

Chapter 9

Data Interface with External Software

Chapter 9 *Data Interface with External Software*

Chapter 9 Data Interface with External Software

In most cases, VisualFEA's main data files are handled through its own user interface. Thus, their internal contents are hidden to the end users. In some cases, however, the contents and structures of VisualFEA file are desired to be known for interface between VisualFEA and other software.

As already described in Chapter 6, you may use VisualFEA together with other solvers provided by a third party, or developed by yourself. The data within the VisualFEA file should be delivered directly, or by way of intermediate data interface programs, to external solvers. In other words, it is necessary for the programs to read the data within the VisualFEA file. If it is the case, the file contents and structures of VisualFEA as well as of the interfacing software should be known, as detailed in the following sections.

If the analysis data generated by other software are to be postprocessed by VisualFEA, they should be written in the form readable by VisualFEA. In this case, the software should supply a data file with full ingredient and structure of a VisualFEA data file. The software may customize VisualFEA's postprocessing items by recording appropriate information on the data file.

In postprocessing stage, the displayed data items are selected using popup menus within contouring or vector dialog box. There are default popup menus set by VisualFEA. However, the items of the popup menus can be customized by the external software, if necessary. The popup menu related data including menu strings are defined by the external solver and recorded in the VisualFEA file.

VisualFEA files are written using Macintosh or Windows API functions related with file opening, reading and writing. Their contents are not accessible in text form. Computer program codes reading or writing VisualFEA files should be written using Macintosh or Windows API's file utility functions and strictly following the file structure and format described in this chapter.

File position

The file consists of a number of data items. The beginning of each item is marked by a corresponding file position which indicate the relative location of the data within the file. Both Macintosh and Windows API have functions setting and getting these file positions. Thus, the data need not be read sequentially. Instead, data items can be located, in any order, using the file position record, and read the item starting from the position.

There are two file position records: modeling data position and analysis data position. The file position of the modeling data items are recorded in the modeling data position record, and those for the analysis result items are recorded in the analysis data position record.

The modeling data positions are obtained and recorded while the VisualFEA file is being saved at the end of modeling. The analysis data file positions are picked and written together with analysis results during processing stage, whichever one of VisualFEA's own solver or an external solver is used for processing. If the analysis results computed by external solvers are going to be visualized by VisualFEA, it is the responsibility of the solver to keep track of the file positions and record them together with the analysis data.

Input data for external solver

If a finite element model created by VisualFEA is to be analyzed by an external solver, the modeling data should be passed to the solver as input data. The following items are the data necessary for finite element processing.

- Node data including nodal coordinates
- Element data including element node connectivities
- Structural boundary conditions for structural analysis
- Element properties such as material properties or section properties
- Temperature distribution, its gradient for structural analysis
- State of frame member joints for frame analysis
- Equivalent nodal forces evaluated from load conditions
- Equivalent nodal heats evaluated from heat conditions
- Number of integration points applied for each element shape and order
- Options to be applied for processing

Before reading the above items, the external solver should read the following data items:

- Modeling data master record
- Modeling data position record

The modeling data master record consists of information on the property of the

file, and the modeling data position record has the relative positions of the above items within the VisualFEA file.

Output from external solver

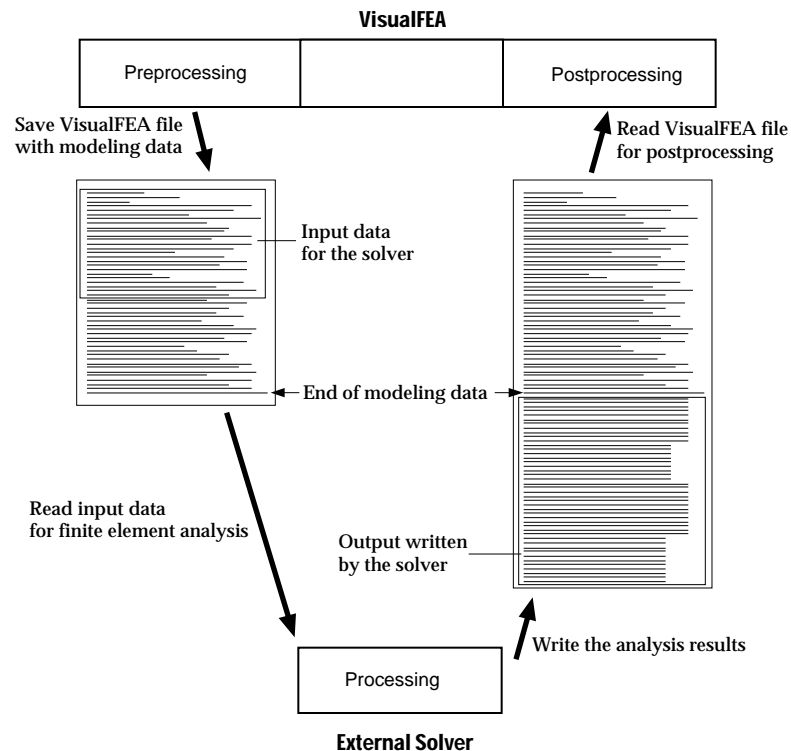
Once the modeling data of a VisualFEA file are read and processed by an external solver, the resulting data can be postprocessed by VisualFEA again. In order to make the output data readable from VisualFEA, they should be written over the original VisualFEA file. In other words, the output data should be appended to the end of the modeling data in the VisualFEA file. The output data items include:

- Data defining custom menu items for contouring
- Data defining custom menu items for vector display
- Data items for graphical visualization

The output part also has the master variables and the file positions, namely:

- Analysis data master variables
- Analysis data file positions

The parts of the VisualFEA file other than above output items should be kept intact by the external solver. Otherwise, Visualfea can neither read nor process the file properly.



<Data flow for combined use of VisualFEA and an external solver>

Other data for user interface, graphical modeling and rendering

VisualFEA file contains not only the data necessary for finite element analysis but also other data related to user interface, graphical modeling, rendering and etc. These data are not directly used in finite element processing, but are essential to maintaining and managing the source of data in graphically interactive work environment. These data can be classified into 3 types: geometric data, source of assignment data, and input and view control data.

The geometric data are graphical data for hierarchical construction of finite element model, and consist of

- Coordinates of curve end points
- Curve division ratio
- Linked curve data
- Primitive surface data
- Surface mesh data
- Volume mesh data

The source of assignment data implies the native data used in interactive work environment. The actual data used in processing are extracted from this source.

- Load condition data
- Heat boundary condition data

The load condition data are used for interactive assignment of load conditions, and are the source of equivalent nodal force data which is actually used infinite element analysis. Likewise, heat boundary condition data are the source of equivalent nodal heat data. The input and view control data are related to viewing transformation, rendering and input environment of graphical models.

- View transformation
- Construction planes
- Symbol variables

There are also the kind of data which serve for the source of interactive graphical modeling as well as for direct use in finite element processing:

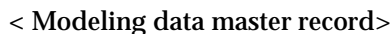
- Node data
- Boundary condition data
- Element property data
- Temperature data
- Member joint data

The data described in this section are also important for interface with other CAD or graphical modeling software. Data may be imported from such software for further processing by VisualFEA. Or, VisualFEA data may be exported for use in other software.

VisualFEA files consist of many data items as described in the previous sections. The structure and the contents of each data item are described in more detail in this section. The contents of VisualFEA files consist of two parts: modeling data and analysis data. Data items of the two parts are indicated in the figure at the beginning of this chapter. The modeling data items are generated at the stage of preprocessing. The analysis data items are obtained as a result of finite element processing, either by VisualFEA or by an external solver.

Modeling data are generated at the preprocessing stage through interactive user operations, and written on VisualFEA file when the file is being saved. The data are written in the file, item by item, in accordance with the sequence described here. But, the position of each item can be located by the modeling data position record, and thus can be read in arbitrary order.

Modeling data master record consists of file identifications and information necessary for reading the file. This record is chiefly used for compatibility between different platform, and between different VisualFEA versions. There are 30 entries of 4 byte long integers in the record as shown in the following figure. The blank spaces are reserved for future use.



- OS flag : The value of this flag is used to identify the platform under which the file was created. If the current and the file's platforms are identical, this

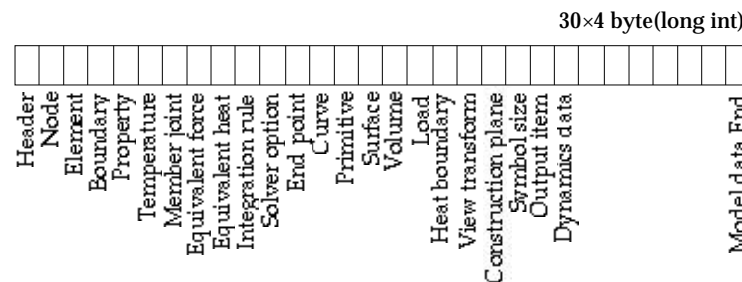
value should read '12345678'. Otherwise, the value gets a different number.

Binary data are represented differently under MacOS and Windows platforms. VisualFEA reads files created under different platform with automatic data conversion.

- **VisualFEA version No. :** This is to identify the version of VisualFEA which created the file.
- **File I.D. :** Each file is given an I.D. which is a random number generated at the time when the file is saved.
- **Number of variable node data items :** A part of node data has record length which may be varied from version to version. Thus, the information on the length is necessary for compatibility between VisualFEA versions.
- **Number of variable element data items :** A part of element data has record length which may be varied from version to version. Thus, the information on the length is necessary for compatibility between VisualFEA versions. The length is represented by the number of items.
- **Number of variable curve data items :** The length of variable part of the element data. Similar to the above item.
- **Number of variable surface data items :** The length of variable part of the curve data. Similar to the above item.
- **Number of variable volume data items :** The length of variable part of the surface data. Similar to the above item.
- **Number of variable volume data items :** The length of variable part of the volume data. Similar to the above item.

- Modeling data position record

Modeling data position record has the information on the offset distance, in byte, from the beginning of the file to the starting point of each data item. The record has 30 entities of 4 byte long integers. The blank spaces are reserved for future use.



< Modeling data position record>

- Header: The position of header record.
- Node: The position of node data.

- Element: The position of element data.
- Boundary: The position of structural boundary condition data.
- Property: The position of element property data.
- Temperature: The position of temperature data.
- Member joint: The position of frame member joint data.
- Equivalent force : The position of equivalent nodal force data.
- Equivalent heat: The position of equivalent nodal heat data.
- Integration rule : The position of integration rule record.
- Solver option : The position of solver option record.
- End point: The position of curve end point data.
- Curve: The position of curve data.
- Primitive: The position of surface primitive data.
- Surface: The position of surface mesh data.
- Volume: The position of volume mesh data.
- Load: The position of load condition data.
- Heat boundary: The position of heat boundary condition data.
- View transform: The position of view transformation data.
- Construction plane: The position of construction plane data.
- Symbol size: The position of Symbol size record.
- Output item: The position of output item record.
- Dynamics data: The position of Dynamics data record.
- Modeling data end: The position of the end point of the modeling data.

■ Header record

Header record has the basic information on the file including the analysis type, sizes of model ingredients, etc. The record has 30 entities of 4 byte long integer. The blank spaces are reserved for future use.

