

1 STRUCTURAL FILE MANIPULATOR

StruMan is designed for being an efficient structural file manipulator in connection with modelling of framed structures.

The user may give all input on one file, or distribute the data on several files. All control parameters are specified in the Analysis Control File. Structure data can also be read from this file, but is usually given on one separate file. The specific content of these files is not important, as long as all data are present.

StruMan supports following input formats:

- SESAM
- SESAM Substructures
- UFO
- ABAQUS (Subset according to PATABA 3.1A StyleGuide)
- SACS

and StruMan recognizes the different input file formats **automatically**.

StruMan exports the structural information to several different file formats, (see page 1-5).

The structural files may be written with a text editor, or generated by preprocessors.

The UFO structural file format combined with the possibility to export the structural model for visualization and verification in **XFOS** makes StruMan being a simple but in many cases an efficient **Preprocessor**.

1.1 GENERAL INFORMATION

The input data are organized in records, each record starting with a record identifier of up to eight significant characters. Each record may consist of one or more lines of data, terminating on the next record identifier. Each line may be **up to 132 characters long**. The data items may be integer or real data.

The data records may be given in an arbitrary order.

In this manual, each record is presented in a standard frame. Each frame represents either one single record or a sequence of similar records.

1.2 FORMATS

The special StruMan control data, SESAM input data and UFO input are read by SAM/FIL, a FORTRAN free-format reader and decoder. This means that the data items may be written anywhere on the line, **as long as the specified order is satisfied**. The data items must be separated with at least one blank (exceeding blanks are ignored). **Note that blank is exclusively interpreted as a delimiter, and can not be used to specify a zero value as accepted by standard FORTRAN READ.**

Other input formats, (F.inst SACS, ABAQUS) follows the actual file system input definitions and the following information is not relevant except for the StruMan Control File!

Important:

All digits, letters and/or special symbols in a data item must be given consecutively without blanks.

Comments

Lines with following characters in the first column are interpreted as comments, and are simply ignored. Comments may occur anywhere in the input data stream.

'
*

%
!

Example:

```
'THIS IS A COMMENT  
# NOTE! COMMENTS ARE IGNORED BY THE PROGRAM
```

In-Line Comments:

The character ! may be used to terminate a data line, and data items on the rest of the line are just ignored.

Example:

```
#   Geo_ID  Do  Thick  
PIPE  10   500  10    ! NEW 96-02-07  
#PIPE  10   600  10    ! OLD 96-02-07
```

This is an useful option in connection with modification/correction of input files.

Alphanumeric Data Items

An alphanumeric data item may consist of one or more characters. The first character is always a letter (A-Z), while the remaining ones may be letters, digits or special symbols (except /, \$, & and blank).

There are no upper limit to the number of characters in an alphanumeric data item. However, only the first 8 characters will be decoded, and all characters in excess of this are simply ignored.

Integer Data Items

- All characters must be digits
- The first digit may be preceded by + or -

Example:

0 1 -27 +66

Real Number Data Items

A real number data entry may consist of up to 3 components, i.e. an integer part i, a decimal part d, and exponent part e. The following 4 basic forms are accepted:

(±)i (±)i. (±)i.d (±).d

These may all be combined with exponent parts yielding the forms:

(±)iE(±)e (±)i.E(±)e (±)i.dE(±)e (±).dE(±)e

Example:

0 +1. -0.2E14+17.E-31.8E+3

Text Strings

Text strings may consist of one or more characters, which may be letters (A-z), digits or special symbols.

72 characters are stored in a text string, beginning at the 9th character of the line.

Example:

