RM2000

Static and Dynamic Analysis

of Spaceframes

USER GUIDE

APPENDIX

TDV Ges.m.b.H.

December 2002

 Heinz Pircher und Partner

Disclaimer and Copyright

Disclaimer

Much time and effort have gone into the development and documentation of *RM2000* and *GP2000*. The programs have been thoroughly tested and used.

The user accepts and understands that no warranty is expressed or implied by the developers or the distributors on the accuracy or the reliability of the program.

The user must understand the assumptions of the program and must apply engineering knowledge and skill to independently verify the results.

Copyright

Contents

1 Scripts

In RM2000, there is a scripting interface based on the TCL script language. Access to the RM2000 database is provided by RM-specific commands in TCL. For specific information about TCL itself, look for textbooks, search the internet (e.g.: http://www.scriptics.com) or check the HTML-based TCL syntax description provided with RM2000. In RM2000, a subset of TCL version 7.3 is implemented. All TCL commands for RM2000 database access are described in the following chapter.

A script is a simple text file **without formatting** (ASCII – text file) containing a sequence of commands. TCL scripts files should be named like 'filename.tcl'. To create a script file, open a text editor (e.g.: by selecting the 'editor' button from the icons at the top of the RM program), write the desired sequence of commands and save it as 'filename.tcl'. Input-Scripts can be started from within RM2000 by selecting the ⇧File ⇨Import Tcl-script option. Select your file ('filename.tcl') from the selection list or input the filename in the 'File' edit field. Choose whether you want to add the input to your project (partial project) or you want to overwrite the existing project by the input (complete project). Select `<OK>` to start the script. Log-, warning- and error messages will appear in the RM log.

Commands begin with a keyword and end at the end of the line. In between there can be parameters for the command. The number of parameters must correspond to the syntax definition of the command given in this chapter.

The '`#`' as the first character of a command will comment it out.
TCL allows the definition of user-defined commands. TDV provides a library with predefined commands. Experienced users can add an own command library.
One of the most important commands is the SERIE command:

```
[SERIE from to step]
```

This command produces a list of numbers beginning from "from" with a numerical distance of "step" up to "to". Examples:

`[SERIE 1 40 10]`	will produce the list `{ 1 11 21 31 }`
`[SERIE 1 5 1]`	will produce the list `{ 1 2 3 4 5 }`
`[SERIE 1 5]`	will produce the list `{ 1 2 3 4 5 }` (default step = 1)
`[SERIE 15.3 5.3 -2.5]`	will produce the list `{ 15.3 12.8 10.3 7.8 5.3 }`

Check the following syntax description of the specific commands whether you can use the **[SERIE from to step]** input with a specific command.

For multiple series, a second command is provided:

[SERIES *from to step start1 step1*]
[SERIES *from to step start1 step1 start2 step2*]
[SERIES *from to step start1 step1 start2 step2 start3 step3*]
[SERIES *from to step start1 step1 start2 step2 start3 step3 start4 step4*]

This command produces a list of lists of numbers beginning from “from”, “start1”, “start2”, … with a numerical distance of “step”, “step1”, “step2”, … until the first list reaches “to”. This command can be used for example for the definition of nodes along a line or for the definition of elements.
Examples:

```
[SERIES 1 40 10   12.0 0.5   15.0 –1.1]
```

will produce the list of lists:

```
{ { 1 11 21 31 } {12.0 12.5 13.0 13.5}  {15.0 13.9 12.8 11.7} }
```

Check the following syntax description of the specific commands whether you can use the [SERIES…] input with a specific command.

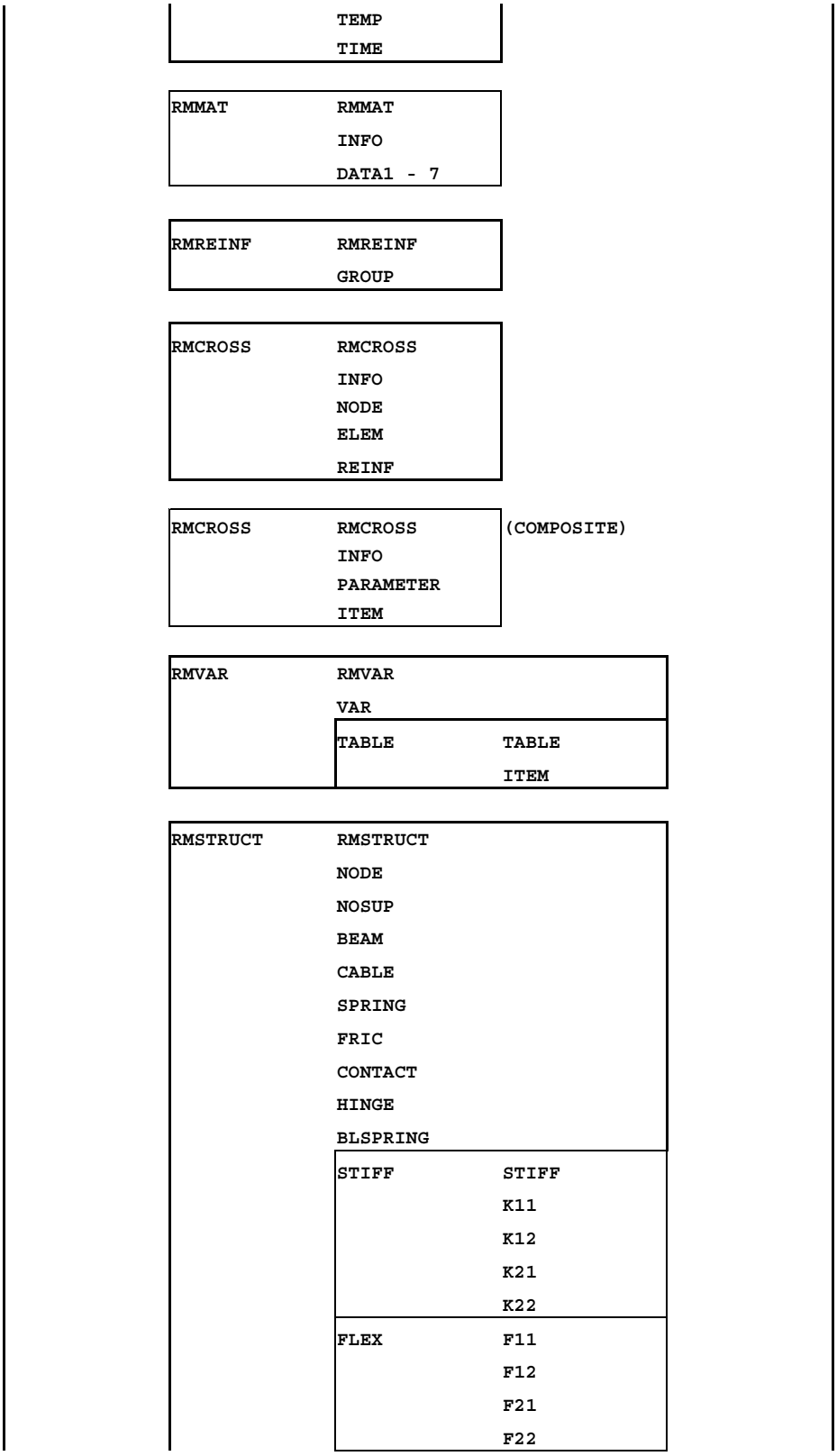
A simple script file creating 11 nodes and 10 beams can look like this:

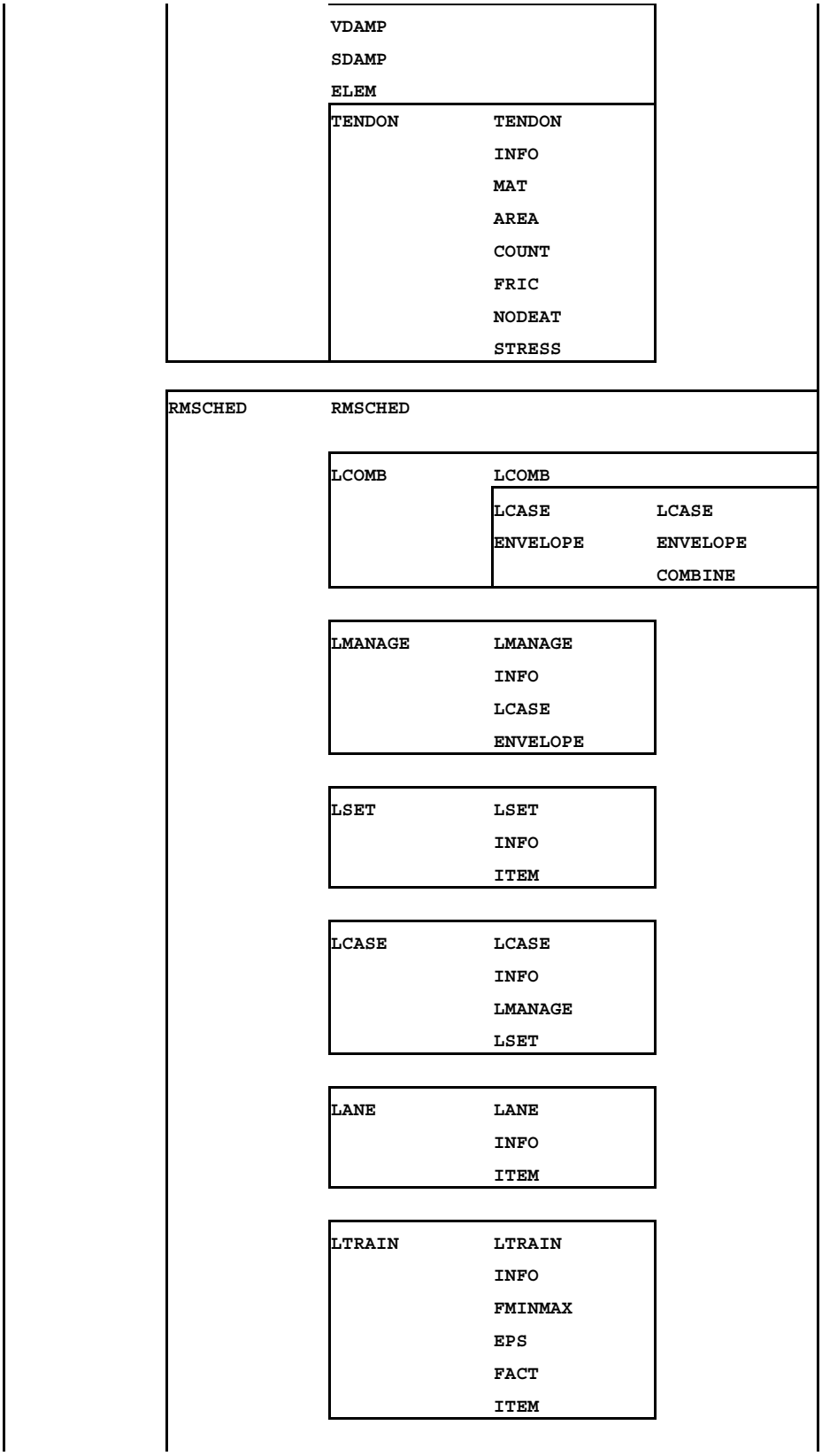
```
# start a RM session
RMJOB START

   # start the structure definition
   RMSTRUCT START

      # create 11 nodes: node 1 at (15.0 / 0.0 / 0.0) to node11 at (32.0 / 0.0 / 0.0)
      NODE [SERIES 1 11 1     15.0 1.7]

      # create element 1-10
      BEAM [SERIES 1 10 1  1 1    2 1]

      # assign the “B55” material to beams 1, 3, 5, 7 and 9
      BEAM [SERIE 1 10 2] MAT “B55”

   # end the structure definition
   RMSTRUCT END

# end a RM session
RMJOB END
```

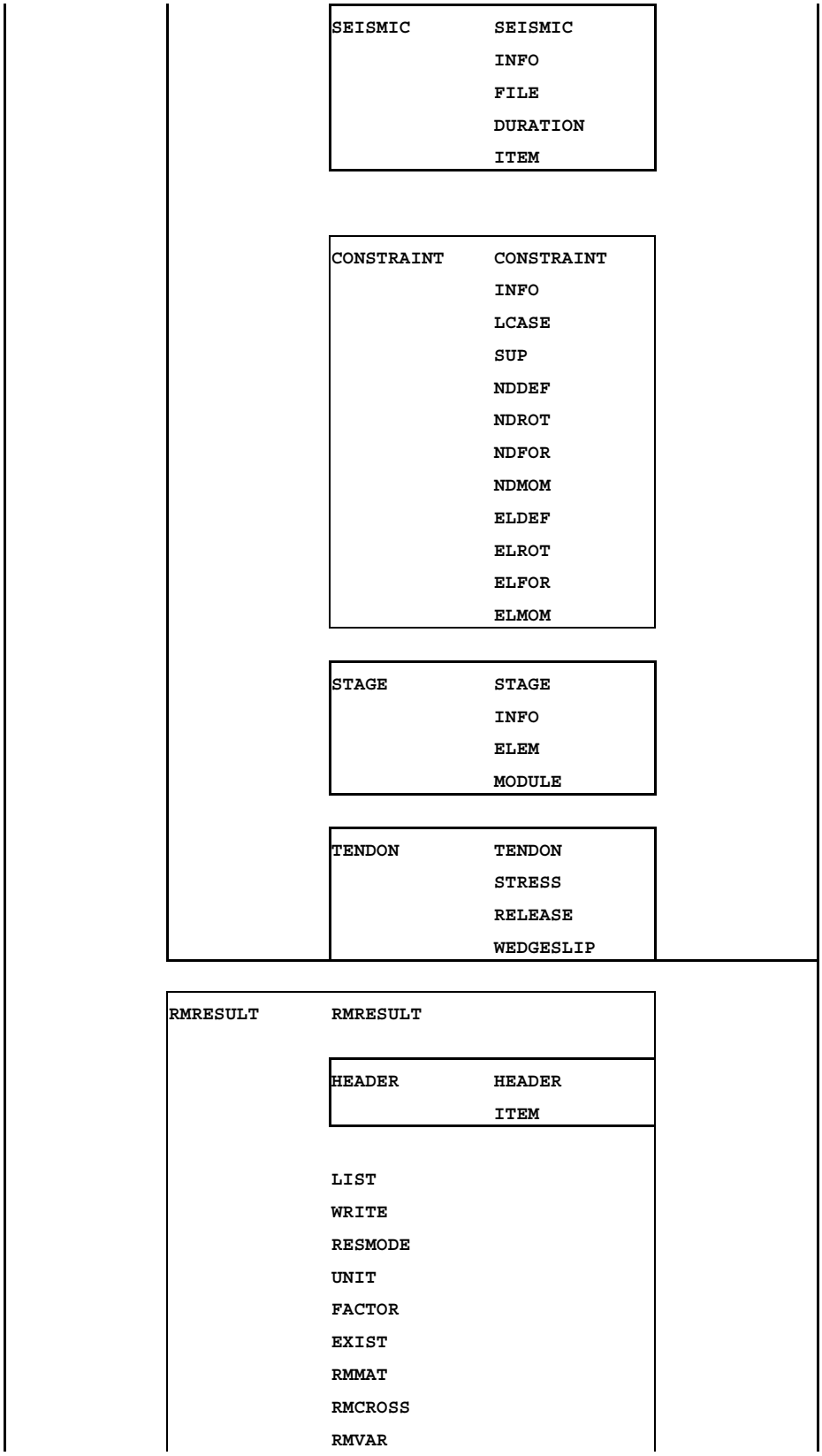
RM-specific commands are divided into different scopes.

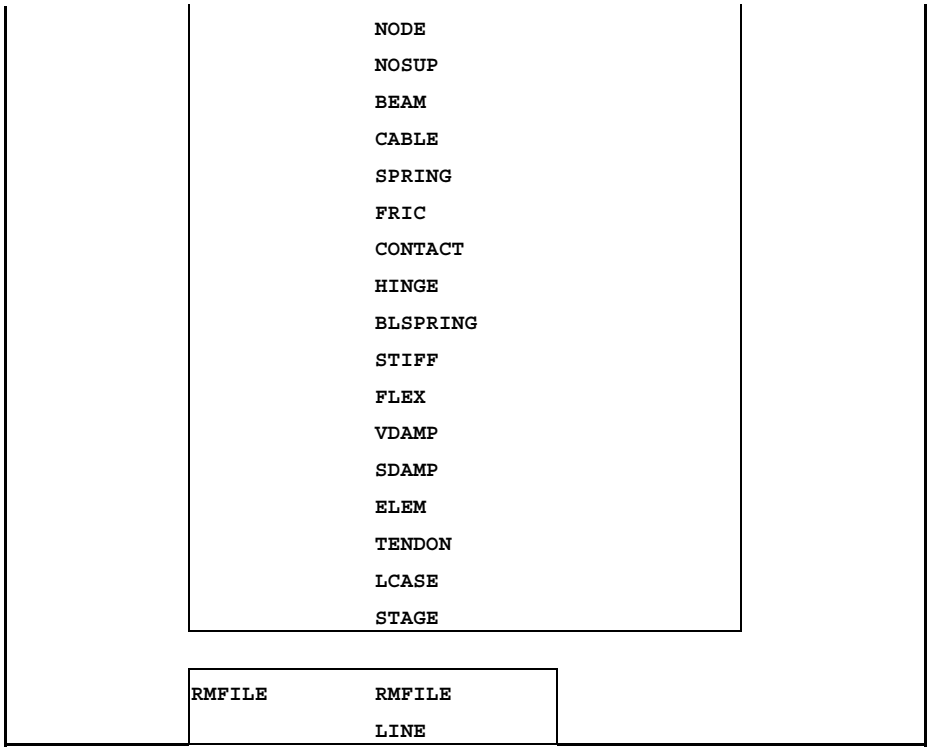
Within a scope, only part of the commands are available. Starting a TCL – Script from within RM2000, scope General is activated automatically. A general overview of scopes and validity of commands is give in the next picture:

SCOPELEVEL				
1	2	3	4	5
RMHALT	RMHALT	RMHALT	RMHALT	RMHALT
RMLOG	RMLOG	RMLOG	RMLOG	RMLOG
RMWARN	RMWARN	RMWARN	RMWARN	RMWARN
RMERROR	RMERROR	RMERROR	RMERROR	RMERROR
RMLANG	RMLANG	RMLANG	RMLANG	RMLANG
RMINPLANG	RMINPLANG	RMINPLANG	RMINPLANG	RMINPLANG
RMDATA	RMDATA	RMDATA	RMDATA	RMDATA
RMJOB	RMJOB			
	RMINFO	RMINFO		
		RMVERSION		
		TEXT		
		PROJDATE		
		STRUCTURE		
		ENVDISP		
		ENVFORCE		
		TENDRES		
		PERMLCTOT		
		NORM		
		LINEAR		
		NONLIN		
		SPECIAL		
		ANGLE		
		LENGTH		
		FORCE		
		MOMENT		
		STRESS		
		TEMP		
		TIME		
		TOL		
		G		
		NEWMARK		
		CRTIME		
		M_OVER_K		
		GLOBDAMP		
		LISTFACT		
		PAGE		
	RMUNIT	RMUNIT		
		ANGLE		
		LENGTH		
		FORCE		
		MOMENT		
		STRESS		


TEMP
TIME
RMMAT
RMMAT
INFO
DATA1 - 7
RMREINF
RMREINF
GROUP
RMCROSS
RMCROSS
INFO
NODE
ELEM
REINF
RMCROSS
RMCROSS
(COMPOSITE)
INFO
PARAMETER
ITEM
RMVAR
RMVAR
VAR
TABLE
TABLE
ITEM
RMSTRUCT
RMSTRUCT
NODE
NOSUP
BEAM
CABLE
SPRING
FRIC
CONTACT
HINGE
BLSPRING
STIFF
STIFF
K11
K12
K21
K22
FLEX
F11
F12
F21
F22


VDAMP
SDAMP
ELEM
TENDON
TENDON
INFO
MAT
AREA
COUNT
FRIC
NODEAT
STRESS
RMSCHED
RMSCHED
LCOMB
LCOMB
LCASE
LCASE
ENVELOPE
ENVELOPE
COMBINE
LMANAGE
LMANAGE
INFO
LCASE
ENVELOPE
LSET
LSET
INFO
ITEM
LCASE
LCASE
INFO
LMANAGE
LSET
LANE
LANE
INFO
ITEM
LTRAIN
LTRAIN
INFO
FMINMAX
EPS
FACT
ITEM


SEISMIC
SEISMIC
INFO
FILE
DURATION
ITEM
CONSTRAINT
CONSTRAINT
INFO
LCASE
SUP
NDDEF
NDROT
NDFOR
NDMOM
ELDEF
ELROT
ELFOR
ELMOM
STAGE
STAGE
INFO
ELEM
MODULE
TENDON
TENDON
STRESS
RELEASE
WEDGESLIP
RMRESULT
RMRESULT
HEADER
HEADER
ITEM
LIST
WRITE
RESMODE
UNIT
FACTOR
EXIST
RMMAT
RMCROSS
RMVAR



1.1 Scope: General

Starting a script from within RM2000, the following commands are available:

- **RMHALT:** Stop the execution of the script immediately.
- **RMLOG:** Put a message to the RM log.
- **RMWARNING:** Put a warning message to the RM log.
- **RMERROR:** Put an error massage to the RM log.
- **RMLANG:** Get the user language for output.
- **RMINPLANG:** Get the user language for input.
- **RMDATA:** Get part of general project data.
- **RMJOB:** Start a new RM Job.