30×4 byte(long int)																													
Analysis subject	Flag of coupled analysis	Number of nodes	Number of active elements	Number of inactive elements	Number of boundary condition sets	Number of element property sets	Number of load condition sets	Number of equivalent nodal loads	Number of member joint sets	Number of heat boundary condition sets	Number of heat convection data	Number of nodal dynamics data sets	Number of load combination sets	Number of curve end points	Number of curves	Number of surface primitives	Number of surface meshes	Number of volume meshes	Number of mesh lines	Number of construction planes	Number of embedded bars	Flag of material nonlinear analysis	Flag of geometric nonlinear analysis	Flag of dynamic analysis	Flag of transient heat analysis	Flag of transient seepage analysis	Flag of sequential analysis		

< Header record >

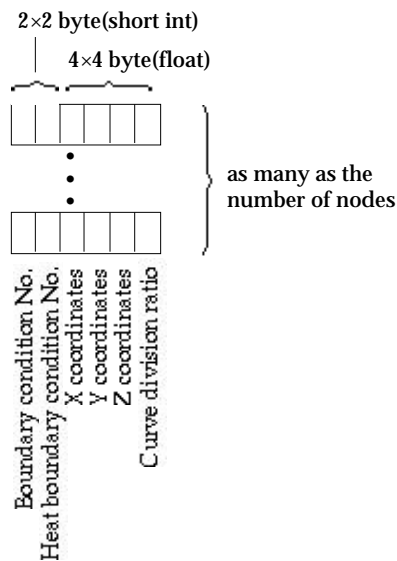
- Analysis subject: The analysis subject is coded by the following numbers. The analysis subject represents the major analysis type which can be mixed with different types of analysis class. (Refer to Chapter 5, "Element Properties".)
 - 0 : plane stress
 - 1 : plane strain
 - 2 : Axisymmetric
 - 3 : plate bending structure
 - 4 : shell structure
 - 5 : 3 dimensional solid structure
 - 6 : 2 dimensional truss
 - 7 : 3 dimensional truss
 - 8 : 2 dimensional rigid frame
 - 9 : 3 dimensional rigid frame
 - 10 : plane heat conduction
 - 11 : axisymmetric heat conduction
 - 12 : 3 dimensional heat conduction
 - 13 : plane seepage
 - 14 : axisymmetric seepage
 - 15 : 3 dimensional seepage
- Number of nodes: The total number of nodes in the model
- Number of active elements: The number of elements used in the finite element analysis. Only elements assigned with element properties are actually included in the analysis.
- Number of inactive elements: The number of elements without element property assignment. The unassigned elements are not included in the analysis, and thus termed as "inactive element."
- Number of boundary condition sets: The number of boundary condition sets which include both assigned and unassigned ones. Unassigned set implies the ones defined for user interface, but not yet assigned to any object.
- Number of element property sets: The number of element property sets which include both assigned and unassigned ones.
- Number of load condition sets: The number of load condition sets which include both assigned and unassigned ones.
- Number of equivalent nodal loads: The number of equivalent nodal loads is the same as the number of nodes with non-zero equivalent nodal load.
- Number of member joint sets: The number of member joint sets which include both assigned and unassigned ones.
- Number of heat boundary condition sets: The number of heat boundary condition sets which include both assigned and unassigned ones.
- Number of heat convection data: The number of heat convection data which

are extracted from heat boundary condition data.

- Number of temperature sets: The number of temperature sets which include both assigned and unassigned ones.
- Number of nodal dynamics data sets: The number of nodal dynamics data sets which include both assigned and unassigned ones.
- Number of curve end points: The total number of end points which define the starting and ending points of curves.
- Number of curves: The total number of curves, both created and generated.
- Number of surface primitives: The total number of surface primitives.
- Number of surface meshes: The total number of surface meshes, both created and generated.
- Number of volume meshes: The total number of volume meshes, both created and generated.
- Number of mesh lines: The total number of straight lines consisting the wireframe meshes.
- Number of construction planes: The number of construction planes defined for aiding the coordinate data.
- Flag of nonlinear analysis: The value indicates whether the analysis is linear or nonlinear: 0 for linear analysis and 1 for nonlinear analysis.
- Flag of dynamic analysis: The value indicates whether the analysis is static or dynamic: 0 for static analysis and 1 for dynamic analysis.
- Number of frontal buffers: The number of data buffers used in frontal solution process.

■ Node data

Node data consist of records with nodal coordinates and assignment information, One record contains following information for a node. There are as many records as the number of nodes.



<Node data>

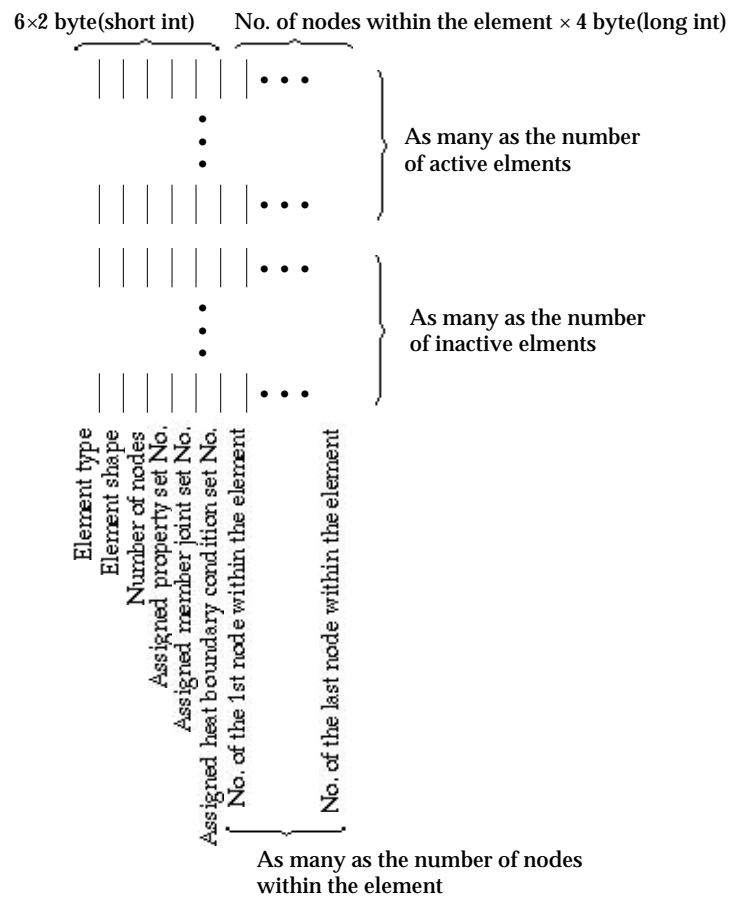
- **Boundary condition No.:** The No. of the structural boundary condition set assigned to the node. This value is zero based. (*Boundary condition sets are numbered starting from 0.*) If this value is -1, the node is not assigned with boundary condition. (2 byte short integer)
- **Heat boundary condition No.:** The No. of the heat boundary condition set assigned to the node. This value is zero based. If this value is -1, the node is not assigned with heat boundary condition. (2 byte short integer)
- **X, Y, and Z coordinates:** The coordinates of the node in X, Y, Z Cartesian coordinate system. (4 byte float for each one of X, Y and Z coordinates)
- **Curve division ratio:** The relative position of the node on the curve. The value is between 0 and 1.0; the node at the starting point of the curve has the value of 0, and the one at the end point of the curve has the value of 1.0. The nodes between the two have the value representing the normalized distance from the starting point.
This value has meaning only for the nodes created by dividing curves. This value is not used for solver. This information is used for future modification of the curve. The curve division ratios are retained after modification of curves. (4 byte float)

■ Element data

Element data consist of records with nodal connectivities and assignment information. One record contains following information for an element. There are records for active elements followed by the records for inactive elements. Active elements are those actually involved in finite element processing. All the elements

assigned with element properties are active. Inactive elements are not used in the processing, but constitute curve divisions, surface meshes or volume meshes. *VisualFEA* represents the segment of curve divisions, surface meshes and volume meshes in the form of element data. These segments become active elements when they are assigned with element properties.

- Element type: Analysis type of the element. Only active elements have element type. (2 byte short integer)
 - 1 : not an element type (for inactive elements)
 - 0 : plane stress
 - 1 : plane strain
 - 2 : Axisymmetric
 - 3 : plate bending structure
 - 4 : shell structure
 - 5 : 3 dimensional solid structure
 - 6 : 2 dimensional truss
 - 7 : 3 dimensional truss
 - 8 : 2 dimensional rigid frame
 - 9 : 3 dimensional rigid frame
 - 10 : plane heat conduction
 - 11 : axisymmetric heat conduction
 - 12 : 3 dimensional heat conduction
 - 16 : interface element
 - 17 : embedded bar
- Element shape: The shape of the element (2 byte short integer)
 - 1 : 2 node line element
 - 2 : 3 node line element
 - 3 : 3 node triangular element
 - 4 : 6 node triangular element
 - 5 : 4 node quadrilateral element
 - 6 : 8 node quadrilateral element
 - 7 : 4 node tetrahedral element
 - 8 : 10 node tetrahedral element
 - 9 : 6 node prism element
 - 10 : 15 node prism element
 - 11 : 8 node hexahedral element
 - 12 : 20 node hexahedral element

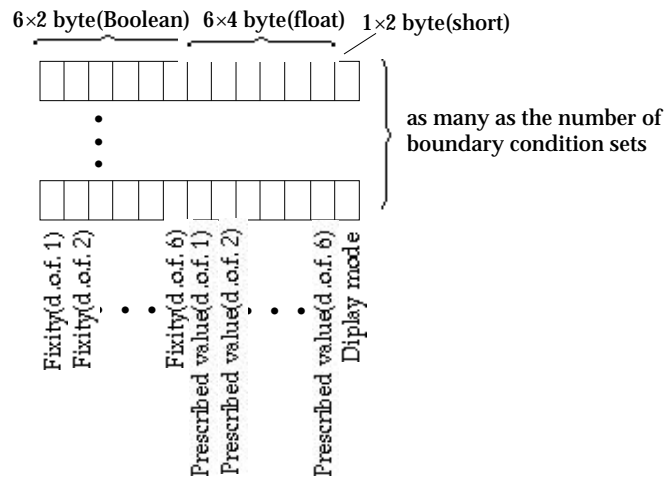


<Element data>

- **Number of nodes:** The number of nodes in the element. (2 byte short integer)
- **Assigned property set No.:** The zero based No. of the element property set assigned to the element. If this value is -1, the element is not assigned with element property set. (2 byte short integer)
- **Assigned member joint set No.:** The zero based No. of the member joint set assigned to the element. If this value is -1, the element is not assigned with member joint set. (2 byte short integer)
- **Assigned heat boundary condition set No.:** The zero based No. of the heat boundary condition set assigned to the element. If this value is -1, the element is not assigned with heat boundary condition set. (2 byte short integer)
- **Node No.:** The zero based No. of each node within the element. (number of nodes within the element x 4 byte long integer)

■ Boundary condition data

The boundary condition data consist of records with structural fixities and either initial displacements or spring constants of the node. One record contains information for one boundary condition set as follows.