This text string uses special characters & \#

and will be stored as

t string uses special characters & \#

1.3 INPUT FILE FORMAT

SESAM See SESAM Part

UFO See UFO Part

ABAQUS See ABAQUS Part

SACS See SACS Part

1.4 STRUMAN INPUT RECORDS

		Page
Analysis Identification	HEAD	1-7
Output File Format		
SESAM file format	SESAM	1-7
UFO file format	UFO	1-7
ABAQUS file format	ABAQUS	1-7
PATRAN file format	PATRAN	1-7
FAROW file format	FAROW	1-7
NIRWANA file format	NIRWANA	1-7
MOVIE.BYU image plot file	MOVBYU	1-7
XFOS binary database	XFOS	1-7
PREFRAME journal file	PreFrame	1-7
Substructure Input		
Merge Sub Structures	MERGSTRU	1-8
Tolerance for merge of nodes	TOLMERGE	1-8
Defining substructures to be merged	SUB_STRU	1-9
New Id's, Substructure Multiplier	SUBSMULT	1-10
Handling of substructure Transformations	SUBTRAN	1-10
General Model Manipulation		
Unit Conversion	CONVUNIT	1-11
Coordinate Transformation	STRUTRAN	1-12
Load Transformations	LOADTRAN	1-12
Combine several loadcases to one loadcase	COMBLOAD	1-13
Shrink, Refine Structural model		
Limitation of structural model	LIMITMOD	1-14
Identifier Add, (mesh refinement)	ID_ADD	1-20
Remove Surplus Properties	SHRIPROP	1-20

The input records is presented in a standard frame with the following format:

RECORD IDENTIFICATOR	Parameter No. 1	Parameter No. 2	Parameter No. 3	Parameter ...
<i>Parameter</i>	<i>Description</i>			
1	Description of contents and default values			
2	Description of contents and default values			
3	Description of contents and default values			
Optional box for comments, notes, exceptions etc.				

NOTE ! Parameters written in **bold** set are mandatory.
Parameters in regular set are optional, and default values will be used if omitted.

Analysis Identification

<pre> HEAD < < Three lines of text identifying the analysis > < > </pre>
<p>Character 9 to 80 from each line are stored as text strings.</p> <p>This record is given only once.</p>

Output File Format

<IDENT>																					
<table border="1"> <thead> <tr> <th style="text-align: left; padding: 5px;"><i>Identifiers</i></th> <th style="text-align: left; padding: 5px;"><i>Description</i></th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">SESAM</td> <td style="padding: 5px;">SESAM format. Output files: <prefix>_sesam.fem <prefix>_sesam_load.fem</td> </tr> <tr> <td style="padding: 5px;">UFO</td> <td style="padding: 5px;">UFO format. Output file: <prefix>_ufo.fem</td> </tr> <tr> <td style="padding: 5px;">ABAQUS</td> <td style="padding: 5px;">ABAQUS format. Output file: ABAQUS.inp</td> </tr> <tr> <td style="padding: 5px;">PATRAN</td> <td style="padding: 5px;">PATRAN format. Output file: patran.inp</td> </tr> <tr> <td style="padding: 5px;">FAROW</td> <td style="padding: 5px;">FAROW format. Output file: farow.inp</td> </tr> <tr> <td style="padding: 5px;">NIRWANA</td> <td style="padding: 5px;">NIRWANA format Output file: nirwana.inp</td> </tr> <tr> <td style="padding: 5px;">MOVBYU</td> <td style="padding: 5px;">MOVIE.BYU format Output file: movbyu.mg</td> </tr> <tr> <td style="padding: 5px;">XFOS</td> <td style="padding: 5px;">XFOS binary database Output file: <prefix>.raf</td> </tr> <tr> <td style="padding: 5px;">PreFrame</td> <td style="padding: 5px;">PREFRAME journalfile Output file: <prefix>.jnl</td> </tr> </tbody> </table>	<i>Identifiers</i>	<i>Description</i>	SESAM	SESAM format. Output files: <prefix>_sesam.fem <prefix>_sesam_load.fem	UFO	UFO format. Output file: <prefix>_ufo.fem	ABAQUS	ABAQUS format. Output file: ABAQUS.inp	PATRAN	PATRAN format. Output file: patran.inp	FAROW	FAROW format. Output file: farow.inp	NIRWANA	NIRWANA format Output file: nirwana.inp	MOVBYU	MOVIE.BYU format Output file: movbyu.mg	XFOS	XFOS binary database Output file: <prefix>.raf	PreFrame	PREFRAME journalfile Output file: <prefix>.jnl	<p>With this record, the user specifies the output format to be used.</p> <p>If no output formats are specified, StruMan reads and checks the input only.</p> <p>This record is given once.</p>
<i>Identifiers</i>	<i>Description</i>																				
SESAM	SESAM format. Output files: <prefix>_sesam.fem <prefix>_sesam_load.fem																				
UFO	UFO format. Output file: <prefix>_ufo.fem																				
ABAQUS	ABAQUS format. Output file: ABAQUS.inp																				
PATRAN	PATRAN format. Output file: patran.inp																				
FAROW	FAROW format. Output file: farow.inp																				
NIRWANA	NIRWANA format Output file: nirwana.inp																				
MOVBYU	MOVIE.BYU format Output file: movbyu.mg																				
XFOS	XFOS binary database Output file: <prefix>.raf																				
PreFrame	PREFRAME journalfile Output file: <prefix>.jnl																				

