SOFiSTiK FEA 2020

TEDDY – TEXT EDITOR AND CADINP – SYNTAX

INTRODUCTION

Version October 2019

www.guidoaltersberger.com

TABLE OF CONTENTS

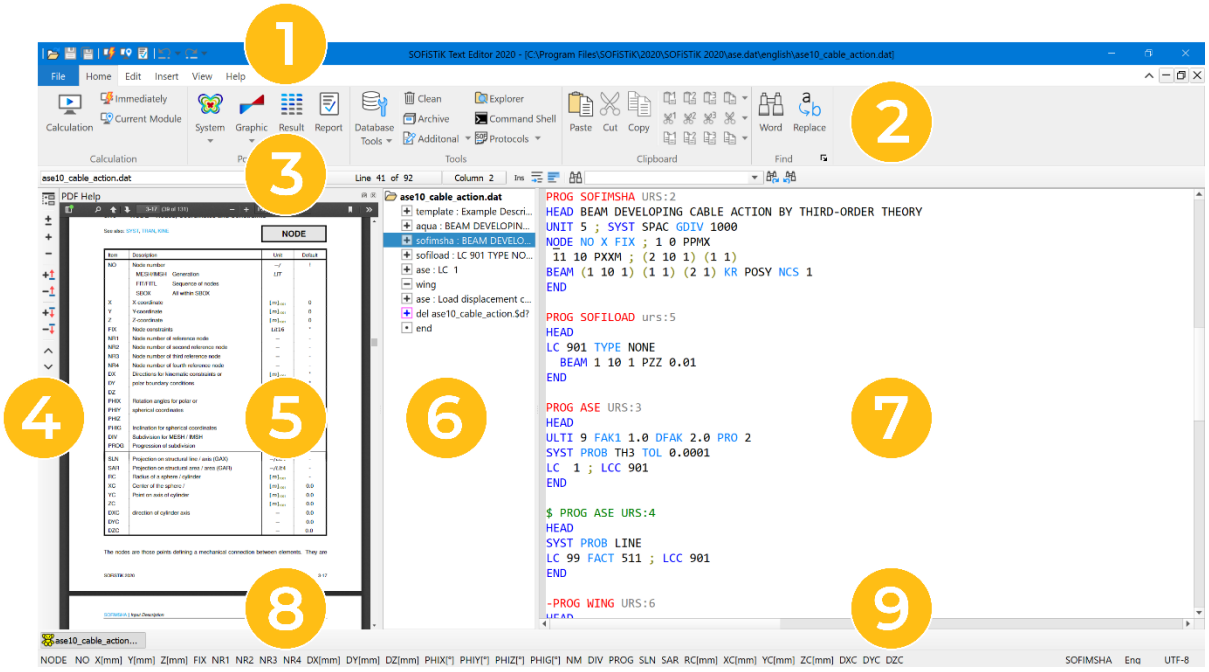
TEXT EDITOR - TEDDY

TEDDY is a text editor which fully supports and utilises the CADiNP language. It provides you with an integrated help system to list all available commands.

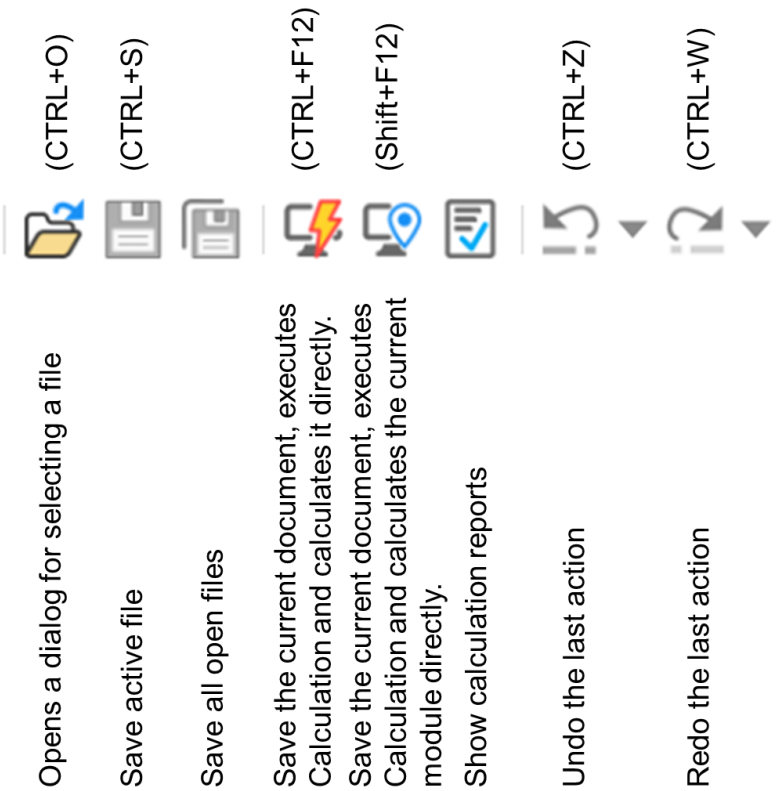
START TEDDY

TEDDY is a windows application and can be started by double-click on the program icon located on the desktop as well as on a DAT-file starts the program TEDDY.exe.

SECTIONS

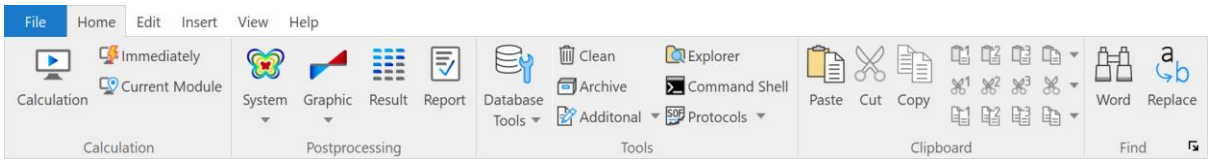


1 MENU

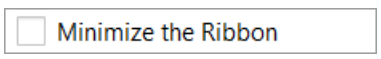


2 RIBBONS

Since version 2020 the application navigation is done using ribbons. “File, “Home”, “Edit”, “Insert”, “View” and “Help” are available.

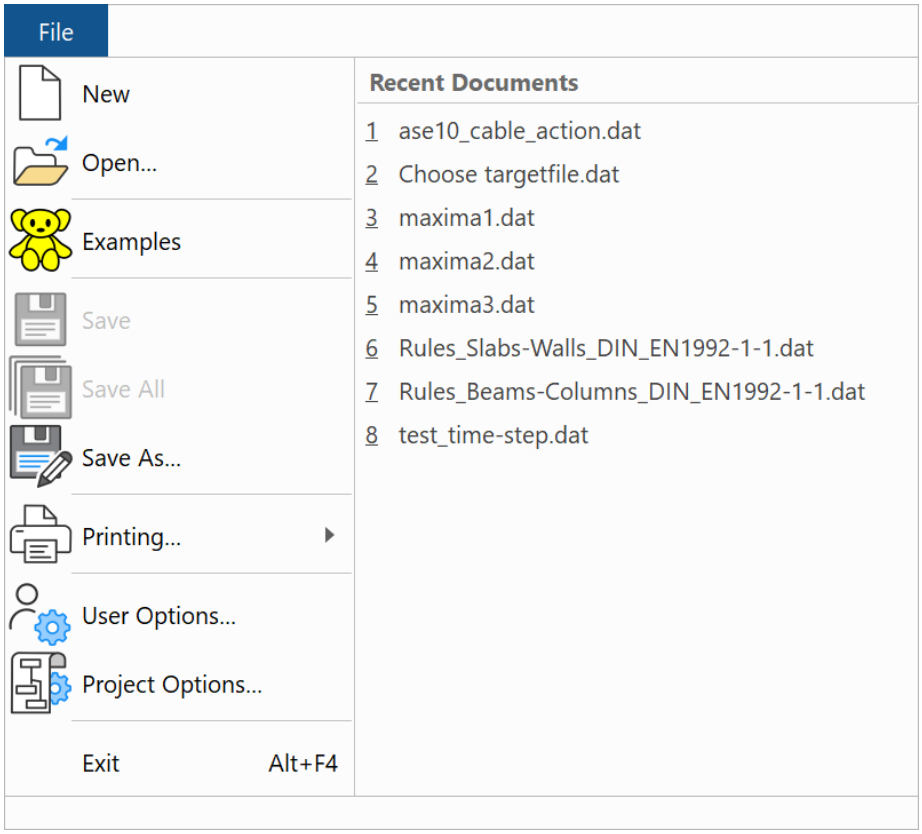


Right-click anywhere at the Ribbon allows minimising the ribbon bar to the names.



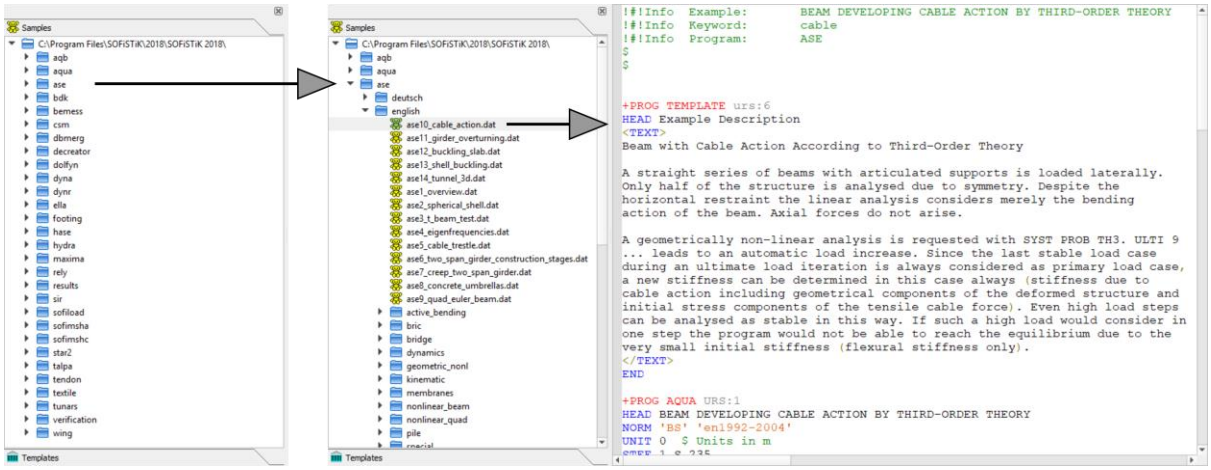
FILE

The File menu contains the default commands such as “New”, “Open”, “Save”, “Save All”, “Save As”, “Print” and “Exit”.



+ EXAMPLES

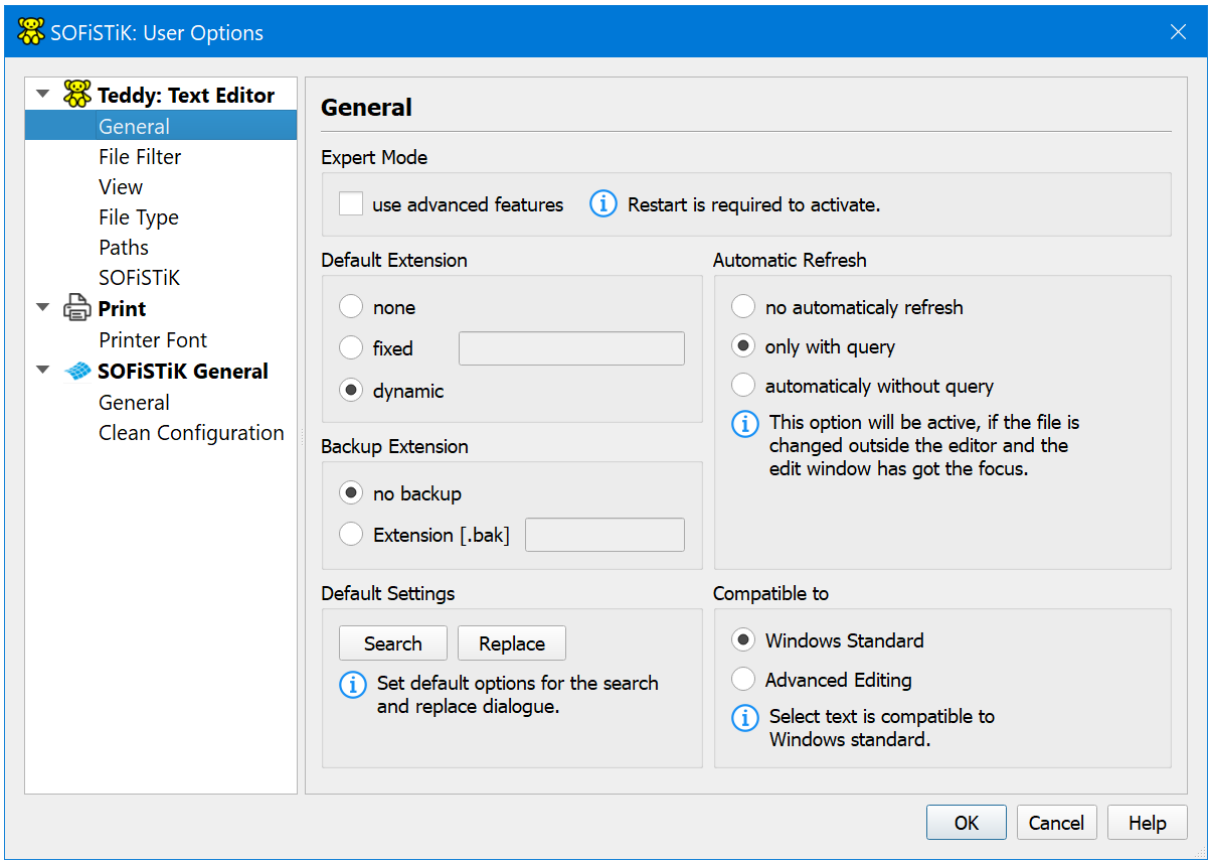
Which this command you'll get access to available teddy examples for all modules and different topics. They are organised in an example browser on the left-hand side of the Input area of Teddy.