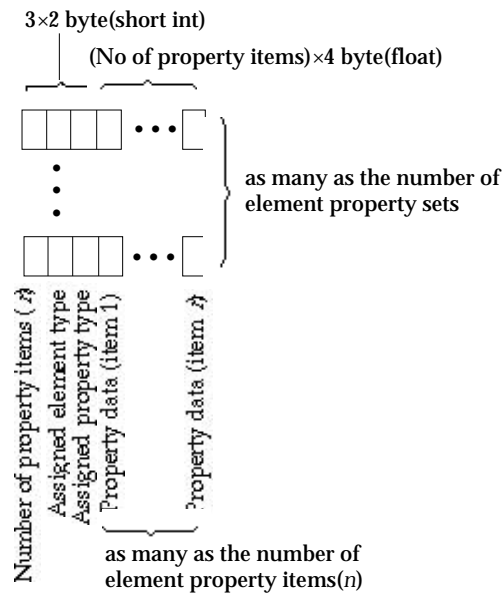


<Boundary condition data>

- **Fixity:** Structural fixity or restraint of each nodal d.o.f. at a node. There are 6 entries of fixity data, but only the entries corresponding to the actual nodal d.o.f. are in use. (6×2 byte short integer)
 - 0 : fixed
 - 1 : free
- **Prescribed value:** Value prescribed for each nodal d.o.f. at a node. There are 6 entries of prescribed values, but only the entries corresponding to the actual nodal d.o.f. are in use. The prescribed values are either initial displacements or spring constants. If the nodal d.o.f. is fixed, the prescribed value represent the initial displacement of the corresponding d.o.f. Otherwise, the value represents the spring constant assigned to the corresponding d.o.f. (6×4 byte float)
- **Display mode:** mode of displaying the prescribed values on “Struct Boundary” dialog. (Refer to Chapter 5) The edit text items of initial displacement, spring constant or both are displayed on the dialog depending on the mode.
 - 0 : Only edit text boxes of initial displacements are displayed.
 - 1 : Only edit text boxes of spring constants are displayed.
 - 2 : All edit text boxes are displayed.

■ Element property data

The element property data have a variable number of data items. The number of data items and the entity of each item are determined by the element type defined for the set, and appear on “Property” dialog.



<Element property data>

- **Number of property items:** The number of element property items are variable, and retrieved by this entry. This value is originally determined by the element type at the stage of preprocessing, but saved and retrieved as an entry of element property data for quick retrieval of the data items. (2 byte short integer)
- **Assigned element type:** The element type is primarily determined by the analysis type. However, VisualFEA allows mixed use of different element types in a model. (2 byte short integer)
 - 1 : not an element type (for inactive elements)
 - 0 : plane stress
 - 1 : plane strain
 - 2 : Axisymmetric
 - 3 : plate bending structure
 - 4 : shell structure
 - 5 : 3 dimensional solid structure
 - 6 : 2 dimensional truss
 - 7 : 3 dimensional truss
 - 8 : 2 dimensional rigid frame
 - 9 : 3 dimensional rigid frame

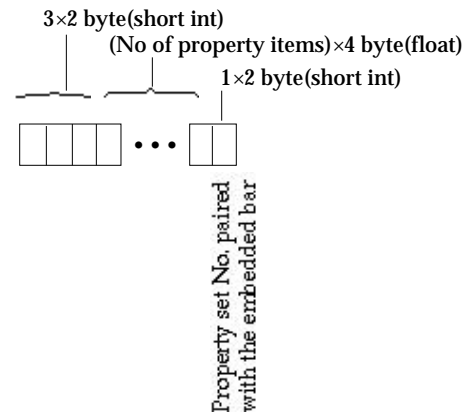
- 10 : plane heat conduction
- 11 : axisymmetric heat conduction
- 12 : 3 dimensional heat conduction
- 13 : plane seepage
- 14: axisymmetric seepage
- 15 : 3 dimensional seepage
- 16 : interface element
- 17 : slip bar
- 18 : embedded bar
- Assigned property type: The type of material property (2 byte short integer)
 - 0 : linear elastic and isotropic
 - 1 : linear elastic and orthotropic
 - 2 : elasto-plastic(appears only for nonlinear analysis)
 - 3 : friction(appears only for interface elements of nonlinear analysis)
- Property data items: The number of data items is retrieved as an entry of a data record, and the entity of each item is determined by the element type as shown in the following table. (number of data items \times 4 byte float)

<Entities of element property items>

Element type	Element property items								
	1	2	3	4	5	6	7	8	9
Plane stress	E_x	E_y	ν_{xy}	t	w_o	α			
Plane strain	E_x	E_y	ν_{xy}	w_o	α				
Axisymmetric	E_x	E_y	ν_{xy}	w_o	α				
Plate	E_x	E_y	ν_{xy}	t	w_o	α			
Shell	E_x	E_y	ν_{xy}	t	w_o	α			
Solid	E_x	E_y	E_z	ν_{xy}	ν_{yz}	ν_{zx}	w_o	α	
Interface element	E_L	E_g	gap	t	*				
Embedded bar	E_{bar}	E_{bond}	A						
Truss 2D & 3D	E	A	w_o	α					
Frame 2D	E	A	I	w_o	α				
Frame 3D	E	ν	A	I_y	I_z	J	w_o	α	β
Plane heat	k_x	k_y							
Axisymmetric heat	k_x	k_y							
Volume heat	k_x	k_y	k_z						

* for plane stress only

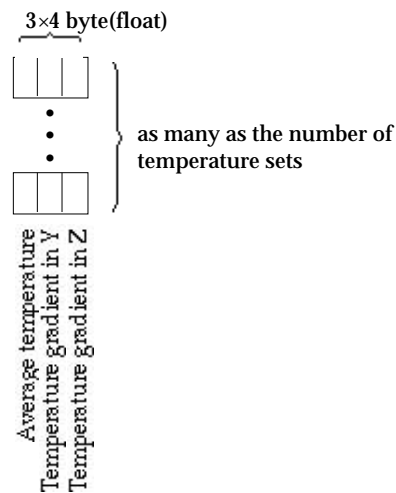
In case the element type is embedded bar, one more data item is added as shown below. The additional item points to the set No. of the property surrounding the embedded bar.



<Element property data for embedded bar>

■ Temperature data

Temperature data are currently applicable only to 2-D and 3-D frame analysis. The data consist of the average temperature of a frame member and the temperature gradient across the member section.



<Temperature data>

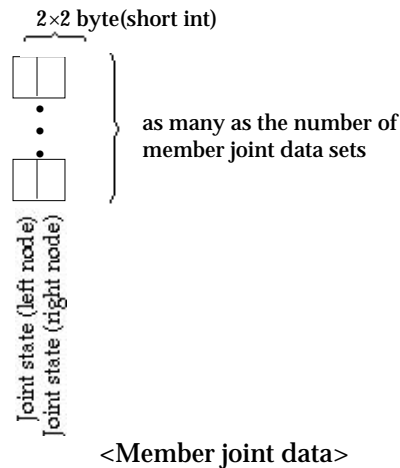
- Average temperature: Average temperature of the frame member(4 byte float)
- Temperature gradient in Y: Temperature gradient across the member section in Y direction. The gradient is equivalent to the temperature difference between top and bottom of the section divided by the height of the section. (4 byte float)
- Temperature gradient in Z: Temperature gradient across the member section in Z direction. This item is used only for 3-D frames. (4 byte float)

Temperature data are not supported by user interface in the current version of VisualFEA. Instead, they are treated as thermal load conditions. Refer to Chapter 5. However, their space is left in VisualFEA file for use by external software.

■ Member joint data

Member joint data are currently applicable to 2-D and 3-D frame analysis. One record of member joint data has only 2 data items: joint states at both end of the member.

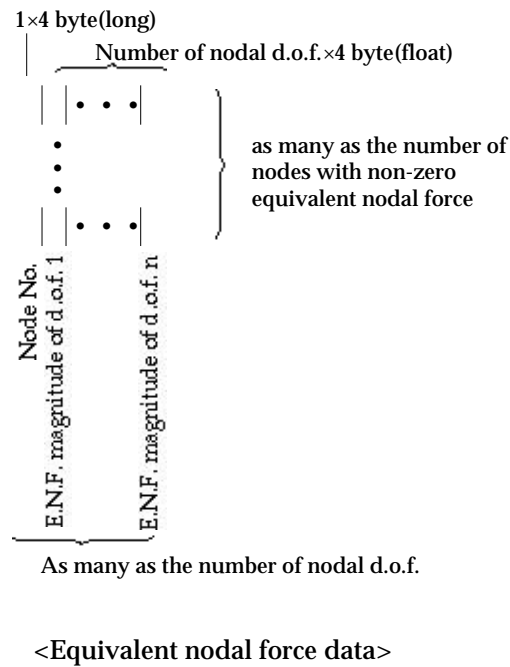
- Joint state (left node): The state of joint at the starting point of the member. (2 byte short int)
 0 : Rigid joint
 1 : Pin joint
- Joint state (right node): The state of joint at the ending point of the member. (2 byte short int)



■ Equivalent nodal force data

The equivalent nodal forces are computed at each node from the load condition data, which is described in one of the later sections. The equivalent nodal forces are directly involved in finite element analysis, while the load condition data are maintained chiefly for graphical user interface. The data has records only for nodes with non-zero equivalent nodal forces. The components of the equivalent nodal forces are as many as the number of nodal d.o.f.

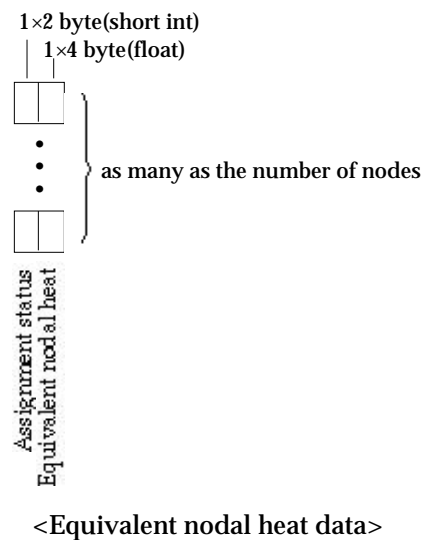
- Node No.: The node No. identifying the node for which the equivalent nodal forces are evaluated. (4 byte long int)
- Equivalent nodal force magnitude: Magnitude of each force component which matches with each one of the nodal d.o.f. (number of nodal d.o.f. \times 4 byte float)



■ Equivalent nodal heat data

This data portion consists of two parts: equivalent nodal heat and convection boundary condition data. Both parts are evaluated from heat boundary condition data.

The equivalent nodal heat data has records for all nodes in the model. Each record has only 2 data entries:



- Assignment status: Flag indicating whether the node is assigned with nodal

heat. (2 byte short int)

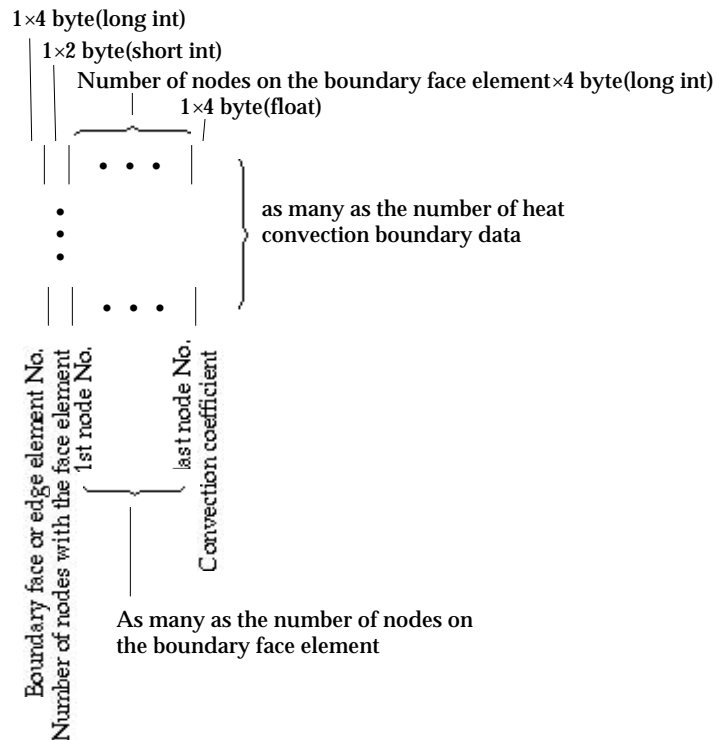
0 : Assigned

1 : Not assigned

- Equivalent nodal heat: Magnitude of nodal heat (4 byte float)

In the second part are the convection boundary condition data, which are evaluated from convection type of heat boundary condition. The data consist of convection coefficients and information on boundary face or edge on which the convection boundary condition is applied.