The syntax for these commands are:

1.1.1 RMHALT:

Syntax	**RMHALT**
Meaning	No parameters, immediately stop script execution. An error message will appear in the RM error log.
Examples	**RMHALT**

1.1.2 RMLOG:

Syntax	**RMLOG "*message*"**
Meaning	Add a message to the RM log.
Examples	**RMLOG "Now start the definition of the bridge."**

1.1.3 RMWARN:

Syntax	**RMWARN "*warning-message*"**
Meaning	Add a message to the RM warning-messages.
Examples	**RMWARN "This warning will appear in the RM log"**

1.1.4 RMERROR:

Syntax	**RMERROR "*error-message*"**
Meaning	Add a message to the RM error-messages.
Examples	**RMERROR "This error will appear in the RM error-log"**

1.1.5 RMLANG:

Syntax	**RMLANG**
Meaning	Returns the selected language for ouput.
Examples	**RMLANG**

1.1.6 RMINPLANG:

Syntax	**RMINPLANG**
Meaning	Returns the selected language for input.
Examples	**RMINPLANG**

1.1.7 RMDATA:

Syntax	**RMDATA DEFLECTION** **RMDATA FORCE**
Meaning	Returns the deflection or force factor chosen in the result dialog.
Examples	**RMDATA DEFLECTION** **RMDATA FORCE**

1.1.8 RMJOB:

Syntax	**RMJOB BEGIN** **RMJOB END**
Meaning	Must be executed at the begin (BEGIN) and at the end (END) of a Script session. RMJOB provides access to the other commands.
Examples	**RMJOB BEGIN** **RMJOB END**

1.2 Scope: RMJOB

After the execution of **RMJOB BEGIN,** this scope is entered. Within this scope, the following commands are available:

- **RMHALT, RMLOG, RMWARNING, RMERROR** as described in chapter 1.1.

- **RMINFO:** Set general options.
- **RMUNIT:** Define units used in TCL file.
- **RMMAT:** Define or change material properties.
- **RMREINF:** Define or change reinforcement properties.
- **RMCROSS:** Define or change cross section properties.
- **RMVAR:** Define or change variables and tables.
- **RMSTRUCT:** Define or change structure.
- **RMSCHED:** Define or change a construction schedule.
- **RMRESULT:** Evaluate results.
- **RMFILE:** Define content of an ASCII file.

Each of these commands enter another scope. These commands are used to provide another set of commands valid in the scope.

The syntax for these commands are:

1.2.1 RMINFO:

Syntax	**RMINFO START** **RMINFO END**
Meaning	Start or end the definition of project parameters, RM-units and factors.
Examples	**RMINFO START** **RMINFO END**

1.2.2 RMUNIT:

Syntax	**RMUNIT START** **RMUNIT END**
Meaning	Set the units for the following TCL data. In order to change the default units in RM, define the units in the RMINFO scope!
Examples	**RMUNIT START** **RMUNIT END**

1.2.3 RMMAT:

Syntax	**RMMAT "*material-name*" *material-type*** **RMMAT END**
Meaning	Start or end the definition of material properties. The material is created if it does not exists yet. The "***material-type***" parameter must be one of the following: **CONC, STEEL, REINF, PRSTRS, WOOD, ALU, OTHER**
Examples	**RMMAT "concrete B50" CONC** **RMMAT END**

1.2.4 RMREINF:

Syntax	**RMREINF BEGIN** **RMREINF END**
Meaning	Start or end the definition of reinforcement groups.
Examples	**RMREINF BEGIN** **RMREINF END**

1.2.5 RMCROSS:

Syntax	**RMCROSS "*cross section name*"** **RMCROSS END**
Meaning	Start or end the definition of a cross section. **RMCROSS** creates a new cross section if cross section "***cross section name***" does not exist yet.
Examples	**RMCROSS "cs15"** **RMCROSS END**

1.2.6 RMCROSS COMPOSITE:

Syntax	**RMCROSS "*cross section name*" COMPOSITE** **RMCROSS END**
Meaning	Start or end the definition of a composite cross section. **RMCROSS** creates a new cross section if cross section "***cross section name***" does not exist yet.
Examples	**RMCROSS "compcs15" COMPOSITE** **RMCROSS END**

1.2.7 RMVAR:

Syntax	**RMVAR BEGIN** **RMVAR END**
Meaning	Start or end the definition of variables and tables.
Examples	**RMVAR BEGIN** **RMVAR END**

1.2.8 RMSTRUCT:

Syntax	**RMSTRUCT BEGIN** **RMSTRUCT END**
Meaning	Start or end the definition of the structure.
Examples	**RMSTRUCT BEGIN** **RMSTRUCT END**

1.2.9 RMSCHED:

Syntax	**RMSCHED BEGIN** **RMSCHED END**
Meaning	Start or end the definition of a construction schedule.
Examples	**RMSCHED BEGIN** **RMSCHED END**

1.2.10 RMRESULT:

Syntax	**RMRESULT BEGIN** **RMRESULT END**
Meaning	Start or end result evaluation. The result scope is entered automatically if the script was started by ⇧RESULT ⇨SCRIPT
Examples	**RMRESULT BEGIN** **RMRESULT END**

1.2.11 RMFILE:

Syntax	**RMFILE "filename.extension"** **RMRESULT END**
Meaning	Start or end the definition of an ASCII file. ATTENTION: Any existing file will be overwritten! Use with care.
Examples	**RMFILE "pl-lc1000.rm"** **RMFILE END**

1.3 Scope: RMINFO

After the execution of **RMINFO START,** this scope is entered. Within this scope, the following commands are available:

- **RMHALT, RMERROR, RMWARNING, RMLOG** as described in chapter 1.1.
- **RMINFO END:** To end this scope.

- **RMVERSION:** RM program version if file was created by export.
- **TEXT:** Set project description text(s).
- **PROJDATE:** Start date of construction.
- **STRUCTURE:** Set active degrees of freedom for structure.
- **ENVDISP:** Set min/max displacement for results.
- **ENVFORCE:** Set min/max forces for results.
- **TENDRES:** Define if tendon results should be saved.
- **PERMLCTOT:** Define the loadcase for the sum of permanent loads.
- **NORM:** Set the active norm.
- **LINEAR:** Set parameters for linear calculation.
- **NONLIN:** Set parameters for non-linear calculation.
- **SPECIAL:** Set special calculation parameters.
- **ANGLE:** Set the unit(s) for angles used in RM.
- **LENGTH:** Set the unit(s) for lengths used in RM.
- **FORCE:** Set the unit for forces used in RM.
- **MOMENT:** Set the unit for moments used in RM.
- **STRESS:** Set the unit for stresses used in RM.
- **TEMP:** Set the unit for temperature used in RM.
- **TIME:** Set the unit(s) for time-related values used in RM.
- **TOL:** Set convergence parameters.
- **G:** Define the gravity constant.
- **NEWMARK:** Set time integration constants.
- **M_OVER_K:** Set the tolerances for m/k.
- **GLOBDAMP:** Set the global damping factor for modal superposition.
- **CRTIME:** Set creeping and shrinking constants.

- **CROSSINT**: Set the parameters for cross section integration.
- **LISTFACT:** Set the multiplication factors for deformation and force output.
- **PAGE:** Set number of lines per page and first page number for listing.

The syntax for these commands are:

1.3.1 TEXT:

Syntax	**TEXT *"Text"***
Meaning	Set the project text. A maximum of 2 text lines is supported.
Examples	**TEXT "Moving load example"**

1.3.2 PROJDATE:

Syntax	**PROJDATE *yyyy – mm – dd***
Meaning	Set the construction start date. Year must be 4 digits.
Examples	**PROJDATE 2001 - 11 – 21**

1.3.3 STRUCTURE:

Syntax	**STRUCTURE *DOF1 DOF2 . . .***
Meaning	Set active degrees of freedom for structure. DOFx must be one of those: **Vx Vy Vz Px Py Pz**
Examples	**STRUCTURE Vx Vy Pz**

1.3.4 ENVDISP:

Syntax	**ENVDISP *DOF1 DOF2 …***
Meaning	Set min/max displacement for results. DOFx must be one of those: **Vx Vy Vz Px Py Pz**
Examples	**ENVDISP Vx Vy Pz**

1.3.5 ENVFORCE:

Syntax	**ENVFORCE** ***DOF1 DOF2 …***
Meaning	Set min/max displacement for results. DOFx must be one of those: **Nx Qy Qz Mx My Mz**
Examples	**ENVFORCE Nx Qy Mz**

1.3.6 TENDRES:

Syntax	**TENDRES LCASE** **TENDRES ALL**
Meaning	Save tendon results in loading case / loading case and superposition results.
Examples	**TENDRES ALL**

1.3.7 PERMLCTOT:

Syntax	**PERMLCTOT** ***number***
Meaning	Define the number of the loadcase containing all permanent loads.
Examples	**PERMLCTOT 1000**

1.3.8 NORM:

Syntax	**NORM** ***"Norm"***
Meaning	Set the active norm. ***"Norm"*** must be a supported norm name as it appears in the RM program (**OE-Norm(B4200), OE-Norm(B4700), DIN, DIN(18800,EC3), Portugal, Norw.Norm-NS, Jap.Norm-JIS, BS5400, AASHTO**)
Examples	**NORM "AASHTO"**

1.3.9 LINEAR:

Syntax	**LINEAR NOSHEAR**
Meaning	Define parameters for linear calculation: **NOSHEAR**: Ignore shear deformations.
Examples	**LINEAR NOSHEAR**

1.3.10 NONLIN:

Syntax	**NONLIN P_DELTA** **NONLIN STAY_CABLE** **NONLIN LARGE_DEFLECTION** **NONLIN MATERIAL** **NONLIN SPRING**
Meaning	Define parameters for non-linear calculation: **P_DELTA**: Calculate with P-Delta effect. **STAY_CABLE:** Calculate with non-linear cable behaviour. **LARGE_DEFLECTION:** Calculate with large deflection theory. **MATERIAL:** Calculate with non-linear material properties. **SPRING:** Calculate with non-linear springs/dampers.
Examples	**NONLIN LARGE_DEFLECTION**

1.3.11 SPECIAL:

Syntax	**SPECIAL PERMLOAD** **SPECIAL STAGE_CONSTRAINTS** **SPECIAL STIFFNESS_ACC** **SPECIAL CSUPDATE_STEELAREA** **SPECIAL CSUPDATE_DUCTAREA** **SPECIAL CSUPDATE_FILLAREA** **SPECIAL EMOD_CREEP** **SPECIAL PRIMARY_TEMPVAR** **SPECIAL PRINT_CSFACT** **SPECIAL PARTFORCE_CREEP** **SPECIAL CSSHEAR_CALCAREA** **SPECIAL TDV_SUPERPOSITION**
Meaning	Define special calculation parameters: **PERMLOAD**: Accumulate permanent load in structure. **STAGE_CONSTRAINTS:** Apply construction stage constraints in structure. **STIFFNESS_ACC:** Accumulate stiffness from LCSum. **CSUPDATE_STEELAREA:** Update cross section by adding steel area. **CSUPDATE_DUCTAREA:** Update cross section by subtracting duct area. **CSUPDATE_FILLAREA:** Update cross section by adding fill area. **EMOD_CREEP:** Update E-modulus by creep **PRIMARY_TEMPVAR:** Create primary state due to temp variable. **PRINT_CSFACT:** Print creeping and shrinking factor. **PARTFORCE_CREEP:** Store part. forces due to creep. **CSSHEAR_CALCAREA:** Calculate shear area for cross sections. **TDV_SUPERPOSITION:** Use TDV superposition method.
Examples	**SPECIAL CSUPDATE_STEELAREA**

1.3.12 ANGLE, LENGTH, FORCE, MOMENT, STRESS, TEMP, TIME:

Syntax	As described in chapter 1.4 (RMUNIT)
Meaning	Set the units used within the interactive RM2000 – program. THIS PART DOES NOT DEFINE THE UNITS USED WITHIN THE TCL-FILE!
Examples	See chapter 1.4 (RMUNIT)

1.3.13 TOL:

Syntax	**TOL** ***Relax N-Iter Tol-1 Tol-2 Tol-3 Tol-4***
Meaning	Define convergence parameters: ***Relax***: Relaxation factor in the Newton-Raphson iteration. ***N-Iter***: Minimum number of iteration. ***Tol-1 .. Tol-4***: Force and deflection limits.
Examples	`TOL 1 40 0.005 0.005 0.00005 0.000015`

1.3.14 G:

Syntax	**G** ***gravity***
Meaning	Define gravity constant. This constant is given in LENGTH STRUCT / s2!
Examples	`G 9.80665`

1.3.15 NEWMARK:

Syntax	**NEWMARK** ***dt c1 c2 Alfa Beta***
Meaning	Define constants for dynamic calculation: ***dt:*** Time increment for NEWMARK time integration. ***c1,c2,Alfa,Beta:*** Constants for NEWMARK time integration.
Examples	`NEWMARK 0.01 0.5 0.25 0 0`

1.3.16 CRTIME:

Syntax	**CRTIME** ***factor*** **LINEAR** **CRTIME** ***factor*** **LOG**
Meaning	Define interpolation type and factor for creeping and shrinking calculations.
Examples	`CRTIME .5 LOG`

1.3.17 M_OVER_K:

Syntax	**M_OVER_K** ***tol***
Meaning	Set the tolerance for m_i/k_i
Examples	**M_OVER_K 0**

1.3.18 GLOBDAMP:

Syntax	**GLOBDAMP** ***dampfactor***
Meaning	Set the global damping factor for modal superposition.
Examples	**GLOBDAMP 0**

1.3.19 CROSSINT:

Syntax	**CROSSINT** ***iter reclevel incr relax tolerance bits***
Meaning	Set iteration count, recursion level etc. for cross section calculations.
Examples	**CROSSINT 500 2 0.25 0.2 1e-6 2**

1.3.20 LISTFACT:

Syntax	**LISTFACT** ***deflectionfactor forcefactor***
Meaning	Set the multiplication factors for deflection and force list-output.
Examples	**LISTFACT 1e3 10**

1.3.21 PAGE:

Syntax	**PAGE** ***nlines firstpage***
Meaning	Set the number of lines per page and number of the first page for list files.
Examples	**PAGE 65 1**

1.4 Scope: RMUNIT

After the execution of **RMUNIT START,** this scope is entered. Within this scope, the following commands are available:

- **RMHALT, RMERROR, RMWARNING, RMLOG** as described in chapter 1.1.
- **RMUNIT END:** To end this scope.

- **ANGLE:** Define the unit(s) for angles.
- **LENGTH:** Define the unit(s) for lengths.
- **FORCE:** Define the unit for forces.
- **MOMENT:** Define the unit for moments.

- **STRESS:** Define the unit for stresses.
- **TEMP:** Define the unit for temperature.
- **TIME:** Define the unit(s) for time-related values.

The syntax for these commands are:

1.4.1 ANGLE:

Syntax	**ANGEL STRUCT *stdangleunit*** **ANGEL RESULT *stdangleunit***
Meaning	Define the angle unit for structure or for results. "***stdangleunit***" must be one of the following: **DEGREE GRAD RAD**
Examples	**ANGEL STRUCT DEGREE** **ANGEL RESULT RAD**

1.4.2 LENGTH:

Syntax	**LENGTH STRUCT *stdlengthunit*** **LENGTH STRUCT *userunit value*** **LENGTH CROSS *stdlengthunit*** **LENGTH CROSS *userunit value***
Meaning	Define the unit for length – values in structure or cross section. "***stdlengthunit***" must be one of the following: **MM CM M IN FT YD** If a user-unit is used, an additional parameter must provide the multiplication factor to get **M.**
Examples	**LENGTH CROSS IN** **LENGTH STRUCT "DM" 0.1**

1.4.3 FORCE:

Syntax	**FORCE *stdforceunit*** **FORCE *userunit value***
Meaning	Define the unit for all force – values. "***stdforceunit***" must be one of the following: **N KN MN T LB KIP** If a non-standard unit is used, an additional parameter must provide the multiplication factor to get **KN.**
Examples	**FORCE KIP** **FORCE MILLINEWTON 0.000001**

1.4.4 MOMENT:

Syntax	**MOMENT *stdforceunit*stdlengthunit*** **MOMENT *userunit value***
Meaning	Define the unit for all force – values. ***stdforceunit*** and ***stdlengthunit*** must match standard units described in 1.4.3 and 1.4.2. If a non-standard unit is used, the user must provide the multiplication factor to get **KN*M.**
Examples	**MOMENT KIP*YD** **MOMENT KN*DM 0.1**

1.4.5 STRESS:

Syntax	**STRESS *stdforceunit/stdlengthunit2*** **STRESS *userunit value***
Meaning	Define the unit for all stress – values. ***stdforce*** and ***stdlength*** must match standard units described in 1.4.3 and 1.4.2. If a non-standard unit is used, the user must provide the multiplication factor to get **KN/M2.**
Examples	**STRESS KIP/IN2** **STRESS KN/DM2 0.01**

1.4.6 TEMP:

Syntax	**TEMP *stdtempunit***
Meaning	Define the unit temperatures used. ***stdtempunit*** must be one of the following: **CELSIUS, FAHRENHEIT**
Examples	**TEMP CELSIUS** **TEMP FAHRENHEIT**

1.4.7 TIME:

Syntax	**TIME SCHEDULE *stdtimeunit*** **TIEM LOAD *stdtimeunit***
Meaning	Define the unit for time – values for the construction schedule and loads (e.g. seismic loads). ***stdtimeunit*** must be one of the following: **YEAR MONTH DAY HOUR MINUTE SECOND** This option is not implemented yet!
Examples	**TIME SCHEDULE DAY** **TIEM LOAD SECOND**

1.5 Scope: RMMAT

After the execution of **RMMAT "*material-name*" *material-type*,** this scope is entered. The "material-type" parameter must be one of the following: **CONC, STEEL, REINF, PRSTRS, WOOD, ALU, OTHER.** Within this scope, the following commands are available:

- **RMHALT, RMERROR, RMWARNING, RMLOG** as described in chapter 1.1.
- **RMMAT END:** To end this scope.

