2.0 1.0 12.3 123.2 3.21
```

4.14. *GLVIEWPOSITIONSCALAR [<ID>]

4.14.1. Description

This block is used for grouping the position result blocks that belong to the same frame and to define the sequence of the result blocks. These results are not supported in GLview Pro.

4.14.2. Directives

%NAME <"Name">		Assign a name to the scalar result.
%DESCRIPTION "Desc"|#ID		Assign a description of the scalar result.
%STEP <StepNo>	Step 1	This directive describes that the following position result blocks belong to step <StepNo>. This directive may appear many times within this block.
%STEPNAME <"Name">	Def "Step n"	Assign a description to the current step. This directive may appear many times within this block.
%STEPTIME <time>		Assign a time stamp to the current step. This directive may appear many times within this block.

4.14.3. Example

```
*GLVIEWPOSITIONSCALAR 1
%NAME "Separate scalars"
%STEP 1
%STEPNAME "Time: 0.0"
1, 3
%STEP 2
%STEPNAME "Time: 1.0"
11, 18
```


4.15. *GLVIEWPOSITIONVECTOR [<ID>]

4.15.1. Description

This block is used for grouping the position result blocks that belong to the same frame and to define the sequence of the result blocks. These results will be available in GLview as separate vector results.

4.15.2. Directives

%NAME <"Name">		Assign a name to the vector result.
%DESCRIPTION "Desc"|#ID		Assign a description of the vector result.
%STEP <StepNo>	Step 1	This directive describes that the following position result blocks belong to step <StepNo>. This directive may appear many times within this block.
%STEPNAME <"Name">	Def "Step n"	Assign a description to the current step. This directive may appear many times within this block.
%STEPTIME <time>		Assign a time stamp to the current step. This directive may appear many times within this block.

4.15.3. Example

```
*GLVIEWPOSITIONVECTOR 1
%NAME "Separate vectors"
%STEP 1
%STEPNAME "Time: 0.0"
1, 3
%STEP 2
%STEPNAME "Time: 1.0"
11, 18
```


4.16. *TRANSFORMATIONRESULT [<ID>]

4.16.1. Description

This block contains a 4*3 transformation matrix for one indexed face set or element block. The transformations are specified as 3 by 4 matrices, one for each step. The transformation matrix is in the following format:

$$M = \begin{bmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \\ 41 & 42 & 43 \end{bmatrix}$$

This is equivalent to the standard 4*4 transformation matrices used with homogeneous coordinates, where the fourth column is taken to be $[0\ 0\ 0\ 1]^T$. The coordinates (x_0, y_0, z_0) are transformed into (x,y,z) as specified in the following equation:

$$[x \quad y \quad z] = [x_0 \quad y_0 \quad z_0 \quad 1] \bullet M$$

4.16.2. Directives

`%IFS_BLOCK_ID #ID`	Def. -1	The IFS block this results belong to.
`%ELEMENT_BLOCK_ID #ID`	Def. -1	The Elements block this results belongs to.

If both `%IFS_BLOCK_ID` and `%ELEMENT_BLOCK_ID` are -1 the tranformation matrix applies to all parts within the step the *TRANSFORMATIONRESULTS block is defined.

4.16.3. Example

```
*TRANSFORMATIONRESULTS 2
%IFS_BLOCK_ID #1
1 0 0
0 1 0
0 0 1
2 1 0
```


4.17. *GLVIEWTRANSFORMATION [<ID>]

4.17.1. Description

This block is used for grouping the transformation result blocks (*TRANSFORMATIONRESULTS) that belongs to the same step and to define the sequence of the transformation result blocks.

4.17.2. Directives

%NAME "Name"		Assign a name to the transformation result.
%DESCRIPTION "Desc"		Assign a description of the scalar result.
%STEP <StepNo>	Step 1	This directive describes that the following result blocks belong to step <StepNo>. This directive may appear many times within this block.
%STEPNAME <"Name">	Def "Step n"	Assign a description to the current step. This directive may appear many times within this block.
%STEPTIME <time>		Assign a time stamp to the current step. This directive may appear many times within this block.
%RESULT_ID <ID>	Block ID	The ID of the result. If –1 or not present, the block ID is used.

4.17.3. Example