- Boundary face or edge element No.: The convection boundary conditions are applied to the inactive elements on boundary edges (2-D case) or on boundary faces (3-D case). This data entry has the element No. on the boundary edge or face. (4 byte long int)
- Number of nodes within the face or edge element: The face or edge consists of a number of nodes. This entry has the number of nodes within the element. (4 byte long int)
- Node No.: All the node No. consisting the boundary face or edge element. (number of nodes within the boundary element \times 4 byte long int)
- Convection coefficient: The convection coefficient of the heat boundary condition set from which the equivalent nodal heat is evaluated. (4 byte float)



<Convection boundary data>

■ Nodal dynamics data

The number of nodal dynamics data sets is included in the data header record, and is always zero for static analysis. The following represents only one set of nodal dynamics data records. These data records occur as many times as the number of nodal dynamics data sets. Each set of data records starts with an attribute classifying the type of nodal dynamics data.

- Type of nodal dynamics data: This data entry represent the type of nodal dynamics data as follows:
 - 1 : Nodal dynamic motion - displacement
 - 2 : Nodal dynamic motion - velocity
 - 3 : Nodal dynamic motion - acceleration
 - 4 : Nodal dashpot constant
 - 5 : Nodal mass

The contents for the rest of the nodal dynamics data records are different depending on the type of nodal dynamics data. In case of nodal dashpot and nodal mass, the data consist of 6 entries which represent the values for each nodal d.o.f.

1×2 byte(short int)



Type of nodal dynamics data

<Attribute classifying the type of nodal dynamics data>

■ Integration scheme record

The integration scheme is set independently for each shape and each order of element. The number of integration points set for various element shapes are saved in the integration scheme data record as shown in the following figure. There are 30 entries of 4 byte long integer in the record. The blank spaces are reserve for future use.

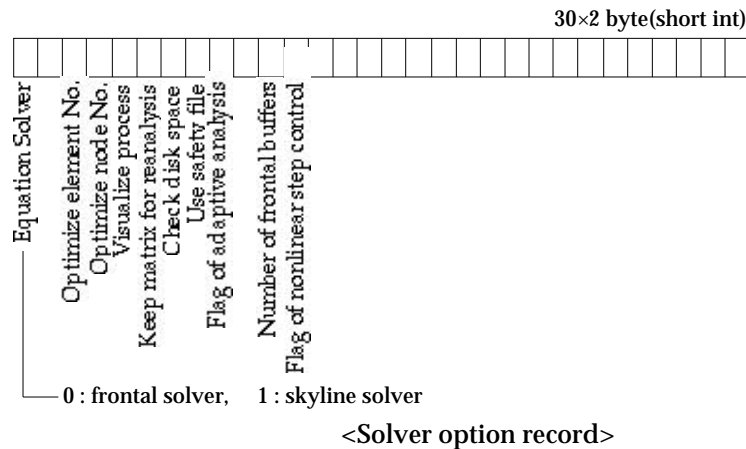
[illegible]

For 3 node triangle element
For 6 node triangle element
For 4 node quadrangle element
For 8 node quadrangle element
For 4 node tetrahedron element
For 10 node tetrahedron element
For 6 node prism element
For 15 node prism element
For 8 node hexahedron element
For 20 node hexahedron element
For 6 node triangle shell element
For 8 node quadrangle shell element
For 4 node quadrangle shell element
For 3 node triangle shell element

- For 3 node triangle element: Number of integration points set for a 3 node triangle element. It is either of 1, 3 or 7.
- For 6 node triangle element: Number of integration points set for a 6 node triangle element. It is either of 1, 3 or 7.
- For 4 node quadrangle element: Number of integration points set for a 4 node quadrangle element. It is either of 1, 4 or 9.
- For 8 node quadrangle element: Number of integration points set for a 8 node quadrangle element. It is either of 1, 4 or 9.
- For 4 node tetrahedron element: Number of integration points set for a 4 node tetrahedron element. It is either of 1, 4 or 5.
- For 10 node tetrahedron element: Number of integration points set for a 10 node tetrahedron element. It is either of 1, 4 or 5.
- For 6 node prism element: Number of integration points set for a 6 node prism element. It is either of 1, 2 or 6.
- For 15 node prism element: Number of integration points set for a 15 node prism element. It is either of 1, 2 or 6.
- For 8 node hexahedron element: Number of integration points set for a 8 node hexahedron element. It is either of 1, 8 or 27.
- For 20 node hexahedron element: Number of integration points set for a 20 node hexahedron element. It is either of 1, 8 or 27.

There are a number of options which are applied for finite element processing. They are saved in the solver option record with 30 entries of 2 byte short integer. Each entry has value of 0 or 1. The option is on if the value is 1 and off otherwise,

except the equation solver option.

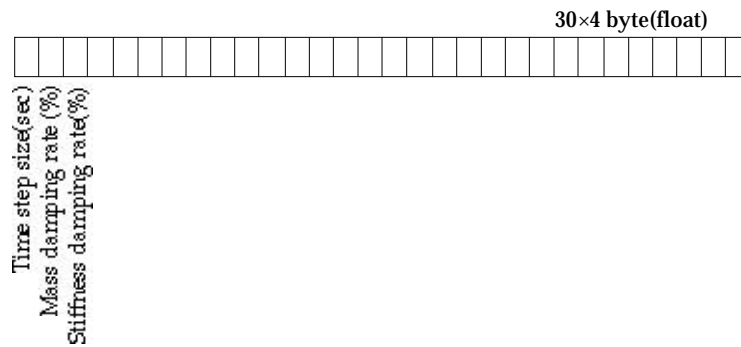


- Equation solver: One of skyline solver and frontal solver is selected as the equation solver.
 - 0 : Frontal solver
 - 1 : Skyline solver
- Optimize element No.: Element number optimization is executed before finite element processing begins, if this option is on.
- Optimize node No.: Node number optimization is executed before finite element processing begins, if this option is on.
- Visualize process: The progress of finite element processing is graphically visualized, if this option is on.
- Keep matrix for reanalysis: The relevant matrix files necessary for reanalysis are not removed at the end of the processing, if this option is on.
- Check disk space: The available disk space is continuously checked while the processing is going on, if this option is on.
- Use safety file: A temporary file is created and used while processing is going on, in order to protect the original file from being damaged under abnormal situation, if this option is on.
- Flag adaptive analysis: An iterative analysis process with adaptive mesh generation is applied if this option is on.
- Number of frontal buffer: This entry specifies the number of matrix data used for frontal solver.
- Flag of nonlinear step control: If this flag is on, the rate load increment is specified individually for each of the load step.

■ Analysis output item record

Each entry has value of 0 or 1. If the value is 1, corresponding item is turned on as an analysis output item. There are 30 entries of 2 byte short integer in the record. The blank spaces are reserved for future use.

- **Inclusion of damping** : This entry indicates whether damping is included in the dynamic analysis.
- **Number of time step** : The number of time steps included in the analysis. This entry is ignored in modal analysis.
- **Number of dynamic modes** : The number of modes computed and used for analysis. This entry is ignored in direct integration.

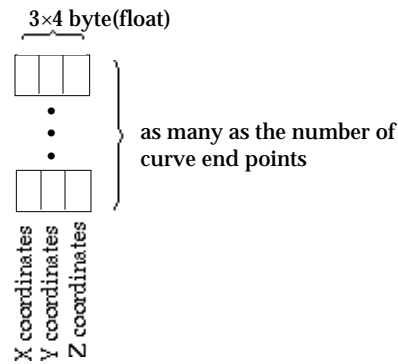


<Dynamic analysis setting record 2>

- **Time step size** : The interval from one time step to the next. Uniform size is assumed for all time steps.
- **Mass damping rate(%)** : The damping ratio applied for mass.
- **Stiffness damping rate(%)** : The damping ratio applied for stiffness.

■ Curve end point data

Every curve has two end points, a starting point and a ending point. The coordinates of these end points are saved as curve end point data.

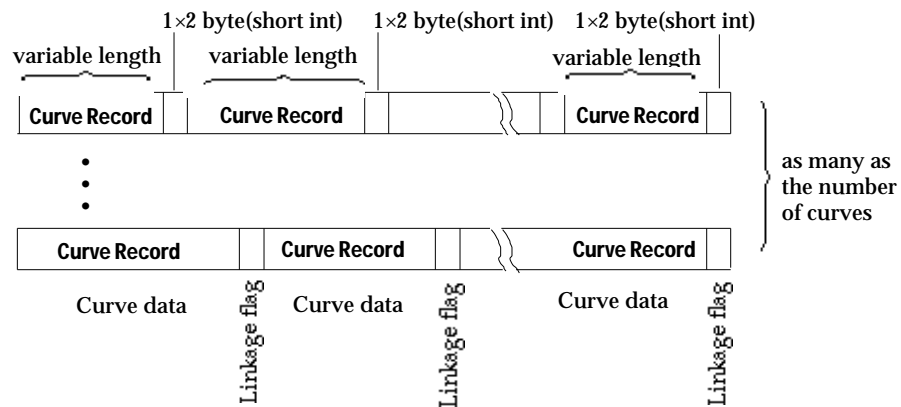


<Curve end point data>

- X, Y, and Z coordinates: The coordinates of the end point in X, Y, Z Cartesian coordinate system. (4 byte float for each one of X, Y and Z coordinates)

■ Curve data

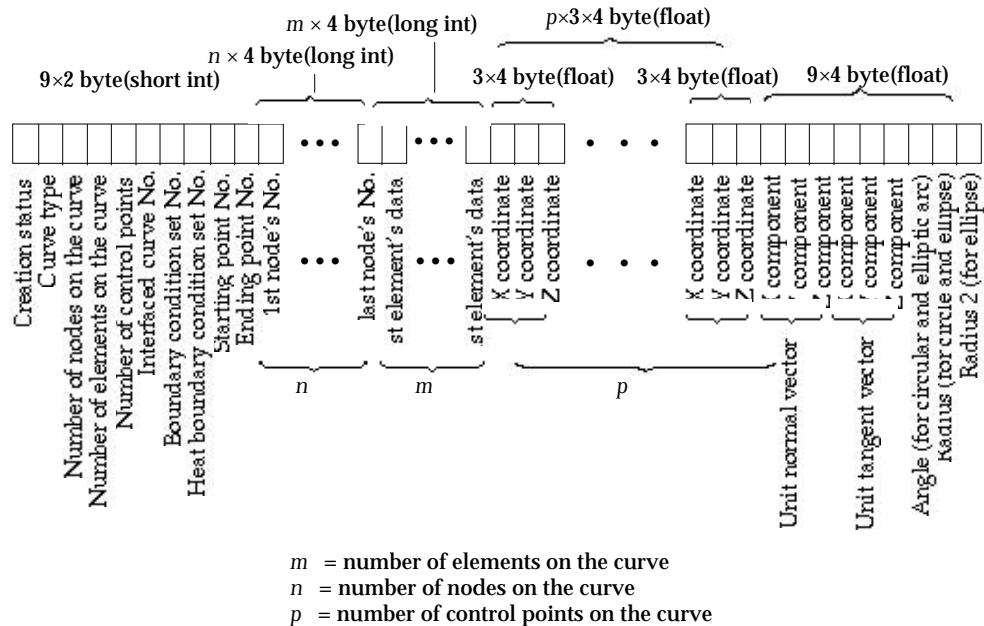
A curve exists either as a single entity or as multiple entities linked together. The curve data are generalized in the form of linked curve as shown in the following figure.