- **INFO:** Define a description for the material.
- **DATA1:** Define **E_Modl, E-Modt, G_Mod, ALFA-T, Gamma, SIG-ZY, CONC, Z-TYP, E_Modex, SIG-allow-pr, SIG-allowSA.**
- **DATA2:** Define **Sigma-F, Sigma-F*, Sigm-V1, Sigm-V2, Sigma1, TAU1, NY*1, Sigma2, TAU2, NY*2, Gamma1, Gamma2, Gamma3.**
- **DATA3:** Define **Sig-p, W28, SIG-allow-ch, SIG-allow-m, TAU1, TAU2, TAU3, TAUB.**
- **DATA4:** Define **SIG-X ZUB, SIG-X DRB, SIG-X ZUH, SIG-X DRH, SIG-X ZUHZ, SIG-X DRHZ** for both full prestress and part. prestress… .
- **DATA5:** Define **SIGG-Q, SIGG-Q+MT, SIGX, SIG1-Q-ST, SIG1-Q-PL, SIG1-MT, SIG1-Q+MT, SIG2-Q+MT, SIG2-G, SIG1-ST, SIG1-PL, SIG1-Q+MT-M.**
- **DATA6:** Define **EPS-PL, EPL-*, SIG-0.2, SIGMA*, SIG-0.2/E, Eps-1, Sig-1, … Eps-8, SIG-8, SIG-ZUS, Xl, WCR, CECO, GAMMA.**
- **DATA7:** Define **PHI(t), EPS(t), RHO(t), EMOD(t).**

The syntax for these commands are:

1.5.1 INFO:

Syntax	**INFO "*message*"**
Meaning	Define a description for the material.
Examples	**INFO "This is the high quality concrete material."**

1.5.2 DATA1:

Syntax	**DATA1 *value1 value2 … value11***
Meaning	Define **E_Modl, E-Modt, G_Mod, ALFA-T, Gamma, SIG-ZY, CONC, Z-TYP, E_Modex, SIG-allow-pr, SIG-allowSA.**
Examples	**DATA1 4.4e+007 0 1.84e+007 1e-005 25 0 0 0 0 0 0**

1.5.3 DATA2:

Syntax	**DATA2** ***value1 value2 … value13***
Meaning	Define **Sigma-F, Sigma-F*, Sigm-V1, Sigm-V2, Sigma1, TAU1, NY*1, Sigma2, TAU2, NY*2, Gamma1, Gamma2, Gamma3.**
Examples	**DATA2 24 0 18 19.2 14 9.2 1.5 16 10.4 2.5 0 0 0**

1.5.4 DATA3:

Syntax	**DATA3** ***value1 value2 … value8***
Meaning	Define **Sig-p, W28, SIG-allow-ch, SIG-allow-m, TAU1, TAU2, TAU3, TAUB.**
Examples	**DATA3 0 60 0 0 0 0 0 0**

1.5.5 DATA4:

Syntax	**DATA4** ***value1 value2 … value12***
Meaning	Define **SIG-X ZU1, SIG-X DR1 ... SIG-X ZU6, SIG-X DR6.**
Examples	**DATA4 0 0.05 0 0 0 0 0 0 0 0 0 0**

1.5.6 DATA5:

Syntax	**DATA5** ***value1 value2 … value13***
Meaning	Define **SIGG-Q, SIGG-Q+MT, SIGX, SIG1-Q-ST, SIG1-Q-PL, SIG1-MT, SIG1-Q+MT, SIG2-Q+MT, SIG2-G, SIG1-ST, SIG1-PL, SIG1-MT-M, SIG1-Q+MT-M.**
Examples	**DATA5 0 0 0 0 0 0 0 0 0 0 0 0 0**

1.5.7 DATA6:

Syntax	**DATA6** ***value1 value2 … value23***
Meaning	Define **EPS-PL, EPL-*, SIG-0.2, SIGMA*, SIG-0.2/E, Eps-1, Sig-1, … Eps-8, SIG-8, SIG-ZUS, Xl, WCR, CECO, GAMMA.**
Examples	**DATA6 0 5 9 4 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**

1.5.8 DATA7:

Syntax	**DATA7 "*form1*" "*form2*" "*form3*" "*form4*"**
Meaning	Define **PHI(t), EPS(t), RHO(t), EMOD(t).**
Examples	**DATA7 "C90cr" "C90sh" "" ""**

1.6 Scope: RMREINF

After the execution of **RMREINF BEGIN,** this scope is entered. Within this scope, the following commands are available:

- **RMHALT, RMERROR, RMWARNING, RMLOG** as described in chapter 1.1.
- **RMREINF END:** To end this scope.

- **GROUP:** Define a reinforcement group.

The syntax for these commands are:

1.6.1 GROUP:

Syntax	**GROUP "*name*"** **GROUP "*name*" *stressgroup*** **GROUP "*name*" *stressgroup* "*materialname*"** **GROUP "*name*" *stressgroup* "*materialname*" "*info*"** **GROUP "name" DATA *maxD reimax reiminA reiminF***
Meaning	Define a reinforcement or stress group or assign parameters to an existing group. If stressgroup is omitted, stressgroup 1 is used. maxD is the limit diameter for CracChk reimax is the maximal reinforcement area per reinforcemenet length reinimA and reinminF define the minimum reinforcement for this group as area = reiminA + reinminF * area of cross section
Examples	**GROUP "upper"** **GROUP "stress2" 2** **GROUP "lower" 1 "B_35"** **GROUP "all" 1 "B_35" "Info for this group"** **GROUP "upper" 0.016 0.1 0.001 0.005**

1.7 Scope: RMCROSS

After the execution of **RMCROSS "*cross section name*",** this scope is entered. Within this scope, the following commands are available:

- **RMHALT, RMERROR, RMWARNING, RMLOG** as described in chapter 1.1.
- **RMCROSS END:** To end this scope.

- **INFO:** Define a description for the cross section.
- **NODE:** Define cross section nodes.

- **ELEM:** Define cross section elements.
- **REINF:** Define reinforcement or stress check points.

The syntax for these commands are:

1.7.1 INFO:

Syntax	**INFO "*message*"**
Meaning	Define a description for the cross section.
Examples	**INFO "This is the middle part cross section."**

1.7.2 NODE:

Syntax	**NODE *number z y***
Meaning	Define a cross section node with coordinates 'z' and 'y'.
Examples	**NODE 17 -4.56 2.50**

1.7.3 ELEM:

Syntax	**ELEM *number node1 node2 node3 node4 ... node9*** **ELEM *number* SHEARY *node1 node2 node3 node4 ... node9*** **ELEM *number* SHEARZ *node1 node2 node3 node4 ... node9*** **ELEM *number* FACTOR *factor-Mx factor-Qy factor-Qz***
Meaning	Define an element by 9 nodes. Add shear information to element. Add factors for shear calculation to element.
Examples	**ELEM 1 1 2 3 4 5 6 7 8 9** **ELEM 2 SHEARZ 2 4 8 6 10 12 14 16 18** **ELEM 2 FACTOR 1.0 0.8 0.8**

1.7.4 ADDPOI:

Syntax	**ADDPOI *ptype "grp" no1 no2 dz angle no1 no2 dy angle "desc"*** **ADDPOI TMPPOI *"grp" n1 n2 dz angle no1 no2 dy angle "desc"* t**
Meaning	Define an additional point (e.g. reinforcement or stress check point). "ptype" must be a valid keyword as described in RM2000. **desc** is a (short) description (name, max., 15 characters)
Examples	**ADDPOI FIBPOI "" 40 60 0 90 77 109 0 90 "RP_1"** **ADDPOI TMPPOI "" 40 60 0 90 77 109 0 90 "RP_1" 32.4**

1.8 Scope: RMCROSS COMPOSITE

After the execution of **RMCROSS "*cross section name*" COMPOSITE,** this scope is entered. Within this scope, the following commands are available:

- **RMHALT, RMERROR, RMWARNING, RMLOG** as described in chapter 1.1.

- **RMCROSS END:** To end this scope.

- **INFO:** Define a description for the composite cross section.
- **PARAMETER:** Define composite cross section parameter.
- **ITEM:** Define cross section parts.

The syntax for these commands are:

1.8.1 INFO:

Syntax	**INFO "*message*"**
Meaning	Define a description for the cross section.
Examples	**INFO "This is a composite cross section."**

1.8.2 PARAMETER:

Syntax	**PARAMETER SYMMETRIC *n1 n2 n3*** **PARAMETER ASYMMETRIC *n1 n2 n3***
Meaning	Define composite cross section parameter for symmetrical / asymmetrical cross section and number of division in different directions.
Examples	**PARAMETER SYMMETRIC *2 2 3***

1.8.3 ITEM:

Syntax	**ITEM "*crossname*" "*materialname*"**
Meaning	Define parts of composite cross section with materials. The cross section "crossname" and the material must already exist in the database!
Examples	**ITEM "*cross1*" "*steel*"**

1.9 Scope: RMVAR

After the execution of **RMVAR BEGIN,** this scope is entered. Within this scope, the following commands are available:

- **RMHALT, RMERROR, RMWARNING, RMLOG** as described in chapter 1.1.
- **RMVAR END:** To end this scope.

- **VAR:** Define variables.
- **TABLE:** Begin table definition (subscope).

The syntax for these commands are:

1.9.1 VAR:

Syntax	**VAR "*variable name*" "*formula*" "*description*"**
Meaning	Define a variable.
Examples	**VAR pi 3.14159 "half circumference of circle with r=1"** **VAR "pi2" "2 * pi" "circumference of circle with r=1"**

1.9.2 Sub-scope: TABLE:

Syntax	**TABLE "*name*" "*description*"**
Meaning	Define a table. The following ITEM command defines table items for this table.
Examples	**TABLE "table 1" "some description"**

After the execution of **TABLE "name",** the table sub-scope is entered. Within this scope, the following commands are available:

- **RMHALT, RMERROR, RMWARNING, RMLOG** as described in chapter 1.1.
- **TABLE END:** To end this scope.

- **ITEM:** Define table entries.

The syntax for these commands are:

1.9.2.1 ITEM:

Syntax	**ITEM "*formula a*" "*formula b*" *interpolationtype***
Meaning	Define a table item. The ***interpolationtype*** defines the interpolation type and must be one of those: **CONST, LINEAR, PAR0, PAR1, PAR2.**
Examples	**ITEM "pi * 1.05" "pi * 1.1" CONST** **ITEM "0.7" "1.0" LINEAR**

1.10 Scope: RMSTRUCT

After the execution of **RMSTRUCT BEGIN,** this scope is entered. Within this scope, the following commands are available:

- **RMHALT, RMERROR, RMWARNING, RMLOG** as described in chapter 1.1.
- **RMSTRUCT END:** To end this scope.

- **NODE:** Define or change nodes.
- **NOSUPP:** Define or change node support.

- **BEAM:** Define or change beams
- **CABLE:** Define or change cables.
- **SPRING:** Define or change static springs.
- **FRIC:** Define or change friction springs.
- **CONTACT:** Define or change contact springs.
- **HINGE:** Define or change hinge springs.
- **BLSPRING:** Define or change bilinear springs.
- **STIFF:** Define elements by stiffness matrix.
- **FLEX:** Define elements by flexibility matrix.
- **ELEM:** Define or change general parameters for elements.
- **VDAMP:** Define or change viscous dampers.
- **SDAMP:** Define or change damper springs.
- **TENDON:** Define tendon geometry.

The syntax for these commands are:

1.10.1 NODE:

Syntax	**NODE *node-number x y z*** **NODE *node-number x y*** **NODE *node-number x***
Meaning	Define a node by three-dimensional coordinates. If the z-coordinate or the y- and the z-coordinate are omitted, they are set to 0.0.
Examples	**NODE 17 10.0 15.0 1.4** **NODE 20 5.0 12.0** **NODE 1 0.0**

The [SERIE from to step] command can be used for node numbers and coordinates.

Syntax	**NODE *node-number* DELETE**
Meaning	Delete node with number “node-number”
Examples	**NODE 20 DELETE**

The [SERIE from to step] command can be used for node numbers.

Syntax	**NODE *node-number* MASS *Mx My Mz Imx Imy Imz*** **NODE *node-number* MASS *Mx My Mz Imx Imy*** **NODE *node-number* MASS *Mx My Mz Imx*** **NODE *node-number* MASS *Mx My Mz*** **NODE *node-number* MASS *Mx My*** **NODE *node-number* MASS *Mx***
Meaning	Assign mass values to a node. If any value is omitted, it is set to 0.0
Examples	**NODE 17 MASS 9.81 9.81 9.81 0.024 0.023 0.024** **NODE 18 MASS 9.81 9.81 9.81**

The [SERIE from to step] command can be used for node numbers.

1.10.2 NOSUP:

Syntax	**NOSUP *node-number* VALUE *Cx Cy Cz CMx CMy CMz***
Meaning	Define node support stiffness values.
Examples	**NOSUP 20 VALUE**

The [SERIE from to step] command can be used for node numbers.

Syntax	**NOSUP *node-number* ECC *ex ey ez***
Meaning	Define node support eccentricity.
Examples	**NOSUP 20 ECC**

The [SERIE from to step] command can be used for node numbers.

Syntax	**NOSUP *node-number* BETA *beta alpha1 alpha2***
Meaning	Define node support orientation.
Examples	**NOSUP 20 BETA**

The [SERIE from to step] command can be used for node numbers.

1.10.3 BEAM:

Syntax	**BEAM *element-number node-number1 node-number2*** **BEAM *element-number node-number***
Meaning	Define a beam between node-number1 and node-number2. If node-number2 is omitted, it is assumed to be node-number1 + 1.
Examples	**BEAM 1 1 5** **BEAM 2 2**

The [SERIE from to step] command can be used for all numbers.

Syntax	**BEAM *element-number* DELETE**
Meaning	Delete element with number “element-number”
Examples	**BEAM 2 DELETE**

The [SERIE from to step] command can be used for element numbers.

Syntax	**BEAM *element-number* MAT "*material name*"**
Meaning	Set the material of element with number element-number. The material must exist.
Examples	**BEAM 3 MAT "reinforced concrete"**

The [SERIE from to step] command can be used for element numbers.

Syntax	**BEAM *element-number* MATVAL "*E*" "*G*" "*Gamma*" "*Alpha-T*"**
Meaning	Set the material values of element with number element-number. Angles must be given in degree (360°).
Examples	**BEAM 3 MATVAL 2.1e8 8.75e7 78 1.2e-5**

The [SERIE from to step] command can be used for element numbers.

Syntax	**BEAM *element-number* CROSS *ecctype* "*cs 1*" "*cs 2*"** **BEAM *element-number* CROSS *ecctype* "*cs 1*"**
Meaning	Assign cross sections to element with number element-number. If a second cross section name is omitted, the same cross section is assigned to both ends of the element. ***ecctype*** defines the type of cross section eccentricity. There can be no eccentricity (o, lower case "O"), local eccentricity (l, lower case "L") and global eccentricity (g, lower case "G") for both coordinates (Y and Z). The forth parameter can, therefore, have one the following values: **YoZo, YoZl, YoZg, YlZo, YlZl, YlZg, YgZo, YgZl, YgZg**
Examples	**BEAM 3 CROSS YoZg "cross section 1" "cross section 2"** **BEAM 4 CROSS YlZo "cross section 5"**

The [SERIE from to step] command can be used for element numbers.

Syntax	**BEAM *element-number* CROSSVAL *Ax Ay Az Ix Iy Iz***
Meaning	Set the values for the cross section for element with number element-number.
Examples	**BEAM 3 CROSSVAL 1.493e-2 0 0 1.917e-6 8.564e-5 2.521e-4**

The [SERIE from to step] command can be used for element numbers.

Syntax	**BEAM *element-number* REINF "*reinf-group*" *x1/l x2/l area* FIX** **BEAM *element-number* REINF "*reinf-group*" *x1/l x2/l area* VAR**
Meaning	Define reinforcement longitudinal geometry and area.
Examples	**BEAM 3 REINF "upper" 0.0 0.8 0.02 FIX** **BEAM 3 REINF "lower" 0.5 1.0 0.03 VAR**

The [SERIE from to step] command can be used for element numbers.

Syntax	**BEAM *element-number* PERIMETER *outerlen innerlen***
Meaning	Set the values for perimeter of the outer and inner cross section shape.
Examples	**BEAM 3 PERIMETER 1.07 1.03**

The [SERIE from to step] command can be used for element numbers.

Syntax	**BEAM *element-number* NPART *n***
Meaning	Set the number of parts for element with number element-number.
Examples	**BEAM 3 NPART 5**

The [SERIE from to step] command can be used for element numbers.

Syntax	**BEAM *element-number* COMP *part1 part2 part3 part4*** **BEAM *element-number* COMP *part1 part2 part3*** **BEAM *element-number* COMP *part1 part2***
Meaning	Define a composite part consisting of elements part1, parts, part3 and part4. If less than 4 parts are given, the composite part consists of less parts.
Examples	**BEAM 5 COMP 1 2 3 4** **BEAM 6 COMP 1 2 3** **BEAM 7 COMP 1 2**

The [SERIE from to step] command can be used for all element numbers.

1.10.4 CABLE:

Syntax	**CABLE *element-number node-number1 node-number2*** **CABLE *element-number node-number***
Meaning	Define a cable between node-number1 and node-number2. If node-number2 is omitted, it is assumed to be node-number1 + 1.
Examples	**CABLE 1 1 5** **CABLE 2 2**

The [SERIE from to step] command can be used for all numbers.

Syntax	**CABLE *element-number* DELETE**
Meaning	Delete element with number “element-number”
Examples	**CABLE 2 DELETE**

The [SERIE from to step] command can be used for element numbers.

Syntax	**CABLE *element-number* MAT "*material name*"**
Meaning	Set the material of element with number element-number.
Examples	**CABLE 3 MAT "st52"**

The [SERIE from to step] command can be used for element numbers.

Syntax	**CABLE *element-number* MATVAL "*E*" "*G*" "*Gamma*" "*Alpha-T*"**
Meaning	Set the material values of element with number element-number. Angles must be given in degree (360°).
Examples	**CABLE 3 MATVAL 2.1e8 8.75^{e}7 78 1.2e-5**

The [SERIE from to step] command can be used for element numbers.

Syntax	**CABLE *element-number* CROSS *ecctype* "*cs 1*" "*cs 2*"** **CABLE *element-number* CROSS *ecctype* "*cross section name*"**
Meaning	Assign cross sections to element with number element-number. If a second cross section name is omitted, the same cross section is assigned to both ends of the element. ***ecctype*** defines the type of cross section eccentricity. There can be no eccentricity (o, lower case "O"), local eccentricity (l, lower case "L") and global eccentricity (g, lower case "G") for both coordinates (Y and Z). The forth parameter can, therefore, have one the following values: **YoZo, YoZl, YoZg, YlZo, YlZl, YlZg, YgZo, YgZl, YgZg**
Examples	**CABLE 3 CROSS YoZg "cross section 1" "cross section 2"** **CABLE 4 CROSS YlZo "cross section 2"**

The [SERIE from to step] command can be used for element numbers.