Substructure Input

MERGSTRU	
<i>Parameter</i>	<i>Description</i>
	<p>With this record, the user switch on the general merging of sub structures specified on special files. The command must be specified on STRUMAN's control file. (The substructure file names will be generated by STRUMAN based on the prefix and substru no).</p> <p>NOTE: No structure file is given under : 'Structrual File prefix.'</p> <p>STRUMAN will ask for 'Substru. file prefix :' and will then open and read the first level substructure files, which could be of any of the legal file formats .</p> <p>File name convension: Substructure no n is located at file <prefix>tn.fem (Example: if prefix troll_ is specified, substructure no 23 is located at troll_t23.fem)</p> <p>This record is given once.</p>

TOLMERGE Tol	
<i>Parameter</i>	<i>Description</i>
Tol	<p>Nodes with distance less than the specified tolerance will be merged in connection with the substructure assembly process.</p> <p>Default tolerance is T_min where T_min is minimum wall thickness detected in the structure.</p>
	<p>With this record, the user defines the tolerance to be used in connection with the substructure assembly process.</p> <p>This record is given once</p>

SUB_STRU		ID	T ₁₁	T ₁₂	T ₁₃
			T ₂₁	T ₂₂	T ₂₃
			T ₃₁	T ₃₂	T ₃₃
			X ₁	X ₂	X ₃
Parameter	Description				
ID	Substructure ID, Note that the file name is composed based on this ID.				
T ₁₁ - T ₃₃	The 9 terms in the Transformation Matrix defined according to the <i>STRUTRAN</i> definition				
X ₁ - X ₃	The 3 terms defining the coordinate translation according to the <i>STRUTRAN</i> definition				
<p>With this record, the user defines one substructure (among several), which should be merged into one total model. Same sub structure ID might be referred to several times, (typically with different coordinate translation definitions), see <i>Example 1</i> below.</p>					
<pre> ===== Example 1 ===== ===== Repeating/Translating same Structure ===== ===== Ufo SubTrans 1 SubsMult 100000 MergStru Sub_Stru 1 1 0 0 ! Frame 1 0 1 0 0 0 1 0 0 0 , Sub_Stru 1 1 0 0 ! Frame 2 0 1 0 0 0 1 27.0 0 0 , Sub_Stru 1 1 0 0 ! Frame 3 0 1 0 0 0 1 </pre>					
<pre> ===== Example 2 ===== = Merging two structures, which has a common co-ordinate system = ===== Ufo SubsMult 1000 MergStru Sub_Stru 101 1 0 0 ! 1'st Substru 0 1 0 0 0 1 0 0 0 , Sub_Stru 201 1 0 0 ! 2'nd Substru 0 1 0 0 0 1 </pre>					
<p>This record may be repeated</p>					

SUBSMULT Multiplier	
<i>Parameter</i>	<i>Description</i>
multiplier	Assembled model ID = Substr. no * Multiplier + Local ID
<p>With this record, the user defines how the external numbers are created during the assembly of substructures.</p> <p>Example 1: Node with ID = 10200 defined in substructure no 7 will get ID no 7010200 with multiplier = 1000 0000</p> <p>Example 2: Element with ID = 100200 defined in substructure no 4 gets ID = 4100200 when multiplier = 1000 000</p> <p>Example 3: Geometry with ID = 655030 defined in substructure 2 gets ID = 20655030 if the multiplier = 10 000 000</p> <p>This record is given once.</p>	