<Linked curve data>

- Curve record: A curve record has information on a single curve as described below. (variable length)
- linkage flag: flag indicating if the curve is linked at the end. (2 byte short int)
 - 0 : No further linkage at the end. The flag is always 0 at the end of linked curves.
 - 1 : curve linked at the end. The flag is always 1 in the middle of linked curves.

A curve record contains information on the geometric composition and the attribute assignment, and other characteristics of a single curve. A curve is composed of end points, control points, nodes and elements.



<Curve record>

- **Creation status:** A curve may be either created by user interaction inputting control points, or generated in the process of finite element mesh generation. The boundary of surface meshes are always defined by curves. The creation status indicates whether the curve was created or generated. (2 byte integer)
 - 0 : Generated.
 - 1 : Created
- **Curve type:** This value represent the type of the curve as follows.
 - 1 : Not a valid curve.
 - 0 : Point
 - 1 : Straight line
 - 10 : Acute arc
 - 11 : Clockwise arc
 - 12 : Counter clockwise arc
 - 13 : Three point arc
 - 14 : Center angle arc
 - 16 : Clockwise circle
 - 17 : Counter clockwise circle

- 18 : Three point circle
- 19 : Center radius circle
- 21 : Quarter ellipse
- 22 : Half ellipse
- 23 : Full ellipse
- 31 : Cubic spline curve
- 32 : B-spline curve
- 33 : Bezier curve
- 34 : Polynomial curve
- 35 : Polyline
- 36 : Segmented curve
- 41 : Rectangle

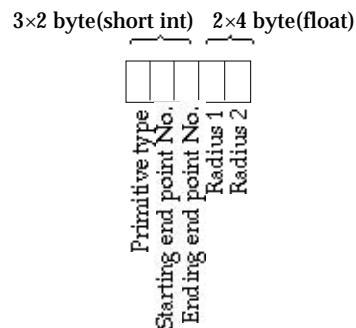
- Number of nodes on the curve: The number of nodes which were created by dividing the curve, or generated together with the curve.
- Number of elements on the curve: The number of elements which were created by dividing the curve, or generated together with the curve.
- Number of control points on the curve: The number of control points which define the geometry of the curve.
- Interface curve No.: When interface elements are assigned along the curve, a duplicate of the curve is generated to form another boundary of the interface gap. In this case, this entry has the pairing interface curve No.
- Boundary condition set No.: If the curve is assigned with a boundary condition, this entry has its No. which is zero based. If boundary condition is not assigned, the value is -1.
- Heat boundary condition set No.: If the curve is assigned with a heat boundary condition, this entry has its No. which is zero based. If heat boundary condition is not assigned, the value is -1.
- Starting point No.: The No. of the end point serving as the starting point of the curve.
- Ending point No.: The No. of the end point serving as the ending point of the curve.
- Node No.: The No. of each node on the curve. This value is zero based. In other words, the node No. starts from 0.
- Element No.: The No. of each element on the curve. This value is zero based. In other words, the element No. starts from 0.
- Coordinates of control points: X, Y and Z coordinates of control points on the curve. Z value is zero for 2 dimensional case.
- Unit normal vector: This entry applies only to circular arc, circle, elliptical arc and ellipse. The directional cosines of the unit normal vector to the curve are saved respectively in X, Y and Z components.
- Unit tangent vector: This entry applies only to closed cubic spline, B-spline or

Bezier curves. The unit tangent vector is tangent to the curve at the starting or the ending point. The directional cosines of this unit tangent vector are saved respectively in X, Y and Z component.

- Angle : This entry applies only to circular arc and elliptical arc. Angle implies the central angle of the curve.
- Radius: This entry applies only to circle and ellipse. This is the radius of a circle, and first radius of an ellipse.
- Radius 2: This entry applies only to ellipse. This is the second radius of an ellipse.

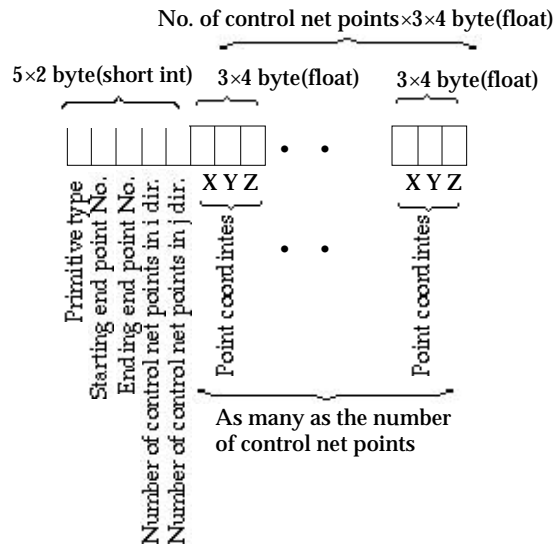
■ Primitive surface data

Primitive surface data consists of primitive records. Of course, there are as many records as the number of primitive surfaces. The records have two different compositions depending on the type of the primitives. The primitive record of sphere, cylinder, and torus is one kind, and that of plane, Lagrangian surface, B-spline surface and Bezier surface is another kind. The primitive record for the first kind is shown in the following figure.



<Primitive record for sphere, cylinder and torus>

- Primitive type: This entry has the code indicating the type of the primitive.
 - 82 : Sphere
 - 83 : Cylinder
 - 84 : Torus
- Starting end point No.: The No. of the end point serving as the starting point of the primitive.
- Ending end point No.: The No. of the end point serving as the ending point of the primitive.
- Radius 1: The first radius of the primitive surface
- Radius 2: The second radius of the primitive surface. Sphere has only one radius, thus this value is ignored in sphere data.



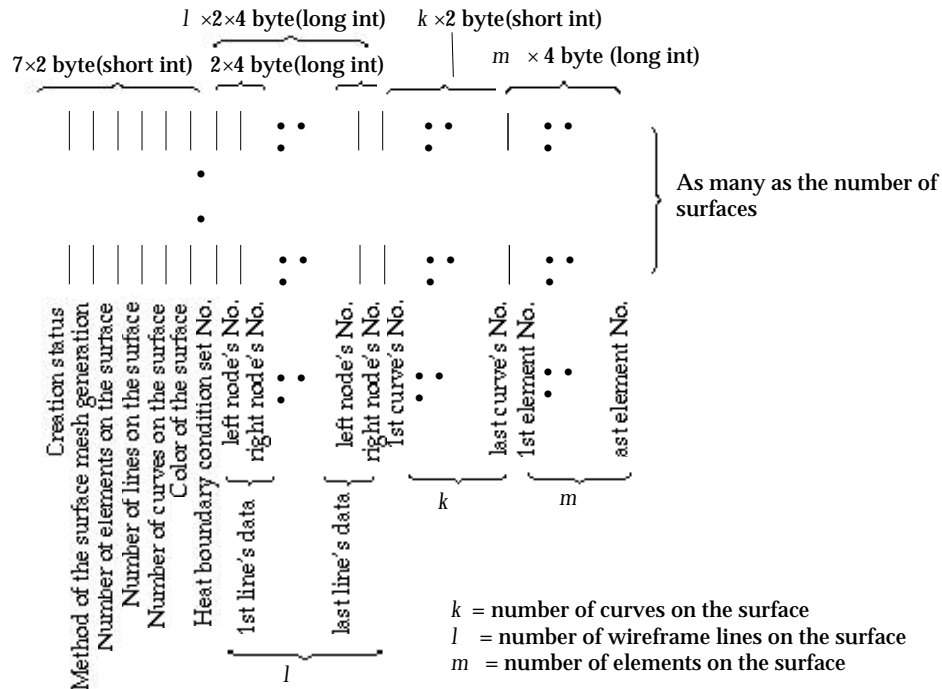
<Primitive record for plane, Lagrangian, B-spline and Bezier surfaces>

- **Primitive type:** The type of the primitive
 - 85 : B-spline surface
 - 86: Bezier surface
 - 87 : Lagrangian surface
 - 90 : Flat plane
- **Starting end point No.:** The No. of the end point serving as the starting point of the primitive.
- **Ending end point No.:** The No. of the end point serving as the ending point of the primitive.
- **Number of control net points in i direction:** The primitive surface is defined by $m \times n$ control net points where m and n represent the number of control points in i and j direction. This entry has the value of m .
- **Number of control net points in j direction:** This entry has the value of n .
- **Control net point coordinates:** The X, Y, and Z coordinates of each of $m \times n$ control net points.

■ Surface mesh data

Surface data have the information on the surface meshes including their compositions, attribute assignments, and other characteristics.

A surface mesh is composed of nodes, elements, wireframe lines and boundary curves. Information on the nodes on the surface can be derived from that of the elements on the surface. And therefore, information on the nodes are not included in the surface data.



<Surface data>

- **Creation status:** A surface mesh may be either created by user interaction using surface mesh generation command, or generated in the process of volume mesh generation. The boundary of volume meshes are always defined by surface meshes.

The creation status indicates whether the surface mesh was created or generated.

0 : Generated.

1 : Created

- **Method of surface mesh generation:** The entry has the code representing the method of surface mesh generation which was applied to create the surface.

0 : Surface mesh generation by 2 edge mapping

1: Surface mesh generation by extrusion

2: Surface mesh generation by 4 edge mapping

3: Surface mesh generation by 3 edge mapping

4: Surface mesh generation by revolution

5: Surface mesh generation by translation

12: Surface mesh generation by automatic triangulation

13: Surface mesh generation by automatic tetrahedronization

14: Volume mesh generation by duplication with "Move"

15: Surface mesh generation by duplication with "Move"

18: Surface mesh generation by duplication with "Revolve"

21: Surface mesh generation by duplication with “Mirror”

24: Surface mesh generation by twisting

28: Surface mesh generation by duplication with “Extrude”

29: Surface mesh generation by projection

- Number of elements on the surface: Number of all elements, either active or inactive, composing the surface.
- Number of lines on the surface: Number of all lines representing the wireframe of the surface mesh.
- Number of curves on the surface: Number of all curves attached to the surface.
- Color of the surface: The color assigned to the surface. (currently not in use, but reserved for future use)
- Heat boundary condition set No.: No. of the heat boundary condition set assigned to the surface.
- Line data: Information on the lines forming the wireframe. A line is represented by two end nodes. Thus, the data for a line consist of 2 entries; the 2 end node's No.
- Curve No.: The No. of each curve in the surface mesh.
- Element No.: The No. of each element in the surface mesh.