As the example files are read-only – saving to a different folder is required to run them successfully.

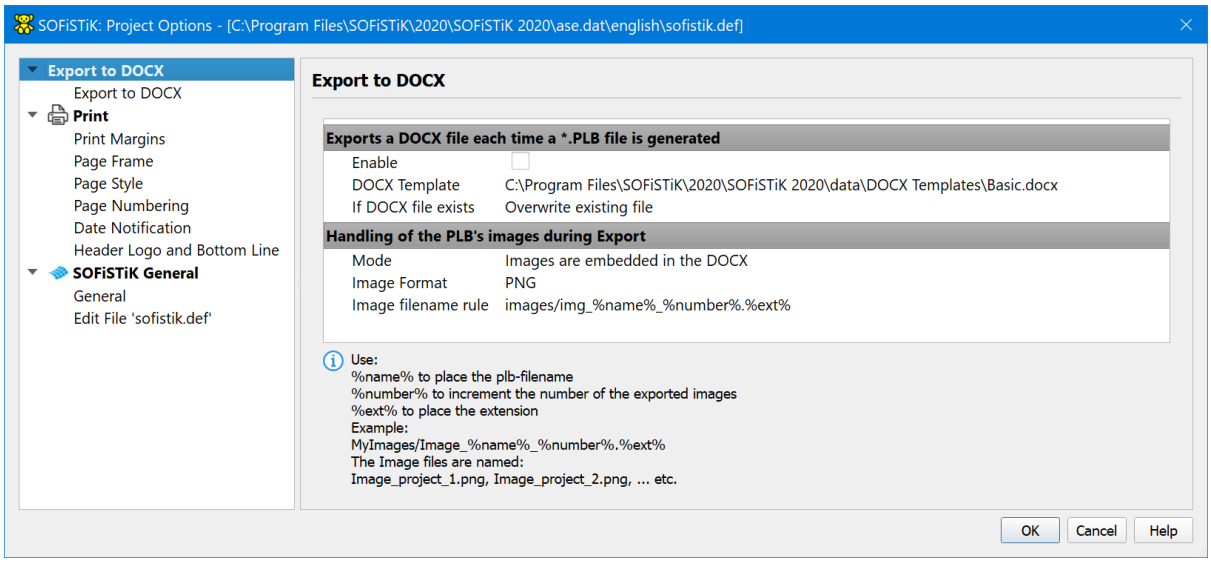
+ USER OPTIONS

The user options of Teddy allow you to adjust the representation, paths, fonts to print the code or even switching between Comma and Decimal Point for the input.



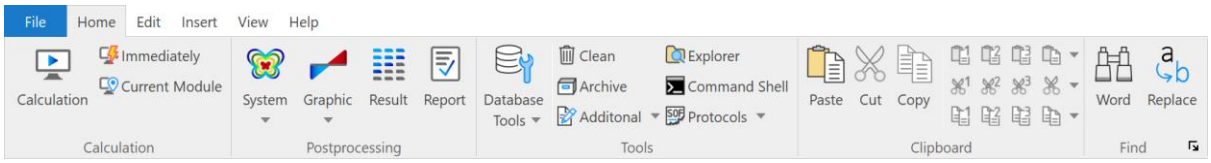
+ PROJECT OPTIONS

Within the Project Options setting such as generating a DOCX report, adjustments of the Report Browser layout and input/output language are available.



HOME

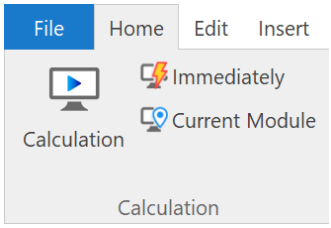
The “Home” ribbon comes with the tabs “Calculation”, “Postprocessing”, “Tools”, “Clipboard” and “Find”.



+ CALCULATION

The command “Immediately” starts the calculation for all activated modules of the input file. Activated Modules are those marked with a “+” in front of the PROG command.

The command “Current Modul” performs the calculation of the current module you’re working at.

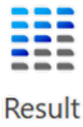


The command “Calculation” opens the “SOFiSTiK Calculation 2020” window. To learn more about it, read chapter SOFiSTiK Calculation 2020.

+ POSTPROCESSING

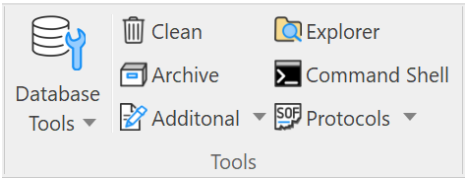
Open the application relevant for the “Postprocessing” from here.

Command	Description
	Open Interactive System Visualisation
	Plots and Visualisation of Results

	Show Tabular Results
	Show Calculation reports

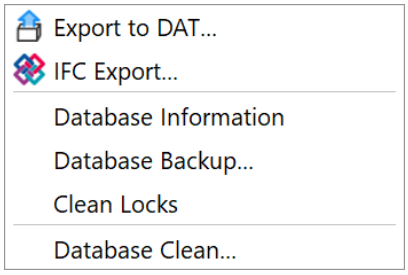
+ TOOLS

The tools available are mainly related to the SOFiSTiK database. However, there are some commands to access projects files more comfortable.



- DATABASE TOOLS

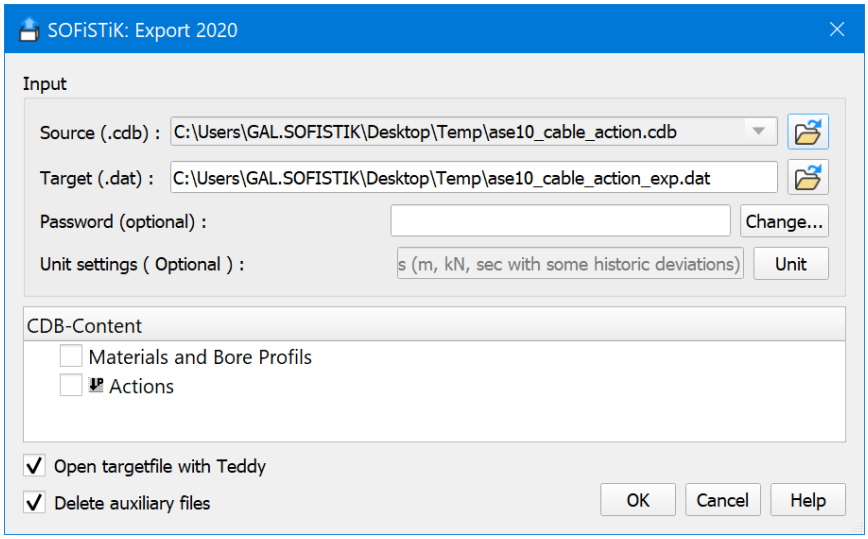
Database related tools are accessible through the “Database Tools” command.



- Export to DAT …

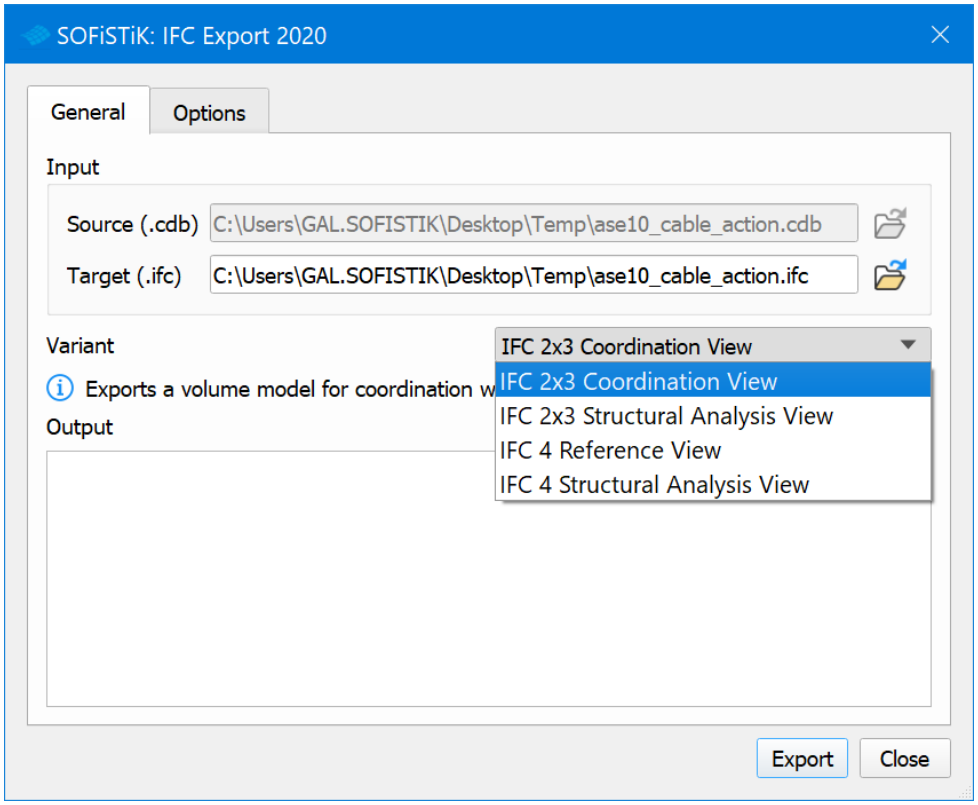
The “Export to DAT …” command allows converting information from the database to a text input file (dat-file).

You’ll get the possibility to refine the export in the “SOFiSTiK: Export 2020” window after starting the command. After confirming the export, a dat-file is generated and saved in the selected target folder.



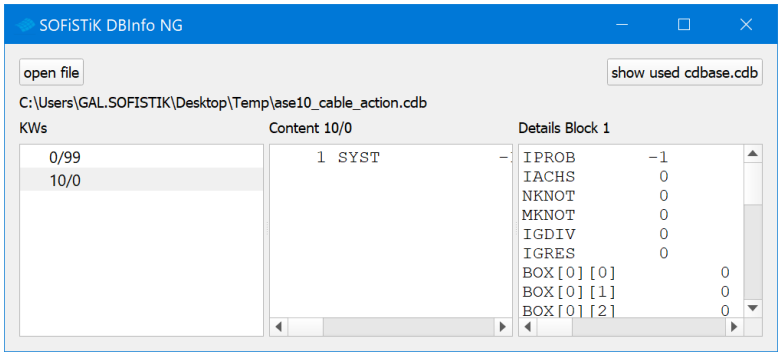
- IFC Export

Exports the model to IFC file format.



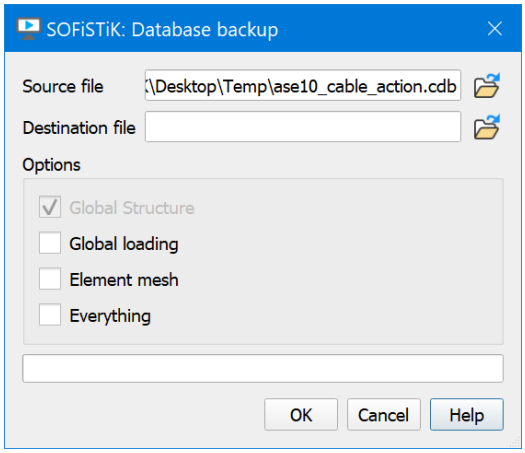
- Database Information

The “Database Information” command allows getting insights into the database structure. Which is useful when extracting information through the available interfaces such as C#, Python or the @KEY access within CADiNP.



- Database Backup

Although the database can be generated at any time with the input files, it's recommended to make a backup. The "SOFiSTiK: Database Backup" is a useful tool to create this backup. Alternative you can also copy the *CDB to a backup folder.

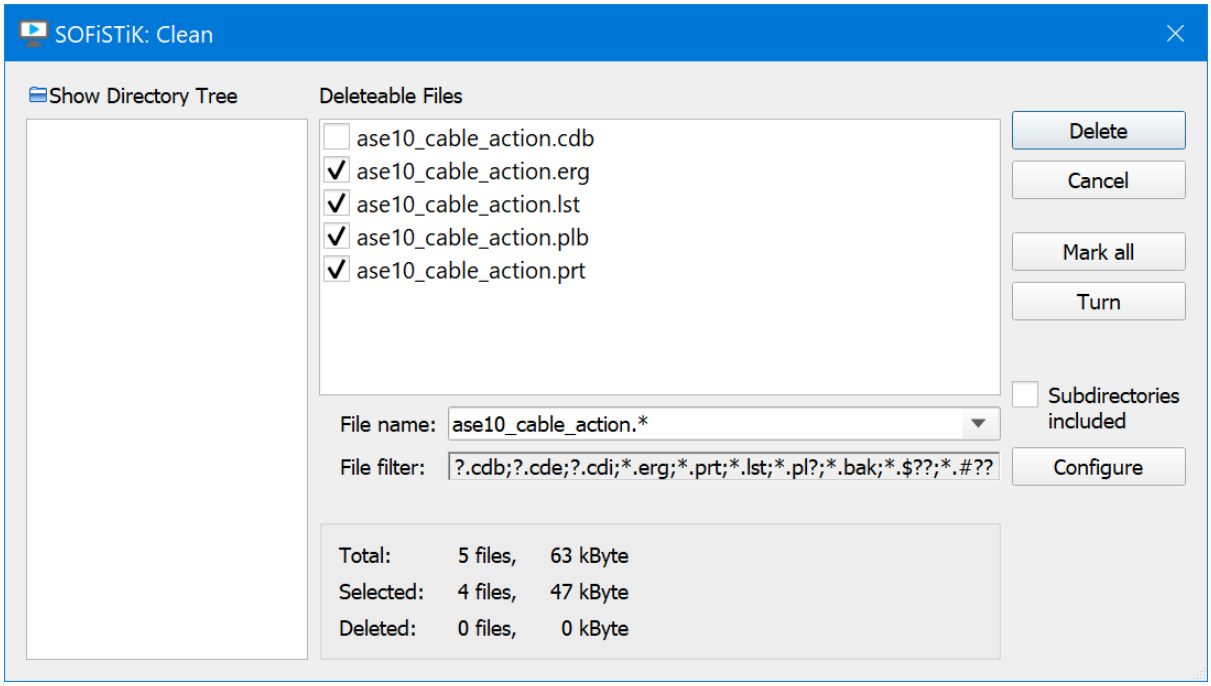


- Database Clean

The "Database Clean" command is a controversial one. As cleaning the database from elements, load cases or actions isn't reducing the size of it.