```
*GLVIEWTRANSFORMATION 1
%STEP 1
%STEPNAME "Time: 0.0"
2,3
%STEP 2
%STEPNAME "Time: 1.0"
12, 13
```


4.18. *CROSSECTIONS [<ID>]

4.18.1. Description

This block contains a collection of cross sectional data.

4.18.2. Directives

%TYPE <type>		Type of cross section, e.g. I, pipe etc.

Defined types:

TYPE	Parameters
IORH	hz : Height bt: Width of top flange tt: Thickness of top flange ty: Thickness of web bb: Width of bottom flange tb: Thickness of bottom flange
PIPE	dy : Outer diameter t: Thickness of wall
CYLINDER	dy : Outer diameter join: If 1, the beams should be joined if possible, else the beams are drawn independently.
BOX	hz: Height by: Width tt: Thickness of top flange ty: Thickness of webs tb: Thickness of bottom flange

4.18.3. Example

```
*CROSSECTIONS 1
%TYPE IORH
0.5 0.01 0.01 0.1 0.1 0.01
```

4.19. *DIRECTIONS [<ID>]

4.19.1. Description

This block contains a collection of data that specifies the direction of a cross section in relation to its nodes by a vector. The components of the vector perpendicular to the beam axis at a given node specifies local z-axis or (“up-vector”). The beam axis is the local x-axis, and a right hand coordinate system is used.

4.19.2. Directives

None.

4.19.3. Example

```
*DIRECTIONS 1
0.0 0.0 1.0
```


4.19.4. Restrictions and limitations

Current definition does not support eccentricities nor rotation of cross sections at each end of the beam.

4.20. *GLVIEWSTATEINFO[<ID>]

4.20.1. Description

This block defines meta data for states on the VTF file. For each step a state ID, name, ref. values etc. can be specified.

4.20.2. Directives

%STATE_ID <ID>	Step number	The ID of the State. The following directives applies to this state until a new %STATEID is given
%STEP <StepNo>	-1	The step number this state maps to. Should be -1 for group states
%STATE_NAME <"Name">	Def "State n"	The name of the specified state
%REF_VALUE <value>		A reference value for the state (time, frequency, load case).
%REF_TIME	Default	Reference type is time.
%REF_FREQUENCY		Reference type is frequency.
%REF_LOADCASE		Reference type is load case.
%REF_OTHER		Reference type is other.
%GROUP		This token indicates that the state is a group (has no step connection, but groups other states)
%PARENT <ID>	-1	Defines the parent of the specified state. If not -1, the state must exist.

4.20.3. Example

```
* GLVIEWSTATEINFO 1
%STATE_ID 1000
%STATE_NAME “Loadcase 1”
%GROUP

%STATE 1
%STEP 1
%STATE_NAME “Occurence 1”
%REF_VALUE 1
%REF_TYPE LOADCASE
%PARENT 1000

%STATE 2
%STEP 2
%STATE_NAME “Occurence 2”
%REF_VALUE 2
%REF_TYPE LOADCASE
%PARENT 1000
```


4.21. *GLVIEWDISPLACEMENT[<ID>]

4.21.1. Description

This block is used for grouping the result blocks that belong to the same frame and to define the sequence of the result blocks. These results will be available in GLview as displacements results.

4.21.2. Directives

%NAME <"Name">		Assign a name to the displacement result.
%DESCRIPTION "Desc"		Assign a description of the displacement result.
%STEP <StepNo>	Step 1	This directive describes that the following result blocks belong to step <StepNo>. This directive may appear many times within this block.
%STEPNAME <"Name">	Def "Step n"	Assign a description to the current step. This directive may appear many times within this block.
%STEPTIME <time>		Assign a time stamp to the current step. This directive may appear many times within this block.
%RESULT_ID <ID>	Block ID	The ID of the result. If –1 or not present, the block ID is used.
%RELATIVE		The displacement result are specified as displacements relative to the original nodes
%ABSOLUTE	Default	The displacement result are specified as new global node positions, just like *NODES.

4.21.3. Example