SUBTRAN Type	
<i>Parameter</i>	<i>Description</i>
Type	<p>Type = 0: $\mathbf{X}_G = \mathbf{T}^T (\mathbf{X}_L - \mathbf{T}_{trans})$ (default, according to SESAM)</p> <p>Where:</p> <ul style="list-style-type: none"> \mathbf{T}^T is the rotational transformation matrix (3 x 3) defined by $\mathbf{X}_L = \mathbf{T} \mathbf{X}_G$ \mathbf{X}_L is coordinates in local system \mathbf{X}_G is coordinates in global (assembly) system \mathbf{T}_{trans} is the coordinates of the vector pointing from local origin to the global origin. The vector is defined in the local coordinate system. <p>Type = 1: $\mathbf{X}_G = \mathbf{T}^T \mathbf{X}_L + \mathbf{T}_{trans}$</p> <p>With following difference from Type = 0:</p> <ul style="list-style-type: none"> \mathbf{T}_{trans} is the coordinates of the vector pointing from global origin to the local origin. The vector is defined in the global coordinate system.
<p>With this record, the user defines how the coordinates of substructures are transferred to global (assembly) system, (relevant in connection with MERGSTRU / SUB_STRU only).</p> <p>This record is given once.</p>	

General Model Manipulation

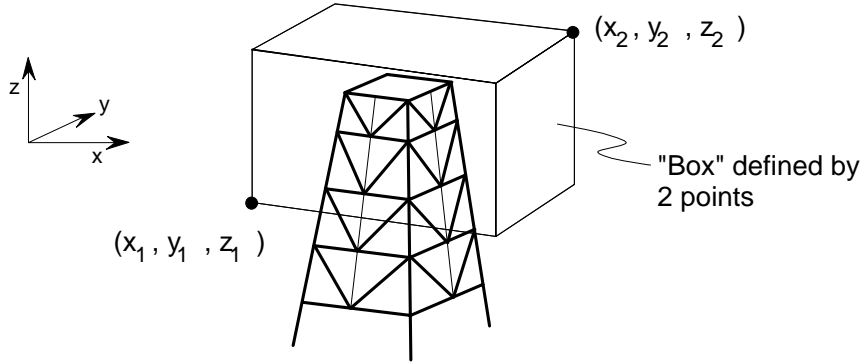
CONVUNIT		Length_Fac	Force_Fac	Mass_Fac
<i>Parameter</i>	<i>Description</i>			
Lengt_Fac	New Length = Old Length * Length_Fac			
orce_Fac	New Force = Old_Force * Force_Fac			
Mass_Fac	New MAss = Old_Mass * Mass_Fac			
<p>With this record, the user defines factors to be used during the Unit Conversion Process.</p> <p>Example 1: Input model is defined in Feet, Kips and lbs, and conversion to SI units should be preformed, (m, N and kg) Following three factors will transfer all structural data to a consistent model in SI- units: Length_Fac : 0.3048 Force_Fac : 4454.0 (9.81 * 454) Mass_Fac : 0.454</p> <p>Example 2: Input model is defined in m, kN and kg and conversion to m, N and kg should be performed. Following three factors will transfer all structural data to a consistent model: Length_Fac : 1.0 Force_Fac : 1000.0 Mass_Fac : 1.0</p> <p>This record is given once.</p>				

STRUTRAN	
	T_{11} T_{12} T_{13} T_{21} T_{22} T_{23} T_{31} T_{32} T_{33} X_1 X_2 X_3
<i>Parameter</i>	<i>Description</i>
$T_{11} - T_{33}$	The 9 terms in the Transformation Matrix defined through: $X_L = T X_G$
$X_1 - X_3$	The 3 terms defining the vector pointing from local origin to the global origin. The vector is defined in the local coordinate system.
<p>With this record, the user defines the transformation matrix to be used during transformation of the input structural model to the model defined in the new coordinate system.</p> <p>Switches ON the transformation operation</p> <p>This record is given once</p>	

LOADTRAN	
<i>Parameter</i>	<i>Description</i>
<p>With this record, the user defines that loads should be transferred into the new coordinate system defined under STRUTRAN.</p> <p>This record may be repeated.</p>	