Syntax	**CABLE *element-number* CROSSVAL *Ax Ay Az Ix Iy Iz***
Meaning	Set the values for the cross section for element with number element-number.
Examples	**CABLE 3 3 CROSSVAL 1.493e-2 0 0 1.917e-6 8.564e-5 2.521e-4**

The [SERIE from to step] command can be used for element numbers.

Syntax	**CABLE *element-number* PERIMETER *outerlen innerlen***
Meaning	Set the values for perimeter of the outer and inner cross section shape.
Examples	**CABLE 3 PERIMETER 1.07 1.03**

The [SERIE from to step] command can be used for element numbers.

Syntax	**CABLE *element-number* NPART *n***
Meaning	Set the number of parts for element with number element-number.
Examples	**CABLE 3 NPART 5**

The [SERIE from to step] command can be used for element numbers.

1.10.5 SPRING:

Syntax	**SPRING *element-number node-number1 node-number2*** **SPRING *element-number node-number***
Meaning	Define a spring between node-number1 and node-number2. If node-number2 is omitted, it is assumed to be node-number1+1.
Examples	**SPRING 1 1 5** **SPRING 2 2**

The [SERIE from to step] command can be used for all numbers.

Syntax	**SPRING *element-number* DELETE**
Meaning	Delete element with number “element-number”
Examples	**SPRING 2 DELETE**

The [SERIE from to step] command can be used for element numbers.

Syntax	**SPRING *element-number* VALUE *CVx CVy CVz CRx CRy CRz***
Meaning	Set the spring values for element with number element-number.
Examples	**SPRING 3 VALUE 1e8 1e5 1e5 1e8 1e8 1e8**

The [SERIE from to step] command can be used for element numbers.

1.10.6 FRIC:

Syntax	**FRIC *element-number node-number1 node-number2*** **FRIC *element-number node-number***
Meaning	Define a friction spring between node-number1 and node-number2. If node-number2 is omitted, it is assumed to be node-number1+1.
Examples	**FRIC 1 1 5** **FRIC 2 2**

The [SERIE from to step] command can be used for all numbers.

Syntax	**FRIC *element-number* DELETE**
Meaning	Delete element with number “element-number”
Examples	**FRIC 2 DELETE**

The [SERIE from to step] command can be used for element numbers.

Syntax	**FRIC *element-number* VALUE *Ax ny nz Ix Iy Iz***
Meaning	Set the spring values for element with number element-number.
Examples	**FRIC 3 VALUE 0.014 0.01 0.01 10 8.5 13**

The [SERIE from to step] command can be used for element numbers.

1.10.7 CONTACT:

Syntax	**CONTACT *element-number node-number1 node-number2*** **CONTACT *element-number node-number***
Meaning	Define a contact spring between node-number1 and node-number2. If node-number2 is omitted, it is assumed to be node-number1+1.
Examples	**CONTACT 1 1 5** **CONTACT 2 2**

The [SERIE from to step] command can be used for all numbers.

Syntax	**CONTACT *element-number* DELETE**
Meaning	Delete element with number “element-number”
Examples	**CONTACT 2 DELETE**

The [SERIE from to step] command can be used for element numbers.

Syntax	**CONTACT *element-number* VALUE *CVx CVy CVz CRx CRy CRz***
Meaning	Set the spring values for element with number element-number. Values are defined by formulas.
Examples	**CONTACT 3 VALUE "CVxF" "CVyF" "CVzF" "CRxF" "CRyF" "CRzF"**

The [SERIE from to step] command can be used for element numbers.

1.10.8 HINGE:

Syntax	**HINGE *element-number node-number1 node-number2*** **HINGE *element-number node-number***
Meaning	Define a hinge spring between node-number1 and node-number2. If node-number2 is omitted, it is assumed to be node-number1+1.
Examples	**HINGE 1 1 5** **HINGE 2 2**

The [SERIE from to step] command can be used for all numbers.

Syntax	**HINGE *element-number* DELETE**
Meaning	Delete element with number “element-number”
Examples	**HINGE 2 DELETE**

The [SERIE from to step] command can be used for element numbers.

Syntax	**HINGE *element-number* CROSS *ecctype “cs 1” “cs 2”*** **HINGE *element-number* CROSS *ecctype “cs 1”***
Meaning	Assign composite cross sections to hinge spring with number element-number. If a second cross section name is omitted, the same cross section is assigned to both ends of the element.
Examples	**HINGE 3 CROSS YoZo “composite_cs1” “composite_cs2”** **HINGE 4 CROSS YoZo “composite_cs”**

The [SERIE from to step] command can be used for element numbers.

Syntax	**HINGE *element-number CVx CVy CVz CRx CRy CRz***
Meaning	Define spring values for hinge springs.
Examples	**HINGE 2 VALUE 1e8 1e8 0 0 1e8 0**

The [SERIE from to step] command can be used for element numbers.

1.10.9 BLSPRING:

Syntax	**BLSPRING *element-number node-number1 node-number2*** **BLSPRING *element-number node-number***
Meaning	Define a bilinear spring between node-number1 and node-number2. If node-number2 is omitted, it is assumed to be node-number1+1.
Examples	**BLSPRING 1 1 5** **BLSPRING 2 2**

The [SERIE from to step] command can be used for all numbers.

Syntax	**BLSPRING *element-number* DELETE**
Meaning	Delete element with number “element-number”
Examples	**BLSPRING 2 DELETE**

The [SERIE from to step] command can be used for element numbers.

Syntax	**BLSPRING *element-number* VALUE CVx CVx-*value* *Fx-limit*** **BLSPRING *element-number* VALUE LOADED CVy CVy-*value* *Qy-yield*** **BLSPRING *element-number* VALUE LOADED CVz CVz-*value* *Qz-yield*** **BLSPRING *element-number* VALUE LOADED CRx CRx-*value*** **BLSPRING *element-number* VALUE LOADED CRy CRy-*value*** **BLSPRING *element-number* VALUE LOADED CRz CRz-*value*** **BLSPRING *element-number* VALUE UNLOADED CVy CVy-*value* *Qy-yield*** **BLSPRING *element-number* VALUE UNLOADED CVz CVz-*value* *Qz-yield*** **BLSPRING *element-number* VALUE UNLOADED CRx CRx-*value*** **BLSPRING *element-number* VALUE UNLOADED CRy CRy-*value*** **BLSPRING *element-number* VALUE UNLOADED CRz CRz-*value***
Meaning	Set the spring values for element with number element-number. Different values for a loaded spring (Fx > Fx-limit) and an unloaded spring (Fx < FX-limit) must be provided.
Examples	**BLSPRING 4 VALUE CVx 1e7 1e4** **BLSPRING 4 VALUE LOADED CVy 5e7 1e2** **BLSPRING 4 VALUE UNLOADED CRx 100**

The [SERIE from to step] command can be used for element numbers.

1.10.10 STIFF:

Syntax	**STIFF *element-number node-number1 node-number2*** **STIFF *element-number node-number***
Meaning	Define a element with stiffness matrix between node-number1 and node-number2. If node-number2 is omitted, it is assumed to be node-number1+1.
Examples	**STIFF 1 1 5** **STIFF 2 2**

The [SERIE from to step] command can be used for all numbers.

Syntax	**STIFF *element-number* DELETE**
Meaning	Delete element with number “element-number”
Examples	**STIFF 2 DELETE**

The [SERIE from to step] command can be used for element numbers.

1.10.10.1 Sub-Scope STIFF – VALUE:

Syntax	**STIFF *element-number* VALUE BEGIN** **STIFF *element-number* VALUE END**
Meaning	Start or end the definition of stiffness values for element “element-number”.
Examples	**STIFF 1 VALUE BEGIN**

Between **STIFF** ***element-number*** **VALUE BEGIN** and **STIFF** ***element-number*** **VALUE BEGIN,** the following commands are available:

Syntax	**K11** ***line-number*** **value1 value2 value3 value4 value5 value6** **K12** ***line-number*** **value1 value2 value3 value4 value5 value6** **K21** ***line-number*** **value1 value2 value3 value4 value5 value6** **K22** ***line-number*** **value1 value2 value3 value4 value5 value6**
Meaning	Set the elements in the stiffness matrix. The matrix itself looks like: K11 K12 K21 K22
Examples	**K11 1 0.456 8.435 0.0 1.234 4.321 5.2**

1.10.11 FLEX:

Syntax	**FLEX** ***element-number node-number1 node-number2*** **FLEX** ***element-number node-number***
Meaning	Define a element with flexibility matrix between node-number1 and node-number2. If node-number2 is omitted, it is assumed to be node-number1+1.
Examples	**FLEX 1 1 5** **FLEX 2 2**

The [SERIE from to step] command can be used for all numbers.

Syntax	**FLEX** ***element-number*** **DELETE**
Meaning	Delete element with number “element-number”
Examples	**FLEX 2 DELETE**

The [SERIE from to step] command can be used for element numbers.

1.10.11.1 Sub-Scope FLEX – VALUE:

Syntax	**FLEX** ***element-number*** **VALUE BEGIN** **FLEX** ***element-number*** **VALUE END**
Meaning	Start or end the definition of stiffness values for element “element-number”.
Examples	**FLEX 1 VALUE BEGIN**

Between **STIFF** ***element-number*** **VALUE BEGIN** and **STIFF** ***element-number*** **VALUE BEGIN,** the following commands are available:

Syntax	**F11** ***line-number*** **value1 value2 value3 value4 value5 value6** **F12** ***line-number*** **value1 value2 value3 value4 value5 value6** **F21** ***line-number*** **value1 value2 value3 value4 value5 value6** **F22** ***line-number*** **value1 value2 value3 value4 value5 value6**
Meaning	Set the elements in the stiffness matrix. The matrix itself looks like: F11 F12 F21 F22
Examples	**F11 1 0.456 8.435 0.0 1.234 4.321 5.2**

1.10.12 VDAMP:

Syntax	**VDAMP** ***element-number node-number1 node-number2*** **VDAMP** ***element-number node-number***
Meaning	Define a viscous damper between node-number1 and node-number2. If node-number2 is omitted, it is assumed to be node-number1+1.
Examples	**VDAMP 1 1 5** **VDAMP 2 2**

The [SERIE from to step] command can be used for all numbers.

Syntax	**VDAMP** ***element-number*** **DELETE**
Meaning	Delete element with number “element-number”
Examples	**VDAMP 2 DELETE**

The [SERIE from to step] command can be used for element numbers.

Syntax	**VDAMP** ***element-number*** **VALUE** ***CVx CVy CVz CRx CRy CRz alpha***
Meaning	Set the spring values for element with number element-number.
Examples	**VDAMP 3 VALUE 1e5 1e5 1e5 1e3 1e3 1e3 1.5**

The [SERIE from to step] command can be used for element numbers.

1.10.13 SDAMP:

Syntax	**SDAMP** ***element-number node-number1 node-number2*** **SDAMP** ***element-number node-number***
Meaning	Define a damper spring between node-number1 and node-number2. If node-number2 is omitted, it is assumed to be node-number1+1.
Examples	**FRIC 1 1 5** **FRIC 2 2**

The [SERIE from to step] command can be used for all numbers.

Syntax	**SDAMP *element-number* DELETE**
Meaning	Delete element with number “element-number”
Examples	**SDAMP 2 DELETE**

The [SERIE from to step] command can be used for element numbers.

Syntax	**SDAMP *element-number* VALUE *CVx1 CVx2 CVx3 dx1 dx2 dx3 alpha CVvx***
Meaning	Set the spring values for element with number element-number.
Examples	**SDAMP 3 VALUE 1e5 1e5 1e5 0.4 0.2 0.2 12 123**

The [SERIE from to step] command can be used for element numbers.

1.10.14 ELEM:

Syntax	**ELEM *element-number* ECC BEGIN *ex ey ez*** **ELEM *element-number* ECC BEGIN *ex ey*** **ELEM *element-number* ECC BEGIN *ex*** **ELEM *element-number* ECC END *ex ey ez*** **ELEM *element-number* ECC END *ex ey*** **ELEM *element-number* ECC END *ex***
Meaning	Set the eccentricity values of the specified element at the begin or at the end of the element. If ez or ey and ez are omitted, they are set to 0.0.
Examples	**ELEM 17 ECC BEGIN 0.0 2.5 0.2** **ELEM 18 ECC BEGIN 0.0 2.8** **ELEM 17 ECC END 0.0 2.8** **ELEM 18 ECC END 0.0 3.0 –0.3**

The [SERIE from to step] command can be used for element numbers.

Syntax	**ELEM *element-number* RELEASE GLOBAL BEGIN *DoF1 DoF2 …*** **ELEM *element-number* RELEASE GLOBAL END *DoF1 DoF2 …*** **ELEM *element-number* RELEASE LOCAL BEGIN *DoF1 DoF2 …*** **ELEM *element-number* RELEASE LOCAL END *DoF1 DoF2 …***
Meaning	Set the degree of freedom of the start or end of the element for local or global coordinate systems. Possible values for degree of freedom (***DoF***) are: **Vx Vy Vz Rx Ry Rz**
Examples	**ELEM 17 RELEASE GLOBAL BEGIN Rx Ry Rz** **ELEM 17 RELEASE GLOBAL END Vy Ry** **ELEM 18 RELEASE LOCAL BEGIN Vx Vz Rz** **ELEM 18 RELEASE LOCAL BEGIN Rx**

The [SERIE from to step] command can be used for element numbers.

Syntax	**ELEM *element-number* BETA *beta alpha1 alpha2 L*** **ELEM *element-number* BETA *beta alpha1 alpha2*** **ELEM *element-number* BETA *beta alpha1*** **ELEM *element-number* BETA *beta***
Meaning	Set angle values for specified element. If any value is omitted, it is set to 0.0. Angles must be given in degree (360°).
Examples	**ELEM 18 BETA 15.05 –17.4 30.0 8.0** **ELEM 19 BETA 0.0 90.0 0.0 3.0** **ELEM 20 BETA 12.0**

The [SERIE from to step] command can be used for element numbers.

Syntax	**ELEM *element-number* AGE *age shrinktime humidity temperature***
Meaning	Set age of the element, time for shrinking already consumed, relative humidity and ambient air temperature.
Examples	**ELEM 20 AGE 12.0 5 75 24**

The [SERIE from to step] command can be used for element numbers.

Syntax	**ELEM *element-number* DUCTMIN begin end**
Meaning	Set web width reduction due to ducts in web for shear capacity check at begin / end of element.
Examples	**ELEM 20 DUCTMIN 0.12 0.12**

The [SERIE from to step] command can be used for element numbers.

Syntax	**ELEM *element-number* SKIP FIB** **ELEM *element-number* SKIP ULT** **ELEM *element-number* SKIP SHEAR** **ELEM *element-number* SKIP RCD**
Meaning	Skip fibre stress check, ultimate load check, shear capacity check or RC design for element “element-number”. The options can be combined.
Examples	**ELEM 20 SKIP FIB SHEAR**

The [SERIE from to step] command can be used for element numbers.

1.10.15 Sub-scope: TENDON:

Syntax	**RMTENDON** ***tendon-number*** **INTERNAL** **RMTENDON** ***tendon-number*** **EXTERNAL** **RMTENDON END**
Meaning	Start or end the definition of a tendon. The second parameter defines the tendon type. If the tendon already exists, it is cleared (all definitions are removed).
Examples	**RMTENDON 1005 INTERNAL** **RMTENDON 500 EXTERNAL** **RMTENDON END**

The **TENDON** command opens a new sub-scope for tendon geometry definition. Within this scope, the following commands are available:

- **RMHALT, RMERROR, RMWARNING, RMLOG** as described in chapter 1.1.
- **TENDON END:** To end this scope.

- **INFO:** Set an info text for the tendon.
- **MAT:** Define the tendon material.
- **AREA:** Define the tendon area values.
- **COUNT:** Define the number of tendons in the tendon group.
- **FRIC:** Define the tendon friction values.
- **NODEAT:** Define a tendon geometry point.
- **STRESS:** Activate elements in this tendon.

The syntax for these commands are:

1.10.15.1 INFO:

Syntax	**INFO "*message*"**
Meaning	Set an info message for the tendon.
Examples	**INFO "tendon group 6 for 3rd construction stage"**

1.10.15.2 MAT:

Syntax	**MAT "*material name*"**
Meaning	Set the material of the tendon. The material must exist in RM2000.
Examples	**MAT "steel 1790"**

1.10.15.3 AREA:

Syntax	**AREA** ***tendon duct***
Meaning	Set area values of the tendon.
Examples	**AREA 0.0015 0.0050**

1.10.15.4 COUNT:

Syntax	**COUNT** ***n***
Meaning	Set the number of geometrically identical tendons for the tendon group.
Examples	**COUNT 10**

1.10.15.5 FRIC:

Syntax	**FRIC** ***friction wobble***
Meaning	Set the friction value and the wobble factor for the tendon.
Examples	**FRIC 0.25 0.15**

1.10.15.6 NODEAT:

Syntax	**NODEAT** ***x y z*** **NODEAT** ***x y z*** **STRAIGHT** **NODEAT** ***x y z*** **STRAIGHT** ***number*** **NODEAT** ***x y z*** **VECTOR** ***dx dy dz***
Meaning	Define a tendon geometry point by three coordinates. If the keyword STRAIGHT is added (e.g.: for external tendon geometry), the next tendon part is a straight line. If "number" follows the keyword "STRAIGHT", a new element with that number is created later during recalc. If VECTOR is added, the next tendon part starts with the defined direction.
Examples	**NODEAT 5.0 -4.0 0.8** **NODEAT 15. 2. 1.0 STRAIGHT** **NODEAT 25.0 -6.0 1.0** **NODEAT 27.0 -5.6 0.8 VECTOR 1.0 0.2 –0.04**

Syntax	**NODEAT ELEM *elemnr pos*** **NODEAT ELEM *elemnr pos ecctype ey ez*** **NODEAT ELEM *elemnr pos ecctype ey ez* ADDPOI *"name"*** **NODEAT ELEM *elemnr pos ecctype ey ez anlgetype ay az*** **NODEAT ELEM *elemnr pos ecctype ey ez anlgetype ay az*** **ADDPOI *"name"*** **NODEAT ELEM *elemnr pos* STRAIGHT** **NODEAT ELEM *elemnr pos ecctype ey ez* STRAIGHT** **NODEAT ELEM *elemnr pos ecctype ey ez* ADDPOI *"name"* STRAIGHT** **NODEAT ELEM *elemnr pos* STRAIGHT *newelemnr*** **NODEAT ELEM *elemnr pos ecctype ey ez* STRAIGHT *newelemnr*** **NODEAT ELEM *elemnr pos ecctype ey ez* ADDPOI *"name"*** **STRAIGHT *newelemnr*** **IN THIS DOCUMENTATION, SOME COMMANDS ARE DIVIDED INTO TWO LINES. IN A TCL FILE, ALL COMMANDS MUST BE WRITTEN IN ONE LINE!**
Meaning	***ecctype*** can be **ANGLE, ANGLENO, ANGLEQ.** For **ANGLEQ**, a valid pointname must be declared with **ADDPOI *"name".*** ***angletype*** can be **ECC, ECCG, ECCNO, ECCGNO, ECCQ, ECCGQ.** For **ECCQ** and **ECCGQ**, a valid pointname must be declared with **ADDPOI *"name".*** Define a tendon geometry point by reference to an element. “pos” is the relative element coordinate (0.0: start <= pos <= 1.0: end). ey and ez are optional eccentricity values. Eccentricities with option **ECC** are defined relative to the centre of gravity line of the element. If the option **ECCNO** is used, the eccentricity values are defined relative to the line defined by the node coordinates which define the position of the element (line between node at start and node at end of the element). If **ECCQ** is used, the eccentricity values are defined relative to the line defined by the additional point "name" defined the element cross section. For global eccentricity values, use **ECCG, ECCGNO** and **ECCGQ** instead of **ECC, ECCNO** and **ECCGQ**. If ANGLE is added, the next tendon part starts with the defined direction. Angles refer to the centre of gravity (**ANGLE**) or to the line defined by the node coordinates which define the position of the element (**ANGLENO**) or to the line defined by the position of the two additional points in the cross section defined by "name" (**ANGLEQ**). If the keyword **STRAIGHT** is added, the next tendon part is a straight line. If “number” follows the keyword **STRAIGHT**, a new element with that number is created later during recalc.
Examples	**NODEAT ELEM 17 0.0** **NODEAT ELEM 17 1.0 STRAIGHT** **NODEAT ELEM 18 0.5 ECC -2.0 0.3** **NODEAT ELEM 18. 0.7 ECCQ -1.0 0.0 ADDPOI "tend1" STRAIGHT 120**

	NODEAT ELEM 19. 0.1 ECC 0.0 0.0 ANGLENO 45.0 0.6 **NODEAT ELEM 19. 0.1 ECCNO 0 0 ANGLEQ 45.0 0.6 ADDPOI "tend2"**

1.10.15.7 Relative element-numbers

If an element-number in the NODEAT command is preceded by a '#', e.g., '#2', the number is interpreted as the 'index'th element which is stressed in this tendon. In this case the STRESS command must precede the NODEAT command!