```
*GLVIEWDISPLACEMENT 1
%NAME "Displacements"
%DESCRIPTION "Displacements results"
%STEP 1
%STEPNAME "Time: 0.0"
1, 3
%STEP 2
%STEPNAME "Time: 1.0"
11, 18
```


4.22. *SET [<ID>]

4.22.1. Description

A block defining an element set (or group) in the VTF file. A set can have items from multiple blocks (parts). A VTF file can contain many sets, and the sets can be overlapping (one element can be in more than one set). A set has an ID which must be unique for the VTF file. A set item could be specified by either an ID or a one based index.

4.22.2. Directives

%NAME "Name"		The name of the set
%SET_ID <ID>		The ID of the set
%GEOMETRY_ID <ID>	-1	The ID of the geometry block the set refers to
%MAP_ITEM_IDS	FALSE	Specifies that the set items element IDs.
%MAP_ITEM_INDICES	TRUE	Specifies that the set items are one based element indices.
%TOTAL_NUM_ITEMS		Number of items in the set. Not needed, but will speed up loading
%BLOCK #ID		Specifies that the following IDs or indices refer to the given element block.

4.22.3. Example

```
*SET 1
%NAME "Set 1"
%MAP_ITEM_INDICES
# Define an element set with the elements with indices 1,10 and 1234 from
# element block with ID 101 and index 23 from element block 102
%BLOCK #101
1
10
1234
%BLOCK #102
23
```


4.23. *2DPLOTDATA [<ID>]

4.23.1. Description

This block is used to define a number of variables that can be plotted against each other. All variables must have the same number of values, called number of rows. This creates in effect a 2D table or 2D matrix with the variables in the columns and the rows in the rows. Each variable is given a name and optionally a type (TIME, STEP) to indicate the contents of the variable.

4.23.2. Directives

%NAME <"Name">		Assign a name to the displacement result.
%SEPARATOR <TAB|SPACE|char>	SPACE	The token separating the values
%VARIABLE_NAMES		Indicates that the next line(s) of the block specifies the name of the series. Each name should be in double quotes “”, and separated by the given separator or a new-line.
%VARIABLE_TYPES	NONE	Indicates that the next line(s) of the block specifies the type of the variables. The variable types should be separated by the given separator, and it is not necessary to specify a type for all variables. Legal values: TIME | STEP | NONE It is fully optional to specify the variable types.
%DATA		Indicates that the following line(s) contain the plot data. The number of values specified in this section must be a multiple of the number of series defined in the series names. The items should be separated by a the given separator or a new-line

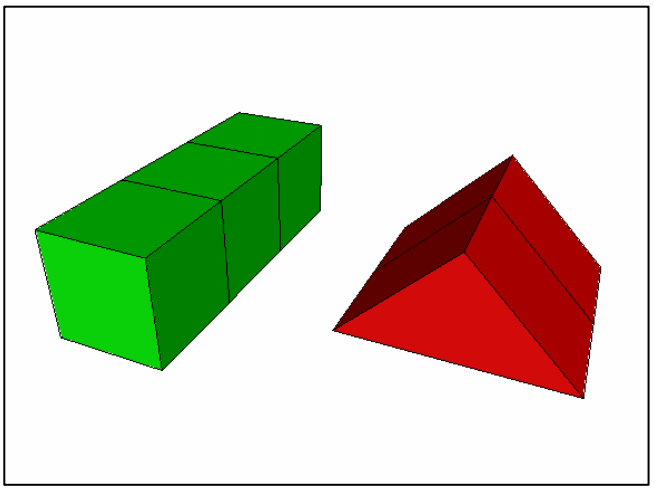
4.23.3. Example