■ Volume mesh data

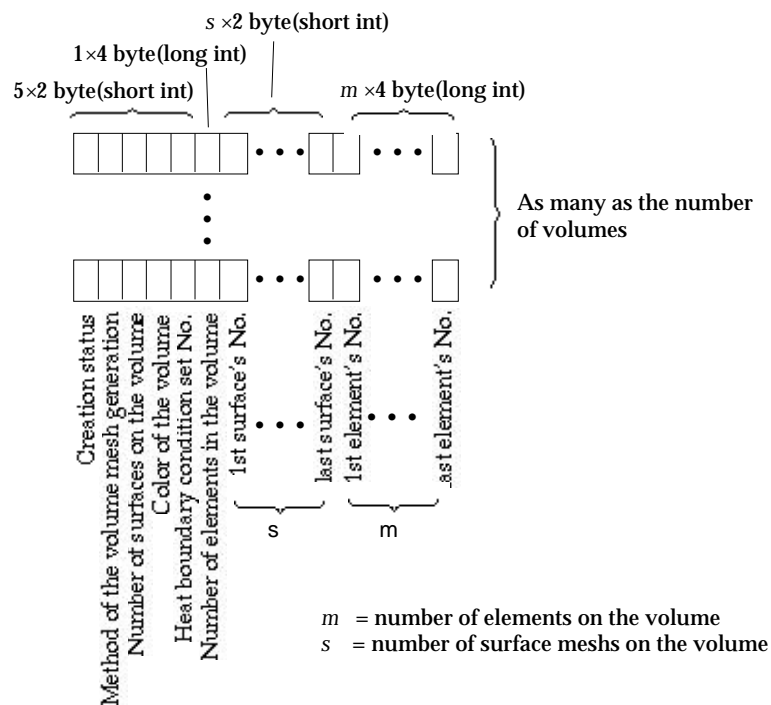
Volume data have the information on the volume meshes including their compositions, attribute assignments, and other characteristics.

A surface mesh is composed of nodes, elements, surface meshes. Information on the nodes on the volume can be derived from that of the elements on the volume. And therefore, information on the nodes are not included in the volume data.

- Creation status: Not used in the current version. Reserved for future use.
- Method of volume mesh generation: The entry has the code representing the method of volume mesh generation which was applied to create the volume.
 - 6: Volume mesh generation by 2 surface mapping
 - 7: Volume mesh generation by extrusion
 - 8: Volume mesh generation by box edge mapping
 - 9: Volume mesh generation by tetrahedron edge mapping
 - 9: Volume mesh generation by revolution
 - 10: Volume mesh generation by translation
 - 17: Volume mesh generation by duplication with “Revolve”
 - 20: Volume mesh generation by duplication with “Mirror”
 - 27: Volume mesh generation by prism edge mapping
 - 23: Volume mesh generation by twisting
- Number of surfaces on the volume : The number of surfaces attached to the

volume. The boundary surface of a volume is composed of surface meshes.

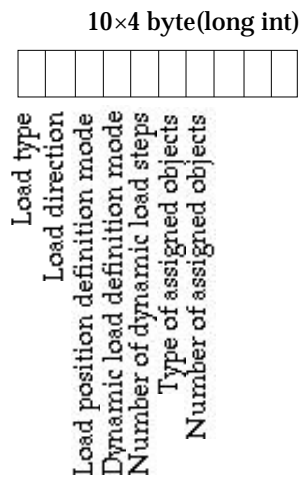
- Color of the volume : Code of the color assigned to the volume. This entry is not used in the current version, but reserved for future use.
- Heat boundary condition set No.: No. of the heat boundary condition set assigned to the volume mesh.
- Number of elements in the volume: Number of all elements, either active or inactive, composing the volume mesh.
- Surface No.: The No. of each surface in the volume mesh.
- Element No.: The No. of each element in the volume mesh.



<Volume data>

■ Load condition data - header record

These data are maintained chiefly for graphical user interface, and involved indirectly, through equivalent nodal forces, in finite element analysis. They are also the source from which the equivalent nodal forces are computed. The contents of load conditions vary depending on their type as described below.



< Load condition data >

- Load type: The code representing the type of the load.
 - 0: Nodal force
 - 1: Point force
 - 2: Uniform force
 - 3: Non-uniform force with linear variation
 - 4: Non-uniform force with parabolic variation
 - 5: Bilinear force
 - 6: Nodal moment
 - 7: Point moment
 - 8: Uniform moment
 - 9: Body force
 - 10: Hydrostatic in X
 - 11: Hydrostatic in Y
 - 12: Hydrostatic in Z
 - 13: Self straining force
 - 14: Thermal force
- Load direction: The code representing the direction of the load.
 - 0: X direction
 - 1: Y direction
 - 2: Z direction
 - 3: Direction normal to curve
 - 4: Direction tangent to curve
 - 5: Direction normal to surface
 - 6: Direction tangent to surface and normal to X direction
 - 7: Direction tangent to surface and normal to Y direction
 - 8: Direction tangent to surface and normal to Z direction

- Load position definition mode: This entry designates how the position of the load is defined. This applies only to mid-point force or mid-point moment, and is ignored for other load types. The position of mid-point force or moment can be defined either by the actual distance from one end of the structural member, or by the relative distance represented by the ratio to the length of the member. Distance ratio is equivalent to the distance divided by the total length of the member. The load position definition mode can be applied also for a curve instead of a member.
 - 0: Actual distance
 - 1: Relative distance
- Dynamic load definition mode: Time dependent dynamic loads can be defined either as a sinusoidal harmonic load or as a transient load.
 - 0: Harmonic load
 - 1: Transient load
- Number of dynamic load steps: The number of steps representing the dynamic load history. This data entry is effective only for transient load, and ignored for static loads or harmonic loads.
- Type of assigned objects: The type of the object to which the load condition is assigned.
 - 1: Not assigned
 - 1: Curve
 - 2: Surface
 - 3: Volume
 - 4: Element
 - 5: Node
 - 7: Surface primitive
 - 11: Line element
 - 12: Surface element
 - 13: Volume element
- Number of assigned objects: The type of the object to which the load condition is assigned.

■ Load condition data - attribute record

This data record consists of load attributes such as its magnitude, position and etc. Only one of the following 3 records is applied depending on whether the load is static, dynamic transient, or dynamic harmonic,

4×4 byte(float)

--	--	--	--

Value 1
Value 2
Value 3
Value 4

< Load attribute record for a static load>

4×4 byte(float)

--	--	--	--

•
•
•

--	--	--	--

Value 1
Value 2
Value 3
Value 4

As many as the number
of load time steps

< Load attribute record for a transient load>

4×4 byte(float)

--	--	--	--

Value 1
Value 2
Value 3
Value 4
Starting time
Ending time
Phase lag time
Angular velocity(radian/sec)

< Load attribute record for a harmonic load>

- Value 1, Value 2, Value 3, Value 4: These entries are data entered using the editable text items of “Load Condition” dialog. Their contents vary depending on the load type as shown below. Also refer to Chapter 5.
- Starting time: The time from which the harmonic load becomes effective.
- Ending time: The time when the harmonic load ends.
- Phase lag time(t_0): Phase lag to the starting point of the sinusoidal loading curve.
- Angular velocity(ω): Radian per second.

A harmonic load is defined by the following sine wave form

$$F \sin \omega (T - T_0)$$

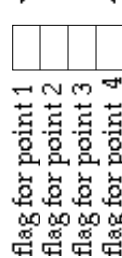
The amplitude of the load is given by value 1, 2, 3 or 4 in the load attribute record.

<Contents of load condition values>

Load Type	Value 1	Value 2	Value 3	Value 4
Nodal force	P	-	-	-
Mid-point force	P	d/L or L	-	-
Uniform force	w	-	-	-
Trapeziform force	w_1	w_2	-	-
Parabolic force	w_1	w_2	w_3	-
Bilinear force	w_1	w_2	w_3	w_4
Nodal Moment	M	-	-	-
Mid-point moment	M	d/L or L	-	-
Uniform moment	M	-	-	-
Body force	a	-	-	-
Thermal load	T_{av}	T_{gr}	-	-
Self-straining force	P	-	-	-
Hydrostatic in X,Y,Z	p_1	p_2	h_1	h_2

For bilinear force, the coordinates of the 4 corner points defining the distribution of the load should be supplied in addition as follows.

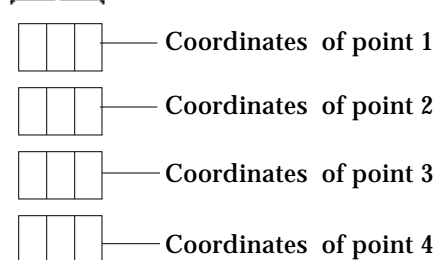
4×2 byte(short int)



Flags indicating whether the coordinates of each of the 4 corner points are defined by data input.

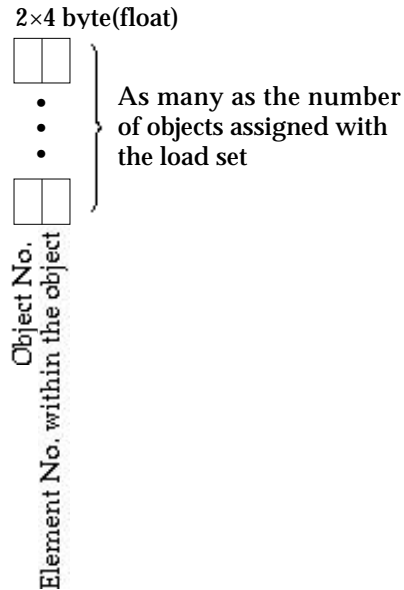
=0 : the coordinates of the point are not yet defined
=1 : the coordinates are defined.

3×4 byte(float)



<Coordinates of 4 corner points of bilinear force>

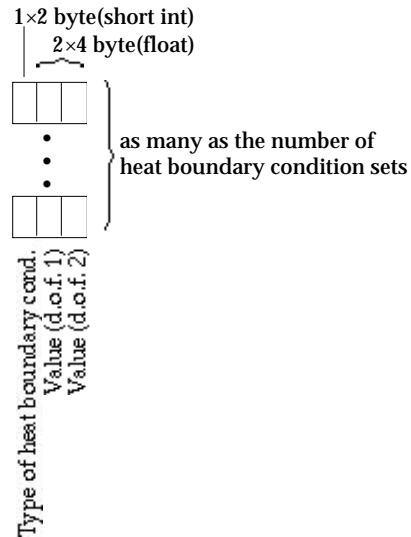
■ Load condition data - assigned objects



- Object No.: The No. of the object to which the load condition is assigned. Or, if the object type is “Element”, this entry has the No. of the object to which the element belongs.
- Element No. within the object: The No. of the element within the object. This entry is used, only in case the Object type is element.

■ Heat boundary condition data

The heat boundary condition data are maintained chiefly for graphical user interface, and involved indirectly, through equivalent nodal heat and convection boundary conditions, in finite element analysis. They are also the source from which the equivalent nodal heats and convection boundary conditions are computed. The items of heat boundary conditions vary depending on their type as described below.



<Heat boundary condition data>

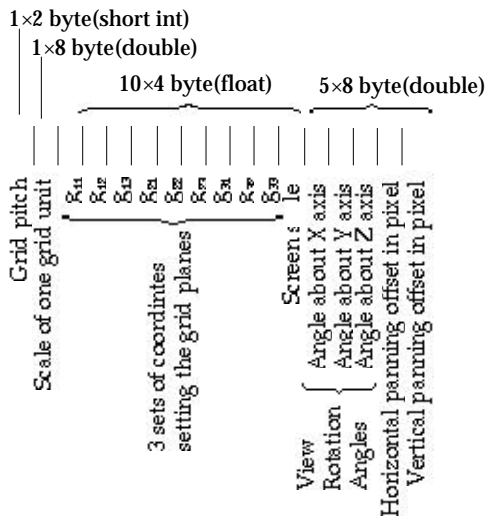
- Heat boundary condition type: The code representing the type of the heat boundary condition.
 - 0: Temperature
 - 1: Convection
 - 2: Heat flux
 - 3: Point source
 - 4: Element source
 - 5: Region source
- Value 1 - 4: These entries are data entered using the editable text items of “Heat Boundary” dialog. Their contents vary depending on the type of the heat boundary condition as shown below. Also refer to Chapter 5.