COMBLOAD		Comb_Case
		L_Case ₁ Factor ₁
		L_Case ₂ Factor ₂
		L_Case ₃ Factor ₃
		L_Case _n Factor _n
<i>Parameter</i>	<i>Description</i>	
comb-case	The result of the combination is collected in a load case with number (ID) = Comb_Case	
L_case ₁	First Load_Case contributing to the combination.	
Factor ₁	The loads in this load_case are multiplied with Factor ₁ etc...	
<p>With this record, the user defines a load combination to be generated and written to file as a new load_case.</p> <p>Example: COMBLOAD 3</p> <pre> 4 1.3 3 2.0 10 0.7 13 -0.55 </pre> <p>Generates a new load case 3 as follows:</p> $I_case3 = (I_case4 * 1.3 + I_case3*2.0 + I_case10*0.7 - I_case13 *0.55)$ <p>This record is given once.</p>		

Shrink, Refine Structural model

LIMITMOD type x1 y1 z1 x2 y2 z2	
<i>Parameter</i>	<i>Description</i>
type	Description type: Type = 1 : Elements inside box are included Type = -1 : Elements outside box are included
x1, y1, z1	Coordinates of point 1 used to define the limitation "box"
x2, y2, z2	Coordinates of point 2 used to define the limitation "box"
	See figure below. Default: Entire structure included.
<p>With this record, the user may extract the part of the entire structure which is of interest. All elements inside the defined "box" (type = 1) or outside the "box" (type = -1) will be included in the element model. Elements which do not fulfil the limitation test are disregarded. (Confer XFOS "Edit/Clip/Box")</p> <p>This record may be repeated.</p>	
	

LIMITMOD type x1 y1 z1 x2 y2 z2 x3 y3 z3 dist	
<i>Parameter</i>	<i>Description</i>
type	Description type: Type = 2 : Elements closer to plane than 'dist' are included Type = -2 : Elements closer to plane than 'dist' are excluded
x1, y1, z1	Coordinates of point 1 used to define the limitation plane
x2, y2, z2	Coordinates of point 2 used to define the limitation plane
x3, y3, z3	Coordinates of point 3 used to define the limitation plane
dist	Distance from plane See figure below. Default: Entire structure included.
<p>With this record, the user may extract the part of the entire structure which is of interest. All elements closer to the defined plane (type = 2) or at larger distance from the plane (type = -2) will be included in the element model. Elements which do not fulfil the limitation test are disregarded. (Confer XFOS "Edit/Clip/Plane")</p> <p>This record may be repeated.</p>	

LIMITMOD type iel1 iel2 iel3	
<i>Parameter</i>	<i>Description</i>
type	Description type: Type = 3 : Specified elements only are included Type = -3 : Specified elements are excluded
iel1, iel2 ...	Specification of the elements to be included/excluded
<p>With this record, the user may extract the elements, element by element included/excluded in the structural model.</p> <p>NOTE! Type = 3 is not available together with other LIMITMOD types, and all elements have to be specified in the same LIMITMOD record.</p> <p style="padding-left: 40px;">Type = -3 may be used together with the other LIMITMOD types.</p> <p>LIMITMOD 3 is given once! LIMITMOD -3 may be repeated.</p>	

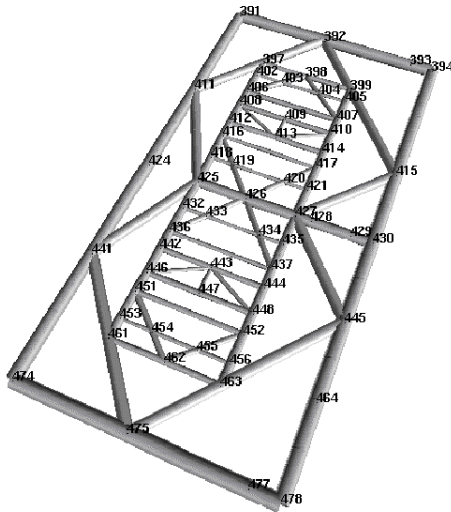
LIMITMOD Type	
<i>Parameter</i>	<i>Description</i>
Type	Type = 100 activates the model shrink procedure which changes sub_divided members into one beam element per physical member.
<p>With this record, the user switches on the model shrink procedure. Elements referring to the same cross_section ID and are all a part of the same physical member are merged into one element.</p> <p>This record is given once</p>	