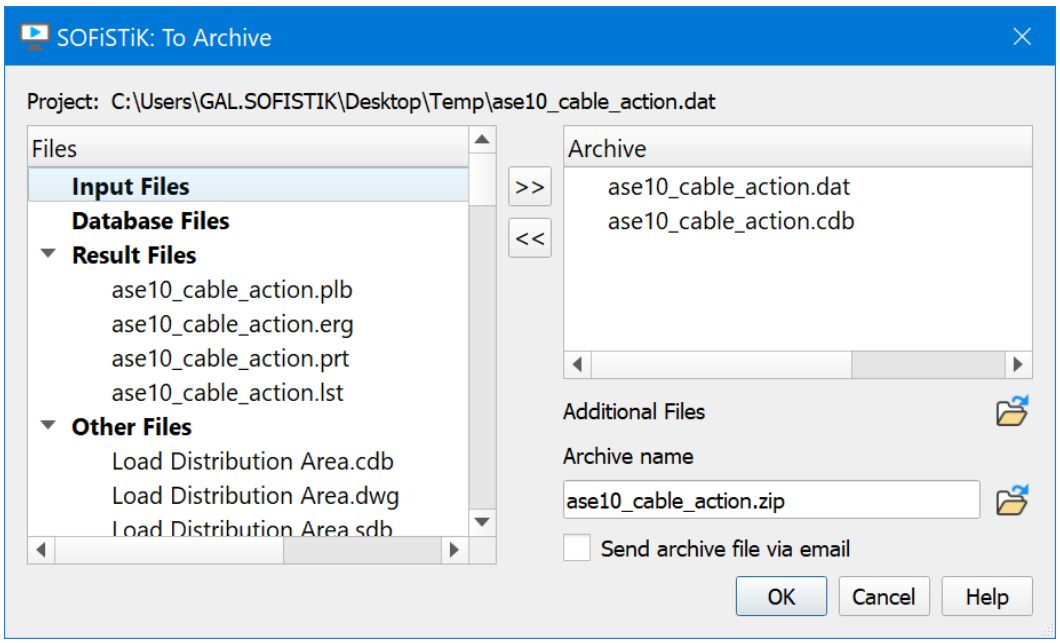
- CLEAN

The "Clean" command allows you to clean up your project folder. Refining the selection of the different files if possible, in the dialogue box "SOFiSTiK: Clean".



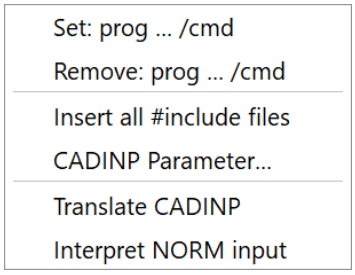
- ARCHIVE

To share project relevant files with colleagues or even to share them with the SOFiSTiK support team. The "Archive" command is a comfortable way of doing that. Foles can be selected in the dialogue box "SOFiSTiK: To Archive".



- ADDITIONAL

There is a set of additional options available in this command.



- EXPLORER

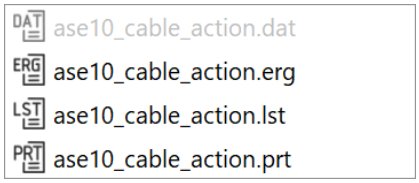
The “Explorer” command is a quick way to open the project directory.

- COMMAND SHELL

“Command Shell” allows to easily access the windows command line and navigates directly to the project folder.

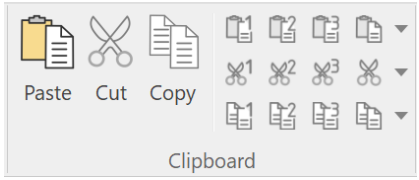
- PROTOCOLS

Available protocols within the project folder can be easily accessed from this command. Files such as *.dat, *.erg, *.lst and *.prt can be opened straightforward in the Text Editor.



+ CLIPBOARD

Especially when you’re using a text interface copy/paste commands are welcome options. The “Clipboard” tab provides you with the most common ones, but also the capability to save snippets temporally to different register.

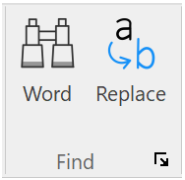


Command	Description	Shortcut

	Paste text from clipboard	CTRL+V
	Cuts out selected text and copies it to the clipboard	CTRL+X
	Copies the selected text to the clipboard	CTRL+C
	Paste text from register #1	SHIFT+CTRL+1
	Cuts out the selected text and saves it to register #1	
	Copies the selected text and saves it to register #1	CTRL+1

+ FIND

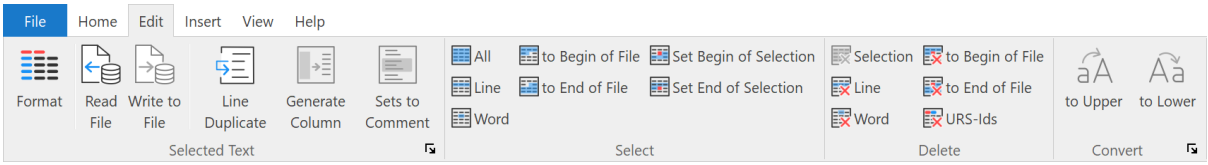
Allows you to find and replace text.



Command	Description	Shortcut
	Search for the word at the current cursor position.	CTRL+F3
	Search and replace text in the document	CTRL+Q+A or F4

EDIT

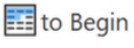
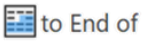
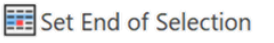
The “Edit” Tab comes with plenty of selection and formatting tools.



+ SELECTED TEXT

Command	Description	Shortcut
 Format	Formats selected text or current module in a proper tabular layout.	CTRL+B
 Read	Insert file as text.	CTRL+K+R or ALT+R
 Write to	Save selected text as a file.	CTRL+K+W or ALT+W
 Line Duplicate	Duplicates current line to next line.	CTRL+K+: or F8
 Generate Column	Generate a column with numbers. Columns selection in “Edit Bar” must be active.	CTRL+K+Z or ALT+Z
 Sets to Comment	Sets selected lines to comment and vice versa.	CTRL+O+C or ALT+C

+ SELECT

Command	Description	Shortcut
 All	Select the entire text.	CTRL+A
 Line	Select text of the current line	CTRL+K+L or ALT+L
 Word	Select word at the current cursor position.	CTRL+K+V or ALT+V
 to Begin of File	Select text from the top up to current cursor position.	CTRL+K+POS1
 to End of File	Select text from the bottom up to the current cursor position.	CTRL+K+END
 Set Begin of Selection  Set End of Selection	Defines the start and end position of the selection.	CTRL+K+B or CTRL+ F7 CTRL+K+K or CTRL+F8

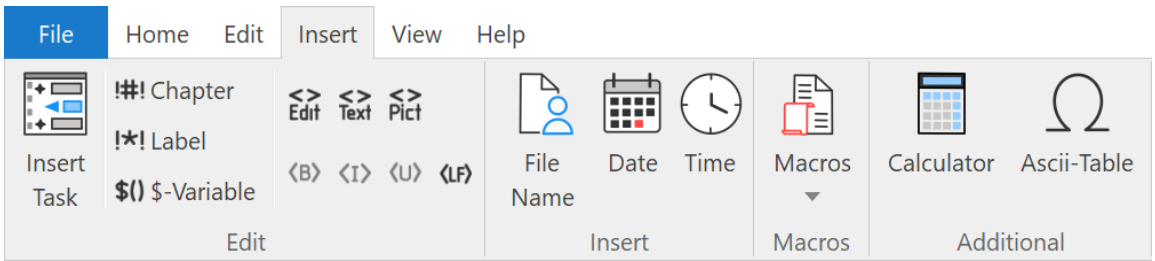
+ DELETE

Command	Description	Shortcut
Selection	Delete selection.	CTRL+K+Y or DEL
Line	Delete current line	CTRL+Y
Word	Delete word at the current cursor position	CTRL+T
to Begin of File	Delete entire text before current curser position	
to End of File	Delete entire text after current curser position	
URS-Ids	Delete all URS-Ids	CTRL+O+Y

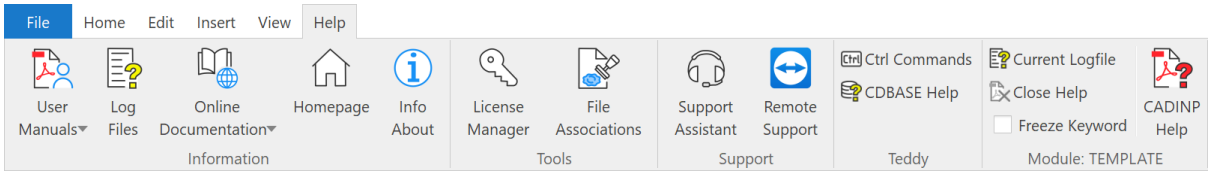
+ CONVERT

Command	Description	Shortcut
to Upper	Convert selected text into capital letters.	CTRL+Q+U
to Lower	Convert selected text into lower letters.	CTRL+Q+L

INSERT



HELP



Within the “Help” tab you can find “User Manuals”, “Logfiles” and the linked to the “Online Documentation”. Also, useful applications such as the “License Manager” as well as the “Support Assistant” and “Remote Support” can be accessed.

It’s recommended to use the SOFiSTiK Support Wizard command to raise a support ticket request.

More Text Editor specific commands are within the section “Teddy” and “Module: …”.

+ TEDDY

- CTRL COMMANDS

In addition to access commands through the ribbons and menu shortcuts are available. I just picked a few of the most used shortcuts to represent in the below chapters.

- Function Keys

Most important functions are accessible through the function keys:

Key	Description
F1	Help to Active Input Record
F2	Search
F3	Repeat Search
F4	Search and Replace
F5	Mark Block Start or End

Key	Description
F6	Copy Marked Block
F7	Move Marked Block
F8	Duplicate Current Line
F9	Change Window
F12	Start Analysis

- Combinations with the CTRL key (selection)

Key	Description
Ctrl + A	Mark All
Ctrl + C	Copy Selected Block
Ctrl + V	Paste Selected Block
Ctrl + X	Cut Selected Block

Ctrl + S	Save File
Ctrl + Y	Delete File
Ctrl + Z	Undo

- Combinations with ALT key (selection)

Key	Description
Alt + L	Select Several Lines (Block of Lines)
Alt + B	Select Several Columns (Column Block)
Alt + H	Remove Block Markers
Alt + R	Read File as Block
Alt + W	Write (Selected) Block to File
Alt + Z	Copy Selected Block

- CDBASE HELP

To access information in the database using @KEY, the “CDBASE Help” is an essential document. It gives you insights at which location I the database structure the information is saved.

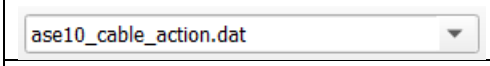
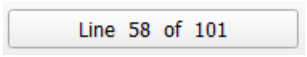
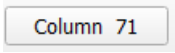
+ MODULE: …

The section “MODULE: …” gives you additional information about the current module. Depending on in which program module you work - the “…” will be replaced with the module name. Also, the Interactive CADiNP Help can be controlled from here.

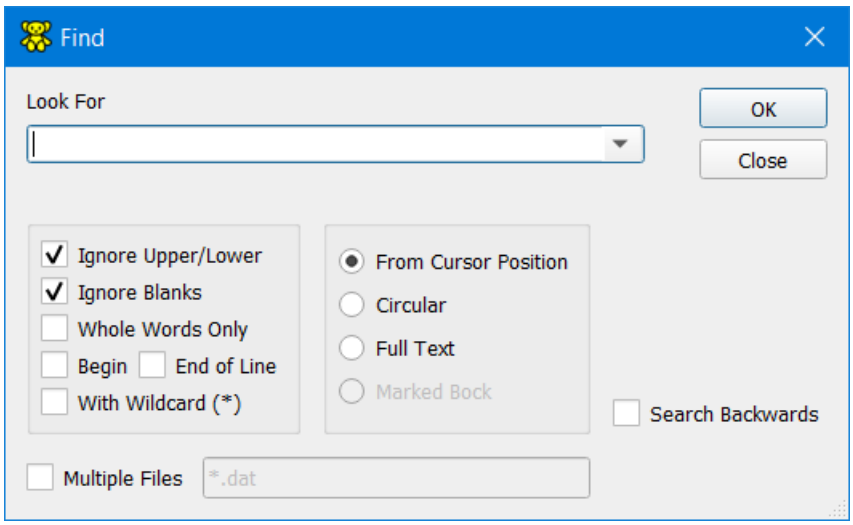
Command	Description
Current Logfile	Shows the latest logfile of the active module.
CADINP Help	Open the Interactive CADiNP Help
Close Help	Close the Interactive CADiNP Help
Freeze Keyword	Suppress live command tracking in the Interactive CADiNP Help

3 EDITBAR

The “Editbar” is useful to open recent dat files, navigate to a specific location in the input file and to search.