Example:

```
… some other commands

RMTENDON 1007 INTERNAL

STRESS 10
STERSS 20
STRESS [100 160 5]

NODEAT ELEM #2
NODEAT ELEM #3
NODEAT ELEM #5
…
```

The first STRESS command will stress element 10, the second element 20. The third STRESS command will stress elements 100, 105, 110, 115, 120, … 160. The sequence of elements stressed is:

10 20 100 105 110 115 120 125 130 135 140 145 150 155 160

The reference to this elements by index (using the '#' character), results in the following elements:

#2 → element 20 (second element stressed).
#3 → element 100 (third element stressed).
#5 → element 110 (fifth element stressed).

1.10.15.8 STRESS:

Syntax	**STRESS** ***element-number*** **STRESS** ***element-number age*** **STRESS** ***element-number age ts***
Meaning	Activate element(s). Optionally the age and shrinking time for the element(s) can be set here.
Examples	**STRESS 15** **STRESS [SERIE 100 500 50] 12** **STRESS [SERIE 1000 1010] 15 40**

The [SERIE from to step] command can be used for element numbers.

1.11 Scope: RMSCHED

After the execution of **RMSCHED BEGIN,** this scope is entered. Within this scope, the following commands are available:

- **RMHALT, RMERROR, RMWARNING, RMLOG** as described in chapter 1.1.
- **RMLOAD END:** To end this scope.

- **LCOMB:** Define load combinations.
- **LMANAGE:** Define load management.
- **LCASE:** Define load cases.
- **LSET:** Define loads and load sets.
- **LANE:** Define lanes.
- **LTRAIN:** Define live loads.
- **SEISMIC:** Define seismic loads.
- **STAGE:** Define construction stages.
- **STRESS:** Define tendon stress actions.

The syntax for these commands are:

1.11.1 Sub-scope: LCOMB:

Syntax	**LCOMB BEGIN** **LCOMB END**
Meaning	Start or end the definition of load combination entries.
Examples	**LCOMB BEGIN** **LCASE END**

The **LCOMB** command opens a new sub-scope for load combination definition. Within this sub-scope, the following commands are available:

- **LCOMB END:** End the sub-scope.

- **LCASE:** Add a load case to a load combination.
- **ENVELOPE:** Add a superposition file to a load combination.

1.11.1.1 LCASE:

Syntax	**LCASE *number combination-rule*** **LCASE END**
Meaning	Start or end the definition of load combination entries for loadcase "number". "combination-rule" must be a valid combination rule from RM2000 (SUPADD, SUPAND, SUPANDX, SUPOR, SUPORX)
Examples	**LCASE 1500 SUPADD** **LCASE END**

1.11.1.2 ENVELOPE:

Syntax	**ENVELOPE *filename.sup combination-rule*** **ENVELOPE END**
Meaning	Start or end the definition of load combination entries for superposition file "filname.sup". "combination-rule" must be a valid combination rule from RM2000 (SUPADD, SUPAND, SUPANDX, SUPOR, SUPORX)
Examples	**ENVELOPE 1500 SUPANDX** **ENVELOPE END**

Both commands (LCASE and ENVELOPE) open a new sub-scope for load combination entries. Within this sub-scope, the following commands are available:

- **LCASE END**
- **ENVELOPE END:** End the sub-scope.

- **COMBINE:** Add an entry for load superposition.

1.11.1.3 Sub-scope: COMBINE:

Syntax	**COMBINE *number factor*** **COMBINE *number factor1 factor2***
Meaning	Combine the loadcase (envelope file) with the superposition load case "number" (1-24) with the specific factors. "factor1" is the favourable factor, "factor2" is the unfavourable factor.
Examples	**COMBINE 6 0.5** **COMBINE 23 1.1 0.9**

1.11.2 Sub-scope: LMANAGE:

Syntax	**LMANAGE "*name*"** **LMANAGE END**
Meaning	Start or end the definition of load management entries.
Examples	**LMANAGE "G1"** **LMANAGE END**

The LMANAGE command opens a new sub-scope for load management definition. Within this sub-scope, the following commands are available:

- **LMANAGE END:** End the sub-scope.

- **INFO:** Define a description for the load manage entry.
- **LCASE:** Add a load case to the load management.
- **ENVELOPE:** Add a superposition file to a load combination.

1.11.2.1 INFO:

Syntax	**INFO "*message*"**
Meaning	Define a description for the load set.
Examples	**INFO "This is load management set 1."**

1.11.2.2 LCASE:

Syntax	**LCASE *number* TOTAL** **LCASE *number* PRIMARY** **LCASE *number* SECONDARY**
Meaning	Add a load cases to the load management record. Up to three load cases can be added. Either the primary state result, the secondary state result or the total result will be added.
Examples	**LCASE 1000**

1.11.2.3 ENVELOPE:

Syntax	**ENVELOPE *envname.sup type***
Meaning	Add a load envelope to the load management record. "***type***" must be one of these: **SUPADD SUPAND SUPANDX SUPOR SUPORX**
Examples	**ENVELOPE env1.sup SUPORX**

1.11.3 Sub-scope: LSET:

Syntax	**LSET *number*** **LSET END**
Meaning	Start or end the definition of a load set.
Examples	**LSET 101** **LSET END**

The LSET command opens a new sub-scope for load set definition. Within this sub-scope, the following commands are available:

- **LSET END:** End the sub-scope.

- **INFO:** Define a description for the load set.
- **ITEM:** Add a load action to the load set.

1.11.3.1 INFO:

Syntax	**INFO "*message*"**
Meaning	Define a description for the load set.
Examples	**INFO "This is a self weight load case."**

1.11.3.2 ITEM:

Syntax	**ITEM *command from to step proj data1 data2 … data6***
Meaning	Define a load action for the load set. "command" must be one of the RM2000 keywords!
Examples	**ITEM "F" 1 1 1 d 1e5 1e5 0 1e2 1e2 0** **ITEM "QTL" 2 3 1 Qz 1 1 1 .3 0 .2**

1.11.4 Sub-scope: LCASE:

Syntax	**LCASE *number type*** **LCASE END**
Meaning	Start or end a load case description. "type" defines the type of load: L, U, LU for load, unload, load+unload.
Examples	**LCASE 5 LU** **LCASE END**

The LCASE command opens a new sub-scope for load case definition. Within this sub-scope, the following commands are available:

- **LCASE END:** End the sub-scope.

- **INFO:** Define a description for the load case.
- **LMANAGE:** Associate this load case with a load management record.

- **LSET:** Add a load set to this load case.

1.11.4.1 INFO:

Syntax	**INFO "*message*"**
Meaning	Define a description for the load case.
Examples	**INFO "This is a self weight load case."**

1.11.4.2 LMANAGE:

Syntax	**LMANAGE *type* "*name*"**
Meaning	Associate load case with load management "name" using a special type. "type" defines the load management type (U, I, T, A for unknown, increment, total sum, automatic).
Examples	**LMANAMGE I "cx1"**

1.11.4.3 LSET:

Syntax	**LSET *number const-fact "var-fact"***
Meaning	Add a load set to a load case.
Examples	**LSET 10 1.0 "table1"** **LSET [SERIE 1 21 10] 1.0 ""**

The [SERIE from to step] command can be used for numbers.

1.11.5 Sub-scope: LANE:

Syntax	**LANE *number*** **LANE END**
Meaning	Start or end a lane description.
Examples	**LANE 5** **LANE END**

The LANE command opens a new sub-scope for lane definition. Within this sub-scope, the following commands are available:

- **LANE END:** End the sub-scope.

- **INFO:** Define a description for the lane.
- **ITEM:** Define a lane position by a RM2000 command.

1.11.5.1 INFO:

Syntax	**INFO "*message*"**
Meaning	Define a description for the lane.
Examples	**INFO "This is the middle lane."**

1.11.5.2 ITEM:

Syntax	**ITEM *command flag data1 data2 data3 … data7***
Meaning	Add a lane position. Command must be one of the RM keywords for lane definition!
Examples	**ITEM POS3D d 0.2 2 2 0.5 0 0 1.5**

1.11.6 Sub-scope: LTRAIN:

Syntax	**LTRAIN *number*** **LTRAIN END**
Meaning	Start or end the definition of a traffic load .
Examples	**LTRAIN 101** **LTRAIN END**

The LSET command opens a new sub-scope for traffic load definition. Within this sub-scope, the following commands are available:

- **LTRAIN END:** End the sub-scope.

- **INFO:** Define a description for the traffic load .
- **FMINMAX:** Define the overall multiplication factors.
- **EPS:** Define the sensitivity factors.
- **FACT:** Define the specific multiplication factors.
- **ITEM:** Add a traffic load action to the traffic load.

1.11.6.1 INFO:

Syntax	**INFO "*message*"**
Meaning	Define a description for the traffic load.
Examples	**INFO "This is a uniform traffic load case."**

1.11.6.2 FMINMAX:

Syntax	**FMINMAX *factmin factmax***
Meaning	Define the maximum and minimum evaluation factors.
Examples	**FMINMAX 1.0 0.7**

1.11.6.3 EPS:

Syntax	**EPS *Vx Vy Vz Rx Ry Rz N Qy Qz Mx My Mz***
Meaning	Define the sensitivity factors (threshold values) for all forces and moments.
Examples	**EPS 1e-10 1e-10 1e-10 0 0 0 1e-10 0 0 1e-10 1e-10 1e-10**

1.11.6.4 FACT:

Syntax	**FACT *Vx Vy Vz Rx Ry Rz N Qy Qz Mx My Mz***
Meaning	Define the multiplication factors for the influence line for all forces and moments.
Examples	**FACT 1 1 1 1 1 1 1 1 1 1 1 1**

1.11.6.5 ITEM:

Syntax	**ITEM *command data1 data2 … data5***
Meaning	Define a load train action. The load train action name must match a valid RM2000 Ltrain action name!
Examples	**ITEM LIF 5 1 10 1 0**

1.11.7 Sub-scope: SEISMIC:

Syntax	**SEISMIC *number rule*** **SEISMIC END**
Meaning	Start or end the definition of a seismic load. “rule” must be one of the following: **ABS** for Absolute sum addition, **SRSS** for Pythagoreian addtion, **DSC** or **CQC**.
Examples	**SEISMIC 101 ABS** **SEISMIC END**

The LSEISMIC command opens a new sub-scope for seismic load definition. Within this sub-scope, the following commands are available:

- **SEISMIC END:** End the sub-scope.

- **INFO:** Define a description for the load set.
- **FILE:** Define the filename for the modal file.
- **DURATION:** Define the duration of the seismic event.
- **ITEM:** Add a load action to the load set.

1.11.7.1 INFO:

Syntax	**INFO** ***message***
Meaning	Define a description for the seismic event.
Examples	**INFO This is a self weight load case.**

1.11.7.2 FILE:

Syntax	**FILE** ***filenamd.mod***
Meaning	Define the name of the modal file.
Examples	**FILE modfile.mod**

1.11.7.3 DURATION:

Syntax	**DURATION** ***time***
Meaning	Define the duration of the seismic event..
Examples	**DURATION 10.5**

1.11.7.4 ITEM:

Syntax	**ITEM** ***type Vx Vy Vz fact "name"***
Meaning	Define a seismic response spectrum graph. "***type***" must be one of those: **D V A** **"Vx Vy Vz"** vector defining the direction of displacement (**D**), velocity (**V**) or acceleration (**A**). "fact" is the damping factor. "name" is the name for the graph.
Examples	**ITEM A 0.2 1.0 –0.1 0.95 "acceleration 1"**

1.11.8 Sub-scope: CONSTRAINT:

Syntax	**CONSTRAINT** ***constraint-set-number*** **CONSTRAINT** ***constraint-set-number*** **VARINFIX** **CONSTRAINT END**
Meaning	Start or end the definition of additional constraints. **VARINFIX** means that the variable loadcase results are already included in the fix results.
Examples	**CONSTRAINT 5** **CONSTRAINT END**

The **CONSTRAINT** command opens a new sub-scope for the definition of additional constraints. Within this sub-scope, the following commands are available:

RMHALT, RMERROR, RMWARNING, RMLOG as described in chapter 1.1.

- **CONSTRAINT END:** To end this scope.

- **INFO:** Add a description for the constraints set.
- **LCASE:** Define constraints for load cases.
- **SUP:** Define constraints for superposition files.
- **NODE:** Define constraints for nodes.
- **ELEM:** Define constraints for elements.

The syntax for these commands are:

1.11.8.1 INFO:

Syntax	**INFO "*message*"**
Meaning	Define an info message for the constraints set.
Examples	**INFO "Load constraints"**

1.11.8.2 LCASE:

Syntax	**LCASE *from to step* +** **LCASE *from to step* VAR** **LCASE *from to step* FIX *factor***
Meaning	Define constraints for a serie of loadcases (from – to – step).
Examples	**LCASE 100 800 100 VAR** **LCASE 110 110 1 FIX 1.0**

1.11.8.3 SUP:

Syntax	**SUP "supfilename.sup" *minmax* FIX factor**
Meaning	Define constraints for a superposition. ***minmax*** defines the specific result used and must be one or a more out of the following: **MAXVx, MAXVy, … MAXMy, MAXMz, MINVx, MINVy, … MINMy, MINMz.**
Examples	**SUP "ltrain.sup" MAXMx MAXMz 1.5**

1.11.8.4 NDDEF:

Syntax	**NDDEF *from to step DoF * factor +*** **NDDEF *from to step DoF * factor operator value*** **NDDEF *from to step DoF * factor* IN *value1 value2***
Meaning	Define nodes (from – to – step) for a constraints set. ***DoF*** is the degrees of freedom for the constraint (Vx, Vy, Vz). The result is multiplied by factor. If + is given instead of an operator, the result is added to the next result. ***operator*** must be one of those: **EQ, LT, GT** (for equal, less than or equal, greater than or equal).
Examples	**NDDEF 1 100 5 Vy * 1.0 +** **NDDEF 200 200 1 Vy * 1.2 IN 1e4 1e5**

1.11.8.5 NDROT:

Syntax	**NDROT *from to step DoF * factor +*** **NDROT *from to step DoF * factor operator value*** **NDROT *from to step DoF * factor* IN *value1 value2***
Meaning	Define nodes (from – to – step) for a constraints set. ***DoF*** is the degrees of freedom for the constraint (Rx, Ry, Rz). The result is multiplied by factor. If + is given instead of an operator, the result is added to the next result. ***operator*** must be one of those: **EQ, LT, GT** (for equal, less than or equal, greater than or equal).
Examples	**NDROT 1 100 5 Rx * 1.0 +** **NDROT 200 200 1 Rx * 1.2 EQ 0**

1.11.8.6 NDFOR:

Syntax	**NDFOR *from to step DoF * factor +*** **NDFOR *from to step DoF * factor operator value*** **NDFOR *from to step DoF * factor* IN *value1 value2***
Meaning	Define nodes (from – to – step) for a constraints set. ***DoF*** is the degrees of freedom for the constraint (Nx, Qy, Qz). The result is multiplied by factor. If + is given instead of an operator, the result is added to the next result. ***operator*** must be one of those: **EQ, LT, GT** (for equal, less than or equal, greater than or equal).
Examples	**NDFOR 1 100 5 Rx * 1.0 +** **NDFOR 200 200 1 Rx * 1.2 EQ 0**

1.11.8.7 NDMOM:

Syntax	**NDMOM** ***from to step DoF * factor +*** **NDMOM** ***from to step DoF * factor operator value*** **NDMOM** ***from to step DoF * factor*** **IN** ***value1 value2***
Meaning	Define nodes (from – to – step) for a constraints set. ***DoF*** is the degrees of freedom for the constraint (Mx, My, Mz). The result is multiplied by factor. If + is given instead of an operator, the result is added to the next result. ***operator*** must be one of those: **EQ, LT, GT** (for equal, less than or equal, greater than or equal).
Examples	**NDMOM 1 100 5 Mz * 1.0 +** **NDMOM 200 200 1 Mz * 1.2 LT 1500**

1.11.8.8 ELDEF:

Syntax	**ELDEF** ***from to step x/l DoF * factor resmodes +*** **ELDEF** ***from to step x/l DoF * factor resmodes operator value*** **ELDEF** ***from to step x/l DoF * factor resmodes +*** **IN** ***val1 val2***
Meaning	Define element (from – to – step) for a constraints set. ***x/l*** is the position within the element (0 = begin of element, 1 = end). ***DoF*** is the degrees of freedom for the constraint (Vx, Vy, Vz). The result is multiplied by factor. ***operator*** must be one of those: **EQ, LT, GT** (for equal, less than, greater than, less than or equal, greater than or equal). ***resmodes*** defines the result typ: **SPLIT/JOIN** and **PRIMARY/SECONDARY**. Without a resmode, the total element result are chosen. **SPLIT** is used for adding composite section results to section results. **JOIN** is used for adding results from parts of the composite section to section composite section results.
Examples	**ELDEF 1 1 1 0.6 Vx SPLIT PRIMARY * 1 +** **ELDEF 2 2 1 1.6 Vx SPLIT PRIMARY * 1 IN –0.1 0.1**

1.11.8.9 ELROT:

Syntax	**ELROT** ***from to step x/l DoF * factor resmodes +*** **ELROT** ***from to step x/l DoF * factor resmodes operator value*** **ELROT** ***from to step x/l DoF * factor resmodes + IN val1 val2***
Meaning	Define element (from – to – step) for a constraints set. ***x/l*** is the position within the element (0 = begin of element, 1 = end). ***DoF*** is the degrees of freedom for the constraint (Rx, Ry, Rz). The result is multiplied by factor. ***operator*** must be one of those: **EQ, LT, GT** (for equal, less than, greater than, less than or equal, greater than or equal). ***resmodes*** defines the result typ: **SPLIT/JOIN** and **PRIMARY/SECONDARY**. Without a resmode, the total element result are chosen. **SPLIT** is used for adding composite section results to section results. **JOIN** is used for adding results from parts of the composite section to section composite section results.
Examples	**ELROT 1 1 1 0.6 Vx SPLIT PRIMARY * 1 +** **ELROT 2 2 1 1.6 Vx SPLIT PRIMARY * 1 IN –0.1 0.1**

1.11.8.10 ELFOR:

Syntax	**ELFOR** ***from to step x/l DoF * factor resmodes +*** **ELFOR** ***from to step x/l DoF * factor resmodes operator value*** **ELFOR** ***from to step x/l DoF * factor resmodes + IN val1 val2***
Meaning	Define element (from – to – step) for a constraints set. ***x/l*** is the position within the element (0 = begin of element, 1 = end). ***DoF*** is the degrees of freedom for the constraint (Nx, Qy, Qz). The result is multiplied by factor. ***operator*** must be one of those: **EQ, LT, GT** (for equal, less than, greater than, less than or equal, greater than or equal). ***resmodes*** defines the result typ: **SPLIT/JOIN** and **PRIMARY/SECONDARY**. Without a resmode, the total element result are chosen. **SPLIT** is used for adding composite section results to section results. **JOIN** is used for adding results from parts of the composite section to section composite section results.
Examples	**ELFOR 1 1 1 0.6 Vx SPLIT PRIMARY * 1 +** **ELFOR 2 2 1 1.6 Vx SPLIT PRIMARY * 1 IN –0.1 0.1**

1.11.8.11 ELMOM:

Syntax	**ELMOM** ***from to step x/l DoF * factor resmodes +*** **ELMOM** ***from to step x/l DoF * factor resmodes operator value*** **ELMOM** ***from to step x/l DoF * factor resmodes* + IN *val1 val2***
Meaning	Define element (from – to – step) for a constraints set. ***x/l*** is the position within the element (0 = begin of element, 1 = end). ***DoF*** is the degrees of freedom for the constraint (Mx, My, Mz). The result is multiplied by factor. ***operator*** must be one of those: **EQ, LT, GT** (for equal, less than, greater than, less than or equal, greater than or equal). ***resmodes*** defines the result typ: **SPLIT/JOIN** and **PRIMARY/SECONDARY**. Without a resmode, the total element result are chosen. **SPLIT** is used for adding composite section results to section results. **JOIN** is used for adding results from parts of the composite section to section composite section results.
Examples	**ELMOM 1 1 1 0.6 Vx SPLIT PRIMARY * 1 +** **ELMOM 2 2 1 1.6 Vx SPLIT PRIMARY * 1 IN –0.1 0.1**

1.11.9 Sub-scope: STAGE:

Syntax	**STAGE** ***stage-number*** **STAGE END**
Meaning	Start or end the definition of construction stages. Stage "stage-number" is cleared (all action, … are removed) if it already exists.
Examples	**STAGE 5** **STAGE END**

The **STAGE** command opens a new sub-scope for construction stage definition. Within this sub-scope, the following commands are available:

RMHALT, RMERROR, RMWARNING, RMLOG as described in chapter 1.1.

