

TpaCAD

2.4.30

Program Editor



Tecnologie e Prodotti per l'Automazione

This documentation is property of TPA S.r.l. Any unauthorized duplication is forbidden. The Company reserves the right to modify the content of the document at any time.

Table of Contents

1 Overview 1

1.1 Introduction 1
1.2 Activation and operating mode 2
1.3 Access right to the system 6
1.4 Multilingual support 7
1.5 Format compatibility 8
1.6 New functionalities 8
1.7 System requirements 10
Running TpaCAD on virtual machines 10
1.8 Checking control at TpaCAD launch 10
1.9 Helps 11

2 Versions and updates 12

2.1 Versions from 2.4.0 (May 22, 2020) to 2.4.30 (August 04, 2026) 12
2.2 Versions from 2.3.1 (March 08, 2019) to 2.3.20 (December 02, 2020) 20
2.3 Versions from 2.2.0 (December 04, 2017) to 2.2.15 (March 28, 2019) 22
2.4 Versions from 2.0.0 (April 08, 2016) to 2.1.18 (September 26, 2018) 25
2.5 Versions from 1.4.0 (April 10, 2015) to 1.4.9 (March 30, 2016) 29
2.6 Versions from 1.3.0 (February 10, 2014) to 1.3.11 (March 31, 2015) 32
2.7 Versions from 1.1.0 (December 06, 2012) to 1.2.4 (October 10, 2013) 35

3 Graphic interface 39

3.1 Overview 39
3.2 List of Shortcuts and Mouse Keys 43

4 Working with the programs 46

4.1 Creating a program 46
Create a program according to a template 46
4.2 Opening and importing a program 46
Import a program from external format 47
Import with generation of more files 48
Import a program in TpaEdi32 format 48
Import a program in EdiCad format 48
Opening a program-piece created in an external environment 49
Starting TpaCAD from Resource Management 49

Reports while opening a program 49
Recording format of a program-piece 50
4.3 Open a program from the recent file list 50
4.4 Drag (drag & drop) 51
4.5 Printing a program 51
4.6 Saving a program 51
Names that cannot be used 52
4.7 Optimizing a program 52
4.8 Print the program label 52
4.9 Exporting a program 53
4.10 Converting an archive of programs 53
4.11 Optimizing an archive of programs 54
4.12 Seeing the Program Optimization Preview 55
4.13 Plant 55
4.14 Operating Environment 56
4.15 Multiple instances of TpaCAD 56
4.16 Tool table 56
4.17 Information on external components linked to TpaCAD 58
5 How to configure the graphic representation 59
5.1 Customize views 59
5.2 Customizing View in Tool Compensation 62
5.3 Checking the view 63
5.4 Three-dimensional representation 64
5.5 Special Views and View filters 65
5.6 About profile 65
6 Piece 67
6.1 Graphic display of the Overall View 67
6.2 Piece geometry 67
6.3 Assignments 72
Area of assignments 72
Dimensions, Execution modes and Properties 72
LxHxS 72
Execution mode 73
Properties 73
"o" Variables 74
"v" Variables 75
"r" Variables 75
Edit wizard 76
Recovering "r" variables from an existing program 78
Special settings 79
Additional info 80
Modelling 80
Fictive faces 80

Information on fictive faces in the local status bar 82
Curved fictive faces and modelling 83
Example 1 83
Example 2 85
Example 3 86
Example 4 87
Section of constraints 88
Optimizations 88
Sequences 88
6.4 Advanced assignments 89
Exclusions 89
Layers 90
Special filters 91
7 Face 94
7.1 Graphic display of the Face View 94
7.2 How to open it 96
7.3 ASCII Text Area 97
8 Piece-Face 101
8.1 Overview 101
8.2 How to open it 101
8.3 Working assignment area 101
8.4 ASCII text area 102
8.5 F Field 102
8.6 Representation 102
8.7 Execution sequences 103
9 Workings 104
9.1 Working type 104
Simple and complex workings 104
Application point 105
Technology 106
Oriented geometries 107
Technological priority 111
Graphic representation 111
9.2 Profile 111
Profile workings 111
Profile building 112
Application point 112
Programming the angles 113
Tangent lines and intercept lines 113
Path 114
Assigning the technology 115
Multiple setups 117
Opening and closing a profile 118
Hooking the profiles 120
Simple profiles 121
Tool compensation 122
Variation of the compensation 125
Variation of the side to compensate 127

Display 127
Executing a profile with sharp corner cut 128
Compensation of the correction diameter in the case of a conical tool 130
Assignment of profiles in piece-face 130
9.3 Logical Instructions 131
IF ... ELSEIF ... ELSE ... ENDIF Structures 131
Exit instruction 133
Error instruction 133
Warning instruction 134
J Variables 134
Global functions 136
9.4 Subroutine 137
Subroutine 137
Assigning the subroutine variables 139
Positioning a subroutine 141
Programmed application point 143
Point hook 143
Final application point 144
Applying workings to the correct face 144
Induced (automatic) calls 145
Selecting induced faces 145
Positioning induced calls 146
<j> variable solution in automatic induced calls 146
Direct calls 147
Programmed induced calls 147
<j> variable solution in programmed induced calls 149
Applying geometric transforms 149
Repetitions in subroutine running 150
Repetitions with free distribution 150
Repetitions with matrix distribution 151
Seeing the development of a subroutine 152
Nesting subroutine calls 153
Edit wizard and assisted functionalities 153
9.5 Programmed tools 154
Advanced use of the programmed tools 157
9.6 Automatic Faces 158
9.7 Insertion of Geometric Entities from Drawing Menu 160
9.8 Inserting Bookmarks 166
9.9 Change and Insertion 167
Selecting the insertion point in a program list 167
Selection 168
General Selection Commands 169
Change of the active working 169
General commands of Change in face program 169
Change of Properties 170
General Purpose Change Commands 171
Find 173
Replace 174
Replace variable 176
Solve 178
10 Tools 180
10.1 Introduction 180

10.2 General tools 180
Centring and alignment 180
Translation 181
Rotation 182
Modify (menu of Graphics) 184
Symmetries 184
Explosion 185
Advanced considerations 186
Repetitions 188
Free repetitions 188
Rectangular series 188
Circular series 189
Repetitions on a profile 191
10.3 Profile Tools 191
Change a profile segment 191
Change corner into arc 194
Change the line in the path 194
Apply entry to profile 195
Apply exit to profile 196
Close profile 197
Invert profile 198
Scale profile 198
Pull profile 200
Split profile 201
Take off each profile segment 203
Extend 203
Fillet profile 204
Chamfer profile 206
Minimize profile 208
Fragment profile 209
Linearize Z 211
Profile Unions 212
With translation 212
With connection segment 212
Connect consecutive profiles 213
Move setup to closed profile 214
Apply setup to profile 215
Apply multiple setup 215
10.4 Constructions 215
Compensated profile 215
Apply bridges to profile 216
Apply Z feed 219
Apply profile repetition 221
Duplicate profile 223
Cut profiles 224
Profile Building 225
Divide on intersection points 226
Text generation 227
Generate spline from polyline 234
Emptying of areas 237
Rotating profiles on a Cartesian plane 241
10.5 Nesting construction of profiles 243
Nesting 243
Nesting solution 251
10.6 Advanced tools in face program 253

Create fictive face from geometry 253
Create surface from geometry 254
Create modelling from geometry 255
Create font from geometry 255
10.7 Useful tools 258
Dimensioning 258
Measures 259
10.8 Overall Program Tools 259
Apply technology 260
Convert [mm]-[inch] 260
Validate profiles 260
Apply reduction to profiles 262
Apply fragmentation to profiles 262
Apply connection to profiles 262
10.9 Overall program transforms 263
Rotate the piece 263
Mirror the piece 264
Overturn the piece 264
11 Parametric Programming 266
11.1 Introduction 266
11.2 Variables and Numeric Parameters 266
11.3 Functions 267
11.4 Variables and String Parameters 267
11.5 Numerical Formats of Special Use 269
11.6 Expression Terms 270
Operators 270
Arithmetic 270
Logical 270
Brackets, Separators 271
Variable Arguments 271
General arguments 271
Execution mode 273
Environment Settings 274
Piece Variables 275
References to Piece Variables 276
Assignments relative to the application of subroutine or macro 277
Setting of custom sections 281
Global Variables 282
Auxiliary functions 282
Mathematical functions 282
Trigonometric Functions 285
Outlines of trigonometry 285
Functions 286
Functions which operate on strings 287
Logical Functions 290
Technological Functions 292
Technological parameters assigned with symbolic formalism 292
Access functions to a generic plant group 293
Access functions to a machine level for the Configuration Of Head Groups 293
Tool access functions 294
Function of direct access to matrix plants 298
Multi-Purpose Geometry Library Functions 300

Functions for calculating angles 301
Functions for calculating distances 304
Function of detection points on geometric elements 305
Point rotation functions 310
Mirror functions 310
Angle rotation functions 312
Segment correction functions with offset 313
Functions of coordinate conversion and of faces information reading 314
Algebraic functions 319
Access functions to programmed working information 320
Custom functions 323

12 Error Messages 324

12.1 General Errors 324

1 - Error in procedure 324
2 - Error in memory allocation 324
5 - Error in file access 325
6 - Error accessing the Clipboard 325
7 - Error accessing an Undo temp file 325
13 - System level doesn't allow execution of this operation 325
18 - Current working is invalid 325
36 - Maximum number of workings per face was overcome 326
38 - Can't insert this working on the current face 326
39 - The tool can't use a required working 326
41 - Errors assigning working properties 326
42 - No modifications or replacements have been affected 326
49 - This tool may only be applied to profiles 327
281 - File reading: unexpected file end 327
282 - File reading: section closure not found 327
283 - File reading: invalid face identifier 327
284 - File reading: working identification number not valid 327
286 - File reading: error file decoding 328
287 - File reading: the program is not compatible with environment assignments 328

12.2 Specific errors in applying tools 328

50 - Tool did not interpret transforms 328
51 - This tool may be applied only to a simple profile 328
53 - Minimize the profile: reduction angle exceeds the value of 90.0° 328
54 - Fragment profile: maximum length of traits is null 328
55 - Apply bridges to profile: invalid number of bridges [minimum 2; maximum 255] 329
56 - Apply bridges to profile: invalid length of bridges or compensation exceeding tool diameter 329
59 - Apply bridges to profile: invalid or unassigned thickness of bridges 329
60 - Apply bridges to profile: cannot distribute bridges on profile (reduce the number of bridges) 329
61 - Profile inversion: complex codes encountered that cannot be inverted 329
62 - Apply tool: complex code of profile end does not terminate with a profile trait 329
63 - Displace setup in profile: the position coincides with the current setup 330
64 - The tool can be applied to a closed profile 330
67 - Fillet or chamfer profile: radius assigned is null 330
68 - Cut profile: shown position already coincides with setup 330
69 - Cut profile: shown position already terminates a profile 330
70 - Enter / Exit profile: reference working is unassigned 330
71 - Apply tool: can't link an entry before profile 331

72 - Enter profile: displacement unassigned for initial point of profile 331
73 - Exit profile: displacement unassigned for profile final point 331
75 - Join profiles: second profile not properly identified 331
78 - Join profiles: profiles are separated 331
79 - Stretch profile: non-modifiable complex codes were encountered 331
80 - Stretch profile: amplification or reduction factor unassigned or equal to 1.0 332
82 - The tool required too many repetitions 332
85 - Apply tool: the profile assigns arcs in a plane different from xy 332
86 - Profile exit: can’t hook on a downward exit 332
88 - Apply tool: unable to apply setup, reference code missing 332
92 - The tool didn't introduce displacements on any axis 333
93 - The tool introduced a null rotation 333
94 - The tool didn't introduce repeated applications 333
95 - Develop text: the text was truncated at the maximum size allowed for the development of the curved geometry 333
96 - Develop text: the conic section of the development is not valid 333
98 - Create text: insufficient font height (min = eps * 100) 333
99 - Develop text: invalid arc of development 334
294 - Emptying of areas: profile is not closed 334
295 - Emptying of areas: profile unsuitable for the assigned tool 334
296 - Emptying of areas: tool radius assigned is null [min: 10*epsilon] 334
297 - Emptying of areas: coverage exceed the tool radius 334
298 - Emptying of areas: depth range includes Z=0.0 334
299 - Emptying of areas: invalid Z air coordinate 334
300 - Emptying of areas: excessive profile number to evaluate (max: 300) 335

12.3 Errors In Parametric Programming 335

101 - Parametric programming: string too long 335
102 - Parametric programming: invalid syntax 335
103 - Parametric programming: <r> variable recalled by name not found 336
105 - Parametric programming: value exceeds range allowed (-3.4E+30; 3.4E+30) 336
106 - Parametric programming: solution of string parameter too long (max = 260 chars) 336
109 - Parametric programming: invalid context for subroutine arguments 336
111 - Parametric programming: invalid context for use of variables <$> 336
112 - Parametric programming: invalid context for use of variables <r> 336
113 - Parametric programming: invalid context for use of variables <v> 337
114 - Parametric programming: invalid context for use of variables <o> 337
115 - Parametric programming: invalid context for use of variables <j> 337
116 - Parametric programming: invalid context for use of working name 337
117 - Parametric programming: invalid index to a variable <r> 337
118 - Parametric programming: invalid index to a variable <j> 337
119 - Parametric programming: invalid index to a variable <$> 337
120 - Parametric programming: invalid index to a variable <v> 338
121 - Parametric programming: invalid index to a variable <o> 338
122 - Parametric programming: function with too many operands (max 30) 338
123 - Parametric programming: function without operands 338
124 - Parametric programming: function with wrong operands number 338
125 - Parametric programming: division by zero 338
126 - Parametric programming: value of trigonometric function (sin, cos) beyond range -1 +1 338
127 - Parametric programming: square root of negative value 338
128 - Parametric programming: exponentation with invalid exponent [min: 0; max: 10] 339
129 - Parametric programming: invalid geometrical library function 339
130 - Parametric programming: function missing required argument 339
132 - Parametric programming: invalid angle for tangent calculation 339

134 - Parametric programming: too many nested calls of custom function (max: 5) 339
135 - Parametric programming: invalid use of custom function 339
136 - Parametric programming: invalid use of arguments arg# res# 339
137 - Parametric programming: arg# argument invalid index or name 339
138 - Parametric programming: res# argument invalid index or name 340
139 - Parametric programming: error calling custom function 340
140 - Parametric programming: error while using functions reserved for custom functions 340
141 - Parametric programming: invalid index to var# argument 340

12.4 Errors in processing variable geometries 340

22 - It's impossible the deletion of a face which has workings assigned 340
144 - Variable geometries: invalid or not assigned reference face 340
145 - Variable geometries: not all the face vertices are distinguished 340
146 - Variable geometries: face vertices are aligned 340
147 - Variable geometries: invalid face polar geometry 341
148 - Variable geometries: invalid rotation plane 341
149 - Variable geometries: impossible to assign the face third point 342
150 - Variable geometries: invalid depth point 343
165 - Variable geometries: invalid face curvature radius 344
166 - Variable geometries: geometric solution error of the surface 344
167 - Variable geometries: max. number of elements in the surface 345

12.5 Errors compiling face program 345

151 - The <operative code name> working code is invalid 345
152 - <parameter name> parameter: invalid value 345
153 - <parameter name> parameter: set format $nn 345
155 - <field name> property: invalid value 346
156 - <field name> field: the value doesn't comply with the minimum set 346
157 - <field name> field: the value exceeds the maximum set 346
158 - Modelling: invalid code or sequence code 346
161 - Too many or not available automatic faces 346
162 - Field F: invalid value 347
190 - Working exceeding the application limits (axis <axis name>) 347

12.6 Errors in works on profile 347

192 - Tool radius computed as infinite 347
193 - Tool radius null 347
194 - Invalid arc 347
195 - Invalid intersection line 347
196 - Invalid entry tangent 348
197 - Invalid exit tangent 348
198 - Point computed external to traits 348
199 - Intersection non-existent 348
200 - Invalid arc (points are not distinguished) 348
201 - Invalid arc (points aligned) 348
202 - Oval: invalid radius 348
203 - Oval reduced to a circle 348
204 - Oval: null or invalid axis / axes 348
205 - Ellipse/Oval: start point exterior conic extents 349
206 - Rectangle: invalid axis/axes or radius 349
207 - Polygon: invalide number of sides 349

12.7 Errors in subroutine or macro 349

208 - Programmed tool: no correspondence found 349
209 - Application Invalid encrypted program 349
210 - Invalid subroutine name 349
211 - Subroutine doesn't exist 349
212 - Shown file hasn't a valid format for subroutine 349

213 - Face number not valid 350
214 - Element of technological reference not applied 350
216 - Subroutine read failure 350
217 - Subroutine name unassigned 350
218 - Curve creation can't be applied 350
219 - Emptying cannot be applied 350
220 - Rotation cannot be applied 350
221 - Inversion cannot be applied 350
222 - Mirror x cannot be applied 351
223 - Mirror y cannot be applied 351
224 - Stretch cannot be applied 351
225 - Programmed tool: one or more workings had been excluded 351
226 - Too many nested subroutine calls (max 5) 351
227 - Custom error number <custom error code> 351
228 - Impossible to assign font (invalid name) 351
229 - Impossible to assign device for font creation 352

12.8 Errors in logical conditions 352

230 - Number of unloaded ELSE or ENDIF exceeds the loaded IF 352
231 - Number of unloaded ENDIF lower than loaded IF 352
232 - Invalid code after an open IF 352
233 - Number of unloaded ENDFOR greater than loaded FOR 353
234 - Number of unloaded ENDFOR lower than loaded FOR 353
235 - Number of FOR instructions exceeds the maximum admissible (max = 500) 353
236 - Number of now running iterations of FOR cycles exceeds the maximum value (max = 100000) 353
237 - An ENDIF instruction is used to close a FOR cycle 353
238 - ENDFOR instruction used to close an IF cycle 354

12.9 Errors in global function assignment 354

239 - An ELSEIF instruction is used in an IF cycle after an ELSE 354
240 - Custom function name is unassigned 354
241 - Custom function name is invalid 354
242 - Error during the carrying out of custom function: returns are unassigned 354

12.10 Errors in multiple setups (profiles) 354

245 - Development of multiple profiles exceeds maximum number of workings that may be assigned on a face 355

12.11 Errors in the assignment of technological parameters for profile and point workings 355

250 - Impossible to apply setup to an open profile as reference code is missing 355
251 - Impossible to apply a technological point as reference code is missing 355
252 - Impossible to assign open profiles 355
253 - Technology replacements were carried out 355
254 - Could not replace a technology 355

12.12 Errors assigning Entry/Exit segments to profile 355

271 - Enter/Exit profile: cannot solve a 3D arc 355
272 - Enter/Exit profile: programmed geometry is not compatible with the request for tool compensation 356
273 - Enter/Exit profile: cannot solve a covering segment, if the profile is not closed 356

12.13 Errors in applying tool corrections 356

261 - Tool compensation: correction exceeds the arc radius 356
262 - Tool compensation: correction exceeds the line 356
263 - Tool compensation: applied reduction to profile 357
265 - Tool compensation: error during correction on different xy-plane, with intersection solution of the line segments 357
266 - Tool compensation: error for correction in different xy-plane 357

267 - Tool compensation: inverting a correction should resolve an intersection or resume an interruption 357
268 - Tool compensation: suspension of correction without consecutive resumption had been required 357
269 - Tool compensation: suspension and consecutive resumption of correction can't compute a connection 357
270 - Tool compensation: a suspension and consecutive resumption of correction must verify geometric continuity of line segments 358
12.14 Errors of fragmentation and linearization of arcs in planes different from xy 358
255 - 3D arc linearization exceeds the maximum number of lines 358
256 - Impossible to linearize 3D arcs as reference linear code is missing 358
13 TpaCAD Customization 359
13.1 Environment 359
Startup 359
Activity 360
Working edit 361
Saving 363
Import format 364
13.2 Colours 366
Graphics 366
Layer 368
Construct 368
Field O 369
Nesting 369
13.3 Views 369
Customize views 369
Customize graphics 372
Grids and patterns 375
Mouse 377
13.4 Technology 377
Default codes 378
Default technology 379
13.5 Customize the "prototype" file 380
14 "Client" workings creation 381
15 Conversion Program 384
15.1 From DXF to TpaCAD format 384
15.2 From TpaCAD to DXF format 384
Parameters 384
Workings and Layers 384
Programmed Workings 385
Point workings (operation code between: 1-1000) 385
Setup workings (operation code between: 1-1000) 385
Profile working of linear typology 385
Profile working of arc typology (xy plan) 386
Profile working of arc typology (no xy plan) 386
15.3 From ISO format to TpaCAD format 386
Settings 387
15.4 From TpaCAD format to ISO format 388
Settings 389
Syntax and Examples 390

Milling 391
Drilling working 392
15.5 From TpaCAD format to Edicad format 393
Translation Mode 393
General information on piece 393
Programmed workings 393
Point workings 393
Logical workings 394
Setup 394
Profile 395
15.6 TpaCAD Program 396
Header lines 397
Advanced Tools In Face Program 397
Section of assigned working in face program 398
Working: Hole 398
Working: Mill setup 400
Working: Line 400
Working: Arc in face plan 401

1 Overview

1.1 Introduction



TpaCAD version 2.4.30

TpaCAD is a CAD/CAM system built in a graphic environment which allows the creation, the modification, and the import of working programs and the development of customized Macros and Subroutines to program numeric control machines in the wood, metal, marble and plastics industry.

Work area offers menu and multifunctional control panels (Ribbons) arranged to create a simplified environment of working program building.
The working program is provided in working lists arranged in the application face.
The basic geometry on which a program is defined is a parallelepiped, assigned with three dimensions (length, height and thickness) and six application faces. To this basic geometry you can add generically composite and oriented planes (variable geometries: flat, curved, or surface) to which a working list can be associated.

The working program is represented both in graphical and text format with immediate interaction between the two representations.
The representation in graphical format can be made in 2D or in 3D or in plane development of the piece (box view), where the 2D or box graph allow the viewing of the workings on the plane of a single face, while the 3D graph allow the viewing of all workings on a piece. The representation can be rotated (on three rotation planes, independent the one of the other), enlarged, reduced (with multiple level zoom) or centred, depending on requirements.

The **graphic** representation allows the interactive selection of a working or of a set of workings, where you can activate multiple display filters. The graphic representation is based on multiple tools:

- cross cursor
- ruler
- constant step or scattered element grid with possibility of customizing the grid elements (mesh, vertices,...).

The representation in **text** format allows a structured view of the face program. It includes, in fact, all programmed blocks, and also, those blocks which do not have an associated graphic representation:

- blocks of conditional statements (IF.. ELSEIF.. ELSE.. ENDIF)
- blocks corresponding to programmed errors;
- assignments of local variables;
- commented blocks.

More specifically, the program text appears indented, such as to highlight the structure assigned by the logical conditions inserted.
The text format is an ASCII representation of the program and enables:

- single or multiple selections;
- in the case of complex workings (subroutines and macros), the display of single workings which correspond to its development;
- in the case of complex workings applied on more faces (called induced), the display of the working list that correspond to the development of each face.

The workings can be inserted by selecting from a graphic palette, by inserting geometric elements and by applying CAD tools such as, for example, text writing and emptying of closed areas.
Workings can be modified as follows:

- by acting directly on every single working;
- by applying modification common to a set of workings;
- by applying geometric transforms to a set of workings (translation, symmetry, repetitions);
- by applying profile handling tools (scale, inversion, disconnections, breaks, tool compensation);

A lot of **tools** are available for a targeted handling of the concerned working program:

- general tools: translation, rotation, symmetries, repetitions, serial repetition on established paths, exploded view of subroutines or macros;
- profile tools: inversion, scale, technology application, joints of profiles, editing of vertices, closing and opening application, fragmentation and minimization, interruption, extent of profiles;
- CAD tools: generation of texts, emptying of area, cut of profiles, generation of spline curves from polylines, shape nesting, use of workings which apply geometric transforms.

A peculiar aspect in the assignment of workings is the **parametric programming**, allowing the use of

- piece variables
- mathematical, geometrical, statistical, logical, string-manipulation functions
- technological functions

The parametric programming can be used to assign program variables, variable geometries and working parameters.
It is important to highlight the possibility to assign custom functions. These are functions which work out a logic of calculation defined according to custom needs, and which can be used at every programming level.

Functions and variable arguments, available in parametric programming, make an effective and complete control on the context in which TpaCAD and the individual working program are operating:

- technology
- configuration settings
- execution mode
- geometrical characterizations of the piece

The multi-purpose geometric library function is particularly useful, since it provides an immediate solution to problems of geometrical nature, even very complex ones.
The high number of functions and variable arguments, available in parametric programming, has suggested to design **contextual helps** during programming, so:

- you can select the function (or variable argument) form a sorted list
- you can request to display of a help concerning the call syntax of a function

General piece assignments enable a great number of logical conditions which characterize the **executive composition** of a working program. Therefore, one only recorded program can generate an unlimited number of different machining plans with different geometrical characterizations (dimensions, assignments of the work faces), execution mode (normal, mirrors), exclusions, reassignment of program variables (offsets, cycle variables).

The panel and/or shape Nesting function offers a nesting system integrated with the programming environment of the single pieces, with:

- dynamic opening of nesting lists
- automatic detection of programs generated by Tpa CAD
- direct import from other formats (G-code, DXF)
- direct generation of reports, production files, labels.

1.2 Activation and operating mode

The program is available in three operating modes:

- **Essential**
- **Standard**
- **Professional**

The **Essential** mode corresponds to the minimum working level: this mode does not have any direct correspondence to previous versions of the product (see: TpaEdi32) and it is designed for an environment specifically targeted to editor functions.

The **Standard** mode corresponds to an intermediate operating level, already called basic working level.

The **Professional** mode corresponds to the advanced operating level. Compared to the basic level, the additional commands and functions are:

- text generation
- features of assignment and application of custom fonts
- emptying of area
- generation of spline curves
- cutting tools and profile construction
- position utilities
- assignment of fictive faces with the indication of reference face
- full functionality of programmed induced calls
- tool for creation of Fictive faces from programmed geometry
- assignment of automatic faces
- extension of parametric programming with addition of Custom functions and of the global function codes
- insertion of workings applying geometrical transforms (STOOL codes)
- extension of functionalities for tool compensation (suspension and side change in profile compensation)
- completeness of Display criteria during printing
- Bookmarks assignment functionality
- snap functionality between elements assigned in different faces, enabled in interactive procedures
- automatic conversion in other file formats during program storage

- possibility to add a custom typology of files in program reading

The enabled commands in **Professional** mode are only highlighted inside the manual by the symbol



To the Professional mode specific functionalities can be added:
- assignment of non-flat working faces (curved faces)
- assignment of piece modelling by extrusion
- assignment of composite working faces (surfaces)

TpaCAD operating functioning is protected by the presence of an USB hardware key according to the enquiries of the machine manufacturer. The hardware key can be moved from a computer to another, allowing the operation in Professional, Standard or Essential Mode on different TpaCAD installations, obviously not at the same time. As a matter of fact, the presence of the key is controlled on each request for the execution of specific commands.

ATTENTION: the hardware key can be programmed to allow the use of the TpaCAD installation package, but for the execution environment only. In this case the launch of TpaCAD fails (a message in English informs that the key excludes the program launch). Let us talk about functionalities: **Off-Line**.
An **Off-Line** programmed key can recognize one of the three above-mentioned modes: Professional, Standard or Essential.

If the reading of the enabling key fails, the **Demo** mode is enabled and has unlimited time. When the Demo mode is active, a window appears to notify that the selected installation does not allow to launch a complete function.

If this window is displayed, even if a properly programmed hardware key exists, it means that something is malfunctioning:
- the hardware key is not properly read or is not inserted into the proper port.

In this case you should perform all the needed tests.

In the **Demo** mode not all program functions are available, more specifically:
- the minimum access level is always active;
- it is not possible to save the programs;
- it is not possible to optimize the programs;
- it is not possible to create, modify, or delete the user workings;
- it is not possible to create, modify, or delete the custom fonts;
- it is not possible to change the plant.

As a standard condition, the **Demo** mode works under Professional mode, and makes some advanced and specific functionalities available, such as:
- fictive faces, also in the definition of curved faces and surfaces
- modelling
- use of custom font
- use of base custom functions.

It is also possible to activate the Standard or the Essential mode, in order to value the differences between the different modes. From the  menu, the command **DEMO operating mode**  , is available. It is activated in Demo mode only and if the program is closed.

The table below compares the three operating modes:

	Professional	**Standard**	**Essential**
Database of the workings			
• Full use of database marked by the manufacturer	√		
• Use of unsigned custom databases	√	√	
• Modification of unsigned custom database	√	√	
• Management of database of client workings	√	√	√
Configuration Management			
• Modification of TpaCAD configuration	√	√	√
• Definition and Modification of the Custom Features customized by the manufacturer	√		

	Professional	Standard	Essential
• Plant selection	√	√	
• Definition of Global Variables	√	√	
• Control of Advanced Configuration	√		
Piece configuration			
• Geometry of the piece in Absolute System	√	√	
• Assignment of fictive faces in local systems	√		
• Assignment of parameters to the fictive faces	√	√	
• Assignment of curved faces (fictive, automatic)	√Θ		
• Assignment of surfaces	√Θ		
• Modelling for extrusion	√Θ		
• Sequence of the workings	√	√	
• Maximum personalization of customizable sections	√	√	[note[1]]
• Possibility of locking the modification for program sections	√	√	
• Management of the piece-face	√	√	√
[note[1]] the Essential functionality excludes the management of the custom sections called: Section of additional info, Section of constraints			
Opening/Creation of a program			
• Opening programs in another format (imports: DXF, ISO,…)	√	√	√
• Customizing a single program import	√	√	
• Opening basic macro-programs	√	√	
• Creating custom macro-programs	√	√	
• Creating protected programs or subroutines	√	√	√
• Adding a custom extension for program opening	√		
• Maximum program dimensioning (number of programmable lines)	√	√	[note[1]]
• Conversion command of Program storage	√	√	√
[note[1]] the Essential functionality limits the number of programmable workings to 100000 (for each face)			
Saving a program			
• Recording programs in another format (DXF, ISO...)	√		
• Recording custom macro-programs	√	√	
Program print			
• Availability of programmable options	√	√	
• Maximum control of the programmable options	√		
General program utilities			
• Completeness of General commands for Program modification (Edit menu)	√	√	[note[1]]

	Professional	Standard	Essential
• Maximum customization of the modification and display filters	√	√	[note2]
• Logical exclusions	√	√	
• Creation of fictive faces from programmed geometries	√		
• Creation of surfaces from programmed geometries	√Θ		
• Creation of modelling from programmed geometries	√Θ		
• Creation of custom fonts from programmed geometries	√		
• Measurements on the piece	√	√	√
• Creation of dimensioning	√		
• Optimization preview	√	√	
• Overall program transforms	√		

[note1] (menu: Edit, group: Set) the Essential feature excludes the command: Set Special filters
[note2] (menu: Edit, group: Modify) the Essential feature excludes the command: Solve

Working codes			
• ISO curve application codes	√	√	
• Emptying process codes for closed areas	√		
• Text generation codes using system fonts	√		
• Text generation codes using custom fonts	√		
• Tool (STOOL) programming codes	√		
• Additional codes for the application of logical cycles (global functions)	√		
• Programming codes for inclined planes (automatic faces)	√		
Working customization			
• Enabling "Standard" (C, L, B, O, M, K) properties	√	√	√
• Enabling "additional" (N, K1, K2, V) properties	√	√	
• Possibility of locking specific property values	√	√	
• Programming property parameters	√	√	
• Additional features in tool compensation (suspension and side change)	√		
• Functionalities for executing sharp corners in the profiles	√		
• Development of programmed induced calls (in piece-face)	√	√	
• Development of programmed induced calls (in all the faces)	√		
• Possibility of calling a macro-program from generic SUB codes	√	√	
Drawing features			
• Modification of the current working by interactive procedure	√	√	
• Drawing menu (points, arcs, circles, ellipse, polylines)	√	√	√
• Use of bookmarks	√		
• Snap between faces	√		

	Professional	Standard	Essential
• Snap on modelling elements	√Θ		
Tools and constructions			
• General tools completeness	√	√	[note[1]]
• General tools completeness for profile modification	√	√	[note[2]]
• Offset application to profiles	√	√	
• Application of bridges to profiles	√	√	
• Application to profiles of depth feed	√	√	
• Profile rotation on coordinated planes	√	√	
• Profile Cut and selective Construction tools	√		
• Emptying of area	√		
• Text generation using system fonts	√		
• Text generation using custom fonts	√		
• Spline curve generation	√		
• "Profile Nesting" tools	√Θ		
[note[1]] (menu: Tools, group: General) the Essential feature excludes the commands: Generic Symmetry, Repetition on a profile, Explosion [note[2]] (menu: Tools, group: Change profiles) the Essential feature excludes the commands: Pull profile, Extend			
Parametric Programming			
• Use of the working name	√	√	
• Use of Custom functions	√		
Graphic options			
• Grid management for points (custom grid)	√	√	

Key:
√ handled entry
Θ requires an additional HW setting

1.3 Access right to the system

Next to the operating mode defined by the enabling key, TpaCAD includes different access levels to the system.

- **User** is the level with most access restrictions. It is not possible to modify any of the protected settings, open or modify programs of macro, open or modify files assigning the workings used in the TpaCAD program. At the launch of TpaCAD this access level starts.
- **Assistance** can be used to assign an access level or to modify a program. The level also allows to modify a limited part of the TpaCAD configuration and it is higher than the User level.
- **Manufacturer** is the level used to configure and write programs with macro typology. At this level all possible changes can be made.

The access to each mode is conditioned by the knowledge of the relating key word. The User level has the lowest access level, while the Manufacturer level has the highest access level.
To access to the level required, the procedure is the following:

1. With almost one TpaCAD application opened, press the **[CTRL + * (asterisk)]** key short-cut. A window opens where to enter the password corresponding to the level. Alternatively, there is an icon to the right of of the Application bar of Windows ®: by clicking the icon with the right mouse key, it is possible to display a menu in which the **Change pass level** command appears.
2. Enter the key word of the level required and select the button **[OK]**.

If the entered password is not correct, the error message "Warning! Wrong Password!!!" appears.
Once the password has been entered, you have already logged into the corresponding access level.

The access level that is selected with the same mode as above is common to the TPA environment, which is installed and works in the computer.

ATTENTION: in TpaCAD a local mode to change the access level can be made available by activating the **Stand-alone** functionality (according to the configuration). More specifically, this second mode:
- is added to the already described mode and it does not replace it. That is, the procedure that can be activated by the keyboard short-cuts **[CTRL+* (asterisk]** still works also in TpaCAD;
- is available only if the **Stand-alone** functionality is active (according to the configuration) and if the machine manufacturer has activated a local account in the TpaCAD environment;
- the manufacturer must release the activated account (password) to the customer;
- allows you to activate an account at the **Manufacturer** level in TpaCAD only. This means that the access level, that is selected here, is not active for the TPA environment, which is installed and working in the computer.

The **Password Level** command can be selected from the menu . When the inserted password is correct, the operator can change it, in order to customize the privileged access to their own installation of the cad system.

ATTENTION: The machine manufacturer can activate a local account in the TpaCAD environment only after accessing the same command from the **Manufacturer** level recognized for the TPA environment, and activated through:
- **[CTRL + * (asterisk)]** key short-cut; or
- command of the Windows application bar; or
- the same **Password level** command selected from the menu of TpaCAD.

1.4 Multilingual support

TpaCAD supports the display of text in several languages. Normally, full support is provided in ten languages:

- English
- Italian
- French
- German
- Spanish
- Czech
- Russian
- Dutch
- Polish
- Chinese

Translated helps are available for these languages in contextual and printable version.

The translation of interface messages only is available for the following group of languages:

- Hebrew
- Hungarian
- Japanese
- Latvian
- Slovenian
- Bulgarian
- Romanian
- Portuguese
- Swedish
- Turkish

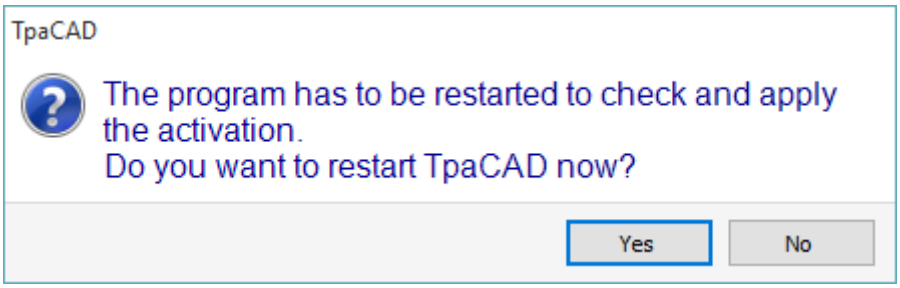
There is no guarantee that the messages and the manuals are fully updated and aligned at every release of version. Messages that have not yet been translated into their specific language are always provided in English.

Apart from these, it is possible to extend the support to other languages, if needed. Some limitations are presented by languages, such as those of the East, requiring the use of a special set of characters or those that do not follow the typical left -> right orientation of the Western languages.

Change of language
The language may be changed at any log-on level. To change the selection of the language you have to use the **[CTRL + /]** short-cut key or click on the icon from the "**application tool bar**" of Windows®.
In the opening window, select the requested language and click the button **[OK]**.

ATTENTION: if TpaCAD is not in a TPA environment, the language change is locally available for the application by activating the **Stand-alone** functionality (from the configuration). In particular:

- in the status bar of TpaCAD there is an additional working selection list of the available languages. Even if not enabled, the cell in the status bar shows the abbreviation of the active language. If the user wants to change the language when the program is closed, they can confirm the direct restart of TpaCAD as suggested in the following window:



otherwise the language is not changed immediately, but only at the next restart of TpaCAD.

1.5 Format compatibility

TpaCAD program format is not compatible with the previous CAD TPA (TpaEdi32, Edicad) versions. However, TpaCAD can read programs of previous versions.
To create programs with TpaCAD in a version compatible with TpaEdi32, it is necessary to use the command Save in its proper format.

1.6 New functionalities

Below is a list of the most significant new functionalities available in TpaCAD, compared to the previous version of the program, TpaEdi32.
The fully redefined graphical interface is certainly the most evident difference; however, there are not only graphical differences. Let us see the main aspects:

- the use of Ribbon, that has replaced the classical menu, has led to a full revision of the command group. There are less Ribbon tabs than the previous primary options of the menu
- some controls can be "repositioned" within a graphic area of the application program. For example, it is possible to move the area of the Working data to the right or to the left;
- the area for displaying both commands and errors is now structured into a group of tabs automatically updated, also including the table of the program <j> variables and the debug window that in TpaEdi32 respond to dialog boxes selectable from the menu
- the area for the complex assignments of the program (dimensions, variables…) remains structured into a group of tabs and is always displayed. So, you can see the dimensions and the variables or a setting of a custom section also during the assignment of a working in a face of the piece
- the area of the ASCII text can be now expanded and downsized: it is possible to "close" the ASCII text and open it again where needed, for example, to assign logical condition branches
- the area of graphic representation of the piece has been totally restyled. Effects of transparency and brightness, as well as the ability to assign a background image to the piece and/or to the current face that adapts to the affected area, make the visualization closer to a three-dimensional view of the workpiece and to the material which it is made of (wood, glass…)
- the assignment area of the single program line (working) is enriched by a command bar giving the possibility to scroll the list of working, quickly move a program line, insert a copy of the current working, change the options of the same assignment area
- the interactive areas manage specific commands activated by contextual menus, that are displayed when clicking the right mouse button on the corresponding area and are made up according to the operating context. This aspect makes uniform the operation in the various areas and also allows the management of more customizations in the menus of commands themselves: for example, the area of graphic representation radically changes the local menu, with the activation of interactive procedures
- the working selection bar is fully restyled and widely customizable.

There are many functional changes:

- a program in TpaCAD native format or compatible with one of the configured import modules can be directly opened through the dragging method in the graph area of TpaCAD
- the program open window incorporates all the choices you can make when uploading a piece:

  - activation of an import module and relating accessory assignments
  - opening of a program in native format as a copy
- list of recent programs increased in number with a local menu
- the number of import and export configurable modules is increased to 8
- the graphic representation of the piece allows the display of the development on the plane of the parallelepiped base faces (box view), besides the traditional 3D (three-dimensional) and 2D views (view on the XY face plane)
- the graphic representation displays all the workings assigned in the piece, also in face view: different colours allow the user to distinguish the working current list from the other ones
- the graphic representation interacts also at the level of the direct change of face view by clicking directly within the overall area of the face
- the graphic representation of a profile can select directly from the menu among the different bounding box viewing options (horizontal, vertical, trim functionality). For instance: the functionality of full or linear segment, meeting for instance the needs of *seeing* both an empty area (full segment) and the overall dimension of a single profile
- in the working assignment it is now possible to integrate the direct settings with interactive acquisition of coordinates (coordinates of an application point or of a rotation centre or of an auxiliary point)
- the resolution of the induced call is now integrated in the management of the master call, with consequent deletion of the induced rows in the text of the program and better comprehension of the functionality itself. The expansion required on the master call allows you to directly find the assigned workings for the additional calls
- the functionality of direct assignment of Client application codes of subroutines had a significant development, allowing the final user of TpaCAD to tailor their own database of workings covering a considerable part of his own customization needs
- the working selections are maintained also with the change of the active view, making possible the application of the overall program tools (technology assignment, reduction, fragmentation, linearization and profile connection) also considering the active selections, besides the application of active filters of view and change
- the commands to search and replace the working elements are extended to the whole program; furthermore, a correspondence and/or a replacement of properties can be considered, as well. The replacement of parametric forms now can carry out both selective replacements of variables or variable arguments and generic replacements of sub-strings
- the commands of overall property assignment allow also the direct assignment of a parametric setting
- new commands allow the user to insert directly in the program list predefined structures of logical cycles: IF… ENDIF, IF…ELSEIF…ELSE…ENDIF, making also possible the direct association of the logical structure with a group of workings
- the help menu of parametric programming can always be recalled by clicking the right mouse button and includes the access to all the program variables (‘o’, ‘v’, and ‘r’) and also to technological information
- compared to the assignment mode of the information needed, the selection of the Tools is unified: directly from the window you can integrate direct settings with other ones acquired in an interactive way. This has led to the development of several tools, as well as their same simplification.
- the activation of interactive procedures keeps the view active (2D, 3D or box view) and pays a particular attention to the helps during the execution of the procedure, by means of synthetic but functional tooltips directly in the graphic area and explanatory messages in the command area
- the snap modes that can be activated during the execution of interactive procedures enable to consider:
  - the depth of snap elements
  - assigned elements in different faces,
  - graphic references appropriated added (markers, bookmarks), the use of which allows the snap extension not only to all the programmed entities on the piece, but also to accessory entities suitably found
  - a snap either on dotted or horizontal/vertical lined grid

  The snap between faces allows, for instance:
  - the hole positioning in face 3 (front) "on" a hole in face 1 (top)
  - the measure of the distance between programmed elements in different faces
- the assignment of the sequences allows the user to find and to select the working graph and the area selection. Furthermore, the list includes the construct workings
- more and better configurations allow the user to arrange the windows for the assignment of the filters in such a way that it fully meets the specific needs of the single application
- the assignment of local parameters to the TpaCAD environment allows for a more efficient use, both on stand-alone mode and integrated with TPA environment
- specific tests and developments are aimed to the treatment of "large programs" (for example, the number of programmed lines is greater than 250,000).

Let us see the changes concerning directly a program:
- in the editor phase, some standard file properties and further customized properties of the program are displayed and in this latter, the change status can be configured for each section of a specific program. It is also possible to force a privileged setting for the unit of measurement of a program. Adding the management of a progressive number for the storage allows the user to optimize the generation and the maintenance of the optimized files.
- the execution modes are stored in the program and it is possible to automatically assign a particular execution
- the maximum number of the <o> and <v> variables is increased to 16 (from 8)
- programming fictive faces adds the possibility to assign curved planes and/or surfaces

- the maximum number of the auxiliary parameters defined in the assignment of a fictive face is increased to 5 (from 3)
- it is possible to assign a piece modelling, selectable among three different modes (dedicated section, assignment in variable geometries or in programmed workings)
- two new fields, called K1 and K2, are added to the working properties. The maximum assignable value is 255; for the new fields it is possible to assign display and block filters for the change and program exclusions
- opening a program written in Edicad or TpaEdi32 format, K, K1 and K2 properties can retrieve programming previously assigned in working parameters: this allows a finer structuring of the workings, with an optimal use of the now available properties
- management of IF…ELSE…ENDIF adds the ELSEIF instruction
- a new operating code (EXIT) allows to improve the executive control of an IF…ELSE…ENDIF cycle, allowing the user to program the direct output from the cycle or from the program flow itself
- programming in face-piece is strengthened, so, for example, it is possible to recall the application of a working in automatic face by indicating the name of the face
- large new STOOL codes (workings of programmed tools) increase the integration of the Tool and Working capacities in a considerable way. The user can change or create profile for:
  - tool compensation
  - application of fillets and bridges
  - fragmentation and linearization
  - linearization or progress in developing the depth
  - profile joints

  The basic operation of the new codes is similar to that of the corresponding advanced tools in profile, with the advantage that the created profiles dynamically adapt to changes of the original profiles, besides the fact that on the whole it can work on more than one profile.
- the functionality of generating spline curves adds the management of the *Cardinal spline* and of the *Paths*
- functionality of custom font assignment and application is added.

1.7 System requirements

Before installing TpaCAD check that the computer has the minimum hardware and software requirements.

Basic installation
Required operating systems: Microsoft® Windows® 7 Enterprise, Ultimate, Professional or Home Premium; Microsoft® Windows Vista® Enterprise, Business, Ultimate or Home Premium (SP1 or next version); or Microsoft® Windows® XP Professional or Home edition (SP2 or next version), Microsoft® Windows® 8, Microsoft® Windows® 10.
Minimum RAM memory required: 4 GB.
The installation requires 700 MB of free hard disk space.
Minimum requirements for graph card and monitor: 1024x768 32-bit colours with 256 MB Ram.
We recommend the user to check and update the graph card drivers of the installation computer in order to optimize the TpaCAD performances.

TpaCAD installation with Nesting functionality
Microsoft® Windows® 10, Windows 11 operating system required.
Minimum RAM required is 16 GB.
Installation requires 700 MB of free hard disk space.
Minimum video card and monitor requirements: 1024x768 32-bit color with 256 MB Ram.
A core count of at least 8.
It is recommended to check and update the graphics card drivers on the installation computer, in order to optimize the performance of TpaCAD.

TpaCAD uses OpenGL SW and requires almost the 1.2 version; if this version is not installed in the computer, the application program closes after a warning message.

Running TpaCAD on virtual machines

There is no guarantee regarding the running of TpaCAD on virtual machines. Running TpaCAD on these machines with respect to the running process on the same hardware with a native Windows may lead to a reduced graphic quality and a slowdown of the overall performances.

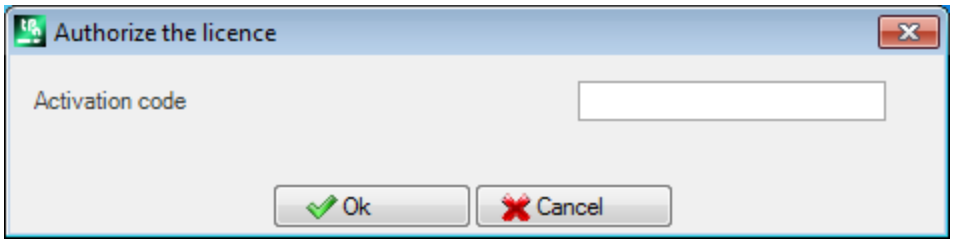
Performance problems depend on the type of the virtual machine and assigned configuration.

1.8 Checking control at TpaCAD launch

At the launch of TpaCAD, it is possible to diagnose some situations that do not allow the program to work. For all these situations a message in English appears and the application is closed.
Let us see the possible cases:

- **"Critical error !!"**: reports a severe error occurred while initializing. This error can occur due to an incorrect installation of the TpaCAD package or because the code or data files required for the functioning of the application have been damaged. We suggest a complete installation.
- **"Error in configuration loading:.."**: signals a severe error situation occurred while loading the TpaCAD configuration file. This error message can notify a specific reporting, such as the lack of file or an access error or an error whose format has been found in the same file. According to the error cases TpaCAD is able to suggest a method to restore the situation such as setting a default configuration or recovering the configuration from a backup copy. The error may indicate that the configuration file has been removed or damaged.
- **"OpenGl: unsupported version (min: 1.2) !!"**: the installed version of the OpenGL software is not compatible with the application. It is necessary to upgrade at least to the 1.2.version number. This message is typical of a working on Virtual machines, that often use very old versions of the OpenGL software.
- **"It's not checked the license to use TpaCAD !!"**: a licence to use "Off-line" the TpaCAD package has been recognized.
- **"Too many instances !!"**: 4 (four) instances of the application program have been already launched.

At the launch of TpaCAD and after a custom package installation, the window below may appear:



This window shows that the performed installation has copied a custom database of workings which requires a particular procedure to be activated: let us say that the database was signed by the machine manufacturer and that also an activation code to be assigned is requested when the program starts. The activation code has to be communicated by the machine manufacturer and assigned to each installation: for instance, when you install the package on two different computers, in both cases the activation process must be activated,
The operation of activation can also be postponed, but this may reduce the possibilities of using the package:

- close the window after setting the required code to fully activate the usage licence. In this case a message warns that the TpaCAD program is being closed: restart TpaCAD to check that the operation was successful. If the window requiring the authorization appears again at the launch, please contact the machine manufacturer;
- to postpone the operation, please close the window without setting the code: at the next launch of TpaCAD, the window will appear again.

If the license activation is not successfully executed, the custom working database is moreover loaded, but it will not be fully working.

At the launch of TpaCAD situations of partial or damaged operational features can be detected. In this case a window shows a list of the anomalies found. The list can also be recalled from the menu  **-> Check signals**.

Possible signals can concern following cases:

- the use of the manager of the USB hardware key failed: TpaCAD works in test mode only (demo);
- the activation of the demo mode following the failure to recognize the usb hardware key: consequently, the application program works in a limited way:
- the use of the manager of the access levels failed: consequently, it works at operator level only;
- the use of the manager for the messages failed: as a consequence, the messages are displayed, totally or partially, in Italian.
- the need to activate the license to use the custom database of the workings.

1.9 Helps

The TpaCAD installation includes some manuals available as online-helps in the HELP folder. They can be printed or directly invoked from the program, in the HELP folder.

The manuals are always available in the languages of the installation.
The installation of updates or of a new product version can update the manuals.

2 Versions and updates

2.1 Versions from 2.4.0 (May 22, 2020) to 2.4.30 (August 04, 2026)

Version 2.4.30 (August 04, 2026)

New features:
- Nesting: the **Cutting profiles** flag in the parts list, if set, remains set, even when changing the program type from tcn to shaped.
- Added parameters to blade workings to refer the working application point to the maximum working depth and not to the entry point into the face
- Modification to the Lamello 3ax working so that it can also be performed without a **Professional** key
- In the **Sequences** assignment page, in the Face column, the face number is also displayed in addition to the panel image

Corrections:
- Fixed a serious error when opening a file in a shared One Drive folder if synchronization is active
- Fixed an issue where selecting a polyline in the **Draw** menu would not retain the settings on the first point
- Fixed issue where **Optimization preview** did not display the profile of shaped tools.

Version 2.4.29 (April 15, 2026)

Corrections:
- Nesting: preview of the scrap sheet was not visible when **Save label in LABEL subfolder** flag was enabled.
- Nesting: the scrap sheets were not saved if the flag enabling the creation of labels was disabled.
- Nesting: the options introduced in the **Save the results** window were not applied in some cases.
- Nesting: with the Automatic label positioning flag enabled and the label size check, when the panel had one of the dimensions smaller than one of the label dimensions and therefore it was necessary to rotate the label to fit into the piece, the label bitmap was not created and the file name was written as "*"
- Nesting: fixed the display of tooltips for the buttons for selecting a part and in the title of the File Open window called by selecting the buttons themselves.
- Nesting: fixed placement issue in square mode
- Fixed problem in mill-radius compensation with arcs on zx plane.
- Fixed problem in the **Rotate on a Cartesian plane** tool and in **STOOL STPLANE,** when the **Resolve Zx plane** flag was enabled in TpaCAD configuration.
- Lamello with blade and lamello with cutter workings were not graphed correctly when the Positive Z axis flag was enabled in the configuration when entering the face
- Lamello working with blade: the rotation Angle parameter on the lower face was not managed
- Cut of rectangle working: when the working started on one side and the bridges and filled radius were enabled, it could happen that the cut on one or more sides went beyond the size of the rectangle's side
- Fixed the maximum value of the **Number of bridges [AN]** parameter of the Cut of rectangle working in the workings database.

Version 2.4.28 (February 03, 2026)

New features:
- Nesting: added flag in Nesting Configuration to choose whether the folders of the labels of the scrap and waste sheets should be created as sub-folders or at the same level as the LABEL folder.
- Nesting: added management of the zig-zag entry option in the Cut of Rectangle working.
- A new tool has been added to the working toolbar that allows you to explode a SUBISO or SISO working into as many SUBISO and SISO operations as there are tool changes in the recalled ISO file. The ISO file must be written according to Tpa specifications.

Corrections:
- The **Apply overall program activations** configuration option was applied only if **Complete from prototype file in external program reading** and **Complete the workings read by an external program** options were enabled.
- Fixed display of a message in the status bar when the display of mill-radius compensation coordinates was enabled.
- Nesting: labels were printed even when label printing was not active, but the **Automatic label positioning** selection was active.

Version 2.4.27 (December 12, 2025)

New features:
- Nesting: in the label wizard of labels for waste and scrap sheets, only usable information is displayed.

Corrections:
- Nesting: in the label wizard of labels for waste and scrap sheets, only generic information such as date and unit of measurement could be configured.
- Nesting: if a sheet of non-defined material in the nesting configuration was present in the scrap sheet database, TpaCAD generated a serious error.
- Nesting: on the label, the overall sheet graphics were drawn incorrectly if the RightToLeft flag was enabled.
- Nesting: the help page, called by the Label Wizard for waste and scrap sheets, was not opening.
- Nesting: in the nesting label, the overall sheet graphics were not drawn if the flip nesting flag was enabled, even when the management of the labels on the lower face was not enabled.

Version 2.4.26 (October 20, 2025)

New features:
- Nesting: added generation of SVG format file to graphically represent the nesting solution.
- Nesting: in the Customize tab when you change the value in the **All** margins field, the values of the outer margins are changed and not the value of the inner margin.
- TpaCAD: added a flag in TpaCAD configuration that allows you to apply the tools applied during import also to TCN files recognized as created by external programs.

Corrections:
- Nesting: in some situations, the scrap label storage folder was not created. The error displayed was "Generic error in GDI+"
- Nesting: advanced scrap cutting was only enabled if label insertion was also enabled
- Nesting: if the DPI value for label printing, set in the nesting configuration, was different from 300, the edges could be positioned incorrectly.
- Nesting: fixed an error in sorting parts when sorting in a spiral outward direction
- TpaCAD: when a subroutine path contained spaces, it was always considered an absolute path. This could limit the portability of programmes from one custom to another.
- TpaCAD: fixed an issue where if a macro was called in a subroutine that in turn called a protected macro, a warning was generated “225 one or more jobs have been excluded”
- TpaCAD: fixed an issue that occurred when sequence management was enabled and workings were generated via the STOOL working: the same sequence numbers were assigned in the piece matrix, making the workings indistinguishable. The maximum number of assignable sequence levels is 5.

Version 2.4.25 (April 14, 2025)

New features:
- Nesting: changes to improve the placement of pieces with tool diameter different in the sheet
- Working Lamello with 3ax milling tool: added flag for anti-chipping management

Corrections:
- Nesting: fixed the problem of sheet sorting with priority enabled in presence of scrap sheets
- Nesting: fixed issue where nesting profiles were overlapping if the **Automatic label positioning** flag was disabled and the **Overlapping nesting profiles** flag was enabled

Version 2.4.24 (January 29, 2025)

New features:
- Change to the range of values that can be set in configuration to the **Minimum angle** parameter for the **Sharp corner cut** functionality
- Improved testing for recognition of an arc assimilated to a circle when performing cutter radius correction

Version 2.4.23a (November 25, 2024)

Corrections:
- Working Lamello with milling tool: profile data was not reported during optimization

Version 2.4.23 (October 23, 2024)

New features:
- Nesting:

1. added option of automatic positioning of piece label, when it is not possible to place it in the centre of the piece.
2. added scrap cuts on the sheet even when a scrap sheet is created from it
3. added label-barcode working of the scrap sheets in the nesting solution
4. added information in xml format report file on scrap sheets, label position, piece type and scrap pieces
5. added option in Nesting configuration, which allows placing the label on piece, even when the label size is larger than the size of the same work piece.
6. added new advanced scrap cutting operation with the ability to place labels in the cut-out areas
7. added Label Wizard to create specific labels for scrap sheets and areas generated from scrap cuts
8. added the option to hide the display of V and O variables
9. added working Lamello with milling tool on 3 axes machine

Corrections:
- Nesting: fixed situations of overlapping pieces and off-panel placements.

Version 2.4.22a (June 05, 2024)

New features:
- Workings **RECTANGLE** and **RECTANGLE EMPTYING (GREEK PATTERN)**: updated the database of workings and the macro Rect.tmcr with the addition of the parameter **Tolerance on the diameter**. This parameter is available and managed only if the technology replacement due to wear and tear is active.

Corrections:
- Nesting: fixed error when handling extra parts that in some cases made TpaCAD close down.

Version 2.4.22 (May 21, 2024)

New features:
- Added option **Store display status of active face when exiting the program**, enabling storing display status of active face when exiting TpaCAD
- Added command **Edit subroutine** to the Button bar in *Working assignment area,* to open a subroutine in another instance of TpaCAD
- In the tool **Apply bridges to profile** and **STOOL: APPLY BRIDGES** to profile the minimum number of bridges or interruptions is 1
- Nesting: added management of scrap sheets
- Nesting: changed the graphics management of custom images, which are now adapted to the container and the stretch is not executed
- Nesting: extra parts are managed even if the number of parts available is 0

Corrections:
- Nesting: fixed error where custom images were not loaded into the label, if the file path was longer than 100 characters
- Nesting: in some cases scraps were not added at the end of all the workings
- Nesting: fixed overlapping of pieces and placements outside the panel

Version 2.4.21 (March 11, 2024)

New features:
- Nesting: when starting the nesting functionality, checks and flags have been inserted on the global technology: tool compensation must not be set
- Nesting: in the creation window of manual clusters the command for checking collisions between parts has been added
- Nesting: images inserted in the label both from **Label Wizard**, CSV file, and from table of Nesting programs
- Nesting: row number increased in the program table from 300 to 500
- Workings **Lamello** modified: technological parameters added (machine, group, Electrospindle, tool and tool type) to the fixing hole

Corrections:
- Nesting tool and nesting solution: if the option **Use a comma as decimal separator** was enabled in the TpaCAD configuration, the workings resulting from nesting procedure were not created
- Nesting: the copy and paste function of numerical values in the **Pieces** table and **Sheets** table did not work

- Fixed issue whereby, in particular cases, TpaCAD generated a serious error if the Windows Print spool was disabled
- Fixed issue in the program description: in case a blank row was left between one comment row and another, the rows present after the blank one were deleted on the next opening
- Fixed various issues about mill-radius compensation, geometric library

Versions 2.4.20 (November 15, 2023)

New features:
- Nesting: added option **Optimize all in True Shape** in Nesting configuration (for further information see TpacadNt manual), which allows the optimization of the scrap areas produced by manual cluster nesting
- Nesting: the option **Shapes: Evaluate the external geometries** is also applied in the case of Nesting True Shape
- Nesting: in the graphical window of the manual cluster configuration the command of snap between rectangles, panels, and rectangular shaped parts has been added
- Nesting: in the graphical window of the manual cluster configuration two buttons to enable/disable displaying the overall dimension of the profiles with linear segment and the reference to the part placement have been added

Corrections:
- Emptying STOOL: in some very particular cases, the starting point of the emptying path was not properly calculated
- SRECT: if a value smaller than the emptying tool diameter was set to rectangle length, TpaCAD generated a serious error
- Nesting: if in the part list a key was pressed on a cell of the **Name** column, TpaCAD generated a serious error
- Nesting: when the unit of measurement of label was inch and program was inch, the label was not properly created
- Nesting tools: when in TpaCAD configuration, on the **Export** tab for the configured exporter the options **Enable in "Save"**, **Enable in "Export"**, **Enable in "Nesting"** were enabled and the **Nesting solution** tool was executed when saving the solution, TpaCAD generated a serious error.
- Nesting: when there was a manual cluster in the part list, the value of the **Solution number** field in the sheet solution window was not correctly written, and **Restart** button was enabled, although all its parts were rectangle or panel type
- Nesting: when there were manual clusters in the part list, a wrong value in **Pieces** field of the sheet solution window was displayed
- Nesting: when the **View areas** flag was enabled, the colour filling the part was white in some cases, instead of the part colour

Version 2.4.19a (September 6, 2023)

Corrections:
- Fixed GDI+ error issue when saving labels containing QrCode. The problem was introduced in version 2.4.19.

Versions 2.4.19 (July 31, 2023)

New features:
- Added universal tool management
- Nesting: the label is not inserted into a piece, if its dimensions are greater than the dimensions of the piece
- Modification to the Lamello working: added **Rotation angle** parameter of the working on the plane
- Modification to the Lamello working with milling tool: added the parameters **Groove length** and **Rotation angle** of the working on the plane
- Nesting: added option defining how the **Priority** field value is interpreted in parts, clusters and sheets
- Nesting: improvement of graphic window commands for creating a cluster: added the ability to move and rotate a part directly from the keyboard

Corrections:
- Nesting: in the Length [DL] and Height [DH] parameters of the BARCODE working the values set in the label Wizard were not reported
- Nesting: fixed cases where manual clusters were placed overlapping
- Nesting: the first piece of the list was not deleted when using the single line deletion command
- Nesting: the box with the active colour was not displayed in the colour selection window or in the sheet viewing window
- TpaCAD: fixed problem during the emptying procedure when the **Empty outwards** option is enabled. The improved emptying procedure has led to the recovery of many areas that were not previously emptied, but it was not taken into account that these areas always have to be emptied after primary emptying
- In a multiple setup, the replacement of the worn tool, if enabled, was done only on the first setup.

- TpaCAD: fixed case of serious error when closing the TpaCAD Configuration window. The problem was found on PCs with Intel Graphics UDH 730 or Intel Iris xe graphics chipset and was solved by updating the chipset drivers
- Print program and nesting label: in some cases the label was printed smaller than the requested size

Version 2.4.18 (May 18, 2023)

New features:
- Added Lamello working with milling tool
- Added the ability to insert information in the QR code without additional aliases
- Nesting: added manual cluster management

Corrections:
- TpaCAD: fixed problem in TpaCAD operation from the command line, so even if the flag **Prompt for the execution of a command notification** was disabled, the confirmation request window was always displayed.

Version 2.4.17 (February 24, 2023)

New features:
- TpaCAD: added a command in the start mode of TpaCAD from the command line (/W), which saves the file without showing the saving confirmation window. The new command is idFileSaveSilentTo|"full pathname". Example: +idFileSaveSilentTo|"c:\albatros\product\programBase.tcn"

Version 2.4.16 (February 17, 2023)

New features:
- Nesting: introduced the management of labels on bottom face when Nesting-flip operation is enabled
- STOOL Rectangle emptying (Greek pattern): a parameter assigning a coverage value as a percentage on radius was added
- Menu Favourite tools: a menu with a list of favourite tools was added to the quick access toolbar

Corrections:
- Editing of workings: in some cases the confirmation key for inserting the working was disabled
- Emptying procedure: some calculation limit situations for which emptying was not performed were corrected
- Emptying procedure: the procedure for managing the residual areas was improved
- Nesting functionality: in the report file the data concerning the central position (XC;YC) on the overall rectangles of placements was written without considering the value of margins applied to the sheet.
- TpaLangs would close with an error if the ALT key was pressed without having any message file open.

Version 2.4.15 (December 20, 2022)

New features:
- Optimization of scrolling of the workings in the ASCII text area
- Added shortcut keys [CTRL+2] and [CTRL+3] to enable the working data area and the ASCII text area
- X Fitting, Y Fitting, X Repeat and Y Repeat workings: the database of workings and Repeaty.tmcr and Repeatx.tmcr macros were updated with the addition of a parameter to duplicate the working on the opposide side
- Addition of the description window of the list of shortcuts: it is recalled up by pressing the icon  .
- STOOL Rectangle emptying: a check on the radius value was added. If the set value is lower than the value of the tool radius, an error message is signalled.

Corrections:
- Solved the problem where the shortcut keys [CTRL+W] to recall the **Window Zoom** command, [CTRL+Shift+W] for the **Previous Zoom** command and F6 for the **Extension Zoom** command were enabled only after recalling the commands from the contextual menu.

Version 2.4.14 (October 25, 2022)

New features:
- Nesting functionality: the management of the rotation of the edges of the labels when the rotation is 180 or 270 degrees was added in *TrueShape* nesting. In rectangular nesting the rotation of the edges is 0 or 90 degrees.