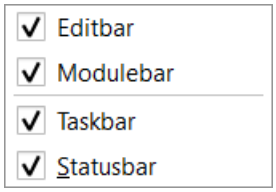
Command	Description	Shortcut
	Recent file	
	Shows current line Jump to a specific line Click to the left end of the button - jump to start of the input file. Click to the right end of the button - jump to the end of the input file	CTRL+Q+N or CTRL+G
	Go to column Shows current column Jump to a specific column	
	Switch between the insert and overwrite mode	
	Autotab on/off	
	Select text as a line or column block.	CTRL+K+N
	Search forward Search backwards	F3 SHIFT+F3
	Set find options and search text. The “Find” dialogue box opens.	CTRL+Q+F or F2

“Find” dialogue box



Errors during the calculation are documented in red letters in the Result Browser. Besides details about the error itself, the location of the error in the input is printed by line and column number. Using the numbers to jump to the location in big input files is useful.

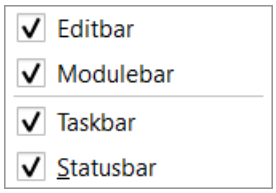
Right-click on the “Editbar” opens the context menu to activate/deactivate the bar.



4 MODULEBAR

Command	Description	Shortcut	Comment
	Module tree		Module Tree on/off. 6 Moduletree
	+/- Prog		Switch current module on/off
	+ Prog		Switch all Modules on
	- Prog		
	Up to curser + Prog		Activate all modules from the first to the current one
	Up to curser - Prog		Deactivate all modules from the first to the current one
	From curser + Prog		Activate all modules from the current to the last one
	From curser – Prog		Activate all modules from the current to the last one
	Previous module Next Module	SHIFT+CTRL+J CTRL+J	Jump between different modules one by one.

Right-click on the “Modulebar” opens the context menu to activate/deactivate the bar.



5 INTERACTIVE HELP

Open the help manual from the “Help” ribbon and the tab “Module:”. It’s the last command “CADINP Help”. Alternatively, hit the “F1. See Chapter Module: … for available commands.

The page of the manual - matching the current input command of the current program opens. The names of the manuals, which are in PDF format, are composed with the program name and a particular character sequence (*_0 German or *_1 English) to indicate the language. In a navigation window, you can see the table of contents, which allows you to jump to the theoretical - or the example sections in the manuals. As soon another location in the input area has selected the location in the manual help changes accordingly to the relevant manual and information.

It is possible to deactivate the automatic tracking of the location in the help manual by activating the “Freezing Keyword” checkbox.

You can close the help manual with the “Close Help” command.

6 MODULETREE

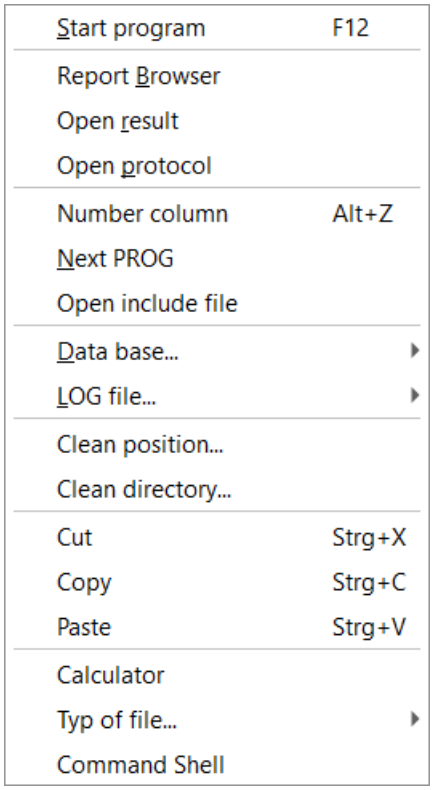
All defined modules, chapters, labels and system commands show up in the task tree. Activating a deactivation of modules and system commands can be done by ticking the (+) or (–) in front of the name. Alternatively, by using the “Modulbar”, see chapter 4 Modulebar.

Keeping the order of the modules reasonable is essential. No analysis can be performed if there are no loads available.

7 INPUT AREA

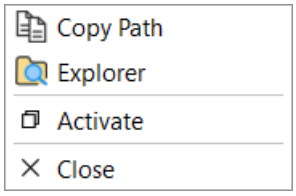
This is the place where the CADiNP Syntax needs to be entered.

Right-Click somewhere at the input area opens the context menu with plenty of handy commands.



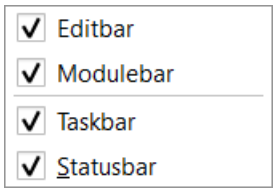
8 TASKBAR - FILE / PROJECT TABS

All open files/projects in TEDDY are shown in the taskbar. By right mouse click on a tab, the context menu with further options opens.



Confirming the selected command in the context menu can be done either by left mouse click or by right mouse click.

Right-click on the “Taskbar” opens the context menu to activate/deactivate the bar.

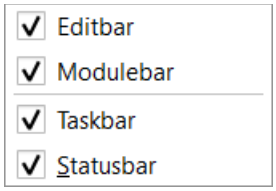


9 STATUS BAR - COMMAND SUMMARY, ACTIVE MODULE NAME AND UNIT INFORMATION

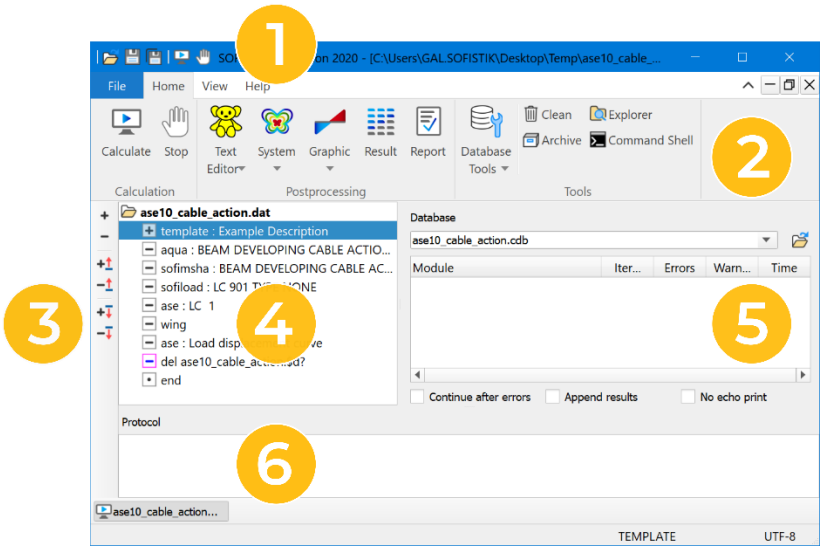
The status bar divides into two parts

- Summary of available items of the entered command (left-hand side) and
- Active module and information about chosen Units (right-hand side).

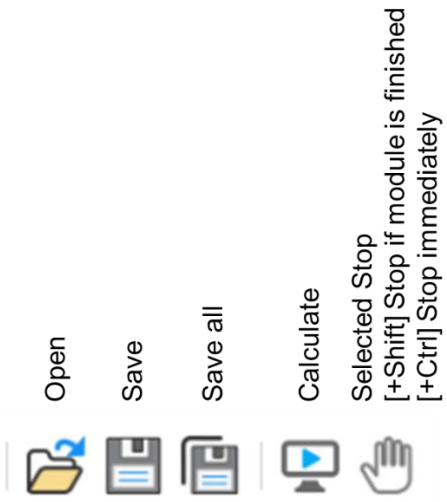
Right-click on the “Statusbar” opens the context menu to activate/deactivate the bar.



SOFISTIK CALCULATION 2020



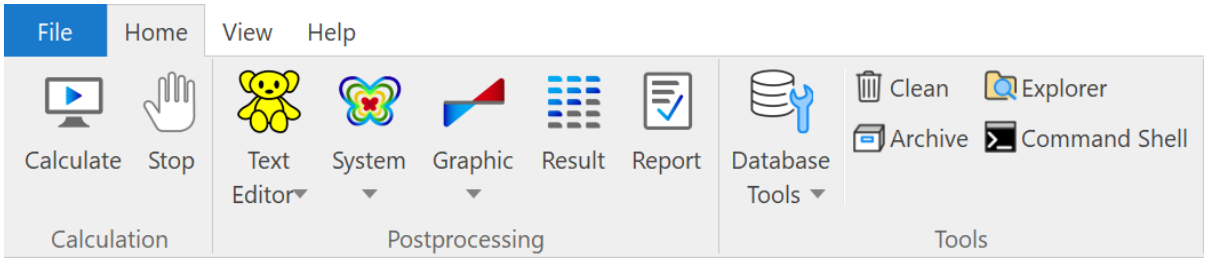
1 MENU



2 RIBBONS

This section comes with the available ribbons of “SOFiSTiK Calculation 2020”. The selection is slightly different as in the Text Editor itself.

HOME



CALCULATION

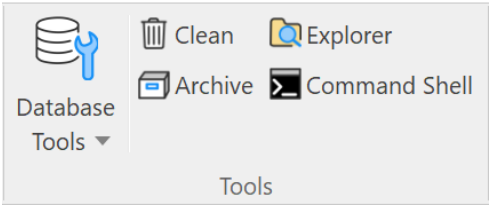
Command	Description	Shortcut
 Calculate	Calculate all activated modules	F12
 Stop	Selected Stop	[+Shift] Stop if module is finished [+Ctrl] Stop immediately

POSTPROCESSING

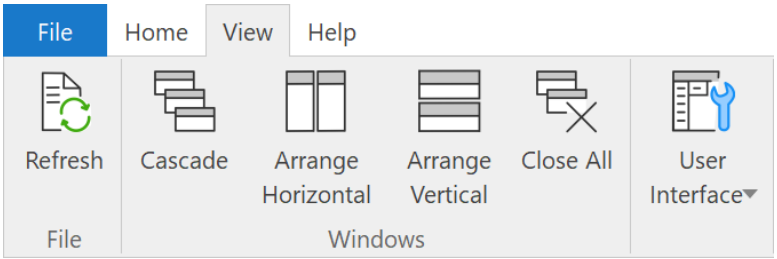
Command	Description
 Text Editor	Open Text Editor
 System	Open Interactive System Visualisation
 Graphic	Plots and Visualisation of Results
 Result	Show Tabular Results

 Report	Show Calculation reports

TOOLS



VIEW



FILE

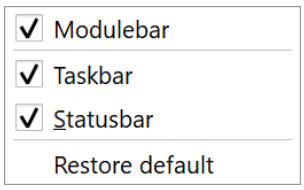
The “Refresh” command does an update of the “Module List”.

WINDOWS

Arrange the windows in “Cascades”, horizontally, vertically or close them all.

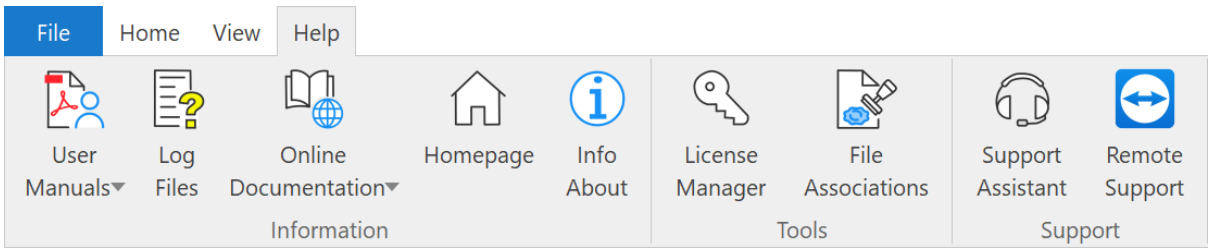
USER INTERFACE

By default, all toolbars are activated. By unticking the checkboxes, you can switch them off individually.



HELP

The “Help” tab is almost similar to the one available in the Text Editor Teddy. Only the text Editor relevant sections “Teddy” and “Module: Template” isn’t there.



3 MODULE BAR

The commands within the “Module Bar” allows to activate/deactivate modules.

Command	Description
+	Activate current program module
−	Deactivate current program module
+↑	Activate all program module upwards exclusive current.
−↑	Deactivate all program module upwards exclusive current.
+↓	Activate all program module downwards exclusive current.
−↓	Deactivate all program module downwards exclusive current.

4 MODULE LIST

The module list allows to double-check preselected modules as well as to activate/deactivate modules before starting the calculation.

5 CALCULATED MODULES

I this section you get an overview of the performance of the calculated modules. Furthermore, an option to select a database if the “SOFiSTiK Calculation 2020” window is used without interaction of the Text Editor.

Below the program module list three checkboxes are available.

Command	Description
☐ Continue after errors	Proceed with the calculation, although error my occur. This option should only be used if you’re aware of the background of the error.
☐ Append results	Add new results to the existing report.
☐ No echo print	Suppress the text output.

6 PROTOCOL

At every single calculation, a protocol is generated. It gives detailed information about the program module performance as well details about errors and warnings. The protocol is saved as *.prt file within the project folder.