- **RMSTAGE END:** To end this scope.

- **ELEM:** Activate or deactivate elements.
- **INFO:** Set description of stage.
- **MODULE:** Add action.

The syntax for these commands are:

1.11.9.1 INFO:

Syntax	**INFO "*message*"**
Meaning	Define an info message for the construction stage.
Examples	**INFO "Tendon 4-10 stressing."**

1.11.9.2 ELEM:

Syntax	**ELEM ACTIVATE** ***element-number*** **ELEM DEACTIVATE** ***element-number***
Meaning	Activate specified element or series of elements. If step of a series is omitted, it is assumed to be 1.
Examples	**ELEM ACTIVATE 15** **ELEM ACTIVATE [SERIE 100 500 50]**

The [SERIE from to step] command can be used for numbers.

1.11.9.3 MODULE:

Syntax	**MODULE** ***module-name "in1" "in2" "out1" "out2" time description***
Meaning	Add an action to the construction stage. The action name must match the module names in RM2000 (e.g., Calc, Creep, …)
Examples	**MODULE Calc 1 "" "" * 0.0 "Recalc loadcase 1"** **MODULE Creep "" 5 601 * 14 "Creep action 5"**

1.11.10 Sub-scope: TENDON

After the execution of **TENDON BEGIN,** this scope is entered. Within this scope, the following commands are available:

- **RMHALT, RMERROR, RMWARNING, RMLOG** as described in chapter 1.1.
- **TENDON END:** To end this scope.

- **RELEASE:** Add a release action.
- **STRESS:** Add a stress action.
- **WEDGESLIP:** Add a wedge-slip action.

The syntax for these commands are:

1.11.10.1 STRESS:

Syntax	**STRESS** ***tendon*** **LEFT** ***"stress-group"*** **ABS** ***force*** ***"description"*** **STRESS** ***tendon*** **RIGHT** ***"stress-group"*** **ABS** ***force*** ***"description"*** **STRESS** ***tendon*** **LEFT** ***"stress-group"*** **REL** ***factor*** ***"description"*** **STRESS** ***tendon*** **RIGHT** ***"stress-group"*** **REL** ***factor*** ***"description"***
Meaning	Add a stress action at the left or right end of the tendon. The stress force can be defined by an absolute force or a relative factor.
Examples	**STRESS 1 LEFT "CS1" ABS 1.70e3 "First stress action"** **STRESS 2 RIGHT "CS1" REL 1.05 "Second stress action"**

The [SERIE from to step] command can be used for tendon numbers.

1.11.10.2 RELEASE:

Syntax	**RELEASE** ***tendon*** **LEFT** ***"stress-group"*** **ABS** ***force*** ***"description"*** **RELEASE** ***tendon*** **RIGHT** ***"stress-group"*** **ABS** ***force*** ***"description"*** **RELEASE** ***tendon*** **LEFT** ***"stress-group"*** **REL** ***factor*** ***"description"*** **RELEASE** ***tendon*** **RIGHT** ***"stress-group"*** **REL** ***factor*** ***"description"***
Meaning	Add a release action at the left or right end of the tendon. The final force can be defined by an absolute force or a relative factor.
Examples	**RELEASE 1 LEFT "CS1" ABS 1.65e3 "First release action"** **RELEASE 2 RIGHT "CS1" REL 0.95 "Second release action"**

The [SERIE from to step] command can be used for tendon numbers.

1.11.10.3 WEGDESLIP:

Syntax	**WEDGESLIP** ***tendon*** **LEFT** ***"stress-group"*** ***slip*** ***"description"*** **WEDGESLIP** ***tendon*** **RIGHT** ***"stress-group"*** ***slip*** ***"description"***
Meaning	Add a stress action at the left or right end of the tendon.
Examples	**WEDGESLIP 1 LEFT "CS1" 0.006 "Left wedge slip action"** **WEDGESLIP 2 RIGHT "CS1" 0.006 "Right wedge slip action"**

The [SERIE from to step] command can be used for tendon numbers.

1.12 Scope: RMRESULT

After the execution of **RMRESULT BEGIN,** this scope is entered. Within this scope, the following commands are available:

- **RMHALT, RMERROR, RMWARNING, RMLOG** as described in chapter 1.1.
- **RMRESULT END:** To end this scope.

- **HEADER:** Define header for list file output.

- **LIST:** Data output for list file.
- **WRITE:** Data output without list headers.
- **RESMODE:** Set output mode.
- **UNIT:** Get user-units from RM2000.
- **FACTOR:** Get user-units factor from RM2000.
- **EXIST:** Check if an object exists.
- **RMMAT:** Get material information.
- **RMCROSS:** Get cross section information.
- **GROUP:** Get reinforcement goup information.
- **RMVAR:** Get variables information.
- **NODE:** Get node information and results.
- **NOSUPP:** Get node support information and results.
- **BEAM:** Get beam information and results.

- **CABLE:** Get cable information and results.
- **SPRING:** Get spring information and results.
- **FRIC:** Get friction spring information and results.
- **CONTACT:** Get contact spring information and results.
- **HINGE:** Get hinge spring information and results.
- **BLSPRING:** Get bilinear spring information and results.
- **STIFF:** Get stiffness element information and results.
- **FLEX:** Get flexibility element information and results.
- **VDAMP:** Get viscous damper element information and results.
- **SDAMP:** Get damper spring element information and results.
- **ELEM:** Get element information and results.
- **TENDON:** Get tendon information and results.
- **LMANATE:** Get load manager information.
- **LCASE:** Get load case information.
- **LSET:** Get load set information.

The syntax for these commands are:

1.12.1 Subscope HEADER:

Syntax	**HEADER MODULE** **HEADER DATA** **HEADER END**
Meaning	Starts or ends the definition of list file header for either module description or data description.
Examples	**HEADER MODULE** **HEADER END**

Within the subscope **HEADER**, only one command is valid:

1.12.1.1 ITEM:

Syntax	**ITEM "*Header line text*"**
Meaning	Define text for either module or data header. Text is truncated at 80th character.
Examples	**ITEM "Node coordinates: x y z"** **ITEM "----------"**

1.12.2 LIST:

Syntax	**LIST "*data*"**
Meaning	Add a data line to the output file (RM-List format). For formatted output, use the TCL command [format]. See TCL-syntax for this option.
Examples	**LIST [format "Node %i, x: %9.4f" $node $x]**

1.12.3 WRITE:

Syntax	**WRITE "*data*"**
Meaning	Add a data line to the output file (no headers). For formatted output, use the TCL command [format]. See TCL-syntax for this option.
Examples	**WRITE [format "Node %i, x: %9.4f" $node $x]**

1.12.4 RESMODE:

Syntax	**RESMODE PRIMARY** **RESMODE SECONDARY** **RESMODE TOTAL** **RESMODE LOCAL** **RESMODE GLOBAL** **RESMODE SPLIT** **RESMODE COMPOSITE**
Meaning	Set the mode for forces evaluation via TCL. Default modes are: **TOTAL, COMPOSITE and LOCAL.**
Examples	**RESMODE PRIMARY GLOBAL**

Syntax	**RESMODE GETTYPE** **RESMODE GETRELATIVE** **RESMODE GETSPLIT**
Meaning	Get the mode for forces evaluation via TCL. Default modes are: **TOTAL** (for type) **LOCAL** (for relative mode) and **COMPOSITE** (for split mode).
Examples	**RESMODE GETSPLIT**

1.12.5 UNIT:

Syntax	**UNIT ANGLE STRUCT** **UNIT ANGLE RESULT** **UNIT LENGTH STRUCT** **UNIT LENGTH CROSS** **UNIT FORCE** **UNIT MOMENT** **UNIT STRESS** **UNIT TEMP** **UNIT TIME SCHEDULE** **UNIT TIME LOAD**
Meaning	Get the unit name of the specific unit.
Examples	**UNIT LENGTH STRUCT**

1.12.6 FACTOR:

Syntax	**FACTOR ANGLE STRUCT** **FACTOR ANGLE RESULT** **FACTOR LENGTH STRUCT** **FACTOR LENGTH CROSS** **FACTOR FORCE** **FACTOR MOMENT** **FACTOR STRESS** **FACTOR TEMP** **FACTOR TIME SCHEDULE** **FACTOR TIME LOAD**
Meaning	Get the unit factor of the specific unit (relative to standard units).
Examples	**FACTOR LENGTH STRUCT**

1.12.7 EXIST:

Syntax	**EXIST CROSS** *"crossname"* **EXIST CROSSNODE** *"crossname" nodenumber* **EXIST CROSSELEM** *"crossname" elemnumber* **EXIST MAT** *"materialname"* **EXIST REINFGROUP** *groupnumber* **EXIST NODE** *nodenumber* **EXIST ELEM** *elemnumber* **EXIST TENDON** *tendonnumber* **EXIST LCOMB** *loadcasenumber/envelopefile* **EXIST LMANAGE** *"loadmanagename"* **EXIST LSET** *loadsetnumber* **EXIST LCASE** *loadcasenumber* **EXIST LCASE** *loadcasenumber* **RESULT** **EXIST LANE** *lanenumber* **EXIST LTRAIN** *loadtrainnumber* **EXIST SEISMIC** *seismicnumber* **EXIST STAGE** *stagenumber*
Meaning	Check if an object exists.
Examples	**EXIST CROSS "cross1"** **EXIST LSET 100** **EXIST TENDON 1024** **EXIST LCOMB 20** **EXIST LCOMB "env.sup"**

1.12.8 RMMAT:

Syntax	**RMMAT GETTOTAL** **RMMAT GETALL**
Meaning	Get the number or list of material names of defined materials.
Examples	**RMMAT GETTOTAL** **RMMAT GETALL**

Syntax	**RMMAT** *matname* **GETINFO** **RMMAT** *matname* **GETEMOD** **RMMAT** *matname* **GETSIGPRE** **RMMAT** *matname* **GETSIGPRESA** **RMMAT** *matname* **GETW28** **RMMAT** *matname* **GETFCTM** **RMMAT** *matname* **GETFYD** **RMMAT** *matname* **GETXI**
Meaning	Get material info / values (E-Modl, SIG-allow-pr, SIG-allowSA, W28, SIG-allow-m, SIGMA*, XI) of defined material.
Examples	**RMMAT B45 GETEMOD** **RMMAT PRST GETSIGPRESA**

Syntax	**RMMAT *matname* GETSIG *eps*** **RMMAT *matname* GETEPS *sigma*** **RMMAT *matname* GETSIG *eps* *gamma*** **RMMAT *matname* GETEPS *sigma* *gamma***
Meaning	Get sigma(eps) or eps(sigma) form nonlinear working diagram. If gamma is given by user, the material gamma is ignored and gamma is used for calculation.
Examples	**RMMAT B45 GETSIG -3.0** **RMMAT PRST GETEPS 200000**

1.12.9 RMCROSS:

Syntax	**RMCROSS GETTOTAL** **RMCROSS GETALL**
Meaning	Get the number or list of names of defined cross sections.
Examples	**RMCROSS GETTOTAL** **RMCROSS GETALL**

Syntax	**RMCROSS *crossname* GETINFO** **RMCROSS *crossname* GETH0** **RMCROSS *crossname* GETH1** **RMCROSS *crossname* GETH** **RMCROSS *crossname* GETW0** **RMCROSS *crossname* GETW1** **RMCROSS *crossname* GETW** **RMCROSS *crossname* GETECCY** **RMCROSS *crossname* GETECCZ** **RMCROSS *crossname* GETBETA** **RMCROSS *crossname* GETECCY SHEAR** **RMCROSS *crossname* GETECCZ SHEAR** **RMCROSS *crossname* GETBETA SHEAR** **RMCROSS *crossname* GETO** **RMCROSS *crossname* GETOIN**
Meaning	Get cross section info / values: geometric values: height below / above COG (centre of gravity), overall height, with left/right of COG, overall width, eccentre of COG, beta angle, perimeter outside/inside.
Examples	**RMCROSS rect001 GETINFO** **RMCROSS rect001 GETECCY**

Syntax	**RMCROSS *crossname* GETAX** **RMCROSS *crossname* GETAY** **RMCROSS *crossname* GETAZ** **RMCROSS *crossname* GETIX** **RMCROSS *crossname* GETIY** **RMCROSS *crossname* GETIZ** **RMCROSS *crossname* GETIW**
Meaning	Get cross section values: area, shear areas, moments of inertia.
Examples	**RMCROSS rect001 GETAX** **RMCROSS rect001 GETIW**

All RMCROSS commands can be followed by the keyword **CUTRIGHT**. After this key, four coordinates (z1, y1, z2, y2) defining a line between P1(z1,y1) and P2(z2,y2) must follow. Then, the results (cross area, moments of inertia, eccentre, ...) are calculated from the section that remains on the right side if the shape of the section is cut by the line defined.

Syntax	**RMCROSS *crossname* REINFGRP *groupname* GETY** **RMCROSS *crossname* REINFGRP *groupname* GETZ**
Meaning	Get the eccentres of the centre of gravity of all reinforcement points within group ***groupname***.
Examples	**RMCROSS rect001 REINFGRP reinf_top GETY**

1.12.10 GROUP:

Syntax	**GROUP GETTOTAL** **GROUP GETALL**
Meaning	Get the number or list of names of defined reinforcement groups.
Examples	**GROUP GETTOTAL** **GROUP GETALL**

Syntax	**GROUP *groupname* GETINFO** **GROUP *groupname* GETMAT** **GROUP *groupname* GETMAXD** **GROUP *groupname* GETAMAX** **GROUP *groupname* GETAMIN** **GROUP *groupname* GETAMINFACT**
Meaning	Get the info, material, diameter and area(s) for reinforcement group ***groupname***.
Examples	**GROUP "top" GETMAT**

1.12.11 RMVAR:

Syntax	**RMVAR GETTOTAL** **RMVAR GETALL**
Meaning	Get the number or list of names of defined variables.
Examples	**RMVAR GETTOTAL** **RMVAR GETALL**

Syntax	**RMVAR *varname* GETINFO**
Meaning	Get the info for variable ***varname***.
Examples	**RMVAR time GETINFO**

1.12.12 NODE:

Syntax	**NODE GETFIRST** **NODE GETLAST** **NODE GETTOTAL** **NODE GETFTS *from to step*** **NODE GETALL** **NODE GETACTIVE**
Meaning	Retrieve data from stucture nodes: - number of first node - number of last node - total node count - nodes within from – to - step - list containing all node numbers - list containing all active node numbers
Examples	**NODE GETFIRST** **NODE GETTOTAL** **NODE GETACTIVE**

Syntax	**NODE number GETX** **NODE number GETY** **NODE number GETZ**
Meaning	Retrieve node coordinates
Examples	**NODE 100 GETX**

Syntax	**NODE number GETMX** **NODE number GETMY** **NODE number GETMZ** **NODE number GETIMX** **NODE number GETIMY** **NODE number GETIMZ**
Meaning	Retrieve node mass data
Examples	**NODE 100 GETIMZ**

Syntax	**NODE number ISACTIVE**
Meaning	Checks if node "number" is active.
Examples	**NODE 100 ISACTIVE**

1.12.13 NOSUP:

Syntax	**NOSUP GETALL** **NOSUP GETACTIVE** **NOSUP GETFTS *from to step***
Meaning	Retrieve data from node supports: - list of all nodes with node support - list of all active nodes with node support - list of all nodes with node support within from – to – step
Examples	**NOSUP GETACTIVE** **NOSUP GETFTS 100 1000 50**

Syntax	**NOSUP *nodenumber* GETVALUE**
Meaning	Retrieve stiffness values for node support. The result will be an array with elements **Cx Cy Cz CMx CMy CMz** The result must be assigned to a variable using the "setarr" – command.
Examples	**NOSUP 100 GETVALUE** Example in use with the setarr – command: **setarr C [NOSUP 100 GETVALUE]** **set x [expr $C(Cx) / 123.0]**

Syntax	**NOSUP *nodenumber* GETECCX** **NOSUP *nodenumber* GETECCY** **NOSUP *nodenumber* GETECCZ**
Meaning	Retrieve eccentricity values for node support.
Examples	**NOSUP 100 GETECCY**

Syntax	**NOSUP *nodenumber* GETBETA** **NOSUP *nodenumber* GETALPHA1** **NOSUP *nodenumber* GETALPHA2**
Meaning	Retrieve node support orientation angles
Examples	**NOSUP 100 GETBETA**

1.12.14 BEAM:

Syntax	**BEAM GETFIRST** **BEAM GETLAST** **BEAM GETTOTAL** **BEAM GETFTS from to step** **BEAM GETALL** **BEAM GETACTIVE**
Meaning	Retrieve data from beams: - first beam number - last beam number - total number of beams - all beams within from to step - list with all beam numbers - list with all active beam numbers
Examples	**BEAM GETFTS 1000 1200 7** **BEAM GETTOTAL** **BEAM GETACTIVE**