Corrections:
- Solved the problem where in **Optimization preview** the modelled piece was not displayed correctly
- Nesting functionality: in the table for the definition of the nesting pieces, the column for enabling mirroring was displayed only if all types of files were enabled
- Nesting functionality: font changed from "Verdana" to "Microsoft Sans Serif" in the Nesting report printing to have a better font definition
- Nesting functionality: problem of 90 degree rotation solved in case of Nesting TrueShape
- Assignment of **C Property or Comment** to several workings: if the workings present in the sequence list were commented, when the comment flag was removed, they were no longer present in the sequence list
- Nesting: in the case of Nesting TrueShape if the pieces were rotated, length and height dimensions were displayed interchanged in the labels

Version 2.4.13 (July 15, 2022)

New features:
- Display filter for tool type added in tool table

Corrections:
- Solved problem where working flags set in insertion or modification were not evaluated correctly
- Palette of workings: in some cases a working, which was not applicable on the active face, was still displayed
- Nesting functionality: piece numbering in the graphic area has been increased to 9999
- Nesting functionality: if label management was deactivated, the rotation assignment was not changed
- Nesting functionality: in the nesting report, even in the case of shapes, the total occupied area was displayed based on the panel size. This meant that the occupied area was greater than the sheet area
- Reading ISO file: T, F, S, and correction fields, which are written in the ISO file, were never read, even if they were not assigned to the external or internal setup
- Reading ISO file: if a unit conversion was applied, it was not applied to the coordinates of the centres and arc radii
- Solved several problems related to: emptying of areas, cutter radius correction, geometric library, profile attachment tools
- Export module: in an inch program the value of the F speed field was exported as an integer value. It now keeps 3 decimal places.
- Macro ISO.TMCR: solved problem of assignment of correction values and tool radius
- Workings: corrections to 5-axis curve, ISO Subroutine and Curve workings

Version 2.4.12 (May 10, 2022)

New features:
- All prototype files are saved only in native TPACAD format and not in ASCII format
- Improved the interactive acquisition procedure on the 2D representation in the presence of induced calls
- Tabulation of faces: indication of the induced lines has been added
- Tabulation of faces: image of indication of fictive face is displayed for all fictive faces present
- Nesting functionality: the option **Shapes: Evaluate the nesting geometries** has been added to the Nesting configuration. This selection affects pieces that are inserted into the nesting project as Shaped workpieces. Assess the space outside the overall rectangle of *Nesting geometry* for assigning an original area around, so as to safeguard placements contiguous to the *Nesting geometry*
- Nesting functionality: management of Nesting-flip operations added
- Nesting functionality: the management of head and tail workings has been added to the prototype file using a specific formalism. Workings in face 1 and 2 are read (if Nesting-flip functionality is active), descriptions are kept also in the *.TCN files of nesting solution
- Nesting functionality: the **Minimum area** option added in the Nesting configuration. It applies the bridges only to small pieces. In this way, the small pieces are excluded from the management of the optimised cut.
- Import module from ISO format: values assigned to the setup as T field and S field are also propagated to the profile
- Import module from ISO format: if there are more G0 the values of the rotating axes are propagated if they are not assigned to the subsequent G0 instructions
- TpaLangs application: the length of message text identifiers has been increased to 250 characters

Corrections:
- Nesting tools: fixed error situations of placement in case of manual cluster of islands
- Nesting functionality: solved situations where the zeta feeds were applied to both the pre-cut and the cut
- Nesting functionality: completion programs were not recorded if the upper face had no workings
- Export module in ISO format: when exporting an oriented profile with explosion of input and output segments, the rotating axes were not saved
- TpaLangs application: the Save button was not enabled when editing only the message identifiers

Version 2.4.11 (March 28, 2022)

Corrections:
- Solved an error situation in case of opening a program, if the nesting functionality bit was not enabled on the hardware key

Version 2.4.10 (March 21, 2022)

New features:
- Modifications to SSIDE code functioning
- Modifications in *Nesting* functionality (added possibility to apply a format export to nesting results; K field internal use eliminated)
- Modifications in application of *Exclusions* (apply in: optimization, export, optimization preview)
- Addition in export to DXF format module (added selection of 2D export)

Corrections:
- Solved error situations in management of induced calls (use of j variables)

Version 2.4.9 (January 11, 2022)

New features:
- Modification in saving a program with name assignment (Windows reserved names filtered out. Ex.: "con", "null", …)
- Modifications in *Nesting* functionality (*TrueShape* nesting angular step changed to 5° from 10°; changes in label management)
- Modifications in *DXF* format import (import of entities with layer recognition: added option to enable profile connection)

Corrections:
- Solved problem in Machine/Drawing environment switch functionality (selecting the menu command required the **Professional** key)
- Solved error situations in *DXF* format import (reduction of *spline* curves)
- Solved minor errors
- Solved error situations in *Nesting* (sorting of pre-cut profiles of shapes, generation of scrap cut profiles, generation of labels with rotation applied)

Version 2.4.8 (July 29, 2021)

New features
- Modifications in the label creation, program, and nesting procedures (image file resolution improved)

Corrections:
- Solved issue in the "*SZSSHAPE: Profile shaped reduction*" (it did not adapt the reduction development with first external assigned technology)
- Solved minor errors

Version 2.4.7 (July 06, 2021)

New features:
- Modifications in *Nesting* functionality (generation of pre-cut profiles for *shapes*)
- Evaluation criterion of programmed technology modified (see paragraph: *Workings -> Working type -> The Technology*)
- Minor changes in configuration of application

Corrections:
- Error solved in applying snap on complex workings
- Solved error situation in *Nesting* (solutions of *TrueShape* placements, generation of scrap cut profiles, removal of temporary files)
- Solved error situation in *ISO* format import process (elimination of inserted rows in profile)

- Solved error situations in *TpaLANGS* application (menu status update, particular cases in assigning the reference language)
- Solved minor errors

Version 2.4.6 (May 14, 2021)

New features:
- Decreased the maximum number of TpaCAD instances (to 4)
- General face tools: eliminated the option to use the data in *clipboard*

Corrections:
- Solved error situation in using the tool *Change profile segment* (case of: arc exit tangent modification)
- Solved error situation in using text development workings (*Inclination angle* was only applied with request of text distribution on a geometrical element)
- Solved error situations in *Nesting* (applying bridges to shape profiles: the tool compensation option was applied twice; use of *shapes* generated by DXF files: with spline entities, the islands could be eliminated)
- Solved error situations in parametric programming solution (functions working on strings, with use of r variable specified by name)

Version 2.4.4 (February 11, 2021)

Corrections:
- Solved error situations in *Nesting* (cutting profile generation: the pre-cut technology was not used, in case of optimized profile; in cutting profiles with zig-zag entry/exit management, it wrote Z coordinate wrong)
- Solved error situations in the graphical representation of surfaces

Version 2.4.3 (January 25, 2021)

New features:
- Additions in *Nesting* function (QR code field in label; added fields in piece assignment)
- Modification in *Print the program label* functionality (possibility to assign QR code field)
- Addition in export to ISO module (setting added to convert to standard XY system)

Corrections:
- Solved error situation in export to *DXF* format module (cases of arc with subtended arc >180°)

Version 2.4.2 (December 02, 2020)

New features:
- Several modifications/additions

Corrections:
- Solved minor errors

Version 2.4.1 (September 15, 2020)

New features:
- Modifications /Additions in *Nesting* functionality (added options: *Apply pre-cut to all pieces*, *Move up the cut of small pieces*; added management of optimized pre-cut profiles)
- Additions in *Nesting* functionality (piece edging)
- Modifications in advanced tool *Create font from geometry* (graphic highlighting in profile selection)

Corrections:
- Solved error situation when using *Custom functions*
- Solved error situation when importing from *PZA* format (case of direct execution of PZA files with reading of outdated version of files)
- Solved error situations in graphical representation (piece and/or face trim solutions in case of: curved faces or surfaces, pivotable blade)

Version 2.4.0 (May 22, 2020)

New features:
- Modification in TpaCAD configuration (setting *Advanced user* removed, given as always active)
- Modification in TpaCAD configuration (setting *Optimize graphics* removed, given as always active)
- Addition in reading and importing ISO files (assignment of rotating axes, setup assignment)
- Modification in command *Print the program label* (possibility to save a file added)
- Added command in function *Nesting* (Save the report (.XML))
- Addition in parametric programming (function: *geo[plface;..]*)
- Addition in importation from *DXF* format (Transform the segmented polyline into a circle)

Corrections:
- Solved error situations in graphic representation (case of locked piece Rotation)
- Solved error situations in solution of 3D arc (direct assignment of the direction of rotation)

2.2 Versions from 2.3.1 (March 08, 2019) to 2.3.20 (December 02, 2020)

Version 2.3.20 (December 02, 2020)

Corrections:
- Aligned to version 2.4.1

Version 2.3.14 (March 31, 2020)

New features:
- Addition in parametric programming (function: *prrot*)

Corrections:
- Solved error situations in report of *Nesting* (wrong sheet size reported)
- Solved problem in the working LAMELLO (when applied on side face, the correspondence of the tool was not tested completely)
- Solved situation in Assignment of the Sequences (particular cases of visualization of the application point for the complex codes)

Version 2.3.13 (March 10, 2020)

Corrections:
- Solved error situations in solution of *Nesting* (cases of profile expanded workings)

Version 2.3.12 (February 27, 2020)

New features:
- Update of the certified signature for the executables to the algorithm with the *sha256* classification
- Modifications when importing from *PZA* format (identification of the stroke side of a panel, sorting in two programs in case of workings on bottom face)
- Modification in the management of format importing (import cases with creation of more files)

Corrections:
- Solved error situations in solution of the rotation direction for 3D arc
- Solved problem in configuration of the working *STZREPATT*
- Solved some minor errors

Version 2.3.11 (January 30, 2020)

Corrections:

- Solved error situations in the standard window showing the Technological table (miscellaneous in Tooling parameters)
- Solved error situations in importing from *PZA* format (assignment in configuration window; miscellaneous in choosing milling-cutter type tools)

Version 2.3.10 (January 14, 2020)

Corrections:
- Solved error situation in assigning the *Local technology* (selection of tool work face on all the side faces: it did not work)
- Solved error situations in graphic representation (the grid could apply incorrect assignments)

Version 2.3.9 (December 02, 2019)

New features:
- Modifications/additions in *Nesting* functionality (reading of "*.csv" list file: added fields, management of dynamic *header*)
- Modifications/additions in *Nesting* solution (modifications in accepting of non-closed nesting profiles)
- TpaLANGS Application: report window added
- Modifications in importing from *PZA* format (management of *Spindle rotation* added, modifications in working of grove on the panel edge)
- Working added in the "Programmed tools" group (SWEEP)
- Modifications/Additions in workings in the "Programmed tools" group

Corrections:
- Solved error situations in solution of parametric programming (the "*geo[alfa]*, geo[beta]" functions did not work in case of curved faces or surfaces)
- Fixed situations in nesting solution (generation of optimized cut profile: cases of segments that were cancelled)
- Fixed situations in *Nesting* functionality (shape reading from "*.dxf" file: particular importing situations, dimension assignment)
- Solved error situations in graphic representation (cases of: excessive zoom with graphics update, error in pattern rotation)

Version 2.3.8 (October 29, 2019)

New features:
- Added commands in functionality *Nesting* (List of recently opened projects, Commands added in solution browsing)
- Modifications/additions in *Nesting* solution (cutting technologies: speed, zig-zag entry)
- Added application functionality *TpaLangs* (non-numerical message identifiers)
- Modifications in exporter to *ISO* format (added setting page, technology update, profile translating options)
- Modifications in importing from *PZA* format (plant technology update, interpretation of the entity POLICURVA with tool compensation already applied)

Corrections:
- Solved error situations in the application *TpaLangs* (language cases filtered)

Version 2.3.7 (September 23, 2019)

Corrections:
- Solved error situations in graphical representation (cases of: modelling with fill)
- Solved error situation in window *Save File* (saving in disk unit)
- Solved error situations in basic workings (generic insertion, Cabineo)

Version 2.3.6 (August 01, 2019)

New features:
- Added commands in functionality *Nesting* (Save the unused pieces, Save the results upon completion)

- Addition in TpaCAD *Customization* (table of colours used in functionality *Nesting*)
- Additions in *Nesting customization* (*Covering between profiles* for true-shape solution, miscellaneous in *Label wizard*)
- Addition in functioning of *Nesting True Shape* (grid placement)
- Added functionality of *Print the program label*
- Added group workings *Insertions* (Lamello, Cabineo)

Corrections:
- Solved error situations in execution of *overall program transforms* (Rotate the piece, Mirror the piece, Overturn the piece)
- Solved error situations in nesting solution

Version 2.3.5 (May 27, 2019)

New features:
- Additions in functionality *Nesting True Shape* (piece grain)
- Modifications on complex workings of profile solutions (added parameter *Technological priority*)
- Added functionality in working *X Repeat*, *Y Repeat* (distribution of drillings on the sides)

Corrections:
- Solved error situation in the program's switch of *unit* of measurement (it stopped working)
- Solved situations in nesting solution
- Solved error situations in graphic representation (cases of: drawing of elements added in *small* lines)
- Solved error situation in *exporting to DXF format* (the export failed if a default extension for the converted files was not assigned)

Version 2.3.4 (April 09, 2019)

New features:
- Additions in *Inner Technology* (spindle rotation)
- Added functionality *Diameter compensation in conical tool compensation*

Corrections:
- Solved situations in nesting solution

Version 2.3.1 (March 08, 2019)

New features:
- Addition of the *Nesting True Shape* functionality
- Additions in the project of *Nesting of programs* (types of pieces, fields and piece list preview)
- Modifications in *Nesting of programs* configuration
- Review of *Nesting profile* tools
- Addition of *Panel grain* management
- Addition in parametric programming (variable argument: *cnf*; functions: *rempty*, *pngru*, *pnstool*, *pntool*)
- Addition of graphic representation of a shaped tool
- Added management of technological replacement for wear of milling tools
- Added workings in the "Mill Special workings" group (*Rectangle emptying (Greek pattern)*, *Door*)
- Modifications to *Door* workings (possibility to rotate the development along the horizontal axis)
- Addition in DXF import module (case of polyline similar to a circle)

2.3 Versions from 2.2.0 (December 04, 2017) to 2.2.15 (March 28, 2019)

Version 2.2.15 (March 28, 2019)

New features:
- Manuals and contextual help in foreign languages added

Corrections:

- Solved error situations in graphic representation (colour of ruler, cursor on surface)

Version 2.2.14 (February 27, 2019)

New features:
- Manuals and contextual helps in foreign languages added

Corrections:
- Solved error situations in graphical representation (cases of: uses of low-performance graphics cards)
- Solved error situation in *Save File* window (it did not display the initial file name)
- Solved error situation in modification of the *Global Technologies* (the changed enabling could not be acquired)

Version 2.2.13 (January 09, 2019)

Corrections:
- Solved error situation resulting from updating the OS to version 1809 (error reporting cases: "*Visual styles-related operation resulted in an error because visual types are currently disabled in the client area*")
- Solved error situation in *General tool* request on large programs (cases of: slowdown in command start)
- Solved error situation in format *import* (technological application for Setup and/or Punctual working)
- Solved error situations while importing from *PZA* format (cases of: groove interpreted as a milling work)

Version 2.2.12 (December 06, 2018)

New features:
- Added messages in Turkish
- TpaLANGS application program: addition of manuals in French, Spanish and Russian

Corrections:
- Solved error situation in selection of lines in ASCII Text
- Solved error situation in graphic display of workings applied to Surface

Version 2.2.11 (November 16, 2018)

Corrections:
- Solved error situations in the functionality of *Nesting of panels* (reset of the execution sequences)
- Solved error situations while importing from *PZA* format (cases of: groove interpreted as a milling work)
- Solved error situation in the box of *Save File* (filter of invalid characters)
- Solved error situation in applying *Mirror Tool* to workings of the Mill special workings group (rectangle, polygon, …)

Version 2.2.10 (October 31, 2018)

Corrections:
- Fixed error situations in *Nesting of panels* functionality (assignment of r variables, application of an edge inside the placements, signalling pathname of length > 100 characters, elimination of workings based on geometric placements)
- Solved error situations when importing from PZA format (cases of: groove of sawing work performed with a profile, sizing of segments in entry/exit profile while shaping a piece)
- Improved graphic printing functionality (the image size is reduced to the useful area of representation)

Version 2.2.9 (September 28, 2018)

Corrections:
- Solved error situation while printing labels in *Nesting of panels* functionality (case of: printing with stretched image)

Version 2.2.8 (September 19, 2018)

Corrections:
- Solved error situation in the parametric programming help window (some entries in the "pr\.." list were incorrect)
- Solved situation of incorrect graphic representation (cases of: workings in closed surfaces)
- Solved error situation in DXF format import (case of: incorrect import of circle as a point)

Version 2.2.7 (June 27, 2018)

New features:
- Addition of ruler management in graphic representation with Box view
- Extension of SSIDE code operation (development also in subroutine typology)
- Addition in parametric programming (formatting in string variable)
- Addition of management of the assignment with largely Unicode characters in the *Nesting of panels* (list of materials, assignment in the nesting and label project)

Corrections:
- Solved error situation determining the deactivation of the custom sections (entered with the 2.2.3 version)
- Solved error situation of *Nesting of panels* (case of generation of equal panels)
- Solved error situation when reading custom messages, if they are in a split format (one file per language)

Version 2.2.3 (March 22, 2018)

New features:
- Addition of ruler management in 2D graphic representation
- Additional items in the drawing menu (Rectangle with centre and corner, inclined Rectangle, Star)
- Addition of working in the group of "Polygons" (star)
- Addition of working in the group of "Mill special workings" (star)
- Additional export module to DXF format (module to be configured: *TpaSpa.DxfCad.v2.dll*)

Corrections:
- Solved error situation in graphic display of vertical/horizontal overall dimensions (slowness of the view due to incorrect fragmentation criteria and linearisation of curved elements)
- Solved error situation in graphic display of the overall dimensions (cases of: visualization of spurious triangulations)
- Solved error situation when calling export modules (after an export process, the configuration of the module itself could not be opened)

Version 2.2.2 (January 30, 2018)

New features:
- Additional commands of *Overall transforms of the piece* (menu *Apply*: Rotate, Mirror, Overturn)
- Additional commands of *General tools* (menu *Tools*: Centre and align to the face)
- Additional command in *Nesting of programs* (workings outside the placements can be excluded)
- Additions in parametric programming (functions: "sign[nn]", "geo[angm;..]")
- Additional working in the group of "Mill special workings" (trigonometric function)
- Additional items in the group of "Mill special workings" (assignment of external technology)
- Changes to application of the Customization setting *Show always the start point in profile segments*

Corrections:
- Solved error situation while importing DXF format (assignment of piece dimensions for the single ellipse)
- Solved error situation in ISO import process (assignment of piece dimensions)
- Solved error situation in Assignment of Sequences (the selection of the graphic selection was not active when the non-sequencing workings were displayed)
- Solved error in graphic display of the vertical overall dimensions (case of trim to the face where the tool is oriented parallel to the face)
- Solved error situations in the graphic rotation of the piece
- Solved error in the application of TpaLangs.exe (cases of system language recognition)

Version 2.2.1 (December 19, 2017)

Corrections:
- Solved error situation in ISO import process (import process not performed)

Version 2.2.0 (December 04, 2017)

New features:
- Reorganization of the distribution of localized files (application messages, context and document help)
- Reorganization of basic Database messages for workings in DLL files
- Additional functionality in *Report window* (shows the codes of the suite in the key)
- Additional command *New from template* (program creation from model)
- Addition of new command menu of the entries for *Last open programs*
- Additional feature in *Nesting of programs* (the user may not generate cut profiles and new fields in assigning the labels)
- Additional feature in the configuration: setting change level in (Piece settings, Custom sections)
- Additional feature in *Essential* (advanced user, profile construction tool)
- Additional functionality of *Modelling* from profiles programmed in the face
- Addition in Customization (Enable in graph: *Deactivate the additional graphic elements for a large program*)
- Additional feature in the menu View on (additional menu to command *Overall view in 3D graphs*)
- Additional feature in *General Tool* (graphic interaction also reproduces the workings concerning the transform)
- Additional feature in *Rotate* general tool (rotation to the minimum overall dimension)
- Additional sections in the configuration of the import from DXF (*Subroutines* & *Layers, Subroutines* & *Blocks*)
- Additional workings in the group of "Mill special workings" (different door typologies)
- Additional features in the working of *Rectangle* application (option of fillet or chamfer solution)
- TpaLangs.exe application program to manage the custom message files.

Corrections:
- Solved problems in *Nesting profile* advanced tool (identification of internal profile in group of paths)

2.4 Versions from 2.0.0 (April 08, 2016) to 2.1.18 (September 26, 2018)

Version 2.1.18 (September 26, 2018)

New features:
- Modifications of abnormal situations originated by Windows 10 OS updates
- Added specification of import from PZA format (workpiece flip, assorted additions)

Corrections:
- Solved error situations in *Nesting of programs* function (cutting profile generation with multiple strokes; cases of error generated by complex code explosion)
- Solved error situations in importing from DXF format (importing ellipse sections)
- Solved some minor errors

Version 2.1.11 (November 29, 2017)

New features:
- Modifications in the way program types are associated with the TpaCAD application (case of updated versions of OS Windows 10)

Corrections:
- Solved situations of incorrect navigation in the technology table (line and/or machine with multiple groups)
- Solved error situation in importing the DXF format (opening a file required an exclusive access)

Version 2.1.10 (October 27, 2017)

New features:

- Additional specifications in importing from the DXF format (piece rotation with xy axis exchange, addition of significant "&.." prefixes).
- Additional specifications in importing from the ISO format (settings: *Dimension G code*, *Quadrant of the machine*)

Corrections:
- Solved error situations in importing from the DXF format (case of assignment of Z position for polyline associated with Macros&Layers)

Version 2.1.9 (October 13, 2017)

New features:
- Addition of specification of the import from DXF format (general setting: *Import empty panel*)
- Addition of specification of the import from ISO format (linearization of *little* arcs)

Corrections:
- Solved error situations in the general application of the *Convert unit of measurement of the piece*
- Solved error situations in applying the graphic activation of *Adjust the overall dimension of the profiles to the length of the segments*
- Solved *Ellipse for 3 points* drawing command issue (wrong snap application on the X/Y axes)
- Solved error situation in solution of *Nesting of programs* (cases of sheets duplications, solution of several solution groups, technological assignment of cut profiles)
- Solved error situation in *Essential* functionality (use of *Global technologies*)
- Solved problem of *ISO file format recognition* (case of: "TCN" file in Unicode format)

Version 2.1.8 (September 18, 2017)

New features:
- Addition of specifications of the import from DXF (general setting: *Do not reposition the entities*)

Corrections:
- Solved error situations while assigning the solution sheets of *Nesting of programs* (cases of choosing the best solution, in the event of *severe warning* that now no longer locks; evaluations of overall dimensions of the applied workings)
- Solved problem in the working STOOL: STMULTI (turned the initial settings into a solution of following profiles)
- Solved problems in the advanced tool *Nesting of profiles* (use in piece-face)
- Solved problem in the parametric programming solution (mixed use of the characters of decimal separator)

Version 2.1.7 (July 07, 2017)

New features:
- Additional working in the group of "Setup" (SETUP ISO 5x)
- Additional working in the group of "Subroutines" (SUBROUTINE ISO)
- Additional working in the group of "Tools" (Z feed + Bridges)
- Additional features in the application workings of Pocket (assignment of the *centre* position)

Corrections:
- Solved situations of error assigning solution tabs of *Nesting of programs* (case of errors resulting from *rotation* of complex workings and *Tool compensation*)
- Solved problem in the *Repetitions with circular development* general tool (it always applied the rotation of the repetitions)
- Solved problem in the *Cut single elements of a profile* tool (when not performing)
- Solved problem in the *Duplicate profile* construction tool (if an exception occurs with request for working duplication)

Version 2.1.6 (June 23, 2017)

Corrections:
- Solved situations of failure in restoring the application layout
- Solved wrong association of tool typology in *Technology table* (cases of sawing work typologies)

Version 2.1.5 (June 15, 2017)

New features:
- Additions in the piece-face management (block, import)
- Additions in the *Nesting of programs* functionality (label management)

Corrections:
- Solved error situations in interactive matching with the workings (piece-face)
- Solved error situations in label layout assignment in *Nesting of programs* functionality (button activations, …)
- Solved error situation in importing from PZA format (development of multiple milling passes and /or sawing works; Spindle and Tool maximum value increased from 1000 to 10000)

Version 2.1.3 (April 27, 2017)

Corrections:
- Solved situations in opening secondary windows (case of opening on secondary screen)
- Solved error situation in opening the *Open File* window (case of failed solution with assigned path on a string of more than 180 characters)
- Solved issue in *STOOL*: *Working Z feed* (cases of setting a number of pitches and of the final Z position: possible generation of a spurious error)
- Solved situations in configuration service of the DXF format importer (assignment in the window)

Version 2.1.2 (April 04, 2017)

Corrections:
- Solved issue in *Nesting* functionality (prototype management)

Version 2.1.1 (March 20, 2017)

New features:
- Additions in parametric programming (functions: "geo[beta;..]", "geo[alfa;..]")
- Additional items of *Nesting profile*
- Addition in assignment of oriented geometries (4 or 5 axis interpolation)
- Additions in *Rotate* general tool
- Addition in the graphic menu (*Modify* command)
- Addition in "Apply bridges to profile" tool (option of bridges/interruptions)
- Updates of the documentation and translation of messages

Corrections:
- Solved problem in opening a program save window (local disk filters, network folders, cloud)
- Solved issue in *Rotate* general tool (cases of a rotation that does not correspond to the graphical interaction)
- Solved issue in *Vertical mirror* general tool (cases of arcs with assigned Y coordinate of the centre)
- Solved problem in *Take off each profile segment* tool (error in inserting and graphic representation)
- Solved problems in *Move setup in closed profile* tool (profile assigned with setup + arc A01)
- Solved problems in *Tool compensation* tool (cases of change correction side applied to a circle)
- Solved problem in *Create surface from geometry* advanced tool (circle)
- Solved problems in *Nesting* functionality (step-by-step management)
- Solved graph problems (light effects in three-dimensional visualization)

Version 2.1.0 (December 02, 2016)

New features:
- Addition of the functionality of Nesting of programs
- Additional general instrument for *Repetitions on a profile*
- Global technologies management (set in *Customize TpaCAD*)
- Management in Import external format (set in *Customize TpaCAD*, *Program overall activations, Validate profiles*)
- Addition of draw functionality (length of the stroke for a Line inserted in tangent continuity)

- Addition in Customization (Enable in graph: Adjust the overall dimension of the profile to the length of the segments)
- Addition in assignment of oriented geometries (entry and exit segments: possibility to exclude the *Tangent tracking*)
- Addition in assignment of oriented geometries (4 or 5 axis interpolation)
- Additional option in the configuration of the import from DXF (a polyline can be split into single segments)
- Additional specification from DXF (any path can be associated to the *Workings & Layers*, *Macros & Layers* functionalities)
- Additional unit in the configuration of the import from DXF (*Logical Workings & Blocks*)
- Addition in the DEMO functionality (user-level accessible parts of the configuration can be changed)

Corrections:
- Solved slowdown problem in the graph of the program
- Solved problem while executing the interactive mode of *Snap on face* (cases of TpaCAD stops)
- Solved problem in the *Arc with the radius* drawing command (impossible to insert the radius directly in the menu)
- Solved problem in the STZLINE programmable tool (each profile segment was assigned as a single profile)

Version 1.4.14 (October 06, 2016)

New features:
- Updates of the documentation and translation of messages

Corrections:
- Solved problem in the *Profile inversion* (case of: the profile assigned with one only segment could lose the modification in the depth)
- Solved problem with the warning situations in the field programming in the custom section
- Solved problem in the tool *Apply bridges to profile* (case of profile whose first segments are little, could no longer apply a bridge)
- Solved problems in the commands of the **Find/Replace** commands (case of replacing failed for programs with many lines)
- Solved problem in emptying of areas (case of very fragmented profiles and generation of spurious closed areas)
- Solved problem concerning the operation with some languages (for example, Turkish) (case of failure in starting the Custom Optimizer)
- Solved problem in executing format imports (case of importing non-ANSI characters)

Version 2.0.0 (April 08, 2016)

New features:
- Addition in configuration: "Surfaces" (Piece settings, Piece geometry)
- Addition in configuration: "Apply transforms to the oriented geometry" (Piece settings, General)
- Addition in configuration: "Sharp corner cut" (Piece settings, General)
- Addition in configuration: "Plant technological component (Environment, Components)
- Addition in configuration: "Turn the speeds into [m/min] or [inch/sec]" (Open and save, Assign in Piece matrix and Export)
- Functionality of: "Tooling technology"
- Addition in "Global variables" functionality (menu commands)
- Addition in the File menu (Optimize an archive of programs)
- Addition in the drawing menu (Arc for the radius, Circle through 2 points, Circle through 3 points, Ellipse through 3 points, Helix, Spiral)
- Additional item in the construction tool "Apply bridges to profile" (distance between bridges)
- Additional item in the "Profile reduction" tool (linear minimization and more)
- Addition of profile tool "Take off each profile segment"
- Addition of construction tool "Divide on intersection points"
- Modification of the local menu for the graphic control (Selection group, commands: Cut, Copy, Paste)
- Addition in the command palette of the working data-entry (entry: "Apply to workings of the face (if automatic)")
- Addition of working in the group "Single arcs" (A27: (tgin, R, A, CW))
- Addition of workings in the group "Polygons" (A48: Helix; A49: Spiral)
- Addition of working in the group "Mill special workings" (HELIX: Helix; TWIST: Spiral)

- Additional workings in the group "Mill special workings": milling to progressive reduction of rectangle, polygon, oval, ellipse, circle, pocket, generic profile
- Additional workings in the group "Tool": milling to progressive and shaped reduction of a generic path
- Additional workings in the group of "Tools": Validate profiles
- Additions in parametric programming (functions: "geo[sub;..]", "geo[param;..]", "geo[lparam;..]")
- Additional print functionality in TpaWorks application
- Additional option in the configuration of the import from DXF (TpaSpa.DxfCad.v2.dll): possible application of a multiplication factor to the assignments of parameters extrapolated from Layer or Blocks

2.5 Versions from 1.4.0 (April 10, 2015) to 1.4.9 (March 30, 2016)

Version 1.4.9 (March 30, 2016)

New features:
- Additional feature in the import of ISO format (TpaSpa.IsoToTpa.v2.dll): the characters at the beginning to recognize a valid format have been extended
- Modification in the application of the default technology in import from an external format (DXF, ISO, …): the assignment does not change the value of the *Construct* property (field B)

Corrections:
- Solved problems concerning the graphic representation of oriented profiles
- Solved problems concerning the graphic representation of conical sections (excessive fragmentation in the representation of the vertical overall dimensions)
- Solved problems concerning the graphic application of the trim cut
- Solved problem concerning the execution of the command *Group together* (situations of incorrect selections occurred)
- Solved problem concerning the execution of the command *Paste*, selected from the menu for graphic control (ASCII text was not updated)
- Solved problem in the window of program opening (in case of import from the external format, the side related to the customization was no longer visible)

Version 1.4.8 (January 26, 2016)

New features:
- Additional option in the configuration of the import from DXF (TpaSpa.DxfCad.v2.dll): the direction of rotation of the closed profiles can be forced

Corrections:
- Solved display problems when operating on an operating system with an oriental interface language (see: Menu, font quality)
- Solved problems in developing oriented texts (dimension and relative placement of characters)

Version 1.4.7 (December 14, 2015)

Corrections:
- Solved problem in Extend profile tool (case of extension of the segment to the horizontal line, it assigned the x coordinate)
- Solved problem in the Assign technology profile tool (the Tangent tracking parameter was not maintained)
- Solved problem in the File window -> Open (in the navigation through folders)
- Solved problem in Configuration (case of edit of settings in Custom section)
- Solved problems in parametric programming (mathematical operator "?" and function "geo[sub;..]")
- Solved problem of start with the language set in the system (case of Spanish)

Version 1.4.6 (October 19, 2015)

New features:
- Additional functionalities to the working code "STOOL": Fragment and linearize (the user can fragment the linear segments too and also not linearize the arcs)

Corrections:
- Solved problems concerning the graphic representation of oriented profiles

Version 1.4.4 (October 15, 2015)

New features:
- Addition in the DEMO functionality (command to choose the working level: Essential, Base, Professional)
- Addition of selection of the colour to be applied to the application program Style
- Additional features in the Scale and Pull tools (interactive acquisition of the scale factor)
- Additional feature in the General tools (automatic interactive acquisition on the vertices of the overall rectangle)
- Additional features in the View on menu (Info group: closed profile, multiple setups, entry/exit segments)
- Addition in Customization (Add selections)
- Additions in Customization (Retrieve Bookmarks)
- Additional feature in Customization (length of the tool for the setup graphic representation of the null overall dimension)
- Modification of the local menu of graphic control (additional commands of profiles, creation of Zoom and Navigate groups)
- Additional command "Group together" command (Edit menu, Modify group): thanks to this command the selections in the list become consecutive
- "If.. ElseIf.. Else.. EndIf" command (Edit menu, group of Blocks)
- Additional feature of Logical block (Edit menu, Blocks group): application of the block to a group of selections
- Additional feature in curved face assignment (selection of start/closure in tangent)

Corrections:
- Solved problems in solution of parametric programming (functions: geo[pxf/pyf/pzf;..], geo[lparam;..])
- Solved problem whilst managing the Sequences: the selection in the failure of graphic area
- Solved problem on the Undo command performed after having inserted Logical block
- Solved problem in performing commands of Replacement invoked from General View (replacement was applied in piece-face only)

Version 1.4.2 (July 13, 2015)

New features:
- Additional feature in the Development Text tool (Inclination angle)
- Additional feature in the Text development codes (Inclination angle)
- Additional feature in the Development of the Cardinal curves (the development calculates the tangent lines on the initial/end point, in order to keep the same development, when the initial/end point changes)
- Addition of DXF file import module (TpaSpa.DxfCad.v2.dll)

Corrections:
- Solved problem in the text development: revised the conversion limits of corner in arc (case of profile with too many corners)
- Solved problem in the Geometry Configuration of the piece (selection in the face origin list)
- Solved problem of the profile in the window of File -> Open: when the multiple typology was selected (case of format importer) the selection of a file was not possible.
- Solved problem of the application of the graphic command *Extended Zoom* (stopped working)

Version 1.4.1 (May 05, 2015)

New features:
- Addition in configuration: "Epsilon used in logical comparisons"
- Additional features in customization: graphic customization for geometric profiles, for emptying and construct profiles
- Solved the management of the Sequences: the movement of lines in Drag&Drop mode applies the selection of insertion above/below
- Additional feature in the Fictive face window settings: additional information calculated for the face

Corrections:

- Solved the problem in managing the Sequences: moving more lines in drag&drop mode inverted the insertion order
- Solved problem in the Profile reduction (tool and STOOL working): cases of arc reduction with unchanged coordinate (it could choose the wrong coordinate for the centre programming)
- Solved problem in applying the commands Create fictive face from geometry, Create modelling from geometry (case of: circle)
- Solved problems in the graphic representation: trim application of the overall dimensions

Version 1.4.0 (April 10, 2015)

New features:
- Addition in configuration: activation to manage the Unicode format for TCN programs
- Addition in configuration: activation to manage the XML files to save the environment configuration
- Addition in configuration: view of the paths and remarkable files
- Addition in configuration: activation of "Complete the workings read by an external program"
- Addition in configuration: assignment of the dimension for the fields of the Custom sections
- Addition in configuration: modification of the profile for the Sharp corner cut
- Extension of the "Stand-Alone" functionality
- Extension of the language managed in the basic operation
- Additions in the management of the prototype file (cases of differentiation according to the piece type; management of type and access levels)
- Addition of management of recognition of primary or secondary instance, in the case of TpaCAD multiple instances
- Addition of management of automatic recovery in reading TCN files
- Addition in the group of Edit commands: Redo
- Addition in the "Validate Profiles" global Tool
- Addition of the option "Reduce data matching search" in the global Tool "Profile connection"
- Addition in the groups of Find and Replace the management of correspondence also for the Name of the working
- Addition in the Translate tool (placement of the overall rectangle)
- Additions in the "Develop the text" tool (spacing and distribution mode; distribution on the conic section; RightToLeft)
- Additions in the codes of Text development (spacing and distribution mode; distribution on geometric element; RightToLeft; technology retrieval from external working)
- Addition in the tool "Create font from geometry" of the assignment of the position for the next positioning
- Additional features in the Entry/Exit tool (typology: Approach)
- Additional feature in the Enter/Exit profile of Setup (typologies: Approach, Removal, Coverage)
- Additional functionality to the STOOL workings (possible retrieval of the technology from the external working)
- Additional functionality to the STOOL workings (possible retrieval of the workings from the previous call level)
- Additional functionalities to the working code "STOOL: code Z feeds" (selection of the development axes)
- Additional workings in the group of "STOOL": "STMULTI: Repeat profile", "STFILLET: Fillet profile", "STCHAMFER: Chamfer profile".
- Additional functionality "NOP working: Null operation"
- Additional functionalities in the Setup workings for special geometries (curved faces or areas)
- Addition in "A32 working: Double arch "(intermediate connection)
- Additional functionalities in the parametric programming (functions: geo[param;..], geo[lparam;..])
- Additions in the interactive placement (snap on the face vertices)
- Additions in View of the profiles, activation and trim option of the overall dimensions
- Addition in the View on menu of the option "Show all fictive faces"
- Addition in the status bar of the application program (*Apply to a copy of the workings*)
- Addition in the settings of Customization, it is possible to confirm the insertion of a working also when an error occurs
- Additional feature in the use of working properties ("Delete at the end of the development"; propagation of a null value)

Corrections:
- Blocked drag interpretation program in TpaCAD (opening with drag-drop) with a command in execution

2.6 Versions from 1.3.0 (February 10, 2014) to 1.3.11 (March 31, 2015)

Version 1.3.11 (March 31, 2015)

Corrections:
- Solved problem with the TpaCAD launchpad (TpaCAD remains in the background)
- Solved problem in technological assignment applied while importing a format (no properties assigned)
- Solved problem in configuration (Custom section setting, with displacement in a list of the items of a section)
- Solved problem in the resolution of induced calls (case of no solution)
- Solved problem in resolution of curved faces (case of curving of the face assigned on the Y axis)

Version 1.3.10 (February 27, 2015)

Corrections:
- Solved problem related to the TpaCAD start-up window (in case of memory errors in case of multiple and close starts)
- Solved problem when applying the Repetition Tool (case of number of repetitions greater than 32767)
- Solved problem while reading a program (case of string working parameters, with a setting containing "WC", "WB" sub-strings: they could lead to interpret direct assignment of some properties, such as comment, construct)
- Solved problem in the graphic representation of arcs programmed in curved faces (radius of curvature of the face less than the radius of the arc. The arc could be visualized with an excessive linearization)

Version 1.3.9 (January 27, 2015)

New features:
- Modifications in the snap application on the vertices of the face (the mouse position is attracted on the sides of the face, with the addition of the possibility to lock one of the two coordinated axes)
- Addition of the option in configuration of import from DXF (all the blocks can be excluded)

Corrections:
- Modifications of the application for the tool compensation (profile reduction in conic section compensation)
- Solved problem of wrong determination of the direction of rotation in 3D arc
- Solved problems when importing from DXF format (maximum number of elements of a profile, exclusion of non-assigned blocks)

Version 1.3.8 (December 04, 2014)

New features:
- Modification in the graphic representation of invalid working (case of point working or setup with invalid operation code): it does not apply the technological information read by the working

Corrections:
- Solved problems in the commands of the **Find/Replace** group (case of replacement of an invalid code)
- Solved problems for the management of the **Apply Z feed** tool (case of #xy arc - wrong translation of the centre)

Version 1.3.7 (November 11, 2014)

Corrections:
- Solved problem in case of importing a program with a technology application (case of wrong processing for tests containing complex codes)
- Solved activation problems in the Configuration window (also at the access level Constructor the maintenance level was applied)
- Solved problems in the *Develop Texts* window (*Invert segment* selection was not managed)
- Solved problems applying the **General Tools** (translate, rotate, mirror) (case of: profile without programming the starting point)

- Solved problems in the solution of Starting segment/End of the profile of 3D Arc
- Solved problem of anomalous slowdown in graph of the oriented profile

Version 1.3.6 (October 09, 2014)

Corrections:
- Solved problem of display of the working applied in curved face

Version 1.3.5 (September 18, 2014)

New features:
- Addition of management of a local constructor account
- Management of management of automatic updates (that is: to modify the database of the workings)
- Modification in the data-entry box management of the current working (in case of automatic confirmation, the control does not go to the next working)
- Addition of interactive functionalities to get to the position (management of the direction buttons for a discrete movement of the mouse)
- Addition in configuration: various items in the feature of technological controls
- Addition of the functionality "Off-line"

Corrections:
- Solved some problems in the Configuration window

Version 1.3.4 (August 27, 2014)

Corrections:
- Solved problem in the management of the order window, that can be used from the section of the custom settings

Version 1.3.3 (July 31, 2014)

New features:
- Addition in enabling processes to the Assistance/Maintenance level (modifications of the Menu and Working Palettes composition in Configuration are now possible)
- Addition in configuration: various items in the feature of technological controls
- Addition in configuration: access level to the Global Variables
- Addition in customization: activation of auxiliary columns of ASCII text
- Addition in customization: activation of the automatic confirmation in working acquisition
- Addition in customization: activation of the auto-recover of a program
- Additional items in interactive disposition of the main work window
- Modifications concerning the recognition and the control of the languages with RTL orientation (Right-to-left)
- Addition in management of the Sequences: additional feature of drag&drop, colour of column

Corrections:
- Addition of recognition of a folder or a file assigned with a link (example: shortcut on the desktop)

Version 1.3.2 (May 15, 2014)

New features:
- Addition of interaction in status bar (selection of a current line)
- Modifications in *3D view* of the vertical setups
- Modifications in view of profiles in Tool Compensation (the setup is brought now on the right profile)
- Additional item in the group of commands for the *Place at line*
- Additional item in the group of *Measures*
- Additional item in the group of *Customize the views* (view of the overall dimension of the profiles)
- Additional item in the Button bar in the *Working assignment area* (command: "Reset working")
- Additional features in *List of Preferred Workings* (shortcut)
- Additional features of *Area of ASCII text* (modification of the properties from the cell of column header, colour column)

Corrections:
- Wrong correspondence of commands in the menu of graph (Zoom-In and Zoom-out commands were reversed)

Version 1.3.1 (April 2, 2014)

New features:
- Addition in configuration: selection for the interpretation of the reports in the management of the program *Custom sections*
- Modification in the management of diagnostic signals, in case of minimized viewing area

Corrections:
- Solved problems concerning the use of a non-standard DPI
- Solved the problem while launching the configuration of the exporter to DXF format

Version 1.3.0 (February 10, 2014)

New features:
- Addition of functionality "Essential" (recognition from USB hardware key)
- Addition of double configuration control ("Draw environment", "Machine environment")
- Addition of functionality "Stand-Alone"
- Addition in configuration: selection for the application of "Programmed induced calls"
- Addition in configuration: selection for "Automatic assignment of variables r"
- Addition in configuration: selection for parametric programming of "V" working property
- Addition in configuration: selection to direct modification in ASCII text of working "B" property
- Addition in configuration: activation to manage the "Global Variables" and the command to assign the related list
- Addition in configuration: various items in assignment of piece-face
- Addition in configuration: enabling of "Curved Faces"
- Additional features in configuration of the Piece matrix assignment and of the modules of Format conversion to writing (fragmentation of the arcs assigned in curved faces)
- Additional features in configuration of the reading format conversion modules (Import as subroutine)
- Addition of the possibility to disable all the real faces
- Addition in configuration: control of the working palette in one group
- Additional features in the control of the prototype file (cases of differentiation according to the piece type; management of type and access levels)
- Modifications in default technological working customization: now, it can be assigned also for the unmanaged real faces
- Additional commands of Transforms in the window to assign the Variable Geometry (fictive or automatic face)
- Additional geometric typologies in the Modelling section (arc and line with solution of tangent continuity; fillet on rectangle corners; oval)
- Extension in automatic propagation of "r" variables (performing also the propagation of the variables used with symbolic name)
- Extension applying "Induced programmed calls" (applies in all faces, recognizes the setting of enabled or excluded faces, programs the application point)
- Additional feature in Find/Replace commands (button to "assign" the current working automatically; command to create a list of the correspondence found)
- Additional feature in command "Create font from geometry" (management of character multiple assignment)
- Additional features in the window of General Tools (management of the help menus of the parametric programming)
- Addition of Info group in the View menu
- Additional features in the management of the data entry window of the current working (restore of the selection in the command palette; storage of the opening state of the nodes)
- Modifications in the "Client working" window (you can assign new buttons in the palette of the workings)
- Additional features in the management of the lock of the program sections (possibility to hide the locked sections)
- Parametric programming, additional capabilities
- Modifications in the database of the workings: assigned the dimension of the parameters for the complex workings
- Modifications in the working database (SSIDE [2021] working]: additional parameters to assign the application point)

- Modifications in the working database (NSIDE [2020] working]: additional parameters to assign a curved face)
- Modifications in the database of the workings (POSITION [1112]: additional parameters to assign the height of the font and the decimal places in characters; interactive mode of measures; writing font modified
- Modifications in the machining database (WARNING [2019] machining: added the parameter to switch to error status if in execution mode)
- Addition in managing the option of the "Global variables" in the help menu of the parametric programming
- Addition in configuration of the import from DXF (import unit of measurement)

Corrections:
- Solved solution of planes to be modelled (issues of coinciding planes)
- Solved some shortcut management cases for edit commands (cut, copy, paste)
- Solved working customization problem in status bar
- Solved some minor problems or graphic representation (profile overall dimension)
- Solved some problems while executing the "machine change" (Restoring layout and style)
- Solved some problems in the TpaWorks application program (while executing the "Check the working"; create/delete a node; management of Machine change)
- Solved problem in the TpaWorks application program (while executing the "Check the working" command)
- Solved minor problems

2.7 Versions from 1.1.0 (December 06, 2012) to 1.2.4 (October 10, 2013)

Version 1.2.4 (October 10, 2013)

Corrections:
- Solved problem in solution of STZLINE working code (Linearize in Z): a distinct profile was resolved for each segment of the original profile

Version 1.2.3 (September 10, 2013)

New features:
- Modifications of the export module to EdiCad format (modifications of the created text format, so that it can be read also by very old EdiCad versions)
- Addition of interpretation of the tool compensation in addition to the export module to EdiCad format

Corrections:
- Solved problem in program import process in EdiCad format (it did not execute the assignments of operating code and/or of the parameters for the first working of the working database, that normally is: X Sawing)

Version 1.2.2 (July 22, 2013)

Corrections:
- Wrong tool graphic representation in oriented setup programming

Version 1.2.1 (July 08, 2013)

New features:
- Additional command "Find and Select" (Edit menu, Modify group) performing selecting working selections
- Additional commands in local menu of ASCII text (selecting a part of the profile)
- Additional "Zoom In/ZoomOut" command
- Additional customizations related to the graphic interaction
- Best graphic selection of the current working in 3D view
- Addition in configuration of setting to enable the "Stand-alone" functionality
- Addition in configuration of modules to convert the format (in reading and writing)
- Addition in configuration of setting to enable the "Technology" window
- Best management of the error window (displaying up to a maximum number of errors)
- Modified criteria associating the images to the 'O' property
- Best interactive management in Profile Tools (composition of Snap menu, graphic construction)
- Best management of "Preferred workings" (commands of)

- Best recognition of the current language (best flexibility in language encoding)
- Additional features of program loading in Edicad/TpaEdi32 format (recovery of the profile custom codes)
- Modification of the initialization criteria for the program Optimization (enabling conversions and tooling)
- Additional installation of an Optimization module, compatible with VB6 environment
- Additional features in loading technology specifications (tool-holder and tool-change enquiry)
- Updated help for the workings in language (in English)

Corrections:
- Solved some problems of induced calls management (situation of locked application or incomplete graphic representation
- Solved problem in "Symmetry around a horizontal axis" (an acquired position was not assigned in the window)
- Solved problem in the tool application "Connection between profiles" (case of profiles with depth variation)
- Solved some problems in the Configuration window
- Solved problem in program Opening (exception thrown by some "cloud" resources in use)
- Solved problem in a standard window for technology representation (application lock if no group is enabled in the configuration)
- Solved some problems in Program optimization (assigning the current tooling)
- Solved some problems in the TpaWorks application ("Auxiliary" attribute window of working parameter: management of assignment tables for the "Active Status" of the parameter)
- Solved problem in assignment window of the current working (case of condition of the parameter active status, if compared to equal/different value)
- Solved minor problems
- Solved installation problem of the application ("silent" execution; enquiries were managed, anyway)

Version 1.2.0 (April 23, 2013)

New features:
- Additional feature in configuration of "Execution mode" section (configuring the active execution modes)
- Additional feature in configuration: selection for rotary axes in "Oriented setup" (configuring the rotation plane for the B axis)
- Best match between configuration and Hardware key (the configuration stored is aligned to the operating feature)
- Additional feature in configuration in case of functionality with Enterprise key (selection item "Protected configuration" in the Environment menu, General Setting group)
- Additional management of "Advanced Configuration"
- Additional management of piece Modelling by extrusion (it requires a specific hardware activation)
- Additional "Resolve" command (Edit menu, Modify group) resolving parametric programming
- Parametric programming, some additional functionalities (geo[sub;..], geo[param;..])
- Best recover of TpaCAD start-up layout (synchronized on the change of the main version second digit)
- Best management in the Program opening window
- Best management of the navigation commands of the ASCII text logical branches
- Best view of the working coordinates (for profile segments only)
- Best management of the programmable custom errors by means of the added distinction between base errors and customization error)
- Best diagnostic reports in Custom optimization call (using a more specific message)
- Best management in the window of the customizable message changes
- Best interface while assigning the COLOUR typology field of the custom section
- Best command management of "Create fictive face from programmed geometry" (selection in piece-face of)
- Best management in the help window of parametric programming
- Best application of the default technologies for the Setup workings (difference according to the sub-typology)
- Additional features in standard technology loading of Albatros environment
- Best management in standard window of technology representation (logical correspondence and tool image)
- Additional feature in the components of Conversion from/to ISO format (customization of the G-ISO code of punctual working)
- Modifications in the working database (BLADE SETUP [95]: it assigns sub-typology 2)
- Several modifications in base macro-program texts
- Updated helps for the workings in language (Italian)
- Additional manual of the TpaWorks application in language (English)
- Additional specific manual for the modelling (Italian and English)

Corrections:

- Solved problem in TpaCAD start-up with double click on piece-file from Resource Management (path including spaces)
- Solved problem in assignment of Fictive faces (exception situation while executing the "Delete all" command)
- Solved minor problems in the window for the assignment of the current working
- Solved problem in update of the "<j> Variables" window
- Solved problem in Program Optimization (Error recording the custom section in piece matrix)
- Solved minor problems of graphic representation (representation of the standard grid, situation of covering colours)
- Solved configuration problem of the import from DXF (in the page "Macros and levels" the prefixes of the parameters got lost)
- Solved minor problems

Version 1.1.4 (March 05, 2013)

New features:
- Best management of the error window (the "Go to line" button is active also in case of error from general program section)
- Best assignment of the active view while changing "Active View" (active tab of program general information)
- Best loading procedures of an image file (access criteria changed)
- Additional features of the program visualization
- Additional features in the working database (WARNING working, operating code 2019)

Corrections:
- Solved program opening in Edicad/TpaEdi32 format (case of replacement of operational codes)
- Solved problem on file re-opening of customization messages
- Solved application problems in Rotation on Cartesian plane (tool and working)
- Solved problem in display of outgoing segment in a profile (case of last segment programmed expanded)
- Solved minor problems of graphic representation
- Solved minor problems

Version 1.1.3 (February 11, 2013)

New features:
- Additional command "Convert a program archive" (File menu): to read and store a program batch
- Additional workings in the database (increased to three the logical condition nodes of the SUB, STOOL, "Global functions" codes)
- Best command management of "Create fictive face from programmed geometry" (selection in piece-face of)
- Additional features in configuration of import from DXF (default values of the parameters recognized on the layers)
- Additional manual in French

Corrections:
- Solved problems in the change of program typology
- Solved problem related to emptying workings (some available technological parameters)
- Solved minor problems

Version 1.1.2 (January 22, 2013)

New features:
- Additional features in graphic customizations of the current program (colour, texture)
- Additional features in texture display
- Best interactivity in the assignment of the "Path" working
- Additional text preview while inserting a text development working from the system font
- Best management of the writing development by using the custom Font (automatic distribution of)
- Best management of debug file (for subroutine/macro-program application)
- Best encryption of macro-programs
- Best management of the display window for the expanded working list
- Additional features in accessory control of Graphic preview
- Additional features in configuration of the Format export modules
- Additional features in configuration of the command "Optimization preview"

- Additional manual in language (Spanish)

Corrections:
- Solved problem in writing preview with selection of System font (non-supported styles)
- Solved problem of message association to the workings (Importer from DXF)
- Solved minor problems

Version 1.1.1 (December 17, 2012)

New features:
- Additional features in snap on Standard Grid management
- Additional management of working insertion with automatic confirmation
- Additional features in field management for COLOUR typology of custom section
- Parametric programming, additional variable arguments (prgnum)
- Parametric programming, additional functionalities (geo[sub;..])
- Additional features in configuration of the Format export modules

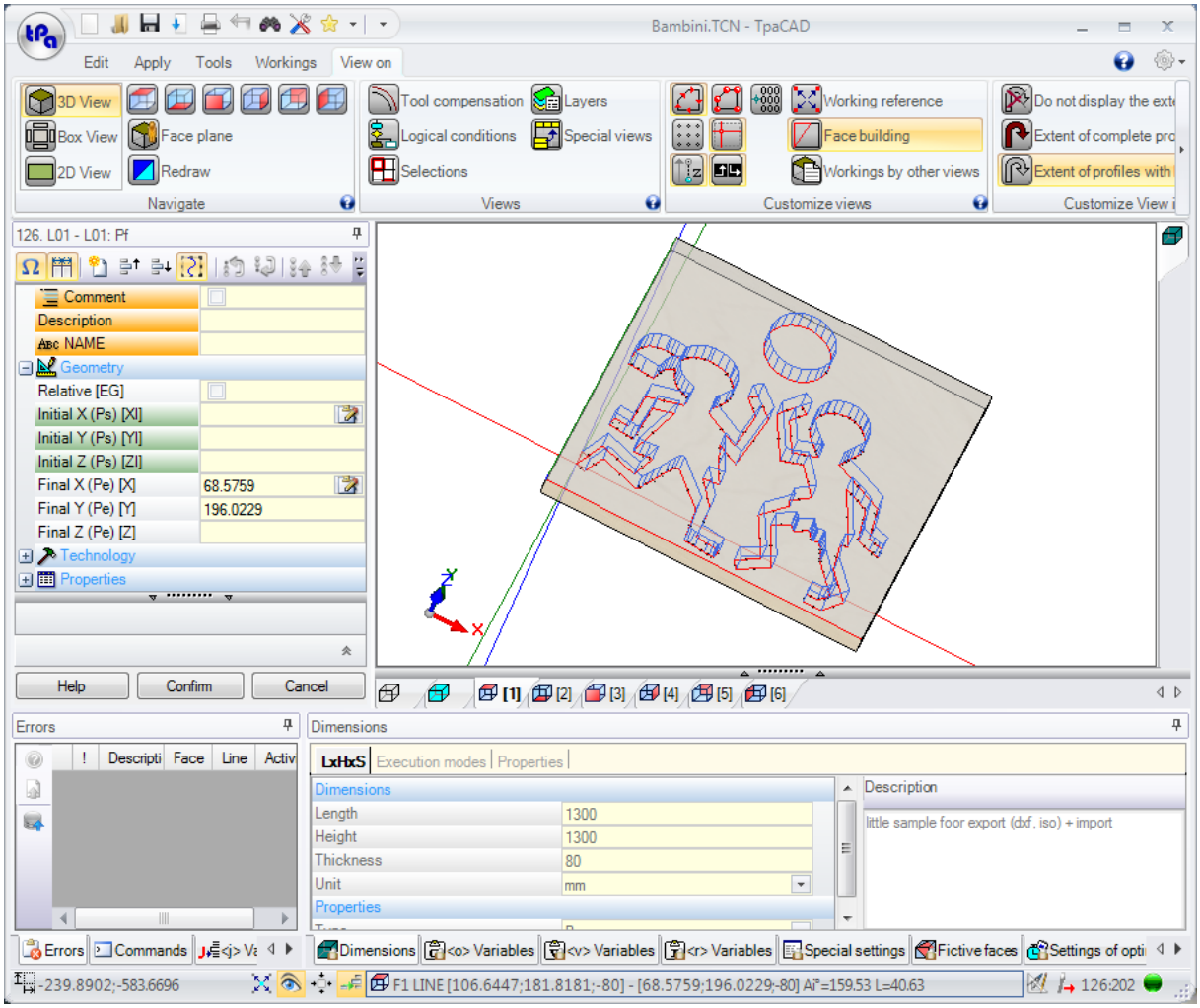
Corrections:
- Solved management problem of additional languages in accessory components (optimizer, importer from DXF)
- Solved problem in custom messages file writing
- Solved problem in the management of the tabs in ASCII text table
- Solved technological assignment problem for punctual workings
- Solved minor problems

Version 1.1.0 (December 06, 2012)

- First official release of the application

3 Graphic interface

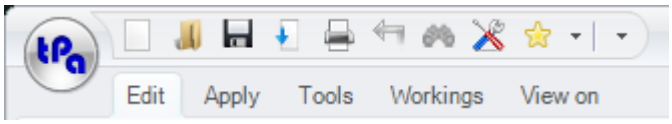
3.1 Overview



The area work of TpaCAD is divided into the following main areas:
Application menu
This is the main menu and corresponds to selecting the button with ***tpa*** (as in the figure) or ***File*** written on it (depending on the style chosen for the menu). It contains commands regarding the program file management, such as New, Open, Save, Print, Close. Some of these commands are shown in the Quick Access Toolbar.
In addition to the commands concerning the direct management of programs, the menu allows the access to the Customization window and the TpaCAD closing command.

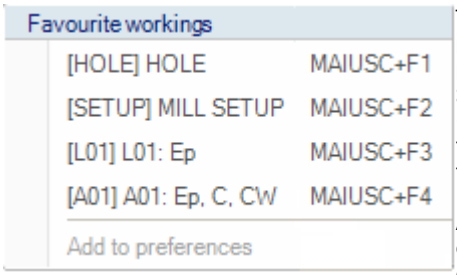
Quick Access Toolbar

The Quick Access Toolbar is a bar containing a set of independent commands in the command tab currently displayed. The bar is placed in the upper left corner next to the icon of the application program.



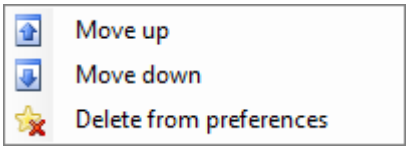
The commands in the toolbar are some of those you can find on the Application menu, besides the Edit tab (Cancel, Redo, Find and Replace) and the Technology table commands.

The selection of the icon  opens a pull-down menu showing the list of the favourite Workings, maximum 15 entries.



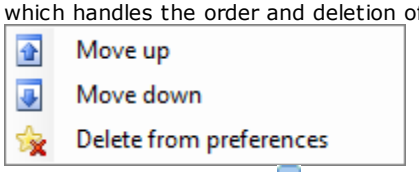
The list in the image is an example: while building a program, it is possible to insert workings directly from the list without accessing the palette of the workings.
Some shortcut keys are assigned to the first 9 entries in the list: from (Shift + F1) to (Shift + F9). The association of the shortcut keys follows their position in the list: moving the position of a working on the list, the shortcut keys change.

Add to preferences: it adds the current working to the list. This command is active in face view and when the program list is not empty: if the current list is already sized to maximum 15 entries, as allowed, the new entry deletes the first entry of the list.
The command is not available in the menu, if the palette of workings is made invisible by the TpaCAD configuration: in this case ho e assume that the list of the workings required for the direct entry is prepared during the configuration of the application and here it cannot be directly changed.
Selecting a working entry with the right mouse button, a local menu opens, that manages the order and the deletion of the items in the list:

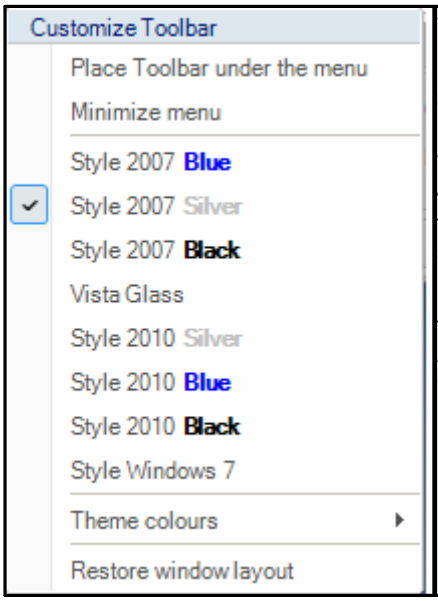


The command for the deletion is not available in the menu, if the palette of the workings is made invisible by the TpaCAD configuration.

By selecting the icon  a pull-down menu displays the list of the Favourite tools. The icon is visible if the favourite tools management is enabled by TpaCAD configuration.
The menu item **Add to preferences** adds the last selected tool to the list of the favourite tools. The command is active in face view and with a non-empty program list. Right-clicking on a tool in the list opens a local menu, which handles the order and deletion of listed items:



The selection of the icon  in the quick access toolbar opens a pull-down menu



the first commands allow the change of position for the ribbon (see later).

Style 2007 Blue
..
A list of entries to choose the TpaCAD presentation style follows.

Theme colours
This command allows to customize the theme colour associated to the style currently used.

Restore window layout
This command allows to recover the original graph of the program.
At TpaCAD launch, it restores the closure layout, recovering the dimension and the position of the controls.

Ribbon

The commands are organized according to similar features in tabs inside a multifunction bar (Ribbon).
This contains up to 6 tabs:
Edit, Apply, Tools, Workings, View on, Nesting
that in turn contain command groups.
The composition of each tab can change according to TpaCAD configuration.

Edit tab:
- Clipboard group
- Edit group
- Place at line group
- Assign property group
- Set group

Apply tab:
- Apply to piece group
- Draw group
- Blocks group
- Advanced group
- Measures group
- Dimensioning group

Tools tab:
- General tools group
- Change profiles group
- Constructions group
- Nesting profiles group

Workings tab:
- set of workings organized according to TpaCAD configuration

View on tab:
- Navigate group
- Views group
- Customize views group
- Customize View in tool compensation group
- Info group

Nesting tab:
- Refer to the Nesting functionality documentation.

Some command sets with immediate application have a button : select the button to open the TpaCAD help on the section describing the commands.
On the right side of the ribbon there are two buttons:
It opens the online program help
It opens a command menu for the available configurations. The menu composition changes according to the configuration arranged for TpaCAD. The menu can include the command for the direct access to other online help files.

The ribbon can be minimized from the Customize quick access toolbar , by selecting the entry in the Minimize menu list, or from keyboard by pressing the [CTRL + F1] shortcut, or by double clicking the title of a tab in the bar itself.

To restore the ribbon from Customize quick access toolbar , select in the list the Minimize menu entry or press on the keyboard the [CTRL + F1] shortuct or double-click on the title of a tab of the bar itself.

Working data area
This is the area where the geometric and technological data of a programmed working is displayed. The control can be moved within the work window with different anchor points at the edge of the window itself or by aggregating it to the space of other controls (Errors, Dimensions, …). To move the control the operator just needs to click on the title bar of the control and, holding down the left mouse button, has to move the pointer on the cell that, among those appearing in the menu, corresponds to the position required.

Graphic area of piece representation
This is the area where the program is graphically displayed and corresponds with the current view.
Overall view
It is possible to choose between the following options:
- graphic three-dimensional representation of the piece (3D view): the piece is displayed in 3D (xyz), complete with all planes (faces), of which the piece consists of and with all the workings applied.

- box view representation: the exploded view of the panel is displayed for the only enabled faces of the parallelepiped piece with the workings of the represented faces applied. The selection of Box View may not be available, according to TpaCAD configuration.

Face View
It is possible to choose between the following options:
- Three-dimensional graphic representation (see general view). The current face and the face workings are coloured to be highlighted with respect to the other faces and workings. The workings of the other faces are grey or their view can be excluded.
- Box view: the current face and the workings of the faces are coloured to be highlighted with respect to the other faces and with respect to the represented workings. The selection of Box View may not be available, according to TpaCAD configuration.
- 2D view: bi-dimensional graphic representation on xy face plane and of the only workings programmed here.

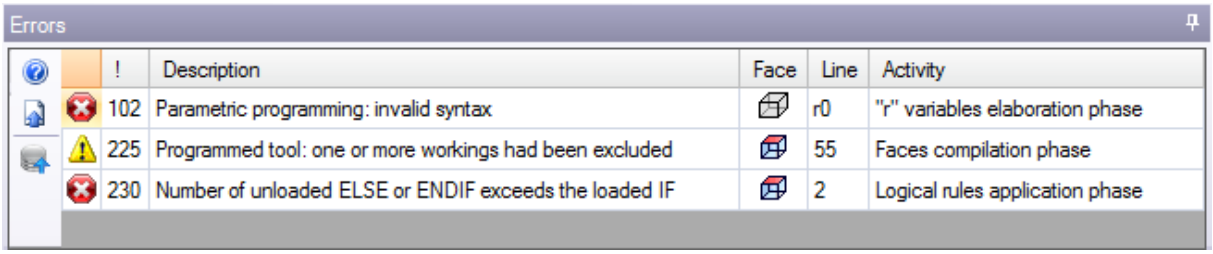
Working list area in ASCII text format
This is the area where the program of the current face in ASCII format is displayed. The area is edited in face view and it is organized in a table:
- each row corresponds to a programmed working line;
- the information of each line is presented in columns. More specifically, the property fields are distinctly shown.

Only a few of the data reported in the table are editable, also according to the TpaCAD configuration.
By means of different anchor activations, the control can be moved around the control of the graphic representation of the piece. To do it, click with the right mouse button the bar marking the area, then select a menu item.

Area of the general assignment of the piece
This is the area to view and set the general program information: Dimensions, variables, Special sections, Variable Geometries, Sequences.

Area to display errors, commands, j-variables, bookmarks
This is the area assigned to display additional information, organized in tabs: errors, commands, <j> variables, bookmarks, and debug files. TpaCAD changes the display of the window in accordance with the current operation. To force the display change, the user just needs to select the corresponding tab.

The area of **errors** shows the errors and warnings detected during the program processing. The displayed errors and warnings refer to the active view, for instance: in the editor of the <r> variables compilation errors are reported; in face view the errors found on the face are displayed.
See in the picture below an example of three warnings:



The first and the third one are error reports, the second one is a warning.
Let us describe the columns that constitute the error window:
- And : respectively error icon and warning icon. A third icon can report a warning situation in TpaCAD; however, the report turns into error, if the execution of the program is required (severe warning).
- **!**: warning identification number
- **Description**: warning description
- **Face**: graphic representation of the face view to which the warning is referred. Moving the mouse cursor on the graphic representation, a help message showing the face number is displayed.
- **Line**: number of the program or variable line to which the warning is related.
- **Working induced in the face..**: graphic representation of the induced face, from which the report is generated (the column is reported only if necessary)

- **Activity**: description of the working process where the problem occurred.

Press the button to recall the contextual help referring the occurred error.
Select to move to the working where the error occurred
Select to request a general working process of the program. In this way a list of errors is generated again.
If on the side bar also the button is visible, that means that the program processing has produced an excessive number of signals: in this case the window shows only the first 100 errors.

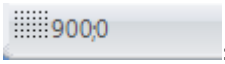
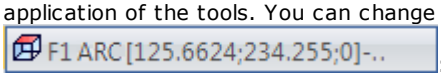
If the display area is minimized, if there are errors, the same area becomes visible, in order to draw the attention of the operator.