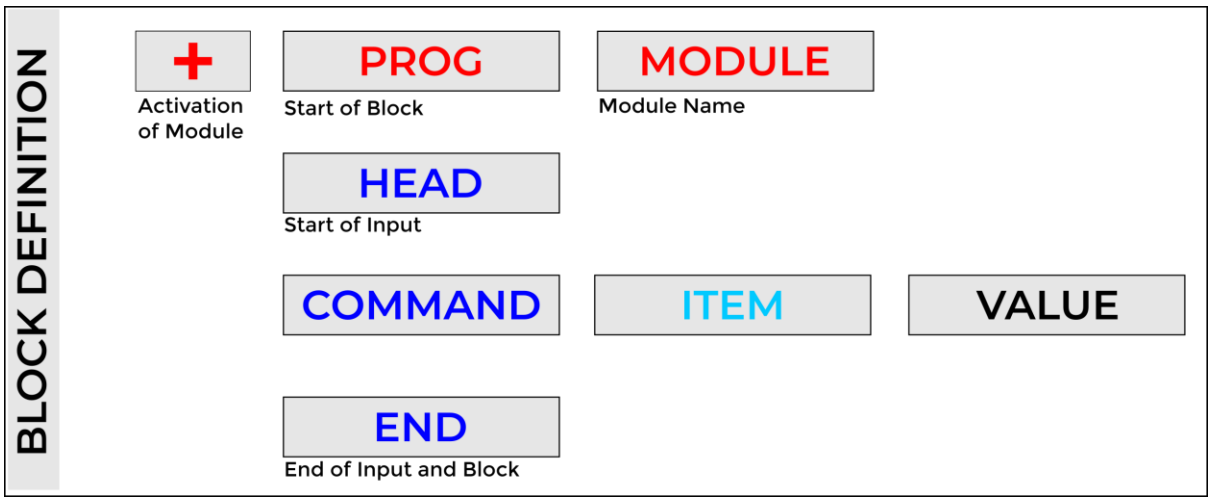
CADINP – INPUT LANGUAGE

Teddy recognised the input commands of CADiNP. It helps to keep track of the input as well as to verify not know commands in different modules immediately.

STRUCTURE OF TEXT INPUT

The general input divides into four main parts which are contained in the Block Definition.

- The begin or start of the Input block is initiated by PROG flowed by the MODULE name. The prefix + or – allows to activate or deactivate the input block.
- HEAD commences the start of the input.
- The input itself is done by different COMMANDS, ITEMS and VALUES depending on the selected module.
- END defines the final input line of the input as well as the block definition.



INPUT SYNTAX

The CADiNP syntax is standardised for all available SOFiSTiK modules.

LET AND STO VARIABLES

We distinguish between two different types of variables #LET and #STO. Both must be defined within a module.

- LET# variables are valid within the module or block and are saved temporarily only.
- STO# variables are valid within as well as outside the module or block. They are global variables and are saved in the CDB (Database).

Both are characterised by a # prefix followed by the name of the variable (up to 16 characters of an alpha-numerical text, 1st character must be a letter) and an optional index.

The name of variables must not contain special characters or collide with a reserved sequence of characters such as SIN, COS, SQR, VERSION.

The unit of a variable is depending on the context, where this variable is used. Thus, it is not possible to use the same variable for a section definition in [mm] and a system definition in [m]. However, it is possible to assign a unit to a variable. The given value is converted immediately, and the value can be used everywhere.

DIFFERENT TYPES OF INPUT VALUES

The following inputs are valid for #LET and #STO variables.

- Numerical
  LET#A 5.1 …Variable #A gets a single value 5.1[-] (No unit).
  LET#A 5.1[mm] …Variable #A gets a single value 5.1[mm] (Unit will be considered).
  LET#10 4,5,6 …generates #10 - value 4, #11 – value 5, #12 – value 6

- Text
  LET#A ‘Test’…Variable #A gets the text ‘test’.

- Vector
  LET#A 4,5,6 …Defines a vector A[4,5,6]
  LET#A(0) 7 …Defines a vector A[7]
  LET#A(2) 9 …Defines a vector A[0,0,9]
  LET#A(0)+1 8 …Defines a vector A[0,8]

A more detailed description of variables, types of values and act and occupied variable names can be found in the Basics Manual (sofistik_1.pdf) chapter 8.2.14.

ARITHMETIC EXPRESSIONS AND FUNCTIONS

In place of a numerical value, any arithmetic expression can be used. The expression may contain parentheses, but no separation characters.

- Allowed operator

Operator	Description
+ -	Addition, Subtraction
* /	Multiplication, Division
** or ^	Exponentiation
== <>	Condition of equality/inequality
>= <=	Relational condition
> <	Relational Condition
& |	And, Or

If no operator is given, multiplication is performed. The bindings of the operators are according to the mathematical rules. The logical operators have the same weight. Thus it is strongly recommended to use parentheses for combined expressions.

- Functions within an expression

Functions	Description
SIN(x), COS(x), TAN(x)	Trigonometric functions
ATN(x), ATN(y,x)	Arc of tangent x or y/x
ARC(x)	Convert angles to arcus
SQR(x)	Square root
ABS(x)	Absolute value
EXP(x)	Exponents of e
LOG(x)	Natural logarithm
LGT(x)	Base 10 logarithm
DIV(x,y), xDIVy, DIV(x/y)	Integer part of x/y
MOD(x,y), xMODy, MOD(x/y)	Division remainder of x/y
MIN(x,y,..),MAX(x,y,..)	Minimum or Maximum numbers
RANDOM(x)	random value between 0 and 1 (x=0 reinitialises the seed)
IIF(expr,val1,val2)	returns val1 if expr is not equal zero and val2 if expr equals to zero.

A more detailed description of arithmetic expression can be found in the Basics Manual (sofistik_1.pdf) chapter 8.2.15.

FUN – DEFINITION OF FUNCTIONS

If an arithmetic expression is needed more often or the meaning of a variable should be changed during the run, it is possible to define a function.

```
LET#F '=FUN(var,formulaexpression)'
```

Example:

```
LET#F '=FUN(x,3*#x**3-2*#x**2+5*#x)'

LET#1 #F(1.234)
```

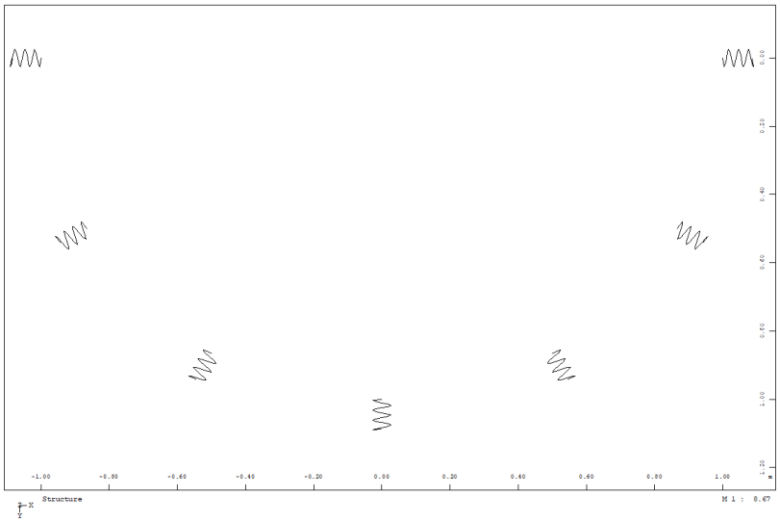
The variable x is substituted by the value of 1.234 and the result of the function saved in variable #1.

The Literal has to start with the character sequence '=FUN(' , followed by a formal parameter name, an expression containing any variables defined so far. Put)' to finish the input at the end.

LOOPS AND JUMPS

The record LOOP initiates loops and terminates by the record ENDLOOP. A loop is executed as many times as determined by a number after LOOP (Default value 9999). The loop will also terminate if the expression following ENDLOOP becomes zero or negative.

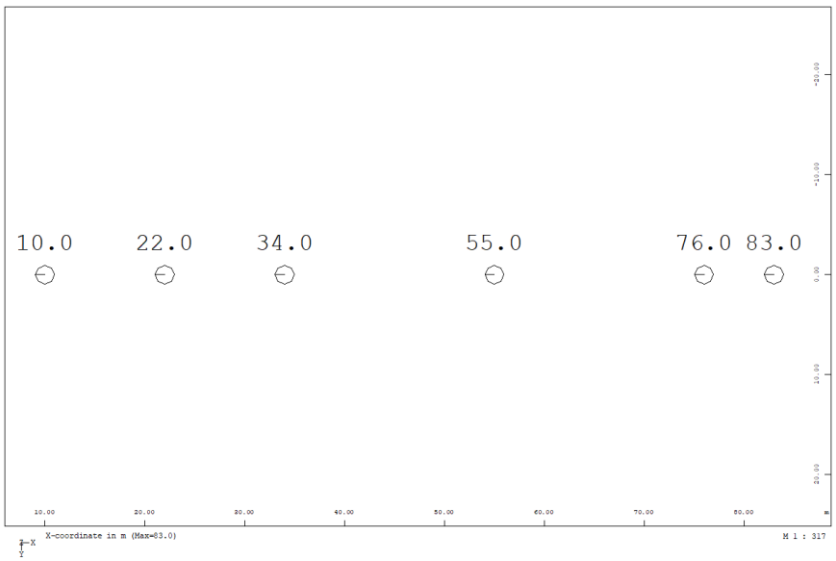
EXAMPLE 1: GENERATION OF NODES AND SPRINGS ALONG A SEMICIRCLE AT EACH 30°



Input option 1	Input option 2
+PROG SOFIMSHA HEAD SYST 3D LET#1 1 LET#2 0 LOOP 7 NODE #1 COS(#2) SIN(#2) SPRI #1 #1 DX COS(#2) DY SIN(#2) CP 1.E5 LET#1 #1+1 LET#2 #2+30 ENDLOOP END	+PROG SOFIMSHA HEAD SYST 3D LET#1 1 LET#2 0 LOOP NODE #1 COS(#2) SIN(#2) SPRI #1 #1 DX COS(#2) DY SIN(#2) CP 1.E5 LET#1 #1+1 LET#2 #2+30 ENDLOOP #2<=180 END

EXAMPLE 2: RUN LOOP OVER ALL ENTRIES OF AN ARRAY

Generate the equal amount of spring elements as values entered in the array A[10,22,34,55,76,83]. Use each entry for the x-coordinate. The loop should stop when getting to the end of the array.



Input:

```
+PROG SOFIMSHA
HEAD
SYST 3D

LET#A1 10,22,34,55,76,83

LOOP#1 A          ! Only the name, #A would be the value 10!
        NODE NO #1+1  X #A(#1) Y 0 Z 0
        SPRI NO #1+1 NA #1+1 DZ 1 CP 1.E5
        ENDLOOP
END
```

A more detailed description of Loops and Jumps can be found in the Basics Manual (sofistik_1.pdf) chapter 8.2.17.

IF – LOGICAL CONDITIONS

Conditional blocks are an vital element of every programming language. The execution of jumps (go to) is not possible within CADINP.

The conditional block is executed if the expression following the IF is greater than zero. You might use the logical expressions for this. Texts are compared with == and != operators only, the comparison is case sensitive. As the complete string on both sides of the operator is considered, no parentheses are allowed, they are part of the strings.

Principle 1	Principle 2

! #1 is the control variable with possible values +1/-1
IF #1
! These lines are input when #1 > 0
! ...
ELSE
! These lines are input when #1 < 0 or = 0
! ...
ENDIF
	IF #1==12
	! These lines are input when #1 equals 12
	! ...
	ELSE
	! These lines are input when #1 does not equal 12
	! ...
	ENDIF

The generation of a CASE construction is available by using a series of additional ELSEIF statements:

```
IF (Condition_1)
....
ELSEIF (Condition_2)
```

....

ELSEIF (Condition_3)

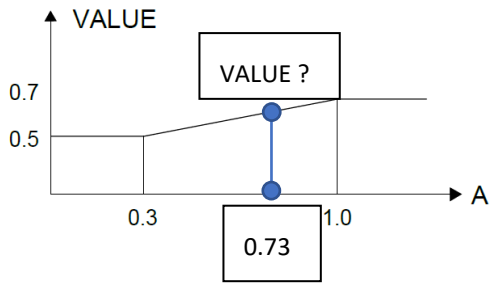
....

ELSE

....

ENDIF

EXAMPLE 3: INTERPOLATION



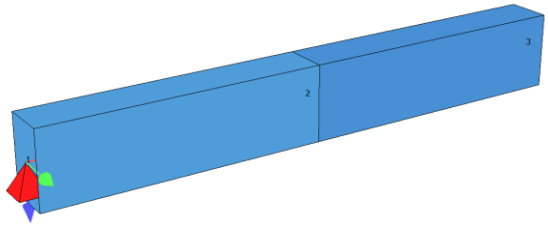
Input	Output

+PROG TEMPLATE
HEAD
 	
LET#A 0.73
 	
IF (#A < 0.3) ! condition 1
LET#VALUE 0.50
ELSEIF (#A>1.0) ! or condition 2
LET#VALUE 0.70
ELSE
LET#VALUE 0.50+0.20*(#A-0.3) ! else this value
ENDIF
	Output value of variable #VALUE = 0.59

#DEFINE – PARAMETER SUBSTITUTION

The declaration of text blocks is made via the input #DEFINE name=text. The name of a text block consists of up to 10 characters without $ prefix. The first character of the name of a text block must be a letter. A parameter in the input data or a block can be referred to by using the syntax $(name). Blanks are not allowed between $ and "(".