Syntax	**BEAM *number* GETNPART** **BEAM *number* GETPARTS**
Meaning	Retrieve element part count / all parts in list
Examples	**BEAM 100 GETNPART** **BEAM 100 GETPARTS**

Syntax	**BEAM *number* GETMAT**
Meaning	Retrieve material name for beam.
Examples	**BEAM 18 GETMAT**

Syntax	**BEAM *number* GETEMOD** **BEAM *number* GETGMOD** **BEAM *number* GETALPHAT** **BEAM *number* GETGAMMA**
Meaning	Retrieve material data from beams: - E - modulus - G - modulus - Thermal expansion coefficient - Gravity
Examples	**BEAM 17 GETGMOD** **BEAM 150 GETGAMMA**

Syntax	**BEAM *number* GETCROSS BEGIN** **BEAM *number* GETCROSS END**
Meaning	Retrieve cross section name for beam begin or beam end.
Examples	**BEAM 18 GETCROSS BEGIN**

Syntax	**BEAM** ***number*** **GETAX** ***position-in-element*** **BEAM** ***number*** **GETAY** ***position-in-element*** **BEAM** ***number*** **GETAZ** ***position-in-element*** **BEAM** ***number*** **GETIX** ***position-in-element*** **BEAM** ***number*** **GETIY** ***position-in-element*** **BEAM** ***number*** **GETIZ** ***position-in-element*** **BEAM** ***number*** **GETO** ***position-in-element*** **BEAM** ***number*** **GETOIN** ***position-in-element***
Meaning	Retrieve cross section values for beam. ***position-in-element*** must be either **BEGIN** or **END** or any x/l value between 0 and 1 (linear interpolation).
Examples	**BEAM 18 GETAY BEGIN** **BEAM 20 GETO 0.7**

Syntax	**BEAM** ***number*** **GETCOMP** **BEAM** ***number*** **ISCOMP** **BEAM** ***number*** **COMPPARTS** **BEAM** ***number*** **COMPPARTS ALL**
Meaning	Retrieve data from beams: - getcomp -> get number of composite element this beam is part of - iscomp -> beam is composed of others? - compparts -> get elements numbers of parts of this comp. element. ALL: get all parts including composite part(s)
Examples	**BEAM 100 GETCOMP** **BEAM 100 ISCOMP**

Syntax	**BEAM** ***number*** **GETREINF** ***reinfgroup*** **BEAM** ***number*** **GETREINF1** ***reinfgroup*** **BEAM** ***number*** **GETREINF2** ***reinfgroup***
Meaning	Get total area / area A1 / area A2 of reinforcemenet for group ***reinfgroup*** for beam ***number***
Examples	**BEAM 100 GETRINF reinf_bot**

Syntax	**BEAM** ***number*** **GETREINFGRP** ***position-in-element*** **BEAM** ***number*** **GETREINFGRP** ***position-in-element*** **LONGITUDINAL** **BEAM** ***number*** **GETREINFGRP** ***position-in-element*** **BEAM** ***number*** **GETREINFGRP** ***position-in-element*** **LONGITUDINAL**
Meaning	Get names of reinforcement groups at beam ***number***. If **LONGITUDINAL** is added, only groups with one or more reinforcement definition points are returned. ***position-in-element*** must be either **BEGIN** or **END** or any x/l value between 0 and 1 (linear interpolation).
Examples	**BEAM 100 GETREINFGRP BEGIN** **BEAM 100 GETREINFGRP 0.5 LONGITUDINAL**

Syntax	**BEAM *number* FIBPOINT GETALL** **BEAM *number* FIBPOINT GETNAME number**
Meaning	Get all fibre stress check points or get fibre stress check point name for number-th point.
Examples	**BEAM 100 FIBPOINT GETALL** **BEAM 100 FIBPOINT GETNAME 2**

Syntax	**BEAM *number* FIBPOINT *name* GETMAT** **BEAM *number* FIBPOINT *name* GETX *position-in-element*** **BEAM *number* FIBPOINT *name* GETY *position-in-element*** **BEAM *number* FIBPOINT *name* GETZ *position-in-element***
Meaning	Get information for fibre stress check point: Material and coordinates at som element position. ***position-in-element*** must be either **BEGIN** or **END** or any x/l value between 0 and 1 (linear interpolation).
Examples	**BEAM 100 FIBPOINT "SP-O" GETMAT** **BEAM 100 FIBPOINT "SP-O" GETY 0.7**

Syntax	**BEAM *number* TENDON GETALL *position-in-element***
Meaning	Get numbers of all tendons at a specific beam point. ***position-in-element*** must be either **BEGIN** or **END** or any x/l value between 0 and 1 (linear interpolation).
Examples	**BEAM 100 TENDON GETALL BEGIN**

1.12.15 CABLE:

Cable access (GETFIRST, ...) as with BEAM!

Syntax	**CABLE *number* GETNPART**
Meaning	Retrieve element part count
Examples	**CABLE 100 GETNPART**

Syntax	**CABLE *number* GETMAT**
Meaning	Retrieve material name for cable.
Examples	**CABLE 18 GETMAT**

Syntax	**CABLE** ***number*** **GETEMOD** **CABLE** ***number*** **GETGMOD** **CABLE** ***number*** **GETALPHAT** **CABLE** ***number*** **GETGAMMA**
Meaning	Retrieve material data from cables: - E - modulus - G - modulus - Thermal expansion coefficient - Gravity
Examples	**CABLE 17 GETGMOD** **CABLE 150 GETGAMMA**

Syntax	**CABLE** ***number*** **GETCROSS BEGIN** **CABLE** ***number*** **GETCROSS END**
Meaning	Retrieve cross section name for cable begin or cable end.
Examples	**CABLE 18 GETCROSS BEGIN**

Syntax	**CABLE** ***number*** **GETAX** ***position-in-element*** **CABLE** ***number*** **GETAY** ***position-in-element*** **CABLE** ***number*** **GETAZ** ***position-in-element*** **CABLE** ***number*** **GETIX** ***position-in-element*** **CABLE** ***number*** **GETIY** ***position-in-element*** **CABLE** ***number*** **GETIZ** ***position-in-element*** **CABLE** ***number*** **GETO** ***position-in-element*** **CABLE** ***number*** **GETOIN** ***position-in-element***
Meaning	Retrieve cross section values for cable. ***position-in-element*** must be either **BEGIN** or **END** or any x/l value between 0 and 1 (linear interpolation)**.**
Examples	**CABLE 18 GETAY BEGIN** **CABLE 20 GETO 0.7**

1.12.16 SPRING:

Spring access (GETFIRST, ...) as with BEAM!

Syntax	**SPRING** ***element-number*** **GETVALUE**
Meaning	Get the spring values for element with number element-number. The result will be an array with elements **CVx CVy CVz CRx CRy CRz** The result must be assigned to a variable using the "setarr" – command.
Examples	**SPRING 3 GETVALUE** Example in use with the setarr – command: **setarr springc [SPRING 3 GETVALUE]** **set x [expr $springc(Cx) / 123.0]**

1.12.17 FRIC:

Fric access (GETFIRST, ...) as with BEAM!

Syntax	**FRIC *element-number* GETVALUE**
Meaning	Get the spring values for element with number element-number. The result will be an array with elements ***Ax ny nz Ix Iy Iz*** The result must be assigned to a variable using the "setarr" – command.
Examples	**FRIC 3 GETVALUE** Example in use with the setarr – command: **setarr springc [FRIC 3 GETVALUE]** **set x [expr $springc(Ix) / 123.0]**

1.12.18 CONTACT:

Contact access (GETFIRST, ...) as with BEAM!

Syntax	**CONTACT *element-number* GETVALUE**
Meaning	Get the formulas for spring constant for element with number element-number. The result will be an array with elements ***CVx CVy CVz CRx CRy CRz*** The result must be assigned to a variable using the "setarr" – command.
Examples	**CONTACT 3 GETVALUE** Example in use with the setarr – command: **setarr springc [CONTACT 3 GETVALUE]** **LIST $springc(VCx)**

1.12.19 HINGE:
Hinge access (GETFIRST, ...) as with BEAM!

1.12.20 BLSPRING:
Blspring access (GETFIRST, ...) as with BEAM!

Syntax	**BLSPRING *element-number* GETVALUE LOADED** **BLSPRING *element-number* GETVALUE UNLOADED**
Meaning	Get the spring constant for element with number element-number. The result will be an array with elements ***CVx CVy CVz CRx CRy CRz Fx QyY QzY*** The result must be assigned to a variable using the "setarr" – command.
Examples	**BLSPRING 3 GETVALUE LOADED** Example in use with the setarr – command: **setarr springc [BLSPRING 3 GETVALUE LOADED]** **set FxLimitLoaded $springc(Fx)**

1.12.21 STIFF:
Stiff access (GETFIRST, ...) as with BEAM!

Syntax	**STIFF *element-number* GETVALUE K11 *row col*** **STIFF *element-number* GETVALUE K12 *row col*** **STIFF *element-number* GETVALUE K21 *row col*** **STIFF *element-number* GETVALUE K22 *row col***
Meaning	Get the matrix element in row / col of submatrix K11 K12 K21 or K22.
Examples	**STIFF 3 GETVALUE K11 1**

1.12.22 FLEX:
Flex access (GETFIRST, ...) as with BEAM!

Syntax	**FLEX *element-number* GETVALUE F11 *row col*** **FLEX *element-number* GETVALUE F12 *row col*** **FLEX *element-number* GETVALUE F21 *row col*** **FLEX *element-number* GETVALUE F22 *row col***
Meaning	Get the matrix element in row / col of submatrix F11 F12 F21 or F22.
Examples	**FLEX 3 GETVALUE F11 1**

1.12.23 VDAMP:

VDamp access (GETFIRST, ...) as with BEAM!

Syntax	**VDAMP *element-number* GETVALUE**
Meaning	Get the spring values for element with number element-number. The result will be an array with elements **CVx CVy CVz CRx CRy CRz alpha** The result must be assigned to a variable using the "setarr" – command.
Examples	**VDAMP 3 GETVALUE** Example in use with the setarr – command: **setarr springc [VDAMP 3 GETVALUE]** **set x [expr $springc(alpha) * 100.0]**

1.12.24 SDAMP:

SDamp access (GETFIRST, ...) as with BEAM!

Syntax	**SDAMP *element-number* GETVALUE**
Meaning	Get the spring values for element with number element-number. The result will be an array with elements ***CVx1 CVx2 CVx3 dx1 dx2 dx3 alpha CVvx*** The result must be assigned to a variable using the "setarr" – command.
Examples	**SDAMP 3 GETVALUE** Example in use with the setarr – command: **setarr springc [SDAMP 3 GETVALUE]** **set x [expr $springc(CVvx) / 3.6]**

1.12.25 ELEM:

Syntax	**ELEM GETFIRST** **ELEM GETLAST** **ELEM GETTOTAL** **ELEM GETFTS from to step** **ELEM GETALL** **ELEM GETACTIVE**
Meaning	Retrieve data from stucture elements: - first element number - last element number - total element count - all element within from to step - list containing all element numbers - list containing all active element numbers
Examples	**ELEM GETTOTAL** **ELEM GETACTIVE** **ELEM GETFTS 100 200 10**

Syntax	**ELEM *number* GETTYPE** **ELEM *number* ISACTIVE**
Meaning	Retrieve data from stucture elements: - type of element (BEAM, CABLE, SPRING, …) - activation status
Examples	**ELEM 100 GETTYPE** **ELEM 100 ISACTIVE**

Syntax	**ELEM *number* GETNODE1** **ELEM *number* GETNODE2** **ELEM *number* GETNODE BEGIN** **ELEM *number* GETNODE END**
Meaning	Retrieve node numbers at begin and end of element.
Examples	**ELEM 100 GETNODE1**

Syntax	**ELEM number GETX *position-in-element*** **ELEM number GETY *position-in-element*** **ELEM number GETZ *position-in-element*** **ELEM number GETX *position-in-element*** **ELEM number GETY *position-in-element*** **ELEM number GETZ *position-in-element***
Meaning	Retrieve element coordinates. ***position-in-element*** must be either **BEGIN** or **END** or any x/l value between 0 and 1 (linear interpolation).
Examples	**ELEM 100 GETX BEGIN** **ELEM 101 GETY 0.7**

Syntax	**ELEM** ***number*** **GETECCX** ***position-in-element*** **ELEM** ***number*** **GETECCY** ***position-in-element*** **ELEM** ***number*** **GETECCZ** ***position-in-element*** **ELEM** ***number*** **GETECCX** ***position-in-element*** **ELEM** ***number*** **GETECCY** ***position-in-element*** **ELEM** ***number*** **GETECCZ** ***position-in-element***
Meaning	Get (user defined) eccentricity values for element. ***position-in-element*** must be either **BEGIN** or **END** or any x/l value between 0 and 1 (linear interpolation).
Examples	**ELEM 17 GETECCZ END**

Syntax	**ELEM** ***number*** **GETBETA** **ELEM** ***number*** **GETALPHA1** **ELEM** ***number*** **GETALPHA2** **ELEM** ***number*** **GETLENGTH**
Meaning	Get element orientation angles
Examples	**ELEM 17 GETBETA** **ELEM 150 GETALPHA1** **ELEM 102 GETLENGTH**

Syntax	**ELEM** ***number*** **GETRELEASE GLOBAL BEGIN** **ELEM** ***number*** **GETRELEASE GLOBAL END** **ELEM** ***number*** **GETRELEASE LOCAL BEGIN** **ELEM** ***number*** **GETRELEASE LOCAL END**
Meaning	Retrieve beam realeases for beam begin or beam end for local or global coordinate system. The result will be an array with elements **Vx Vy Vz Rx Ry Rz** for translational (V…) and rotational (R…) releases. The result must be assigned to a variable using the "setarr" – command.
Examples	**ELEM 18 GETRELEASE**

Syntax	**ELEM** ***number*** **GETAGE** **ELEM** ***number*** **GETSHRINK** **ELEM** ***number*** **GETHUMIDITY** **ELEM** ***number*** **GETTEMP**
Meaning	Retrieve element data: - initial age of beam - time the element has had to shrink up to the current moment - relative humidity (%) - ambient temperature
Examples	**ELEM 18 GETRELEASE**

1.12.26 Node / Node support result access:

Syntax	**NODE *number* GETLC *lcnumber***
Meaning	Retrieve result data for node "number" caused by loadcase "lcnumber". The result will be an array with elements **Vx Vy Vz Px Py Pz** The result must be assigned to a variable using the "setarr" – command.
Examples	**NODE 100 GETLC 5** Example in use with the setarr – command: **setarr def [NODE 100 GETLC 5]** **LIST $def(Vx)**

Syntax	**NOSUP *number* GETLC *lcnumber***
Meaning	Retrieve result data for NOSUP "number" caused by loadcase "lcnumber". The result will be an array with elements **Vx Vy Vz Px Py Pz Nx Qy Qz Mx My Mz** The result must be assigned to a variable using the "setarr" – command.
Examples	**NOSUP 100 GETLC 5** Example in use with the setarr – command: **setarr f [NOSUP 100 GETLC 5]** **LIST $f(Mz)**

Syntax	**NOSUP *number* GETSUP *supfile minmax***
Meaning	Retrieve result data for element "number" caused by superposition "supfile". Parameter **minmax** selects the specific result for a maximum or minimum value. **minmax** must be one of **MAXVx, MAXVy, … MAXMy, MAXMz** or **MINVx, MINVy, … MINMy, MINMz.** The result will be an array with elements **Vx Vy Vz Px Py Pz Nx Qy Qz Mx My Mz** The result must be assigned to a variable using the "setarr" – command.
Examples	**NOSUP 100 GETSUP "suppos.sup" MAXPz** **NOSUP 100 GETSUP "suppos.sup" MINQz** Example in use with the setarr – command: **setarr f [NOSUP 100 GETSUP "suppos.sup" MINQz]** **LIST $f(Qz)**

1.12.27 Element result access:

Result access is provided the same way for all element types. Strain and stress values are only available for beams and cables. As an example, the BEAM command is used here. Replace BEAM with SPRING, ... for other results.

Forces and deformations:

Syntax	**BEAM *number* GETLC *lcnumber* AT BEGIN** **BEAM *number* GETLC *lcnumber* AT END** **BEAM *number* GETLC *lcnumber* AT *position***
Meaning	Retrieve result data for beam "number" caused by loadcase "lcnumber". The result will be an array with elements **Vx Vy Vz Px Py Pz Nx Qy Qz Mx My Mz** The result must be assigned to a variable using the "setarr" – command. "positions" is 1 for begin and "npart" for end of element.
Examples	**BEAM 100 GETLC 5 AT BEGIN** **BEAM 100 GETLC 5 AT 3**

Syntax	**BEAM *number* ADDLC *lcnumber* AT *position dof value dof value …***
Meaning	Add results to loadcase for a specific element point. "positions" is 1 for begin and "npart" for end of element.
Examples	**BEAM 100 ADDLC 5 AT BEGIN NX 1000 MY 120 MZ 230**

Syntax	**BEAM *number* GETSUP *supfile* AT BEGIN *minmax*** **BEAM *number* GETSUP *supfile* AT END *minmax*** **BEAM *number* GETSUP *supfile* AT *position* *minmax***
Meaning	Retrieve result data for beam "number" caused by superposition "supfile". Parameter **minmax** selects the specific result for a maximum or minimum value. **minmax** must be one of **MAXVx, MAXVy, … MAXMy, MAXMz** or **MINVx, MINVy, … MINMy, MINMz.** The result will be an array with elements **Vx Vy Vz Px Py Pz Nx Qy Qz Mx My Mz** The result must be assigned to a variable using the "setarr" – command.
Examples	**BEAM 100 GETSUP "suppos.sup" AT BEGIN MAXPz** **BEAM 100 GETSUP "suppos.sup" AT 3 MINQz**

Syntax	**BEAM *number* ADDSUP *lcnumber* AT *pos. minm. dof val dof value …***
Meaning	Add results to superposition for a specific element point. "positions" is 1 for begin and "npart" for end of element.
Examples	**BEAM 100 ADDSUP 5 AT BEGIN MAXNx Nx 1000 My 120 Mz 230**

Strain and stress (BEAM and CABLE only!):

Syntax	**BEAM *number* LCSTRAIN *lcnumber* AT BEGIN POS z y** **BEAM *number* LCSTRAIN *lcnumber* AT *position* POS "name"**
Meaning	Retrieve strain for beam "number" caused by loadcase "lcnumber". The position for which strain will be returned can be defined by coordinates (y, z) or by the stress check point name.
Examples	**BEAM 100 LCSTRAIN 5 AT BEGIN POS -1.0 0.25** **BEAM 100 LCSTRAIN 5 AT 2 POS "Stressu"**

Syntax	**BEAM *number* LCSTRESS *lcnumber* AT BEGIN POS z y** **BEAM *number* LCSTRESS *lcnumber* AT *position* POS "name"**
Meaning	Retrieve strain for beam "number" caused by loadcase "lcnumber". The position for which strain will be returned can be defined by coordinates (y, z) or by the stress check point name.
Examples	**BEAM 100 LCSTRESS 5 AT BEGIN POS -1.0 0.25** **BEAM 100 LCSTRESS 5 AT 2 POS "Stressu"**