The area of the **commands** shows:
- the command exit status, for example Creation of a new program or Open an existing program
- the sequence of the executed commands
- the sequence of the operations to be performed during the execution of a command.

The area of **J variables** shows the table of <j> variables sorted in 10 rows and 10 columns.
The area of **Bookmarks** shows a table where it is possible to assign significant auxiliary positions for the interactive procedures. This area is available according to the TpaCAD configuration.

Status bar
The status bar shows several pieces of information that change according to the operations you are performing.



-  : it displays the mouse position on face view. If an interactive procedure is activated, for example the insertion of an element from the Draw menu, the icon displayed may show the type of active snap (on grid or entity) and the mouse position takes the snap into account
-  : if selected, it shows that the snap to grid is enabled
-  : if selected, it shows that the program display is active. The command is located in the status bar according to the TpaCAD configuration
-  : if selected, it shows that the snap cross-hair is to be displayed in interactive procedure when request of face and/or entity snap is active
-  : if selected, it shows that the next insertion of a working will be done after the current working. Cancel the selection in order to perform an insertion before the current working: once the insertion is made, the selection status will be automatically changed to active.
-  : it shows the last setting chosen for the selection *Apply to a copy of the workings*, available in the application of the tools. You can change the status directly.
- 

: it shows the geometric and technological information of the current working, and points to the application face (icon and face number)
-  : it shows that the previous field displays the real coordinates of the selected working. If View in Tool compensation is selected and clicking the icon, the compensated coordinates and the icon  are displayed. Clicking again the icon, the user comes back to the real coordinates and the icon  is displayed again.
- 

: it shows the progressive number of the current program line and the total number of the lines.
-  : the image is visible only if more instances of TpaCAD are running in the same environment and the active one does not correspond to the first instance started.
-  : if it is green, it informs the user that TpaCAD is waiting for commands. If it is red, it informs that TpaCAD is performing a processing phase (for instance, the graphics is being updated).

3.2 List of Shortcuts and Mouse Keys

View Window

[CTRL+W]: Window Zoom
[CTRL+right mouse key]: Zoom In-Out
[CTRL+Shift+W]: Previous Zoom
F6: Extension Zoom command
X: upward rotation around horizontal axis
[Shift+X]: downward rotation around horizontal axis
Y: leftward rotation around vertical axis
[Shift+Y]: rightward rotation around vertical axis
Z: clockwise rotation
[Shift+Z]: counterclockwise rotation

F3: activates box view
F2: activates 3D view
F4: activates 2D view
F5: redraws
F7: activates the display of the tool compensation

F8: activates the display of the logical conditions

Fictive faces

[CTRL+left mouse click]: selects or deselects the line on the line header cell

Sequences

[CTRL+left mouse click]: selects or deselects the line on the line header cell
[Shift+(left mouse key pressed)]: starts the area selection
[Shift+CTRL+(left mouse key pressed)]: the workings in the selected area are added to the current selections in the table
[CTRL+(left mouse click)]: selects or deselects the working closest to the mouse position
[left mouse click]: it moves the current line to the working closest to the mouse position, resetting all the selections

Menu for inserting geometric entities and acquiring dimensions

I: reverses the direction of rotation of the arc
L: switches from arc to line when constructing a polyline
A: switches from line to arc when constructing a polyline
C: closes on the starting point when constructing a polyline
P: uses the last point entered
F: snap between faces
D: snap on depth
G: snap to grid
X: stop X axis
Y: stop Y axis
T: forces exit in tangency of a segment from a segment
Z: cancels the last segment
S: snap on entity, enables snap commands
[Enter]: finishes insertion
[Escape]: cancels the insertion phase

Snap on entity Menu

[CTRL+P]: snap on a programmed point
[CTRL+N]: snap on the coordinates of the point closest to the cursor
[CTRL+M]: snap on a midpoint
[CTRL+C]: snap on the centre of an arc
[CTRL+I]: snap on intersection point
[CTRL+O]: snap on a perpendicular point to segment
[CTRL+T]: snap on a point tangent to segment
[CTRL+Q]: snap on point of quadrant change
[CTRL+E]: snap on the face edge
[CTRL+F]: snap on a bookmark
[CTRL+X]: snap on a point of a modelling profile

Selection of a working

Arrow up: moves the active line to the previous block
Arrow down: moves the active line to the next block
Previous Page: moves the active line 10 lines earlier
Next Page: moves the active line 10 lines later
Home: moves the active line to the first line of the program
End: moves the active line to the last line in the program

Selection in graphic view

[Shift+left mouse key]: selects the workings of the identified area
[Shift+left mouse key+ALT]: selects the workings of the identified area by extending the selection to the entire profile
[Shift+left mouse key+CTRL]: selects the workings of the identified area by keeping the previous selections

Selection in ASCII text

[Shift+left mouse key]: selects from the active line to the line pointed to by the mouse
[CTRL+left mouse key]: selects or deselects the program line pointed to with the mouse

General Selection Commands

[CTRL+A]: selects all the face workings

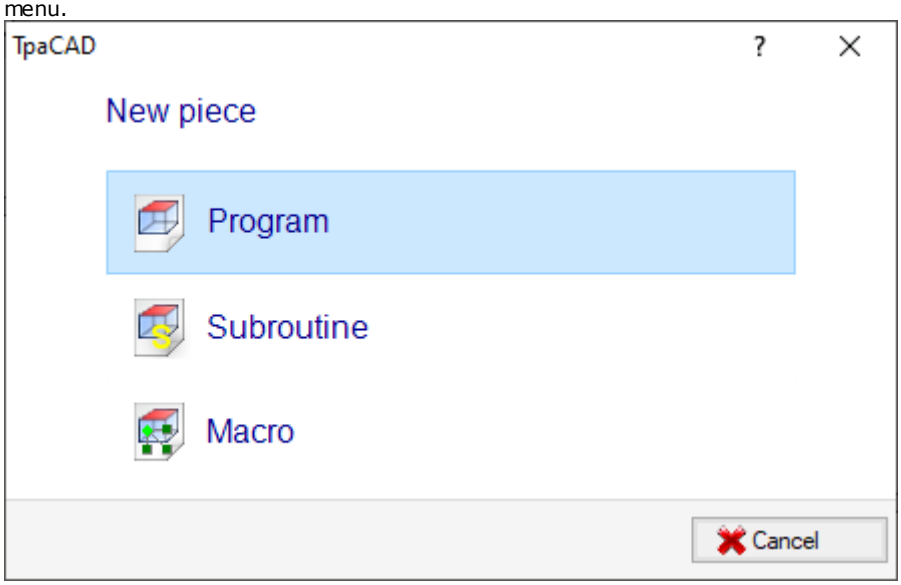
General Purpose Change Commands

[CTRL+C]: copies the selected text or workings to the clipboard
[CTRL+V]: pastes the text or the workings from the clipboard
[CTRL+X]: cuts the selected text or workings
[Del]: deletes the workings
[CTRL+Z]: undoes the last command
[CTRL+Y]: redoes the last undo
[CTRL+F1]: minimizes or maximizes the ribbon
[CTRL+2]: activates the working data area
[CTRL+3]: activates the working list area in ASCII text format

4 Working with the programs

4.1 Creating a program

TpaCAD creates programs, subroutines and macros through the command **New** (icon) from the Application menu.



Normally, we choose to create a **Program**. When it is necessary to define once and for all the set of workings to be used repeatedly in a program, we choose to create a **Subroutine**. The possibility to create a **Macro** program is active only if the access level is equivalent to Manufacturer.
Depending on the configuration, the command **New** can directly start the generation of a **Program** without opening the selection window.
The piece typology can be directly changed in the editor phase.

During the creation process, the new program is initialized by using the default prototype program (PIECE.TCN, in the folder TPACADCFG\CUSTOM). In case this default program is not installed, a 500*500*20 mm program is created. The prototype program can be sorted by typology: see paragraph Customize the "prototype" file.

By clicking on the icon of the window, the contextual help of the current window is set up.

TpaCAD can create and/or open one program at a time. However, multiple instances can be started, up to maximum 4.

Create a program according to a template

To create a new program by choosing a model, select **New from template** from the Application menu, then in the dialog box select an already stored program and confirm.
A model is a prototype program that already contains the desired settings for the initialization of the new program. The dialog box shows the content of the default folder of the models.
You can confirm the creation of the program without using a template by clicking on **Open without template**. In this case the command continues, as in the previous case, and uses the default prototype program.
The command **New from template** creates one **Program** only.

4.2 Opening and importing a program

TpaCAD allows you to create programs, subroutines and macros through the command **File -> Open** (icon) in the Application menu.
TpaCAD saves two typologies of files:

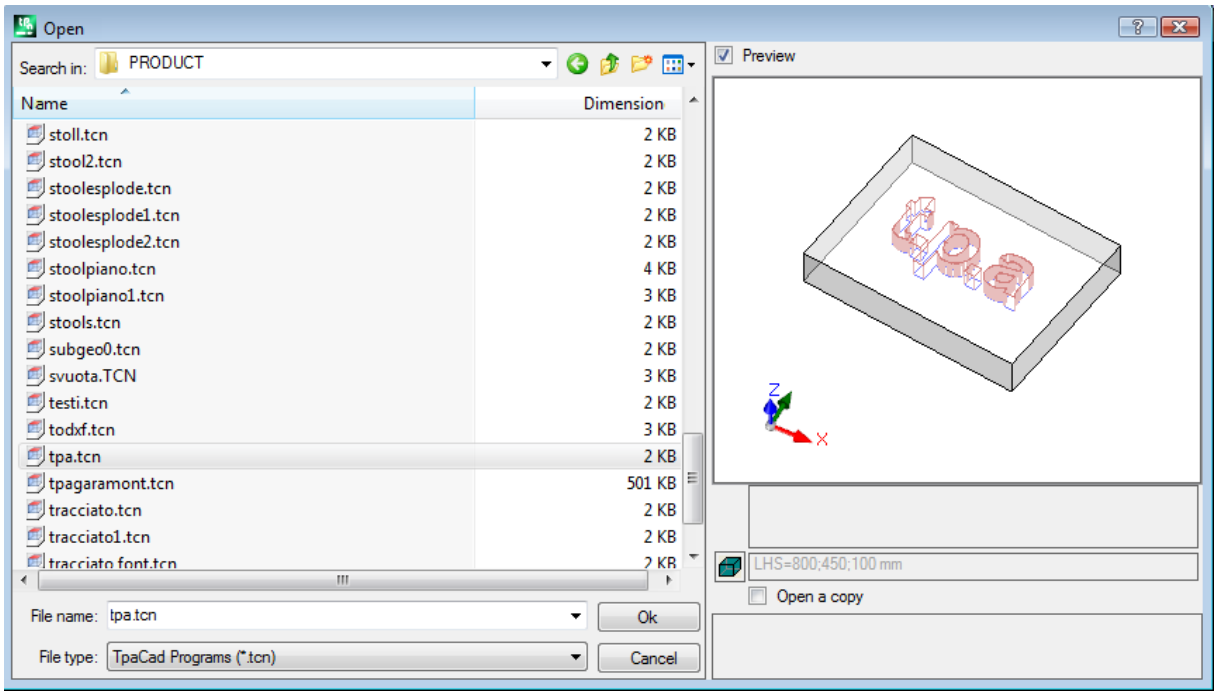
- TCN extension: default for programs and subroutines (type of file: **TpaCAD Files**)

- TMCR extension: default for macros (type of file: **TpaCAD Macro**).

This command opens a custom *Resource manager* window:
- the two indicated typologies are reported in the **File type** list. The Macro typology is shown only if the access level allows its opening.
- Select the typology "All files (*.*)" to set no display filters: with these selections it is possible to open only those programs directly recognized by TpaCAD application.

For a program or a subroutine in any case it is not compulsory to assign the extension corresponding to the selected typology. However, it may be helpful to the immediate recognition of the program-pieces.
If the program selected in the list is recognized as a program-piece, it is possible to display that graphic preview (**Preview**).
In the window comment and size are also displayed.
If the size of the selected program is very large, the preview is temporarily disabled to avoid an excessive time in the graphic display. However, if you need a program preview, you have to enable the option **Preview**. The size beyond which a program must be considered "large" should be defined in ***Customize -> Environment -> Saving***.

Next to the area reserved for the comment, two images can be displayed that show if the selected program opens as read only and/or write only .
Select the option **Open a copy** to open the file as a copy: the program is uploaded from the selected file, "(2)" is added to the name and is considered as a new program for saving purposes.



Import a program from external format

If configured by the machine manufacturer, on the window you can select typologies of formats different from the TpaCAD format. Base procedures are carried out by external components, linked to TpaCAD.
Select first the required typology from the list displayed in the window (for example: "DXF files (*.dxf)"), then a file of the selected typology: the confirmation of the opening directly starts the conversion of the file to a TpaCAD format.
The execution of the graphic preview for this file depends on the configuration settings defined by the manufacturer of the machine. A program opened by format conversion is considered as created by the New Program command.

If a file is opened by external conversion format, global tools on the program can be activated, according to the definitions in **Customize -> Environment -> Import format**. More specifically, the activation may be:
- operational in an automatic way, that is without any request for confirmation;
- depending on a proposal requested by the operator with a message "Do you want to apply the automatic assignments set?"

The automatic assignments involve the application of particular tools to the program:

- general settings read from the PIECE.TCN prototype file (stored in the folder TPACADCFG\CUSTOM); more specifically: execution mode, <o> and <v> variables, custom sections.
- application of technology to open profiles (that do not start with a setup working) or assigned with a geometric setup code
- application of technology to the point workings assigned with code of geometric point
- reduction of the fragmentation for linear profile segments, with evaluation of an angle of cumulative reduction
- fragmentation and linearization of the profile arcs
- automatic connection of profiles to verify their geometric continuity.

Before opening a file in an external format, according to the TpaCAD configuration, it may be possible:

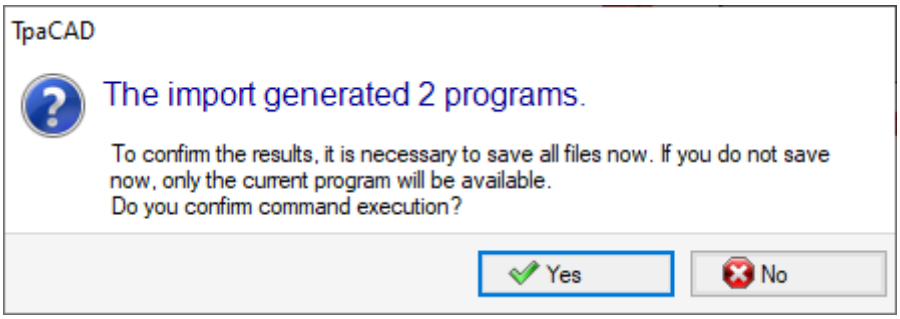
- to set possible arguments that can be used for the conversion: in the opening window a field  is displayed, where to assign the arguments. The field is initialized with the default arguments, assigned by the manufacturer of the machine. It is important that the user who sets the arguments of conversion that knows their meaning.
- to request the customization of the single conversion as managed by the conversion module (for example, if the program is in DXF format, you can indicate which layers should be converted and which should be excluded from the conversion, instead).

ATTENTION: if all six real faces are disabled, a program import assigns the subroutine typology.

Import with generation of more files

A particular case may correspond with an import with generation of more TCN files.
For instance, the situation may correspond with creating a second program with the workings of face 2 (bottom face).
In this case, everything will occur as indicated above, and a warning will show that this particular situation is taking place.

The picture coincides with the case of 2 programs created:



Proceed confirming the immediate saving in order to save the importing of both programs.
Otherwise, only the first imported program will be available, while the second one (or the following ones) will not be recoverable.
In our example, the two files are saved by choosing the position and the name of the first file. The second one (or the following ones) is saved with the addition of an increasing suffix to the assigned name.

Import a program in TpaEdi32 format

A program written with TpaEdi32 can be directly uploaded and processed. The program format is automatically recognized by selecting one of the (*.TCN, *.TMCR, *.*) TpaCAD file typologies. Opening a program written in TpaEdi32, the execution modes can be automatically assigned by the settings defined in the prototype file PIECE.TCN: to carry out the assignments, it is enough to confirm when the prompt message box is opened.
We remind you that a program generated with TpaCAD can be read by TpaEDI32, only if it is saved in TpaEdi32 format.

Import a program in EdiCad format

A program written with Edicad can be directly uploaded and processed: the program format is automatically recognized, by selecting one of the TpaCAD file typologies (*.TCN, *.TMCR, *.*).
But it is not possible to operate in a reverse way: a program generated with TpaCAD cannot be read by EdiCad.
To read a macro written with EdiCad, it is necessary to save it in ASCII format directly in Edicad.
Opening a program written in EdiCad, it is possible to automatically assign various settings by means of the settings defined in the PIECE.TCN prototype file. To carry out the assignments, it is enough to confirm when the

dialog box is opened. Besides the execution modes, it is now possible to automatically assign the custom sections including Special settings, Additional Info, Constraint section, Optimization settings.

Information retrieved during the import:

- The three offsets are retrieved in the first three "o" variables
- The cycle variables are retrieved in the "v" variables
- Fictive face information: the similar face assignment is retrieved in the direction of z axis
- In parametric programming every use of , (comma) is replaced by ; (semicolon)
- In the reading of program workings: the parameter corresponding to the comment is retrieved as working comment (for example: IF, FOR, …)
- In the reading of the program workings, a few working codes are reassigned with other equivalent codes
- The programs assigned in EdiCad as sub-cycles are retrieved with macro typology

Information lost during the import:
- Sequence field assignment in single face
- The assignment of shaped workpiece

Non-retrievable information during the import:
- Technological functions of parametric programming concerning multi-drill heads
- The sub-cycle call syntax with the "*" character to address the subroutine call in TPACADCFG\SUB directory is no longer supported.

Opening a program-piece created in an external environment

A program recognized as a program-piece can be identified as non-created in the TpaCAD environment: this is the usual situation of some programs generated by an import or by a management system.

In these cases, opening the program can integrate programming
- of general program information (variables, custom sections) and/or
- of workings with default settings.

This behaviour is determined by what assigned in **Configuration of TpaCAD**.

Starting TpaCAD from Resource Management

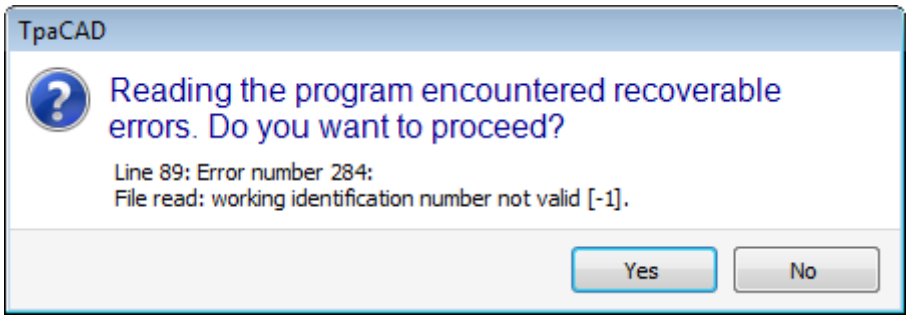
You also can start TpaCAD directly from the Resource Management, requiring the opening of a TCN extension file.
In this case, when the program starts, no functions of Plant selections or of Operational environment are managed, even if they are enabled.

Reports while opening a program

While opening a program that is recognized as a TpaCAD format (including files recorded by TpaEdi32 and Edicad applications) you may detect situations of formal error. These are mainly situations of non-observance of the syntax required for a TpaCAD program and usually correspond to files generated by an external process.
For the description of the error reports read the chapter Error Messages.
There are two cases of errors:
1. non-recoverable errors
2. errors that are deemed recoverable.

The first case necessarily leads to the cancellation of the command to open the program.
The situations you may encounter are:
- memory allocation failure: this is a serious situation that reveals that the system is running out of the available memory
- file recorded with a non-univocal formalism: workings stored in both ASCII and native format are recognized. Undoubtedly, the file was not created by a CAD of Tpa.
- there are unmanaged sections, without indication of the closing line of the section itself. Undoubtedly, the file was not created by a CAD of Tpa.

The second case can be solved by activating an automatic procedure of error recovery. Below, a possible report:



The window shows the first error situation found and warns that recoverable errors only were found. If you enable the recovery procedure, at the end of the reading, the user is warned that the operation was successful. To validate the program, you will have to save it. The program is marked as modified and at program termination you will be requested if you want to save it.

This recovery procedure is enabled in TpaCAD only. The reading of the same program fails when it is executed in the machine.
In the event that the program has been generated by an external application, the opening with a recovery procedure can indicate permanent errors in creating TCN files, which is recommended to correct: in fact, the recovery procedure can lead to the automatic deletion of significant parts of the program itself just because of a wrong original syntax.

Recording format of a program-piece

A program recognized as a program-piece, regardless of the extension of the file and of the recognized type, is a text format file, registered with ANSI or Unicode encoding.

Both encodings are always recognized.
The encoding of a program intervenes, for example, in the programming of the description of a program, in the development of a text or in the name that a subroutine can have.

ANSI encoding is based on an original basic encoding layout of 95 printable characters:

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

then extended to 255 characters, by means of a particular system of code pages, corresponding to the local settings of the system. This system allows representing specific characters of a language, or group of languages, using in any case a limited number of codes. The result is obtained by assigning different representations to the same code, as the encoding that is set in the system modifications. If, for example, we consider one of the stressed characters based on a local setting valid for Western Europe (**à**, **ò**, …) and if we record the character in an ANSI format file, the same file opened in Cyrillic or Hebrew culture setting shows a different character as a result of the recognition of a different active code page.

However, this encoding is not sufficient when you work, for example, with Asian languages that generally have many more characters than the languages normally encoded with 255 characters. The solution is the management of the files with Unicode encoding:

- you can read files recorded with both encodings
- when you create a new program, the default Unicode encoding is automatically assigned
- when you save a program, you can choose which encoding to use.

Please refer to paragraph Saving a program for further details.

4.3 Open a program from the recent file list

The File menu displays a list of recently opened files up to a maximum of 10 entries. Double-click a name in the list to directly open the program.
Click with the mouse right key on a name to open a simple menu:

Open file path: opens the *Open* dialog box directly on the folder where the File is stored

Open a copy: opens the File as a copy. The program is uploaded from the selected file, "(2)" is added to the name and is considered as a new program for the saving purpose.

Remove from the list: deletes the entry from the list.

4.4 Drag (drag & drop)

If you drag a file, for example from Resource Management, and drop it within the TpaCAD working area, you can open a program in TpaCAD format. If the file is not recognized as TpaCAD format, a possible conversion is checked by means of the configured import modules. If the check is positive, conversion and subsequent opening of the file are performed in accordance with the mode described in the previous paragraphs.

The drag is ignored, if a procedure is under way, such as a window opening and waiting for a command to be competed.

4.5 Printing a program

TpaCAD prints the active program in the view represented in the graphic area with the command **File -> Print -> Print graphics** (icon ) from the Application menu.

Active zoom and pan, view filters, special views and all the active graphic elements are considered (cursor, grid, edge points and profile direction arrows,...).
Printing a program may require the graphics upgrade due to particular complex working settings, that may in fact require a print customization. In this case, the graphic representation is updated first, then a confirmation is requested to start the Print procedure; when the command is over, the graphics returns to its original status.
Print customizations correspond to program lines containing specific conditions, for instance they may correspond to additions of: writings and/or elements of dimensioning, hatches, auxiliary contours. Anyway, they represent aspects concerning macro-program texts.

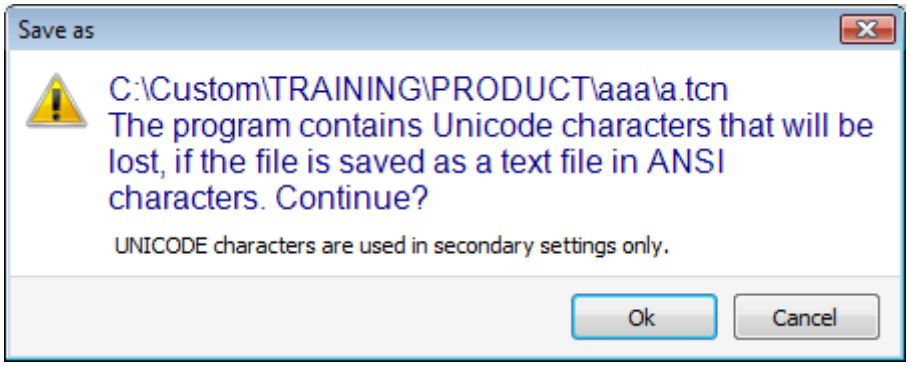
4.6 Saving a program

TpaCAD stores the current program through the command **File -> Save** (icon ) in the Application menu. In case the program being edited is new or if the command **File -> Save as** is selected, the window to assign the file name and its storage location is displayed. The extension to be assigned to the file can be chosen among those suggested or can be defined by the user. The default extensions are: TCN for programs and subroutines, TMCR for macro. In case of storage of a macro-program, only the TMCR extension is shown: compulsory for a correct recognition.
We suggest not to assign the TMCR extension (by default for macro) to programs or subroutines.

If the program is modified compared to the last storage, a message shows the situation and asks to confirm the saving.
During the execution of the command **File -> Save as**, if an existing program path in TpaCAD format is selected, no storage is performed, if the file to be overwritten is internally write-protected with level access higher than the current one (in the opening window, this icon would appear: ).
It is also possible to choose which encoding to use to save, selecting between:

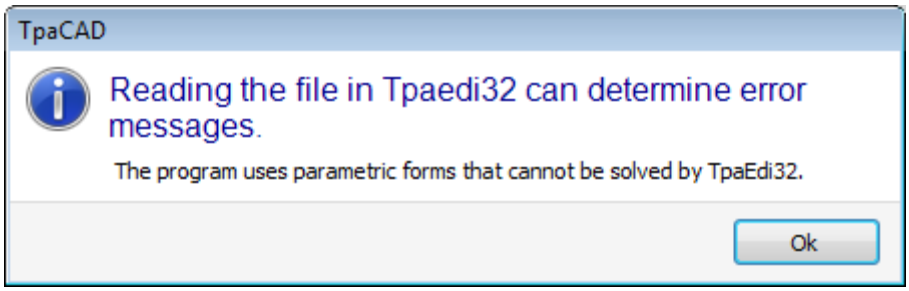
- ANSI
- Unicode.

If you select the ANSI format and the program uses Unicode characters, a message reports the loss of information.



The warning in figure indicates that Unicode characters are used in secondary settings, as descriptions of programs or variables or workings are. The loss while recording a secondary setting changes the video representation of a text, but it does not change the interpretation of a program.
A non-secondary setting is, for example, the programming of a string variable: the loss of information changes the interpretation of the program, causing error situations.

If configured by the manufacturer of the machine, select to save in a format compatible with TpaEdi32. In this case the file format is ANSI only. Compatibility is to be intended as a possibility to read the program with TpaEdi32, without reporting on version incompatibility; as for the actual possibility of program interpretation, the compatibility is bound to the workings and parametric programming in use that should not include new functionalities of TpaCAD. At the end of the saving process a message appears that may also indicate a certain incompatibility in the interpretation of the program in TpaEdi32, as in the following figure:



The storage procedure can be followed by other procedures configurable by the machine manufacturer.
In particular:
- program conversion in an external format (e.g.: ISO format)
- program optimization

These procedures are not activated in case of saving of macro-programs.
Sometimes, enabling these procedures may be time demanding; when clicking inside the graphic area, the user is warned by a message window that the program filing is not finished yet.

Names that cannot be used

Some names cannot be used as they are reserved by the operating system.
Below is the list:

CON, PRN, AUX, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9

The names listed above cannot be used with any file extension: for instance, "con.tcn", "con.tmcr", or "con".

4.7 Optimizing a program

TpaCAD optimizes the current program through the command ***File -> Optimize*** (icon ) in the Application menu.
The program optimization is performed by an external component linked to TpaCAD in accordance with the criteria defined by the machine manufacturer.
If the program has been modified or if it is a new program, the storage process is performed before the optimization. The parameters used for the optimization are those set in the program until the command is selected: execution mode, exclusions, dimensions, variables…
It has already been said that the program optimization can be implemented after a save, as well.
However, the optimization on direct request is generally fuller, and the reports can be stored and/or memorized.

4.8 Print the program label

TpaCAD prints the label associated with the current program through the command ***File -> Print the program label*** (Icon ) from the Application menu.

The label is created according to a format set during the TpaCAD configuration.

A first window allows selecting whether to print the label or to save it to a file.
In case of printing, a second window allows confirming or selecting the printer to be used for printing.

4.9 Exporting a program

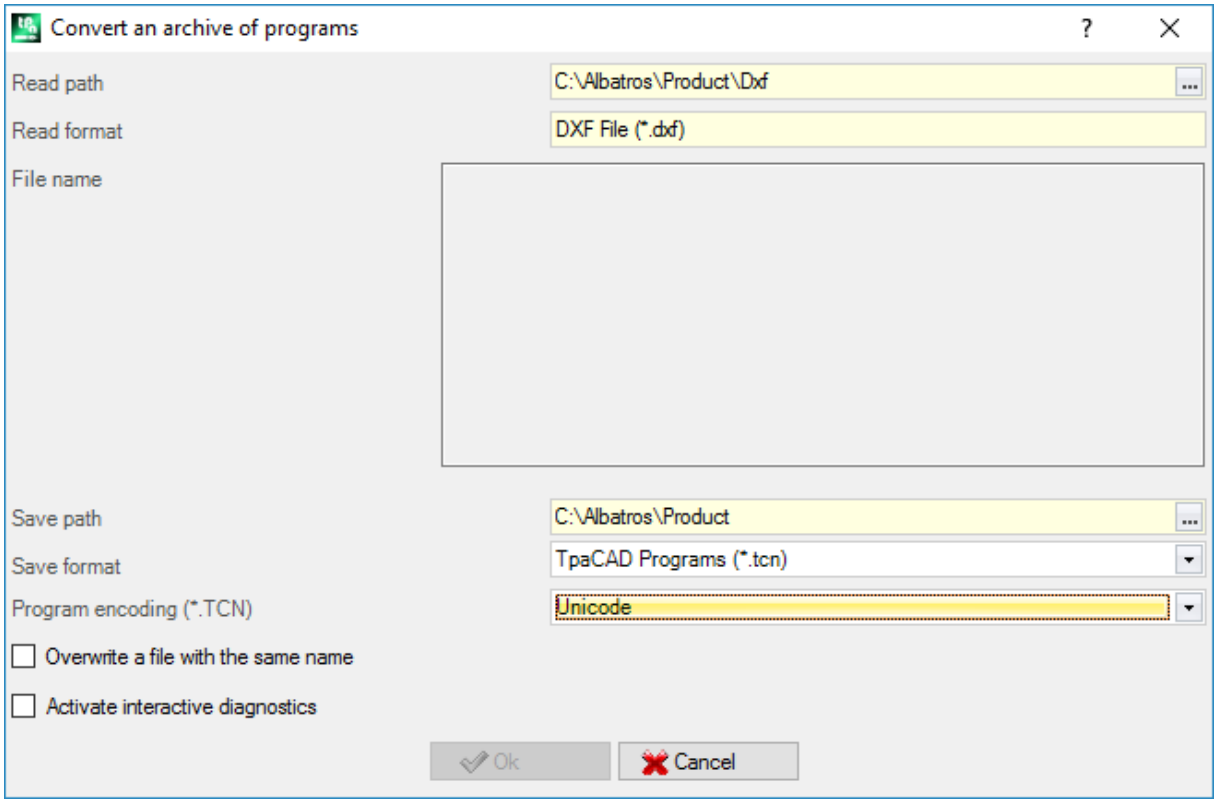
TpaCAD exports the active program in one of the formats configured by the manufacturer of the machine. Export procedures are carried out by external components, linked to TpaCAD.
The command is enabled from the menu **File -> Export** from the Application menu. A menu to select the conversion type appears. Examples of possible selections:

- Edicad file
- DXF file
- ISO file

On selecting a conversion, a program is saved, if needed, and the window to assign the file name and its storage location can appear. The parameters used for the export process are those set in the program until the command is selected: execution mode, exclusions, dimensions, variables.

4.10 Converting an archive of programs

Select the command ***File -> Convert an archive of programs*** (icon ) from the Application menu.



This command starts a program list, specifying one of the different format reading types and the following storage, specifying one of the different format saving types.
Data are saved without executing the optimizer.

Let us see the detailed settings:

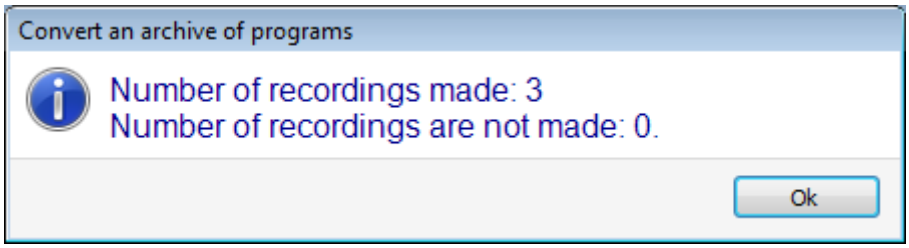
- **Read path**: initialized with the opening path of the programs, it assigns a disk unit and the folder for the program reading. To start the path research select the button on the field, opening the file search box. More specifically:
  you can make a multiple selection of the programs:
  the managed format types correspond to those views for the Program opening command.
  - TCN extension: default for programs and subroutines (type of file: **TpaCAD Files**)

  - "All files (*.*)" to not set display filters: with those type selections you can open only programs directly recognized by TpaCAD application program
  - typologies corresponding to configured import modules.
- **Read format**: it indicates the format type selected in the file search box;
- **File name**: it indicates the programs selected in the file search box;
- **Save path**: initialized with the recording path of the programs, it assigns a disk unit and the folder for the program saving. To start the path research, select the button on the field to open the path search box;
- **Save format**: it shows the list of the possible format types for the program saving.
  - TCN extension: it saves the program in **TpaCAD File** format, with assigned TCN extension
  - "All files (*.*)" saves the program in **TpaCAD File** format: if the programs have been opened without importing the format, the file extension is not changed; otherwise the extension is deleted;
  - typologies corresponding to configured import modules.
- **Program encoding (*.TCN)**: for saving purposes only in **TpaCAD Files** format, selecting between:
  - ANSI
  - Unicode.
- **Overwrite a file with the same name**: select to overwrite existing programs
- **Activate interactive diagnostics**: select to manage an interaction in the window, whenever an error situation occurs. In this case, when you receive a report, you may request an immediate cancellation of the command.

ATTENTION: if all six real faces are disabled, a program import in TpaCAD format assigns the subroutine type.

In case of import with generation of more TCN files, all files are saved.

Confirming the settings in the box, the command is started and executed until it is completed. At the end of the execution, a window shows the number of the processes finished in a correct way or not.



In the area of Commands, it is possible to examine the whole course of the proceeding more specifically and see, if any, the reason of each single wrong processing.

Under the request to save in format of **TpaCAD Files**, when the command execution is completed and at least one processing has been performed, the path assigned for saving *TCN programmes is set as the last one opened for the next program opening.

4.11 Optimizing an archive of programs

Select ***File -> Optimizing an archive of programs***  from the Application menu.

This command allows to start the optimization in a list of programs that should be already recorded in **TpaCAD Files** format. The optimization of the program is performed by an external component linked to TpaCAD in accordance with the criteria defined by the machine manufacturer.

Let us see the detailed settings:

- **Read path**: this field is initialized with the program opening path. It sets a disk drive and a folder to read programs. Through the button  the file search window opens.

More specifically:
- it is possible to make a multiple selection of the programs;
- the managed format types correspond to:
  - TCN extension: default for programs and subroutines (type of file: TpaCAD Files)
  - "All files (*.*)" not to set display filters: with those selections you can open only programs directly recognized by TpaCAD.
- **Read format**: displays the type of the chosen format in the file search window
- **File name**: displays the programs selected in the file search box.

- **Activate interactive diagnostics**: select to manage an interaction in the window, whenever an error situation occurs. In this case, it is possible to request an immediate cancellation of the command at every signal.

When the settings in the window are confirmed, the command is started and performed until it is complete or at the first error situation, if the **Activate interactive diagnostics** case is selected.

At the end of the execution, a warning shows the number of the processes unsuccessfully terminated.
In the area of the Commands, it is possible to examine the whole course of the proceeding more specifically and see, if any, the reason of each single wrong processing.

4.12 Seeing the Program Optimization Preview

It is possible to request the display of a program arriving at its executive step: the command ***Optimization preview*** can be selected from the tab **View on**. This command is enabled in Overall view.
When selecting the command, the program is saved, if needed, and a window to assign the file name and its storage location may appear.

The *Optimization preview* allows the user to see how the program with execution request will be processed, in accordance with the current settings (dimensions, variables). The result can greatly differ from the result normally displayed in editor phase, due to the application of:
- ✓ different parametric assignments,
- ✓ different logic conditions,
- ✓ application of functionality of: Multiple setup, Tool compensation, Arc fragmentation.

If the program process determines error situations, a window opens showing all diagnosis reports and the command for *Preview* is cancelled.

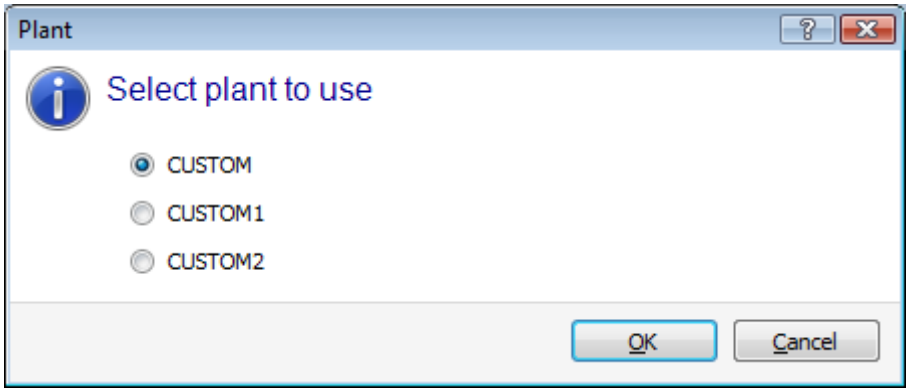
An independent display window opens, in which specific commands and functionalities may be available. Closing the window causes the closure of the command.

4.13 Plant

A plant consists of one or more machines. The machines (or modules) consist in groups, divided into subgroups and devices. Usually, the plant is unique, thus no change is provided for. Sometimes, more configurations are to be installed for different plants.

Plant selection window can be recalled from the menu  **-> Plant** or the selection may be required when TpaCAD is started. This is an optional item to be configured. It is only always active, when the programs are closed.

The window shows the list of the configured plants in alphabetical order, marking the current selection.



Select the name of the plant you want to use and confirm by pressing the button **[OK]**.
We want to stress how noteworthy is the changing of the work Plant: in effect, it is equivalent to work with installations performed on two computers and functioning on two different plants.
The functioning on multiple Plants requires a customized installation of the TpaCAD environment and in general of the whole installed TPA software environment. The selection in TpaCAD of a plant far different from that selected by default does not change the operation of the external TPA environment.

4.14 Operating Environment

TpaCAD can manage a second work environment, called "Draw", which is an alternative to the normal environment used, called "Machine". Switching from the "Draw" environment may depend on the access level.
If available, the switching command between the two environments is on the main bar and it is enabled with closed program and at the layer set as in configuration of TpaCAD:

 shows that the "Draw" environment is active

 shows that the "Machine" environment is active.

As an option, the selection can be requested when TpaCAD starts. This is an optional item to be configured, always active only if no program is opened. To the extent that the access level requires, the selection window of the work environment is always proposed, when the next instance of TpaCAD is launched.

The control of the "Draw" environment can meet particular requirements, such as:
- a highly specific programming environment to enable the geometries and/or the section of the piece and/or the menu composition
- a subroutine and/or macro-program development environment, which is necessarily very diversified.

Controlling the "Draw" environment can also be convenient only to differentiate an environment for normal use of TpaCAD, marked by simplified menus and one more rich and powerful, but requiring more experience in using the program.

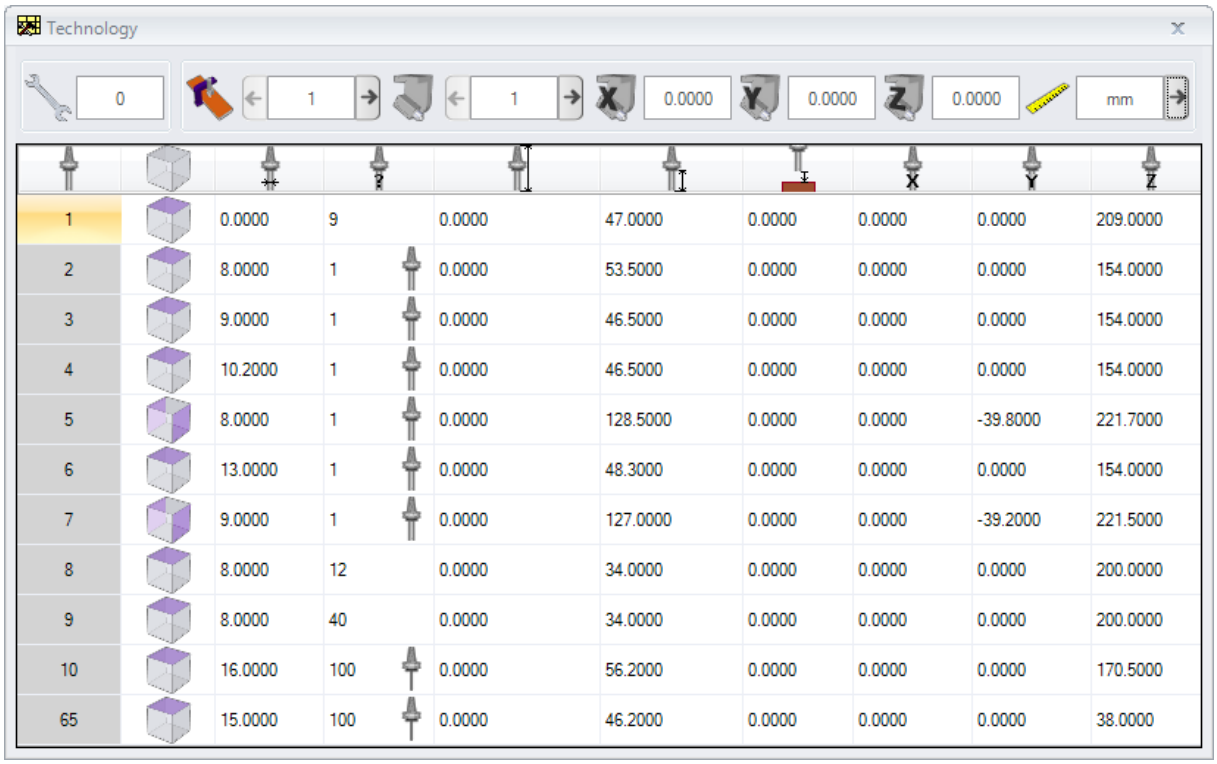
4.15 Multiple instances of TpaCAD

As already mentioned, TpaCAD can create and/or open one program at a time, but you can start multiple instances of the application (up to 4).
For each instance, it is possible to select the Plant and/or the Operating environment, if and as it is provided by TpaCAD configuration.
More specifically, it is possible to perform Copy/Paste operations of workings among different instances.

When you start multiple instances on the same Plant, only the first one is granted the possibility to save changes of Configuration and Customization of TpaCAD.
In the case of non-primary instance, in closing the application, a message informs that no modification in setting and/or customization of TpaCAD will be saved. In a very similar way, the opening of Configuration windows informs that it is not possible to make changes, since the accesses have been limited to all the instances recognized as secondary instances.

4.16 Tool table

TpaCAD normally operates in a plant technological context. It is directly interfaced to one or more machines, of which it knows the assignments concerning the groups and the working tools.
The technological assignment of the tools is of primary interest for workings which can be executed in a program, and it is usually possible to display the table of tools available for workings.
The command is on the Quick Access Toolbar.



1	0.0000	9	0.0000	47.0000	0.0000	0.0000	0.0000	209.0000
2	8.0000	1	0.0000	53.5000	0.0000	0.0000	0.0000	154.0000
3	9.0000	1	0.0000	46.5000	0.0000	0.0000	0.0000	154.0000
4	10.2000	1	0.0000	46.5000	0.0000	0.0000	0.0000	154.0000
5	8.0000	1	0.0000	128.5000	0.0000	0.0000	-39.8000	221.7000
6	13.0000	1	0.0000	48.3000	0.0000	0.0000	0.0000	154.0000
7	9.0000	1	0.0000	127.0000	0.0000	0.0000	-39.2000	221.5000
8	8.0000	12	0.0000	34.0000	0.0000	0.0000	0.0000	200.0000
9	8.0000	40	0.0000	34.0000	0.0000	0.0000	0.0000	200.0000
10	16.0000	100	0.0000	56.2000	0.0000	0.0000	0.0000	170.5000
65	15.0000	100	0.0000	46.2000	0.0000	0.0000	0.0000	38.0000

If the universal tool management is enabled in the TpaCAD Configuration, the first page shows the list of universal tools and the machine and group fields have a value of 0.

The display window can change according to the configuration assigned to TpaCAD. The figure above shows how the plant tooling appears: the configuration features of the tools of a group are listed in a table, where it is possible to select the groups of each machine of the plant. The minimum structure of a technological configuration includes one machine with one group only.



Tooling

It shows the number of the current tooling. This area is visible if the tooling can be selected in the program. The field is unchangeable and shows an integer numerical value greater or equal to 0. The tooling is a picture of the way the groups of a machine are prepared: it is a machine structure. Managing more tooling processes is typical of a plant with one machine only.



Machine

Machine selection field: it allows you to scroll through the machines configured in the machinery. If the universal tool management is enabled in the TpaCAD Configuration, the Machine field can assume value 0.

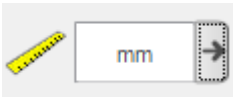


Group

Selection field of the group on the machine: it allows you to scroll through the groups configured in the selected machine. If the universal tool management is enabled in the TpaCAD Configuration, the Group field can assume value 0.

The X, Y, Z correctors of the group are displayed beside:





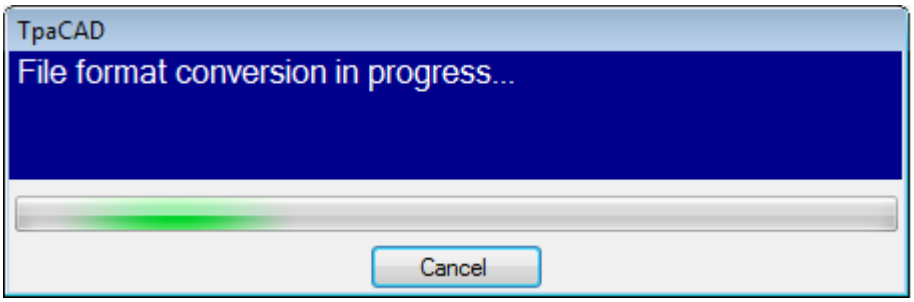
Display unit of measurement

Selection field of the representation unit of the parametric data. The selection can be between [mm] or [inch]. By opening the window, the unit of measurement of the active program appears.
The table provides the list of the tools arranged in accordance to the active selections (tooling, machine, group). Each row displays the significant information on a tool: face of working, diameter, typology, lengths, correctors, rotation speed, movement speed. The presentation order can be set for all columns.
The Technology window can be displayed also during the insertion of a working, by selecting the button  at the option **Tool**. After closing the window by double-clicking a row of the tool table, the current selection is brought again to the technological fields of the machine (machine, group, tool).

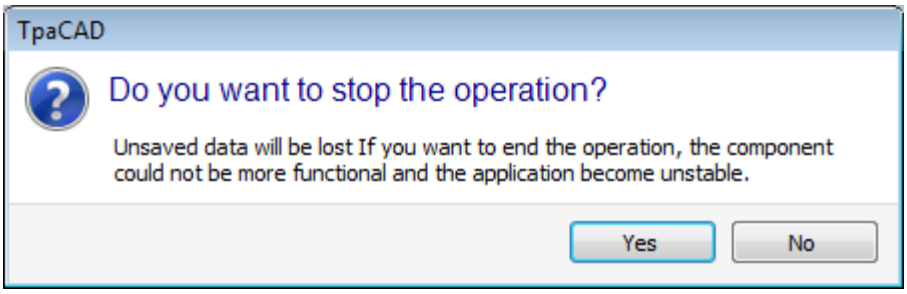
The Technology window can also be displayed to help the parametric programming. The closure of the window by double-clicking a cell of the tool table integrates the field in programming with the selection of the technological function corresponding to the technological selected fields (machine, group, tool, information typology).

4.17 Information on external components linked to TpaCAD

As already mentioned, some overall procedures concerning the programs are performed by procedures started in external components, linked to TpaCAD. That is: import from external format, export to an external format.
The performance of external procedure locks the normal TpaCAD execution, waiting until the same procedure stops.
After a few seconds from the start of the format conversion, if the procedure is not finished, a window appears showing the status



and can control the forced closing. In this case, a following window informs about the need to pay attention to such an operation.



The request to stop this procedure is carried on if you think that the running program does not answer any more; however, this is an extreme solution. According to the dimension of the program that is being processed, we suggest that you wait a few seconds before ending an operation that presumably requires only some time to be completed. In case you detect a real situation of anomalous functioning, you should break the procedure and report on the situation to the supplier of the system.

5 How to configure the graphic representation

5.1 Customize views

The commands to enable or disable the display of the visual elements in the area for the graphic representation of the panel, are partly available in the customization group in Customize -> Views -> Customize views. Let us see the menu selections in the **View on** tab below.



Profile direction: it enables or disables the display of the direction arrows on profile segments. This display is applied if the display of the tool diameter overall dimensions is excluded. For the marked profiles of construct, geometry or emptying, the application is set as in ***Customize -> Views -> Customize graphics***.



Points on profiles: it enables or disables the display of profile edge points (little circles). This display is applied if the display of the tool diameter overall dimensions is excluded. For the marked profiles of construct, geometry or emptying, the application is set as in ***Customize -> Views -> Customize graphics***.



Working coordinates: it enables or disables the display of added elements and coordinates concerning the current working. Complex or construct workings (subroutines, macro-programs) are excluded. For example, if the current working is an arc, the coordinates for the edge points of the arc, of the centre and the initial radius are displayed like a linear segment between the arc start point and the centre.



Overall view in 3D graphics: it enables or disables in bulk the display of the overall dimensions in the three-dimensional graphics (vertical dimensions), in accordance with the corresponding parameters in the dialog box opened by ***Customize -> Views -> Customize graphics***. This option is significant only in a three-dimensional representation. For the marked profiles of construct, geometry or emptying, the application is set as in ***Customize -> Views -> Customize graphics***.
The button applies the activations assigned from the menu managed by the button itself:



Show point and setup extent in 3D graphics



Show profile extent in 3D graphics

For the point workings, for example a drilling, the 3D overall dimension only appears if the depth of the piece (not over the piece) is given and it is the representation of a cylinder whose diameter is equal to the diameter set or taken from the technology.
As for the profiles, the actual view mode of the 3D overall dimensions is ruled by a further group of selections. More specifically:

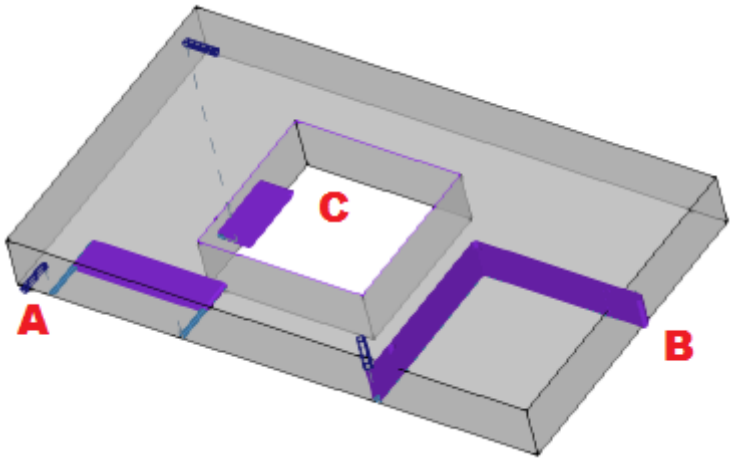


Limit the 3D overall dimensions: it enables or disables the evaluation of specific limit conditions (trim) of the 3D overall dimensions of the profiles. If this option is not selected, a 3D overall dimension corresponding to the tool length is represented; the only added evaluation concerns the face programmed depth:

- if over the piece: only the hatched segment appears, without the added overall dimensions (the actual representation is determined by its setting in ***Customize -> Views -> Customize graphics***)
- Otherwise, it shows the overall dimensions with the technological data of the tool applied (length and diameter).

In the following picture: three profiles are assigned to the piece:

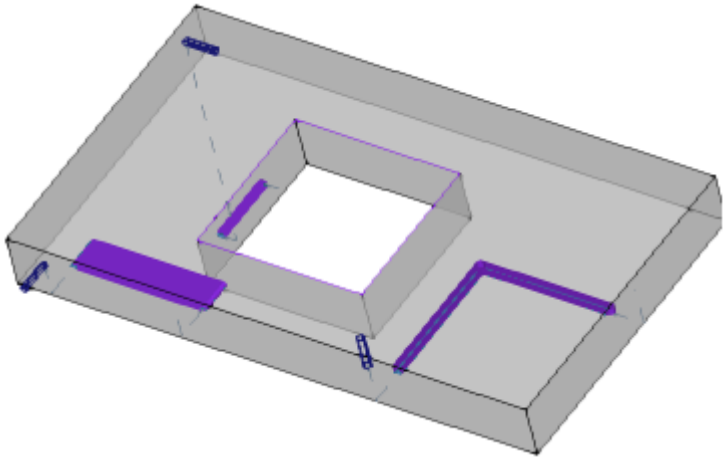
A. Oriented profile programmed in face 1, cutting a side face, tool oriented horizontally. The profile starts and ends outside the XY plane of face 1 and in a position over the piece (always in relation to face 1): the Z clearance segments are hatched, while the segments outside the XY area of the face are represented by a continuous line.
B. Vertical profile programmed on face 1. The profile starts and ends outside the XY plane of face 1, but in a position not over the piece: the whole profile appears with the overall dimensions applied;
C. Vertical profile programmed on fictive face (ex.: 7). The profile starts and ends outside the XY plane of face 7 and in a position over the piece: the segments going in/out from the piece are programmed in order to avoid collisions with the piece. The only Z clearance segments of the face are hatched.



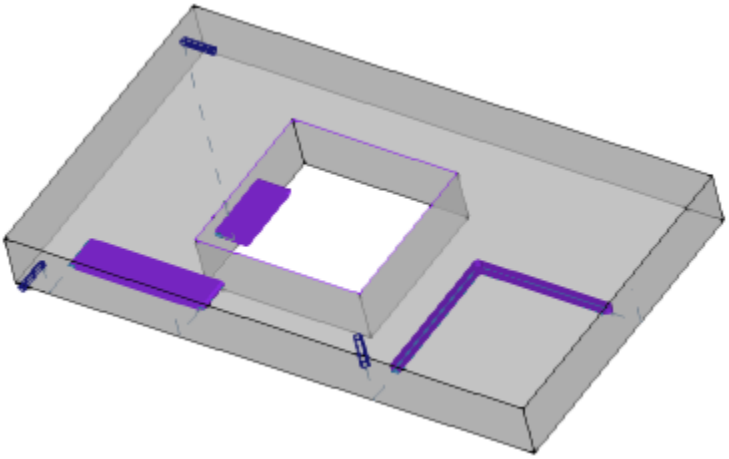
If this option is selected, the evaluation mode offers more aspects:

- first of all, for each segment of the profile the programmed depth is calculated, in the XY usable area of the face (within length and height of the face): Z clearance programmed segments or outside the XY area of the face are considered over the piece.
- the selection in the group of the three options is calculated:

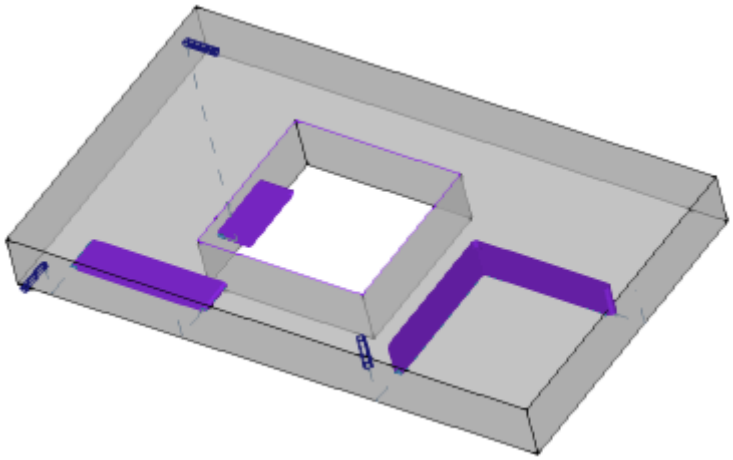
 **Trim the overall dimension at the plane of the face**: it shows the overall dimensions of the segments **not over the piece** only (see previous point) and with an extension of the vertical overall dimension that is limited in the XY plane of the face. The picture shows how the segments over the piece are now those outside the XY area of the face



 **Trim the overall dimension at the piece**: it shows the overall dimensions of the segments considered **not over the piece** only (see previous point) and with an extension of the vertical overall dimensions that is restricted to the intersection to the piece (original parallelepiped) and anyway not greater than the length of the tool. In case of curved face or surface, the vertical overall dimension interprets as: **Trim the overall dimension at the plane of the face**.



Trim the overall dimension at the length of the tool: it shows the overall dimensions for the segments considered **not over the piece** only (see previous point) and with an extension of the vertical overall dimension that is equal to the length of the tool. The picture shows the same overall dimension for all the segments not over the piece corresponding to the tool length of each profile.



Overall dimension of the profiles: it enables or disables the display of the overall dimension of the tool diameter on the profiles (horizontal dimensions), according to the selection in the area of **Customize View in tool compensation**. The option is applied to the original profiles (non-corrected profiles).

Entry and exit segments: this option enables and disables the display of the entry and exit segments in the original profiles (non-corrected profiles) and non-construct profiles. The entry and exit segments are always shown in the representation of construct profiles or with active correction.

Grid: it enables or disables the display of the step grid. As for grid settings, please refer to ***Customize -> Views -> Grids and patterns***.
ATTENTION: in any case the grid may be not displayed, if the current zoom does not allow the "distinction" of the element for the grid itself.

Special grid: it enables or disables the display of the special grid (the command may not be available). It is about a grid directly assigned for individual points and defined, during configuration, by the machine manufacturer. The grid activation is interpreted only in top or bottom face view.
ATTENTION: in any case the grid may be not displayed, if the current zoom does not allow the "distinction" of the element for the grid itself.

Cursor: it enables or disables the cross-cursor display identifying the active working. The cursor is centred on the working application point and it is displayed in 2D or in 3D, according to the active view. The cursor uses the colours of the three axes (RGB): the X axis is red (R=red), the Y axis is green (G=green), the Z axis is blue (B=blue). In the piece overall view, the cursor is displayed in sequence assignment. As for cursor settings, please refer to ***Customize -> Views -> Customize views***.

Working references: it enables or disables the display of the graphic item that indicates the reference set for the active working. This command is only available if an interpretation of the Field O as a

reference (corner or face side) is recognized. In the piece overall view, the reference is displayed in sequence assignment.

Face building: in case of fictive faces, where the programming of the three significant points of the face does not correspond to a system of three orthogonal axes, this option enables or disables the display of the construction between the programmed y axis (not perpendicular to the face x axis) and the calculated y axis (perpendicular to the face x axis).

Show all fictive faces: in face view, this option enables or disables the view of the fictive faces, excluding the current face. If the view is disabled (non-active selection), all the faces with variable geometry are excluded from the graphic representation and this exclusion concerns the workings applied to the faces itself. This command allows you to simplify the view in case of a program with many assigned planes.
If the current face is not the piece-face, the selection is also applied to the automatic faces assigned in piece-face.
The selection is ignored in the view of piece-face: all the faces assigned on the piece are displayed and you can exclude the representation of the workings programmed on the view of the single faces (see the command **Workings by other views**).

Workings by other views: in face view, this option enables or disables the display of the workings programmed on the other face views. This command allows you to simplify the display in case of a complicated program.

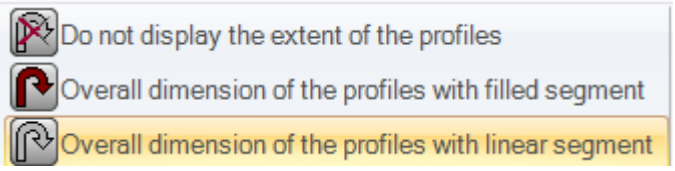
On the status bar, following commands are displayed:

Snap to grid: if enabled, this command limits the cursor movement for the active or the default grid vertices (step grid). Snap to grid influences the display of the mouse position on the status bar and affects:
- the acquisition of coordinates in some tools
- the direct application of geometric items.

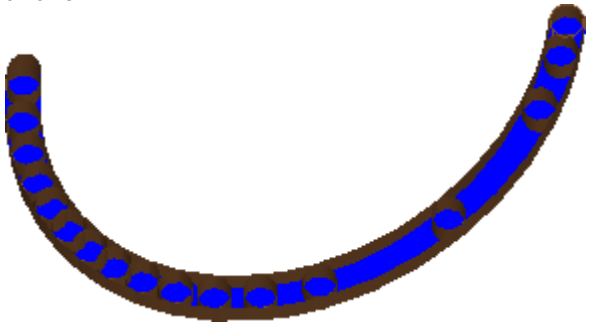
View program: if selected, it shows that the program display is active. The command is located in the status bar according to the TpaCAD configuration
At TpaCAD launch, the field is always active, apart from its status on application program exit. The command unit is designed to read very large programs allowing time management reduction. In case of field activation, a message requires a confirmation. In the same way, a confirmation is required also during a program reading, if the field is not selected.
If the field is not selected, interactive modes are also not available to assign the current working and tool application. In these cases, a message shows why it is not possible to perform the command.

5.2 Customizing View in Tool Compensation

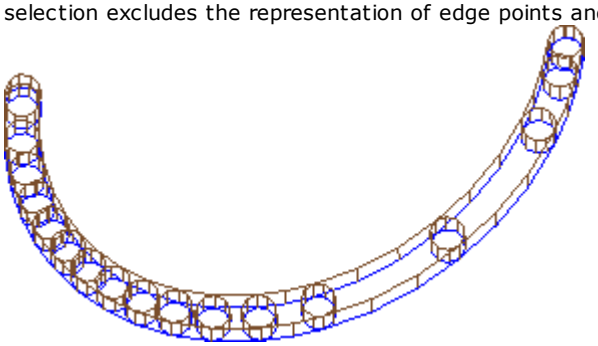
In the View on tab, let us now examine the group Customize View in Tool Compensation. The selections of this group are always applied when the view in Tool compensation is active and, when **Overall dimension of the profiles** is selected, also in the normal view.



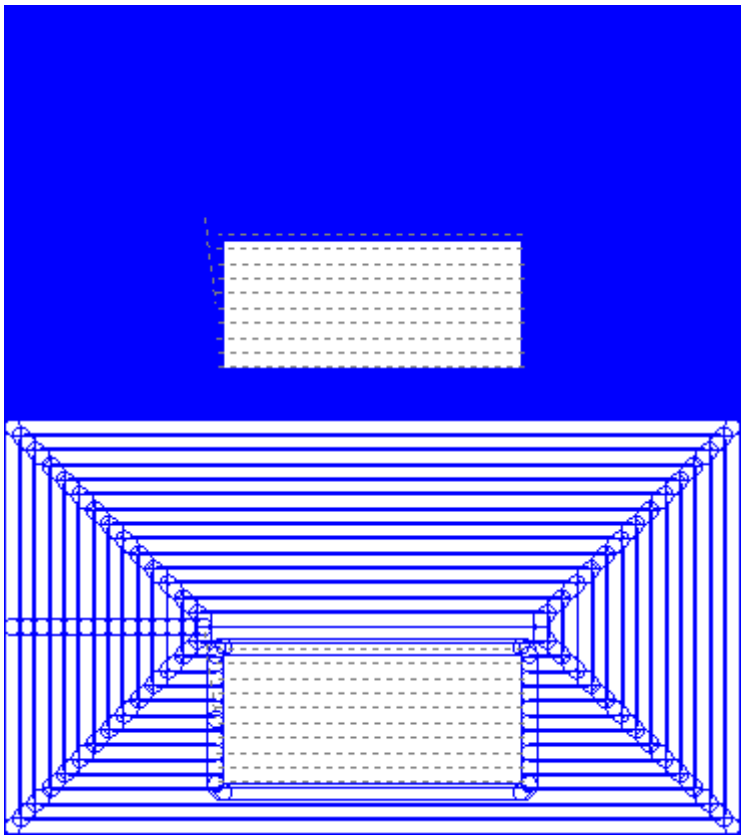
Do not display the extent of the profiles: this item represents all the profiles with unit thickness.
Overall dimension of the profiles with filled segment: the profiles are represented with a full segment whose thickness is equal to the tool extent. This selection excludes the representation of edge points and direction arrows.



Overall dimension of the profiles with linear segment: the profiles are represented with a thickness equal to the tool extent but with a non-full profile. In the figure the external contours of the extents are displayed. This selection excludes the representation of edge points and direction arrows.



The full segment view can be particularly useful in case of emptying profiles, in order to estimate the real removal of material. The figure makes the difference between the two representations clear.



However, the following items are represented with unit thickness:
- construct profiles
- profile segments over the piece (for the options, see: ***Customize -> Views -> Customize views***).



Original profiles in compensation: if enabled, the compensated profiles and the original ones (non-compensated profiles) are displayed. If not enabled, only the compensated profiles and the profiles that do not apply any compensation are displayed. This selection affects the tool compensation view.

5.3 Checking the view

Zoom and Pan commands allow you to enlarge, reduce, relocate what is displayed on the panel or on the face. Zoom and Pan only modify the dimension of the area represented within the view window. They are activated from the contextual menu recalled in the graphic view area by pressing the right mouse button.



Pan: it moves the panel with the mouse in the graphic area. After selecting the command a special cursor appears: hold down the left or the right mouse button and drag in the desired direction. If you

release the button, the command closes.
To directly activate the command through the mouse, hold down the right button and move in the desired direction.



Default view: it restores the piece view to 3D (three-dimensional) on the rotations corresponding to the default View.



Assign View by default: it assigns the current rotations of the piece (in 3D view) as default view. At the launch of the program, the default view is assigned and the 3D view is selected.