EXAMPLE 4: SINGLE-SPAN BEAM USING #DEFINE PARAMETER FOR SYSTEM AND CROSS-SECTION DIMENSIONS



Input	Output
#DEFINE LENGTH=3.70 #DEFINE WIDTH=300 #DEFINE HEIGHT =500	Definition of the global parameter for system length, cross-section width and height-
+PROG AQUA HEAD NORM DC 'EN' NDC '1992-2004' CONC NO 1 TYPE C FCN 25 STEE NO 2 TYPE B CLAS 500 SREC NO 1 H $(HEIGHT) B $(WIDTH) END	Design code definition. Material. SREC: Rectangle cross section number 1 300/500mm
+PROG SOFIMSHA SYST 3D NODE NO 1 X 0 Y 0 Z 0 FIX PP NODE NO 2 X $(LENGTH)/2 Y 0 Z 0 NODE NO 3 X $(LENGTH) Y 0 Z 0 BEAM NO 1 NA 1 NE 2 NCS 1 BEAM NO 2 NA 2 NE 3 NCS 1 END	Definition of the System by creating nodes using the $(LENGTH) parameter. Beam elements with cross-section number 1 between node 1-2 and 2-3.

#INCLUDE – BLOCK DEFINITION

The #DDEFINE command allows, also, to defining single parameters only creating entire text/input blocks. It allows defining recurring text blocks only once and to call it on any location in the input files.

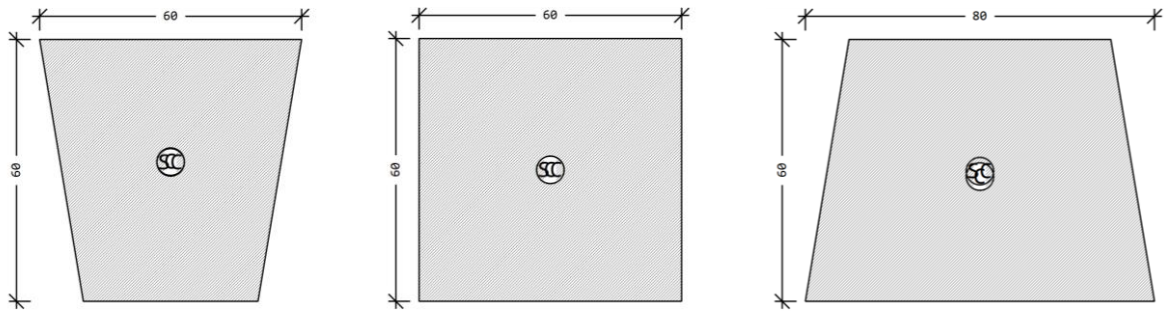
Input	Description
#DEFINE name	Start of the block name
#ENDDEF	End of block
#ENDDEF name	End of Block
#UNDEF name	Delete a defined block
#INCLUDE name	Insert block name content from memory or file

EXAMPLE 5: TEXT BLOCK DEFINITION FOR SIMPLE TEXT OUTPUT

Input	Description and Output
#DEFINE TEXTOUTPUT TXE HELLO WORLD #ENDDEF TEXTOUTPUT	Definition of the Text block
 +PROG TEMPLATE HEAD #INCLUDE TEST END	Considered input to execute +PROG TEMPLATE HEAD TXE HELLO WORLD END

+ EXAMPLE 6: TEXTBLOCK DEFINITION TO USE IT MULTIPLE TIMES FOR CROSS-SECTION DEFINITION.

Defining a trapezoid cross-section by changing the length of the bottom edge only (40 / 60 / 80 mm)



Input	Description and Output
#DEFINE SECT $PROG AQUA POLY TYPE OPZ MNO 1 VERT NO 1 Y #1/2 Z -#3/2 NO 2 Y #2/2 Z #3/2 #ENDDEF SECT	Definition of the text block based on AQUA input commands.
 +PROG AQUA HEAD NORM DC 'EN' NDC '1992-2004' STEE NO 1 TYPE S CLAS 355 SECT NO 1 ; LET#1 60,40,60 #include SECT SECT NO 2 ; LET#1 60,60,60 #include SECT SECT NO 3 ; LET#1 60,80,60 #include SECT END	Considered input to execute +PROG AQUA HEAD NORM DC 'EN' NDC '1992-2004' STEE NO 1 TYPE S CLAS 355 SECT NO 1 ; LET#1 60,40,60 POLY TYPE OPZ MNO 1 VERT NO 1 Y 60/2 Z -60/2 NO 2 Y 40/2 Z 60/2 SECT NO 2 ; LET#1 60,60,60 POLY TYPE OPZ MNO 1 VERT NO 1 Y 60/2 Z -60/2 NO 2 Y 60/2 Z 60/2 SECT NO 3 ; LET#1 60,80,60 POLY TYPE OPZ MNO 1 VERT NO 1 Y 60/2 Z -60/2 NO 2 Y 80/2 Z 60/2 END

A more detailed description about #DEFINE and #INCLDUE can be found in the Basics Manual (sofistik_1.pdf) chapter 10.2 and 10.3.

APPLY – INTEGRATION OF DATA DURING THE CALCULATION

With the APPLY command, a file can be included between two module input blocks, but not inside a module. The referred file should contain one or more module +PROG.

While the #INCLUDE command inserts data before the calculation is executed, the APPLY command executes the referred file during the computation. Data can be generated during a calculation and after that inserted at the right position in the timeline of the process.

As the program modules, the APPLY is controlled with the signs +/-. +APPLY execute the command, -APPLY skip the command.

A more detailed description of APPLY can be found in the Basics Manual (sofistik_1.pdf) chapter 10.4.

SYSTEM COMMANDS

Any shell command can be integrated into the analysis. Parameter substitution is possible within the system commands input lines. (The name of the project replaces character string $(PROJECT))

*SYS Will be executed as in any batch file if the last PROG line was active.

+SYS Will be executed as in any batch file.

-SYS Will not be executed, all following *SYS lines are deactivated.

Use basic commands of the operating system (Windows)

- SYS COPY
- SYS DEL
- SYS REN

Call a module/program by command lines

- SYS SPS

Input	Description
+SYS del "$(PROJEKT).$D1”	Deletes stiffness matrix
+SYS wait "name.exe"	WPS is initiated to wait with the processing of further modules until the active process finished “+SYS -wait for name.exe”.

A more detailed description of system commands can be found in the Basics Manual (sofistik_1.pdf) chapter 10.8.

MANAGING UNITS

The SOFiSTiK programs allow input and output in the most suitable engineering units. The units used in the database are expressed in SI-units kN, m, sec. However, other units can also be used for input and output (e.g. N, mm or cm2) as well as a different number of decimals.

Three categories of units are distinguished:

- mm – Fixed Unit
  Input is always required in the specified unit.

- [mm] – Explicit Unit.
  Input to specified the default unit. Alternatively, an explicit assignment of a related unit is possible such as 2.5[m].

- [mm]$_{1011}$ – Implicit Unit.
  Implicit units are categorised semantically and denoted by a corresponding identity number (shown in green). Valid categories referring to the unit ”length” are such as geodetic elevation, section length and thickness.
  The default unit for each category is defined by the active unit set (design code specific). An explicit input can overrule the input default. The default unit is a unit set 5 (Eurocode, NORM UNIT 5)

DOCUMENTATION AND HEADINGS OF INPUT AND OUTPUT

CHAPTER AND LABEL

A TEDDY input file can be structured by chapter entries and by label entries. The chapters and labels are shown in the Teddy module tree, and both are used as jump labels.

- !#!Chapter
  The chapter allows defining a Super-Block in Teddy. The entire block can be activated or deactivated by double-click on the chapter icon Chapter in the module tree. The symbol changes to either !+! or !-!. !#! has the same meaning as !+!. In the result file, PLB chapters can be opened and closed with the Report Browser.

- !*!Label
  Labels are for jump marks only and need to be placed within the Block definition (module). Result Browser doesn’t consider labels for navigation.

ADD COMMENTS TO THE INPUT

The comment characters $, ! or // turns the right-handed content of the input line into a comment. All comment operators should be separated from the comment itself with a blank as well as a spare line to the previous input line.

```
+PROG AQUA $ Text to print as a header
HEAD
NORM DC 'EN' NDC '1992-2004'
CONC NO 1 TYPE C FCN 25
STEE NO 2 TYPE B CLAS 500

$ Rectangle cross section defined below
SREC NO 1 H $(HEIGHT) B $(WIDTH) $ Cross Section number 1
END
```

TEXT OUTPUT IN REPORT BROWSER

Three commands are available to add information to the output file and represent it in Result Browser.

TXB

Insert text on the very top of the module report.

- Call of a single variable: TXB USER_TEXT #VAR
- Call values of a vector: TXB USER_TEXT #(#VAR(2))
- Calculation of a variable: TXB USER_TEXT #(#VAR*100)
- Spacing and decimal precision of the value: TXB USER_TEXT #(#VAR,3.2)

TXE

Insert a text on the very end of the module report.

The same inputs as for TXB can be used for TXE.

<TEXT> MULTILINE_TEXT </TEXT>

Insert a multiline text on the defined position of the module report.

Refer to an external file: <TEXT,FILE=filename.txt>

A more detailed description about TXA, TXE and <TEXT> can be found in the Basics Manual (sofistik_1.pdf) chapter 8.3.4 to 8.3.6.

TEDDY AND THE GRAPHICAL INPUT ENVIRONMENT OF SOFISTIK

Besides the standalone application, TEDDY is accessible from almost all graphical applications such as SOFiSTiK Structural Desktop, SOFiPLUS (AutoCAD), Revit, McNeel Rhinoceros, Result Viewer and WinGRAF. SOFiSTiK Structural Desktop comes with the most powerful interaction with Teddy.

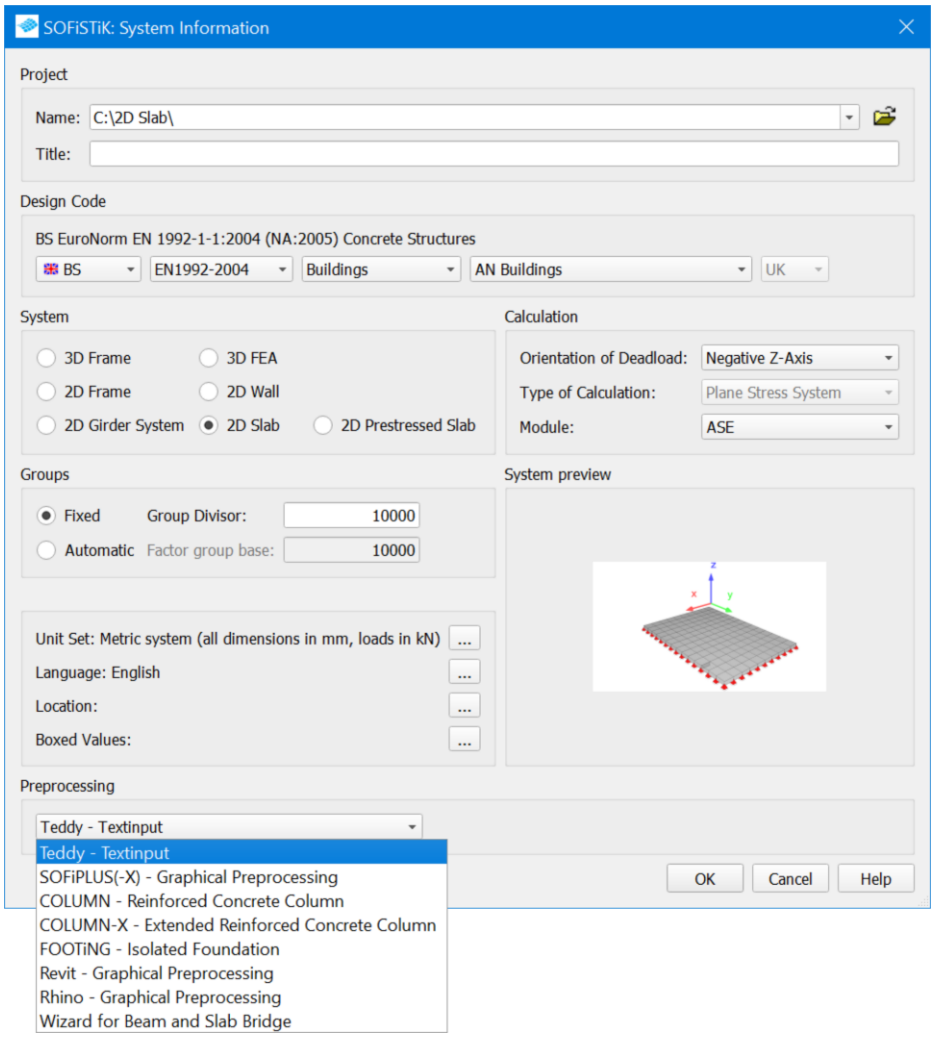
SOFISTIK STRUCTURAL DESKTOP

SOFiSTiK Structural Desktop allows to easily access and organise all available modules in graphical user tasks as well as teddy input. It controls pre-processing, processing and post-processing.