Syntax	**BEAM *number* SUPSTRAIN *supfile* AT BEGIN *minmax* POS z y** **BEAM *number* SUPSTRAIN *supfile* AT END *minmax* POS number** **BEAM *number* SUPSTRAIN *supfile* AT *position minmax* POS "name"**
Meaning	Retrieve strain for beam "number" caused by superposition "***supfile***". The position for which strain will be returned can be defined by coordinates (y, z), by the numberth stress check point or by the stress check point name. **minmax** must be one of **MAXVx, MAXVy, … MAXMy, MAXMz** or **MINVx, MINVy, … MINMy, MINMz OR MAX, MIN. "MAX" "MIN"** will return the maximum or minimum stress value selected out of all **MIN/MAX** values.
Examples	**BEAM 100 SUPSTRAIN "suppos.sup" AT BEGIN MAXPz POS -1.20 0** **BEAM 100 SUPSTRAIN "suppos.sup" AT 2 MINMz POS "STRESSP5"**

Syntax	**BEAM *number* SUPSTRESS *supfile* AT BEGIN *minmax* POS z y** **BEAM *number* SUPSTRESS *supfile* AT END *minmax* POS number** **BEAM *number* SUPSTRESS *supfile* AT *position minmax* POS "name"**
Meaning	Retrieve stress value for beam "number" caused by superposition "***supfile***". The position for which stress will be returned can be defined by coordinates (y, z), by the numberth stress check point or by the stress check point name. **minmax** must be one of **MAXVx, MAXVy, … MAXMy, MAXMz** or **MINVx, MINVy, … MINMy, MINMz OR MAX, MIN. "MAX" "MIN"** will return the maximum or minimum stress value selected out of all **MIN/MAX** values.
Examples	**BEAM 100 SUPSTRESS "suppos.sup" AT BEGIN MAXPz POS 4** **BEAM 100 SUPSTRESS "suppos.sup" AT 2 MIN POS "STRESSP5"**

Syntax	**BEAM *number* GETINFLINE *inf-file* AT BEGIN *minmax*** **BEAM *number* GETINFLINE *lane-nr* AT BEGIN *minmax*** **BEAM *number* GETINFLINE *inf-file* AT END *minmax*** **BEAM *number* GETINFLINE *lane-nr* AT END *minmax***
Meaning	Retrieve results from influence line from lane nr. ***lane-nr*** or directely from *.inf file for beam "number". **minmax** must be one of **MAXVx, MAXVy, … MAXMy, MAXMz** or **MINVx, MINVy, … MINMy, MINMz.**
Examples	**BEAM 100 GETINFLINE lane001.inf AT END MAXMz** **BEAM 100 GETINFLINE 2 AT BEGIN MAXVx**

1.12.28 TENDON:

Syntax	**TENDON GETFIRST** **TENDON GETLAST** **TENDON GETTOTAL** **TENDON GETFTS from to step** **TENDON GETALL** **TENDON GETACTIVE**
Meaning	Retrieve data from stucture tendons: - first tendon number - last tendon number - total tendon count - all tendons within from to step - list containing all tendon numbers - list containing all active tendon numbers
Examples	**TENDON GETTOTAL** **TENDON GETACTIVE** **TENDON GETFTS 100 200 10**

Syntax	**TENDON *number* GETTYPE** **TENDON *number* GETINFO** **TENDON *number* GETMAT** **TENDON *number* GETCOUNT** **TENDON *number* GETAREATEND** **TENDON *number* GETAREADUCT** **TENDON *number* GETBETA** **TENDON *number* GETFRIC**
Meaning	Retrieve data from stucture tendons: - type of tendon - info text - material name - count - area of tendon and area of duct - beta and friction
Examples	**TENDON 101 GETMAT** **TENDON 101 GETAREATEND** **TENDON 101 GETBETA**

Syntax	**TENDON *number* ISGROUTED** **TENDON *number* ISACTIVE**
Meaning	- Retrieve status of tendon
Examples	**TENDON 1 ISGROUTED** **TENDON 2 ISACTIVE**

Syntax	**TENDON *number* ELEM GETALL**
Meaning	Retrieve list of all elements assigned to tendon ***number.***
Examples	**TENDON 1 ELEM GETALL**

Syntax	**TENDON *number* GETALLS** **TENDON *number* GETS AT *elem* BEGIN** **TENDON *number* GETS AT *elem* END** **TENDON *number* GETS AT *elem x/l*** **TENDON *number* GETS AT *elem* S1** **TENDON *number* GETS AT *elem* S2**
Meaning	Retrieve geometrical data for a tendons: - all s values of all definitions points (tendon points) - s value at element start or element end (-9999 if not exist) - s value for a element x/l - get lowest and highest s value for element
Examples	**TENDON 101 GETALLS** **TENDON 101 GETS AT 102 BEGIN** **TENDON 101 GETS AT 102 0.7** **TENDON 101 GETS AT 102 S1**

Syntax	**GET* = GETX or GETY or GETZ** **TENDON *number* GET* *s-value*** **TENDON *number* GET* *s-value* LOCAL *elem*** **TENDON *number* GET* *s-value* LOCAL *elem* CROSS** **TENDON *number* GET* *s-value* LOCAL *elem point-name*** **TENDON *number* GETVEC *s-value*** **TENDON *number* GETVEC *s-value* LOCAL *elem*** **TENDON *number* GETVEC *s-value* LOCAL *elem point-name***
Meaning	Retrieve geometrical data for a tendons: - global coordinate (x/y/z) at a specific s value - coordinate (x/l, ey/ez) at a specific s value local to an element - coordinate (x/l,ey/ez) at s local to an element cross section - coordinate (x/l,ey/ez) at s local to an add. point in the el. cross section - tendon vector at a specific s-value, global oriented - tendon vector at s, oriented relative to the element axis - tendon vector at s, oriented relative to the cross point axis the vector result must be assigned to a variable using the setarr command!
Examples	**TENDON 101 GETY AT 12.345** **TENDON 101 GETX AT 12.345 LOCAL 101** **TENDON 101 GETZ AT 12.345 LOCAL 101 BOT** **TENDON 101 GETVEC AT 12.345 LOCAL 102**

Syntax	**TENDON *number* GETXL *s-value* LOCAL *elem***
Meaning	Retrieve the position within the element ***elem*** for a specific s.
Examples	**TENDON 1 GETXL 3.245 LOCAL 101**

Syntax	**TENDON *number* GETLENGTH**
Meaning	Retrieve the total 3d – length of the tendon.
Examples	**TENDON 1 GETLENGTH**

Syntax	**TENDON *number* GETBEND *s-value*** **TENDON *number* GETBEND *s-value* Y** **TENDON *number* GETBEND *s-value* Z** **TENDON *number* GETBEND *s-value* Y LOCAL *elem*** **TENDON *number* GETRADIUS *s-value*** **TENDON *number* GETRADIUS *s-value* Y** **TENDON *number* GETRADIUS *s-value* Z** **TENDON *number* GETRADIUS *s-value* Y LOCAL *elem***
Meaning	Retrieve bend or radius of tendon at position s either: - 3D radius - radius in global x/y or x/z plane - radius in x/y or x/z plane of an element
Examples	**TENDON 1 GETBEND 3.245** **TENDON 1 GETBEND 3.245 Y** **TENDON 1 GETBEND 3.245 Z LOCAL 102**

Syntax	**TENDON *number* GETLC *lcnumber* AT *elem* BEGIN** **TENDON *number* GETLC *lcnumber* AT *elem* END** **TENDON *number* GETLC *lcnumber* AT *elem position***
Meaning	Retrieve normal force for tendon “number” caused by loadcase “lcnumber” at element "elem". “positions” is 1 or "BEGIN" for begin and “npart” or "END" for end of element.
Examples	**TENDON 101 GETLC 5 AT 102 BEGIN** **TENDON 101 GETLC 5 AT 102 2**

Syntax	**TENDON *number* GETSUP *supfile* AT *elem* BEGIN *minmax*** **TENDON *number* GETSUP *supfile* AT *elem* END *minmax*** **TENDON *number* GETSUP *supfile* AT *elem position* *minmax***
Meaning	Retrieve normal force for tendon “number” caused by superposition “supfile” at element "elem". Parameter **minmax** selects the specific result for a maximum or minimum value. **minmax** must be one of **MAXVx, MAXVy, … MAXMy, MAXMz** or **MINVx, MINVy, … MINMy, MINMz.**
Examples	**TENDON 101 GETSUP “suppos.sup” AT 102 BEGIN MAXNx** **TENDON 101 GETSUP “suppos.sup” AT 102 2 MINMz**

Syntax	**TENDON STRESSLABEL GETALL** **TENDON STRESSLABEL *label* GETITEMS** **TENDON STRESSLABEL *label* ITEM *itemnumber* GETNAME** **TENDON STRESSLABEL *label* ITEM *itemnumber* TENDON** **TENDON STRESSLABEL *label* ITEM *itemnumber* POSITION** **TENDON STRESSLABEL *label* ITEM *itemnumber* TYPE** **TENDON STRESSLABEL *label* ITEM *itemnumber* VALUE**
Meaning	Retrieve stress actions for tendon stress - all stress labels in project - all item numbers for a stresslabel - name of action, tendon-number, position (LEFT/RIGHT), value-type (FORCE, FACTOR), value
Examples	**TENDON STRESSLABEL GETALL** **TENDON STRESSLABE cs1 GETITEMS** **TENDON STRESSLABE cs1 ITEM 2 GETNAME**

1.12.29 LMANAGE:

Syntax	**LMANAGE GETALL**
Meaning	Get a list of all load manage entries.
Examples	**LMANAGE GETALL**

1.12.30 LCASE:

Syntax	**LCASE GETALL** **LCASE GETALLUSER**
Meaning	Get a list of all loadcases for which results exist. Get a list of all user-defined loadcases.
Examples	**LCASE GETALL** **LCASE GETALLUSER**

Syntax	**LCASE *lcase* GETINFO** **LCASE *lcase* GETLOADINFO** **LCASE *lcase* GETTYPE** **LCASE *lcase* GETLSETS** **LCASE *lcase* GETLSET *number* FACTOR** **LCASE *lcase* GETLSET *number* FORMULA** **LCASE *lcase* GETLSET *number* INCREASE**
Meaning	Get a list of all loadcases for which results exist. Get a list of all user-defined loadcases.
Examples	**LCASE 500 GETINFO** **LCASE 500 GETLOADINFO** **LCASE 500 GETLSETS** **LCASE 500 GETLSET 501 FACTOR**

1.12.31 LSET:

Syntax	**LSET GETALL**
Meaning	Get a list of all loadsets.
Examples	**LSET GETALL**

Syntax	**LSET *lset* GETINFO** **LSET *lset* GETITEMS**

Meaning	Get info for loadset.
	Get a list of all loadset items. This list consists of multiple entries which must be assigned to an tcl-array variable. The content of this array variable is: “from”, “to”, “step”, “proj”, “ndata”, “data1”, “data2”, …
	Example:
	foreach item [LSET 500 GETITEMS] {
	setarr lsetitem $item
	LIST “from: $lsetitem(from), … data1: $lsetitem(data1)”
	}

Examples	**LSET 500 GETINFO** **LSET 500 GETITEMS**

1.12.32 STAGE:

Syntax	**STAGE GETALL** **STAGE GETIRST** **STAGE GETLAST** **STAGE GETTOTAL**
Meaning	Get a list of all stages, get the number of the first / last the number of stages.
Examples	**STAGE GETALL** **STAGE GETIRST**

Syntax	**STAGE *stage-number* GETTIME** **STAGE *stage-number* GETDURATION** **STAGE *stage-number* GETINFO** **STAGE *stage-number* GETACT** **STAGE *stage-number* GETDEACT** **RMSERIE elementserie**
Meaning	Get informations for stage. Get a list of all elements activated / deactivated in stage ***stage-number***. To produce a FROM – TO – STEP list, use the RMSERIE command with the result!.
Examples	**STAGE 1 GETACT**

Syntax	**STAGE** ***stage-number*** **GETACTION TOTAL** **STAGE** ***stage-number*** **GETACTION** ***action-number*** **NAME** **STAGE** ***stage-number*** **GETACTION** ***action-number*** **INPUT1** **STAGE** ***stage-number*** **GETACTION** ***action-number*** **INPUT2** **STAGE** ***stage-number*** **GETACTION** ***action-number*** **OUTPUT1** **STAGE** ***stage-number*** **GETACTION** ***action-number*** **OUTPUT2** **STAGE** ***stage-number*** **GETACTION** ***action-number*** **INFO** **STAGE** ***stage-number*** **GETACTION** ***action-number*** **TIME** **STAGE** ***stage-number*** **GETACTION** ***action-number*** **DURATION**
Meaning	Get total number of action or action information for n-th action.
Examples	**STAGE 1 GETACT**

1.13 Scope: RMFILE

After the execution of **RMFILE "filename",** this scope is entered. Within this scope, the following commands are available:

- **RMFILE END:** To end this scope.

- **LINE:** Add a line to the file.

1.13.1 LINE:

Syntax	**LINE "content"**
Meaning	Add a line to the file. If one of the following characters are use in the content, must be replaced by the character preceeded by a backslash (\): [ ] { } " $
Examples	**LINE "PLFTXT LB 10.000 -12.000 0.000000 \"Scale 1:1000\""**

2 Data conversion from RM7 to *RM2000*

It is possible to export most of the important input data prepared for a project using RM7 into the *RM2000* database directly.

The RM7 project directory must be opened before starting the data transfer and after generating the structural system and the tendon layout a SYSAK file run will activate the whole structural system. Only the materials and cross-sections used in the project will now be transferred.

Note: *For exporting data from RM7 to RM2000 V7.52.02 or higher must be used. If it was not shipped with RM2000, order it from your support office!*

2.1 What can be transferred?

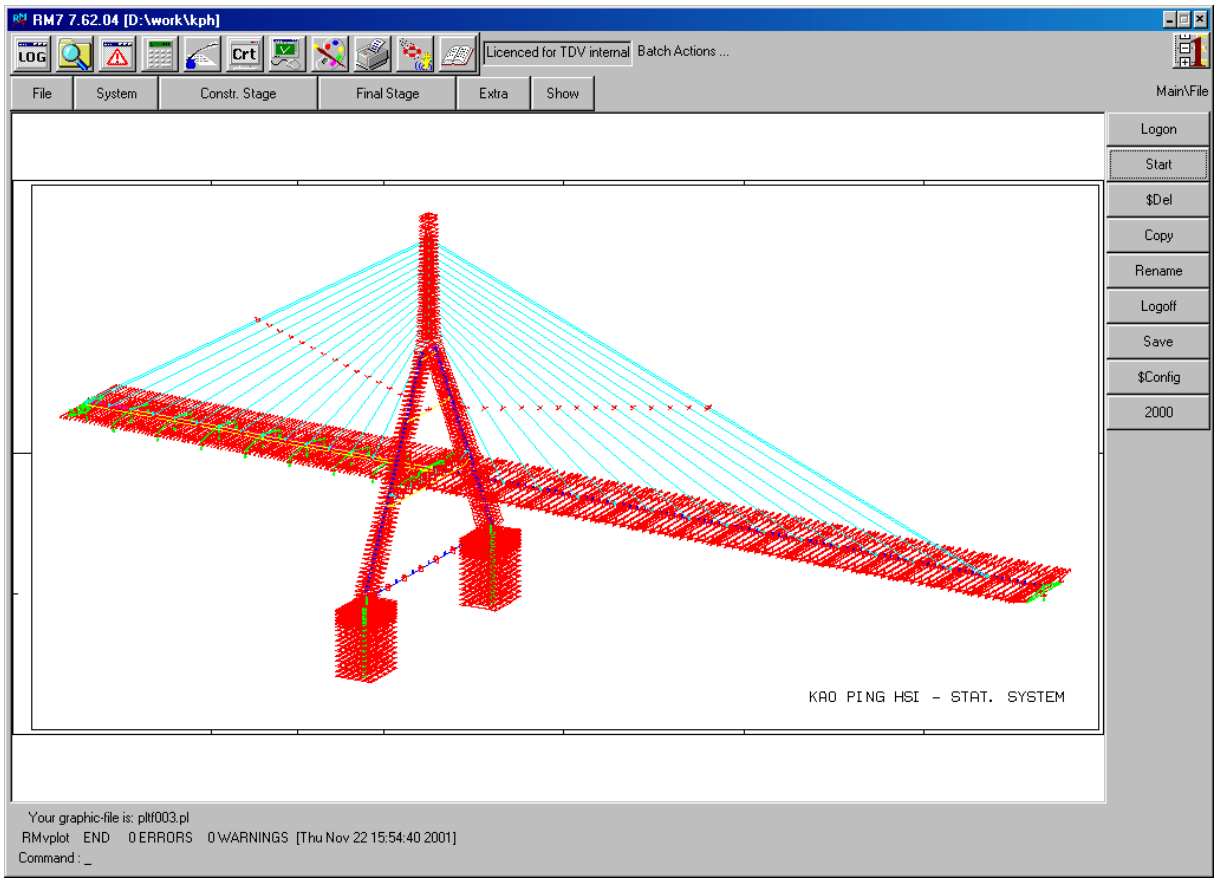
The following data can be transferred to the *RM2000* database:

- Node definitions: node coordinates, node supports, node support eccentricities, node orientation, node mass…….
- Element information: element assignments, element eccentricities, element hinges, element orientation…….
- Cross-sections (both those prepared interactively and numerically) exept thin walled sections generated with QWOST
- Material assignment…….
- Tendon geometry…….
- Composite section definition (use the environment variable SET COMP=1)…….

2.2 How to do it?

1. Change into the appropriate RM7 project directory
2. Start RM7
3. Activate the whole system
4. Select 

: the following menu will appear on the right:



5. Select 

6. The *RM2000* database will be created:

   - rm-bin01.rm8
   - rm-bin02.rm8
   - rm-bin03.rm8
   - rm-bin04.rm8

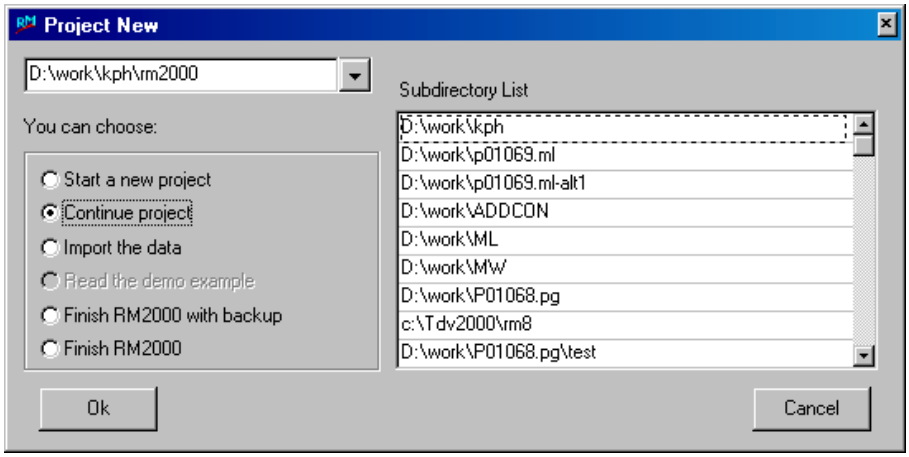
   The above *RM2000* database will remain in the same directory as the RM7 project from which it was converted – see "How to continue in *RM2000*" for new directory recommendations.

2.3 How to continue in *RM2000*?

To avoid data confusion, transfer or copy the newly created *RM2000* database into a new project directory before starting *RM2000* (in that directory!).

Select the new project directory by first selecting the 'Project New' window arrow – see screen shot below:

7. Now start *RM2000* by choosing 'continue project'……..



8. Complete the input data by defining Load Management, Load Sets, Loading Cases, LOADS AND CONSTR. SCHEDULE, Tendon Schedule…….(Refer to "*RM2000* Getting Started" for guidance on the preparation of this data)