Extension Zoom: it scales the draw area so as to display the piece or the face in their full extent, in the maximum representation scale allowed. More specifically, the piece or the face view are centred with an additional visual margin. The command can be activated through the key function **[F6]**.



Window Zoom: it selects a rectangular window whose content will be enlarged, in the maximum representation scale allowed. After selecting the command, a special cursor appears: hold down the left mouse button and drag it until the desired window comes into sight. If you release the button, the command closes.
The command can be activated by the short-cut **[CTRL+W]**.



Previous Zoom: it redoes the previous view (with a storage up to 10 levels). The command can be activated by the short-cut **[CTRL+Shift+W]**.



Zoom All: it scales the draw area so as to display the piece and the workings in their full extent.



Zoom In-Out: it activates the command of dynamic variation of the representation scale. After selecting the command, a special cursor appears: hold down the left (or right) mouse button and drag upwards to increase the zoom, downwards to decrease the zoom
The command can be activated through the keyboard short-cuts **[CTRL+right mouse key]**.



Zoom In: the drawing representation is enlarged.



Zoom Out: the drawing representation is reduced.

Zoom In/Zoom Out using the mouse: this command is always active. It increases or decreases the current representation scale (zoom in or zoom out). To zoom in, scroll the mouse wheel up, to zoom out scroll the mouse wheel down.

The commands for the 3D rotation of the piece are activated from the keyboard or from the mouse.
Upward rotation: the piece rotates upward, with horizontal rotation axis. The rotation is activated from the keyboard by selecting the key **[X]**, and ends when the key is released.
Downward rotation: the piece rotates downward, with horizontal rotation axis. The rotation is activated from the keyboard by selecting the keyboard short-cuts **[Shift+X]**, and ends when the short-cuts are released.
Leftward rotation: the piece rotates leftward with vertical rotation axis. The rotation is activated from the keyboard by selecting the key **[Y]**, and ends when the key is released.
Rightward rotation: the piece rotates rightward with horizontal rotation axis. The rotation is activated from the keyboard by selecting the keyboard short-cuts **[Shift+Y]**, and ends when the short-cuts are released.
Clockwise rotation: the piece rotates clockwise on the view plane, with rotation axis perpendicular to the view. The rotation is activated from the keyboard by selecting the short-cut **[Z]**, and ends when the short-cut is released.
Counterclockwise rotation: the piece rotates counterclockwise on the view plane, with rotation axis perpendicular to the view. The rotation is activated from the keyboard by selecting the short-cuts **[Shift+Z]**, and ends when the short-cuts are released.
Rotation using the mouse: to rotate the piece, you need to hold down the left mouse button and move the cursors in the direction towards which the piece should be turned.

5.4 Three-dimensional representation

The commands to select the graphic display are grouped in the **View on** tab in the **Navigate** group.



3D View: it activates the three-dimensional representation. If the 3D View is active, the piece can be rotated on three planes that are assigned by activating the commands from keyboard or mouse, as described in the previous section.
This command can be activated from the contextual menu in the area of graphic view, by the commands of the tab **View on** or from keyboard through function key [F2].



Box View: it activates the bi-dimensional representation of the panel exploded view. The faces of the parallelepiped only are represented. The command can be activated from the contextual menu in the area of graphic view, by the commands of the tab **View on** or from keyboard through function key [F3]. If the box view is active, all rotation commands of the piece are disabled.



2D View: it activates the bi-dimensional representation of the selected face. This command can be activated from the contextual menu in the area of graphic view, by the commands of the tab **View on**

or from keyboard through function key [F4]. If the 2D view is active, all rotation commands of the piece are deactivated.

Below is a list of rotation commands for a defined plane (with 3D view).
All commands can be activated from the contextual menu called in the area of the graphic view (group: Navigate) or in the **View on** menu tab.


View from the top: the piece is placed in 3D view from the top face (face 1).


View from the bottom: the piece is placed in 3D view from the bottom face (face 2).


View from the front: the piece is placed in 3D view from the front-side face (face 3).


View from the back: the piece is placed in 3D view from the back-side face (face 5).


View from the right: the piece is placed in 3D view from the right-side face (face 4).


View from the left: the piece is placed in 3D view from the left-side face (face 6).


Face plane: the piece is placed in 3D view from the current face.


Redraw: it creates again the overall view display, with application of all currently assigned graphic settings (visual items, zoom, pan, special views and view filters). This command can be selected in the **View on** menu tab or through keyboard with the function key [F5].

5.5 Special Views and View filters

The commands to activate the special views and the view filters are grouped in the **View on** tab in the **Views** group.
The Special Views and the View filters are applied to the piece as a whole, even if they are activated in face view, and are all together in the representation resulting from their application.


Selections: it activates the view of only the selected workings.


Tool compensation: it enables or disables the view of the tool compensation. If the application procedure of the tool compensation has detected some errors:
- with selection from face view, the special view is not activated. With selection from overall view, the special view is activated only for the faces that have been verified as correct;
- in the Error area, it is possible to see the error situations.

The command can be activated by the function key **[F7]**.


Logical conditions: it enables or disables the application and the view of the logical conditions. If the selection is active, only the workings that are verified according to the programmed logical conditions are displayed. In particular:
- the construct workings can be totally excluded from the display, if the option in **Customize -> Views -> Customize views** is configured.
- the open profiles (without setup) can be totally excluded from the display, if set in the configuration by the machine manufacturer;
- if some exclusions are assigned, these last ones are applied and evaluated in the same way as the logical conditions.

If the application procedure of the logical conditions has detected some errors:
- with selection from face view: the special view is not activated. With selection from overall view: the special view is activated only for the faces that have been verified as correct
- in the Errors area, the error situations are displayed.

The command can be activated by the function key **[F8]**.


Layers: this option activates the display of the only workings that are assigned on Layers whose visibility status is active, as set by the command Piece -> Advanced assignments -> Layers. This command is not available, if the management of Layers is not enabled.


Special views: it assigns the visibility status of the workings according to their association with one or more significant assignments (properties, technology, ...), as set by the command Piece -> Advanced assignments -> Special filters. This command may not be available in the menu.

5.6 About profile

The **Info** group in the **View on** tab shows significant information about the current profile (in face view).
If the current working is not part of a profile: no field is compiled and no icon is shown as checked.



Length of profile: this option shows the 3D length of the current profile, including any enter/exit segments programmed in the setup.



Area: this command shows the area of the current profile, if it is closed. In evaluating the closed profile, any possible segments going in and out programmed on the setup are excluded.



It displays an image showing the movement of the tool for the current line. The information message (tooltip) that is displayed moving from the mouse cursor to the image describes the movement of the tool: descent to work coordinate, descending and rising movement over the piece, tool movement in the piece.



Apply multiple setups: this icon is checked, if the profile applies some multiple setups.



This profile is closed: the icon is checked, if the profile is geometrically closed in all the coordinates (XYZ). In the evaluation, possible entry/exit lines programmed on the setup are excluded.
In the case of profile with multiple setups, the evaluation concerns the first programmed setup.



Apply entry to profile: the icon is checked, if an entry line is programmed on the setup and it is correctly solved.
In the case of profile with multiple setups, the evaluation concerns the first programmed setup.



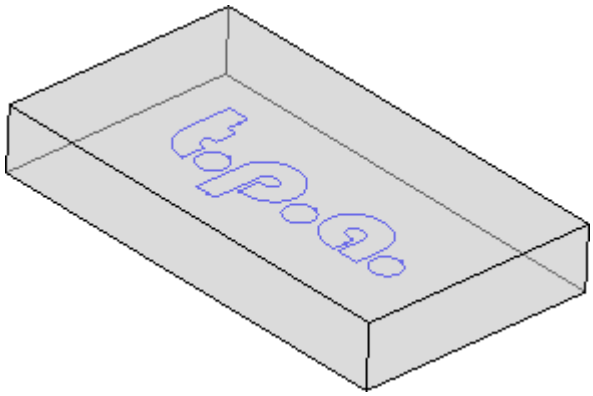
Apply exit to profile: the icon is checked, if an exit line is programmed on the setup and it is correctly solved.
In the case of profile with multiple setups, the evaluation concerns the first programmed setup.

6 Piece

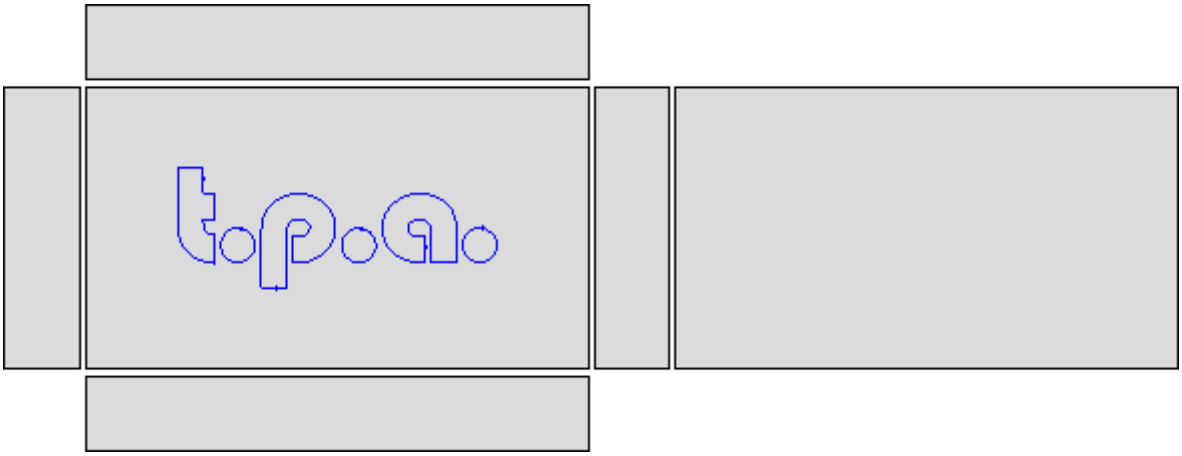
6.1 Graphic display of the Overall View

The piece is represented in three-dimensional view, with any fictive faces assigned also externally to the basic parallelepiped or in box view, without the fictive faces.

Representation of the piece in three-dimensional view
The workings are represented in the space so that the real overall dimensions are visible in all directions.



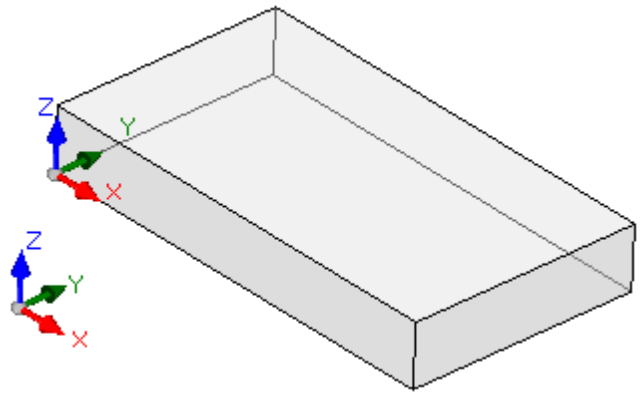
Representation of the piece in box view
In the piece, the faces of the basic parallelepiped are represented in exploded view. The workings are represented in the face plane on which they are applied.



6.2 Piece geometry

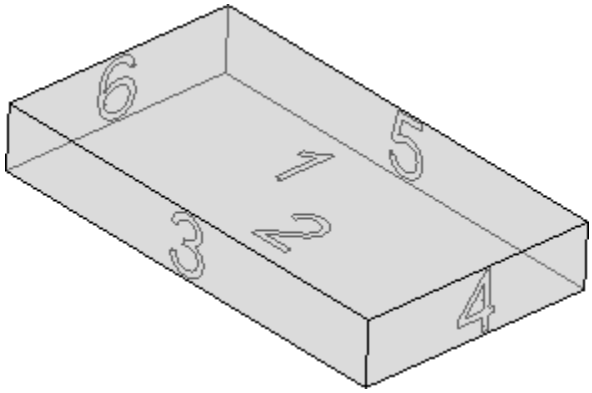
The piece is a parallelepiped object, consisting of:
- three dimensions: length, height and thickness. The three dimensions are indicated by the letters: **l**, **h**, **s**.
- six faces.

TpaCAD uses a three-dimensional system of fixed Cartesian coordinates, named **Absolute Reference System of the piece**, that is common to all pieces and is assigned as shown in the picture:



- the axes are indicated: X, Y and Z
- system origin is located on piece left bottom corner
- X axis is associated to piece length dimension (indicated by: l) and has a positive direction rightward
- Y axis is associated to piece height dimension (indicated by: h) and has a positive direction inward
- Z axis is associated to piece thickness dimension (indicated by: s) and has a positive direction upward.

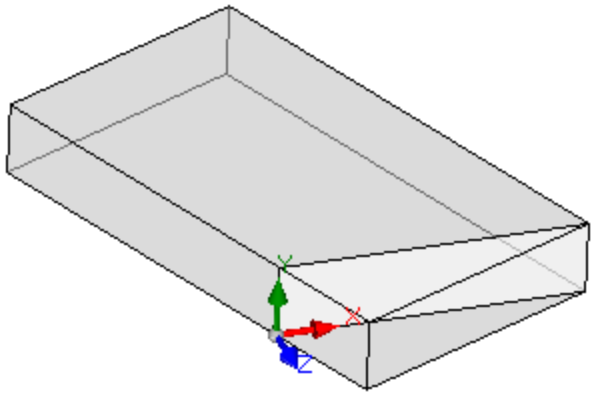
The six faces of the parallelepiped are now indicated as **real faces** and are numbered from 1 to 6. The figure shows the **automatic** numbering of the faces:



- top face is number 1
- bottom face is number 2
- front view face is number 3
- right side view face is number 4
- opposite face to the front view is number 5
- opposite face to the right-side view is number 6.

TpaCAD can be configured to operate with a numbering different from the automatic one: in this case we say that a **custom** numbering is assigned, anyway using the face numbers from 1 to 6.
TpaCAD can be configured not to manage one or more of the real faces.

In addition to the six real faces other faces can be assigned, generally located in the piece, from now named **fictive**.



Fictive faces are numbered from 7 to 99 and:

- they can be internal, partially or totally external to the piece

- they can be tilted in any way with respect to the absolute Cartesian coordinate system of the piece
- a fictive face can be assigned on a single (flat, curved) or composite element (surface)

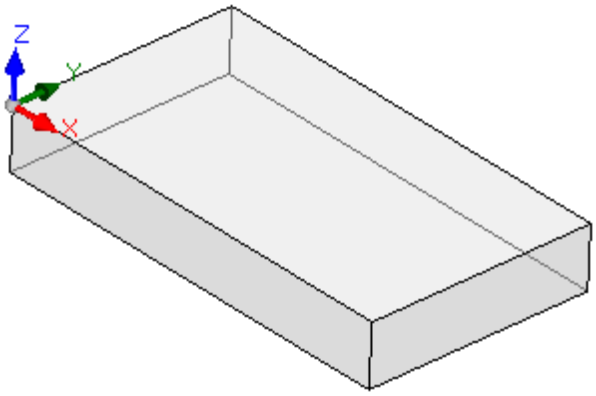
In particular operational functionalities (see programming in Piece-Face) additional faces can be assigned, generally located in the piece, from now on named **automatic**. Automatic faces are numbered from 101 to 500 and they can be only assigned to a single element (flat, curved).

In any case, a fictive (or automatic) face is assigned, the XY plane is always traceable back to a rectangular sheet.

The working programming in the piece is always referred to one face and uses the three-dimensional Cartesian coordinate system of the face. In this case also we talk about three XYZ axes, where:
- the face plane assigns the X and Y axes
- the direction perpendicular to the face plane assigns the Z axis, that we indicate as depth axis.

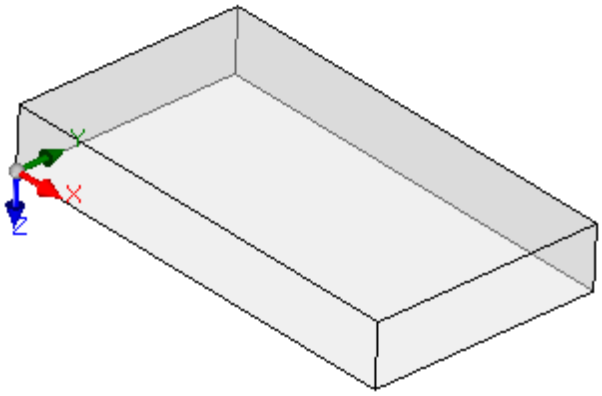
In the **Face reference system**:
- X axis is associated to the length dimension of the face (from now on indicated by: **lf**)
- Y axis is associated to the height dimension of the face (from now on indicated by: **hf**)
- Z axis is associated to the thickness dimension of the face (from now on indicated by: **sf**).

Let us examine now the Reference systems of the real faces, as automatically assigned:
Faces 1 and 2:



Face 1:

face dimensions
lf=l
hf=h
sf=s



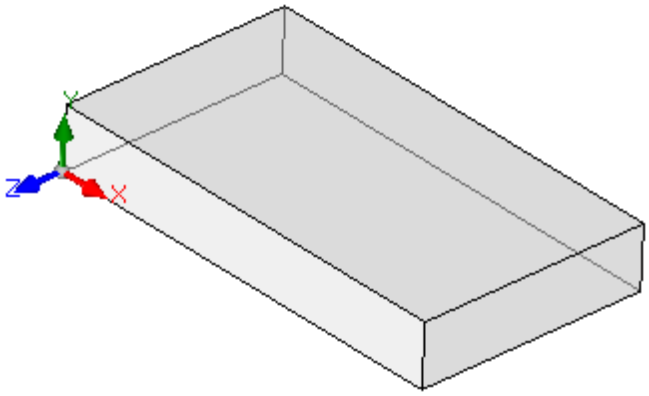
Face 2:

face dimensions:
lf=l
hf=h
sf=s

the local systems of the faces 1 and 2 are similar:
- X axis has the same orientation and direction as the X axis of the Absolute Reference System of the piece
- Y axis has the same orientation and direction as the Y axis of the Absolute Reference System of the piece
- Z axis has the same orientation as the Z axis of the Absolute Reference System of the piece, but in face 2 it has opposite direction.

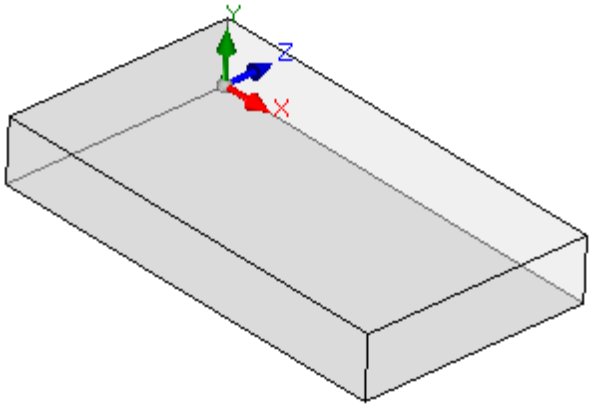
Compared with the Absolute Reference System of the piece, the origin point of the faces:
- in face 1 it is in (0; 0; s);
- in face 2 it is in (0; 0; 0).

Faces 3 and 5:



Face 3:

face dimensions
lf=l
hf=s
sf=h



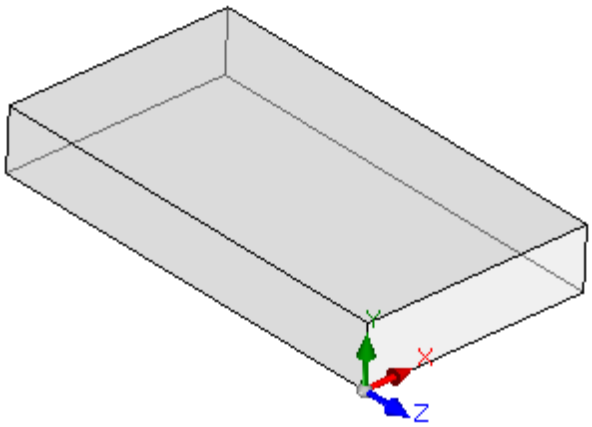
Face 5:

face dimensions
lf=l
hf=s
sf=h

The local systems of the faces are similar
- X axis has the same orientation and direction as the X axis of the Absolute Reference System of the piece
- Y axis has the orientation and direction as the Z axis of the Absolute Reference System of the piece
- Z axis has the same orientation as the Y axis of the Absolute Reference System of the piece, but in face 3 it has opposite direction.

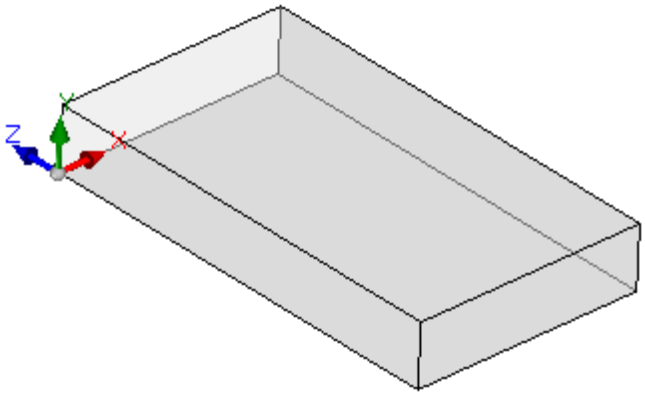
Compared with the Absolute Reference System of the piece, the origin point of the faces:
- in face 3 it is in (0; 0; 0)
- in face 5 it is in (0; h; 0)

Faces 4 and 6:



Face 4:

face dimensions
lf=h
hf=s
sf=l



Face 6:

face dimensions
lf=h
hf=s
sf=l

The local systems of the faces are similar:
- X axis has the orientation and direction as the Y axis of the Absolute Reference System of the piece
- Y axis has the orientation and direction as the Z axis of the Absolute Reference System of the piece
- Z axis has the same orientation as the X axis of the Absolute Reference System of the piece, but in face 5 it has opposite direction.

Compared with the Absolute Reference System of the piece, the origin point of the faces:
- in face 4 it is in (l; 0; 0);
- in face 6 it is in (0; 0; 0).

In each face, the point indicated as origin of the three axes corresponds to identically null assignments (value: 0.0) for the three point coordinates. Let us see how the coordinates of a generic point, referred to a face, change. Let us suppose face 1, with dimensions of the piece (l=100; h=800; s=20):
- the X coordinate of the point will be positive, when it moves rightwards from the origin, along the direction shown by the (red) arrow of the X axis, while negative values place the point in the opposite half-plane, on the left side of the Y axis of the face;
- the Y coordinate of the point will be positive, when it moves from the origin, along the direction shown by the (green) arrow of the Y axis, while negative values place the point in the opposite half-plane, lower than the X axis of the face;
- the Z coordinate of the point will be *generally* positive, when it moves upwards from the origin, along the direction shown by the (blue) arrow of the Z axis, while negative values place the point "below" the XY plane found by the face.

This means that a point that is located in the centre of the XY plane of the face, entering the piece at 10 mm from the face plane, has the following coordinates: X=500; Y=400; Z=-10. The same point, placed over the piece, will have Z coordinate with inverted sign: Z=10.
As far as the sign to be assigned to the Z coordinate is concerned we have used the term *generally*. The one described above is considered as the mostly used situation: negative depth values determine the space occupied by the tool in the piece, while positive values assign positions over the piece; this logic is applied to all faces of the piece (real and not).
However, it is possible to operate with the convention that is exactly opposite that described, if defined in TpaCAD Configuration.
Later in this manual the operations will be performed with the above-mentioned convention.

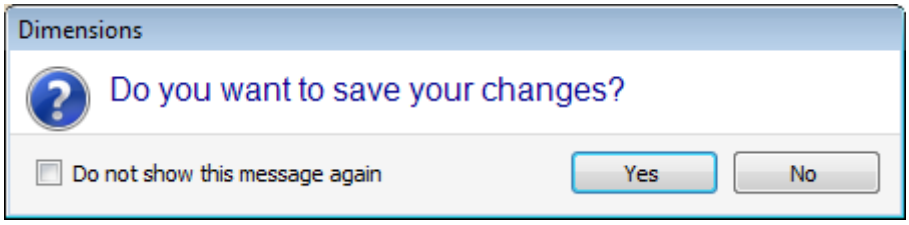
Local systems of piece real faces can be locally assigned to TpaCAD configuration in a different way, by moving front XY plane origin on a different corner and/or rotating orientation of X and Y axes. In this case we say that a **custom** piece geometry has been assigned.

From TpaCAD configuration you can opt for a work "plane" geometry, where a programming assigned on one or more planes with x, y coordinates of working is considered significant. A depth as z coordinate, but not a dimension in z, is assigned. In this case we say that a piece geometry is assigned in the **Absolute system**:
- the piece assigns the two dimensions of Length and Height according to which the piece is represented; anyway, a minimum thickness is assigned to the piece, but in automatic way;
- the piece base programming is limited to face 1 and the piece is represented by a rectangle;
- anyway, it is possible to assign further work planes, like fictive or automatic faces.

6.3 Assignments

Area of assignments

The area of assignments is arranged on more pages and it is always visible. Each page shows and sets a group of piece assignments.
The editing mode is directly active in general view. In face view this mode is activated by selecting the option **Edit** from the contextual menu or by double clicking the title of the Assignment area. To confirm or cancel the changes, select respectively the options **Apply** and **Cancel** from the contextual menu or press the **[Enter]** key. In this last case, the confirmation dialog box may appear.

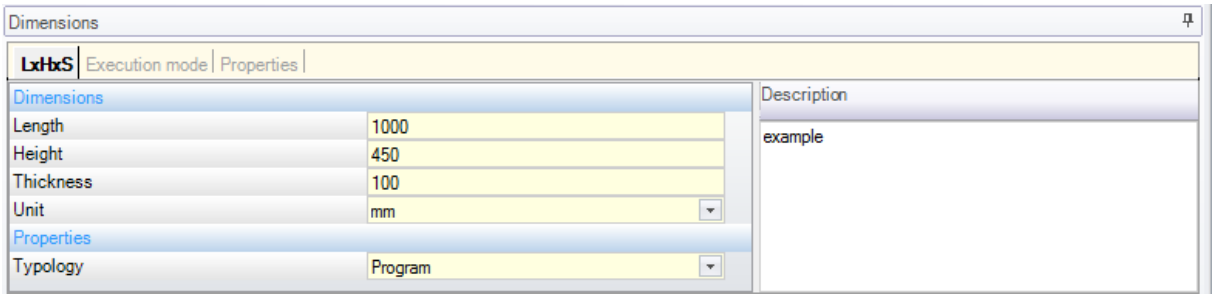


Select **[Yes]** to apply the changes, **[No]** to keep them pending.
Select the option **Do not show this message again** and confirm the dialog box with **[Yes]** to change the confirmation setting in TpaCAD Customization to **Automatic confirmation**.
If you select a command from the menu or select a face view and there are unsaved changes, these same changes can be saved and applied automatically on confirmation or can be trashed, according to the settings in TpaCAD Customization.

Dimensions, Execution modes and Properties

LxHxS

General information are assigned, such as dimensions and unit of measurement, piece typology, access level and comment.



- **Length, Height, Thickness**: piece dimensions. The three fields accept strictly positive numerical settings (> 0.0), containing maximum 20 characters. The piece dimensions can be used symbolically to assign variables or working parameters; the symbolic names of the dimensions are respectively: l, h, s. (**See chapter Parametric Programming**). The piece dimensions can be reassigned during the program execution, even if the settings stored in the original program remain unchanged. The **Thickness** of the piece can be not displayed, as from TpaCAD configuration.
- **Unit**: unit of measurement of the piece ([mm] or [inch]). (The field may be set as not editable in TpaCAD configuration)
- **Typology** of the program. The typology can be: program, subroutine and macro, can be changed. The macro typology is only shown if the access level is equal or higher than manufacturer. The subroutine typology may be not suggested, if its creation in TpaCAD configuration is only allowed from the non-minimum access level.
- **Opening and change level**: they assign respectively the minimum access level to open and record a program. Levels higher than the user's current access level cannot be set and the change level is assigned at least equal to the access level. If the current access level corresponds to the Operator level, the entries are not displayed.
- **Description**: it is a text given as a comment to the program. The maximum length of the text is 500 characters.

The area of **errors** shows the full list of messages (errors or warnings) detected during the program process.

Execution mode

The default program execution modes are assigned.



This is an optional page.

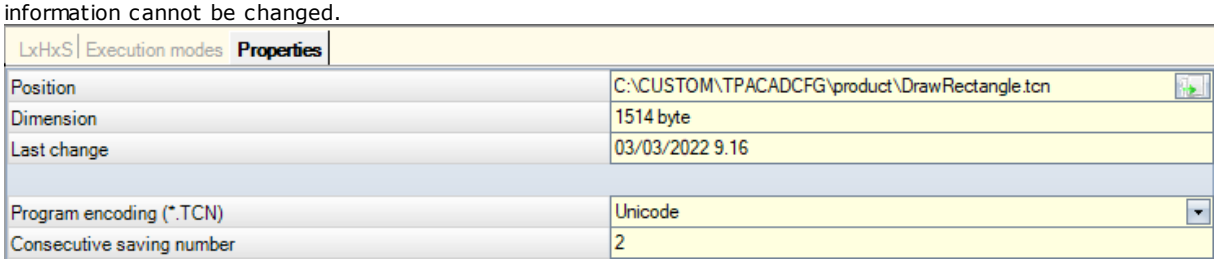
- **Work area**: it assigns an identification number of the work area. This is a custom parameter; therefore it takes a specific meaning for each application. using the parametric programming the parameter corresponds to the prarea variable arguments.
- **Execution**: program execution mode. Listed items:
  - Normal
  - Mirror x
  - Mirror y
  - Mirror xy

  (The actual number of items in the list can be reduced according to the configuration of TpaCAD).
  Using the parameter programming the parameters correspond to the prgn, prgx, prgy, prgxy variable arguments.
- **Locator offsets in work area**: they assign the stroke position of the selected work area with respect to machine home position. Using the parameter programming the parameters correspond to the prqx, prqy, prqz variable arguments.

Optionally, the execution modes can be assigned from the plant technology by a wizard selection: in a list of items select the work area. Then, according to this selection, the kind of execution and the step position can be automatically updated by reading them from the technological configuration.
The piece execution modes can be reassigned during the program execution, even if the settings stored in the original program remain unchanged.

Properties

This page displays some standard file properties and other customized properties of the program TpaCAD. The information cannot be changed.



Standard properties

- **Position**: it shows the file's full path. The field is empty if the program is new. The icon, right to the field, runs the command that copies the file path in the Clipboard.
- **Original file position**: it shows the full path of the original file in case of conversion in reading format.
- **Dimension**: it shows the size of the file in bytes, as reported at the time of reading (0 if the program is new)
- **Last change**: it shows the date of the last change of the file, as reported at the time of reading (the field is empty if the program is new)

Customized properties

- **Program encoding (*.TCN)**: it shows the codification assigned to the program (ANSI or Unicode). The field can be modified. If the program is new, the suggested setting is Unicode, otherwise it corresponds to the file codification as read. If the selection is changed from Unicode to ANSI, and the program uses Unicode settings, a message indicates the loss of information in case the program needed to be saved in ANSI format;
- **Consecutive saving number**: it shows a numerical value, automatically increased at each file saving in TpaCAD environment;
- **Read-only attributes**: if visible, it follows the list of program information whose change is locked and that may concern: dimensions (LxHxS), execution mode, variables (<o>, <v>, <r>), fictive faces, custom sections, sequences. A situation of modification lock can result from the external generation of programs in TpaCAD format (for example by the import from an external format) or from the reading of a file already saved. A lock situation can always be changed at manufacturer access level.

In case of change locked section, the picture of a padlock marks this status .
Particularly, the lock can clearly concern the Dimension page and/or that of the Execution mode. The description of the program can still be changed, even if the Dimensions are not editable.

- **No visualization if read-only**: select to make the sections in read only invisible. This is a selection applied to all locked sections, except for the Dimension page, that always stays visible.

"o" Variables

This is an optional page.
When configuring TpaCAD the maximum managed number of "o" variables is defined, between 0 (unmanaged section) and 16. They are numerical variables, whose meaning is normally, but not necessarily, unique for all programs.
The page arranges the "o" variables in a table: each row corresponds to a variable.

<o> - variables

	Value	Name	[..]	Edit
o0: X Offset	500	ofx	[mm]	l/2
o1: Y Offset	0	ofy	[mm]	0
o2: Z Offset	0	ofz	[mm]	0
o3	0			0
o4	0			0
o5	0			0
o6	0			0
o7	0			0

- **Heading**: it shows the name (o0, …, o15) and custom title of the variable (Example: X Offset).
- **Value**: it shows the value resulting from the solution of the expression defined in the **Edit** column. The field cannot be modified.
- **Name**: it shows the symbolic name associated to the variable that can be used in parametric programming. The field cannot be modified, and it is assigned during TpaCAD configuration. As in the figure, "ofx" is the symbolic name of the o0 variable: the variable can be called as "o0" or "o\ofx". The column is not displayed, if none of the "o" variables has an assigned symbolic literal name.
- **[..]**: it shows the unit of measurement of the variable:
  - if the variable defines a coordinate, the unit of measurement is expressed in [mm] or in [in]
  - if the variable defines a speed, the unit of measurement can be expressed in [m/min] or [mm/min] or [in/s] or [in/min], according to TpaCAD configuration
  - if the variable is dimensionless, it is not assigned.

The field cannot be modified. The column is not displayed, if none of the "o" variables has an assigned dimension.

- **Edit**: it is the variable assignment field. The field can be modified and can assign a number, a parametric or numerical expression. The maximum length of the field is 100 characters. In the figure, it is possible to see an example of parametric expression: the o0 variable is set ="l/2", where l indicates the length of the piece. The value calculated for the expression is 300, as shown in the **Value** column. An example of numeric expression is ="500/2", which returns value 250. The modification of the active field begins when any alphanumeric key or the F2 function key is pressed.
- **Description**: it displays the descriptive text of the variable, that, for example, can provide guidance on the meaning of the variable. The description is defined during the configuration and cannot be changed. The column does not appear when no description is assigned.

The setting of an "o" variable can be parametrized on the piece dimensions (l, h, s) and the Execution modes, while it cannot use other variables ('v', 'r' and the same 'o' variables either) or assignments of variable geometries or custom sections.
The 'o' variables of the piece can be assigned again while running the program, even if the settings stored in the original program remain unchanged.

The commands to modify the variables are contained in the context menu, that can be called when you click the right mouse button on the area for the variable window

Import from file: it imports the assignments of the variables from a selected program.

Copy: it copies the settings of the selected variables (the current variable, if there are no selected rows) into the local Clipboard. The copied variables are available to later Paste into the same program or in another one.
To commute the selection of a variable, click the header cell of the corresponding row holding down the key **[Ctrl]**. To remove the selection of the whole list, click on any position of the table.

Paste: it pastes the settings of the variables previously copied in the local Clipboard, considering the names of the variables as follows: 'o0' assigns the 'o0' variable, 'o13' assigns the 'o13' variable. The command is enabled only if a copy of one or more "o" variables is available in the local Clipboard.

Delete: it resets the setting of the selected variables.

Delete all: it resets the setting of all variables.

The error area shows the list of only the errors occurring during the assignment of the "o" variables. An invalid setting is also reported on the individual variable, as in the figure below:

<o> - variables

	Value	Name	[..]	Edit
o0: X Offset	225	ofx	[mm]	l/2
o1: Y Offset	0	ofy	[mm]	h-
o2: Z Offset	0	ofz	[mm]	0

While programming, you can recall an immediate help for those functions and variables available for the parametric programming.

"v" Variables

This is an optional page.
When configuring TpaCAD the maximum managed number of "v" variables is defined, between 0 (unmanaged section) and 16. They are numerical variables, whose meaning is normally, but not necessarily, unique for all programs.
The page arranges the "v" variables in a table: each row corresponds to a variable.

<v> - variables

	Value	Edit	Description
v0	1	1	
v1	2	2	
v2	3	3	
v3: Optimize	4	4	0=x increasing; 1=x descendi...
v4	0	0	
v5	0	0	
v6	0	0	
v7	0	0	

The page is quite similar to that of the "o" variables, to which reference is made.

"r" Variables

The table always lists 300 "r" variables. They are numerical or text variables, whose meaning differentiates for each program.
The page arranges the "r" variables in a table: each row corresponds to a variable.

<r> - variables

	Value	Name	Type	Edit	Description
r0	40	radius	[double]	40	
r1	0		[double]		
r2	0		[double]		
r3	0		[double]		
r4	0		[double]		

- **Header**: (r0, ..., r299) it presents the name of the variable.

- **Value**: it shows the value resulting from the solution of the expression defined in the Edit column. In the case of a *string* variable (see the field: **Type**) the resulting value is shown between double quotation marks. The field cannot be modified.
- **Name**: it shows the symbolic name associated to the variable that can be used in Parametric Programming. The field syntax has a maximum length of 16 alphanumeric lower-case characters. A name is not accepted if already assigned to another r variable. As in the figure, "radius" is the symbolic name of the r0 variable: the variable can be called as "r0" or "r\radius".
-  : it enables or disables the possibility to reassign a variable externally. A "r" variable can be reassigned during the execution of the program or when the program is used as a subroutine. We assume for example that r0 assigns a variable coordinate for the positioning of a drilling working:
  - when the program execution is called, it will be possible to change the value of r0 from an external menu
  - when the subroutine is called in another program, it is possible to change the value of r0 directly during the programming.

  A variable that cannot be reassigned is used for program definition auxiliary settings. The variables that cannot be reassigned normally use those that can be reassigned (for tests, assignments). It can be stated that a re-assignable variable is public, while a variable which cannot be reassigned is local (or private). The selection is disabled in any case if the value of the variable is a parametric expression that uses other "r" variables.
- **Type:** it assigns the type of the variable. Two numerical types (Double, Integer) and a non-numerical type (String) are managed.
- **Edit**: it is the field where the value of the variable is assigned. The field can be modified and may assign a number, a numerical expression or a parametric expression. The maximum length of the field is 100 characters.
- **Description**: it is a text field that can be assigned as a comment to the variable.

The modification of an active editable field begins when any alphanumeric key or the F2 key is pressed.

The "r" variable setting can be parametrized on:
- piece dimensions (l, h, s),
- "o" and "v" variables (o0 - o15, v0 - v15)
- earlier "r" variables in the list (example: r15 can use the r variables from r0 to r14).

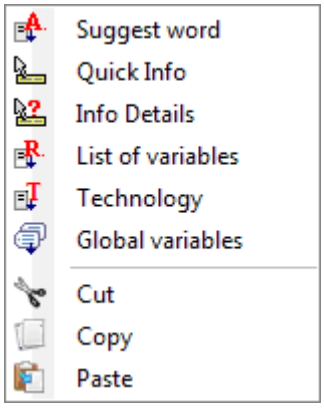
For a more detailed analysis of the possible variable parametrization, see the chapter on Parametric programming.
An unset variable (empty Edit field) has Value = 0.0 and Type = Double and cannot be reassigned.
The type of a r variable is not fixed, but can be:
- **Double**: numerical type; the calculated value (after all the parameters used for the setting) keeps the decimal part. Examples: working positions, movement speed.
- **Integer**: numerical type, similar to the previous case, but the calculated value resets the decimal part. Examples: counters, selection of operation, rotation speed.
- **String**: non-numerical type. Examples: the name of a subroutine, a text. The calculated value is also of String type.

The available commands to modify the variables are contained in the contextual menu that can be called when you click the right mouse button on the area of the window of variables: the use of the commands is totally similar to that of the "o" variables, to which reference is made.

The error area shows the list of only the errors occurring during the assignment of the "r" variables.
As already mentioned for the "o" variables, an invalid setting is reported even for single variables.

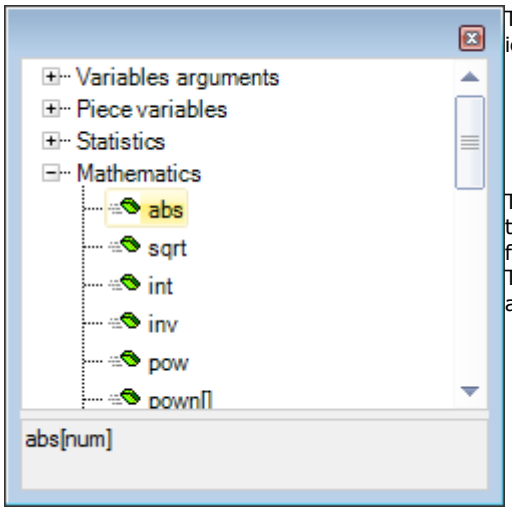
Edit wizard

While programming a field, you can always recall an immediate help menu of functions, variable arguments and available variables for the parametric programming, in addition to the usual change commands for an edit field (Cut, Copy, Paste). Also in this case, this is a context menu, that opens by clicking with the right mouse button the edit field of the variable. This is the full menu:

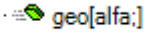


The help menu of the editor wizard can be recalled for: variables (<o>, <v>, <r>), custom section fields, settings of the fictive faces or the assignment of the workings.

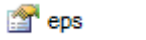
- **Suggest word**: it opens a menu where all functions and arguments of parametric programming grouped in nodes are available. The nodes shown, as well as the composition of each node, depend on more factors:
  - the context where the menu is open (variable assignment, variable geometry or working parameter);
  - type of the field to be assigned: numeric or string;
  - TpaCAD configuration.



The entries in the list are of two typologies, marked by different icons:



: function



: variable argument

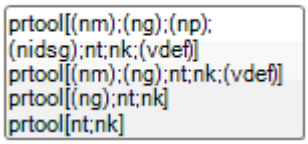
To select an entry: open a node and select the line of interest. In the case of a function, in the lower zone of the window the formats recognized when recalling the function itself are given. To recall the help page describing the function or the selected argument, press the key **[F1]** or the window button **?**.

Confirm the selection by double-click or [ENTER]: the selected entry is inserted in the edit field at the position of the cursor. In the example in figure:

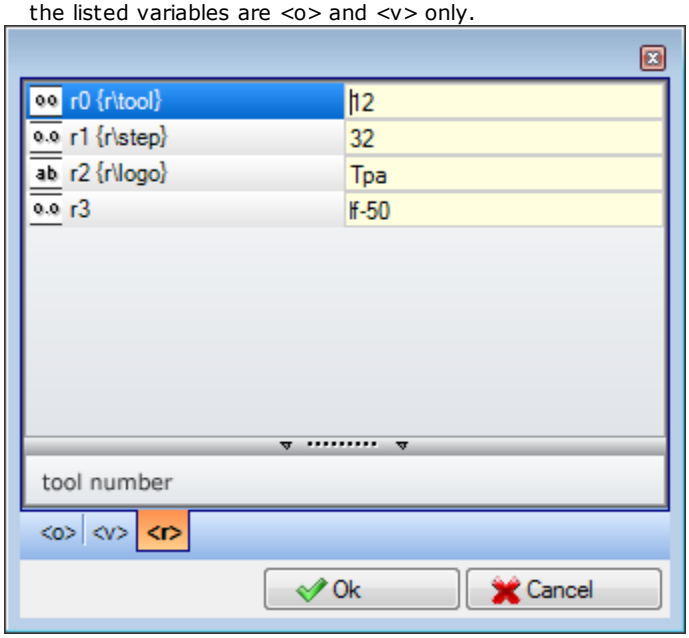
- the confirmation for the *abs* function requires inserting the string "abs" (the format of the function does not obligatorily require the use of square brackets);
- the confirmation for the *pown* function requires inserting the "pown[]" string (the use of the square brackets is obligatory).

- **Quick Info**: it opens a help box (tooltip) concerning how to use the function, where the cursor in the edit field is positioned. The entry does not appear, if the edit field is empty or if the cursor is not positioned on a valid name. The tooltip display is automatically cancelled after a few seconds. The figure illustrates the case of a function managing more formats:

prtool[]



- **Info Details**: it opens the help page of the function where the cursor in the edit field is positioned.
- **List of variables**: it opens a window containing the list of program variables. In case of <r> variable settings, the listed variables are <o> and <v> only.



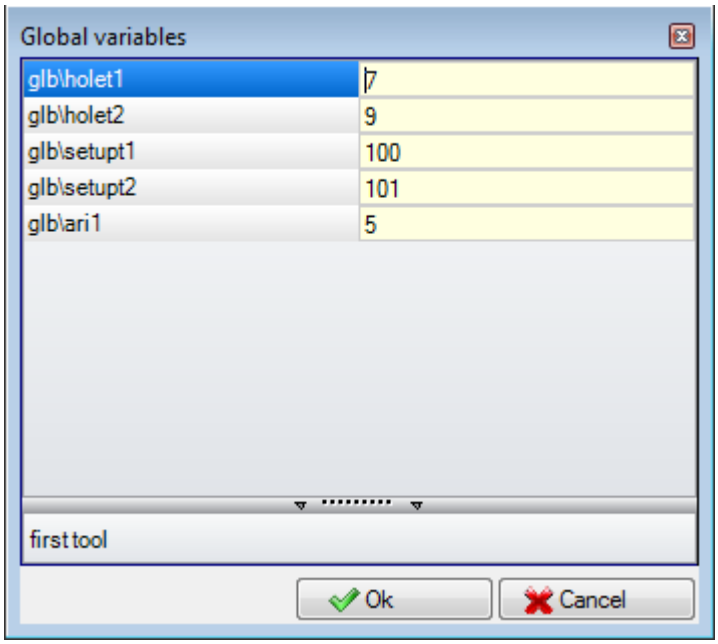
In the figure, the page that displays the <r> variables appears. Each compiled row corresponds to an assigned variable:

- the icon of the first column of each row shows the variable type. In the figure: r0 is of integer type 00, r1 and r3 are of double type 0.0, r2 is of string type ab.
- Then, the automatic name of the variable (r0, r1, …) is given and, if assigned, the symbolic name ("r\tool", …) appears in braces
- in the last column, the value of the variable appears.

If you select a row, in the area under the list, the description of the variable, if assigned, is given. In the figure, the description of the first row is "tool number".

- **Technology**: it opens the technology window. The entry does not appear if no technology is read, or if the technology control does not manage the procedure required, or when a variable or a working parameter of string type is assigned. Select a piece of information (cell in the table or something else) and confirm to insert the call to a technological function in the edit field at the cursor position.

- **List of global variables**: it opens a window containing the list of *Global variables* for TpaCAD environment. This option does not appear, if the management of the variables is not enabled or if no variable is assigned.



The table can show no more than 300 numeric variables, that can be recalled by symbolic name only, with the following format: "glb\xxx" with "xxx" = symbolic name. The variables are assigned in TpaCAD configuration and can be used for each setting.

Recovering "r" variables from an existing program

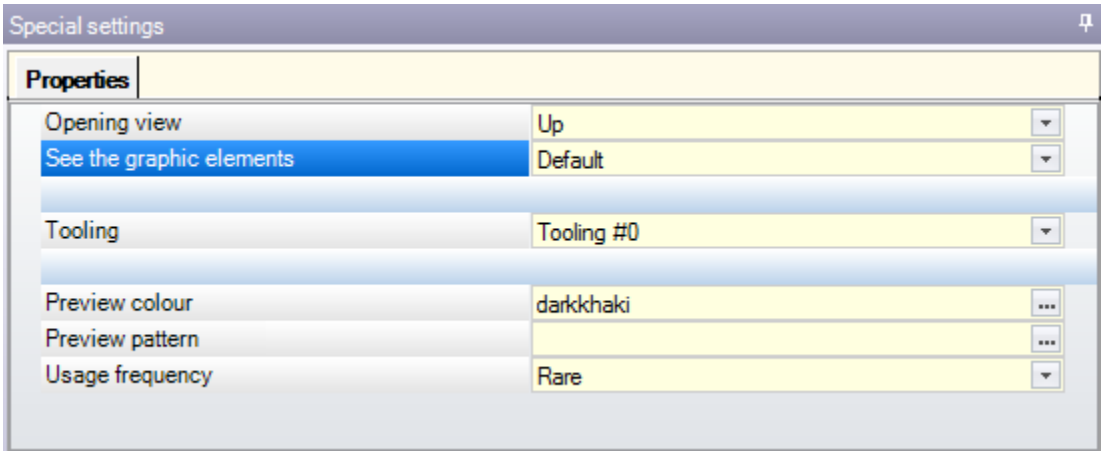
Using the command **Import from file** of the contextual menu, the entire list of "r" variables of another program can be imported.

If it is necessary for some of the variables of a program to be recovered, commands of **Copy** and **Paste** must be used, according to the following procedure:

- open the program, from which the variables must be copied;
- open the page of the "r" variables and select the variables to import. For instance, from r5 to r9;
- select the command **Copy**, from the contextual menu, to copy the selected variables into the local Clipboard;
- open the program, in which the variables must be imported;
- open the page of the "r" variables;
- select the command **Paste**, from the contextual menu to paste the variables previously copied. More particularly, the variables from r5 to r9 are overwritten, according to the example.

Special settings

This is an optional page.
This section includes some significant information, for which TpaCAD must activate particular authentication measures and procedures, in addition to exclusively custom information.
Custom assignments have a meaning which is unknown to the application and are configured by the machine manufacturer in customizing the program.
Even the section title (here: Special settings) may be different, since it can be reassigned at custom level.



You can define up to ten tabs, where the needed fields can be grouped:
- nine tabs organizing the information in the list, as in the figure. Each field has a type of edit (direct, selection from the list, ...) and of format (integer, double, string) assigned;
- optionally, a tab (always in last position) organizing the information in a table: each column with its type of edit and format, where you can duplicate the information on a specific number of rows.

A single item (information) is assigned in a field of specific typology:
- direct edit of numerical value, double type (example: "100.5")
- direct edit of numerical value, integer type (example: "12")
- direct edit of numerical value which can be assigned in parametric form (example: "l-100")
- selection from a list
- list of values to be sorted
- colour selected from a palette of colours
- direct edit of a generic string
- file search (the file open window is displayed)
- folder selection (folder selection window is displayed)

For each item, a Help text can be assigned, displayed in the area below the list. For any further information about the meaning of each item provided in the window, contact the machine manufacturer.

The available commands to modify the section are listed in a contextual menu that can be opened by clicking the right mouse button on the area of the window of Special Settings. In particular:

 it imports the section assignments from a chosen program

 it resets the section to the default settings

 it resets only the current page of the section to the default settings.

The default settings are read by the prototype program.

In Special Settings some items with a significant meaning can be displayed:
- Grain direction: it sets the grain of the panel
- Edges (left, right, top, bottom): it sets the edges of the panel, differentiated by side
- Tooling: it selects the technological outfit in numerical or string format (pathname or file name);
- Patterns and Colour of the panel;
- Display of the Graphic elements: the field allows the deactivation of the representation of arrows, edge points and 3D overall dimensions, with uploading of the program. For example, this option is useful for particularly large programs or ISO curves;
- Preview view: it sets the view to be assigned when the program is launched, with selection among the 6 view faces of the base piece.

The error area shows the list of only the errors that occur during the assignment of the section. An invalid setting is reported also in the corresponding field by means of the image  .

An error report in the context of a custom section can correspond to fields like the following:
- double numerical value (example: "100.5")
- integer numerical value (example: "12")
- numerical value assignable in parametric form (example: "l-100").

In the case of parametric field: all the errors generated by a wrong parametric programming may occur.
A parametric form can use:
- piece dimensions
- "o" and "v" variables
- "r" variables.

An error can also correspond to warnings due to invalid value, set or calculated. The valid values are set in TpaCAD configuration as assigned interval:
- between a minimum and a maximum; or,
- greater than a minimum value; or,
- less than a maximum value.

These messages correspond to warnings, moreover only detected at TpaCAD level, or to errors, as assigned during the configuration phase.

Additional info

This is an optional page .
They are included in the information field of exclusively custom type. The assignments are shown to be configured in dedicated functionality and their meaning is unknown to the application.
What previously presented about the custom information of the Special Settings remains valid.

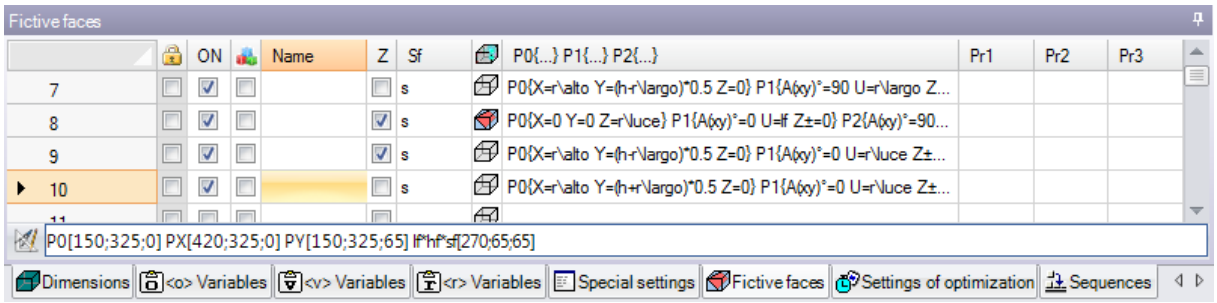
Modelling

This is an optional page.
For the description of the associated capability, please read a specific documentation that can be recalled through the short-cut F1 or a command available in the .

Fictive faces

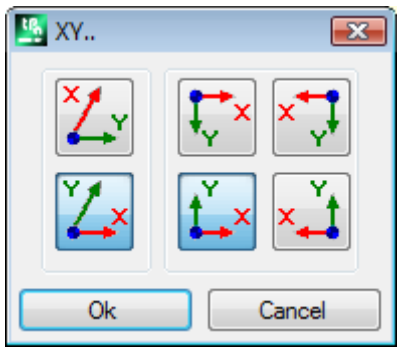
This is an optional page.
A fictive face can be assigned to represent pieces with a complex shape, such as, for example, the cavity for the glasspane in a door, or to aid in programming on planes with any orientation whatsoever. The fictive faces are progressively numbered from 7 to 99.

The page arranges the fictive face in a table: each row corresponds to a face.

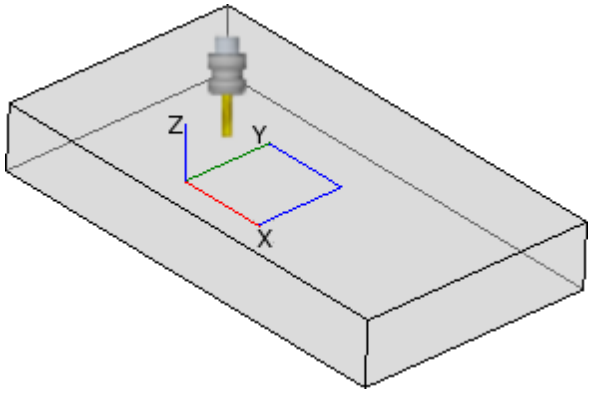


The fictive face page suggests the full list of faces that can be assigned from face 7 to face 99, since it is not necessary to define them sequentially.
The fictive faces assigned in the window are graphically displayed in general view without the programmed workings. Exiting the fictive face setting window, the graphic representation of the program is updated.
Below is the data that can be assigned for a face:
- **Heading** (Example: "7"): shows the face number.
- **ON**: if selected, it enables the assignment of the face.
- : if selected, it means that the face cannot be deleted because it has some programmed workings (the field is automatically set).
- : if selected, it enables the use of the face only as auxiliary to construct other fictive faces. An auxiliary construction face cannot be programmed, nor be considered in the piece geometry. The column shows only if the assignment of the reference face is enabled (see later on).
- **Name**: shows the name of the face.

- **XY**: it opens a window to set the mode of representation of the face XY plane in 2D view: horizontal or vertical X axis and which direction of each of the two-coordinate axes. The column appears only if enabled.



- **Z**: it selects the direction of the z axis over the piece in relation to the xy plane of the piece. **Z** is the depth axis and it is perpendicular to the xy plane of the face. If the field is selected, a xyz left-handed coordinate system is assigned, otherwise a right-handed one. The here assigned direction shows how the tool works (see figure):



  - If the Z positive axis is directed upwards, the tool enters the piece from above: the coordinate system is right-handed (it follows the right-hand rule, with: x axis on the thumb, y axis on the forefinger, z axis on the middle finger).
  - If the Z axis is opposite to what is shown in the figure (positive downwards) the tool enters the piece from below: the coordinate system is now left-handed (it follows the left-hand rule, with: x axis on the thumb, y axis on the forefinger, z axis on the middle finger).

Using the parametric programming, the axis orientation is returned by the geometric library multi-purpose function geo[zface; nside]

- **Sf**: it sets the thickness of the face. If the field is not set, the default value = "s" is used (piece thickness).
- The thickness setting can be parametrized on:
  - piece dimensions
  - "o" and "v" variables
  - "r" variables

  Using the parametric programming, the thickness of the face is returned by the geometric library multi-purpose function geo[sface; nside].
- : it shows the icon of the reference face used assigning the fictive face. This piece of information cannot be modified and is only shown if enabled.
- **P0{...} P1{...} P2{...}** assigns the coordinates of the three face points in a dedicated window (see later on). If the field status is active, the window starts by pressing any alphanumeric key or the function key F2.
- **Pr1,Pr2,Pr3,Pr4,Pr5** sets up to 5 additional face parameters. Value assignment can be parametrized according to the rules valid for **Sf** thickness. The values are interpreted as coordinates (typology: double; unit of measure corresponding to the unit of the piece). The columns are optional. Using the parametric programming, the parameters are returned by the geometric library multi-purpose function geo[pr1; nside], geo[pr2; nside], ..., geo[pr5; nside].

The commands available to change the fictive faces are contained in the contextual menu:

Import from file: it imports the assignments of variable geometries from another program. The File Open window opens and the preview of the assigned faces is managed. **ATTENTION**: running this command does not delete the fictive faces with programmed workings.

Copy: it copies the settings of the selected faces (current face, if there are no selected rows) in the local Clipboard. The copied faces are available to Paste again in the same program or in another one. To select or deselect a face, click on the header cell of the corresponding line by pressing the key **[CTRL]**.

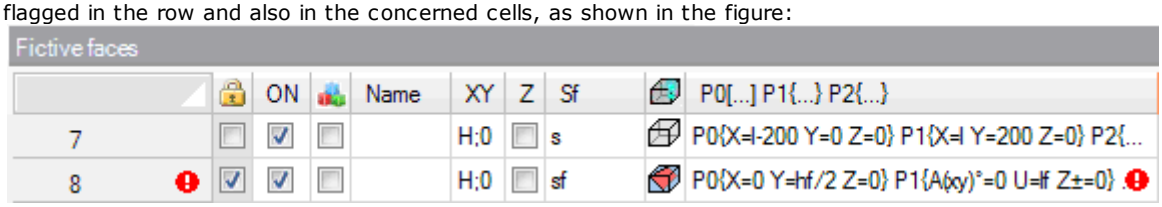
Paste: it assigns the settings as from the local Clipboard. Two cases can occur:
if the local Clipboard assigns one face only: the paste is done on the current face;

if the local Clipboard assigns more than one face: the paste is done respecting the numbering of the faces in the Clipboard. In this way, face 7 is copied on face 7 and so on, until the Clipboard is empty.

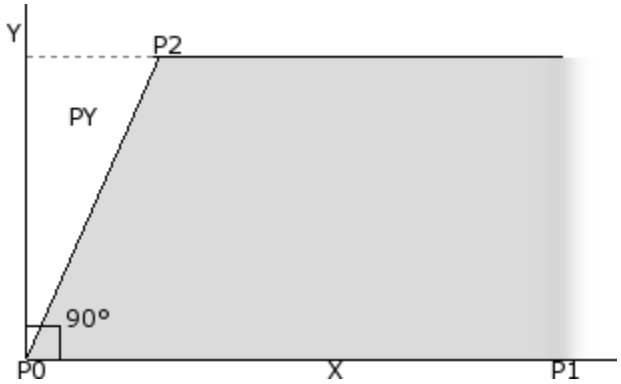
Delete: it deletes the setting of the selected faces or of the current face, if there are no selected rows.

Delete all: it deletes the setting of all the fictive faces that do not have any programmed working and that are not reference faces for faces that cannot be deleted.

The area of the errors shows the list of only the errors occurred while setting the fictive faces. An invalid setting is flagged in the row and also in the concerned cells, as shown in the figure:



An invalid setting can concern more assignments: geometry, thickness, additional parameters.
If the geometry of a fictive face is incorrectly assigned, the XY plane of the face coincides with the reference plane and the Z axis is oriented as in the programming.
A fictive face is always identified through three distinct and not aligned points:



- P0 is the origin of the face xy plane
- P1 is the point that orients the x+ axis;
- P2 is the third point on the xy plane:
  - if the line through P2-P0 is perpendicular to the line through P0-P1: P2 is the point that orients the y+ axis;
  - otherwise: the point that orients the y+ axis in PY is determined.

The P0-P1 distance assigns the length of the face.
The P0-PY distance assigns the height of the face.

From an operational point of view, the points shown in figure are not always known. Not always the values of the coordinates of the three points are known, but for example we know:

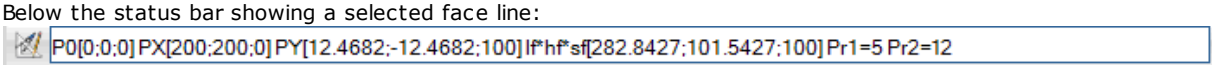
- how tilted the face is in comparison to another one, or
- how tall the face is, or
- where the face plane ends.

The assignment modes of the coordinate points of the face try to enable all these options. In the following paragraphs, there are examples explaining the definition of fictive faces.

A fictive face can be defined similar to another face of the piece when the coordinate points of the faces can be overlapped by translation operations in any direction and/or rotation only on the XY plane of the same face. So, the planes of the two faces must be parallel and it must be possible to overlap the Z semi-axes.
Workings and technology that are normally applied to the real face can also be applied on a fictive face similar to one of the 6 real faces of the piece.
The assignment of a face is fully independent from the assignment of a reference face: the similarity sizes up the geometric conditions of the face after its definition.

Information on fictive faces in the local status bar

Below the status bar showing a selected face line:

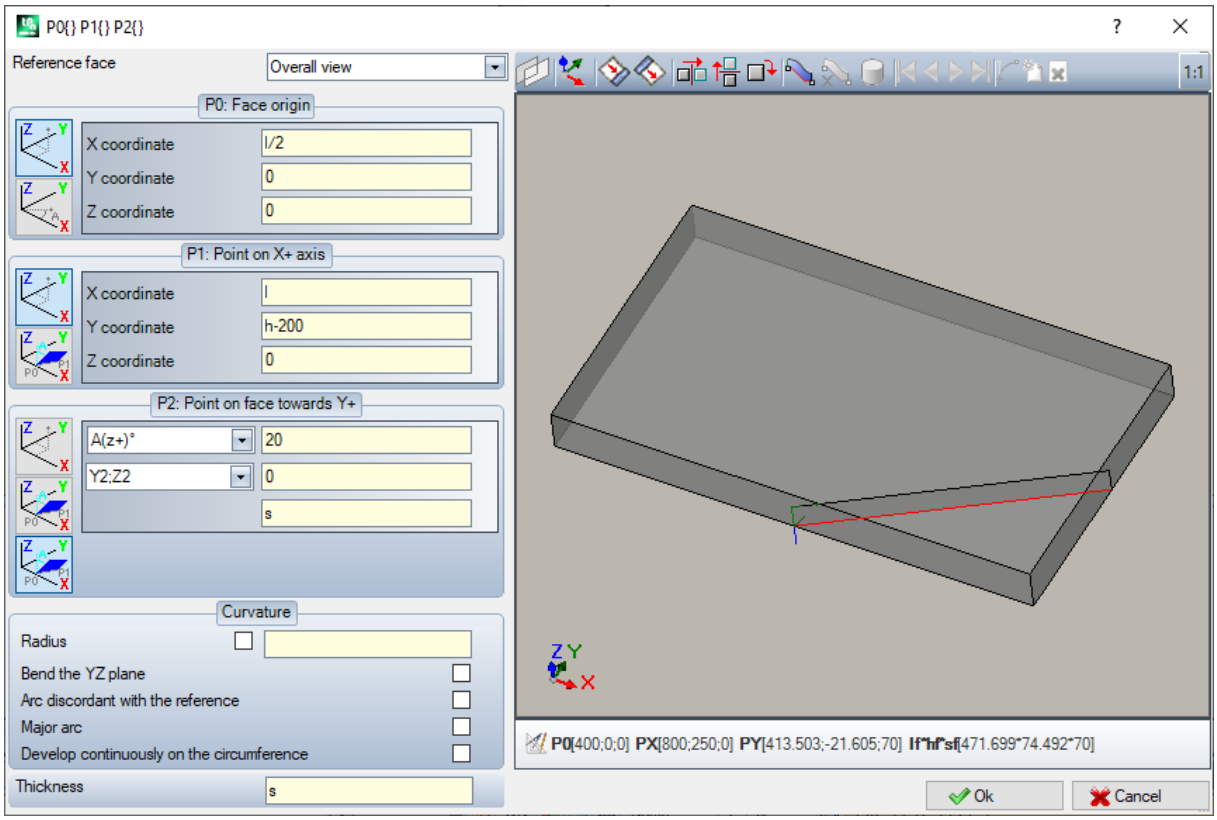


P0[...]	point of origin of the face
PX[...]	extreme point along the x+ axis
PY[...]	extreme point along the y+ axis (calculated)
lf*hf*sf=[...]	face dimensions
Pr1=...Pr5=..	values calculated for the additional parameters, if set.

Curved fictive faces and modelling

The assignment of fictive faces can also manage the capability of piece modelling, assignment of curved faces or of surfaces (please, read a specific documentation, that can be recalled by the command "Modelling help" available in the).

Example 1



- **Reference face**: it selects the xyz system to assign to the fictive face to be defined. It can be the absolute Cartesian system on the piece (selection from the list: Overall view) or the xyz system of another real or fictive face. The fictive curved faces or those assigned as surfaces are excluded from the list.
- : it sets the current face in such a way that it coincides with the selected reference face. The option is disabled, if the reference face is Overall view.
- : the three tools apply a remarkable transformation on the three points of the face. More specifically: horizontal symmetry, vertical symmetry and 90° rotation. The tools modify the three points with numerical settings and in a system of Cartesian coordinates: parametric assignments or polar system selections are lost.
- : in case of warnings of failures detected in defining the fictive face, the text area below the graphic control switches to Error area.

Assigning face points:

- **P0: Face origin**: the x, y, z coordinates of the origin of the fictive face (P0 point) are assigned in Cartesian (first selected bitmap) or polar (second selected bitmap) coordinates
- **P1: Point on X+ axis**: the coordinates of P1 are assigned in Cartesian (first selected bitmap) or polar (second selected bitmap) coordinates
- **P2: Point on face towards Y+**: the coordinates of P2 are assigned in Cartesian (first selected bitmap) or polar (second selected bitmap) coordinates, or the rotation of p0-p2 segment with respect to an axis is assigned.