LIMITMOD	Type	N_Geo
	D ₁	T ₁
	D ₂	T ₂
	D _{n_geo}	T _{n_geo}
<i>Parameter</i>	<i>Description</i>	
Type	Type = 102 activates the remove member procedure followed by the model shrink procedure . Members with Diameter and Thickness equal to any of the listed D/T's are removed from the finite element model. When the actual elements are removed, sub_divided members are replaced by one beam element per physical member.	
N_Geo	Number of D / T pairs defined in the list	
D	Outer Diameter	
T	Thickness	
<p>With this record, the user activates the remove member procedure followed by the model shrink procedure.</p> <p>Example: The detailed model of the conductor frame should be simplified by removing the smaller members. Removal of these members makes it possible to introduce only one element per physical member in the shrunk model, see figures below.</p>		

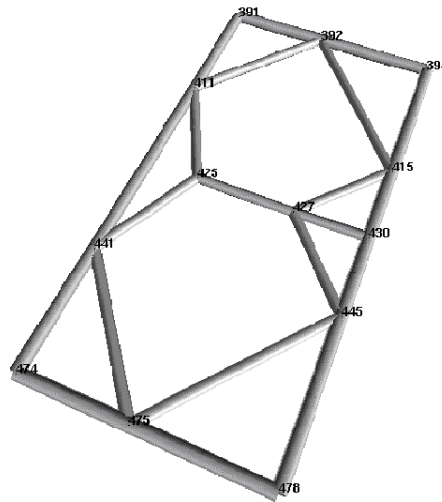
```
LIMITMOD 102 6
          0.219 0.013
          0.219 0.025
          0.219 0.038
          0.323 0.013
          0.323 0.025
          0.508 0.016
```

Beam elements having any of the 6 specified D / T's will be removed from the element model.

This record is given once



Original Frame



Shrunk frame. Original node- and element numbers are kept

LIMITMOD	Type	Tol_D	Tol_T	Flag
<i>Parameter</i>	<i>Description</i>			
Type	Type = 103 activates the model shrink procedure which changes sub_divided members into one beam element per physical member.			
Tol_D	Elements are merged into one if D_1/D_2 is within the specified tolerance $(1.0 - Tol_D) < D_1/D_2 < (1.0 + Tol_D)$. If FLAG = 1 the pipe with the larger diameter is used, otherwise the pipe with the smaller diameter is used.			
Tol_T	Elements are merged into one if T_1/T_2 is within the specified tolerance $(1.0 - Tol_T) < T_1/T_2 < (1.0 + Tol_T)$. If FLAG = 1 the pipe with the larger thickness is used, otherwise the pipe with the smaller thickness is used.			
FLAG	FLAG = 1 Pipe with larger D or T is kept FLAG = -1 Pipe with smaller D or T is kept			
<p>With this record, the user switches on the model shrink procedure which makes it possible for the user to remove f.inst strub/can element in a typical jacket model. The user controls the tolerance to be used during the shrinking process. Original ID's (node, element, material, cross_section, ..) are kept in the shrunked model.</p> <p>NOTE Both the Diameters and the Thicknesses must be within the specified tolerance if the elements should be merged into one.</p> <p>This record may be repeated.</p>				

ID_ADD iadd	
<i>Parameter</i>	<i>Description</i>
iadd	<p>Elements and nodes generated in connection with the REFINE command get external numbers (ID's) as follows:</p> <p>$ID = 100 * (Master_Elemet_ID) + IADD + ipart,$ $ipart = 2, ndivide$ (see REFINE command, UFO part)</p>
<p>With this record, the user defines how generated IDs are created.</p> <p>This record is given once.</p>	

SHRIPROP <Item>	
<i>Parameter</i>	<i>Description</i>
item	<p>Item to be shrunked:</p> <p>Mat : Surplus, identical materials are removed Geo : Surplus, identical geometries are removed Ecc : Surplus, identical eccentricities are removed LocalCoo : Surplus, identical local coord definitions are removed All : Surplus, identical properties (all mentioned above) are removed</p>
<p>With this record, the user specifies that surplus property references are removed if the actual data are "*"identical", (within a tolerance of 1.0E-6). Element references are updated.</p> <p>This record is given once.</p>	