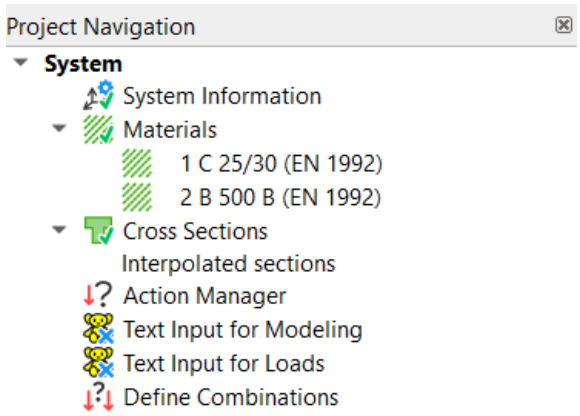
Teddy can be accessed at different places in SOFiSTiK Structural Desktop (SSD). The same capabilities as in the stand-alone application, are available.

SOFISTIK: SYSTEM INFORMATION DIALOG BOX – NEW PROJECT

When starting a new project in SSD it requires to decide what application for the pre-processing of the structure should be used. Several options are available from the drop-down menu of the dialogue box. Besides Revit, SOFiPLUS and some more, **Teddy - Textinput** is the one to define the structure by using Teddy.



After confirming the SOFiSTiK System Information dialogue box by left mouse click, two yellow Teddy tasks show up in the left-handed project tree.

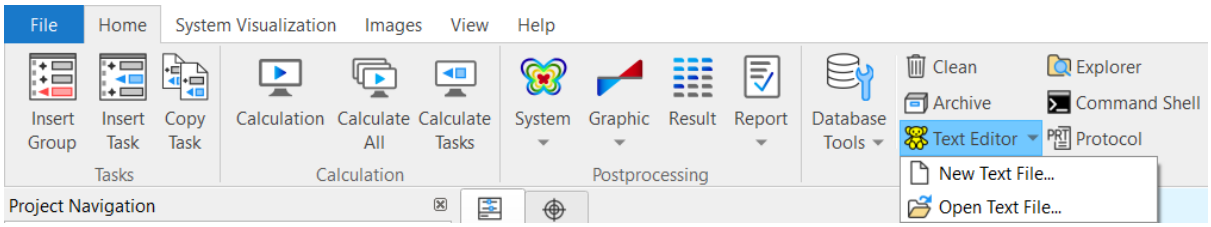


- Text Interface for Model Creation.
  It's a placeholder for the text input of the module SOFiMSHA for system definition.
- Text Interface for Loads.
  It's a placeholder for the text input of the module SOFiLOAD for load definitions.

OPEN A DAT-FILE

SSD supports opening DAT-files as teddy file/project tab within SSD. Leaving SSD isn't necessarily needed. Mainly if external files are referred to the project, it allows you to review and edit the files quite easily.

You can find the command in the "Home" ribbon within section "Tools".



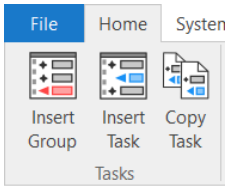
ADD A TEXT EDITOR TASK TO THE PROJECT

Two different types of text editor tasks are available in the SOFiSTiK Structural Desktop. Both can be allocated to the project tree with the "Insert Task" command.

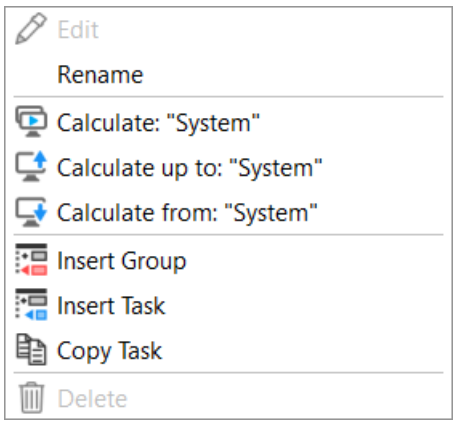
INSERT A NEW TASK

There are three options to allocate a new task to the project tree in the SOFiSTiK Structural Desktop.

- Ribbon "Home" – "Insert Task" Command



- By right-click within the “Project Navigation”

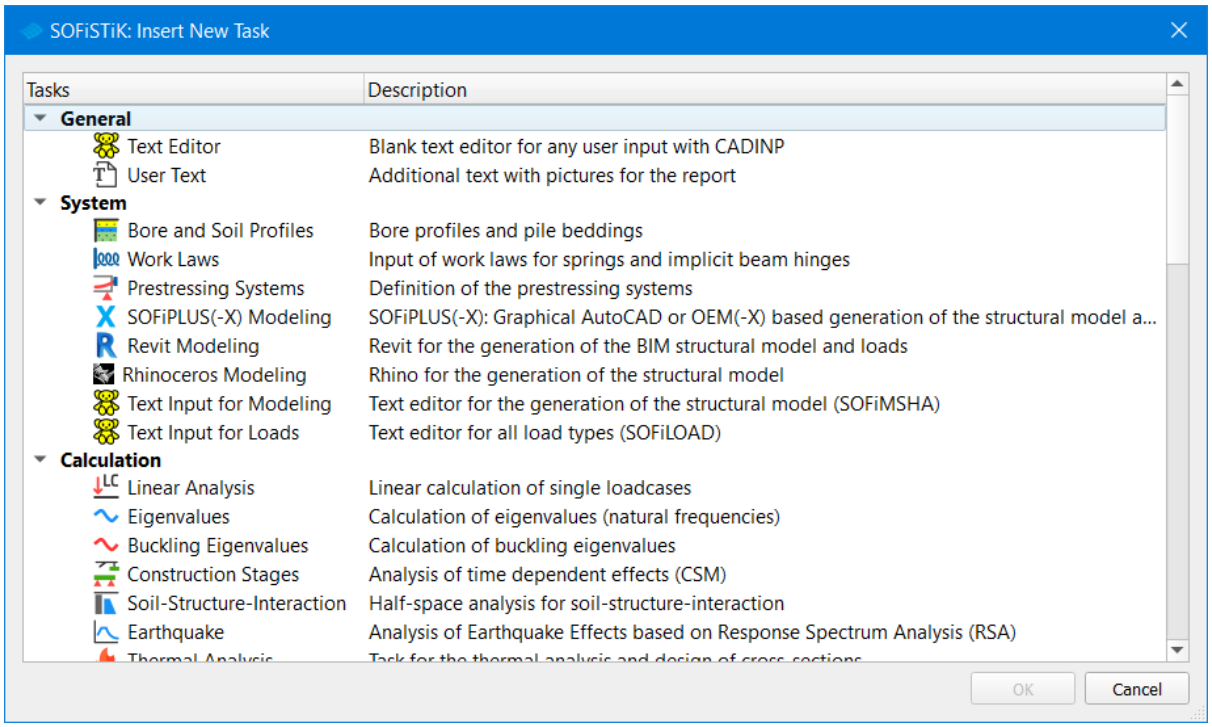


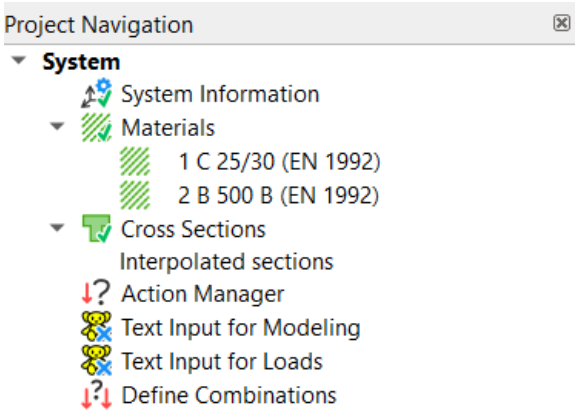
The above options open the “SOFiSTiK: Insert New Task” dialogue box, which represents the library of available tasks.

THE TWO TYPES OF AVAILABLE TEXT EDITOR TASKS IN SSD

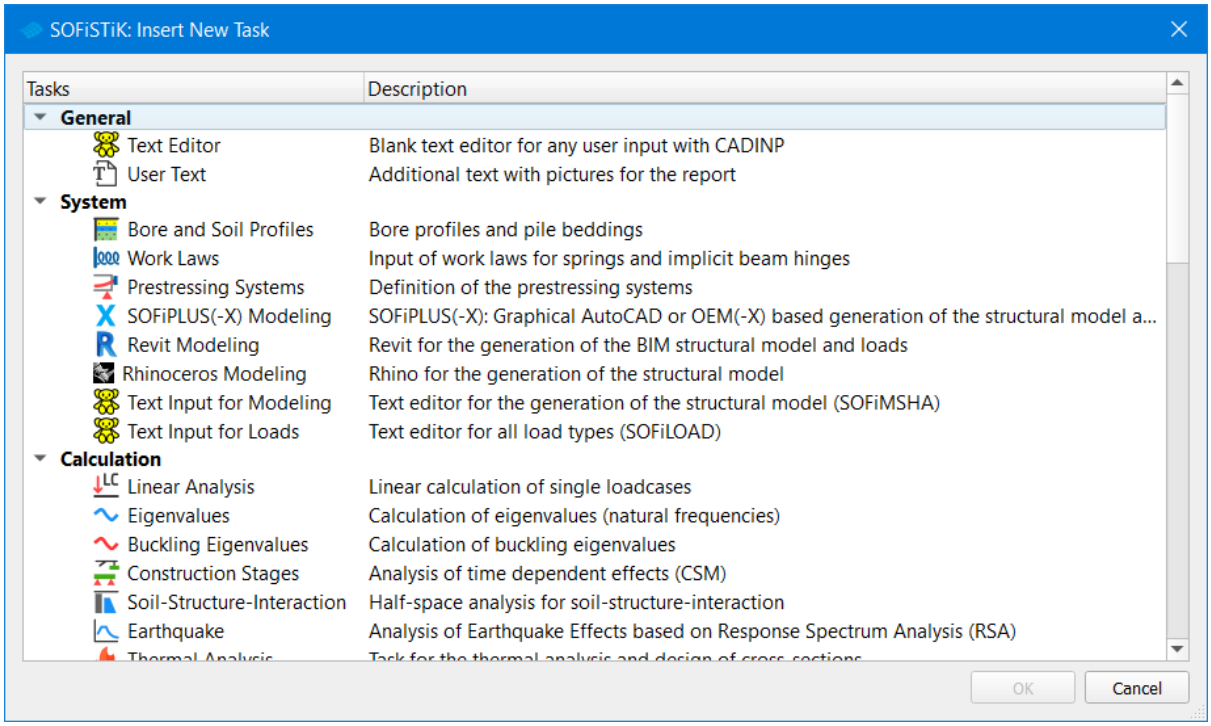
Two types of text editor tasks are available to insert in the project tree.

- The yellow text editor tasks Text Interface for Modelling and Text Interface for Modelling in task group System are linked to system generation and load definition and can be placed in the group “System” within the “Project Navigation” only.
-





- The **Text Editor** task -you can find it in the task group “General”– can be added at a random location in the project tree.

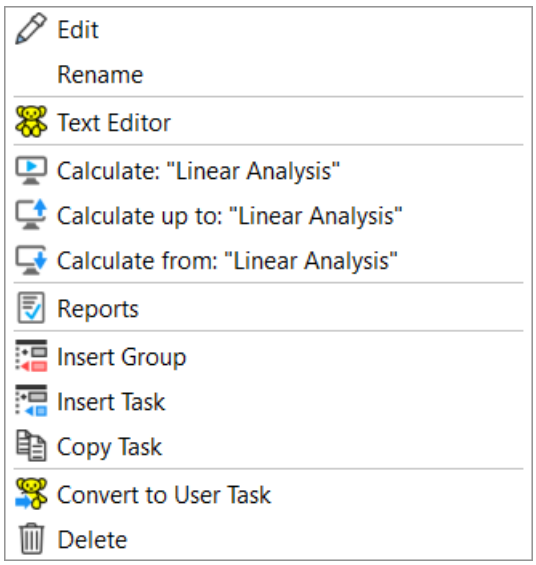


EDIT AND CONVERT GRAPHICAL TASKS

Before running the calculations with the “Calculation” command in the ribbon “Home” in the tab “Calculation” - all input is going to be transferred to text format in the background.

Due to this fact, each graphical user task can be converted to a Text Editor task except the default tasks within the group System at the very top of the “Project Navigation” tree.

By right-click on the task to convert the context menu pops up.



The last but one command

Convert to User Task

converts the graphical task in a text input task.

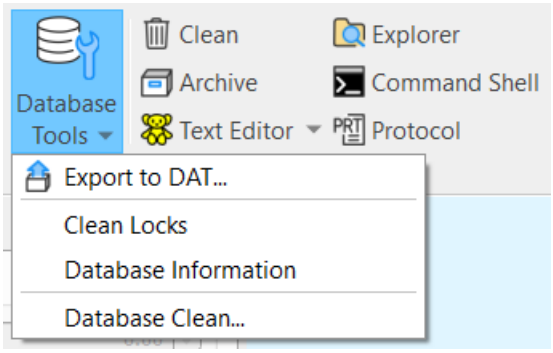
If a graphical task was converted to a text task by mistake, use the undo command. It restores the graphical task and keeps the text input task.

The **Text Editor** command is for viewing purpose only. Any modification of the text input gets lost after the task was accessed with the graphical user interface again.