If this last option is chosen, data in the left cell allows the selection of one of the 6 coordinated semi-axes of the piece:

- A(z+)°, A(z-)°
- A(x+)°, (x-)°
- A(y+)°, A(y-)°.
  The selected semi-axis assigns the reference Y axis of the face (with origin in P0 and y+ direction on the selected semi-axis). The value sets the rotation angle (in degrees) of the y+ axis of the face around its own x axis: the axis rotates in positive direction towards the z+ axis of the face (selection as from: **Z air**)

In the example:

- the reference y+ axis of the face is assigned as Z+ of the piece;
- the rotation angle is 20°:
  - with Z air in the right reference: the face plane rotates towards the outside of the figure
  - with Z air in the left reference: the face plane rotates towards the inside of the figure.

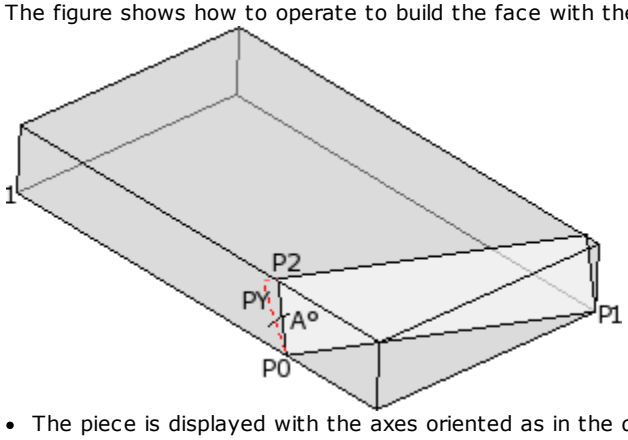
Up to now the face plane has been assigned, but the P2 point is still to be positioned: the y+ semi-axis is fixed, but not the position of P2.
The other drop-down case allows the selection of different ways to complete the P2 assignment:

- hf: assigns the height of the face. P2 point falls on the y+ axis, at the assigned distance (field on the right of the drop-down box). The value is taken as absolute value:
- X2;Y2: it assigns the (X, Y) coordinates of P2, while the z coordinate is calculated with the condition that the point belongs to the face plane
- X2;Z2: assigns the (X, Z) coordinates of P2, while the y coordinate is calculated with the condition that the point belongs to the face plane
- Y2;Z2: assigns the (Y, Z) coordinates of P2, while the x coordinate is calculated with the condition that the point belongs to the face plane

By assigning two of the coordinates, P2 point usually falls out of the y+ axis. In the figure the selection made is Y2;Z2:

- Y2 is set on the field on the right side of the box ("0");
- Z2 is set in the field on the right side, but under the box ("s").

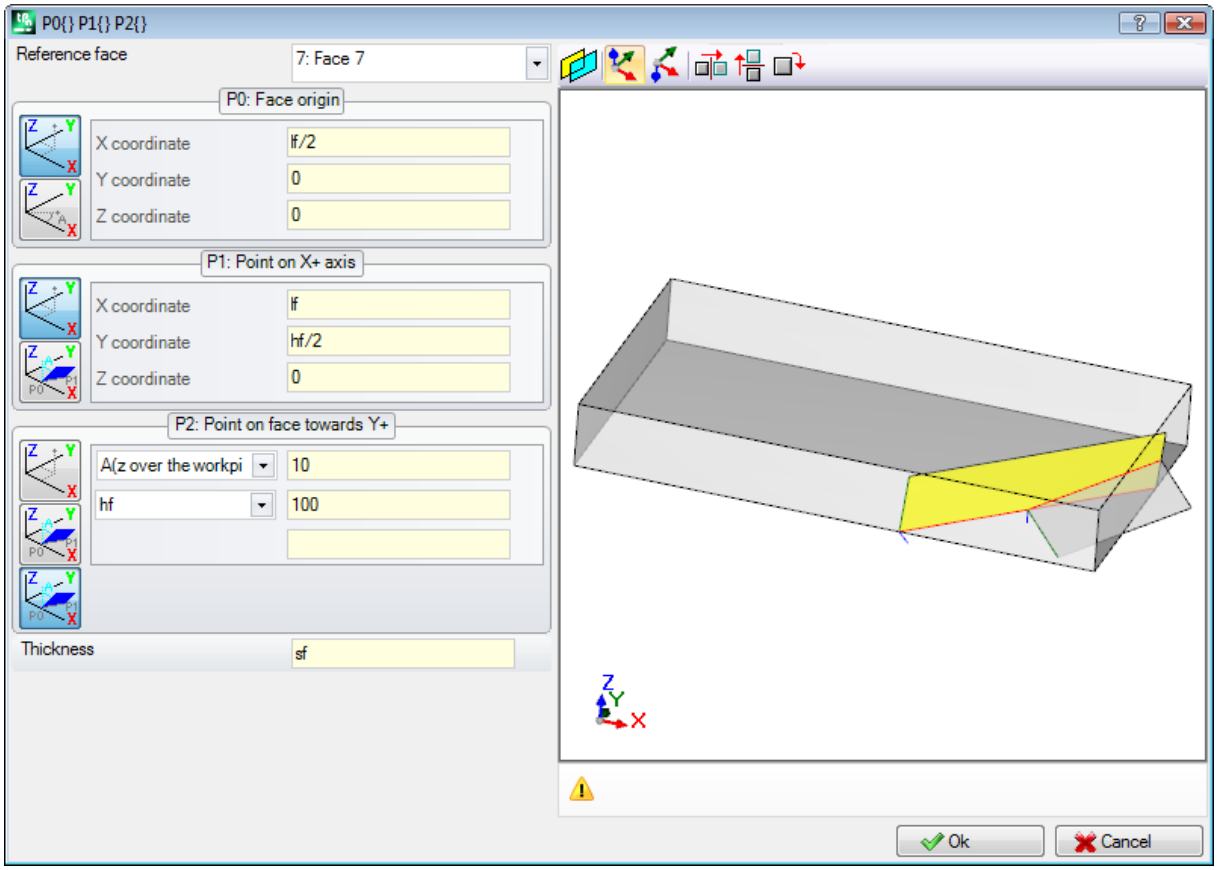
The figure shows how to operate to build the face with the assigned parameter settings:



- The piece is displayed with the axes oriented as in the drawing of the Cartesian coordinates shown bottom left (point 1 is the origin of the axes)
- the origin of the face is P0 and the X+ axis is between P0 and P1
- a semi-axis oriented as Z+ of the piece is indicated from P0: it is the Y+ axis of the face, with null rotation
- the selection of the right Cartesian coordinates orients the Z+ axis of the face towards the outside of the figure
- the Y+ axis is rotated A° (20° -> positive value -> it rotates towards the Z+ axis of the face)
- the (Y, Z) coordinates fix P2 point on the left side face of the piece: P2 forms an angle smaller than 90° with the X axis of the face (from P0 to P1), therefore the P2 projection on the Y axis of the face (PY) is recalculated. The linear segments which join the origin of the face to P2 and P2 to PY indicate that the P2 and PY points do not coincide.

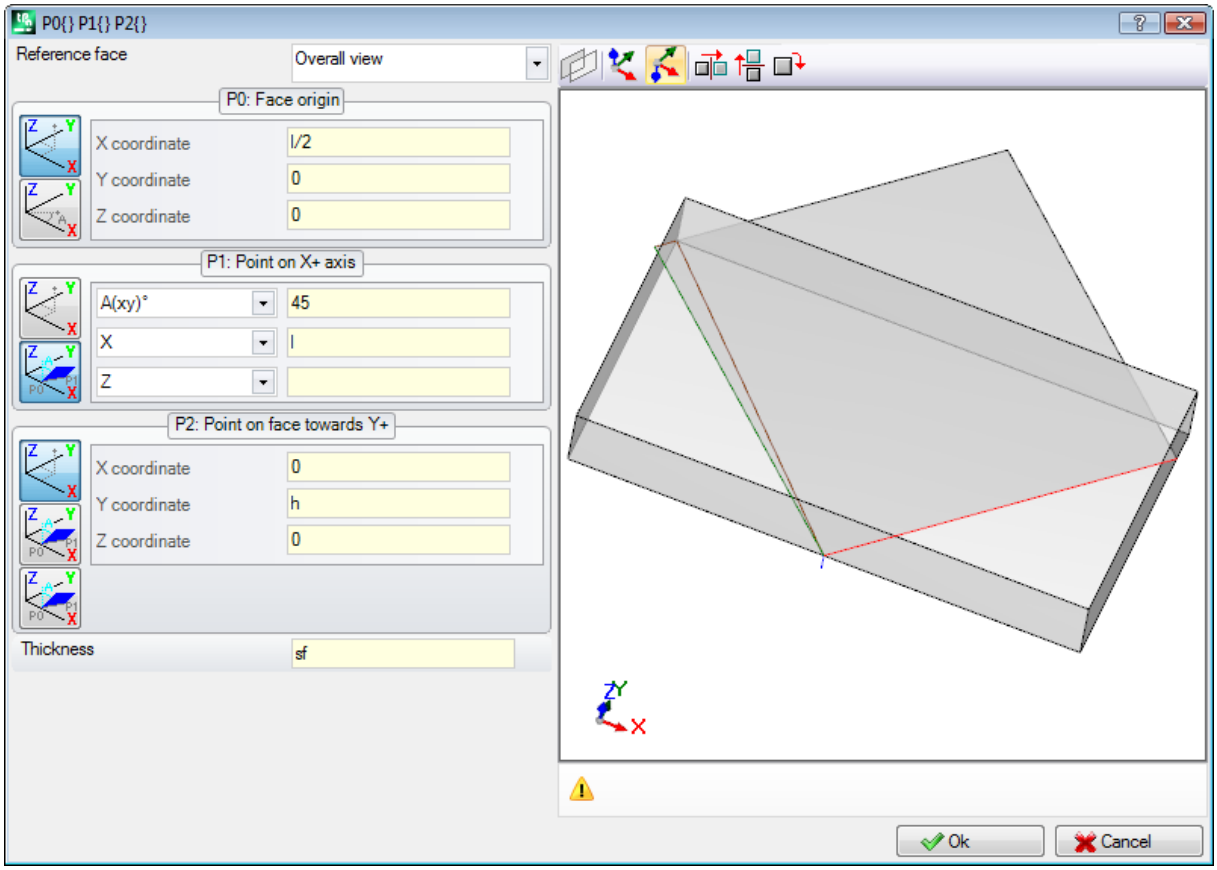
Example 2

In the following example, a fictive face (already defined) as reference face is configured:



- **Reference face**: the selection means that the face is to be assigned by using the xyz system of another fictive face: the face 7. For us, the face 7 is the assigned one with the **Example 1**. The drop-down box of the reference face provides a list of faces:
  - all the real faces of the piece (**ATTENTION**: also the real faces that are not manipulated);
  - the fictive faces assigned on the piece with lower number than that of the face to be defined. If, for example, face 8 is being assigned, it is possible to select a real reference face or, if fictive, only face 7.
- **P0: Face origin**: the first bitmap is selected in the frame. The selection means that the three coordinates of the point are known, but now it is about coordinates assigned on face 7. The coordinate fields locate the face origin on the half of the x axis of face 7 (lf/2; 0; 0):
  - the adoption of lf, hf, sf (face length, height and thickness) variable arguments in parametric programming leads to the use of the dimension values of face 7.
  - **ATTENTION**: the Z Coordinate value, if different from 0, uses the same conventional signs which are valid for the faces (negative or positive in face working);
- **P1: Point on X+ axis**: also in this case, the first bitmap is selected. The selection means that the three coordinates of the point are known, here also assigned on face 7. The fields of the coordinates locate the P1 point to (lf; hf/2; 0);
- **P2: Point on face towards Y+:** the third bitmap is now selected in the frame. The selection means that the inclination of the face to one of the coordinated axes of the reference face (face 7) is known. Assignments are similar to those of the previous example:
  - **A(clearance z)°**: it matches the already described A(z+)° selection, but now the message indicates that the choice falls on the z semi-axis over the piece (similarly: A(z piece)° corresponds to the already described A(z-)° selection, but now the message indicates that the choice falls on the z semi-axis in piece working);
  - **hf**: the P2 point is now determined by setting the face height.

Example 3



- **Reference face**: Overall view is set. The selection means that the fictive face is to be assigned by using the absolute xyz system of the piece.
- **P0: Face origin**: the first bitmap is selected in the frame. The selection means that the three coordinates of the point are known. The fields of the coordinates locate the face origin at half of the x axis of the piece (l/2; 0; 0):
- **P1: Point on X+ axis**: the second bitmap is selected in the frame. The selection means that the polar coordinates of the point on one of the three Cartesian planes of the piece are known (if a reference face was set, we could say: "...on one of the three Cartesian planes of the reference face"). The setting fields shown are different from the previous cases:
  - **A(xy)° 45**: the left drop-down box allows selecting one of the 3 Cartesian planes:
    - A(xy)°: xy rotation plane
    - A(xz)°: xz rotation plane
    - A(yz)°: yz rotation plane

  The value sets the rotation angle in degrees on the plane and the pole (centre) of the polar coordinate system is the P0 point: the axis which comes out of P0 on the plane and with the assigned angle defines the face X+ axis.
  On the three planes, the angle rotates in positive direction:
  - with the X+ axis which closes towards Y+, in case of xy rotation plane
  - with the X+ axis which closes towards Z+, in case of xz rotation plane
  - with the Y+ axis which closes towards Z+, in case of yz rotation plane

  The X+ semi-axis is fixed, but not the position of P1 on this one.
  - **X l**: the left drop-down box allows you to choose among 3 different ways to assign P1 on the plane defined by a polar coordinate system:
    - **U**: it assigns the module of the polar coordinate system (distance of P1 from P0, on the rotation plane). The value is taken as absolute value
    - **X**: it assigns the x coordinate of P1, while the y coordinate is calculated provided that P1 point belongs to x+ axis of the face;
    - **Y**: it assigns the y coordinates of P1, while the y coordinate is calculated provided that P1 point belongs to x+ axis of the face;

  The coordinates selected in the list correspond to the rotation plane chosen:
  - (X, Y) in case of xy rotation plane
  - (X, Z) in case of xz rotation plane
  - (Y, Z) in case of yz rotation plane

In the figure, the option selected is: X1=l.

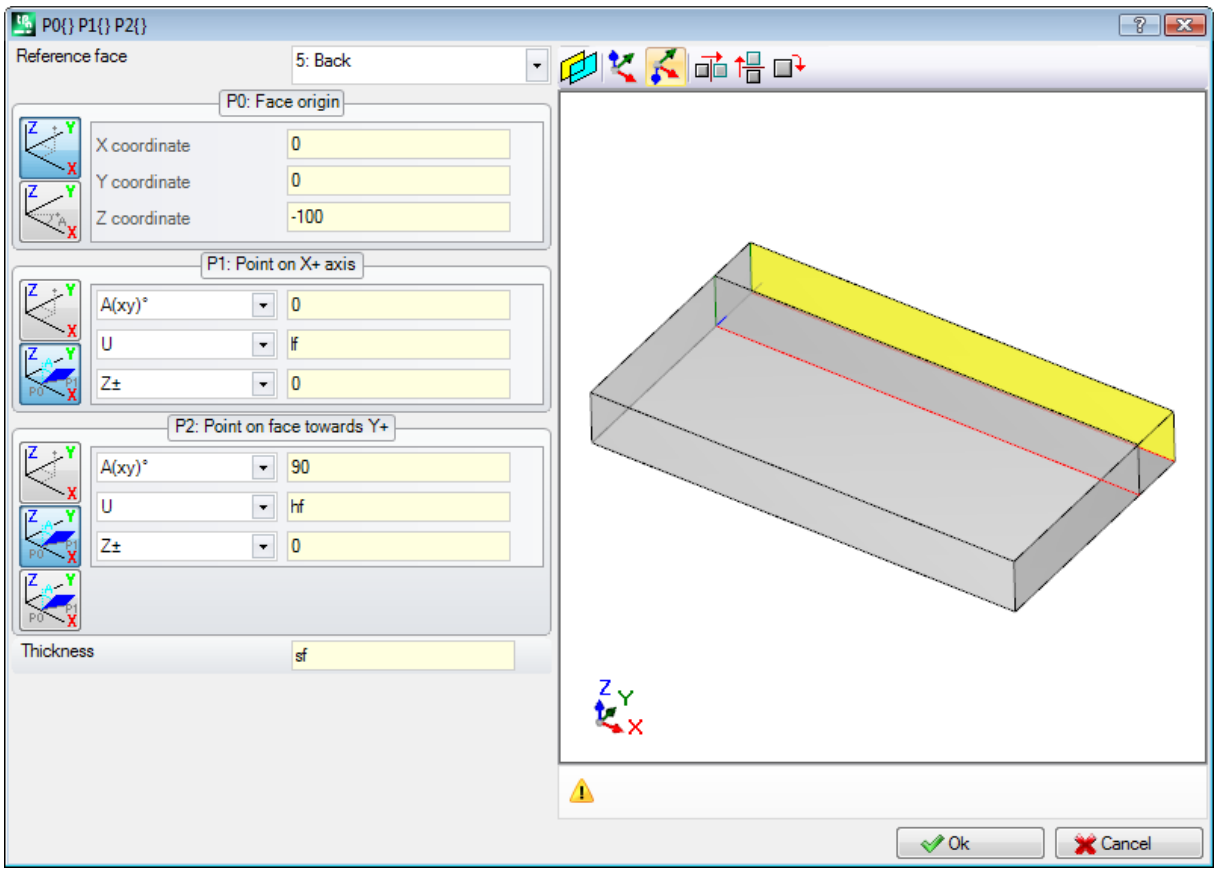
Once the position of P1 on the selected plane of the polar coordinate system has been assigned, the position of P1 on the axis perpendicular to the plane is still to be defined.

- **Z 0**: the left drop-down box allows choosing among 3 different ways to assign P1 on the third axis (in the example: Z axis):
  - Z: it directly assigns the position;
  - Z±: it assigns the variation of position with respect to the assigned value in P0 point;
  - AZ°: it assigns the angular variation with respect to the value assigned to P0 point. The set value must range between -90° and +90°: the value is considered valid, if it is included in the above range, extremes excluded (the difference among values is significant if greater than epsilon = 0.001°). Positive values of the angle determine height increment, negative values determine height reduction.

  The here assigned coordinate corresponds to the axis perpendicular to the rotation plane:
  - z, in case of xy rotation plane
  - y, in case of xz rotation plane
  - x, in case of yz rotation plane
- **P2: Point on face towards Y+**: the first bitmap is selected in the frame. The selection means that the three coordinates of the point are known. The fields of the coordinates locate the P2 point in (0; h; 0). P2 forms an angle smaller than 90° with the x axis of the face (P0 to P1), therefore the P2 projection on the y axis of the face (PY) is recalculated. The linear segments which in the graphic representation join the origin of the face to P2 and P2 to PY indicate exactly that the P2 and PY points do not coincide.

Example 4

Assign a parallel face to a pre-existing one:



Reference face: Face 5: press then the button  . Set the Z coordinate of the P0 point to the –100 height to translate the face along the Z- axis of the reference face so as to obtain the face shown in the figure. If it is necessary, move the P0 point on the x and/or y coordinates: assign values different from 0; if it is necessary assign different length and/or height dimensions: replace lf and/or hf values.

Section of constraints

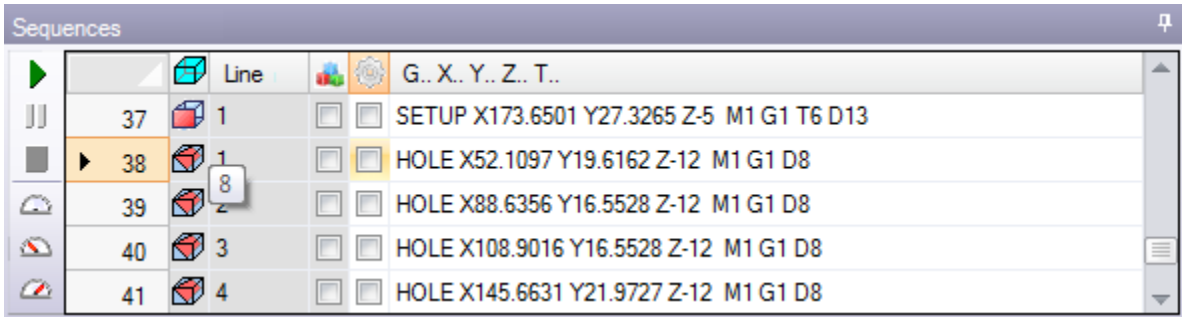
This is an optional page .
It is included in the custom information section. The assignments are configured in the dedicated function and their meaning is unknown to the program.
What previously described about custom information in Special Setting is still valid.

Optimizations

This is an optional page .
It is included in the custom information section. The assignments are configured in the dedicated function and their meaning is unknown to the program.
What previously described about custom information in Special Setting is still valid.

Sequences

This is an optional page.
It allows establishing a specific execution order for all the workings assigned to the piece.
The page arranges the workings in a table: each row corresponds to a working.



- **Heading**: it provides the sequence number
- : it graphically displays the face where the working is programmed. Moving the mouse pointer over the icon, a tooltip with the face number appears
- **Line**: sequence number of the working in the face program
- : the box is checked in case of construct working
- : the box shows the primary colour associated to the working, according to the kind of working (points, setup, profile segment) or operation code
- working optimization flag (optional). The flag interpretation depends on the single application. A single box can be enabled or disabled or, using the commands of the contextual menu, the user can enable or disable more boxes. The field is the only one that can be modified in the table.
- **Working**: description text of the working
- **G..X..Y..Z**: it displays the ASCII name of the working, the application point and the technology assignments (machine, group, head, spindle, diameter, technological priority).

If at least one of the listed workings has an assigned description field or a Name, a column for the corresponding display is added.
The available commands to modify the run sequence list are arranged in the contextual menu, that can be called by pressing the right mouse button in the window area.

Initialize the piece according to the program sequence: it initializes the list according to the automatic order of the faces (top face first, then: bottom, front, right-side, rear, left-side, fictive in numbering order from 7 to 99) and of programming for each face. This command cancels any modification made manually since the start of the session (Cut, Paste, Assignment of optimization flag).

Initialize the face according to the program sequence: it enters the list of the face workings the current line belongs to. For instance, if you select the command on a row that shows a working on face 4, all workings of face 4 are inserted starting from the row itself, keeping the programming order of the face itself.

Enable optimization flag: it enables the optimization flag of the selected rows. This command is optional.

Disable optimization flag: it disables the optimization flag of the selected rows. This command is optional.

Cut: it cuts the selected rows from the table (current rows, if there are no selected rows) and pastes them into the local Clipboard. This command is available, only if the Clipboard is empty. To select or deselect a line, click on the corresponding line header cell by pressing the **[CTRL]** key. To remove the selection from the whole list, click on any point of the table.

Paste: it pastes the content of the local Clipboard on the current row and clears the local Clipboard. This command is available only if one or more rows have been copied into the Clipboard. The rows are entered before or after the current working according to how the Insertion flag is enabled in the status bar.

Enter down: it selects the insertion point above/below the current row.

It is possible to modify the order of the workings also by dragging within the area of the table. The drag is applied to the selected rows (current row, if no row is selected) and it is activated by holding down the right mouse button until the required position in the list is reached. The change of the list by dragging is not conditioned by the status of the local Clipboard.

The Button bar on the left of the table displays the commands useful to simulate the list:



: it starts the graphic simulation. The current working is moved from the first row to the last one of the list, maintaining a constant period. The simulation can be interrupted by clicking the button and restarted by clicking again. The button ends the simulation.
The simulation speed can be changed by means of the buttons:
: sets the default simulation speed
: decreases the simulation speed
: increases the simulation speed

According to the functions assigned in customizing TpaCAD, the graphic representation may display all programmed workings or only the workings for which it is possible to assign the sequence. In this case the program does not show the workings that match the following items:
- working in piece-face
- open profiles
- induced calls of subroutine or macro
- workings for which the management of the sequence is not enabled

In the graphic representation the working matching the selected row in the table is highlighted. The representation may be in 3D, in box view, of single face (2D).

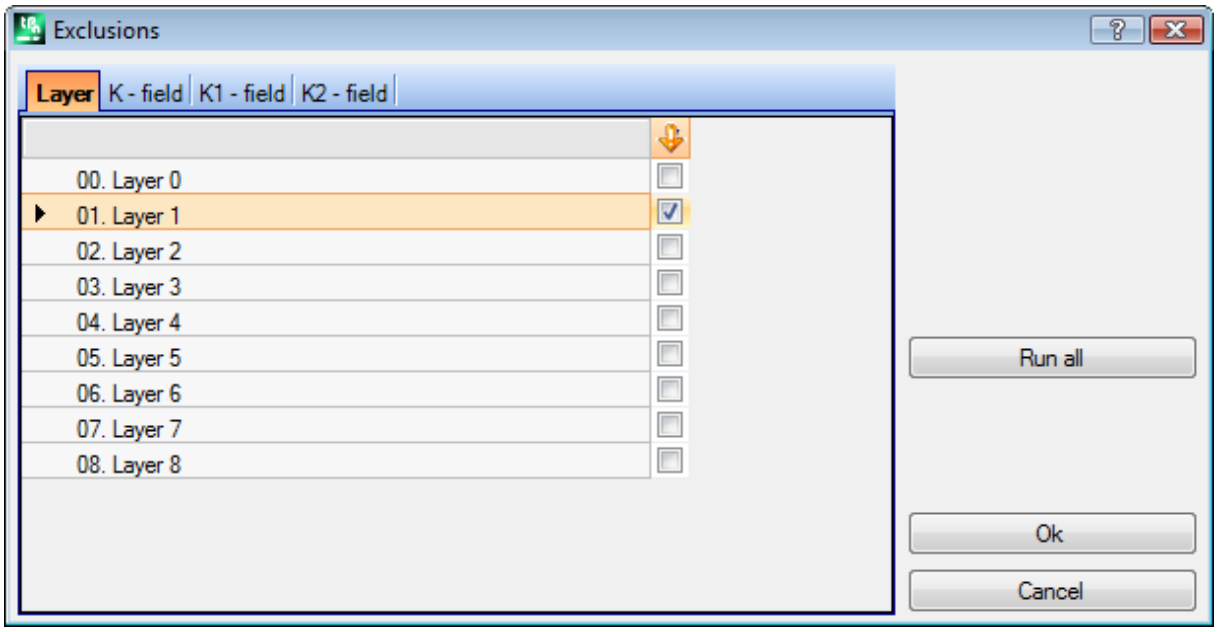
It is possible to find a working or to make selections also directly in the graphic area:
- **[Shift+(left mouse key pressed)]**: starts the area selection. Workings included in the indicated window are added to the current selections in the table. Holding down also the key [CTRL], the previous selections are maintained, otherwise they are reset.
- **[CTRL+(click on the left mouse key]**: it switches the selection status of the working nearest to the mouse position
- otherwise: **(click on the left mouse key)** moves the current row to the working nearest to the mouse position, resetting all the selections.

If you quit the page of the Sequences, a full update of the graphic representation of the program is made.

6.4 Advanced assignments

Exclusions

The command is optional and available in the group **Set** of the tab **Edit**  .
TpaCAD allows you to exclude a set of workings that are identified by a common property: L (Layer), K, K1 and K2. An exclusion is equivalent to an added logical condition, with the essential difference that it does not remain stored in the program. The exclusions here assigned are applied to the program by selecting the special view of the *Logical conditions*, available in the group **Views** of the tab **View on**. The possibility to assign exclusions in the machine is defined by the application managing the execution lists.
Each page of the window is available only if its control is enabled.



- **Layer** to assign the exclusions for the "L" property values: number and name given to the layer are displayed for each layer, for a maximum of 16 values (for subsequent values no exclusion can be assigned). The names of the layers can be modified in the page that opens from the Application menu **Customize -> Colours -> Layer**. The status of the layer is shown in the column whose header is the icon : the check mark in the box shows that the Layer is excluded (in the figure Layer 1).

The first row corresponds to layer 0 (not assigned).
The button **[Run all]** resets all the exclusions set in the page.
Select the button **[Ok]** to assign the Exclusions to the active program as they are assigned and directly activate the view on Logical conditions.

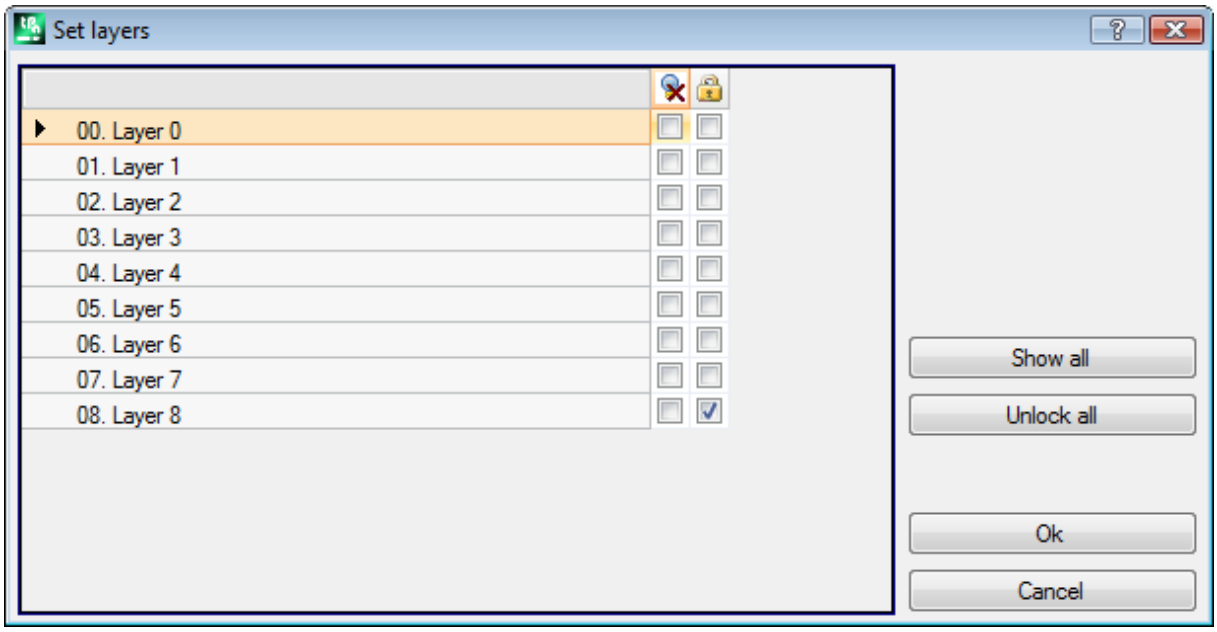
- **K Field**: for the assignment of the exclusions for the "K" property
- **K1 Field**: for the assignment of the exclusions for the "K1" property
- **K2 Field**: for the assignment of the exclusions for the "K2" property

It is also possible for fields (K, K1, K2) to assign the exclusions, for a maximum of 16 values.

Layers

The setting window is recalled from the tab **Edit** in the group **Set** .

The layers available in the configuration of TpaCAD are displayed (in the example: up to layer 8) and in any case up to 16 values.
In this window the user can assign the view and change filters of status for each layer.
The view filters, as assigned here, are applied to the program by selecting the special view *Layers* available in the group **Views** of the tab **View on**.
The change filters instead are directly applied: it is not possible to modify the workings with a value of a locked layer assigned.



- **Heading column**: number and name given to the layer are available for each layer (it can be changed in the page opened from the menu Application **Customize -> Colours -> Layer**).
- : layer view status: the check mark in the check box indicates that the Layer is excluded from the view
- : free or locked layer status: the check mark in the check box indicates that the layer is locked

The first row corresponds to Layer 0 (not assigned).
The button **[Show all]** activates the displayed status for all access levels. The button **[Unlock all]** sets all levels in the free status.

Special filters

The setting window is recalled from the tab **Edit** in the group **Set** .

This is an optional command.

It shows the values of Construct, Fields (O, K, K1 and K2), Technology.
Each page of the window is available only if its control is enabled.
In this window you can assign some view and change filters in the same way as in the Layer window. In the Technology tab only view filters can be assigned.
The view filters, as assigned here, are applied to the program by selecting the special view Special views, available in the group **Views** of the tab **View on**.
The change filters instead are directly applied: for instance, it is not possible to modify the workings with a locked value of O Field or Construct assigned.
For each property page:

- column : property view status. The check mark in the check box indicates that the property is excluded from the view
- column : free or locked property status. The check mark in the check box indicates that the property is locked.

The buttons **[View all]** and **[Unlock all]** deactivate the special filters set for the active page.

Construct
This shows the values of Construct configured in TpaCAD and in any case a maximum of 16 values. Each construct has its corresponding number and name.

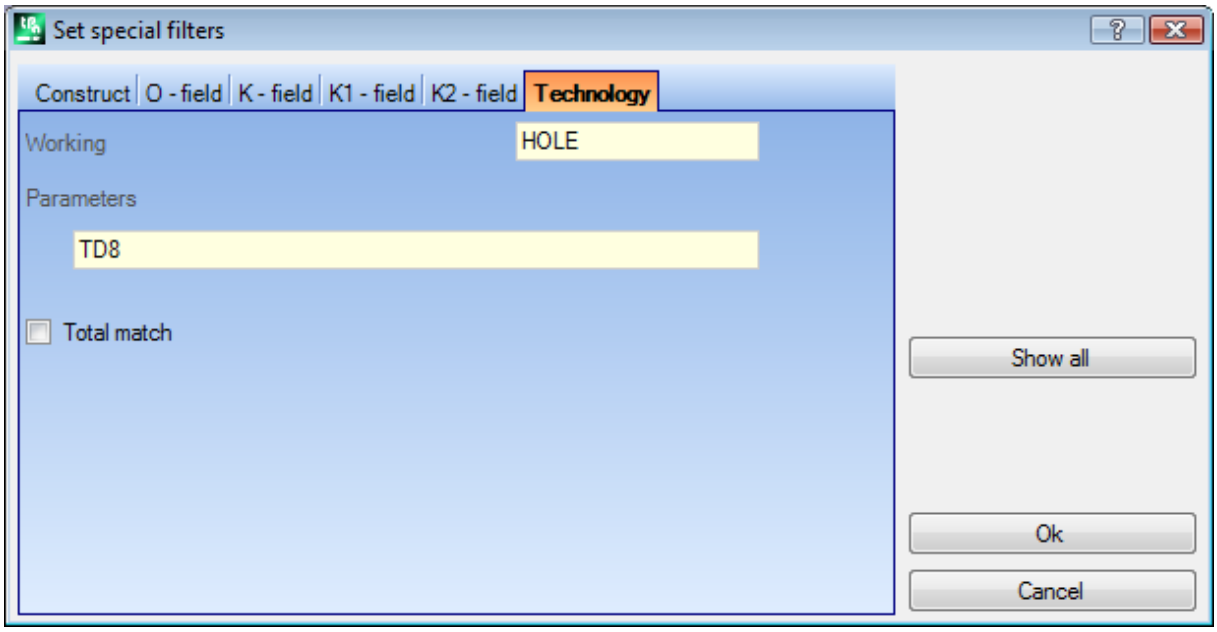
O Field
This shows the values of the O Field configured in TpaCAD and in any case a maximum of 16 values. Each construct has its corresponding number and name. The page is not available even if the assignment of "O" field is enabled on the single profile segments.

K Field

K1 Field

K2 Field
These show the values of Field (K, K1, K2) configured in TpaCAD and in any case a maximum of 16 values.

Technology
It allows you to choose the workings to be displayed by assigning the ASCII code of the working and/or a list of parameters belonging to the working itself. It is possible to set only one view filter.



According to the data set in the example window, only the workings with ASCII code "HOLE" and TD parameter set to value 8 are displayed.
In face view, this button next to the **Working** field allows setting the field to the code of the current working.
Only the **parameters** with technological value are interpreted (examples: machine, group, tool), so in case of a working belonging to a profile, the parameters belonging to the working that opens the profile are evaluated (setup or profile segment).

The entry **Total match** defines the search criteria of the workings that verify the settings. If selected, a query is performed also on possible expanded lists, that is on the workings that are assigned by subroutines or macros. If not selected, the query is performed on the programmed workings only (list shown in the ASCII text).
In the example displayed in figure:
- if the option is not selected, the query is run in the directly programmed workings (HOLE, TD8).
- if the option is selected, the query is also run for the workings (HOLE, TD8) deriving from the programming of a subroutine.

It is not necessary to assign both fields. Thus, if you want to refer to the example in figure:
- if the field Parameter is not assigned: only the workings with ASCII code "HOLE" are shown;
- if the field Working is not assigned: only the workings with TD parameter set to 8 are shown.

For the field of the parameters it is possible to set the following parameters:
- Parameters = "TD=r27" only the workings with TD parameter setting to "r27" are shown
- Parameters = "TM2 TD=r27" only the workings with TM parameter set to value 2 and TD parameter setting to "r27" are shown.

The comparison is done with the parameter setting string.

It is also possible to assign some logical conditions. Examples:
- Parameters = "TMR<=3": only the workings with TMR parameter less than or equal to 3 are displayed;
- Parameters = "TMR#3", "TMR<>3": only the workings with TMR parameter different from 3 are shown;
- Parameters = "TMR>3": only the workings with TMR parameter greater than 3 are displayed;
- Parameters = "TMR>3 GR=r4": only the workings with TMR parameter greater than 3 and GR parameter set to "r4" are shown.

In case of logical condition assignment (i.e.: not identity condition), we **recommend** assigning numerical settings. As a matter of fact:

- in case of numerical setting, the comparison is with the parameter value
- in case of parametric setting, the comparison is with the parameter setting string, with the possibility to consider only the difference between set strings.

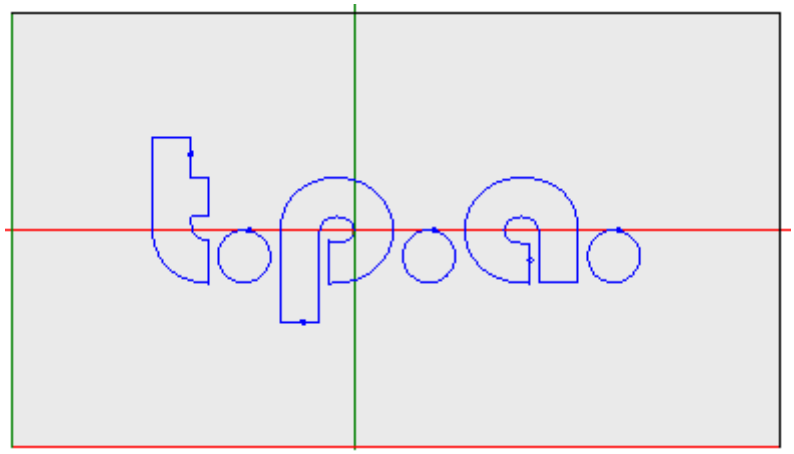
A change in the field of Parameters can determine automatic changes due to automatic checks. More specifically, the parts recognized as name of parameters are assigned with capital letters and assignments recognized as invalid are discarded.
In case of correspondence verified on the opening setup of a profile, this correspondence is applied to the whole profile.
The button **[Show all]** deactivates the special filters for the Technology.

7 Face

7.1 Graphic display of the Face View

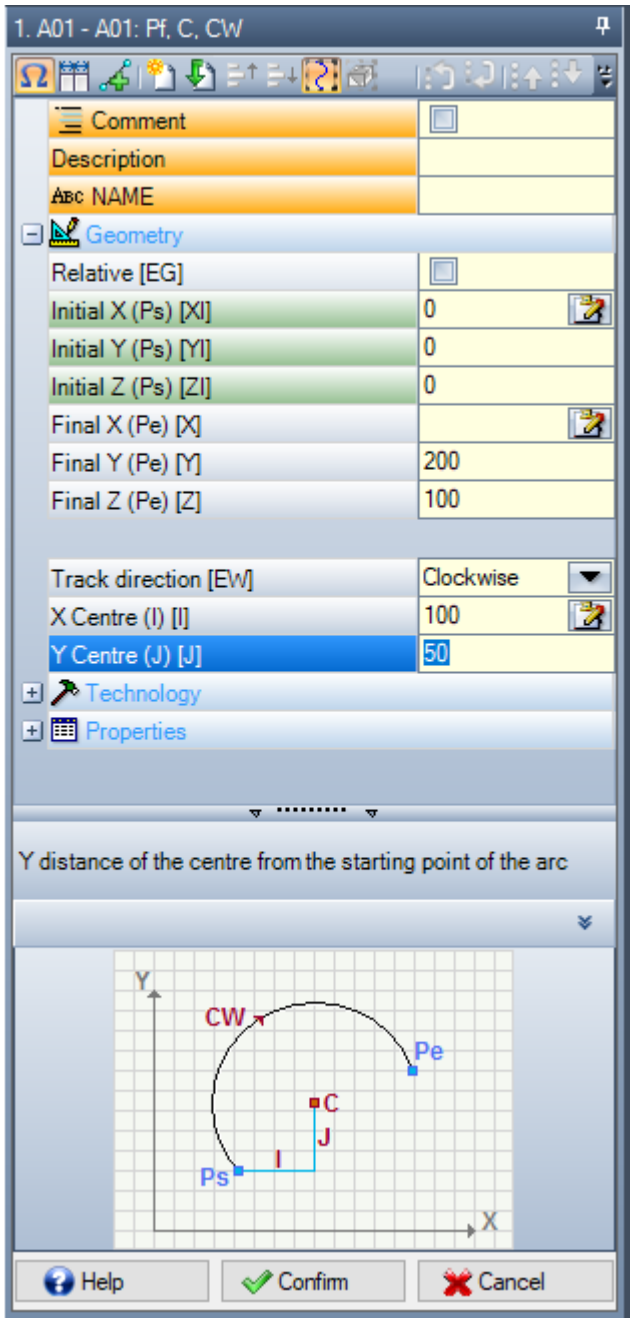
Graphical display area of Face View
The area of Face View can be represented in

- three-dimensional graphics. The current face and the face workings are coloured to be highlighted in contrast to the other faces and workings. The workings of the other faces are grey or their view can be excluded.
- box view: the current face and the workings of the face are coloured to be highlighted in contrast to the other faces and workings
- bi-dimensional graphic representation on the xy face plane. Both the current face and its workings are represented.

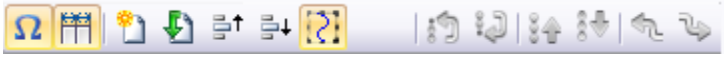


The figure is an example of face graphics. We can also see the cross-cursor full-field on the view. The reference system origin and the face axes are indicated.

Working assignment area
If workings are defined in the face program, the active working data is provided in the **working assignment** window.



- **Header**: it appears under: ASCII Name - Working description. As in the figure "A01 - A01: Pf, C, CW"
- **Parameters and working properties assignment area**: the entries are arranged in a list as direct entries (Name, Description, Comment, ...) or grouped in nodes (Geometry, Technology, Property). The properties have a meaning common to all workings and are arranged homogeneously in all workings: some at the top as direct entries (in the figure: Name, Description, Comment), other grouped in the last node. The parameters are instead more generically different among workings, both in meaning and organization. Near the description of the parameters, ASCII names assigned to the parameters themselves may be displayed in square brackets. To indicate that the parameter is a string the icon ab is added
  Each node in the list is first presented closed or open, according to the setting for the presentation of each working: later presentations of the working keep for each node the last status selected for the view.
- **Text Help Area**: description of the parameter being edited
- **Graphic Help Area**: graphical support to set the geometric data of the working.



The Button bar under the title of the working shows the commands:

Ω if enabled, the windows shows the ASCII names of the parameters, besides the descriptions. For example, as you can see in figure, the **Relative** parameter is called ASCII equal to [EG]

if enabled, the window equally sizes the two header columns of entries and programming. Otherwise, the header column is sized in such a way that the longest text can be fully displayed

this selection corresponds to the setting **Show always the start point in profile segments** in **Customize -> Environment -> Working edit**.

it duplicates the current working and enters the duplicated working after the current one. If the current working is being entered or changed, it assigns the changes first and then copies them. This command is available if enabled in the TpaCAD configuration. This command does not impact, if the current working change option is locked.

Reset working: it assigns the current working to the state that matches the insertion by direct selection from the working palette. This command does not impact, if the current working change option is locked.

: they move the current working to the position of the previous or following row. These commands are available, if they are enabled in the TpaCAD configuration. These commands do not impact, if the current working change option is locked. How these commands work, it depends on the status of this button:

- If not selected, the previous commands affect only the current working, even if it belongs to a profile. Moreover, the current working is moved to a position in the list, even though it breaks a profile before or after the profile;
- if selected: if the current working belongs to a profile, the previous commands move the whole profile. Moreover, if a profile is found before or after, this one is considered as a whole and it is not broken by the movement to the list.

: they move the current working to the first or to the last program list

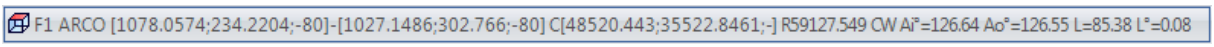
: they move the current working to the previous or the subsequent one

: they move the current working to the opening or closing line of the current profile (if the current working belongs to a profile).

: This command is available only if TpaCAD manages the piece-face and the automatic faces and if it is enabled in piece-face only. If selected, a moving working that creates an automatic face also moves the workings applied to the same face.

: the command **Edit subroutine** is available only if the working calls a subroutine. In this case the subroutine is opened in another instance of TpaCAD where it can also be modified.

The Status bar
In the Status bar the geometrical and technological information about the current working are shown.
An example of composition for an arc as follows:



The information in the status bar are specific for this working.
As in the case of an arc reported here, the complete geometry of the geometric element is shown: end and start points, centre, radius, rotation direction, angles of tangent on extreme points, lengths in the development of the arc (linear and angular).

7.2 How to open it

The face view to be displayed is selected from the Face Selection Bar that is always visible and includes the general view, the piece-face (if managed), the real faces and the fictive faces.



The real faces are the only faces actually enabled by the machine manufacturer during the configuration. The fictive faces are those that have been assigned in Overall View, with the exclusion of the faces set as construction auxiliaries.

Each bitmap of the bar corresponds to a face.

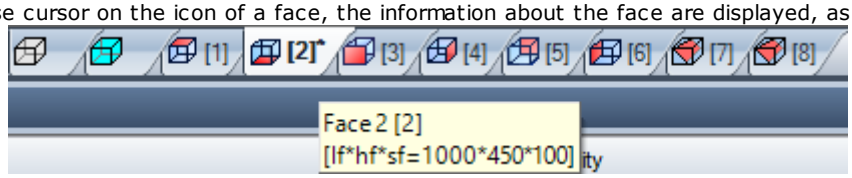
Overall view

 Piece-face

 from face 1 to face 6

 fictive face. In case of more assigned fictive faces: the bitmap is shown only for the first one. The reason is that to reduce the necessity to scroll on the Selection bar among all faces, in case of numerous assigned fictive faces.

Moving the mouse cursor on the icon of a face, the information about the face are displayed, as follows:

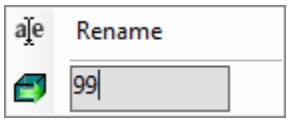


- the name (in figure: "Face 2")
- the number of face program rows, if available (in the figure: "(2)")
- the dimensions (in the figure: "[lf*hf*sf=1000*450*100]").

The numbering of the real faces may change on different applications: it is thus possible to assign a customized numbering.

 If a face has programmed workings, an asterisk is shown as a superscript, next to the face number.

By right-clicking the icon of the current face, a local menu opens:



- **Rename**: select, to change the name of the face. The name of the faces are not saved in a language file, so they cannot be translated. This entry is not available in General View.

- **Go to face**: the following row of the menu is visible only if the scroll down buttons of the *Selection Bar* are active. It may be the case of several assigned fictive faces of a video area specific for the horizontal bar with a greatly reduced size. In the row:
  - an edit field, where to set the number of the face to activate (in the figure: 99);
  - a button corresponding to **Go to face**, to move to the corresponding face view. To select *General view*: set a non-numerical character (or a negative value).

It is possible to activate a certain face in interactive way, directly on the overall view or from the face view (different from the piece-face):
- **select the short-cut ALT and click the corresponding area of the face with the left mouse button; or**
- **the corresponding area of the face by double mouse clicking.**

In case of several graphically overlapped faces: repeat the selection (ALT short-cut+click; or double-click) until the activation of the required face is executed.

7.3 ASCII Text Area

In the ASCII text area the program of the face in ASCII format is displayed. It is possible to directly make changes in the table for only the assignments of:
- **Description**: without a specific enabling, if the program row can be changed;
- **"C" property** (comment): if enabled from TpaCAD configuration and only if the program row can be changed;
- **"N" property** (name): if enabled from TpaCAD configuration and only if the program row can be changed.

The page includes a table of as many rows as the number of face workings. The list is arranged according to the original programming order.

			ASCII Text			M	K	K1	K2	Description
1	aa		HOLE EG0 X55.3982 Y75.7589 Z-12 TD20 TMC1 TR1 TP1	4	0	0	0	0	0	
2	bb		HOLE EG0 X87.3982 Y75.7589 Z-12 TD8 TMC1 TR1 TP1	1	0	0	0	0	0	
3	cc		HOLE EG0 X119.3982 Y75.7589 Z-12 TD8 TMC1 TR1 TP1	0	0	0	0	0	0	
4			HOLE X151.3982 Y75.7589 Z-12 TD8 TMC1 TR1 TP1	0	0	0	0	0	0	
5			IF ESP1=I TST1=0 ESP2=800 LOG1=0 TST2=0 LOG2=0 TST3=0...	0	0	0	0	0	0	
6			HOLE X183.3982 Y75.7589 Z-12 TD8 TMC1 TR1 TP1	0	0	0	0	0	0	
7			HOLE X215.3982 Y75.7589 Z-12 TD8 TMC1 TR1 TP1	0	0	0	0	0	0	

Header: sequential numbering of workings (starting from 1).

Free/locked status for the working: if the cell is active (visible check mark), it means that the ("L") field or the ("B") construct field or the O ("O") property field are locked. The level (either construct or O field) lock prevent the changes of the workings whose assigned field is equal to the locked value. The cell highlighted in different colour indicates that it does not correspond to a directly programmed status, but that it is the results of specific evaluations. The display state of the column can be modified in TpaCAD customization.

Active view: the cell is active (visible check mark), if the working is represented in the graphics. A working is not graphically shown, if one of the following situations occurs:

- the comment flag is active ("C" property)
- it is logical
- a display filter of the properties is active (fields "L", "B", "O") or technology (operating code and/or technological parameters)
- a special view of Logical conditions is active

Also in this case, the cell highlighted in different colour shows that it does not correspond to a directly programmed status. The display status of the column can be modified in TpaCAD Customization.
ATTENTION: the cell status does not depend on the view field setting of the program (in the Status bar).

Logical status: the column is significant, when the View of the Logical conditions is activated. In this case, the cell shows:

- a barred yellow arrow, if the working does not check the logical conditions
- a green arrow, if the working checks the logical conditions and it is not a particular logical instruction
- a stop sign, if the working checks the logical conditions and it is a particular logical instruction (ERROR, EXIT). In this case: the logical conditions can result from external conditions (cycles: IF – ELSE – ENDIF) added to conditions directly programmed on the workings.
- a warning sign, in case of WARNING logical instruction verifying the external logical conditions (cycles: IF – ELSE – ENDIF) and the conditions directly programmed in the workings.

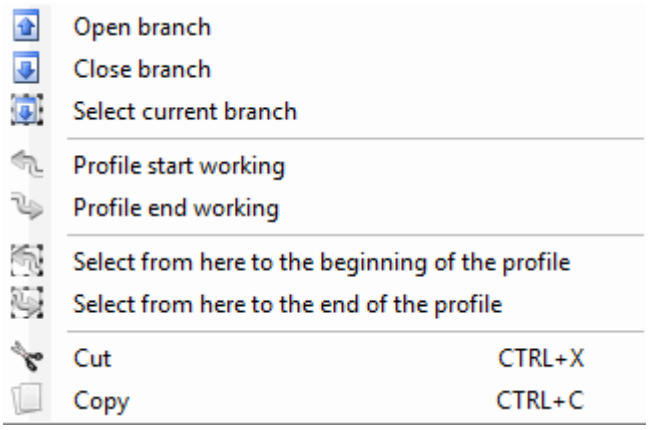
Also in this case, the cell highlighted in different colour shows that it does not correspond to a directly programmed status. The display state of the column can be modified in TpaCAD Customization.

"C" property (comment): the column is not displayed, if the property is not managed. If the cell is selected, this means that the working is in the list, but it does not affect the program. At this purpose, when reference is made to the previous or to the next working, with respect to another, it must be meant that *the comment workings are excluded*. If the cell is selected, in the window of the **working assignment** all the remaining fields, properties and parameters, are disabled, that is they cannot be modified. Possible working changes are brought to the normal status, if the cell is deselected.
ATTENTION: the modification is possible in the window of **working assignment**. The Comment field can be assigned to all workings, without exceptions. As already told: if the activation is verified, the cell can be directly modified.

"N" property (Name): it is an optional column. This is a name assigned to the working. If it is a non-numerical field, no longer than 16 characters, the valid characters are alphanumeric and the first character must be alphabetical. For example, the property is used to apply complex codes of transforms to be applied directly to programmed workings. The Name field can be assigned to all workings, without exceptions.

ASCII Text: it displays the operating code (it interprets the first field of the ASCII text. Examples: "G89", "L01", "A01") and the parameters, in the ASCII format as defined for the working. The column can display an indent for the immediate visualization of the logical program structure, which is evaluated on the basis of IF (IF, ELSE, ENDIF) and FOR cycles (FOR, ENDFOR, in case of macro program text).

In case of current non-comment working, click the ASCII text cell with the right mouse button to open a context menu that, for example, can help moving inside the program.



The entries that can appear in the list are:

- **Open branch**: it moves the current working to the previous program row that starts the current logical cycle (cycles: IF, FOR)
- **Close branch**: it moves the current working to the next program row that closes the current logical cycle (cycles: IF, FOR)
- **Select current branch**: it selects the block of workings that belong to the same logical branch of the current working.
- **Profile start/end working**: it moves the current working to the beginning or to the end row of the profile to which the current working belongs.
- **Select from here to the beginning/to the end of the profile**: it selects the part of the profile between the current working and the beginning/the end of the profile to which the current working belongs.

- **Expand working**: the current working is complex (subroutine or macro call), or it is a multiple profile segment; the command opens a window that shows its development. Each row of the expanded list corresponds to a working, whose geometric, technological information and assigned properties are provided in the same way as a bits of information are shown in the status bar of the current working.

colour of the working: it shows the primary colour associated to the working, according to the kind of working (points, setup, profile segment) or to the operation code. According to the typology, the colour is assigned in TpaCAD Customization or in the database of the workings. The colour shown here is not bound to assign properties (such as: level, construct). The column also appears in the expanded working list.
The display state of the column can be modified in TpaCAD Customization.

"L" property (layer): it is an optional column. It displays the value of the working layer. If the layer value is 0, it means that no layer has been assigned to the working. More specifically: if a value greater than 0 is set, the working can be displayed with an assigned colour (coloured square in the corresponding cell). **ATTENTION**: the value of the layer is mentioned here, and it may also result from the solution of a parametric programming.
The "L" field cannot be assigned on the following kinds of working:

- profile (lines and arcs) for the entire profile it is valid the setup value
- logical instructions (IF loops, assignment of variables…)
- custom workings (points, setup, logical) or complex for which the management in the configuration is disabled.

"B" property (construct): it is an optional column. It displays the value of the Construct field of the working. More specifically: if a value greater than 0 is set, the working can be displayed with an assigned colour (coloured square in the corresponding cell). If shown as a construct, the working is compiled but not executed.
"B" field cannot be assigned on workings, see "L" field.

"M" property: it is an optional column. It displays the value of the M field of the working.
The "M" field cannot be assigned on the workings, see "L" field; however, its management can be enabled on the profile segments.

"O" property: it is an optional column. If a value greater than 0 is set, the working can be displayed with an assigned colour (coloured square in the corresponding cell). If the maximum managed value is not greater than 4, TpaCAD can display a reference for the working (a side or a corner). **ATTENTION**: to display the icons, you must enable in any case some specifications in the configuration of TpaCAD.
The "O" field cannot be assigned on the workings, see "L" field; however, its management can be enabled on the profile segments.

"K" property: it is an optional column. It displays the value of the K field of the working. The "K" field cannot be assigned on the workings: see "L" field.

K1 **"K1" property**: it is an optional column. It displays the value of the K1 field of the working. The "K1" field cannot be assigned on the workings: see "L" field.

K2 **"K2" property**: it is an optional column. It displays the value of the K2 field of the working. The "K2" field cannot be assigned on the workings: see "L" field.

Description: descriptive text that can be assigned to complete or support the assignment of a working. The text can be modified directly in the table.

The configuration of TpaCAD determines which property columns are actually shown in the table.

8 Piece-Face

8.1 Overview

The face-piece is a particular face that does not have its own geometric identification. It can be stated that it represents the piece as a whole, including all faces which characterize it.
Conventionally, we assign to the piece-face:

- Absolute Reference System of the piece
- Piece dimensions (l, h, s)
- 0 as identification number.

The face-piece program allows to assign workings directly to different faces in a single program list. The assignment of a working is referred to its own application face, that is set in an additional field (see later: F field) in a window of working assignment.
A program written in piece-face **cannot be used as subroutine**. For this reason, the face-piece is made available only in case of a piece with program typology.

The program written in face-piece is not the sum of the programs written individually on the other faces but it is added to them.

If the end user of TpaCad needs to write subroutines, they must first program the workings in the views of the enabled faces, in order to apply them later.
A subroutine can be recalled instead in the piece-face of a program.

- The F fields set in the subroutine call declares in which face the subroutine is applied.
- The choice of which face of subroutine should be recalled is made according to the already mentioned modes (topic Entry, included, for instance, in the SUB working parameters).

The machine manufacturer can configure TpaCAD in such a way that only the piece-face can be programmed. This is only valid in the case of pieces with program typology. In the case of pieces with subroutine or macro-program typology, the real configured faces and the eventual programmed fictive are managed anyway, while the piece-face is disabled.

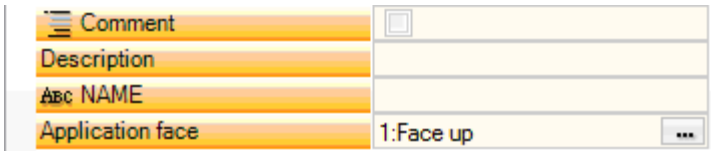
8.2 How to open it

The piece-face is selected from the Selection Bar of the face.



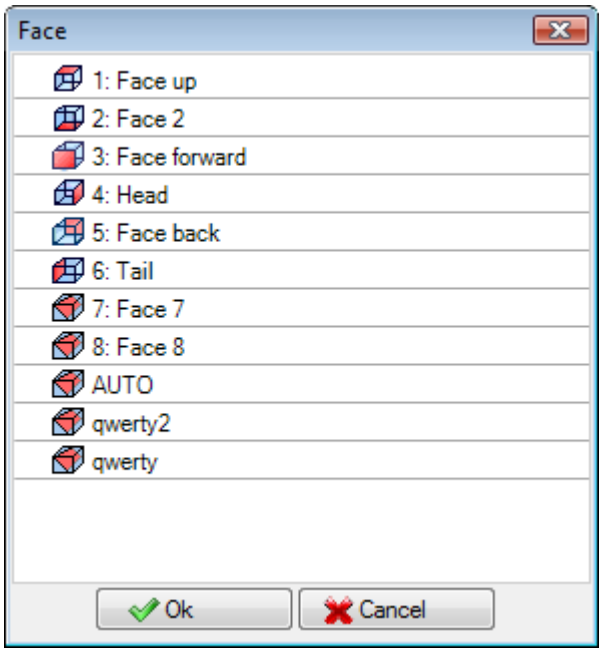
In the graphic area, the piece in 3D view is displayed. No face is marked.

8.3 Working assignment area



These data are the same as for all faces and add the application face or **F field** for each program working.

The field is only set in a dedicated window: select the button ... to display the list of all the managed faces on the piece that can be selected, both real of virtual. No parametric programming is allowed. The F field is significant for all working apart from the logical instructions (IF, ELSE, ENDIF, assignments of J variables…).
The list of the faces can change according to the selected working. For example, a saw working can be generally applied only to face 1 (top side) and 2 (bottom side). Therefore, in the list only the faces 1 and 2 will be displayed. If, before a working, there are some automatic faces assigned, the selection list for the F field also includes the AUTO option, which corresponds to the application into the automatic face.
For further details, see the code of the creation of an automatic face.
Now, let us see a window set in the example for the selection of the **F Field**:



In the list order there are:
- the 6 faces of the base piece (in this example, they can all be selected)
- two fictive faces (7 and 8)
- the AUTO option that corresponds to the last automatic face assigned before the current row of the current program
- two rows (the last ones) that correspond to the direct selection of one of the automatic faces assigned before with a name.

From this list the construction faces (fictive or automatic) are excluded.

8.4 ASCII text area

The information displayed are the same as for the face programs, beyond the application face or **F Field** for each program row.

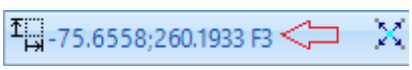
8.5 F Field

The command **F Field** is available in the group **Assign property**  of the tab **Edit**.

8.6 Representation

In piece-face, any face view can be activated: 3D, Box View or 2D.
Unlike the view of other faces, here the current face changes according to the change of the current working. Furthermore, the current face is also the selected face to be worked, for example by inserting a geometric element in an interactive way – point, line, arc. More specifically:
- to work on a face with already assigned workings, it is enough to click next to a working of the face;
- to work on a face without assigned workings: select the short-cut ALT and click the left mouse button in the corresponding face area, or double-click the mouse into the face area. In case of several graphically overlapped faces: repeat the selection (short-cut ALT + click; or double-click) until the activation of the required face is executed.

The working face appears in the status bar next to the mouse position:



8.7 Execution sequences

One feature of the piece-face program is that it directly defines the execution sequences. During the assignments of sequences, the workings programmed on piece-face cannot be sequenced.
The workings in piece-face are executed before every other list of face workings, according to the order in which they are programmed. Therefore, one of the main reasons why the piece-face is useful is that it meticulously groups the program workings that are to be executed in a predefined sequence.
A typical example is represented by the creation of fictive faces during the execution: it is necessary to ensure that the face creation is implemented before working the face itself. In this case it can be useful to assign the cut working of the face on piece-face: no matter how the program will be modified, the face is guaranteed to be created immediately.

9 Workings

9.1 Working type

Simple and complex workings

A working insertion is performed by selecting the working itself in the tab of **Workings**.

The **simple workings** include: individual drillings, individual setups, line and/or arc segments, logical instructions.

Point and setup workings have a direct assignment of technology and geometry.
Main use of setup working is profile opening. In this case, the setup provides for technological information to be used for profiling. A setup can also be used alone, that is not followed by a profile.
On the contrary, a point working is used alone. Examples of point workings are drills and insertions.

The **logical** workings are featured to meet also specific customization needs.
For example, logical workings can be implemented for:
- real-time measure of piece;
- scheduled stops during piece execution;
- limits setting.

Geometric and technological fields can be assigned to logical workings. Anyway, they are not interpreted by TpaCAD. The logical workings are neither displayed in a graphic area, nor calculated in overall dimensions; moreover, any relevant positioning sets by the following working is not applied.
Generally, a group of logical workings is always available besides those eventually assigned according to the specific customization needs. They are what we call logical instruction: if loops (IF - ELSE - ENDIF), ERROR, EXIT, assignment of J-variables.

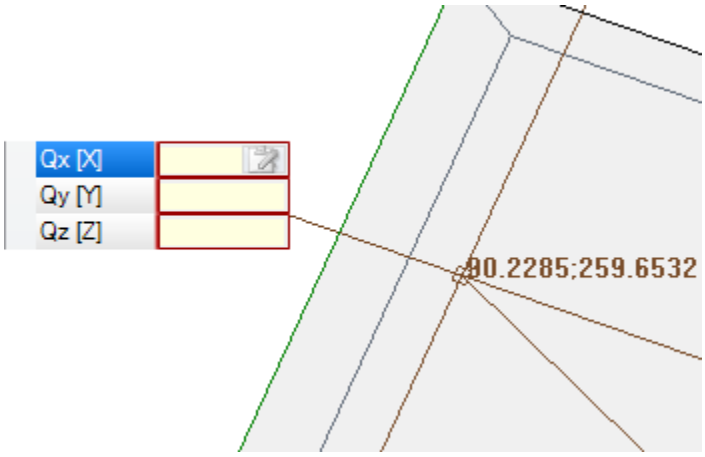
The **complex workings** are defined by combining simple and/or complex workings. They include for example: drilling cycles (FITTING, REPEAT), polygons, sawings.
As already mentioned, a working assigns parameters and properties.
A **working parameter** can always use each parameter setting managed in parametric programming. More specifically:
- piece and/or face dimensions (l, h, s, lf, hf, sf)
- program variables (o, v, r)

The maximum length of a parameter is 100 characters.
The **working properties** generally accept numerical programming only. For some properties, the use of parametric programming can be enabled with the same modes applied to the parameters. In case of parametric programming, the corresponding cell in ASCII text area presents the corresponding numerical value.

Once data has been confirmed, the working is actually entered into the face program only if no error conditions are detected in programming the working itself. In this case, it is necessary to solve problems or cancel the entry operation. Only in the case of assignment of the working in macro test, it is possible to confirm the entry also in case of error and this to compensate the eventuality of a spurious error: for instance, a typical case is the arc programming with the assignment of a geometry, using variables locally set within a FOR loop. Once the face program has been entered, this one is updated both in the graphic representation and in the ASCII text list; so, the entered working becomes active.

Edit wizard
During the programming of the working parameters, several helps are provided to the operator. In a very similar way to what already seen in the variable assignments, pressing the right mouse button within the assignment area of the parameter you are editing, a contextual menu is opened, where following entries are shown:
- **Suggest word**: it opens a menu where all functions and arguments of parametric programming, grouped in nodes, are available
- **Quick Info**: tooltip of the arguments required for the function in use
- **Info Details**: it opens the help page concerning the entered function
- **List of variables**: it opens a window containing the list of assigned variables ("r", "o" and "v").