```
*2DPLOTDATA 1
%NAME "Model Plot data"
%SEPARATOR SPACE
%VARIABLE_NAMES
“Time” “Total displacement” “Step” “vonMises” “1st Principal Stress”
%VARIABLE_TYPES
TIME NONE STEP
%DATA
1 0.2 1 10.2 1.23
2 1.3 2 10.2 2.21
2.3 1 4 10.2 3.21
2.5 1.2 5 10.01 1.42
```


5. Examples

5.1. Minimal VTF element example

```
*VTF-1.00

*NODES 3
%WITH_ID
10      0.0 0.0 0.0
20      1.0 0.0 0.0
30      1.0 1.0 0.0
40      0.0 1.0 0.0
50      0.0 0.0 1.0
60      1.0 0.0 1.0
70      1.0 1.0 1.0
80      0.0 1.0 1.0
90      0.0 0.0 2.0
100     1.0 0.0 2.0
110     1.0 1.0 2.0
120     0.0 1.0 2.0
130     0.0 0.0 3.0
140     1.0 0.0 3.0
150     1.0 1.0 3.0
160     0.0 1.0 3.0

*ELEMENTS 1
%NAME "Hex elements"
%DESCRIPTION "The hexahedron elements"
%NODES #3
%WITH_ID
%HEXAHEDRONS
100     10 20 30 40 50 60 70 80
200     50 60 70 80 90 100 110 120
300     90 100 110 120 130 140 150 160

*GLVIEWGEOMETRY 1
%NAME "Sample model"
%ELEMENTS
10,1

*NODES 10
%WITH_ID
1       2 0 0
2       3 1 0
3       4 0 0
4       2 0 1
5       3 1 1
6       4 0 1
7       2 0 2
8       3 1 2
9       4 0 2

*ELEMENTS 10
%NODES #10
%PENTAHEDRONS
1 2 3 4 5 6
4 5 6 7 8 9
```



5.2. Example - beam definition

```
*ELEMENTS 1
%NAME "Beams supporting top side (type GIORH)"
%NODES #3
%CROSSECTIONS  #1
%DIRECTIONS    #1
%BEAMS
1 2
4 3
5 6
7 8
%CROSSECTIONS  #2
%BEAMS
7 8
8 9

*ELEMENTS 2
%NAME "Bay work northern end (type GIORH)"
%NODES #3
%CROSSECTIONS  #1
%DIRECTIONS    #2
%BEAMS
1 2
4 3
5 6
7 8

*ELEMENTS 3
%NAME "Trussed beam (pipes)"
%NODES #2
%CROSSECTIONS  #2
%DIRECTIONS    #1
%BEAMS
1 2
4 3
5 6
7 8