<Contents of load condition values>

Type	Value 1	Value 2
Temperature	temperature	-
Convection	convection coeff.	ambient temperature
Heat flux	flux	ambient temperature
Point source	heat quantity	-
Element source	heat quantity	-
Region source	heat quantity	-

■ View transformation data

View transform data consists of information on grid setting and view transformation in interactive work environment, the state of which at the time of file saving is retrieved and applied as the initial setting when the file opened. The view transformation data may be saved in an independent view files, if necessary. The record contains the same contents both for a VisualFEA file and for an independent view file.

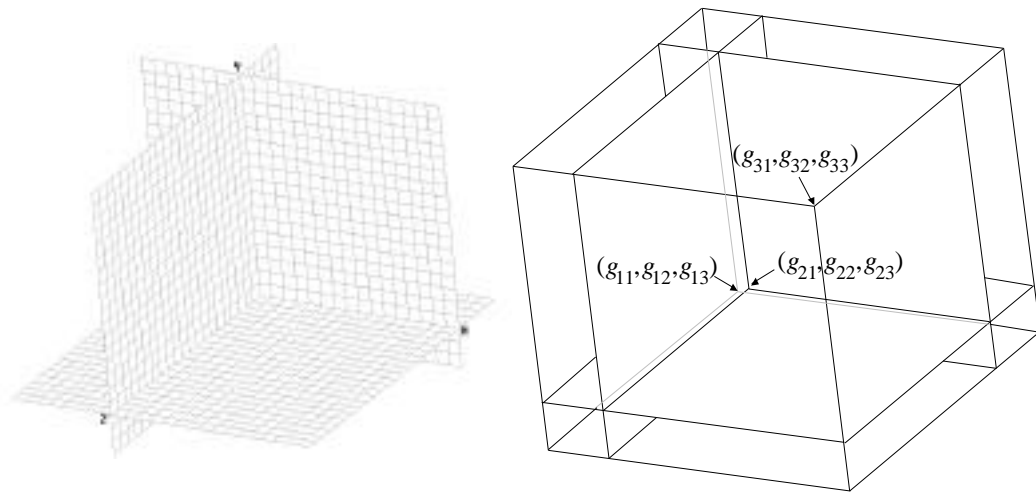


< View transformation data>

- **Grid pitch** : Number of pixels between grid lines on the screen with initial view transformation. This is the value entered “Grid Setting” dialog, or “Preference” dialog as described in Chapter 2.
- **Grid scale** : The actual distance represented by the interval between two adjacent grid lines. This value is also entered using “Grid Setting” dialog, or “Preference” dialog.
- **Grid box coordinates** : 3 sets of XYZ coordinates setting the planes as shown in the figure below. These values are inputted by interactively moving or resizing the grid planes as explained in Chapter 2.
- **Screen scale** : Zoom factor inputted by interactive view scaling action as explained in Chapter 2. The view scale, which is the ratio between the screen coordinates in pixels and the actual coordinates, are determined by

$$\text{view scale} = \text{grid pitch} \times \text{grid scale} \times \text{screen scale}$$

- **View rotation angles** : Angles of view rotation about X, Y and Z axes respectively. These angles are used in computing the view rotation matrix **R** by the following equation.



< Grid box coordinates>

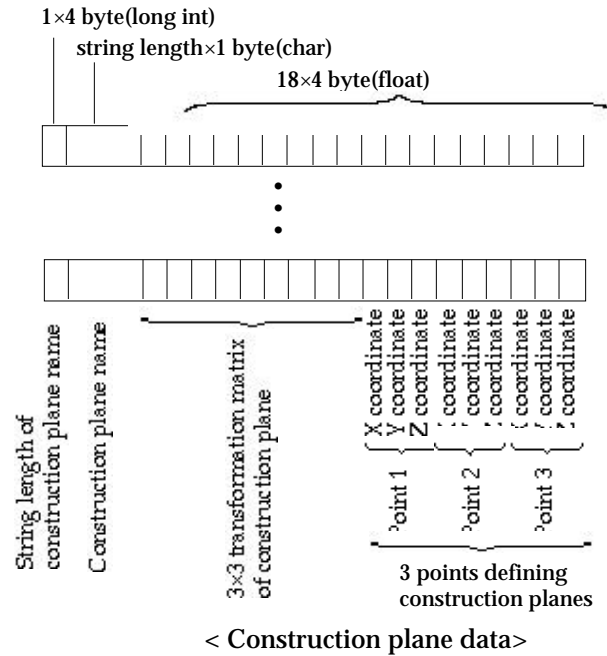
$$\mathbf{R} = \begin{matrix} \cos\theta_z & \sin\theta_z & 0 & \cos\theta_y & 0 & -\sin\theta_y & 1 & 0 & 0 \\ -\sin\theta_z & \cos\theta_z & 0 & 0 & 1 & 0 & 0 & \cos\theta_x & \sin\theta_x \\ 0 & 0 & 1 & \sin\theta_y & 0 & \cos\theta_y & 0 & -\sin\theta_x & \cos\theta_x \end{matrix}$$

- Horizontal panning offset : Horizontal offset distance of screen view in pixels. This value is changed by panning the screen view in horizontal direction as described in Chapter 2.
- Vertical panning offset : Vertical offset distance of screen view in pixels. This value is changed by panning the screen view in vertical direction as described in Chapter 2.

■ Construction plane data

Construction plane is a user defined grid plane with grid points for coordinate input. They are created interactively as described in Chapter 2. A plane in 3 dimensional space is uniquely defined by 3 points on the plane. The planes are not usually parallel to X, Y or Z axis. Their orientation is saved in a 3×3 transformation matrix. Each construction plane has its own name which is saved as a character string. The length of the name is variable.

The construction plane data have as many records as the number of construction planes which is contained in the header record. Refer to Chapter 2 for creation and use of user defined grid plane.



- String length of construction plane name : The number of characters used for the construction plane name.
- Construction plane name: Character string of the construction plane name. The name is endowed at the time of creating the construction plane. And this name is used to retrieve the construction plane.
- Transformation matrix of construction plane: 3×3 matrix setting the orientation of the construction plane.
- Coordinates of 3 points defining the construction plane: X, Y and Z coordinates of 3 points which were inputted in order to define the plane.

■ Symbol size record

A force of a load condition is symbolized by an arrow. The length of the arrow is drawn approximately in proportion to the length of the force magnitude. The proportion can be set differently for various load types. It is also possible to set the reference magnitude of the force for unit length of the arrow differently for various load types. The force types are classified differently from the actual types used in load condition data. They are point load, curve force, surface, volume force, point moment, curve moment, self straining force and thermal force.

The line length of the arrow in pixels is computed as follows:

$$\text{arrow length} = \frac{\text{force magnitude}}{\text{reference magnitude}} \times \text{unit length} \times \text{symbol scale}$$

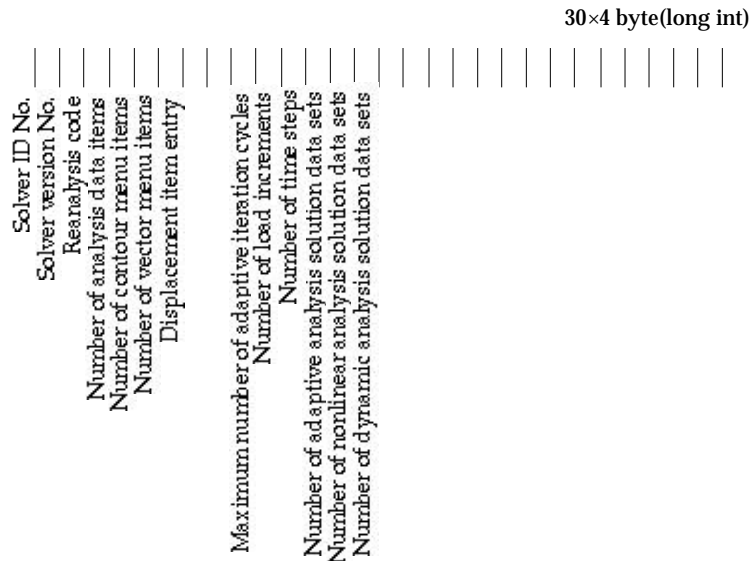
- Symbol scale : One symbol scale for each one of the load types,
- Reference magnitude: One reference magnitude for each one of the load

Analysis data

The analysis data are generated as results of finite element processing either by VisualFEA or by an external solver. In order to visualize the analysis data by VisualFEA, they should be appended to the end of the modeling data in the same VisualFEA file.

■ Analysis data master record

Analysis data master record consists of the solver's identification and basic information on the analysis items. The record has 30 entities of 4 byte long integer. The blank spaces are reserved for future use.



<Analysis data master record>

- Solver ID No.: A number given to the solver in order to identify the external solver. In order to avoid conflict due to assigning the identical number for different solvers, it is advisable to use a large number within the range of 4 byte long integer.
- Solver version No.: Solver's version No if any. This value should be an integer number. This entry may be used to overcome incompatibility between different versions of the solver.
- Reanalysis code: This entry indicates whether the matrix file necessary for reanalysis is retained or not. The matrix file may be either saved for reanalysis or removed at the end of the finite element processing, depending on the setting of the solver options.
 - 0: The matrix file was removed. Reanalysis mode is not allowed.
 - 1: The matrix file was save. Reanalysis mode is allowed.
- Number of analysis data items: The number of data items obtained from the

finite element processing, and written in the VisualFEA file.

- **Number of contour menu items:** The number of popup menu items in the “Contour” dialog for contouring of the analysis data.

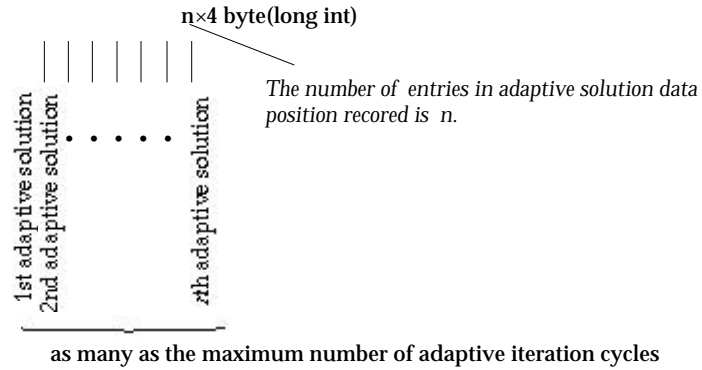
The contour menu items may be customized by the external solver. It is the responsibility of the solver to count the custom popup menu items and write the number into this entry.