For the parameters of a point geometric assignment you can display the icon  , that can be selected to acquire the position corresponding by directly clicking with the mouse in the graphic area. The direct acquisition mode is determined by the TpaCAD configuration and by the field setting for the program display (in the status bar). The acquisition in graphic area is always meant on the bidimensional xy plane and can be optionally extended to the depth coordinate. Following particular cases can be distinguished:
- significant assignment of a point (x and y coordinates): in the working area the fields corresponding to the two coordinate are highlighted. To force the acquisition of only one of the two coordinates, the other coordinate must be locked in the local menu managed in the graphic area (right mouse button)



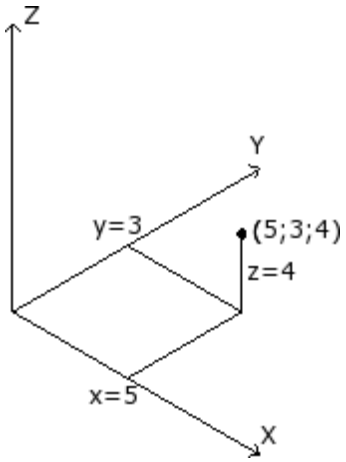
- significant assignment of only one coordinate (abscissa or ordinate): on the working area, only one corresponding field is marked and a horizontal or vertical cursor highlights its meaning.

The local menu managed in graphic area also makes available the activation commands of the snap (grids, entities), normally managed in the interactive acquisition procedures.

Application point

The application point of a working is defined by the assigned coordinates on the XY plane and by the Z coordinate perpendicular to the face plane.
The X and Y coordinates can be assigned in a system of Cartesian or polar coordinates.

Assignment of Cartesian coordinates:



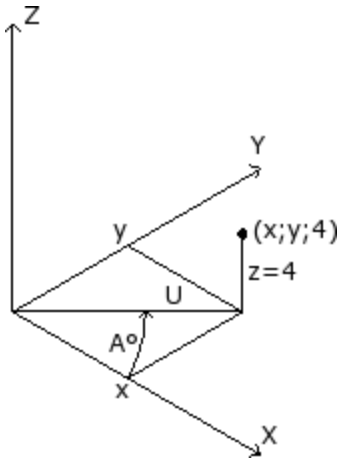
Qx [X]	If/2+50
Qy [Y]	20+32
Qz [Z]	a;4

In a Cartesian system the coordinates are directly assigned as:

- absolute from the face origin, if the box **Relative** is not selected.
- relative from the last position programmed above, if the box **Relative** is selected.

If point is assigned like in figure, with coordinates (x=5;y=3;z=4), but in relative mode with the last position programmed at (x=2;y=2;z=2), the working shall have its own application point at (x=7;y=5;z=6).
The absolute/relative selection is applied to all coordinates.
When **Relative** mode is selected, the absolute mode can be forced on a single coordinate by placing "a;" before the coordinate setting. If a coordinate is not assigned (empty field), the value set for the previous working is propagated.

Assignment of polar coordinates:



Geometry	
Relative [EG]	
X-Centre [I]	0
Y-Centre [J]	0
Qz [Z]	-5
Angle (A°) [A]	45
Module (U) [U]	100

The figure shows the polar coordinates. The Z coordinate is directly assigned, as in the case of assignment with Cartesian coordinates.
The position of the point in the XY plane is specified by giving its distance from a centre and its angle (in degrees) in XY plan from X axis.
In the figure:

- the centre is the origin of the face (0:0);
- the U distance from the centre is: 100;
- the angle is: 45°.

The absolute/relative selection is now applied to the Z coordinate and to the (x;y) coordinates of the polar system centre.
When the **Relative** mode is selected, the absolute mode can be forced on a single coordinate by placing "a;" before coordinate setting.

Technology

A point or setup working has a technology assignment involving the evaluation of the plant architecture. The technological data, in fact, must be defined according to the tool that will perform the working and the group and the machine to which the same tool belongs.

A plant consists of one or more machines, in each of which one or more groups (or heads) can work; they are divided into devices: tools, electrospindles, tool changes. When the technology is applied to a point or to a setup working, reference is made to a tool, that is fitted out in a position (spindle/electrospindle) of a machine head group.
To each head group a maximum device configuration is assigned depending on the specific application. Each machine can be provided with a tool catalogue and a toolholder catalogue, each toolholder can be fitted out with a maximum number of tools, always depending on the application.

General assessment criteria
Assessment criteria for tool programming applied by TpaCAD are examined below, according to a chart of possible cases, on the basis of **priorities** applied to their assessments:

Programming by spindle (or electrospindle) **and tool**

Machine [TMC]	1
Group [TR]	1
Electrospindle [EM]	100
Tool [T]	2
Tool type [TP]	100

In the picture, the **Electrospindle** field is set at 100, and the **Tool** field at 2.
By **Machine** and/or **Group**, setting a value is generally required. If the field is not assigned, the value 1 is set up by default.
By clicking on the icon  , the technological parameters can be directly selected from the window showing the technology.

The value assigned to the field **Electrospindle** sets the device position in a group, while the value assigned to the field **Tool** defines the tool (or the toolholder) to be fitted in the **Electrospindle**, with a possible interpretation of selection in a list of devices or device typology.
According to the technology of the machine, the value assigned to the field **Tool** can also define a toolholder and, in the case of tool-holder fitted out with more tools, it can also show the position in use.
In the example displayed in figure:

- if **Electrospindle** 100 of group 1 is associated to a tool change, then it is fitted with tool number 2;
- otherwise: the **Electrospindle** 100 must necessarily be fitted out with tool number 2.

If only one of the two fields (**Tool** or **Electrospindle**) is set up, reference is made to the case of programming for Electrospindle, described below.

Also the **Tool Type**, that allows a wider specifics on the tool selection, can be assigned. As displayed in the figure, the working is considered as correct and completed, only if **Tool**=2 of **Machine**=1 and **Group**=1 are configured with **Tool type**=100.

The field **Electrospindle** can be assigned by default and not be visible in the window of the working data setting. This is the case of a group with only one configured electrospindle or if the selection of a position on the group cannot be differently fitted out.

Programming by spindle (or electrospindle)

Machine [TMC]	1
Group [TR]	1
Electrospindle [EM]	
Tool [T]	12
Tool type [TP]	1

The following cases can occur:

- only the **Tool** field is available and set: (in the figure with value 12);
- only the **Electrospindle** field is available and set;

- both fields are available, but only one is set (in figure with value 12).

By **Machine** and/or **Group**, setting a value is generally required. If the field is not assigned, the value 1 is set up by default.
The tool is directly chosen in the **Tool** field (or electrospindle), using the current tooling.
If the spindle is not fitted out in the technological parameters, the following cases can occur:
- choice of the default Tool
- error situation.

As in the previous case, **Tool type** can be normally assigned.

Programming by diameter

Diameter [TD]	8
Machine [TMC]	1
Group [TR]	1
Tool [T]	
Tool type [TP]	1

The field **Diameter** is set: here with value 8, while the **Electrospindle** and **Tool** values are not set.
By **Machine**, **Group** and **Tool type**, a value can be generally set. If a value is not assigned, a default one is not set.
Selection criteria for program execution tool remain specific of a single application.

With the **Diameter** field set, a programming by diameter is recognized if the set value for **Tool** is null (=0) and if the set value for **Electrospindle** is null or negative (<=0).
ATTENTION: with versions prior to [2.4.7], recognizing the programming by diameter required that the **Tool** field was not set.

Programming by diameter is typical of drilling workings and, according to the declared available tools, can determine the execution of more drillings in one step.

Default tool
Neither the **Electrospindle**, **Tool**, nor **Diameter** field is set.
The selection of **Machine** and/or **Group** and/or **Tool type** can be forced. Selection criteria for program execution tool remain specific of a single application. Anyhow, programming by **Default tool** might not always be actually operating. In this case, an error during the program optimization is reported.

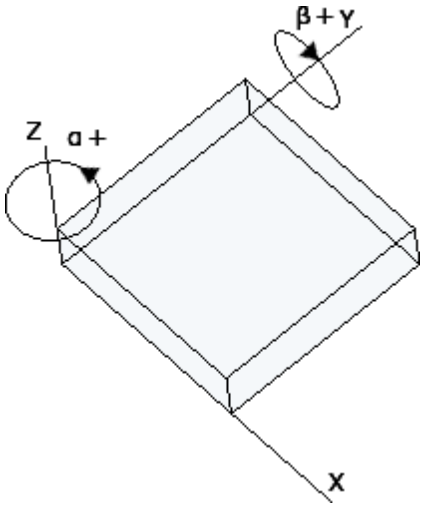
Automatic tool
The selection of the automatic tool takes priority on the settings assigned to fields **Electrospindle**, **Tool** and **Diameter**.
Selection of **Machine** and/or **Group** and/or **Tool Type** can always be forced.
Selection criteria for program execution tools remain specific of the application.

Oriented geometries

The normal working condition corresponds with setting the tool perpendicularly to xy plane of the working face. A setup working can also assign a tool orientation with respect to the face plane, in this case, it is called oriented setup. The following fields define the tool orientation:

- rotation axis (alpha),
- rotation axis around Y axis (beta).

Both rotation axes have a programming **absolute on the piece**.
Tool rotation fields, if assigned for the working, are significant in any case, even if they are not set (in this case, their value is 0).



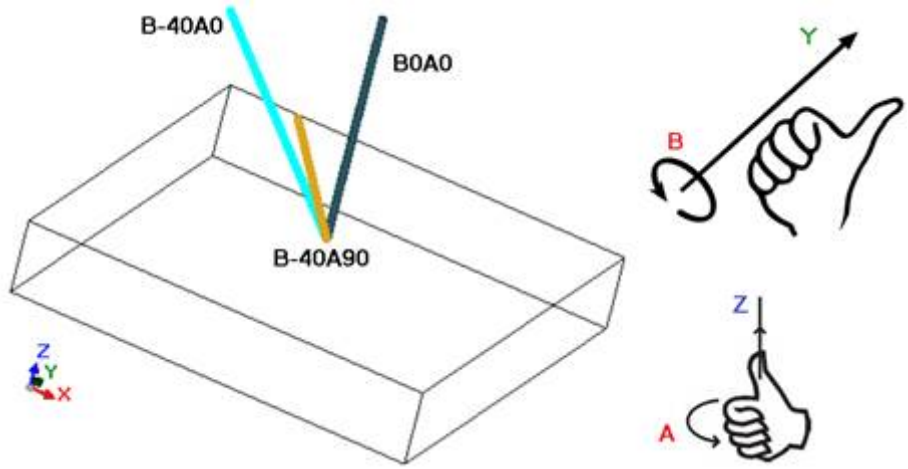
The figure represents a generic piece and the **three absolute Cartesian points**:
- beta rotates around the Y axis
- alpha rotates around the Z axis.

The picture shows the default situation. Following TpaCAD configuration:

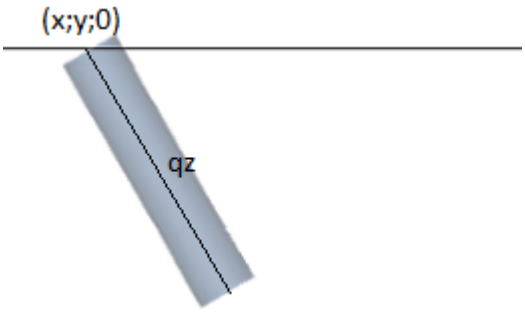
- the value of the rotation set for the *beta* axis can have the opposite sign; or
- the beta axis can turn around the X axis.

The picture shows setup workings that are programmed on face 1:

a. a first setup with (B0;A0)=vertical direction
b. a second setup rotates B anticlockwise
c. a third setup adds a clockwise rotation for A



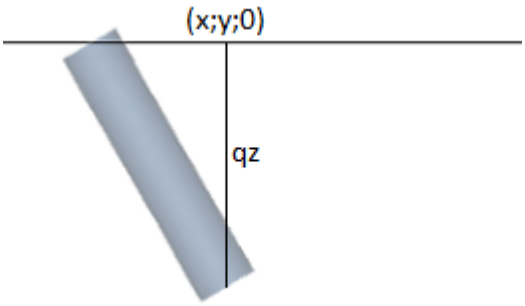
In a setup working that assigns the tool orientation, the XYZ coordinates are significant with respect to the non-oriented setup case.
In fact, it is generally possible to change the programming modes of the application point (coordinates in XY and Z plane for the depth axis) with parameter setting **Orthogonal Plane Z Ref**.



If the parameter **Orthogonal Plane Z Ref** is not selected:
- the X and Y coordinates program the entry point of the tool on the face plane
- the Z coordinate, programmed for the depth, is measured along the tool oriented axis.

The depth is significant with the sign:
- positive value moves the drill bit from the programmed XY position along the opposite resultant of the angles (alpha;beta);
- negative value moves the drill bit from the programmed XY position along the resultant of the angles (alpha;beta).

If the (alpha;beta) angles are set correctly, so that the tool can be placed at the entry of a face:
- positive value takes the tool over the piece;
- negative value takes the working tool into the face.



The tool enters the face plane at the programmed XY coordinates, with direction assigned by the rotation angle and rotation axis around Y axis, in accordance to the programmed depth.

The figure shows the use of the tool into the piece, with side view with respect to the top face plane.
The forthcoming profile respects the assigned orientation on the setup.

This case corresponds to the default situation, which is applied to the programming in an oriented setup, even if the parameter **Orthogonal Plane Z Ref** is not assigned for the setup.

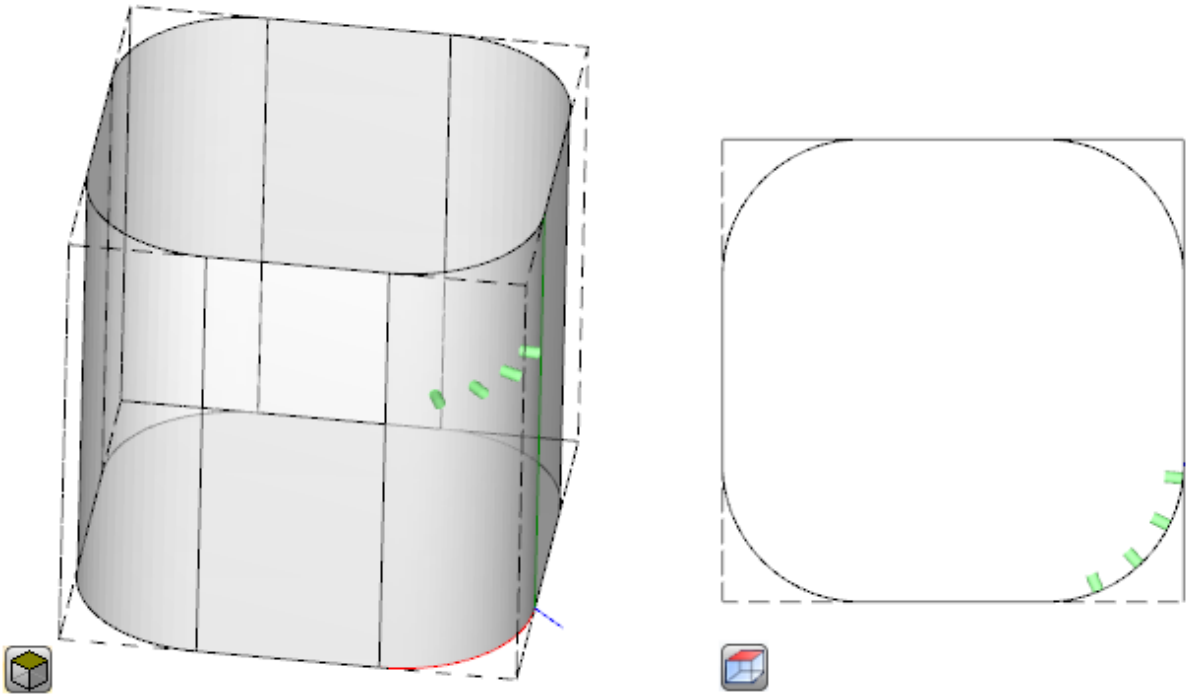
If the parameter **Orthogonal Plane Z Ref** is selected:
- the X and Y coordinates program the point on the face plane that corresponds to the final position of the drill bit;
- the Z-coordinate, programmed for the depth, is measured along the plane orthogonal to the face (depth axis of the face).

The tool enters the face plane with assigned direction, in P' point, so that the assigned position on the three programmed coordinates is observed. The P' point is automatically calculated, while the programming relates to the bit of the tool.

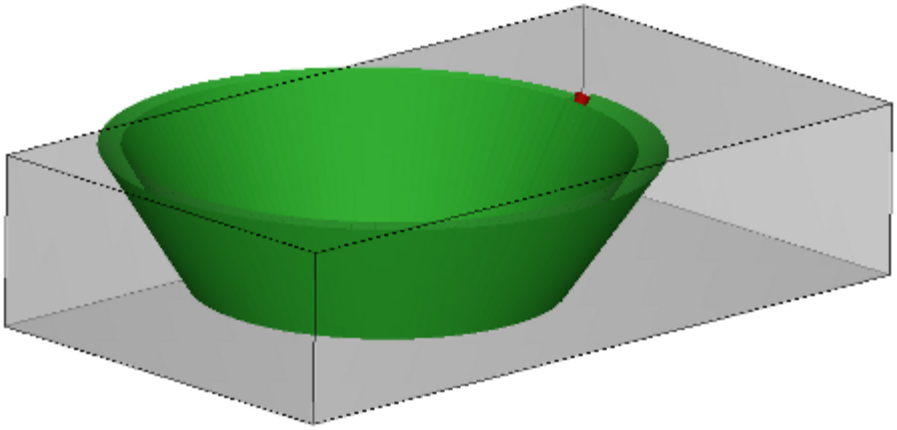
This programming mode is taken on by default in case of setup carried out in a curved face or surface.

Angles (alpha, beta) can be easily programmed by means of:
- parametric forms (**geo[alfa;x;y;z]; geo[beta;x;y;z]**) that return the angles (alpha, beta) corresponding to the vertical direction of the face, but only if it is plane. In case of a curved face or of a surface, the functions calculate the direction of the face with null radius of curvature;
- parametric forms (**geo[alfa]; geo[beta]**) that return the angles (alpha, beta) corresponding to the vertical direction in a specific point of the face, including the cases of curved face or surface;
- selection of **Vertical direction**: to be selected to assign the direction vertical to the face regardless the assignment: plane, curved or surface. The calculated values of the angles (alpha, beta) are not automatically assigned in the corresponding fields, but they are determined for the graphic representation of the working.

In the figure, there is an example of an isolated setup programmed in a curved face: on the left, a three-dimensional representation of the piece; on the right, the view from the top. It is clearly shown that the direction perpendicular to the face changes when the position along the axis of curvature changes.



Programming an oriented setup can enable the mode of **Tangent tracking**, that corresponds to the request to keep the axis perpendicular while carrying out the profile. This mode is used to carry out some non-vertical milling (the axis of the tool is parallel to the XY face plane or to the inclined plane, but not perpendicular to the XY plane) with the need to keep the rotation axis of the tool perpendicular to the profile to be carried out.
In figure, there is an example of milling on the top face of the piece, with the tool that corrects the inclination along the profile.



If the **Tangent tracking** selection verifies an orientation of the tool perpendicular to the face, the same corresponds to the request to keep the axis of the tool perpendicular while carrying out a profile, even when the place of the face changes and in accordance with the following instruction of the case of a vertical setup.

The **Tangent tracking** selection can be associated to the programming of a vertical setup, if it is being carried out on a curved face or on a surface. In these cases, the selection corresponds to requiring that the direction of the tool is always kept perpendicular to the XY face plane, plane now generally variable. Working on geometries of variable face, we can say that a setup is always to be meant as programmed with oriented geometry.
If we now consider the case of a curved face:

- if the selection is active, along the profile, the tool changes direction, so as to remain in the direction perpendicular to the face plane
- if the selection is not active, if selection is not active: the setup is carried out with the tool perpendicular to the face plane and such a direction is maintained along the whole profile.

According to the TpaCAD configuration and/or to a selection in the setup working, the profile can be interpolated in **Tangent tracking** by an interpolation at 4 or 5 axes:

- the 5-axis interpolation mode assumes that both the rotary axes can be repositioned while working

- the 4-axis interpolation mode places the rotation of the head on two rotary axes; however, later relocations can affect the (alpha) axis only, while the position of (beta) remains unchanged.

Interpolation possibilities depend on the physical machine configuration, even more than on the installed functionalities.

Technological priority

The assignment may be available for each working that interprets a technology assignment, therefore, of a point or setup type.
Information interprets a positive integer value (default value: 0) and is available for use in optimization, for the purpose of program execution.

The details of interpretation and use of the information may vary depending on the specific application, but the general terms meet the need to establish criteria of reciprocal priority of execution between similar workings.
The case of primary use concerns the sorting of the execution of the programmed profiles, so as to optimize the tool change operations. It is possible that several profiles use the same technology, but that some should be processed after all the others, regardless of the technology assigned for each profile.

The programming of the **Technological Priority** can solve the problem, placing in order of execution:
- first all profiles with *Technological priority = 0*, with or without groupings optimized for tool
- then the profiles with *Technological priority = 1*, with or without groupings optimized for tool
- ..
- until exhaustion of the profiles.

Graphic representation

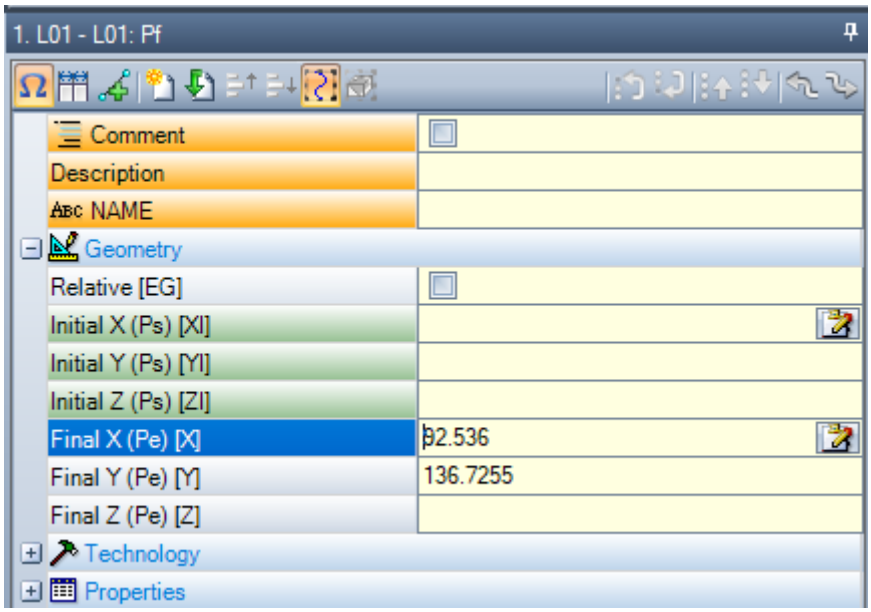
Point and setup workings are represented in bidimensional graphic face by a circle whose diameters is equal to the diameter of the programmed tool; in three-dimensional graph, they are represented by a cylinder whose diameter is equal to the diameter of the programmed tool; its height is equal to the depth overall dimensions of the tool in the face.
In three-dimensional graph of a setup with oriented tool, the tool is represented oriented along the rotation angle and rotation axis around Y axis.
A programmed working with a tool with multiple drill bits is represented by only one circle, whose diameter is equal to the diameter of the first head drill bit.

9.2 Profile

Profile workings

The elementary workings that can take part in the construction process of a profile are located in the **Workings** tabs of different groups:
- Individual segments of lines: they calculate a linear segment;
- Individual arcs: they calculate an arc in the XY face plane;
- Chamfer and fillet: they calculate two linear segments or a linear segment and an arc;
- Multiple arcs: they calculate two or more arcs;
- Circles: they calculate a circle in the xy face plane;
- Arcs in xz, yz, xyz planes: they calculate an arc in the xz, yz, xyz planes;
- Polygons: they calculate a normally closed figure that corresponds to a polygon (rectangle, triangle, hexagon) or to a conic section (oval or ellipse);
- Path (see next paragraph).

All the workings available in these groups have a significant interpretation for the TpaCAD program: each of them calculates a precise interpretation of the programmed geometric information. Generally, we say that each of these workings defines *a profile segment*, with a generic characterization of *linear segment* or *arc*.
The figure shows the parameters for the geometric assignment of the simplest segment linear profile (L01):



Let us examine the assignment of two edge points of the segment:
- Ps (XI;YI;ZI): start point of the segment. The point is normally available in each profile working, but it is only programmed to assign an open profile (that is a profile starting without a setup): normally, the start point of a profile segment execution is automatically defined by the final execution point of the previous segment.
- Pe (X;Y;Z): end point of the segment. The point may not be directly programmable (in some coordinates or partially), according to the geometry of each profile working and in this case, it is automatically defined.

Profile building

A profile is generally defined by a continuous sequence of linear and/or circular segments. This sequence is not necessarily opened by a setup working.
During the execution of the program, the tool selected remains in use from the start point of the profile until the end point, without break. TpaCAD recognizes the profile assignment:
- as a whole geometric development among one or more profile segments;
- with hook of profile parts defined in separate way (hooks of profile parts with subroutines or macros);
- as development of the application of a subroutine or macro.

A profile can start with:
- a setup working that assigns the general associated technology to the execution of the profile itself;
- a profile segment and in this case, it is also called **open** profile. The profile technology can be assigned later, still in editor phase or directly while processing a program for execution purposes.

A profile is considered open if one of the two situations is verified, as follows:
- in the profile segment (arc or line), also only one of the assignment parameters of the start point of the segment is set;
- before the segment of the profile, neither a setup or another profile segment is assigned.

Application point

The profile workings have the application point in the end point. In case of a multiple segment, the application point is the end point of the last calculated segment.

Each profile code calculates a specific geometry on a plane.
Let us consider some examples of different possibilities:
- **L2 [xy(pole, U, A), Zf]**: calculates a linear segment in the space assigned in two geometric components:
  - XY plane: linear segment defined in a polar system;
  - Z direction: with individual component, perpendicular to the face plane.

  The working calculates a linear segment in the space, where each axis has a linear movement.
  The depth axis is Z.

- **A4 [xy(P1,Xf,Yf),Zf]**: calculates an helix assigned in two geometric components:
  - XY plane: circular segment defined in a Cartesian system as an arc through three points;
  - Z direction: with individual component, perpendicular to the face plane.

  The working calculates an helix in the space, with helix axis parallel to the face plane and circular development assigned in the (XY) face plane.
  The depth axis is Z.

- **A5 [xz(Xf,Zf,centre,rot),Yf]**: calculates an helix assigned in two geometric components:
  - XZ plane: circular segment defined in a Cartesian system as an arc through three points;
  - Y direction: with single component perpendicular to the XZ plane.

  The working calculates an helix in the space, with helix axis parallel to the Y axis of the face and circular development assigned in the (XZ) face plane.
  The depth axis is Y.

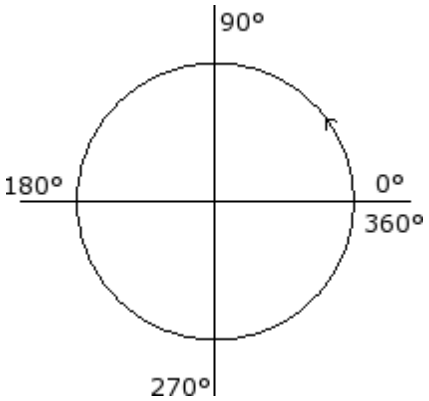
- **A9 [xyz(Xf,Yf,Zf,centre,rot)]**: calculates an arc in the plane generically oriented in the space:
  - no depth axis is assigned.

As already said, usually each profile section can directly assign also the section start point. In this way, the section directly opens a profile. If the profile section does not assign the start point, it is located on the application point of the working assigned before.

Programming the angles

The profile codes often use angle settings:
- The angles are programmed in degrees and decimal degrees (x.xx°)
- The note used is shown in figure: 0° to 360° with counterclockwise rotation.

Negative angles cover the XY plane from the X axis and clockwise rotation.



Tangent lines and intercept lines

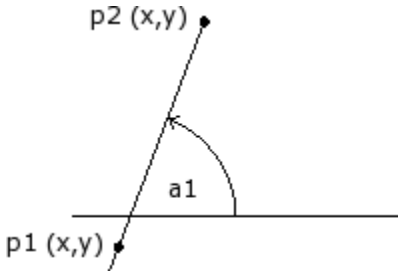
They are geometric elements used in profile codes.

The **tangent line** is a line that sets the tangent condition to programmed profile section (line or arc). It can be:
- an entry tangent line: if tangent is set on section start point;
- an exit tangent line: if tangent is set on section end point.

The **intercept line** is a line that sets the belonging condition for the application point (section end point) to the line itself.
An intercept line can also set tangent conditions on section end point.

With a single linear section, there is no difference between entry and exit tangent lines.

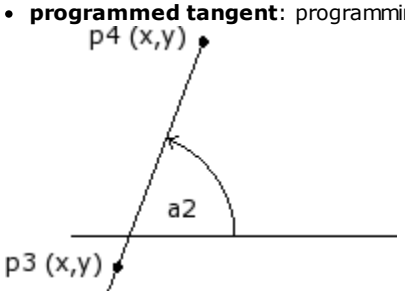
An **entry tangent** is defined as:
- **default tangent**: if set equal to the exit tangent line of the previous profile section
- **programmed tangent**: programming is required for:
-



  - inclination angle of the line (a1); or
  - of the two points (P1 and P2) on the line. Line orientation is defined from P1 to P2.

The angle programming prevails on point programming.

An **exit tangent** is defined as:
- **default tangent**: only with circular section ending on profile setup point. It is set up like initial tangent line of first profile section;
- **programmed tangent**: programming is required for:



  - inclination angle of the line (a2); or
  - of the two points (P3 and P4 on the line). Line orientation is defined from P3 to P4.

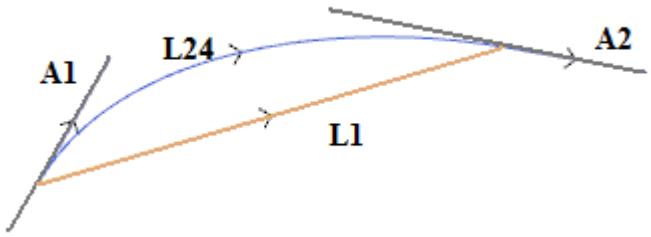
Angle programming prevails on point programming.

An **intercept line** is always directly programmed. Programming can occur with:
- the angle (a2) and a point on line (P3); or
- the two points (P3 and P4) on the line. Line orientation is defined from P3 to P4.

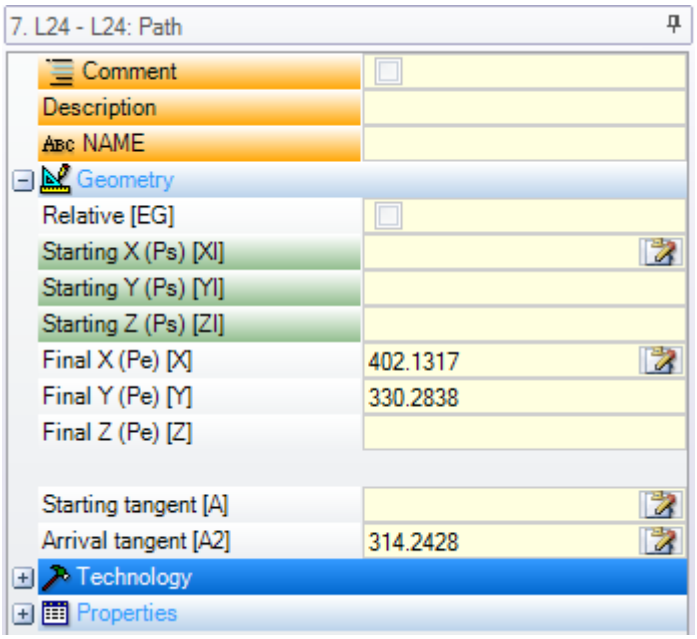
Path

This term *Path* refers to:
- a particular working contribution to the profile definition. The working code is L24;
- a profile made of elements of path type.

More specifically, the term *Path* is given to one of the curve options that can be selected by means of the Spline curve generation tool.
The figure shows the interpolation associated to a L24 code:



L1 is the linear segment that joins the extreme points of the segment (the segment is the comparison term with the generated curve).
L24 is the generated curve that corresponds to the L1 segment:
- **A1** is the starting tangent of the L24 curve;
- **A2** is the closing tangent of the L24 curve.



The L24 working sets the geometric information on:
- the start point of the segment (if it does not continue a previous path);
- end point of the segment;
- starting tangent;
- arrival tangent.

If the starting tangent is not set, it takes:
- previous arrival tangent, on the profile;
- the direction between the extreme points of the segment, if the arrival tangent is not significant.

If the arrival tangent is not set, it takes:
- the direction between the extreme points of the segment.

The two fields of tangents assignments can be modified in an interactive way by selecting the button associated to the field.
The L24 working calculates:
- only one linear segment (L1), if the start and arrival directions coincide;
- a continuous curve (L24), sampled by a sequence of linear micro-segments. The length of the segments is automatically evaluated and the number of the sampled segments is generally high: the theoretical curve, in fact, has a variation of continuous curving and the sampling in micro-segments, however thick they may be, it gives anyway an approximate solution.

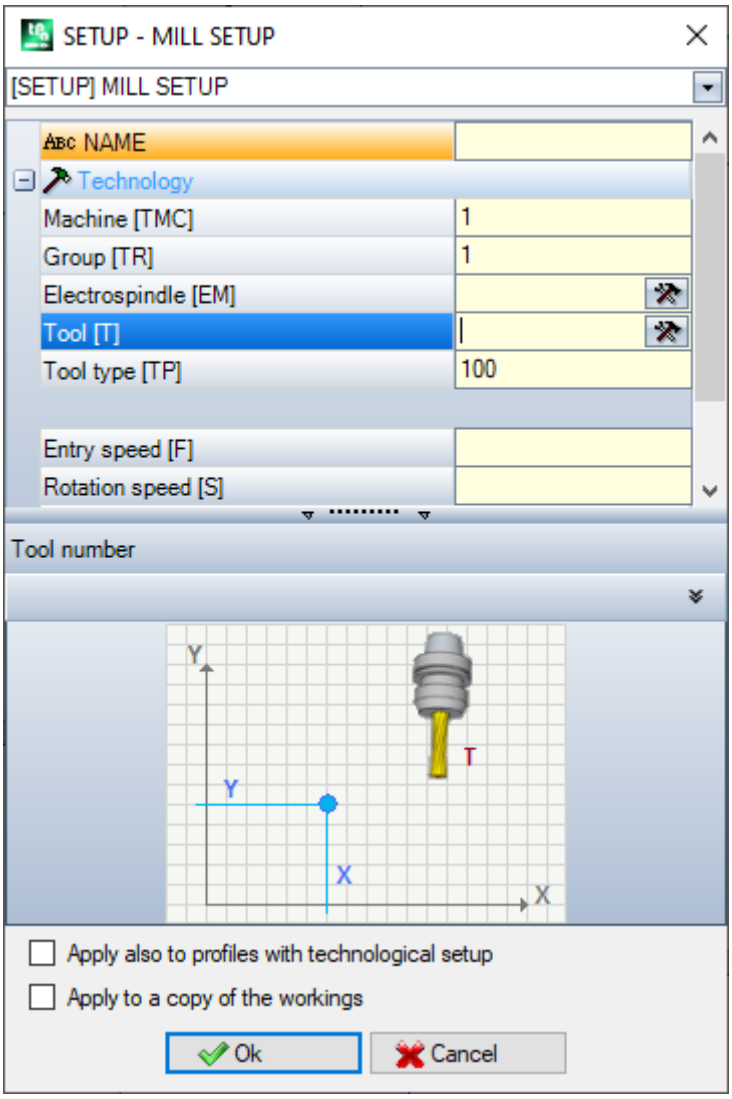
The L24 working is expanded in the list of the micro-segments, that calculate only in a solution of some specific tools.
The application to the L24 workings of advanced tools can be restricted, due to the specific nature of the working itself and may not be normally selected from the working palette: in this case, it can only derive from the *Spline generation tool*.
We can definitely say that the application of advanced tools in the process of handling the curve profiles in *Path* is not considered as a normal rule.

Assigning the technology

It is possible to assign the technological parameters of a profile by entering a setup working at the opening of the profile itself. The profile setup is not necessarily visible. For example, if the profile is fully or partially defined in the application of a subroutine (or macro), the setup can be internally applied to the development of the subroutine.
A profile without setup has been called **open** and it has no expressly assigned technology. In any case, during the execution, the profile always starts with an opening setup and relative assigned technology. We are talking here about the default technology as assigned in the dialog box opened from Application **Customize -> Technology -> Default codes**.
Therefore, although the possibility to manage open profiles simplifies the programming procedure, it must always be clear what a program is intended to apply, during the working. If the profile building requires a different technology from the default one, it is the task of the programmer to assign it directly.

For the working of open profiles, it is always possible to select between different options, as assigned by the machine manufacturer in TpaCAD configuration:
- the profiles programmed as open can always be excluded from the execution (for instance: in the same way as the construct workings);
- the profiles programmed as open can be normally executed, subject to the assignment of the default technology;
- the programming of open profiles can generate an error condition and therefore the program cannot be executed. In this case, the operator should directly assign the technology to each profile.

How to assign the technology to a profile
Technological data can be set by changing the profile manually (inserting and/or changing its setup) or by recalling **Apply setup** of the group **Change profiles** in the **Tool**  tab.



In the window above, the setup code to be assigned is first selected in the window among those in the list (in figure: [SETUP] MILL SETUP) and the list of parameters is updated with the data related to the selected working. Then, set the technological parameters and the working properties and confirm by pressing **[OK]** to apply the assignment.
As highlighted in the figure, it is not possible to assign here the geometric parameters of the setup working.
The technological parameters of a setup working do not only concern the choice of (Machine, Group, Electrospindle, Tool), but also the group of the parameters defining specifically:

- the Tool compensation modes;
- the Profile opening and closure modes.

These aspects are examined in the following paragraph and, as it will be evident, they can considerably modify the final product development.

In the window of the tool, two options are available, as follows:

- **Apply also to profiles with technological setup**: this option also applies the tool to profiles already open with a setup working. If the option is not selected: the tool is applied to the open profiles only or to those beginning with a GEOMETRIC SETUP working (these profiles can result from a format conversion).
- **Apply to a copy of the workings**: this option applies the tool to a copy of the workings and does not change the original lines.

The assignment is applied to profiles with less than a selected element or to the current profile (if no selections are available). Anyway, the application is restricted to profiles that verify the active view filters: selections, logical conditions, levels, special filters. If the tool is applied directly to original profiles, (selected or current) the modification cannot be applied to workings in a locked status (layer, construct, locked O field).

Multiple setups

The profiles to which several setup workings are suitably assigned are defined setups or multiple setups. During the profile building, the profile execution is repeated for a number of times equal to the number of the programmed setups:

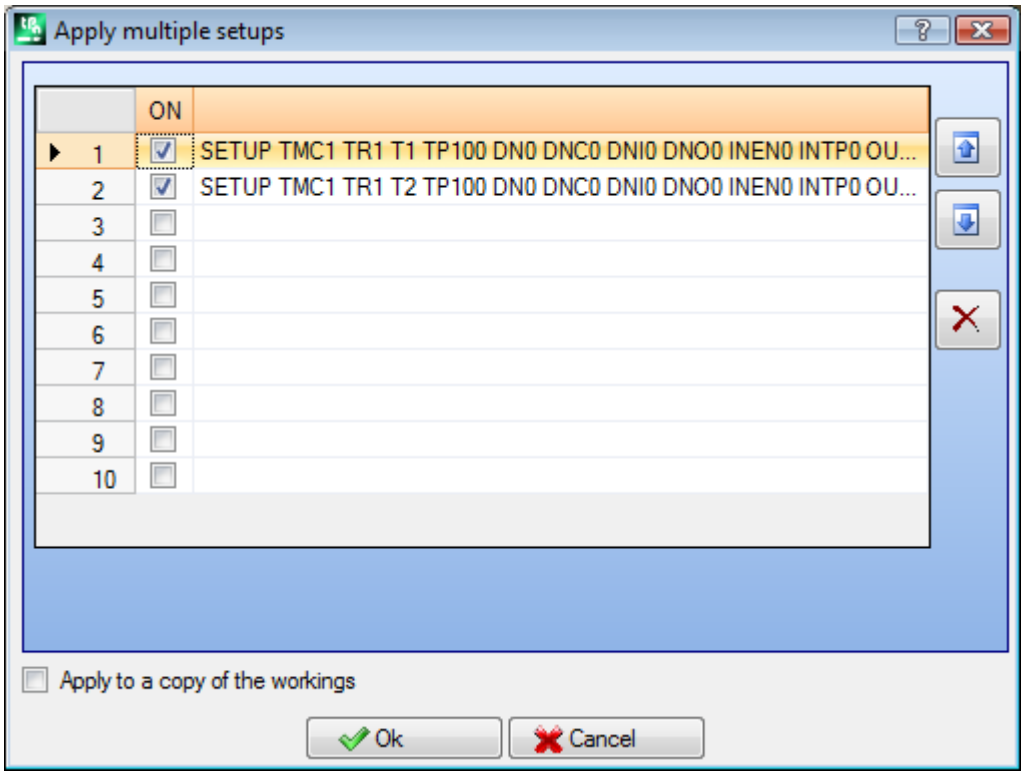
- the profile is executed for the first time with the first setup with the assigned technology;
- the profile is executed for the second time with the second setup and its assigned technology
- and so on for all the other assigned setups.

In this way, it is possible to duplicate the execution of a profile without having to program it several times, even with very different technological assignments. Once again, we emphasize that the technology of each setup defines on the whole:

- the choice of (Machine, Group, Electrospindle, Tool);
- the Tool compensation modes;
- the Profile opening and closure modes.

In TpaCAD and in a multiple profile environment the first setup is only seen while for the following setups the **Point hook** procedure is applied, which make the setups transparent, during the execution of the profile. For example, if the application of tool radius compensation is required, the profile is compensated in accordance with the technology assigned to the first setup.

The technological data of the setups can be set by changing the profile manually (inserting and/or changing its setups) or by recalling the command **Apply multiple setups** of the group **Change profiles** in the **Tools** tab.



The window shows a table with 10 rows. Each row can assign a setup with the same procedure considered for the application of an individual setup. To enable a setup, select the corresponding box in the ON column: a selection window is opened (setup working, technology); to modify an already assigned setup: double-click (or short-cut F2) on the right cell of the concerned row to open the assignment window; to disable an already assigned setup: remove the selection for the corresponding box in ON column.

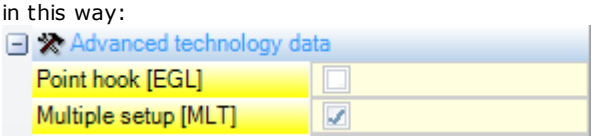
The order of the rows in the table reflects the order of the profile setup assignments: to move a setup, use the buttons on the right side of the table.

The option **Apply to a copy of the workings** is available and applies the tool to a copy of the workings and does not modify the original lines.

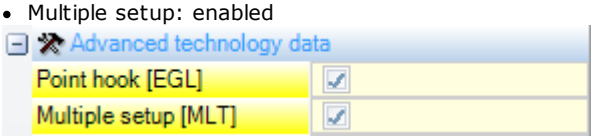
The criteria to detect the profiles concerning the application of the tool are the same as those of the command **Apply setup**.

The assignments now are always applied even to the profiles with already assigned technology: apply the tool also to profiles already open by a setup working or multiple setups.

With assignment of a multiple profile, the pages of the **Advanced technology data** of each setup are modified in this way:



First setup
- Point Hook: not enabled
- Multiple setup: enabled



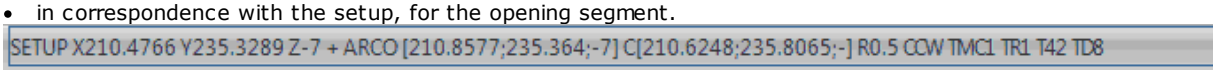
Following setups:
- Point Hook: enabled
- Multiple setup: enabled

Opening and closing a profile

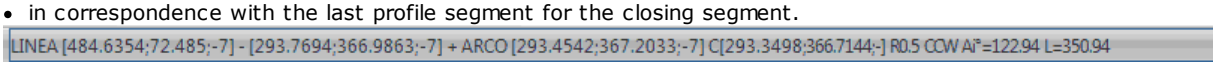
In a profile setup working, it is possible to assign how the profile should be opened and closed.

It is possible to add an opening and closing segment and select its typology (linear segment or arc), length and depth variation of the segment.
- the opening segment moves the setup point in relation to the programmed position.
- the closing segment is performed after the last programmed segment of the profile.

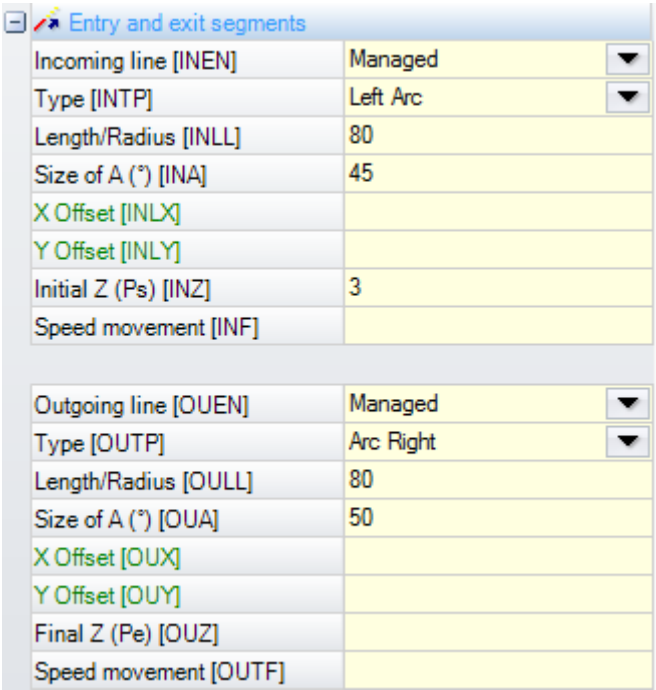
The opening and closing segments are not generated for isolated setups and are always displayed in tool compensation view, while the display without such active view is optional. If displayed, the geometric information of the opening and closing segments are shown in the status bar:
- in correspondence with the setup, for the opening segment.



- in correspondence with the last profile segment for the closing segment.



The parameters that assign the opening and closing segments are grouped in a setup node:



For the opening segment, you can select five **types**, as follows:

- **line**: linear segment, calculated in tangent continuity
- **arc left**: arc in the xy plane on the profile left side, calculated in tangent continuity
- **arc right**: arc in the xy plane on the profile right side, calculated in tangent continuity
- **arc 3D**: arc in oriented plane, calculated in tangent continuity
- **approach**: two linear segments on which the movement along the depth axis and the one in the face plane are sorted. The movement along the depth axis is carried out first, then the movement in the XY face plane. The development of tangent continuity is not guaranteed: if the condition is not verified, the selection of the typology is not applied in case the tool compensation is required.

For the closing segment, you can select six **types**, as follows:
- **line**, **arc left**, **arc right**, **arc 3D**: developed like in the opening segment
- **removal**: two linear segments on which the movement along the depth axis and the one in the face plane are sorted. The movement in the XY face plane is carried out first, then the movement along the depth axis. The development of tangent continuity is not guaranteed: if the condition is not verified, the type selection is not applied if the tool compensation is required.
- **coverage**: can be used only if the profile that finishes in the same setup point (closed profile) covers a portion of the first segment of the profile. The development of tangent continuity is not guaranteed: if the condition is not verified, the selection of the typology is not applied, in case the tool compensation is required.

The values that can be set for the Line type are:
- Length/Radius: length of the segment in the face plane. The programmable minimum value is 50*epsilon. If the entry segment and the exit segment are both enabled, but for this last one no value has been assigned, the value set is propagated from the entry segment to the exit one.

The values that can be set for the **Arc (left, right, 3D)** types are:
- **Length/Radius**: radius of the arc. The programmable minimum value is 50*epsilon. If both the entry segment and the exit segment are enabled, but for this last one no value has been assigned, the value set is propagated from the entry segment to the exit one.
- **Size of A (°)**: angle width of the arc. If the value is not set, the default value is 45°. Minimum value is 1°, maximum value is 270°, if the arc lays on the xy plane, otherwise the maximum value is 90°. If both the entry segment and the exit segment are enabled, but for this last one no value has been assigned, the value set is propagated from the entry segment to the exit one.

The values that can be set for the **Approach/Removal** types are:
- **X Offset**, **Y Offset**: they set the Offset of the two coordinated axes. The value set are summed to the respective coordinates of the setup or of end point.
- **Length/Radius**: length of the segment in the face plane, used if both the previous values are null (both less than: 10*epsilon); in this case, the segment is calculated in tangent continuity. The programmable minimum value is 10*epsilon. If the entry segment and the exit segment are both enabled, but for this last one no value has been assigned, the value set is propagated from the entry segment to the exit one.

The values that can be set for the **Coverage** type are:
- **Length/Radius**: length of the segment in the face plane. If the value is not set, the length of the initial segment of the profile is used.
- **Movement speed**: it sets the speed interpolation on the segments. If the value is not set: the speed assigned on the first segment of the profile is used. If on the exit segment no value is set, the speed assigned on the final segment of the profile is used.

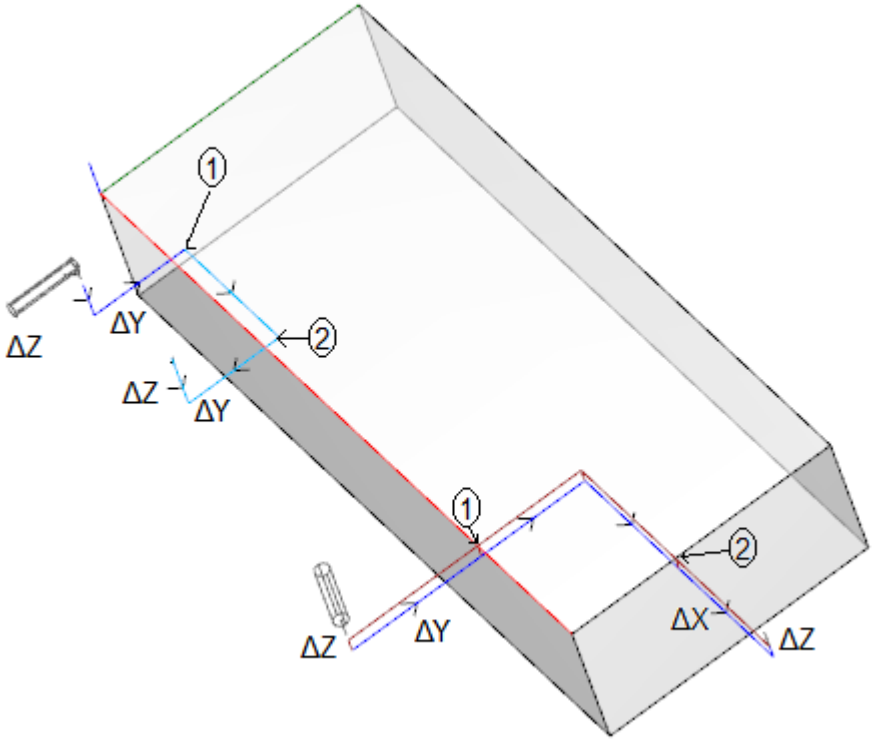
For the entry segment:
- **Initial Z**: sets the initial depth of the segment. The final depth of the segment is the assigned depth for the setup. Its programming is absolute and, if the value is not set, the default value is the value assigned to the **Qz** field (depth assigned to setup). Some clarifications are needed, if the selected typology is a **3D Arc**. First of all, the geometry of the segment depends on the starting segment of the profile. If the initial segment is an:
  - arc in xy plane carries out an arc on the xyz plane
  - arc in xz plane carries out an arc on the xz plane
  - arc in yz plane carries out an arc on the yz plane
  - linear segment carries out an arc on the xyz plane

  The value set for initial Z cannot be generally applied to the start point of the arc, because it is determined by the value set for the angle.
  The initial Z is significant, if only the radius of the arc is not set and it is taken from the value of the variation between the initial Z and the depth assigned for the setup. The variation sign between the initial Z and the depth assigned for the setup determines the solution of the resulting arc so as to enter from the direction set.
  If the initial Z is not set, following cases can be distinguished:
  1. if the profile starts with an arc, on the entry arc, a rotation sense opposite to that of the first profile is set.
  2. if the profile starts with a linear segment, on the entry arc, the entry direction of the coordinate over the piece is set.

For the closing segment:

- **Final Z**: sets the final depth of the segment. The initial depth of the closing segment is the final depth assigned for the profile. Its programming is absolute and, if the value is not set, the default value is the value of the final depth of the profile. If the selected typology is **Arc 3D**: considerations similar to those made for the entry segment shall apply, in order to determine the geometry of the segment and of the final Z.

The type of the **Approach/Removal** segment is useful when you need to control the movements that engage/approach or disengage/remove the tool from the piece. The figure shows two typical situations:



Both profiles are programmed from the top face:
- in one case, profile on the left, the tool is oriented horizontally, in such a way that it enters in a perpendicular way to the side face
- in the other case, profile on the right, the tool is vertical to the face.

The points indicated as **1** correspond with the position programmed for the setup of the profiles.
The points indicated as **2** correspond with the position programmed in end of profile.
In both cases, the setup programs the entry segment in Approach and the exit segment in Removal.

The profile on the left can correspond to the working of a simple cavity on a side face of the piece: the entry and exit segments perform controlled engagement and a disengagement on the overall dimension of the piece and of the work group.
The profile on the right can correspond to the milling of an angle that can be cut, if the milling is carried out beyond the depth of the piece. The entry/exit segments allow a correct engagement and disengagement of the profile outside the overall dimensions of the piece, controlling in this case also the positioning along the depth axis.

Hooking the profiles

A particular aspect in the definition of profiles is the possibility of linking them to each other. It is the option of **Point hook**, available as parameter of the setup workings and of the complex codes.
A point hook always needs **the application** of a relative programming of null displacements. Furthermore:
- if a profile element which can be hooked is assigned before the hook point (setup, arc, line, subroutine which ends its development with a profile element);
- if the active working is a setup working or a complex code, the profile before the hook point continues the profile after the hook point, without executing any of the intermediate setups. In this case, we are talking about **profile connection**.

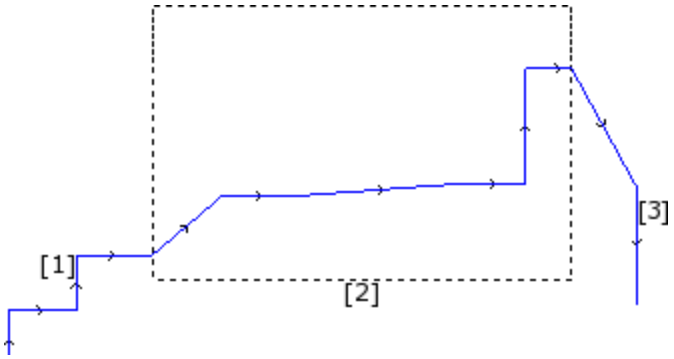
A profile resulting from the connection of different segments is a simple profile to all intents and purposes. The profile technology is usually assigned by the opening setup working. If no opening setup is assigned, also in this case we are talking about open profile.

Let us clarify the meaning of the expression "profile that can be hooked" or, better, which may be the situations that make a complex code (subroutine or macro-program) not to be hooked.
A first case corresponds to a complex code for which an exclusion in the working database is expressed. A typical example concerns the BLADE codes, whose execution does not allow the interpretation of any profile, but of only one linear segment.
Programming a final application point (see the paragraph Workings -> Subroutine -> Positioning a subroutine -> Final application point) excludes the possibility of hooking after a subroutine.

Simple profiles

Thanks to the point hook procedure, it is possible to continue a profile with parts assigned by the application of subroutines or macros. In any case, it is not said that this mechanism always allows to obtain a profile, where the tool selected for the profile building is used from the start point to the end point of the profile, without any disconnection. Let us see a first example:

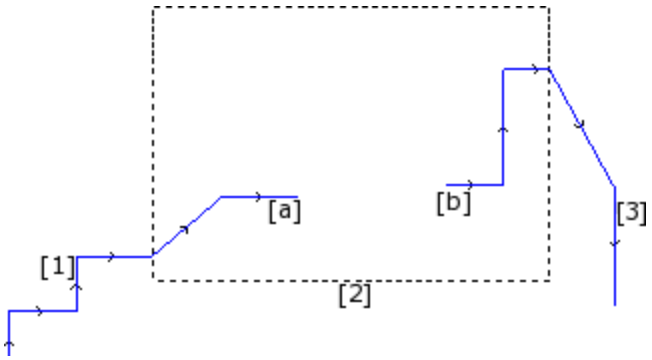
Let us see a **first example**:



The profile shown in the figure above consists of 3 parts:
- **[1]** the first part (on the left) is built with linear segments (it does not matter, if the profile is open or not);
- **[2]** the central part is enclosed in a rectangle: let us assume that it has been obtained by the application of a subroutine (in point hook);
- **[3]** the last part (on the right) is built with linear segments and finishes the profile.

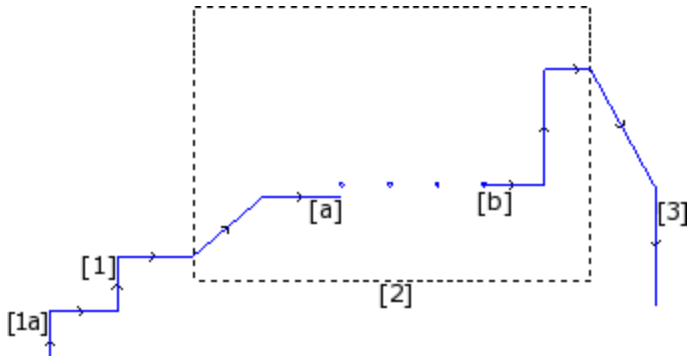
It can be stated that a profile has been built. The working tool remains engaged from the start point to the end point of the profile, without any disconnection.

Let us now see a **second example**:



The representation is similar to the previous one: the difference is that the central part of the profile shows a disconnection.
Is it still possible to say that a profile has been built?
The execution shows indeed two separate profiles:
- the first profile executes the first part **[1]** and continues to the point **(a)** shown in figure;
- the second profile starts from the point **(b)** shown in figure and continues to complete the final part **[3]**.

A **third example** is even more far from the idea of profile:



The subroutine marked with **[2]** now executes:
- in the initial part: a profile (which is connected to the previous profile **[1]**);
- in the central part: four drilling workings;
- in the final part: a profile (which is connected to the next profile **[3]**).

Logically speaking, the profile definition should only be applied to the first of the three examples examined above. In any case, there are some functions specific to a profile for which it is not at all important to make a distinction between the above-mentioned examples. If for example, it is necessary to apply a profile tool which assigns a particular technology to the profile which starts in (**1a**), it can be useful that the tool itself considers the set of workings as a profile, without taking into account how the **[2]** block has been defined: in this case, we are talking about profile defined in any case or complex or extended.
In the first of the three examples examined above, the profile is defined as simple: to all intents and purposes, the **[2]** block can be assimilated to a profile element. Therefore, a profile is said to be simple if it consists of simple profile elements (linear segments or arcs) and/or complex codes (subroutines or macros) which can be assimilated to simple profile elements.

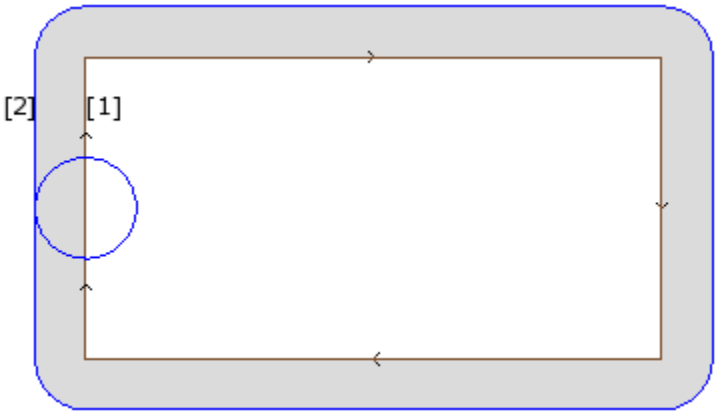
Tool compensation

The request for tool radius compensation activates a mechanism of automatic displacement of programmed trajectories (profiles), to keep into account the actual diameter of the tool which executes the same trajectory.

The command **Tool compensation**  to enable or disable the tool compensation is available in the group **Views** of the tab **View on**.
The compensation tool is applied in the xy plane and cannot be applied to arcs assigned on a plane different from xy, if:
- the original arc is a circle or the arc inverts the direction of the x axis or of the y axis
- the compensated arcs determine a fillet or an intersection solution internal to the segment.

The tool compensation is also applied to the construct profiles.

The following example shows how the tool compensation works:

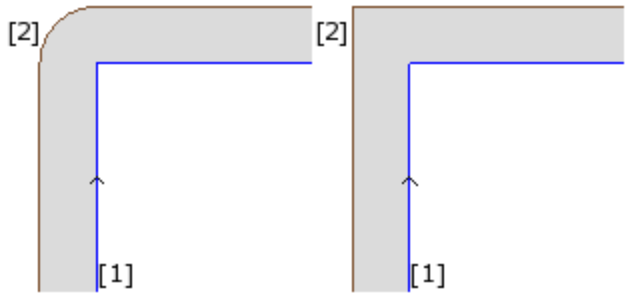


(1) programmed profile:
- rectangle followed clockwise;
- the small circle shown on the left vertical side of the rectangle displays the working tool diameter;

(2) profile obtained by tool compensation:
- it is external to the programmed profile and is followed in the same direction (clockwise);
- the distance between the two profiles is equal to the tool radius.

During the execution of the profile, the inner rectangle has the dimensions with which it has been drawn: according to the requested compensation, the tool works sure enough externally to the trajectory programmed.
If it were necessary to respect the dimensions external to the rectangle, the required compensation should be internal to the rectangle.

Let us see the detail of a rectangle corner of the example above:

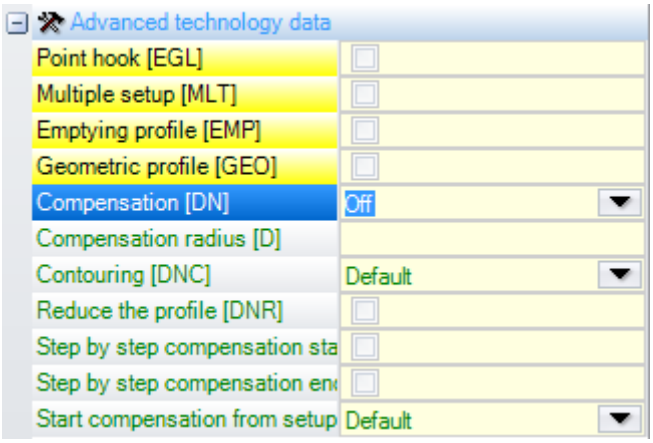


In the left figure the compensated profile moves around the original corner with a radius arc equal to the tool radius; in the figure on the right, the compensated profile continues up to the external intersection point of compensated linear segments.
In the first case, the compensation mode with insertion of **fillets** is applied.
In the second case, the compensation mode with insertion of **intersections** (otherwise called **contouring** compensation) is applied.
The compensation side is established by following the direction of the programmed profile. In the example in figure:

- the left side corresponds to the compensation outside the rectangle
- the right side corresponds to the compensation inside the rectangle.

Setting a compensation radius different from the tool radius allows to increase or to reduce the default compensation. The minimum recognized value corresponds to the epsilon resolution set in the configuration by the machine manufacturer. A setting value lower than epsilon is ignored.

The parameters for the execution of the tool radius compensation are assigned at profile and setup technology level and they may appear only partially, according to the TpaCAD configuration.



The parameters are grouped under the option **Advanced technology data**:

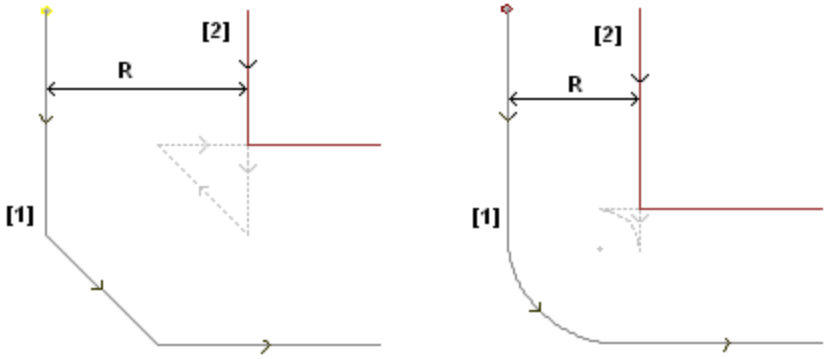
- **Compensation**: it enables the compensation, with direct selection of the compensation side. The listed entries are three:
  - **Off** disables the compensation
  - **Left** enables the compensation on the left side of the profile
  - **Right** enables the compensation on the right side of the profile
- **Compensation radius**: it sets the compensation radius, if it must be different from the tool radius. In TpaCAD configuration, a different interpretation of the value can be established.

- A typical configuration recognizes:

  ✓ the setting of the compensation radius, in case of programming without an initial sign. E.g.: "5", "r4", "prfi[12]/2"
  ✓ a variation to the compensation, to be added to the radius reported for the tool, in case of programming starting with the sign +/-. E.g.: "+2", "-2", "+r4".

- **Contouring**: it enables the compensation mode on the corners. The listed items are three:
  - **Default**: it enables the assigned default mode (in TpaCAD configuration)
  - **Fillets**: it enables the compensation with insertion of fillets
  - **Corners**: it enables the compensation with insertion of the intersections

- **Reduce the profile**: it enables the removal of the segments in the compensated profile, with respect to the original one, on the basis of geometric clearance restrictions exceeding the compensation itself. The figure below shows two typical situations, which can only be solved by enabling the profile reduction:



(1) programmed profiles,
(2) profiles obtained by tool compensation
R compensation radius

The figure on the left shows a profile portion assigned with a chamfer:
- the compensation is applied on the left side of the profile
- the value of the (R) compensation exceeds the chamfer dimension.

If the profile reduction is not activated, the profile compensation fails. An error due to compensation excess on the inclined segment appears.
The compensated profile (2) is obtained only if profile reduction is enabled: the intermediate segment does not appear; sure enough, it has been eliminated in the projection of segments, for the building of the compensated profile.
The dashed segments highlight which would have been the compensated profile, if the compensation applied to the intermediate segment were considered valid. It is clear that the direction of the intermediate segment would be inverted, with consequent alteration of the initial geometry.