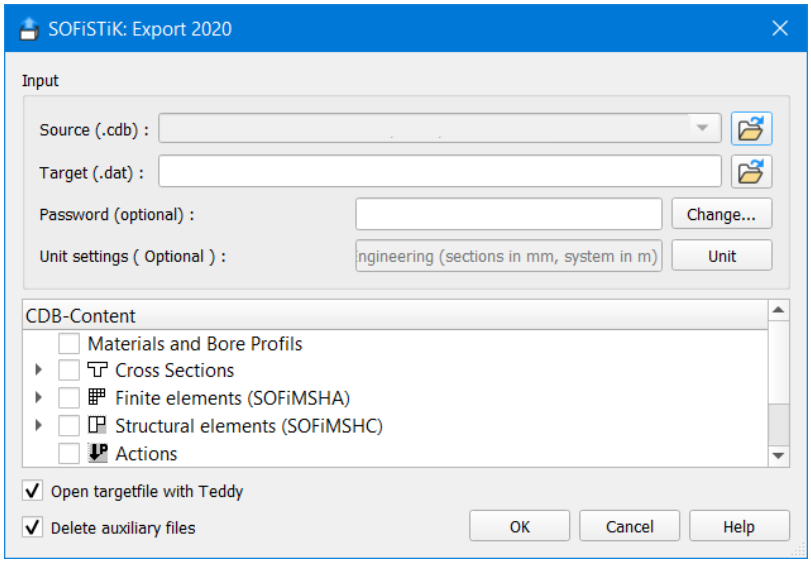
EXPORT TO DAT

Besides converting single graphical tasks to text input SOFiSTiK provides a command to export data saved in the CDB (Central Data Base) to a DAT file. The command is in Teddy available too.

Therefore, use the “Database Tools” command within the “Home” ribbon and tab “Tools”



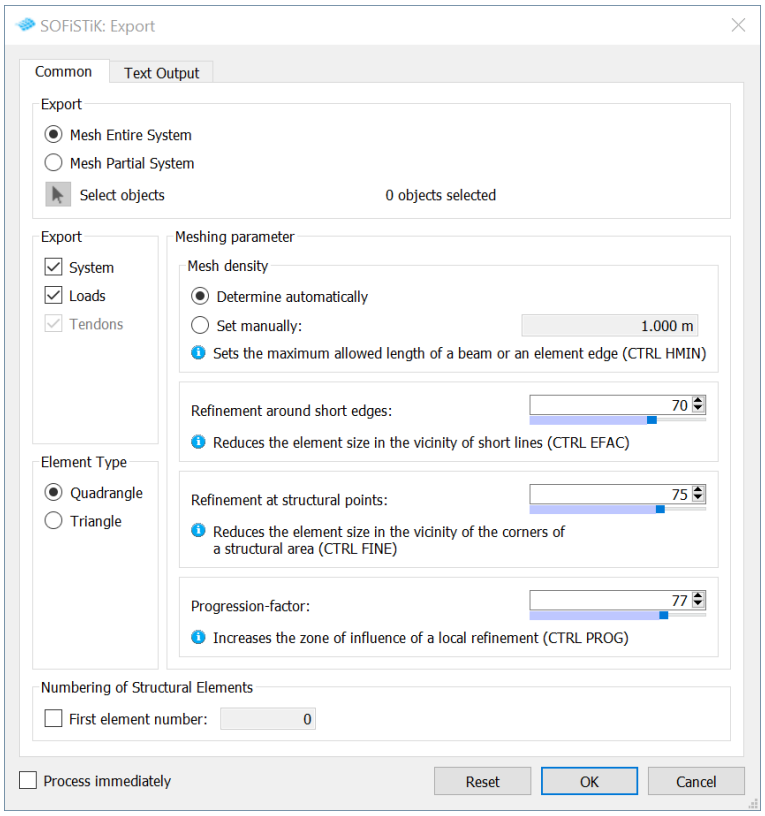
By left-click “SOFiSTiK: Export” opens, and allows to select the information by ticking the checkboxes to generate the target file with the requested information.


SOFiSTiK: Export 2020
Input
Source (.cdb) :
Target (.dat) :
Password (optional) :
Change...
Unit settings (Optional) :
ngineering (sections in mm, system in m)
Unit
CDB-Content
Materials and Bore Profils
Cross Sections
Finite elements (SOFiMSHA)
Structural elements (SOFiMSHC)
Actions
Open targetfile with Teddy
Delete auxiliary files
OK
Cancel
Help

SOFIPLUS (AUTOCAD)

SOFiPLUS represents one of the available pre-processors for graphical modelling of analytical systems. Interfering with the application by text input isn't necessary. However, extracting specific information set graphically in SOFiPLUS to a text input can be of use. Especially, to learn about the input syntax to set up tendons using the TENDON module.

When Exporting the system ensure the checkbox ☐ Process immediately in the bottom left corner is not activated. Confirm the export with OK. The Export dialogue box closes immediately, and WPS doesn't start to execute the calculation.



Go to the project directory and check it for DAT files. The default files are

- name_aqa.dat
- name_ldf.dat
- name_msh.dat
- name_run.dat

If tendons are used in the project, additional files depending on the tendon type are available too.

- name_spt.dat – Tendons in shells
- name_tnd.dat – Tendons in slabs
- name_bpt.dat – Tendons in beams

PROGRAM MODULES

AQB – DESIGN OF CROSS SECTIONS

AQB is used for stress analysis and design of cross-sections created with AQUA. The enhanced version AQBS allows the consideration of the unique features of prestressed concrete and composite structures (prestressing tendons, imposed secondary stresses, creep and shrinkage).

AQUA - MATERIAL AND CROSS SECTIONS

AQUA calculates the properties of cross-sections of any shape and material. The cross-section properties for static analysis are determined, as well as characteristic magnitudes for the calculation of normal and shear stress.

ASE – GENERAL STATIC ANALYSIS OF FINITE ELEMENT STRUCTURES

ASE calculates the static and dynamic effects of general loading on any structure.

The basic version of ASE performs the linear analyses of beams, cables, trusses, plane and volume structures. Plain strain and rotational systems can be analysed with TALPA.

The extended versions of ASE are capable of running:

- Influence surfaces
- Non-linear analyses
- Pile elements with linear/parabolic soil coefficient distribution
- Creep and shrinkage
- Forces from construction stages
- Modal analysis, Time step method
- Material non-linearities
- Geometrical non-linearities
- Membrane elements
- Evaluation of collapse load
- Non-linear dynamics

BDK – BUCKLING RESISTANCE OF STEEL MEMBERS

BDK determines the stability eigenvalues for buckling of a straight individual member. The member, which is taken from the entire system, is calculated with STAR2 or ASE and the buckling resistance check of steel members is performed according to the selected design code.

BEMESS – DESIGN OF PLATES AND SHELLS

The program BEMESS is used to design forces of the analysis programs ASE or superimposed forces of MAXIMA according to various codes. BEMESS is as well capable of extracting extreme stresses from a set of load cases according to the linear elastic theory. In addition to the ultimate limit design, serviceability checks can be done such as crack width control, minimum reinforcement, stress and fatigue checks.

COLUMN – REINFORCED CONCRETE COLUMN

The SOFiSTiK Structural Desktop task “Column Design” provides the nominal curvature according to the standard EN 1992-1-1:2004 Chapter 5.8.8. It can be applied to a single column or a whole structure with a constant normal force distribution.

The nominal curvature is a simplified method, which calculates the approximate bending moment with second-order effects. Though, the approximation is based on the linearization of the dimensioning diagram. The necessity of 2nd order effects is checked in consideration of the slenderness criteria according to the standard EC 2 Chapter. 5.8.3.1.

CSM – CONSTRUCTION STAGE MANAGER

The construction stage manager allows to set up and organise different stages of the project as well performs creep analysis.

DBMERG – DATABASE MANIPULATION

Program DBMERG allows to modify and transfer data within a database or between databases of the SOFiSTiK software.

DECREATOR – GENERATION OF DESIGN ELEMENTS

The program DECREATOR generates design elements along with structural members. Design elements are independent of a member’s finite element discretisation and support defining design sections at any position along a structural line.

DOLFYN – FLUID DYNAMICS POWERED BY CYCLONE FLUID DYNAMICS BV

DOLFYN has been developed by Henk Krüs from Cyclone Fluid Dynamics BV and is available as open-source software (www.dolfyn.net). It performs analysis of

- Fluid mechanic (Navier-Stokes equations)
- Turbulence models for high Reynold numbers
- Free Surfaces for Fluids
- Heat Conduction and Convection

DOLFYN was adopted to the SOFiSTiK environment to extract the mesh, material and boundary conditions out of the SOFiSTiK database as well to save results for postprocessing.

DYNA – DYNAMIC ANALYSIS

The program DYNA performs dynamic analysis of 3-dimensional structures such as

- Static analysis of load cases acting upon three-dimensional structures, as well as planar and axisymmetric structures.
- Static analysis of load cases after second-order theory
- Computation of the natural frequencies of three-dimensional structures.
- Computation of the buckling eigenvalues of three-dimensional structures.
- Implicit direct integration of the equations of motion for structures with arbitrary damping
- Explicit direct integration of the nonlinear equations of motion
- Interaction with load trains and wind loading
- Soil structure interaction with the SBFEM
- Integration of the equations of motion by superposition of the mode shapes.
- Steady-state oscillations and excitation through spectra.
-

DYNR – TRANSIENT RESULTS AND RESPONSE SPECTRA

The program DYNR enables the graphical presentation of transient results from the programs DYNA and HYDRA. It plots time-controlled, and frequency-controlled results as well as calculate response spectra of a given history of loading and acceleration.

ELLA – EXTENDED LIVE LOAD ANALYSIS

ELLA performs the analysis and evaluation of imposed loads acting on beam or shell structures. It is processed by moving load trains along traffic lanes. The envelopes of the scalar variables for the entire structure are the result. The evaluation of influence lines or influence surfaces is necessary to achieve an accurate analysis of larger systems.

FEX-DYN – NONLINEAR EXPLICIT DYNAMICS

FEX-DYN is a FEM analysis kernel for explicit integration of motion equations inside the SOFiSTiK product line. FEX-DYN is efficiently programmed to cope with the physically and geometrically non-linear problems and conceptualised for multiprocessor machines.

FOOTING – FOUNDATIONS

The SSD task FOOTiNG allows based on given loads the calculation of required foundation dimensions – As well to run design checks on existing foundation dimensions. Footing allows assigning vertical forces, moments and horizontal forces. Necessary design combinations are automatically determined. And it is possible to consider 2nd order effects.

HASE – HALF SPACE ANALYSIS FOR STATIC SOIL-STRUCTURE INTERACTION

Based on the half-space theory, module HASE calculates the stiffness matrix representing the soil of the static soil-structure interaction. The soil-structure interaction analysis itself is performed in module ASE.

HYDRA – SEEPAGE AND THERMAL ANALYSIS

The thermal analysis allows the evaluation of the temperature distribution within a solid due to steady or transient heating or cooling or temperature gradient. Groundwater models are used to compute the movement of water (seepage) within a porous solid needed for geology, hydrology, soil mechanics or groundwater supply.

MAXIMA – SUPERPOSITION

MAXIMA performs the determination of extreme values of the internal forces, stresses, displacements and support reactions.

RELY – STRUCTURAL RELIABILITY POWERED BY STRUREL®

Rely is an add-on to the SOFiSTiK program to perform reliability analysis, where the engineering system of interest is modelled using one of the SOFiSTiK finite element modules. The stand-alone software package Strurel powers the kernel of Rely.

SIR – SECTIONAL RESULTS

The program SIR defines 3-dimensional sections in the structure and performs

- Representation of the intersected elements (TRUS, BEAM, CABL, QUAD, BRIC, SPRI) and the graphical representation of their results
- Calculation of the resultant forces and moments including the support reactions
- Generation of a data record to run the design of the sections within the AQB

SIR generates cross-sections and internal forces and moments which can be used straight by AQUA and AQB or AQBS.

SOFILOAD – LOADS AND LOAD FUNCTIONS

SOFiLOAD represents the load generator for all SOFiSTiK-Modules.

SOFIMSHA – IMPORT AND EXPORT OF FINITE ELEMENTS AND BEAM STRUCTURES

SOFIMSHA exports and imports the total volume of nodes and elements. The capability of element generation is available in CADiNP only. Complex structures should be defined via SOFIMSHC or SOFiPLUS.

SOFIMSHC – GEOMETRIC MODELLING

SOFiMSHC is a tool for creating and processing geometric models and finite element structures. It can be used as a stand-alone program within CADiNP as well as integrated as geometry processing module in the SOFiSTiK programs SOFiPLUS, Extensions for Revit and Rhinoceros Interface.

STAR2 – STATICS OF BEAM STRUCTURES 2ND ORDER THEORY

The programs of the STAR-modules enable the computation of the internal forces in any 3-dimensional beam structure by 2nd or 3rd order theory taking into consideration shear deformations as well as various non-linear material effects. (STAR1, STAR2, STAR3)

TALPA – 2D FINITE ELEMENTS IN GEOTECHNICAL ENGINEERING

TALPA includes two programs

- CSG – The Construction Stage Manager Geotechnics performs the automatic generation of the calculation process.
- FEA – Finite Element Analysis – calculation of analysis of different construction stages.

The program supports numerous non-linear material models (e.g. elastoplastic, viscoplastic). Accounting for non-linear support effects, such as friction or slip, is possible, as well.

TENDON – GEOMETRY OF PRESTRESSING TENDONS

TENDON defines prestressing for beams, slabs and folded structures. In addition to the calculation of the prestressing tendon geometry, the program computes the prestressing forces taking into consideration the prestressing process and the friction losses. The input of the tendon geometry can be done graphically as well as numerically.

TEXTILE – CUTTING PATTERN MEMBRANE STRUCTURES

TEXTILE calculates the development of a 3D-curved surface. It includes the computation of the stressed membrane to obtain the cutting pattern areas.

WINGRAF/WING - GRAPHICAL REPRESENTATION OF FINITE ELEMENTS AND BEAM STRUCTURES

WinGRAF is used for the graphical output for the finite element and framework analysis modules. It is capable of representing almost all information saved in the central database.