- **Number of vector menu items:** The number of popup menu items in the “Vector” dialog for contouring of the analysis data.
- **Displacement item entry:** This entry directs the displacement data position entry in the analysis data position record.
- **Maximum number of adaptive iteration cycles:** The maximum number of iteration cycles is set as a criterion of terminating the iteration, prior to adaptive processing. This value is used to secure the space of the adaptive analysis data position record, and is ignored for non-adaptive analysis.
- **Number of load increments:** The value of this entry is used to secure the space of the nonlinear analysis data position record, and is ignored for linear analysis.
- **Number of time steps:** The value of this entry is used to secure the space of the dynamic analysis data position record, and is ignored for static analysis.
- **Number of adaptive analysis solution data sets:** The number of data sets including the intermediate and the final data sets obtained from adaptive analysis process.
 - 0: for non-adaptive analysis.
 - 2: for adaptive analysis in which only the data sets of the initial and the final stages are stored.
 - 3 or greater: for the data sets of the intermediate as well as the initial and the final stages.
- **Number of nonlinear analysis solution data sets:** The number of data sets stored in the intermediate and the final steps of load increments for nonlinear analysis.
 - 0: for linear analysis.
 - 1: for nonlinear analysis in which only the final solution is stored.
 - 2 or greater: for nonlinear analysis in which the intermediate data as well as the final solution data are stored.
- **Number of dynamic analysis solution data sets:** The number of data sets stored for the intermediate and the final time steps of dynamic analysis. This entry is ignored for static analysis.

■ Adaptive analysis data position record

Adaptive analysis data position record has the information on the offset distance, in bytes, from the beginning of the file to the starting point of each data sets stored

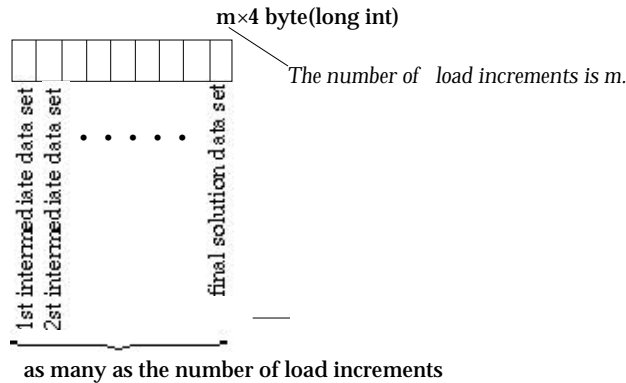
for the intermediate process of an adaptive analysis. The record has entities of 4 byte long integers, as many as the maximum number of adaptive iterations. This record is included only in case of adaptive analysis.



<Adaptive analysis data position record>

■ Nonlinear analysis data position record

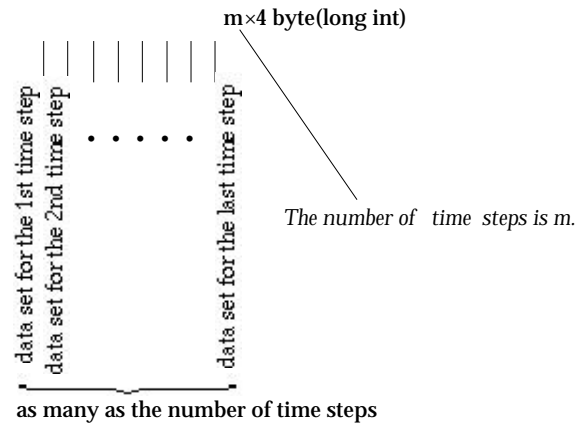
Nonlinear analysis data position record has information on the offset distance, in bytes, from the beginning of the file to the starting point of each data sets stored for the intermediate process of a nonlinear analysis. The record has entities of 4 byte long integer as many as the number of load increments. This record is included only in case of nonlinear analysis.



<Nonlinear analysis data position record>

■ Dynamic analysis data position record

Dynamic analysis data position record has the information on the offset distance, in byte, from the beginning of the file to the starting point of each data sets stored for whole time steps of a dynamic analysis. The record has entities of 4 byte long integer as many as the number of time steps. This record is included only in case of dynamic analysis.

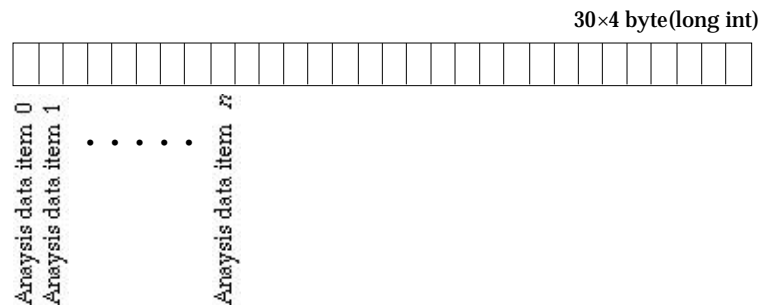


<Dynamic analysis data position record>

■ Analysis data position record

Analysis data position record has information on the offset distance, in bytes, from the beginning of the file to the starting point of each analysis data item. The record has 30 entities of 4 byte long integer. The blank spaces are reserved for future use.

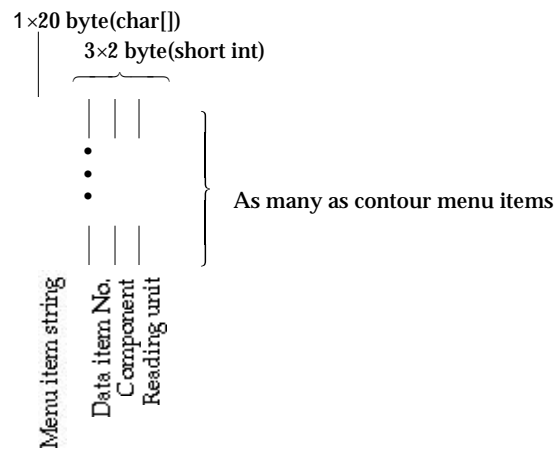
- Analysis data item 1 through n : The positions of analysis data items.



<Analysis data position record>

■ Custom contour menu data

The popup menu items in "Contour" dialog can be customized by the external solver. For this purpose, the solver should write the following menu data in front of the analysis data in the VisualFEA file.



<Custom contour or vector menu data>

- Menu item string: Menu string with 20 characters including blank space. The string appears as an popup menu item.
- Data item No.: The data with this item No. will be displayed when the menu item is chosen.
- Component: The component of the data item to be displayed.
- Entry No. or type: One component data consist of many data unit. Entry No. designates which entry out of a data unit is to be read and displayed. Entry type designates how a data unit is read and displayed. If this entry has a small integer value like 0,1,2,3,..., then this represent the entry No. Otherwise, this is relevant to entry type as follows (The macro in the header file is shown within the parenthesis.):

6000 (DISPLACEMENT_NORM) : Read all entries of each data unit, and make one norm of these values. For example,

$$\ddot{u} = \sqrt{u_0^2 + u_1^2 + u_2^2}$$

7000 (READ_ALL) : Read all entries of each data unit.

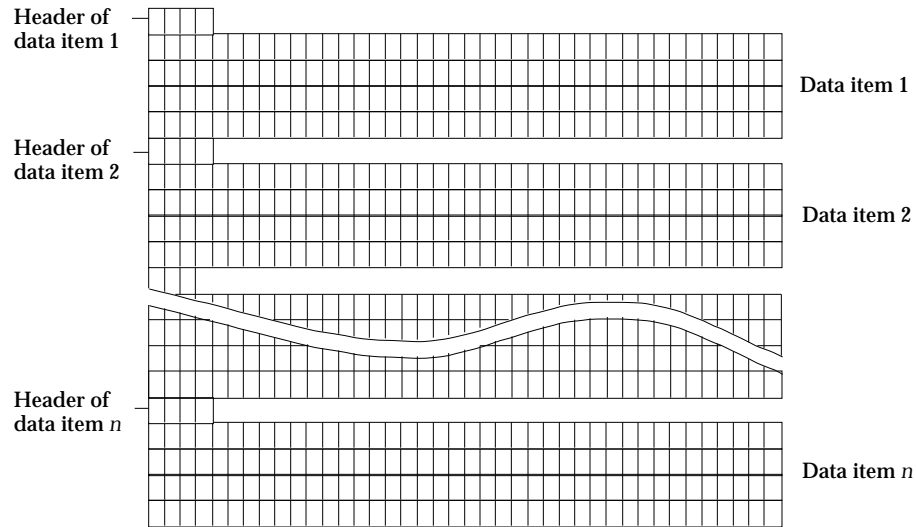
■ Custom vector menu data

The popup menu items in “Vector” dialog can be customized by the external solver in the same way as the case of “Contour” dialog.

■ Analysis data items

The number of analysis data items are specified in the analysis data master record described above. Each data item is obtained as the final products of finite element processing, either by VisualFEA or an external solver. The analysis data have hierarchical construction of entry, data unit, component and item. Each data item

consists of header and main body as shown in the following figure.



< Structure of analysis data >

The header record has the information on the hierarchical structure of the data. A data item is composed of a few components. A component has a number of data units. A data unit has a uniform or a variable number of consecutive data entries.

- **Number of components:** One data item has either only one component or more than one components. This entry specifies the number of components composing the data item.
- **Number of data units:** Data unit is an aggregate of consecutive data entries which are read as one unit. There are number of data units in one component of a data item. This may be , for example, the total number of elements or the number of nodes, used in the model.
- **Unit size:** A data unit may have only one entry or a number of entries. the unit size may be uniform for all data units in the component, or may vary from unit to unit. For example, if the data item is the principal directions, the unit sizes are uniformly the same as the spatial dimension. If the data item is the displacements in a mixed structure, the unit sizes may not be uniform, because the number of nodal d.o.f. are not necessarily the same for all nodes. The following macro is defined for variable nodal d.o.f. in the header file.

2000 (VARIABLE_NODAL_DOF) : A data unit consists of data entries as many as the nodal d.o.f. which varies from node to node.

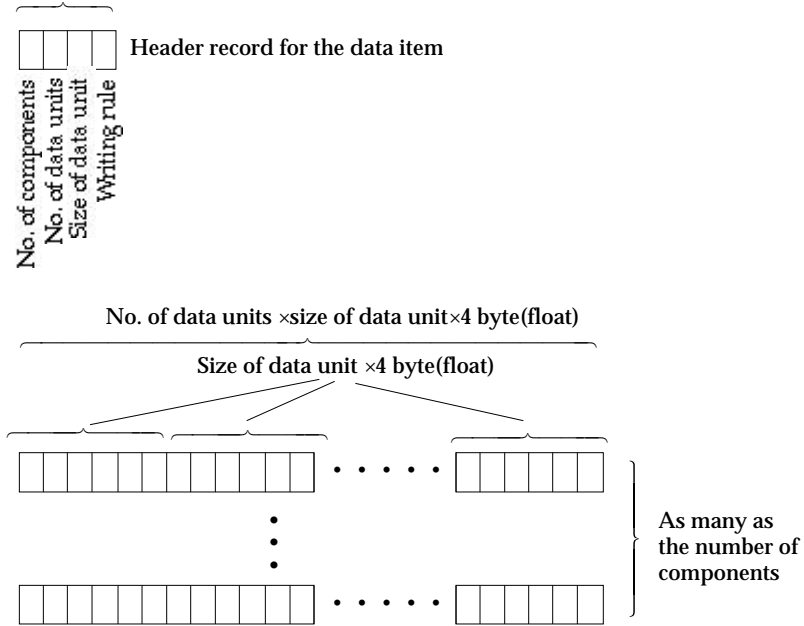
- **Writing rule:** This entry specifies what a data unit is represented by. The following macros are defined in the header file.

1 (WRITE_NODE) : A data unit matches with one node. So, there are as many data units as the number of nodes.

2 (WRITE_ELEM) : A data unit matches with one element. So, there are as many data units as the number of elements.

3 (WRITE_ELEM_NODE) : A data unit matches with each one of the nodes in an element. So, there are as many data units as the sum of the numbers of nodes in all elements.

4×4 byte(long int)



< Structure of an analysis data item >