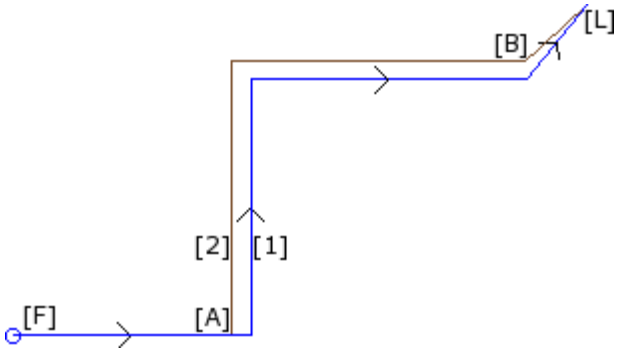
The figure on the right shows a profile portion assigned with a fillet:
- the compensation is applied on the left side of the profile
- the value of the (R) compensation exceeds the radius of the fillet.

If the profile reduction is not activated, the profile compensation fails. An error due to compensation excess on the arc.
The compensated profile (2) is obtained only if the profile reduction is enabled: the intermediate segment does not appear; sure enough, it has been eliminated in the projection of two contiguous segments, for the building of the compensated profile.
The dashed segments highlight which would have been the compensated profile, if the compensation applied to the intermediate segment were considered valid. Also in this case, the direction of the intermediate segment would be inverted, with consequent alteration of the initial geometry.

Profile reduction is only applied where it is necessary (that is in case situations such as those listed above happen) and can also delete several consecutive segments.
It is important to focus on how profile reduction does not take into account in any way the profile as a whole. When a segment must be deleted, a solution of intersection between segments respectively before and after the deleted segment is searched, without examining whether the intersection interferes with other parts of the profile. Therefore, it is recommend to enable reduction only if necessary and, in any case, to examine the compensation made, mostly when there are compensation values which far exceed the extents of the original profile.

- **Step by step compensation startup**: it enables the gradual compensation start on the first segment of the profile. Compensation is calculated from the second segment of the profile and movement on the first segment is linear: from the setup programmed point to the compensated start point of the second segment. In any case, Step-by-step compensation start is not applied if one of the following conditions is verified:
  - the first segment of the profile is not linear
  - the profile is defined by only one segment
  - the first segment of the profile needs a disconnection in compensation (see below).
- **Step by step compensation end**: it enables the gradual compensation start on the last segment of the profile. It is only applied if the last segment is linear. Compensation is calculated up to the last-but-one segment of the profile and movement on the last segment is linear: from the compensated end point of the last-but-one segment to the end point of the programmed profile. In any case, Step-by-step compensation closure is not applied, if one of the following conditions is verified:
  - the last segment of the profile is not linear
  - the profile is assigned by only one segment

- the last segment of the profile needs or continues with a disconnection in compensation (see below).



(1) Programmed profile,
(2) Profile obtained by tool radius compensation.

The profile applies:
- gradual compensation start ([F] which is the first segment);
- gradual compensation closure ([L] which is the last segment).

[A] is the compensated start point of the second segment of the profile;
[B] is the compensated end point of the last-but-one segment of the profile.

- **Start compensation from setup**: it enables compensation from the setup programmed point. The listed items are three:
  - **Default**: it enables the default mode (in TpaCAD configuration)
  - **Off**: it disables the compensation mode
  - **Apply**: it enables the compensation mode;

If the entry is enabled, the compensated profile starts from the setup programmed point to the compensation start point on the first segment with linear movement.

In any case, Start compensation from setup is not applied if one of the following conditions is verified:
- Step-by-step compensation Start is required and applied;
- the first segment of the profile needs a disconnection in compensation (see below).

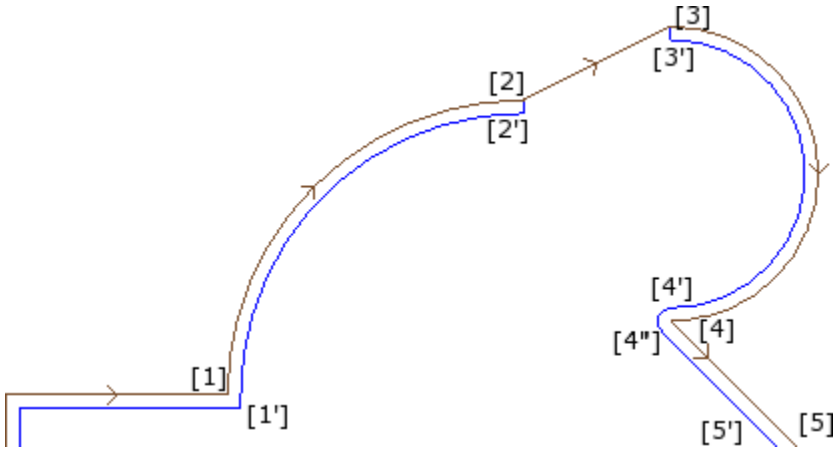
Start compensation from setup is generally used in applications to work on very hard material such as marble, when on the positions programmed for the setups some *pilot holes* are worked by special tools, from which the tool for the profile execution can easily start, without risks of rupture.

Variation of the compensation

The application modes of the tool can also be changed during the profile development. In a profile working, we can assign the **Compensation** parameter, showing up to 4 entries in list, as follows:

- **Unchanged**: the compensation continues unchanged with respect to the previous segment
- **Resume**: if interrupted or suspended, the compensation is restarted
- **Break**: the compensation is interrupted from the current segment to the next restart
- PROFESSIONAL **Suspend**: the compensation is suspended from the current segment to the next restart. This option is available in **Professional** mode only.

Let us examine the example of an application of compensation break:



The concerned profile part is programmed on the following segments:
- ..
- [1] -> [2] (arc)
- [2] -> [3] (line)
- [3] -> [4] (arc)
- ..

The programmed profile shows the direction arrows. The compensation is on the right side of the profile.
Let us examine the profile, where the tool compensation is applied:
- ([1] -> [2]) is compensated on the arc: [1'] -> [2']
- added linear segment: [2'] -> [2]
- original segment line: [2] -> [3]
- added linear segment: [3] -> [3']
- ([3] -> [4]) is compensated on the arc: ([3'] -> [4']) and the fillet ([4'] -> [4"]) is added before the compensation of the following segment.

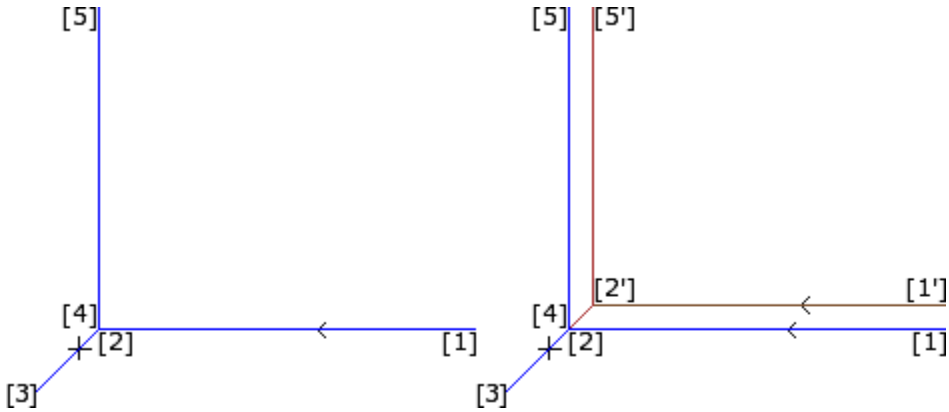
The compensation has not been applied on the linear (segment): [2] -> [3]. More specifically:
- the segment [2] -> [3] coincides perfectly even on the compensated profile;
- the compensation is stopped at the end of the previous segment (arc: [1] -> [2]), by defining the point [2'] as it was the last segment of the profile and adding a linear segment from [2'] to the point [2];
- the compensation is resumed from the following segment (arc: [3] -> [4]), by defining the point [3'] as it was the first segment of the profile and adding a linear segment from [3] to the point [3'].

The compensation as shown is obtained by using the **Compensation** parameter, assigned on the profile segments.
From the example above, we can infer that the compensated profile results from the following settings:
- [setup]: requests the compensation side: Right
- ...
- [1] -> [2]: **Compensation**: Unchanged
- [2] -> [3]: **Compensation**: Break
- [3] -> [4]: **Compensation**: Restart
- ...

It is possible:
- to set a compensation stop also on the first segment of the profile;
- a compensation stop not necessarily shall be deleted by a restart operation: it may carry on to the end of the profile.

The above-described example highlights an aspect concerning the compensation mode applied to corners. The profile setup has necessarily required the application of the compensation mode with research of intersections (Contouring: Corners): the intersection solution in point [1'] seems to highlight it.
But we have seen that compensation has inserted a fillet (arc: [4'] -> [4]) in point [4]: this because the compensation of the two segments which converge to point [4] has found no edge and, therefore, it has added a fillet.

An example of application of suspension in compensation concerns the working of door-frame corners.
The figure below shows the situation of an edge:



On the left, the programmed profile is displayed, with the following direction of segments:
- [1] -> [2]
- [2] -> [3]
- [3] -> [4]
- [4] -> [5]

The corner is on the two intermediate segments (2 -> 3), (3 -> 4). **ATTENTION**: points [2] and [4] coincide.
The compensation is on the right side of the profile.

The figure on the right shows what is necessary to obtain, with tool radius compensation applied:
- first compensated segment: [1'] -> [2']
- added linear segment: [2'] -> [3]

- added linear segment: [3] -> [2']
- last compensated segment: [2'] -> [5'].

The point [2'] is determined by intersecting the two compensated segments obtained in compensation from the two original segments, respectively before and after the corner: (1 -> 2) and (4 -> 5).

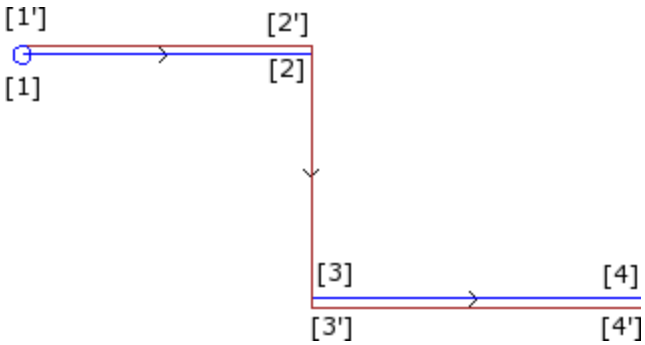
The compensation as shown is obtained by using the **Compensation** parameter, assigned on the profile segments.
From the above-described example of the frame, to obtain the compensated profile as shown in the right figure, you need to set each segment as follows:
- [setup]: requires the compensation side: Right
- ...
- [1] -> [2]: **Compensation**: Unchanged
- [2] -> [3]: **Compensation**: Suspend
- [3] -> [4]: **Compensation**: Suspend
- [4] -> [5]: **Compensation**: Resume
- ...

It is necessary that:
- a request for suspension is not deleted by a restart operation. A suspension to the end of the profile determines an error message in application of tool radius compensation
- the two segments before and after suspension are geometrically consecutive, that is, the first ends in the point where the second starts. Otherwise, a message error appears in application of tool compensation
- the compensation on the two segments (segments before and after suspension) can determine a condition of intersection (and not of fillet). Otherwise, a message error appears in application of tool radius compensation.

Variation of the side to compensate

PROFESSIONAL **Variation of side to compensate** (This option is available in **Professional** mode only.)
In a profile working, it is possible to select the parameter **Change compensation side**, that inverts the side of the compensation (from the left to the right or vice versa).
The activation of this selection is subject to limitations, as follows:
- the request may correspond to a resumption of compensation after an interruption; or
- the previous segments, corresponding to the request, may calculate an intersection of the compensated segments; or
- the previous segments, corresponding to the request, assign an inverted geometry.

Example of application for the inversion to compensate:



- [1] (setup) requires left compensation
- [1] -> [2] (segment): to compensate
- [2] -> [3] (segment): stops the compensation
- [3] -> [4] (segment): restarts the compensation and requires side change.
- ..

The same profile, where the tool compensation is applied:
- ([1] -> [2]) is compensated on the left on: [1'] -> [2']
- added linear segment: [2'] -> [2]
- original segment: [2] -> [3]
- added linear segment: [3] -> [3]
- ([3] -> [4]) is compensated on: ([3'] -> [4']) on the right side.

Display

The commands **Profile overall dimension in compensation** and **Original profiles in compensation** available in the group **Customize Views** of the tab **View on** modify the **View in tool compensation**.
Profile overall dimension in compensation: the compensated profiles and the profiles that do not apply any compensation are represented with a full segment whose overall dimension is equal to the tool extent. For these profiles, the edge points and the direction arrows are not represented.
In any case, the following items are represented with unitary thickness:
- construct profiles
- segments of profile built over the piece.

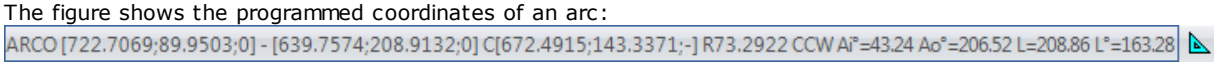
If disabled, compensated profiles are represented with unitary thickness.

Original profiles in compensation: if enabled, the view also shows original profiles (uncompensated profiles); otherwise, the view only shows compensated profiles and the profiles which do not apply any compensation (with direction arrows applied on the displayed segments, if required).

The status bar

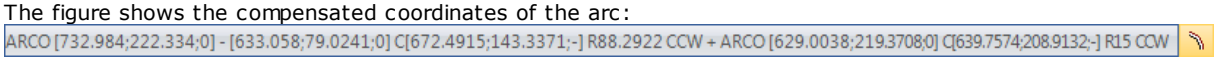
If the Tool radius compensation is activated, the coordinates relative to programmed segments or to those which have been compensated are displayed in the Status bar. To change, click on the picture on the right of the coordinate area.

The figure shows the programmed coordinates of an arc:



ARC	this writing indicates that this is an arc (the writing is managed by message files)
[722.7069;…]	position of the arc start point
-[639.7574;208.9132;…]	position of the arc end point
C[672.4915;…]	position of the arc centre point
R=73.2922	arc radius
CCW	counterclockwise rotation (CW if clockwise)
Ai°=43.24	start angle of the segment (in degrees)
Ao°=206.52	end angle of the segment (in degrees)
L=208.86	length of the segment (in 3D)
L°=163.28	angle of the arc (in degrees)

The figure shows the compensated coordinates of the arc:



ARC	this writing indicates that this is an arc (the writing is managed by message files)
[732.984;…]	position of the compensated arc start point
-[633.058;…]	position of the compensated arc end point
C[672.4915;…]	position of the centre of the arc
R=88.2922	radius of the compensated arc
CCW	indicates anticlockwise rotation
+	indicates that the compensation has added a segment after the arc
ARC	this writing indicates that this is a fillet arc (the writing is managed by message files)
[629.0038;…]	position of the fillet arc end point
C[639.7574;…]	position of the fillet arc centre point
R15	radius of the fillet arc
CCW	indicates anticlockwise rotation of the fillet arc

For the compensated segment, neither start and end angles nor length are provided.

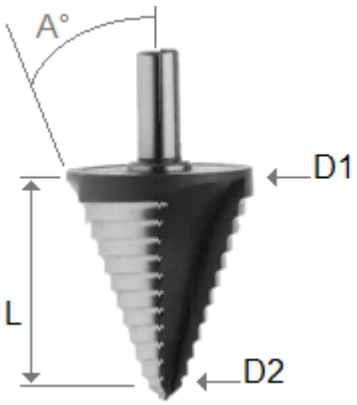
Executing a profile with sharp corner cut

In the setup of a profile, it is possible to require the change of the normal executing logic of a profile in order to achieve a sharp corner cut. This option concerns the execution of the corners of a profile between two profile segments.

The activation is set in the node **Advanced technology data**:

- **Sharp corner cut**: The system works, if some conditions are verified, as follows:
  1. the tool set on the setup is conical
  2. the depth programmed on the setup point engages the tool in the piece.

The figure shows an example of conical tool:



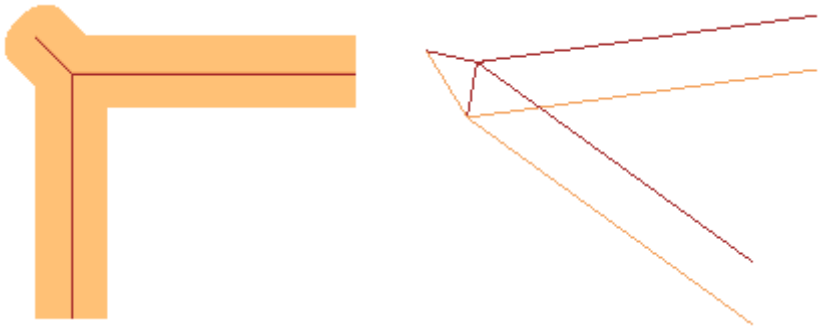
characterized by two extreme diameters and by the angle of the cone. Clearly, this tool hollows out the material according to its used part: the more the programmed depth increases, the greater is the diameter that works on the surface of the piece.

When the conditions are verified, two linear movements are added to the corners:
- rise of the tool up to null depth position and towards the theoretical corner of the cavity of the tool, on the outside of the corner (found by the part of the greater angle). The length of the segment is determined by the shape of the tool, by the geometry of the corner and by the depth programmed on the edge vertex.
- lowering tool that repositions itself on the programmed corner.

The final effect performs a sharper corner up to the maximum limit allowed by the tool.
In configuration, the minimum and a maximum angle of such a corner are defined, so that the addition of linear segments is applied: so, a range of values between 1° and 179° may be configured.
The corners made up by programmed ingoing/outgoing segments to a profile are excluded from the evaluation.
A typical application is used in frame working.

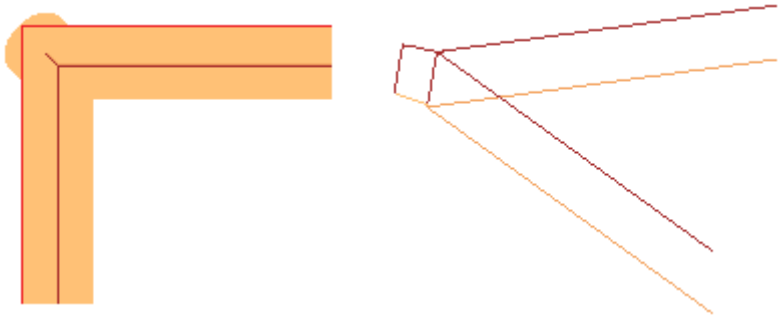
If the profile requires the Tool compensation and always assuming that the conditions required are met:
- the diameter used for the compensation is the reference diameter stated for the tool (in figure: D1)
- the corners of the compensated profile, obtained through intersection solution, are evaluated
- the corners made up by the programmed ingoing/outgoing segments to a profile are excluded
- the corners made up by additional linear segments to apply suspension, interruption and resumption of the tool compensation are excluded.

In the example below: application to a right-angled corner (90°) of the rising up movement:



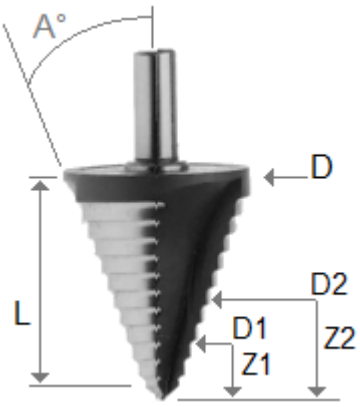
As clearly shown in figure, the rise on the corner is added to the "external part" of the profile. In a rectangular profile, finding the external part could not be more clear, but it is not always the case. A profile can show concave and convex parts and can be closed or not. For all these reasons, four possible options are associated to the **Sharp corner cut** field as follows:
- **Off**: selection excluded
- **Automatic**: automatically finds the external part of the profile. First of all, please check if the profile requires a tool compensation:
  - with compensation on the right, the external part is on the left of the profile, vice versa in the case of compensation on the left;
  - if no tool compensation is required and if the profile is closed, the direction of rotation of the same profile is calculated as follows: clockwise corresponds to the external left part; counter-clockwise corresponds to the external right part;
  - otherwise: the external part of the profile is determined on the geometry of the first useful corner found on the profile.
- **Left**: the external part is placed to the left of the profile
- **Right**: the external part is placed to the right of the profile.

Furthermore, in the configuration the above-mentioned functioning can be more widely applied, including also the case of the non-conical tool. In this case, the additional segments confirm the programmed depth on the corner and, even keeping the same direction on the face plane as the previous case, length is such as to stop the external overall dimensions of the tool at the theoretical corner.



Compensation of the correction diameter in the case of a conical tool

When using a conical tool, it is possible to apply the diameter compensation used in tool correction. The functionality must be enabled in **TpaCAD Configuration** and does not require the Professional mode.
The compensation takes into account the programmed depth on the profile setup, net of a possible entry section to the profile.



With reference to the figure:

- in case of depth of entry into the programmed piece Z1, the diameter marked as D1 is used
- in case of depth of entry into the programmed piece Z2, the diameter marked as D2 is used
- in case of depth of entry into the programmed piece from a value equal to or greater than the useful length of the tool (marked as L), the maximum declared diameter is used, marked as D
- in the case of programmed depth over the piece, the diameter D is used with respect to the piece.

If the option is not selected, the diameter D is always used.

Assignment of profiles in piece-face

Assigning profiles in piece-face requires some further details.
When configuring TpaCAD, it is possible to choose between two different operating modes:

- **Recognition of profile not conditioned by F**: in this operating mode, greater importance is given to the continuity of profiles than to the assignment of the application face. In the case of profile workings (arcs and lines), the recognition of open profile does not take into account the F field assignments related to the current and the previous workings:
  - if the segment opens a profile, it keeps its own original F field programming
  - otherwise: it propagates the F field from the current segment to the previous segments

  In the case of setup or complex working **requiring** a point hook, the F field is propagated from the previous working.

- **Recognition of profile conditioned by F**: in this operating mode, greater importance is given to the assignment of the application face than to the continuity of profiles. In the case of profile workings (arcs and lines), the recognition of open profile keeps into account the F field assignments related to the current and the previous workings and different settings cause the interruption of the continuity of the profile. F field propagation from the current to the previous segments is never applied. In the case of setup or complex working **requiring** a point hook: the F field is not propagated from the previous working, and the point hook does not determine the continuity of the profile, if the previous F field setting is different.

9.3 Logical Instructions

Logical instructions are particular simple workings to which no profile working correspond.
A logical instruction can assign the conditioned execution of one or more workings or execute itself a given function, by condition it or not according to the value of a logical expression (Example: ERROR).

IF ... ELSEIF ... ELSE ... ENDIF Structures

A logical instruction can be entered by recalling the **If.. EndIf**, **If.. ElseIf.. Else.. EndIf**, **If.. Else.. EndIf** command from the group **Blocks** of the tab **Apply** or by selecting the IF, ELSEIF, ELSE, ENDIF working in the group of the LOGICAL INSTRUCTIONS.



The **If.. EndIf** structure is the simplest programmable alternative form. The IF instruction expresses a condition that:
- if TRUE: determines the performance of one or more workings specified after the IF
- if FALSE: determines the non-performance of workings involved.

The ENDIF instruction delimits workings conditioned by IF.
The ELSE instruction can be assigned between IF and ENDIF and denies the condition evaluated by IF.
The **If.. Else.. EndIf** form can be paraphrased as: "if the condition expressed on **If** is valid, it performs the workings specified after **If**; otherwise, it performs the workings specified after **Else**".

A more complex form can be expressed as **If.. ElseIf.. ElseIf.. Else.. EndIf**, that can assign conditions alternative to each other: the first verified calculates the alternative form and it is possible that no condition is verified. If the complex form ends up with an **Else** level, the same is verified as a default alternative, if no condition before is verified.
The result of the logical conditions set in a program is visible by requiring the application of the logical conditions with the command available in the group of **Views** in the tab **View on**. In this active view, only the workings that verify the logical conditions are displayed.
The result of the logical conditions set in IF.. ELSE.. ENDIF cycles does not condition the interpretation of the workings, as programmed in face sequence. Let us clarify the point.
After a IF.. ENDIF cycle that executes a profile, a 100 drilling working on X coordinate is programmed: the X coordinate of the hole is determined by adding the programmed coordinate (100) to the profile end point within the IF cycle, **apart from** the verification on the logical condition for the IF instruction.

Sometimes, after inserting a logical block by selecting the command in the group **Blocks** of the tab **Apply**, the following message may appear:



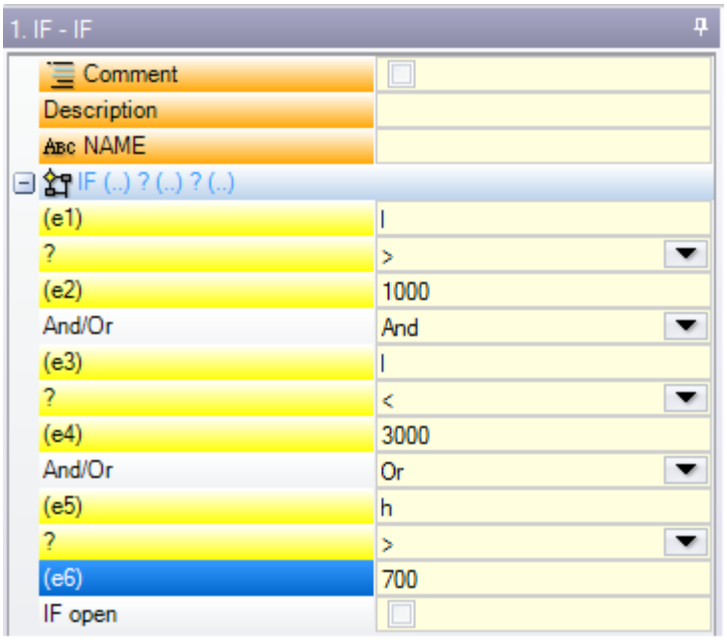
Choose **[Yes]** to insert directly into the block the group of the selections of which the current working is part. In the case as above (Insert If.. Endif), being on the line 5 and having selected the lines from 3 to 12:
- the IF instruction is inserted **before** the line 5
- the ENDIF instruction is inserted **after** the line 12

The logical status of the working is also shown in the ASCII text:

					ASCII Text			M
6	☐	☑		☐	IF ESP1=r\rot TST1=4 ESP2=0 LOG1=0 TST2=0 LOG2=0 TST3=...	0	0	0
7	☐	☑		☐	IF ESP1=r\arcoin TST1=4 ESP2=0 LOG1=1 ESP3=r\radin TST...	0	0	0
8	☐	☑		☐	SETUP EG0 X0 Y-r31 Zr14 TMCr20 TRr21 Tr22 TPifelse[rem...	0	0	0
9	☐	☑		☐	A42 EG0 EW0 I0 Jr31 UYr30*2 Nr13 TAr0 Ur4 ANr\narchirea...	0	0	0
10	☐	☑		☐	ELSE	0	0	0
11	☐	☑		☐	IF ESP1=r24 TST1=4 ESP2=1 LOG1=0 TST2=0 LOG2=0 TS...	0	0	0
12	☐	☑		☐	SETUP EG0 Xr26 Y-r31+r26 Z0 TMCr20 TRr21 Tr22 TPife...	0	0	0

in figure: IF level is verified in the IF.. ELSE.. ENDIF structure.

The condition expressed by IF and ELSEIF instructions can consist of three terms. Let us see an example:



(e1) ? (e2): first term

And/Or: logical condition between first and second terms
(e3) ? (e4): second term

And/Or: logical condition between the result of the second condition and the third term
(e5) ? (e6): third term

The **(e..)** fields appearing in a term have a general parametric setting.

The **?** element in the **(e..)** fields of a term assigns a comparison condition for:

<	strict minority	(example: (e1) < (e2))
<=	minority	(example: (e1) <= (e2))
>	slight majority	(example: (e1) > (e2))
>=	majority	(example: (e1) >= (e2))
=	equality	(example: (e1) = (e2))
<>	difference	(example: (e1) <> (e2))

A term is verified like TRUE, if comparison condition is complied.
ATTENTION: the comparisons between the **(e..)** fields are always evaluated below a minimum deviation equal to 0.001 (comparison epsilon): values differing by less than epsilon are considered as equal.

The logic condition between two relation terms has the following value:

And	if both terms must be verified as TRUE
Or	if it is enough, when only one term is verified as TRUE.

It is possible to set: none, one, two or three condition terms.
If no term is set for an IF: the corresponding level is always verified. In this case, when IF also assigns ELSEIF or ELSE levels, these are never verified.
A similar consideration is also applied to the programming of an ELSEIF. If the control evaluates the instruction (read: no level specified before for the IF cycle is verified), and no term is set, the corresponding level is verified and finishes the IF cycle development.

ELSE and ENDIF are fully feedthrough instructions: they have no assigned fields.

IF.. ELSEIF.. ELSE.. ENDIF condition structures can be nested without restrictions.
The programming shown in figure corresponds to the evaluation of a logic expression:

IF (((I > 1000) and (I < 3000)) or (h > 700)) {..} ENDIF

That is:
if (l) is higher than 1000 **and** (l) is also lower than 3000;
or: if (h) is greater than 700,
then IF instruction is verified as TRUE.

If: l=2000, h=500

(l > 1000)	TRUE
(l < 3000)	TRUE
(h > 700)	FALSE

evaluates: (TRUE and TRUE) or FALSE => TRUE or FALSE => TRUE.

IF open

IF closing with ENDIF instruction is compulsory, unless IF selects the **IF open** field.
In this case, IF instruction only affects the following working, which cannot be:
- setup or profile working;
- a logical instruction itself (IF, ELSEIF, ELSE, ENDIF) or an Application point (in a subroutine).

Any incorrect use of the IF.. ELSEIF.. ELSE.. ENDIF instructions is reported during the application of logical conditions. Error situations are described in the chapter Error in logical conditions.

Exit instruction

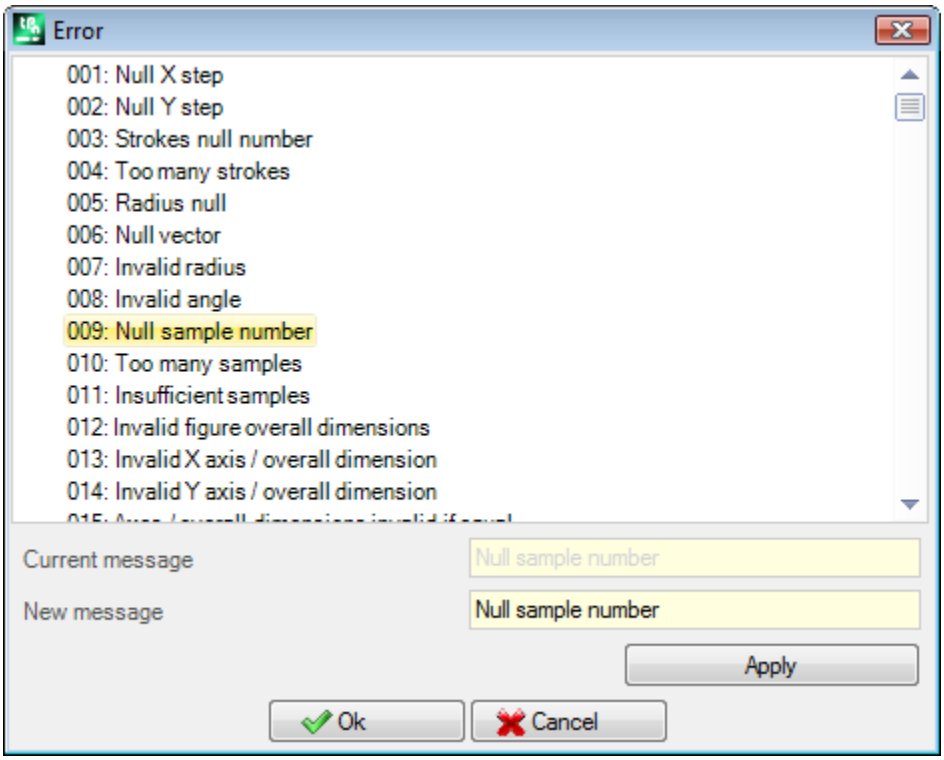
The EXIT instruction allows to force situations of logical conditions by jumping forward while executing the programmed text. Jump condition is expressed in the same formalism as IF instruction.
If the instruction condition is TRUE or is not set, the instruction interprets the jump condition. In this case:
- it determines the direct exit at the nearest nesting level after the IF cycle
- if the instruction is executed outside an IF cycle, the EXIT instruction implies a jump at the end of the face program.

Even though inside an IF cycle, it is anyway possible to force the jump at the end of the program by selecting the RETURN field.
The jump condition is evaluated during the application of the logical conditions only, just as the condition expressed for the IF instruction. If the condition of the instruction is TRUE, the FALSE condition is forced for all the programmed workings in the IF cycle after the EXIT instruction.
If the test result is FALSE, the program development is normally carried on. The test result is TRUE, if no logical condition is required.

Error instruction

The ERROR instruction programs error situations. The error condition is expressed with the same formalism as IF instruction. If the instruction condition is TRUE or is not set, the instruction interprets an error condition.
If the error is generated during the call to a subroutine, its development is not performed and an error is signalled.
If the error is directly generated in the main program text:
- TpaCAD warns of the error situation, when the logical conditions are applied. The stop icon in the ERROR instruction tells that it is verified as TRUE
- during the execution phase, it deletes the program interpretation and stops its execution.

The ERROR instruction can effectively control the validity of parameters and/or variables assigned when recalling a subroutine, or the validity of variables assigned during the program execution.
When the **Error** entry is selected in the working data entry, a list is shown where the errors assigned (number + message) are displayed.



At manufacturer level, a new message can be entered or an existing one can be modified. After selecting the message to be changed or entered, the change must be written in the edit field **New Message**. To confirm the message entered, press the button **[Apply]**.

Warning instruction

The WARNING instruction programs signaling situations; what it has been said about the ERROR instruction remains valid, only now no error is indicated, but only a notice that it does not influence the normal development or execution of programs and subroutines as such.

ERROR running: selecting the field, the user can differentiate the behaviour of the instruction: during the execution, the instruction activates an ERROR situation, deletes the program interpretation and stops his execution.

J Variables

It is sometimes required or only convenient to assign variables during the definition of the face program. For instance, it is required, when the program cannot be entirely defined or needs some information taken from the macro or subroutine application. In general, making local assignments while writing the face program can be more convenient instead of gathering them in the r variable table; this offers a better ease of program comprehension. For this purpose, <j> variables are available. It is about 100 variables of numerical type, identified by name: from j0 to j 99.
<J> variables are local to a face. This means that:

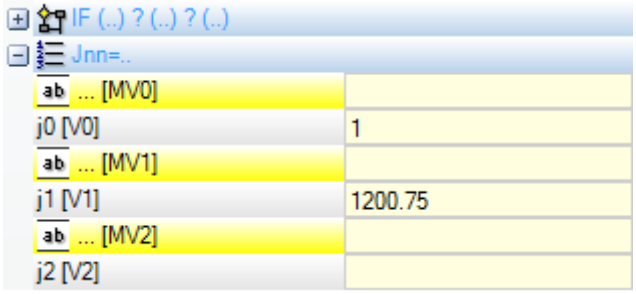
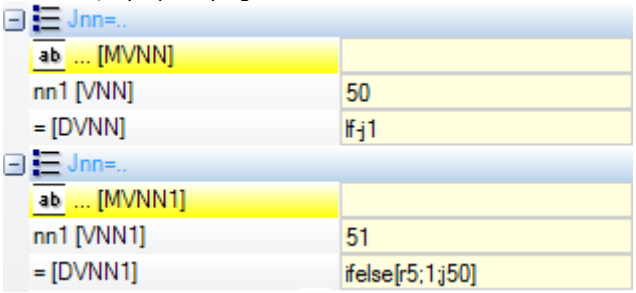
- there is no correlation of assignments and reading of variables among different faces.
- each face program starts with the set of pre-set variables to 0.0 (zero) value.

<J> variables can be used in each working applied to the face, at any level:

- a <j> variable can be used to fix hole diameter, or a working coordinate, or a logic condition
- inside the face, the visibility of variables is global at any level of application. So:
  - the main program can set j5=1
  - the application of a subroutine can modify the j5 value (for example: j5=2)
  - after applying the subroutine, the main program can retest the j5 value finding it changed.

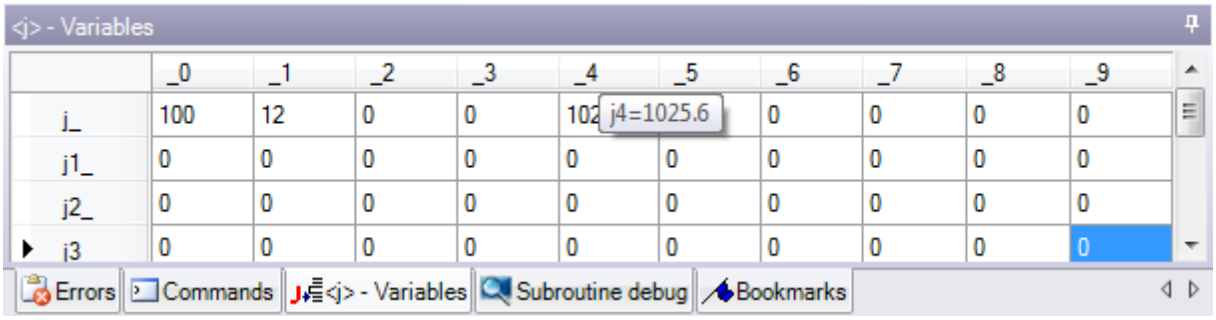
Applying induced subroutines calls, the <j> variables have the values set at the moment of the main call.

Three general instructions for the assignment of <j> variables are defined in the workings *palette*:

	ASSIGN Jnn	This instruction allows to assign one or more <j> variables. If necessary, condition the assignment/assignments with logic conditions (set in **IF (..) ? (..) ? (..)**): • the assignments are executed only if the logical condition set is verified as TRUE • the first Jnn=... node groups a certain number of direct assignments: in the figure the displayed:  IF (..) ? (..) ? (..) Jnn=.. ... [MV0] j0 [V0] 1 ... [MV1] j1 [V1] 1200.75 ... [MV2] j2 [V2] are 2, from j0 to j2. In the example, the first two variables (j0=1; j1=1200.75) are assigned • the following nodes enable to assign the same number of variables, by specifying the variable index.  Jnn=.. ... [MVNN] nn1 [VNN] 50 = [DVNN] lf-j1 Jnn=.. ... [MVNN1] nn1 [VNN1] 51 = [DVNN1] ifelse[r5;1;j50] In the example, two variables are assigned: J50 with "lf-j1" value and J51 with "ifelse [r5;1;j50]" value.
	ASSIGN Jnn with condition (.. ? .. : ..)	This instruction allows to assign one or more <j> variables based on the evaluation of logical statements (set in **IF (..) ? (..) ? (..)**). The programmed assignments are executed anyway, in case the logical condition set is verified: in this case, a part of the assignments is TRUE, a part of the condition is FALSE. Informationally, we speak about a ternary condition. For each variable, a node is assigned, with three available fields: • the first field sets the variable index (value from 0 to 99) • the second field shows the assignment to be made in case of logical conditions verified as TRUE • the third shows the assignment to be made in case of logical conditions verified as FALSE
	ASSIGN Jnn (0 - 99)	This instruction allows to assign all or a group of <j> variables, eventually by condition the assignments with logic conditions (set in **IF (..) ? (..) ? (..)**): • a first field set the initial group index to be assigned (example: 0) and a second field sets the final group index to be assigned (example: 50). If the fields are empty, they do not assign a group but the variables for integer from j0 to j99; • the third field shows the assignment to be made. The assignments are executed only if the logical condition set is verified as TRUE.

Among the instruction parameters, some descriptive texts associated to each single <j> variable are available. The rows are headed as [MV0].. [MV1]..

What is the value of J variables



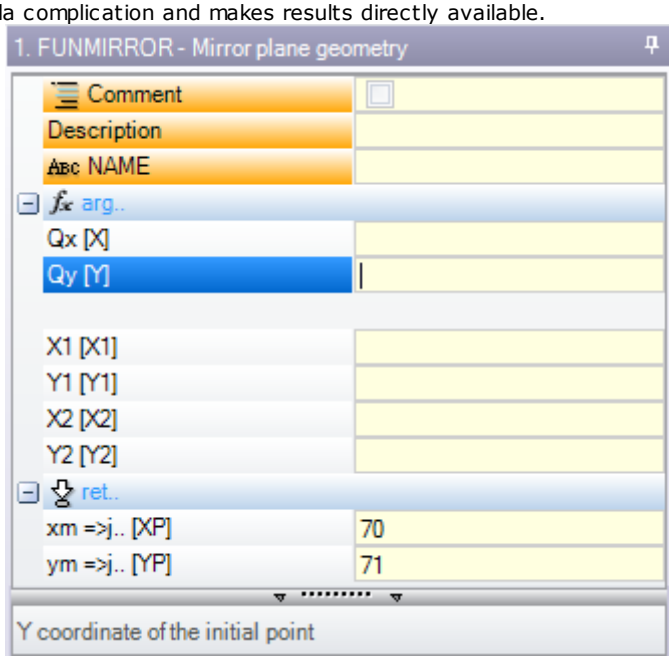
The <j> variables are displayed in the area of Commands in lower left side of the screen.
The window arranges the 100 <j> variables in a table of 10 rows and the same number of columns:
row j_: shows the variables from j0 to j9;
row j1_: shows the variables from j10 to j19;
...
row j9_: shows variables from j90 to j99.

Hovering the mouse cursor over a cell, a help message (tooltip), showing the corresponding variable name and its assigned value (example: "j4=1025.6"), is displayed.
In piece Overall View, all workings take value 0.0.
In Face View: the values shown in the window can change if the current window changes. The window is updated to the state of the variables according to its availability after the current working.

Global functions

Global functions are special logical instructions which allow performing a more or less complex calculation procedure and directly assigning results in <j> variables.
They must be set up in the configuration phase of an application, by assessing specific customization needs in details.

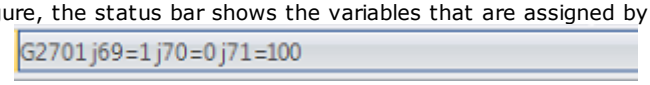
A simple example follows:
The position of a point P is to be determined, with (r0;r1) coordinates and mirrored around a generic axis assigned by two points: P1 (r2;h/2), P2 (l/2;r3). A possible way is that of obtaining formulas for the transform required and assigning the first variable r for x coordinate and the second variable r for y coordinate. If the transformation concerns a single case, this solution can be surely suitable.
Let us suppose that we need to calculate the transformation more times and in different programs: each time the formulas must be remembered and written again.
When using global functions, all formulas can be written once and recalled by using a proper instruction, which does not show the formula complication and makes results directly available.



arg.. groups the arguments required by the instruction:
- coordinates of point to be mirrored (x;y);
- coordinates of two points on axes (P1(x1;y1) and P2(x2;y2)).

ret.. groups the return fields:
- **funmirror =>j..**: it sets the <j> variable index returning the function result (here: j69): for example 1, in case of correct result, 0 in case of incorrect result
- **xm =>j..**: it sets the index of the <j> variable returning the x transformed coordinate (here:j70);
- **ym =>j..**: it sets the index of the <j> variable returning the y transformed coordinate (here:j71).

In case of setting as in figure, the status bar shows the variables that are assigned by the instruction:



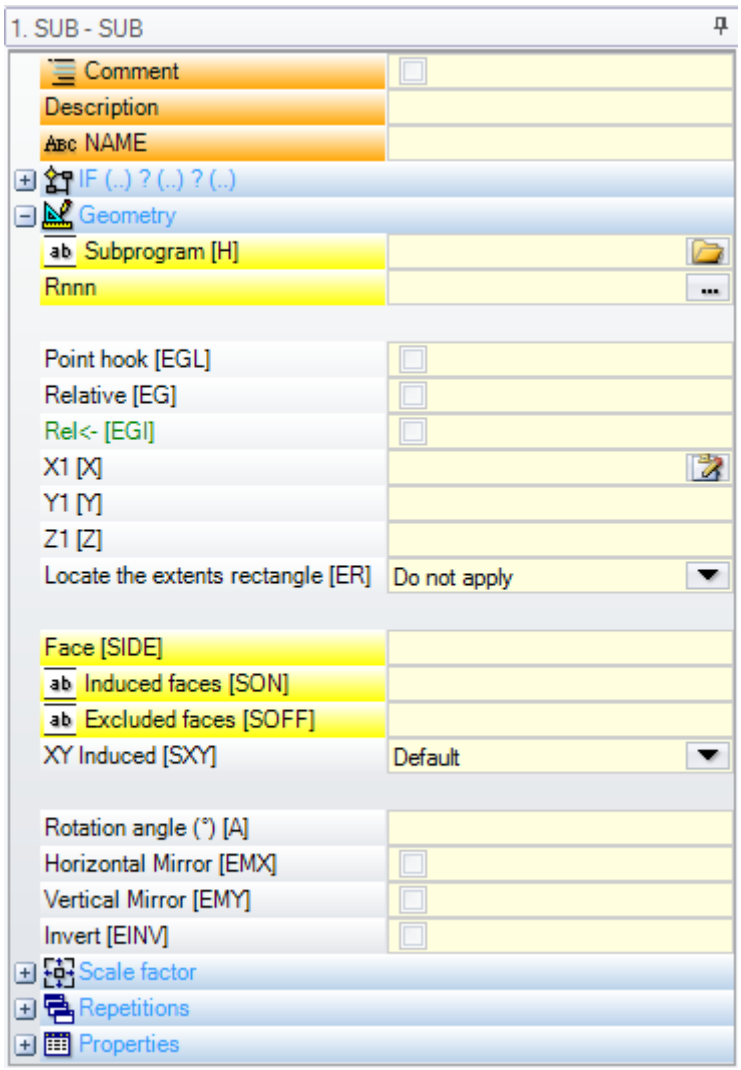
J69=1: correct result of the function (our example does not include cases of invalid solution)
J70=0: mirrored X coordinate
J71=100: mirrored Y coordinate

9.4 Subroutine

Subroutine

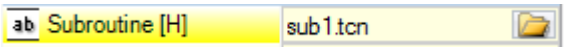
The subroutine is a piece-program file implemented with program or subroutine typology.
In the **Workings** tab of the SUBROUTINES groups, 3 types of codes for the application of a subroutine are defined, as follows:

SUB	manages the geometric transforms and the multiple applications with free repetition
SMAT	manages the geometrical transforms and the multiple applications with matrix repetition
SEMPTY	manages the geometrical transforms except for the scale factor. It does not manage multiple applications. It can generate emptying.



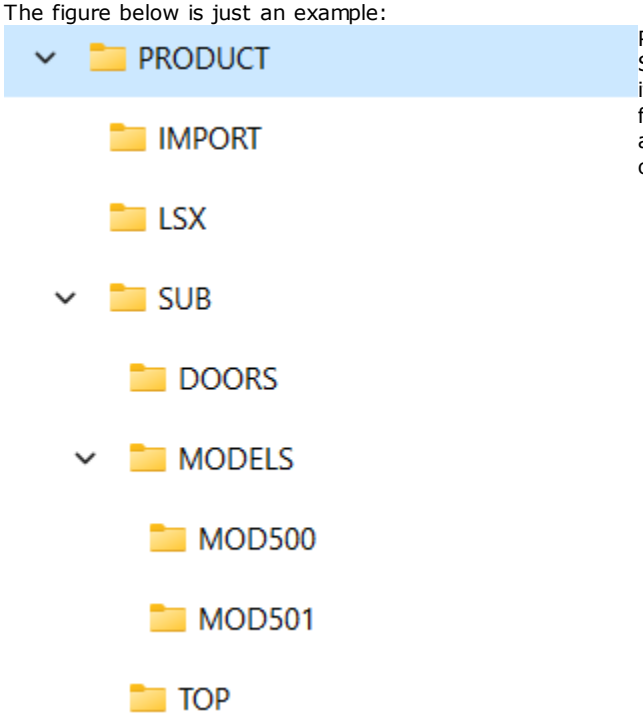
Let us see a few significant examples for the application of a subroutine:
- **"IF (..) ? (..) ? (..)"** node: possibility to condition the application of the subroutine called directly. The subroutine is applied only if the condition is TRUE.

- **Subroutine**: it can be edited also in parametric form or it can be assigned by opening the file open window.
  The icon ab on the left side of the field shows that it is about a *string* parameter. In the *Open Piece* window the research is set in the standard storage folder of the subroutines (SUB). The available file types that correspond to the only program format files are listed in the file open window. If a file of format valid for piece-program is selected, the relevant dimensions, comment and graphic preview are displayed in the window. Closing the window, the name of the selected subroutine is shown in the SUB field. Example:



In this example, the whole localization path of the subroutine is not shown, but only name.extension. An **addressing to the standard storage folder of the subroutines (SUB)** is indeed recognized. This ensures the portability of the programs. If we indeed copy our program to another machine, it is enough to copy also the sub1.tcn subroutine to the SUB directory to make everything work properly. In case of relative addressing, if the program has the macro extension (*.TMCR), it is searched in the macro directory and not in the subroutine (SUB) directory. It is possible to know what type of program is this by reading it and in the case of macro, it is not opened. The subroutine name and extension cannot include the following characters: **\ /:* ? " < > | # %**.

The SUB directory can contain other folders, where to store subroutines.
The figure below is just an example:



PRODUCT is the standard program archive directory; SUB is the standard subroutine archive directory and it is assigned in the program directory; in SUB the following folders are created: DOORS, MODELS, TOP and each of these folders can contain, in their turn, other sub-folders

- if the SUB1.TCN subroutine is selected in "..\PRODUCT\SUB\MODELS\MOD500\", the SUB field is assigned as follows: "MODELS\MOD500\SUB1.TCN"
- if the SUB1.TCN subroutine is selected in the "..\PRODUCT\" program directory, the SUB field is assigned as follows: "..\SUB1.TCN": also in this case a relative addressing is maintained to ensure program portability;
- if the SUB1.TCN subroutine were selected in a sub-folder of the "..\PRODUCT\DOORS\" program directory, the SUB field would be assigned as follows: "..\DOORS\SUB1.TCN": also in this case a relative addressing is maintained to ensure program portability;
- if the subroutine is selected outside the folder of the programs, the SUB field shows the whole subroutine localization path, without ensuring the program portability.
- Rnnn: it sets the subroutine "r" variables which can be reassigned. See chapter Assigning the subroutine variables.

- **Point hook**: possibility to carry on a profile
- **Relative, Relative <-**: absolute or relative mode with respect to the previous listed working
- **Positioning the extents rectangle**: subroutine application point is located according to the overall rectangle, following these options:
  - **Centring in XY**: in correspondence with the overall rectangle centre
  - **X-Y-**: in minimum overall point in both X and Y
  - **X-Y+**: in minimum overall point in X and maximum overall point in Y
  - **X+Y-**: in maximum overall point in X and minimum overall point in Y
  - **Y+**: in maximum overall point in both X and Y
- Available geometric transforms:

- **X1, Y1, Z1**: translation (the fields assign the application point)
- **Rotation angle**: rotation
- **Horizontal mirror, Vertical mirror**: mirrors
- **Inverting**: it reverses the execution of the subroutine

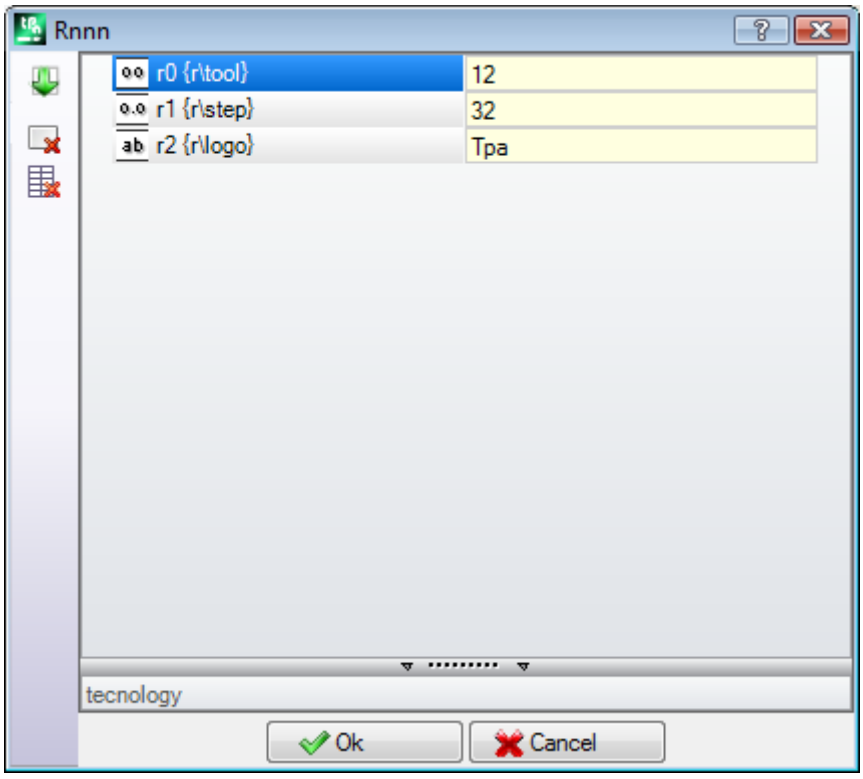
- **Induced faces**: it lists the faces to apply in case of induced calls
- **Excluded faces**: it lists the faces not to apply in case of induced calls
- **Induced XY**: it chooses among different adaptation modes for the application point (positioning) in secondary calls (induced XY)

- **Emptying**: request emptying development
- node **Scale factor**: it sets the scale factor to modify the dimension
- node **Repetitions**: it sets a multiple application of the subroutine with possibility to choose between a free or a matrix repetition.

- **Property**:
  All properties can be assigned on a subroutine code. Let us say, more generally, that this is valid for all complex codes, unless different specification in the configuration of the workings.
  Let us see some particular aspects as follows:
  - "C" field (Comment): the whole working is a comment working and there is not any subroutine application;
  - "L" field (Layer): in case of strictly positive value (> 0), the entire SUB code development takes the value set (the value is propagated). In case of null value (0): the value is normally not propagated (it is about default setting, but it is possible to propagate the 0 value). If the subroutine performs a profile hook (carries on a profile beginning upwards), the value of the "L" field value is propagated from the profile setup;
  - "B" field (Construct): same considerations as for the "L" field;
  - "O", "M" fields: value propagation is decided at configuration level, both for the strictly positive values and for the 0 value. If the subroutine performs a profile hook (carries on a profile beginning upwards), the field value can be propagated from the setup or maintain a different position, as defined in the configuration.
  - "K", "K1", "K2" fields: value propagation is decided at configuration level, both for the strictly positive values and for the 0 value. If the subroutine performs a profile hook (carries on a profile beginning upwards), the field value is propagated from the profile setup.

Assigning the subroutine variables

The **Rnnn** entry sets the <r> variables of the subroutine which can be reassigned and edited only in the dedicated window. Among the set "r" variables, only those which can be reassigned in the subroutine are displayed. This option is not managed, if the SUB field is not assigned or, if the setting is not valid or if the subroutine has no variables which can be reassigned.
The displayed window is:



In the second column, the variable name (r0, r1) and the name in full are displayed, if the name in full is not assigned. In the third column the assignment of the variable is displayed. The icon of the first column of each row shows the variable type. In the picture, r0 is of integer type 00, r1 is of double type 0.0, r2 is of string type ab.

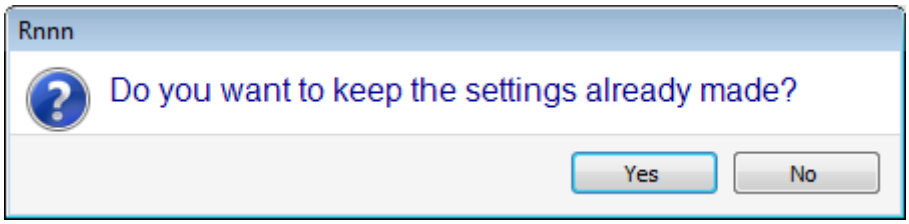
At the insertion of a subroutine the column fields are initialized to the values assigned to the variables in the subroutine. If the field is empty, one of the following two cases can occur:
- 0 value is imposed
- the value assigned to the variable in the subroutine text is imposed.

Its behaviour depends on how TpaCAD is configured by the machine manufacturer.

Assigning a variable, it is possible to require a help for parameter programming; the menu help opens, when you click the edit field of the variable with the right mouse button. In case of invalid setting, for example of syntax error, an error warning is shown directly when confirming the data and the image displayed aside remains as visible recall of the error status. Anyway, the closure of the window with confirmation is not conditioned by error warnings.

The buttons of the Toolbar on the left side of the chart allow to:

Importing the assignments of all the variables from the subroutine. If in the list there are fields already assigned, following window is shown:



Select **[Yes]** to set non-assigned fields only
Select **[No]** to overwrite all fields with the assignments read by the subroutine.

Reset the value of the selected variable

Reset the value of all the variables

To set the variables, the same considerations for each other working field are taken into account. To all intents and purposes, it is about the information related to the working that is being assigned, only with a higher degree of configuration. If the subroutine is edited or if the subroutine name changes, the variable window may change. In particular, it is possible to use each valid setting parameter. Let us consider some examples of assignments for numerical variables:
- r0=r5+32: it uses the r5 program variable;
- r1=100.5: it is assigned only numerically;
- r12=lf/2: it uses the length of the face, the subroutine is applied to.

Any set "r" variables of the subroutine, which cannot be reassigned, are recalculated based on the new settings. In the subroutine text, for example, let us assign two non-re-assignable variables:
- r100=lf-r0*2
- r101=r10

The r100 variable value is assigned with:
- lf: length of the face where the subroutine is applied
- r0=r5+32

The r101 variable value is assigned with the r10 value, as assigned in the subroutine. If the subroutine does not assign r10, the variable is searched among those of the program the subroutine is applied to, as has been shown in the previous paragraph.

Automatic assignment of Rnnn variables
Rnnn variables are automatically assigned when in a subroutine one or more r variables are used without any value assigned (empty field). When a subroutine is recalled by a program, the variables mentioned above are searched in the calling program and, in case of more cascade callings, the backward research of the value to be assigned can continue until to the calling program.
It is an useful mechanism for a fully automatic passage of one or more information into the subroutines, if an entire archive of programs always uses these information. However, unwanted results may appear, if the functionality is not properly used, for example because we forget to keep an r variable free. This mechanism would become fully obsolete using <o> and <v> variables, that in the program are always public. For this reason

we recommend a restricted use of the automatic assignment of the **Rnnn** variables, if really needed and, as a rule, the use of the only variables assigned in explicit way.

To better understand the mechanism of automatic assignment, let us analyse the following example:
in a subroutine, the variable r0 is used to assign the diameter of a drilling tool, but the variable remains not assigned. In this case, in the subroutine the r0 value is assigned as null and of numerical type (double).
Then, the subroutine is recalled into a new program:

- if the program does not assign the r0 variable, all remains unchanged: the diameter of the drilling tool has value 0.0;
- if the program assigns a value 10 to the r0 variable, the application of the subroutine changes: the diameter of the drilling tool now has value 10.0.

You may benefit from a special note, if you use a r variable with symbolic name. For our example, we do not use the variable as "r0", but as "r\fitool" and we do not assign the variable in the subroutine: now, in the subroutine edit a warning is reported (non-serious error) 103 - Parametric programming: "r" variable recalled by name not found.
Then, the subroutine is recalled into a new program:

- if the program does not assign any "r" variable named "fitool", all remains unchanged: the diameter of the drilling tool has value 0.0 and the warning appears again;
- if the program assigns the name "fitool" and a 10 value to a "r" variable, the application of the subroutine changes: now, the diameter of the drilling tool has value 10.0 and the warning disappears.

This mechanism described here to research and assign the variables can always be modified in the configuration of TpaCAD with the following steps:

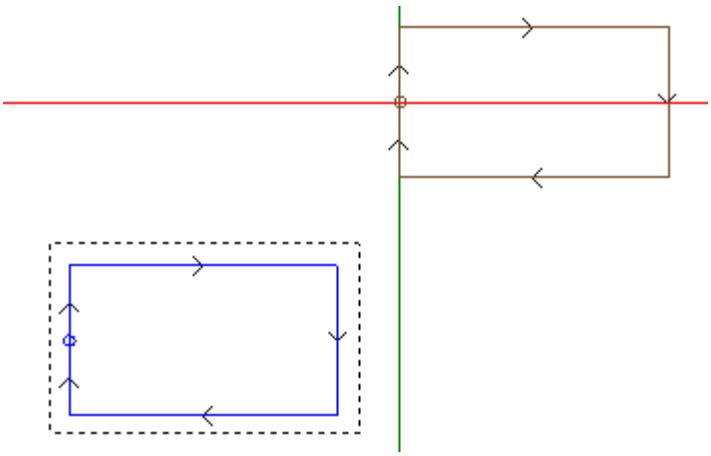
1. Whole Exclusion. In this case, the value of a non-assigned Rnnn variable is always 0.0. In case of use by name of a variable that is not assigned (in the example: "r\fitool"), the diagnosis report corresponds to the error 103 and now, it is not a Warning, but a real Error;
2. Activation in case only of use of a variable by name. Going on with our example, the mechanism of automatic assignment would be activated for "r\fitool", but not for "r0".

Other automatic assignments of the subroutine
If in a subroutine a dimension of the piece (l, h, s) or a 'o'/ 'v' variable, or a custom section setting, or a variable geometry are used, what these information refer to?
The answer is clear: to the main program that calls the subroutine, apart from the point where the subroutine is called (see paragraph: Nesting subroutine calls).

Positioning a subroutine

A subroutine is positioned in the YX face plane and depth with Z direction, perpendicular to the face plane: the values calculated of the three coordinates (x, y, z) define the **application point** (point that we call: P1).
The application point is programmed in a system of **Cartesian coordinates**, where it is possible to assign the coordinates in absolute or relative mode.

When **relative** mode is selected, the absolute mode can be forced on a single coordinate, by placing "a;" before the coordinate setting.

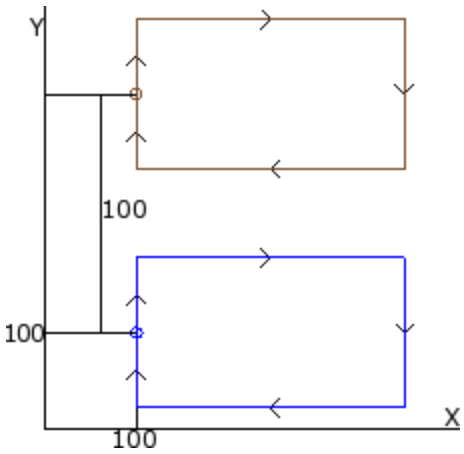


The rectangle selected in figure represents the subroutine development (a rectangle crossed clockwise, with start point located in the middle of the left vertical side).
The cross cursor indicates the application point P1:
subroutine start point is located in P1 (rectangle setup).

When the coordinates of the point P1 are not assigned (empty field), one of the two following cases occurs:

- the translation with respect to the original position of the subroutine is not applied. For example, if two coordinates only are set up for P1 in XY plane, rectangle positioning in Z remains unchanged.
- The coordinate of the previous working is propagated using the same criteria to place a point working (example; single drilling).

Its behaviour depends on how TpaCAD is configured by the machine manufacturer.
If relative positioning mode is active and working is preceded by another complex code (macro or SUB working) **Relative <-** field is also evaluated. If enabled too, application point P1 is considered as related to the application point (P1) of the previous working.

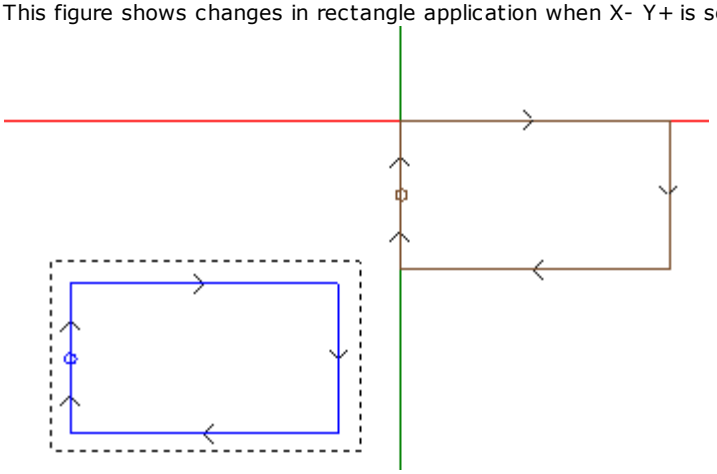


The figure corresponds to two applications of the subroutine in the example (making a rectangle):
- at the bottom, the application point is programmed as absolute at (X=100; Y=100);
- on the top, the application point is programmed as relative, with **Relative <-** and coordinates x = 0 and y = 100:
  - the relative coordinate X=0 sets the x coordinate of the point P1 to the same x coordinate of the point P1 in the first application (bottom rectangle);
  - the relative coordinate Y=100 sets the y coordinate of the point P1 by adding 100 to the y coordinate of the point P1 in the first application (bottom rectangle).

Is it possible to differently choose which subroutine point will be taken to P1, for example, by referring to the rectangle centre, instead of the setup working?

Yes, in different ways.
One method provides the selection of **Locate the extents rectangle** in subroutine application SUB code assignment window. This is a multiple selection field with the following entries:
- **Do not apply**: the field does not affect the positioning of the subroutine
- **Centre in XY**: subroutine overall rectangle centre is taken to P1
- **X- Y-**: overall point in both X and Y is taken to P1
- **X- Y+**: minimum overall point in x and maximum in y is taken to P1
- **X+ Y-**: maximum overall point in x and minimum in y is taken to P1
- **X+ Y+**: overall point in both X and Y is taken to P1

This figure shows changes in rectangle application when X- Y+ is selected:

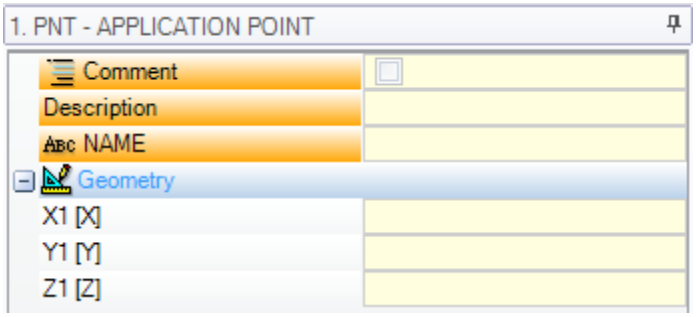


ATTENTION:

it is the simplicity of the example subroutine that helps the correspondence between the subroutine overall rectangle and the figure programmed.