*CROSSECTIONS 1
%TYPE IORH
0.5 0.01 0.1 0.01 0.1 0.01

*CROSSECTIONS 2
%TYPE PIPE
0.15 0.01

*DIRECTIONS 1
0.0 0.0 1.0

*DIRECTIONS 2
1.0 0.0 0.0
```

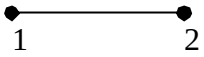
NOTE: Directional data not necessary for pipes. Cross sections not completely defined. Above definitions indicate I-beam by height, flange widths, and thickness data; pipe by outer diameter and thickness.

6. Supported element types

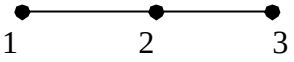
6.1. Points

*POINTS

6.2. Beams

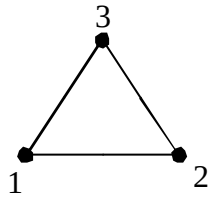


*BEAMS

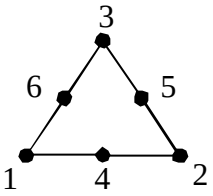


*BEAMS_3

6.3. Triangles

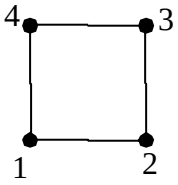


*TRIANGLES

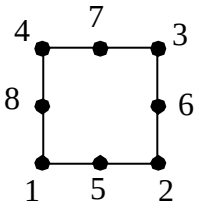


*TRIANGLES_6

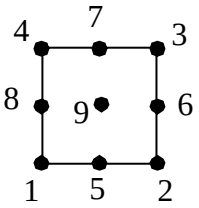
6.4. Quads



*QUADS

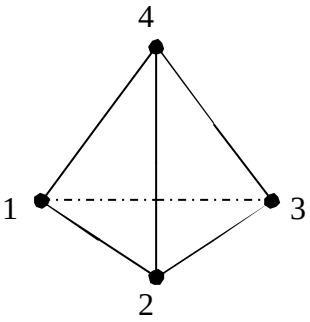


*QUADS_8

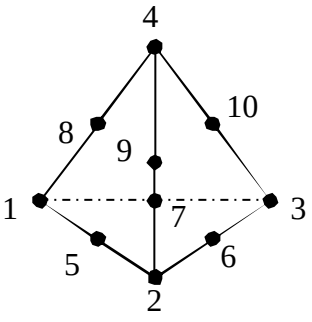


*QUADS_9

6.5. Tetrahedrons

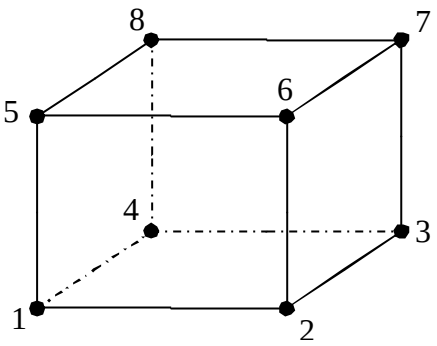


*TETRAHEDRONS

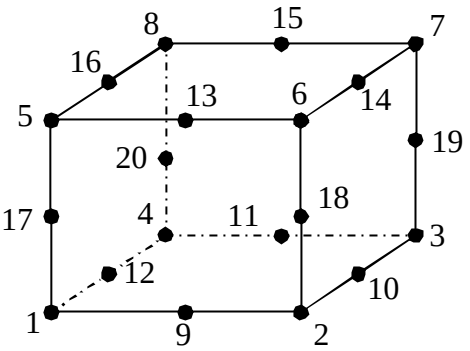


*TETRAHEDRONS_10

6.6. Hexahedrons

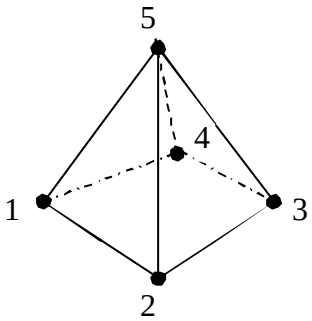


*HEXAHEDRONS

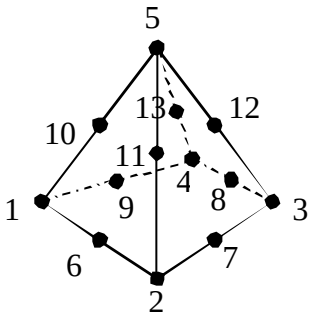


*HEXAHEDRONS_20

6.7. Pyramid

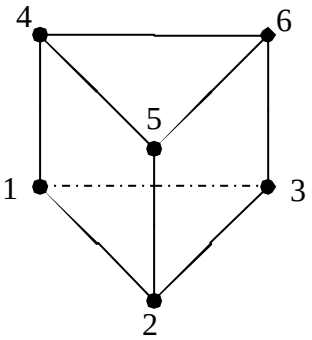


*PYRAMIDS

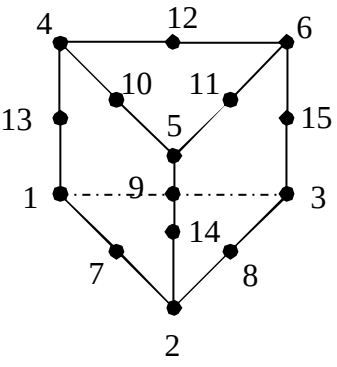


*PYRAMIDS_13

6.8. Pentahedrons



*PENTAHEDRONS



*PENTAHEDRONS_15