Programmed application point

The coordinates of the point to be translated to the application point P1 can be programmed in the subroutine itself.
Reference is made to the **Application point** logical instruction for programming:

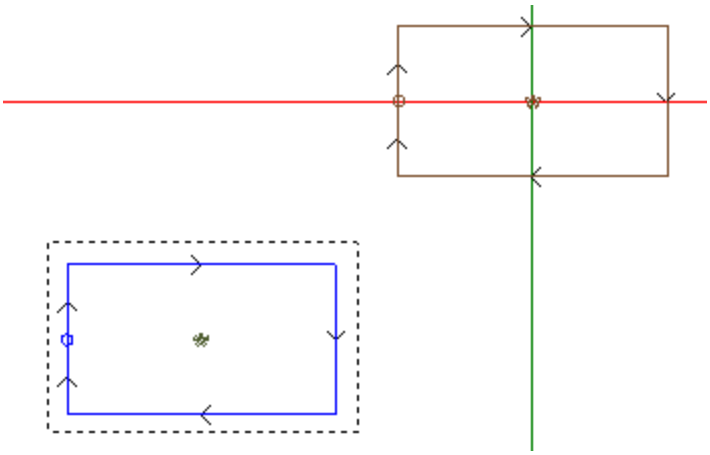


The three fields X1, Y1, Z1 assign the point to be located, when the subroutine itself is recalled.
The programming is interpreted in absolute coordinates and it is valid for all the three coordinates: for the field that are not set, the value is 0.0.
ATTENTION:
The code is interpreted in subroutine application **only**.

In the application of a subroutine, only one application point can be significant: the first one that is verified by the logical conditions.

The application point here assigned does not necessarily need to match a work position. In the example, rectangle centre coordinates can be reasonably set up, as programmed in the subroutine.

The figure shows how the subroutine application changes, by adding the instruction APPLICATION POINT for the rectangle centre.



The instruction APPLICATION POINT in the subroutine is ignored, if the subroutine application SUB code sets a valid selection at the option **Position the overall rectangle**.

Point hook

The selection of point hook option:
- has relative mode with zero shifts for three coordinates in application point (P1) (it makes different setups useless in point P1)
- it makes field selection useless: **Relative <-** and **Position the overall rectangle**
- ignores the application point instruction set in the subroutine.

The point hook **always** applies a relative programming of null shifts.
If a profile element, where a hook can be executed (setup, arc or line, another complex working, that ends its development by a profile element), is available before the SUB code (previous line of a program, not of a

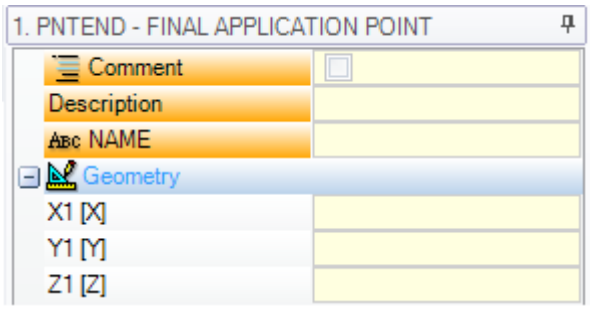
comment) and if the current subroutine starts with a profile element where a hook can be executed (setup, arc or line):
the subroutine application continues the profile started before, and actually excludes the setup execution starting the program itself.
In this case, the point hook has recognized a situation where a **Hook of profiles** occurred.

Final application point

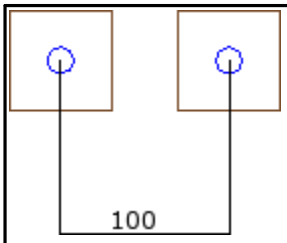
It depends on the working typology itself to determine the last working made in the development of the subroutine. When it is a matter of:
- a punctual working or a setup (for example, a single drilling), the last point worked is determined by its application point;
- a profile trait (line or arc), the last point worked is determined by the last point of the trait;
- a subroutine, the last point worked is determined by the development of the subroutine.

Let us define the following point: **final application point**.
While developing a subroutine, the final application point is relevant to apply:
- repetitions in execution of the subroutine itself;
- subsequent working with assignment of coordinates in a relative mode or in case of propagation of coordinates.

In a subroutine, it is possible to program the coordinates of the FINAL APPLICATION POINT by using the logical instruction appointed before. This instruction is interpreted only within a subroutine and in the case of use of more instructions within the same subroutine, the last one verified by the logical conditions is considered true.



The three fields of X1, Y1, Z1 set the final application point, that does not necessarily need to coincide with the coordinates of a working point. The programming of the coordinates occurs in absolute mode and for the coordinates, that were not set, the value 0.0 is taken up.

	In the example alongside, we can see a subroutine with a hole and a construct rectangle around the hole. The FINAL APPLICATION POINT is set, so that it coincides with the hole. In this way the call, for example, in relative mode of the same subroutine, that was programmed with displacement in X equal to 100, determines its displacement with respect to the hole. If also the APPLICATION POINT of the initial placement overlaps the hole, the position of the hole becomes, to all intents and purposes, the only significant position of the subroutine.

The use of the instruction FINAL APPLICATION POINT excludes the possibility of hooking the subroutine after the working and recognizing the profile continuation.
The instruction FINAL APPLICATION POINT is ignored, if in the code SUB of subroutine call a transform of **Inversion** is set.

Applying workings to the correct face

A subroutine is a piece-program file, whether it is implemented with program or subroutine typology. Therefore, subroutine workings are applied to one or more faces. Generally, we want to indicate which face of the subroutine we apply, setting the parameter **Face** of the SUB working. This setting can determine two different operations:
- activation of the Induced calls: setting has not been made (empty field);
- activation of the Direct call: setting has been made (NON-empty field).

Induced (automatic) calls

The application of the subroutine determines the automatic execution of each non-empty face of the subroutine. They have a face correspondent in the program applying the subroutine itself.
The term "induced" means that the application of the subroutine is propagated, in this case in automatic way, to other faces.
This kind of functioning corresponds to the more generic case of the application mechanism for induced calls. It is also called by the term ***automatic***, in contrast to the mechanism of *Programmed induced calls* (see further).
The application of automatic induced calls works steadily (read: it does not need to be specifically enabled). It is activated, if the *Face* field remains not set.

For example:
- let us create and save the subroutine ONE with the assigned workings:
  - holes in face 1
  - a groove in face 3
  - holes in face 4
- let us create now the program PRG1, select the face 1 and enter a SUB code that recalls the subroutine ONE, and leaving the face field unassigned. Then, we look how the graphic representation of the piece changes according to the entered working:
  - the workings assigned to face 1 of ONE are executed
  - also the workings assigned in ONE on the faces 3 and 4 are executed: we talk about the executions that correspond to *induced calls*. For the program line that determines the development of induced calls we talk about *master call* and for the belonging face we talk about *master face*.
  - let us save now the program PRG1
- let us modify the ONE subroutine by assigning some workings also to face 5
- let us open again the program PRG1: we immediately notice that the subroutine call enters also the workings in face 5
- let us modify again the ONE subroutine by clearing all workings of face 3
- let us now open the program PRG1 again. We notice that the workings in face 3 have been removed.

It is possible to see the structure of the code (SUB or similar) and of all the induced calls by opening the window for expanding the working from the **ASCII text** (see: Seeing the development of a subroutine).

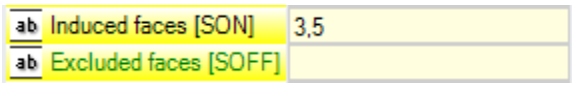
Some peculiarities of the application of an induced call:
- each call corresponds to an additional program line, one for each assigned call, only automatically managed and kept hidden (not visible)
- an eventual relative programming or point hook selection of the main call
- each call applies the status of the J variables that corresponds to the main call.

The automatic induced call mechanism is managed only at the basic programming level. To be clearer about this last concept, let us carry on with the example above:
- let us reopen the program PRG1 and insert a few holes in face 3
- we now create the program PRG2, enter the face 3 in programming and enter a SUB code that recalls the program PRG1, and leaving the face field unassigned. We now see how the graphic representation of the piece changes according to the entered working:
  - the workings assigned to face 3 of PRG1 are executed
    - also the workings assigned in the face 1 of PRG1 and resulting from the recall of the subroutine ONE are executed
    - on the other hand the workings in the other faces of PRG1 resulting first from the application of the subroutine ONE are not executed: recalling the subroutine ONE, it now does not determines any induced call, because it is not at the basic programming level.

If PRG1 had been created with subroutine typology, it itself would have stopped the induced call procedure, avoiding the source of situations of incomprehension.

Selecting induced faces

Induced call application can be selective:

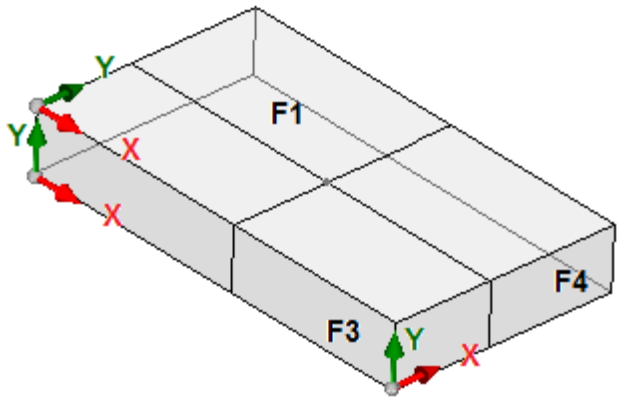


Induced faces: if set, it indicates faces involved in induced call. In figure: "3;5" setting indicates the application of induced calls in faces 3 and 5 only.
Excluded faces: if set, it indicates faces not involved in induced call. In figure: "3;5" setting indicates the application of induced calls in all faces excluding the faces 3 and 5. The field of the excluded Faces is interpreted only if the field of the induced Faces is not set.
In both fields, list the numbers of the faces split up by ";" (semicolon).

Positioning induced calls

In an induced call, the application point can be assigned in different ways, by selecting among various entries of the **Induced XY** field:

- **Default**: the field does not affect the positioning. Apply the assigned mode in TpaCAD Configuration: it is one of the following selections.
- **Adapt XY**: it adapts the application point
- **Go through XY=**: for each induced call, it goes through the fields as set in master call
- **Do not go through XY**: for each induced call, it goes through the fields as non-set.



It is necessary to adjust the application point for induced calls as X and/or Y axes could not correspond between different faces.
Let us consider this figure (three visible faces of piece are shown)

- let us assign a subroutine application in face 1: the application point on the face plane 1 is shown
- calls are induced in the other two indicated faces: face 3 and 4

Now, X and Y axes are examined in the two induced faces:

- face 3: X axis physically corresponds to X axis in face 1, while Y axis has no physical correspondence with Y axis in face 1;
- face 4: X axis physically corresponds to Y axis in face 1, while Y axis has no physical correspondence with X axis in face 1.

Thus, automatic associations could logically appear:

- in face 3: application coordinate X = application coordinate X for face 1; application coordinate Y not assigned;
- in face 4: application coordinate X = application coordinate Y for face 1; application coordinate Y not assigned.

The following table shows the correspondences applied with the **Induced XY** parameter set in the selection **Adapt XY**:

Master face	Induced face	Coordinate in induced face
(1,2)	(4,6)	X = Y coordinate from master face (if not set ="") Y = ""
(1,2)	(3,5)	X= X coordinate from master face Y = ""
(3,5)	(1,2)	X= X coordinate from master face Y = ""
(4,6)	(1,2)	X = Y coordinate from master face (if not set ="") Y=""
(any other case)	(any other case)	X= X coordinate from master face; Y = Y coordinate from master face

More specifically, an induced call in a fictive face always applies the same X and Y settings of the master call.

For the coordinates of the point P1 not assigned on the induced call (empty field), no translation is applied with respect to the original position of the subroutine. The propagation of the coordinates of the previous workings is never applied to the induced calls, as in the master call.

<j> variable solution in automatic induced calls

The development of an automatic induced call, as well, can use <j> variables.
For example:

- we write the subroutine ONE with workings assigned on faces 1 and 3:
  - we program on face 1:
    - line 1: code <HOLE> at x coordinate = 100.0
  - we program on face 3:
    - line 1: code <HOLE>, we set X coordinate = 100
    - line 2: IF (j0 > 0)
    - line 3: code <HOLE>, we set X coordinate = 50 in relative mode
    - line 4: ENDIF

- we write the program PRG1, with working assigned on face 1:
  - line 1: code <SUB>, we apply the subroutine ONE: the subroutine assigns a <HOLE> on face 1 (X coordinate = 100) and develops a programmed induced call on face 3
  - the induced call on face 3 assigns only one <HOLE> at X coordinate = 100

- we change the program PRG1 by inserting new instructions (on face 1):
  - line 1: <ASSIGN Jnn>, we set value j0=1
  - line 2: code <SUB>, we apply subroutine ONE: the subroutine assigns a <HOLE> on face 1 (X coordinate = 100) and develops a programmed induced call on face 3
  - the induced call on face 3 assigns now two holes:
    - <HOLE> at X coordinate = 100
    - <HOLE> at X coordinate = 150
  - line 3: <ASSIGN Jnn>, we set value j0=0
  - line 4: code <SUB>, we apply the subroutine ONE: the subroutine assigns a <HOLE> on face 1 (X coordinate = 100) and develops a programmed induced call on face 3
  - the induced call on face 3 assigns only one <HOLE> at X coordinate = 100.

The automatic induced calls use J variables as assigned at the moment of the main call (in the example: on face 1):
- any change to programmed J variables in the *main* call do not affect their *induced* calls, but the next program lines
- any change to programmed J variables in induced calls affect the single induced call.

Direct calls

The application of the subroutine determines the execution of the only program of the face clearly shown in the *Face* parameter.
Always with reference to the previous example, any face of the ONE subroutine can be applied in face 1 of the PRG1 program, just by typing the corresponding number in the Face field (also in parametric form).

Programmed induced calls

We illustrate a behaviour that is solved as an alternative to the automatic induced calls, with an evaluation priority that will be considered later.
The term "induced" shows that the application of the subroutine is propagated to other faces, not automatically, but in a programmed way by means of a specific working.
The application of programmed induced calls requires a specific activation, which differentiates in functionality and face:
- the functionality may be fully non-operating: in this case, the behaviour is caused by the two cases mentioned before. In case of *Essential* functionality: the functionality is never active;
- otherwise, it is possible in any case to have an activation limited to the application of a subroutine (or macro) on piece-face, or a global activation on every face.

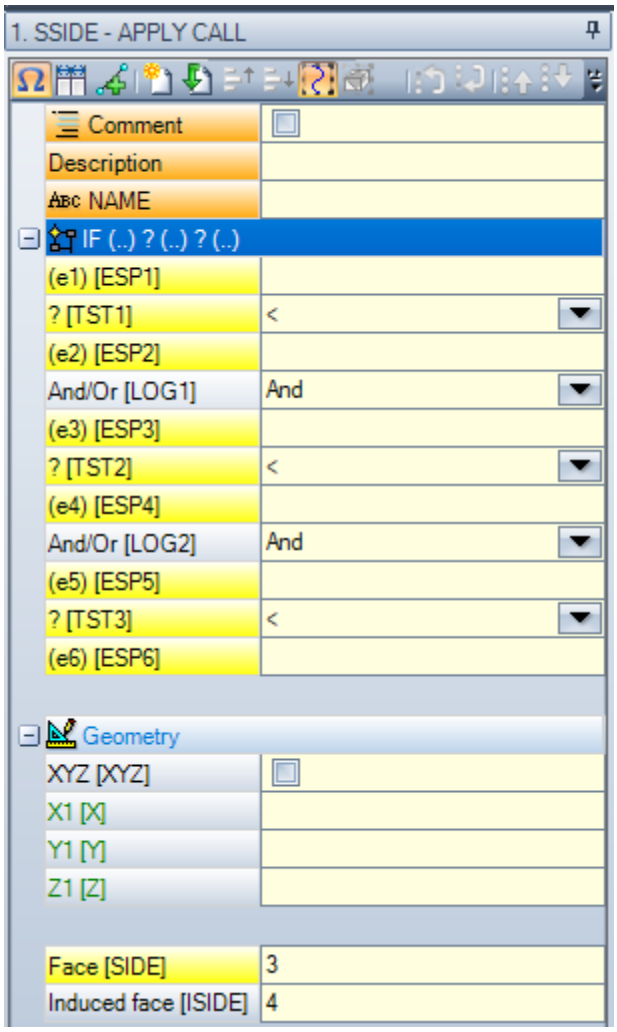
The behaviour of the piece-face may differ or fully match the behaviour of any other face.

The functionality is recognized if the application of the subroutine solves one code or both:
- SSIDE: application code of induced call;
- NSIDE: creation code of automatic face (with application on piece-face).
  and directly excludes the application of automatic induced calls.

To examine both codes, see the following paragraphs.
It is highlighted how the solution of a SSIDE or NSIDE code makes the subroutine call not expandable.

The application of programmed induced calls does not depend on the *Face* field setting:
- if it is set (example: 1), it assigns the subroutine face that must be applied to the current face
- if it is not set: it applies the same face of the subroutine to the current face.

The programming is implemented by the SSIDE working, which can be selected in the SUBROUTINES group of the **Workings** tab. This working can only be entered in a subroutine (or macro) text and becomes operating when the subroutine is called (for example, in piece-face):



- **"IF (...) ? (...) ? (...)"** node: the conditions of application are assigned with a direct IF condition, up to three logical condition terms between two expressions. If the condition is TRUE, the instruction can interpret an induced application of the subroutine call
- **Face**: it sets the subroutine face to be applied in the induced call. If the parameter is not set (read: if the field is empty): the working does not define any added application, but disables a possible solution of *Automatic induced calls*
- **Induced face**: it sets the application face of the induced call

In the example in the figure, the SSIDE instruction programs an induced call on face 4 of the calling program, with application of face 3 of the subroutine.

In evaluating whether a SSIDE instruction can be really applied, other assignments are considered:
- **Induced faces/Excluded faces**: selective settings concerning the faces to be included or excluded. For example, if the field **Excluded faces** specifies face 4, the SSIDE instruction here examined does not apply the programmed call
- excluded applications are those induced in:
  - unmanaged faces or construct faces (if fictive or automatic);
  - the current face (main application face)
  - an already induced face.

- **"Geometry"** node: can assign the specific application point for the induced call programmed here. Check the "XYZ" box to enable the recognition of the application point and to set the coordinates in the following fields (in absolute mode, and the programming is thought to be valid for all 3 coordinates).
  - a non-set coordinate (i.e.: empty field) applies value 0.0
  - if the field is not selected, the solution of the application point of the induced call is defined by the main call (see: Positioning induced calls).

Differently from the automatic induced calls, the mechanism of the programmed induced calls is managed the same way at any programming level.

<j> variable solution in programmed induced calls

Also the development of a programmed induced call can use <j> variables, with added performance compared to the case of automatic induced call.

For example:

- we write subroutine ONE with workings assigned on faces 1 and 3:
  - we program on face 1:
    - line 1: code <ASSIGN Jnn>, we assign value 100 to variable j0
    - line 2: code <SSIDE-APPLY CALL>, we set value 3 in **Face** field, value 4 in **Induced Face** field
  - we program on face 3:
    - line 1: code <HOLE>, we set X coordinate = j0. The hole will be at x coordinate = 0.0.
    - line 2: code <ASSIGN Jnn>, we set value j0 = j0+100;
    - line 3: code <HOLE>, we set X coordinate = j0. The hole will be at x coordinate = 100.0.

- we write program PRG1, with working assigned on face 1:
  - line 1: code <SUB>, we apply the subroutine: the subroutine on face 1 assigns the variable j0 and develops the programmed induced call in face 3;
  - on face 3: the first hole is now executed at X coordinate = 100, the second one at X coordinate = 200.

This example shows how a programmed induced call initially uses J variables as assigned at the moment of the induced call itself (SSIDE code). Further new assignments are added to the initial situation.

Applying geometric transforms

When a subroutine is applied, some geometric transforms can be activated, applied in the order below
If the subroutine applies a macro complex code, whose required transform is not allowed, the user is warned by an error message.

Inversion
The subroutine inversion requires the inversion of the execution order for the developed workings: the last block becomes the first one and so on.
In case of profile, the transform also determines the geometric inversion of the profiles themselves and of the settings of:
- tool compensation (right or left) of each setup.
- selection of entry/exit segments to profile (always on setups), in case of right or left arc setting.

If activated in TpaCAD Configuration, the application of this tool to a profile can apply the mirror technology.
If activated in TpaCAD configuration, the application of the tool to an oriented setup can apply the transform to the orientation axes (only if the current face is plane, that is, it is not curved or assigned as a surface).

Rotation
The subroutine rotation is set in numeric field, with rotation angle programmed (in degrees and decimal degrees) in XY face plane from X axis. The rotation occurs around the application point of the subroutine.
If activated in TpaCAD configuration, the application of the tool to an oriented setup can apply the transform to the orientation axes (only if the current face is plane, that is, it is not curved or assigned as a surface).

Mirror
A subroutine symmetry is set in two selection fields:
Horizontal Mirror: mirrored execution around a vertical axis
Vertical Mirror: mirrored execution around a horizontal axis.
If both entries are selected, options are summed up. The transform, only in case of an active selection, also inverts the settings of:
- tool compensation (right or left) of each setup, only in case of an active selection.
- selection of entry/exit segments to profile (always on setups), in case of right or left arc setting.

If a rotation is required as well, this is executed before the symmetry.
If activated in TpaCAD configuration, the application of the tool to a profile can apply the mirror technology.
If activated in TpaCAD configuration, the application of the tool to an oriented setup can apply the transform to the orientation axes (only if the current face is plane, that is, it is not curved or assigned as a surface).

Scale (stretch branch)
It applies a reduction or amplification factor to the subroutine and is enabled by the following entries:
- **Enable**: if selected, it enables the transform application;
- **Factor**: reduction or amplification factor (minimum programmable: 0,001). Following situations are interpreted:
  - less than 1: reduction applied
  - higher than 1: amplification applied
  - =1: no action.
- **3D scale**: if selected, it also enables in-depth application (face Z axis). Selection is compulsory if the subroutine also runs arcs assigned on a plane different from xy.

Repetitions in subroutine running

SUB codes manage two different modes of subroutine automatic repetition:
- SUB performs the multiple application with free repetition
- SMAT performs the multiple application with matrix repetition

Repetitions with free distribution

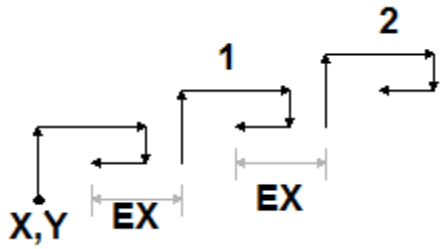
- **Repetitions**: number of repetitions to be added to the base application. The minimum value to enable the repetitions is 1.
- **X, Y, Z Offset**: deviations applied to each repetition. Values are applied as relative and added at each repetition.
- **Relative <-**: if selected, it applies the offsets to the application start point of the previous repetition. An Offset position can be forced to be absolute by entering "a;" before the same **"a;"** position.
- **Point hook**: if selected, it hooks each repetition to the previous one. In this case, it ignores the settings concerning the Offsets X, Y, Z and Relative <- field;
- **Offset A(°)**: it sets the rotation increase by applying it to each following repetition. The initial value is given by the value assigned to the rotation field in the base application. For example, if the base rotation performs a 30° rotation and the Offset A(°) is not set, all repetitions rotate by 30°; on the other hand, if the Offset A(°) =10°, then the first rotation rotates by 40°, the second one by 50° and so on for all the next rotations.

Any mirrored transform assigned for the base application is also applied to repetitions. In particular, transforms are also applied to the corresponding offsets:
- **Horizontal Mirror**: it also mirrors the offset set along the horizontal axis
- **Vertical Mirror**: it also mirrors the offset set along the vertical axis.

Any scale and/or inversion transform assigned for the base application is also applied to repetitions.

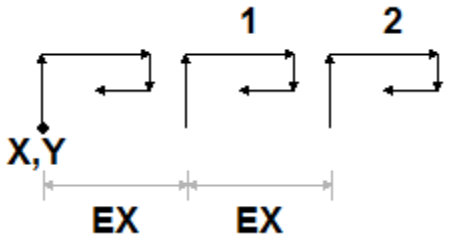
Let us see two examples, where the following common values are set:
the following values are to be set:
- Repetitions: 2
- X Offset: 100
- Y Offset: 0 (not set up)

- **Example 1:**



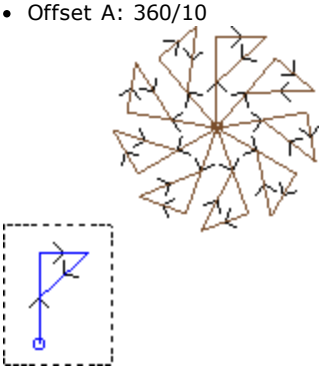
This figure shows the development resulting from the non-active **Relative <-** setting:
- **X,Y**: this is the base application point (can be the point of the overall rectangle, or the application point specified in the subroutine, or the first programmed point).
- **1**: corresponds to the first repetition. Its application point adds 100 in X to the base application of the final position and 0 in Y;
- **2**: corresponds to the second repetition. Its application point adds 100 in X to the base application of the final position and 0 in Y.

Example 2:



This figure shows the development resulting from the active setting of **Relative <-**:
- **X,Y**: this is the base application point
- **1**: corresponds to the first repetition. Its application point adds 100 in X to point P1 and 0 in Y;
- **2**: corresponds to the second repetition. Its application point adds 100 in X to first repetition application point position and 0 in Y.

Example 3:
The example uses a toy windmill, with repeated application of a single element.
the following values are to be set:
- Repetitions: 9
- Relative <-: enabled
- Offset A: 360/10



The section highlighted in figure corresponds to the single element as programmed in the subroutine. All repetitions are applied to the base application point and rotated to complete the round angle.

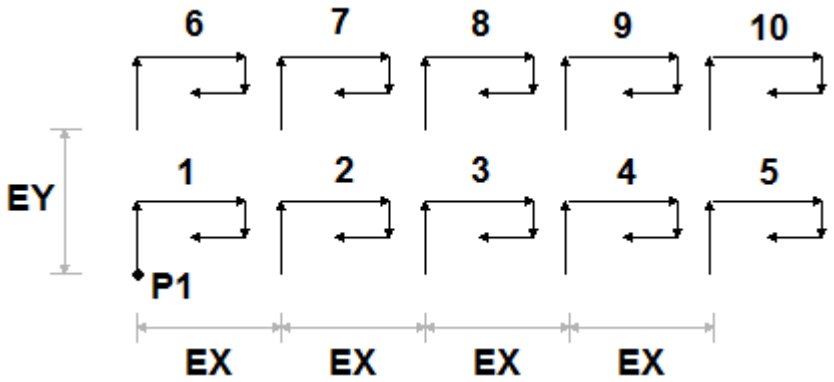
Repetitions with matrix distribution

- **Rows, Columns**: number of rows and columns of the repetition matrix. The minimum value to be enabled for the repetitions is 1, in both fields. The total number of applications made is given by the product (Rows * Columns), base application **included**. Development on the rows is always associated with the Y axis of the face and that on the columns with the X axis of the face.
- **Distance between columns**: distance between matrix columns
- **Distance between rows**: distance between the rows of the matrix
- **Relative <-**: if selected, it applies row and column offsets to the application start point of the previous repetition. An Offset position can be forced to be absolute by entering "a;" before the same **"a;"** position.

Any mirrored transform assigned for the base application is also applied to repetitions. In particular, transforms are also applied to the corresponding offsets:
- **Horizontal Mirror**: it also mirrors the offset set along the horizontal axis
- **Vertical Mirror**: it also mirrors the offset set along the vertical axis.

Any scale and/or inversion transform assigned for the base application is also applied to repetitions.

Let us now see an example, where the following values are set:
- Rows: 2
- Columns: 5
- Distance between rows: 100
- Distance between columns: 100



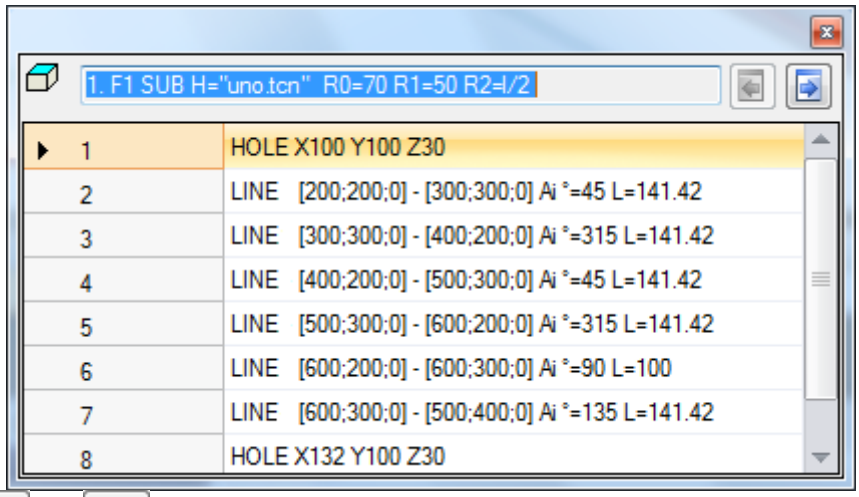
This figure shows the development resulting from the active **Relative <-** setting:
- **P1**: this is the base application point;
- **EX**: this is the distance between columns;
- **EY**: this is the distance between rows.

The total number of the executed applications is (2*5)=10, base application **included**.

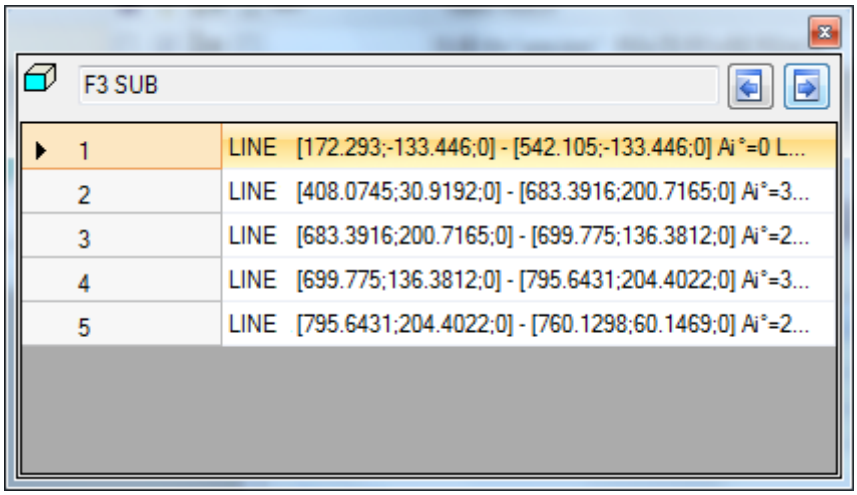
Seeing the development of a subroutine

It is possible to see the development of a subroutine by clicking on the corresponding cell of the **ASCII Text** with right mouse button. The working list is displayed with the geometric, technological information and the assigned properties. The formalism is analog to that used in the status bar.

The figure is related to the application case also in induced calls. In case of normal subroutine call (without induced call) the working list only appears in the window below:



The buttons  and  allow the user to switch to the previous or to the next call. The figure below is an example of induced call:

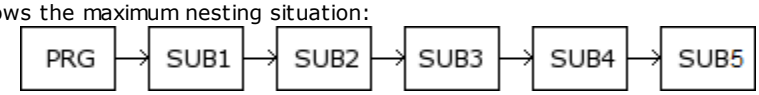


Nesting subroutine calls

It is possible to nest complex codes, assigned as macros or subroutine calls, but with a nesting limit up to 5 calls.

Let us suppose to be in the editor phase of a program (PRG: basic programming level):
- in a face of the piece, we can apply a subroutine call (SUB1)
- SUB1 can make calls to other subroutines. For example to subroutine SUB2
- SUB2 can make calls to other subroutines. For example to subroutine SUB3
- SUB3 can make calls to other subroutines. For example to subroutine SUB4
- SUB4 can make calls to other subroutines. For example to subroutine SUB5
- SUB5 cannot make calls to other subroutines

The figure below shows the maximum nesting situation:

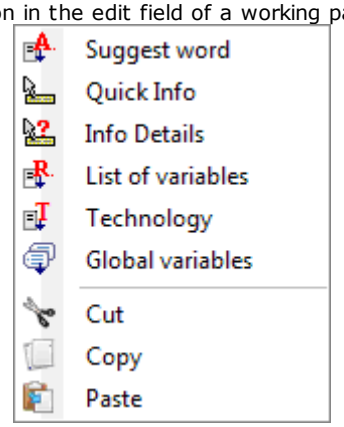


(SUB1, SUB2, SUB3, SUB4, SUB5) must not be necessarily subroutines, they can also be macros.
ATTENTION: also the Programmable tools are complex codes and as such they are considered in the evaluation of the number of nested calls.
The maximum allowed number of nesting automatically decreases by one in case we are in the editor of a subroutine or a macro-program.

Edit wizard and assisted functionalities

While inserting or modifying a working you may generally use the editor wizard functionality. Below, we show a list of the situations that may occur.

We have already written about the possibility to recall a quick help menu of functions, variable arguments and variables available for the parametric programming. Also in this case, it is a context-sensitive menu that can be opened by clicking the right mouse button in the edit field of a working parameter.



The composition of the menu changes according to various kind of evaluation:

- the parameter type to be modified (for example, if it is a string parameter, the **Technology** entry does not appear in the list)
- the available settings (for example: **Cut**, **Copy**, **Paste** commands verify the possibility to be used)
- the environment authorizations (for example: **Technology** or **Global variables** entries appear only if they are managed)
- the environment settings and access level (for example: access to Manufacturer level may display additional entries in the list or increasingly populate the menu associated to some entries).

 : this icon highlights the parameters for which you can activate an interactive acquisition in the graphic area. The acquisition mode is similar to what has been described in **Insertion of Geometric Entities from Drawing menu**. As already mentioned in the paragraph describing the working types, the acquisition can be automatically associated to multiple parameters or can be valid for one field only.

 : this icon highlights the parameters of technological settings, tool and/or electrospindle. By clicking the icon, a technology window opens, where it is possible to choose, in an interaction way, the technology for the working. In some particular cases, it is possible to choose two different technologies in different parts.

 : this icon is used to select the name/names of the workings, for example when using **Programmed tools** (see: next paragraph) or more generally, in all the workings that allow to assign one or more parameters to select the workings indicated by Name.

 : this icon is used to open the dialog box of some properties, for example Layer and Construct. In piece-face, the icon is also used for the Application face.

 : this icon, or other icons, opens the windows associated to dedicated settings such as emptying processes, selection in Font list, selection in the Setup or Point working list.

  : the combination of two icons is used, if the parameter can either select workings indicated by Name or select a working code (for example, a Setup code).

9.5 Programmed tools

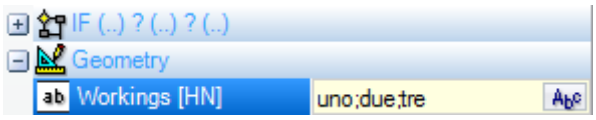


To understand what is meant by workings called ***Programmed tools*** and how they are used, let us examine the following example.
Let us suppose to have to execute the emptying of a closed area. It is possible:

- to apply Emptying tool directly to the profile. In this way, the emptying profile generated will not accept any changes of the original profile. In particular, the emptying process does not take into account any parametric programming of the original profile
- to save the profile in a subroutine and apply the transform by invoking an emptying complex code. This possibility allows the user to have the parametric profile, which can be modified, and consequently adapt the emptying procedure; however, the use of a subroutines is required.

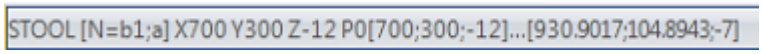
Workings that apply geometric transforms combine the positive features of both the above procedures: in our example, they can apply the Emptying tool directly to a profile, without switching to a subroutine, but allowing the change of the original profile and, automatically, the emptying process, as well.
In the TOOL group of the **Workings** tab we find:
STOOL: applies general transforms (translation, mirrors, rotation, scale, repetitions)
STOOL: EMPTY: empties closed areas
STOOL: SPLINE: generates Spline curves from polylines
STOOL: RADIUS: generates profiles for the compensation
STOOL: ATTAC: generates profiles with application of bridge points
STOOL: ZSTEP: generates profiles with application of progressions in Z
STOOL: MULTI: generates profiles with repetition along the development axis
STOOL: LINE: generates profiles for fragmentation and linearization
STOOL: ZLINE: generates profiles with linearization of the development along Z
STOOL: LINK: generates profiles connecting separated profiles
STOOL: STPLANE: rotates on Cartesian plane.

Which workings do these complex workings apply to?
The complex workings can work on all the workings inserted before and whose **Name (or N Field)** is set in the entry **Workings**. Several workings can have the same Name.



In the example in the figure above, the complex emptying working is applied to all the workings, whose **Name** is "uno", "due" or "tre". If the current working (for example: STOOL) is inserted into the line 10 of the program, the workings are searched in the first 9 lines of the program.
The order in which the workings are examined does not match the order in which they are listed in the **Name** field, but the progressive program number. In the example, the workings whose name is "due" will be applied first, only if they appear in the list **before** the workings whose name is "uno" or "tre".
If the name is assigned on a profile element, this one is considered until the end of the profile by the element that assigns the name. This makes assigning the name on the first element of the profile appropriate (typically: the setup), unless you want to explicitly consider only a part of the same profile, which would be assigned in this case without a setup.
The syntax of the **Workings** field is "name1;name2;...". The names must be separated by the character ';' (semicolon) and the number of the character should not exceed 100.

Also in this case, a parametric programming of the field with variables and string parameters is accepted. The solution of the parametric syntax must produce a "name1;name2;..." string. In the status bar, a string resulting from the solution of the parametric string and from an additional analysis is shown removing possible invalid assignments. In the example, the working is applied to the workings called "b1", "a":



In the **Workings** field, it is also possible to assign search operations by means of the wild character '*':
- if the string is ="*", or defines a name field equal to "*" (example: "aa;*;bb"), the transform is applied to all the previous workings with assigned name
- if a name field ends in '*' character, the search feature is applied to the names that begin with the part of the same set, and end with one or more characters anyway assigned. Example "aa;sp*": searching the "aa" names and all those beginning with "sp".

The additional string analysis removes possible names assigned with invalid characters (note: alphanumeric only) or with improper use of the character '*' (note: it can only terminate a name field). Field changes at this level do not determine a diagnostic message.
Some example can clarify the correspondence of what has been programmed to what has been shown in the status bar:

"aa;*;cc"	interprets: "*"	(the name "*" forces the total search)
"a*a;cc"	interprets: "cc"	(the character '*' does not terminate a name)
"abc*;cc??;d1"	interprets: "abc*;d1"	(the name "cc??" is not valid)
"aa,*,cc"	interprets: ""	(separator is wrong)
"AA;BB"	interprets: "aa;bb"	(leads back to capital letters)

If the Workings field is empty, no working is affected by the transform. Selecting the button  , a window opens and shows the list of the names that can be chosen. If the working programming takes place in piece-face, the only workings assigned to the same face by the complex transform code are accepted.
However, comment working, cycle logical instructions different from (IF.. ELSEIF.. ELSE.. ENDIF) cycle, or EXIT, or J variables writing are excluded.
For all codes, except for the first one of the group (STOOL):
- the complex workings that cannot be expanded are filtered
- only profile workings or logical instructions (IF.. ELSEIF.. ELSE.. ENDIF; EXIT; J variables writing) are evaluated.

For each code, except for the first one of the group (STOOL), following workings are excluded:
- complex non-expandable workings (typical examples: sawings, insertions),
- custom point workings and logical workings both programmed directly and resulting from the expansion of complex workings (example: FITTING of drillings).

In addition, all complex workings are applied expanded.

On the other hand, the STOOL code applies more general criteria, because it is used generally and is not for specific transforms of profiles:
- complex workings are not expanded, but directly applied
- all custom point workings or logical workings affect the transform.

If the workings that match because of the indication of the Name are excluded due to the application of the above-mentioned criteria, a warning message appears in accordance to the error number: 225 – Programmed tool: workings excluded.

Particular attention is focused on the application of *Programmable tools* to logical instructions. While including an IF.. ENDIF branch, make sure that the name required is assigned to the whole structure. An IF situation without closure with ENDIF instruction, for example, determine an error message due to the wrong match of IF and ENDIF.

1	aa	HOLE X100 Y100 Z-10 TD10 TP1
2	aa	HOLE X132 Y100 Z-10 TD10 TP1
3	aa	HOLE X164 Y100 Z-10 TD10 TP1
4	aa	IF ESP1=r0
5	aa	HOLE X196 Y100 Z-10 TD10 TP1
6	aa	HOLE X228 Y100 Z-10 TD10 TP1
7		ENDIF
8	aa	HOLE X260 Y196 Z-10 TD10 TP1
9		STOOL HN="aa" Xf/2

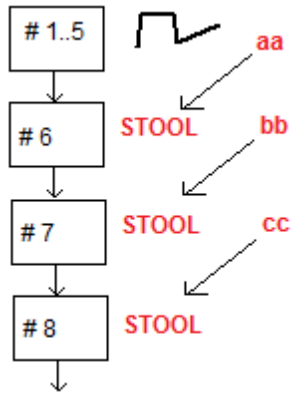
Let us consider the example of the figure:
- from the line 1 to the line 8, the name "aa" is assigned
- the line 7 has no name assigned. This is the ENDIF instruction that closes the IF of the line 4
- Also the line 8 is named "aa".

A STOOL is programmed to the line 9 and is applied to the workings, whose name is "aa". The compilation of the line determines an error situation: an ENDIF number less the IF's occurs.

The program lines that do not verify the logical conditions set are excluded from the transform.

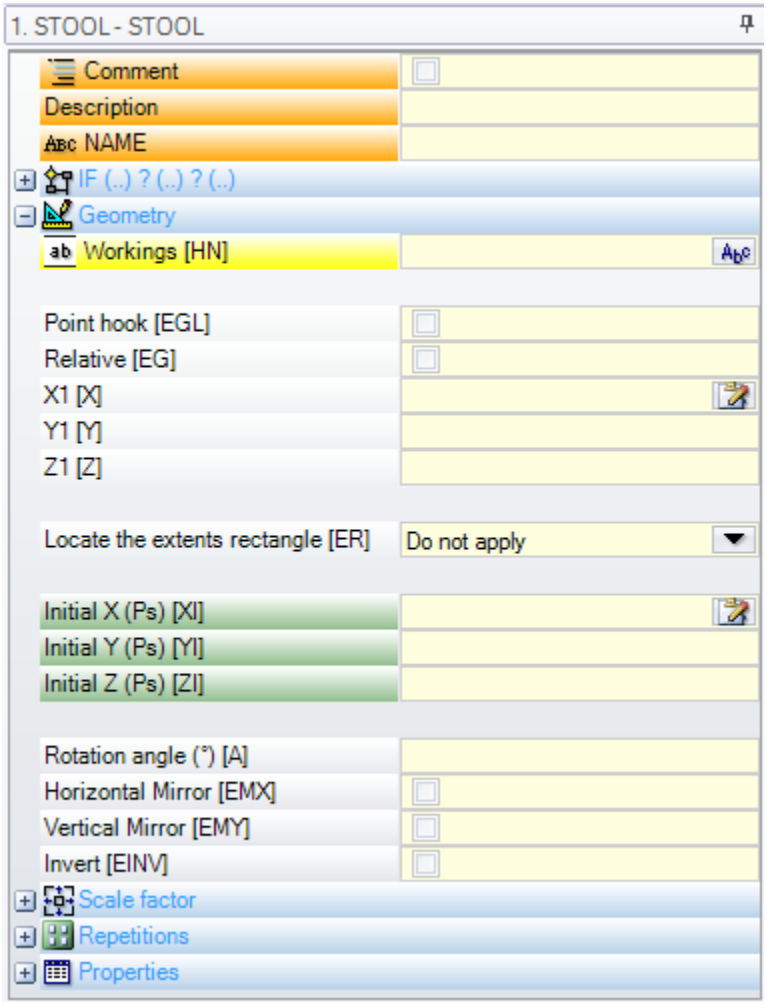
To the original workings affected by the application of the Programmed tool, a possible non-null value of the B Field (construct) is reset to zero. This allows to execute only the workings resulting from the transform code, but not the original ones, as generally requested.
The properties directly assigned to the STOOL working are then applied to the workings that are developed according to the normal criteria applied to all the subroutine codes.
Editing a macro-program, the application of Programmed tools does not determine any immediate development. The current development is only performed during the application of the macro-program itself.

A STOOL code is a complex code equal to a subroutine call or macro and, because of this, it's relevant to evaluate the number of nested calls, even with recursive calls at the same level.
Let us consider this picture:



#1..5 are program lines, assigned with **aa** name: we put a profile
#6 is the next line, a **STOOL** code: applied to the **aa** workings and assigned with **bb** name. This working applies a first nesting level
#7 is the next line, a **STOOL** code: applied to the **bb** working and assigned with **cc** name. This working applies a second nesting level
#8 is the next line, a **STOOL** code: applied to the **cc** workings. This working applies a third nesting level.

Let us now see in detail the STOOL code (apply generic transforms: translation, mirrors, rotation, scale, repetition). A consideration of the other codes is put off to the paragraph that examine the tools matching the codes itself.



The working is similar to the subroutine application SUB code. Now the Workings field replaces the subroutine selection field.
Next to the Application point coordinates (X1; Y1; Z1), there are 3 additional coordinates (initial X; initial Y; initial Z) to define an auxiliary point, functioning as:

- rotation centre;
- mirror axis.

The application node of the repetitions directly shows the selection from two possible schemes: free or matrix. For a full working exam, reference is made to the help, that can be directly called from the window of working data setting.

Advanced use of the programmed tools

The use shown until here for the **Programmable tools** can be seen in the list of the workings being programmed. Now, let us see a multi-step example, that will be described as follows:

- create a program (for example: PRG1);
- insert a working of simple drilling (example: HOLE) with assigned diameter from 10 mm to 150, construct and whose name is "aa";
- recall a subroutine (SUB1) and let transfer the "aa" name to an "r" public string variable:

SUB1 purpose:

- pick the "aa" working;
- examine it and understand if the machine tooling is able to drill or if the programmed diameter requires executing a milling cycle (read: circle emptying at the programmed depth).

According to our statements until now, this does not seem possible: so, it is clear that the problem has a solution, that will be examined later.

- We have already said that the SUB1 subroutine has an "r" public string variable, say r0: on the retrieval of SUB1, the variable of the subroutine is written = "aa";
- the subroutine SUB1 can pick the "aa" working by means of a STOOL code and can assign the **Workings** entry to

  -*r0.

  - ignoring the minus sign (-) at the beginning, the meaning of the residual part ("*r0") is the format of parametric programming of a string parameter;
  - the minus sign (-) at the beginning is interpreted by the STOOL code as a request for searching the workings not before itself (i.e. not in the SUB1 subroutine, but before the call to the SUB subroutine (that is, in the PRG1 program);
- if the STOOL code is programmed as a Construct code, the "aa" working is still recalled as a Construct by the programmable tool;
- at this point, the SUB1 subroutine must program the interpretation of the working called by the STOOL line and decide what to do.

What to do and how to do it depends on the specific problems and it is not here of primary interest.
We are interested, instead, to provide some indications on the possibility to interpret the working that the STOOL line has recalled: in our example a simple drilling of HOLE code.

How can we examine the working developed by the STOOL line?
In this case also, the programming helps us by means of the multi-purpose function of geometric library. More specifically:

- the **geo[param; ..]** function allows to read the primary information of the STOOL line, as the number of the workings that has developed. To do that:
  - set the name to the STOOL working, for example "tt";
  - use the function with the syntax **geo[param;"tt";"#list"]**, for example in a logical cycle instruction (IF.. ELSE) or of assignment of a "j" variable;
  - the function will return a non-null value, if the working called "tt" has developed some workings.

- The **geo[param; ..]** function allows the operator to read the primary information of the STOOL line as the number of the workings that has developed. To do that:
  - use the function with the syntax **geo[param;"tt";"#tip"]**, for example in a logical cycle instruction (IF.. ELSE) or of assignment of a "j" variable. The function will return a numerical value that corresponds to the typology of the first working, that the "tt" working has developed. More specifically: the 0 value corresponds to the typology of a hole;
  - use the function with the syntax **geo[lparam;"tt";"td",1]** to read the value or the drilling diameter programmed on the working itself.

The section dedicated to the **Parametric programming**, where the use of the **geo[lparam; ..]** function is used, says that it is also possible to expand the inspection on the structure of the STOOL working also in the case of developments at layers higher than the first one; so, you are allowed to create subroutines and macro-programs much bigger and more complex than it was proposed before.

9.6 Automatic Faces

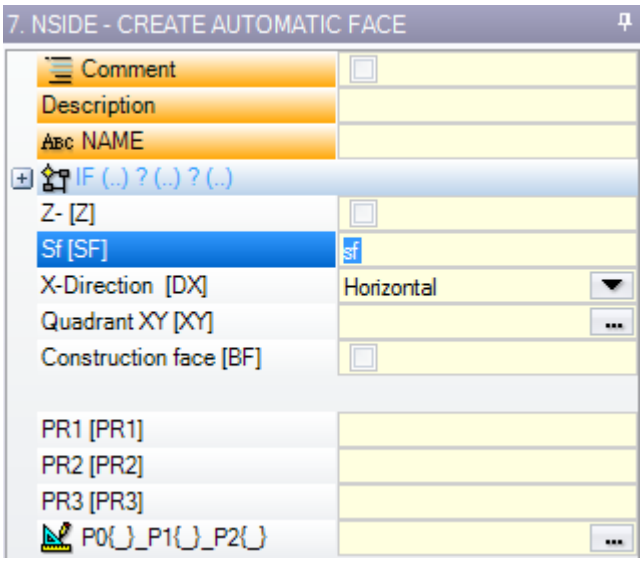


It is an optional operating mode.
The automatic faces are faces directly created during the programming of the piece-face. The face numbering is automatically and sequentially managed, from 101 to 500. The automatic faces can be only seen on piece-face.
The creation of an automatic face enables the following application of the workings to it, always and only in face-piece programming. It is not possible, instead:

- to gain direct access to an automatic face separated view
- to create and/or assign workings to an automatic face from a face different from the face-piece.

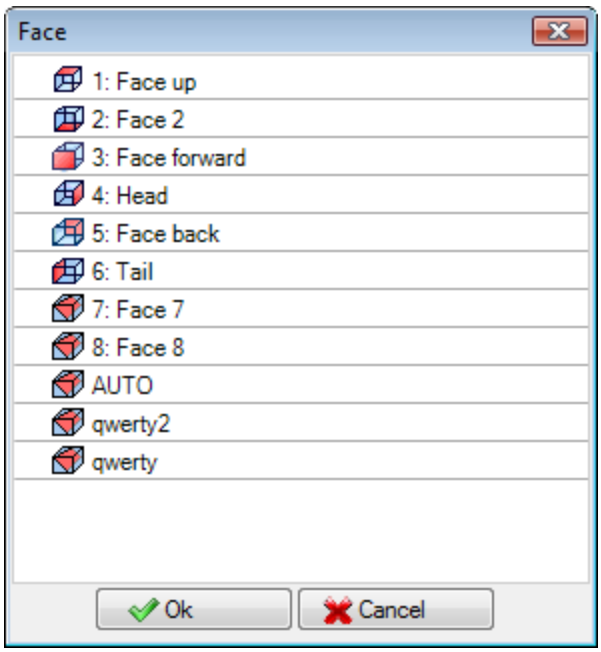
An automatic face cannot be directly selected by means of the number (in automatic way) given to the face: it is possible to access the last assigned face of a face whose name is assigned.
If no names of faces are set or used, the mechanism to use the automatic faces meets the following scheme:

- ...
- assigning an automatic face (automatic number: the first one free, example 105);
- workings are applied to the last created automatic face (last created -> 105);
- ...
- assigning an automatic face (automatic number: the first one free, example 106);
- workings are applied to the last created automatic face (last created -> 106);
- ...



In any point of the piece-face program is therefore available only a specific automatic face: the last created before the selected working. In this sense, we generally call it *application in automatic face*.

Let us now take again the example of selection window opening shown before for the F Field:



In the list order there are:
- the 6 faces of the base piece (in this example, they all can be selected);
- two fictive faces (with numbering 7 and 8)
- an AUTO entry corresponding to the last automatic face assigned before the current program row, construct faces excluded. The automatic face can also be created at a previous expanded level. The last automatic face can also be a NSIDE working applied in a subroutine or macro.
- two rows (last in the list) corresponding to the direct selection of one of the automatic faces assigned before with a name. In this example, we have two automatic faces, whose names are respectively: "qwerty2", "qwerty". In case of faces with repeated names, the last assigned before will be the valid one. From this list, the construction faces are excluded. In case of automatic faces assigned at expanded level (that is, when a subroutine or a macro are called), the same is listed only if the corresponding working in the list of piece-face has an assigned name. In this case, the last face assigned is shown with the same name.

An automatic face is created by the NSIDE working which can be selected in the LOGIC INSTRUCTIONS group of the **Workings** tab. The working can be inserted:
- directly in the face-piece, or

- in the subroutine (or macro) text: in this case it becomes operating when the subroutine is called in face-piece.

- node **"IF (...) ? (...) ? (...)"**: the conditions of application are assigned with a direct IF condition, up to three logical condition terms between two expressions. If the condition is TRUE, the instruction interprets the creation of an automatic face.
- **P0 {} P1 {} P2 {}**: it opens a window identical to the fictive face assignment window to set the corners of the automatic face. The assignment of an automatic face reflects the modes available for the fictive face assignment:
  - reference face
  - three points (P0, P1, P2) to assign the plane
  - Z axis direction
  - thickness
  - graphic representation mode
  - additional parameters
  - setting as **construction face** (which can be used only as reference face for the assignment of a following automatic face and which cannot be programmed)

An automatic face cannot be assigned as *surface*.
If the NSIDE working is called in a face that is different from piece-face (in subroutine or macro-program), it will not be possible to assign a fictive face as reference face.
The graphic representation of the piece-face also includes automatic faces and excludes construction faces.

Programmed induced calls
A programmed induced call can be applied to an automatic face.
With reference to SSIDE code:
- to request application on automatic face, it is necessary to leave the **Induced face** field blank.

In this case, the application will refer to the last automatic face created (listed before).

9.7 Insertion of Geometric Entities from Drawing Menu

The commands to enter directly the geometric entities are in the tab **Apply** of the **Draw** group.
The functionality of the commands is influenced by the setting of the program display (in the status bar): if the entry is not active, a message warns that it is not possible to continue to activate the command.
Select one of the commands of the group to insert geometric elements in interactive way. In Piece-face, the entries of the Draw menu are disabled, if the Box View is active with working currently assigned on a non-real face.

	**Point**: enters a point working. The item is available from the menu, if a default code to work the points is assigned (for the current face or indifferently by face) (see: ***Customize -> Technology -> Default codes*** from **Application** menu).
	**Line**: inserts a linear segment. The item is available on menu, if the COPL01 working code is assigned. Inserts a linear segment, according to the indications provided in the command bar area: **Segment initial point** **Segment final point** If the working COPL01 does not manage the setting parameters of the start point of the segment on the xy plane, it should be possible to hook the linear segment on the previous item in the program list and the start point of the segment is automatically positioned on the hook point.
	**Arc (centre, start, end)**: enters an arc. The item is available on menu, if the working code COPA01 that manages the assigned parameters of the start point of the segment on the xy plane, is assigned. Inserts an arc, according to the indications provided in the command bar area: **centre of the arc**; **radius**; **initial angle**; **final angle**. The rotation of the arc can be defined during the assignment of the first angle. If the **[I]** key is pressed or if the item **Invert rotation** is selected from the contextual menu, the direction of rotation changes from Clockwise (CW) to Counterclockwise (CCW) and vice versa. When the command is selected in the command area, the writing **[CW]** appears in case of clockwise rotation, **[CCW]** in case of counterclockwise rotation.
	**Arc (3 points)**: It inserts an arc defined by three points. The item is available on menu, if the working code COPA04 is assigned. Inserts an arc, according to the indications provided in the command bar area: **initial point of the arc**;

	passage point on the arc; **end point of the arc**. If the working COPA04 does not manage the setting parameters of the start point of the segment on the xy plane, it should be possible to hook the linear segment on the previous item in the program list and the start point of the segment is automatically positioned on the hook point.
	**Arc (Beginning, end, radius)**: inserts an assigned arc through two points and the radius. This item is available in the menu, if the working code COPA11 managing the assignment parameters of the starting point on the xy plane is assigned. It inserts an arc, according to the indications provided in the command bar area: **initial point of the arc**; **end point of the arc**; **radius of the arc**. The rotation of the arc can be defined whilst assigning the radius. If the **[I]** key is pressed or if the item **Invert rotation** is selected from the contextual menu, the direction of rotation changes from Clockwise (CW) to Counterclockwise (CCW) and vice-versa. When the command is selected in the command area, the text **[CW]** appears in case of clockwise rotation, **[CCW]** in case of counterclockwise rotation.
	**Circle**: inserts a circle. The item is available on menu, if the working code COPA45, that manages the assigned parameters of the start point of the segment on the xy plane, is assigned. Inserts a circle, according to the indications provided in the command bar area: **centre of the arc**; **radius**. The circle rotation is assigned clockwise.
	**Circle (2 points)**: inserts an assigned circle through 2 points. This item is available in the menu, if the working code COPA46 managing the assignment parameters of the starting point on the xy plane is assigned. Inserts a circle, according to the indications provided in the command bar area: **starting (and end) point of the circle**; **opposite point on the circle**. The circle rotation is assigned clockwise.
	**Circle (3 points)**: inserts an assigned circle through three points. This item is available in the menu, if the working code COPA46 managing the assignment parameters of the starting point on the xy plane is assigned. It inserts a circle, according to the indications provided in the command bar area: **starting (and end) point of the circle**; **first transition point on the circle**; **second transition point on the circle**. The circle rotation is assigned clockwise.
	**Helix**: inserts an helix. This item is available in the menu, if the working code COPA48 managing the assignment parameters of the starting point on the xy plane is assigned. It inserts an helix according to the indications provided in the command bar area: **centre of the helix**; **final depth, deducted as the distance from the centre along the depth axis**; **radius/starting point**. The rotation of the ellipse is assigned as clockwise and the number of the repetition as equal to 5.
	**Spiral**: inserts a spiral. This item is available in the menu, if the working code COPA49 managing the assignment parameters of the starting point on the xy plane is assigned. Inserts a spiral according to the indications provided in the command bar area: **centre of the helix**; **final depth, deducted as the distance from the centre along the depth axis**; **radius/starting point**; **final radius**. The rotation of the spiral is assigned as clockwise and the number of the repetition as equal to 5.
	**Ellipse**: inserts an ellipse. The item is available on menu, if the working code COPA42, that manages the assigned parameters of the start point of the segment on the xy plane, is assigned. Inserts an ellipse according to the indications provided in the command bar area: **centre** of the ellipse; **edge point** of an axis; **edge point** on the second axis. The ellipse rotation is assigned clockwise.
	**Ellipse (3 points)**: inserts an assigned ellipse through three points. This item is available in the menu, if the working code COPA42 managing the assignment parameters of the starting point on the

	xy plane is assigned. Inserts an ellipse according to the indications provided in the command bar area: **first extreme point** of an axis; **second extreme point** of the same axis; **distance of the centre from the second axis**. The ellipse rotation is assigned clockwise. The starting point of the ellipse is determined on a quadrant change point along the major axis.
	**Arc of ellipse**: inserts an arc of ellipse. The item is available on menu, if the working code COPA43, that manages the assigned parameters of the start point of the segment on the xy plane, is assigned. Inserts an arc of ellipse, according to the indications provided in the command bar area: **centre** of the ellipse; **edge point on the first axis**; **edge point on the second axis**; **initial angle**; **final angle**. The rotation of the arc of ellipse can be defined during the assignment of the final angle. If the **[I]** key is pressed or if the item **Invert rotation** is selected from the contextual menu, the direction of rotation changes from Clockwise (CW) to Counterclockwise (CCW) and vice versa. Command selection is shown in command area by **[CW]** in case of clockwise rotation, **[CCW]** in case of counter-clockwise rotation.
	**Polygon**: inserts a polygon. This item is available on menu, if the working code COPL17 that manages the assigned parameters of the start point of the segment on the xy plane is assigned. Inserts an arc, according to the indications provided in the bar area: number of **sides** (from 3 to 99): the value must be set in a **Dialog box** **Centre** of the polygon **Initial angle**.
	**Rectangle**: inserts a rectangle. This item is available on menu, if the COPL16 working code that manages the assigned parameters of the start point of the segment on the xy plane, is assigned. Inserts a rectangle, according to the indications provided in the command bar area: **First vertex of the rectangle** **Second vertex of the rectangle** opposite the first one.
	**Rectangle (Centre, P)**: inserts a rectangle by assigning the centre and a corner. This item is available on menu, if the COPL16 working code that manages the assigned parameters of the start point of the segment on the xy plane, is assigned. Inserts a rectangle, according to the indications provided in the command bar area: **Centre** of the rectangle **Vertex** of the rectangle.
	**Rectangle (Side, P)**: inserts a rectangle by assigning a side and a point to delimit the opposite side. The rectangle can generally be oriented. This item is available on menu, if the COPL16 working code that manages the assigned parameters of the start point of the segment on the xy plane, is assigned. Inserts a rectangle, according to the indications provided in the command bar area: **First vertex of the rectangle** **extreme point on the first side** point ending the first side **extreme point on the second side** point that determines the position of the side opposite to the one assigned
	**Star**: inserts a star. This item is available on menu, if the working code COPL25 that manages the assigned parameters of the start point of the segment on the xy plane is assigned. Inserts a spiral according to the indications provided in the command bar area: number of **sides** (from 3 to 99): the value must be set in a **Dialog box** **Centre** of the polygon **Initial angle** to assign the external radius **Internal radius** Internal radius and external radius definitions should not be interpreted in an absolute way: in fact, the relationship between both rays can be reversed.
	**Polyline**: inserts a polyline that can be assigned by a contiguous sequence of linear and/or circular (arc) segments assigned by points, according to the indications provided in the commands bar area. The item is available on menu, if the COPL01 working code is assigned. More specifically, it is possible to: • switch from line to arc by means of **[L]** and **[A]** respectively or from the contextual menu by selecting respectively the items **Switch to line** and **Switch to arc** • close a segment on the start point of the polyline using the **[C]** key or from the contextual menu by selecting **Close on starting point command**

	If the working COPA01 does not manage the setting parameters of the start point of the segment on the xy plane, it should be possible to hook the linear segment on the previous item in the program list and the start point of the segment is automatically positioned on the hook point. Switching to arc is available, if the working code COPA04 is assigned.
	**Path**: inserts a path geometric element. This item is available in the menu, if the working code COPL24 managing the assignment parameters of the starting point on the xy plane is assigned. It inserts the element according to the indications provided in the command bar area: **start point of the segment** **end point of the segment** **initial tangent line to the segment** (take the graphic crossbar by placing the mouse in close proximity) **final tangent line to the segment** (take the graphic crossbar by placing the mouse in close proximity) Close the insertion using the **[ENTER]** key.

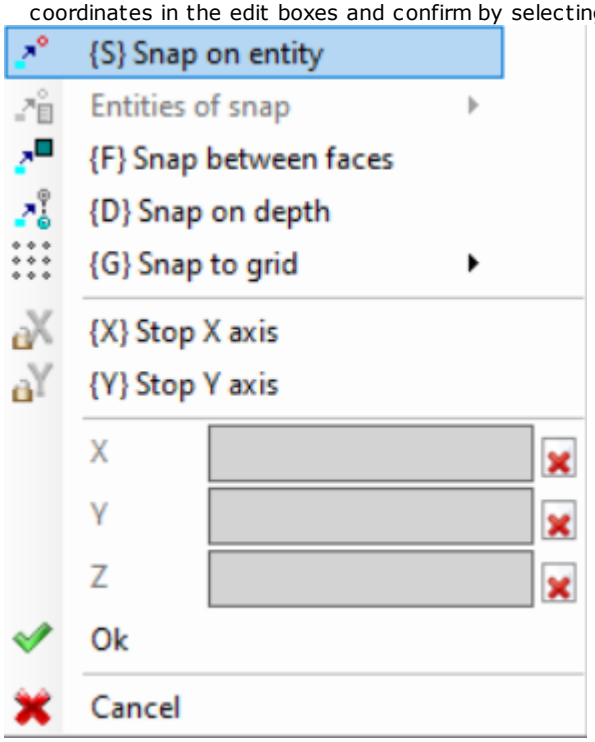
While drawing, the mouse cursor is customized and in the area of the Commands the indication of the element (position on the xy plane, radius, angle, …) that is going to be inserted, is shown.

The insertion step remains active so far as it is not cancelled by **[ESCAPE]** key or by the **Undo** command from the contextual menu.


It is possible to move the mouse with small steps, on one of the two directions, by selecting a direction key (arrows: right, left, up, down) When you release the key, the mouse stops. The applied step, in pixel units, is equal to the double of the value set in *TpaCAD Customization* as **Minimum threshold of the movement of the mouse**. We would like to stress, that the applied movement is not generally associated to one of the coordinated axes of the face or of the piece, but it corresponds to the horizontal or vertical position on the screen.

Opening the contextual menu by means of the right mouse button is always possible: menu composition can always change according to the entity of the selected drawing.

For instance, (x, y, z) coordinates can always be inserted in this way:
- move the mouse cursor to the position required and click the left button
- from the contextual menu that opens by pressing the right mouse button in the graphic area, then enter the coordinates in the edit boxes and confirm by selecting **Ok**.



An obligatory setting condition is highlighted by the display of an image on the left side of the text (exclamation mark) and by a different background colour if the edit field.

In the same way a value with different modes cannot be entered. Be considered the radius of a circle:
- moving the mouse cursor, draw the radius from the centre of the circle to the position required
- from the contextual menu: the value of the radius is directly inserted

To cancel the last setting made in a field, select the icon  near the field.

The availability and the use of the contextual menu of the drawing, as well as the following considerations with regard to the snap modes, applies not only to the step of inserting the drawing menu, but also to other interactive modes of inserting and/or modification of the program: directly from the window of data-entry of the current working, or tool application.

Let us see the other available commands available in the contextual menu:
'P' = Use the last point item is only available when the start point of some geometric entities is to be entered, such as line and arc for three points, and assigns coordinates of last element entered before. The first point of segment can be hooked to previous element in the program list. Hooking is performed by continuing the profile if really possible only. For example, if previous element is a point, command activation locates start point of linear segment on point position, but like the beginning of a new profile. Activation or deactivation can also be performed by pressing the **'P'** key.
'S' = Snap on entity the positioning required is entered also in the coordinates which are determined by the option activated in the menu, that opens by selecting the option **Entities of snap**:
- **Programmed Point**: the positioning is on the programmed point nearest to the cursor. It can be activated also by the **[CTRL+P]** key combination.
- **Nearby point**: the positioning is on the point closest to the cursor (for example: the positioning is found along an arc or a linear segment). It can also be activated by the **[CTRL+N]** key combination.
- **Midpoint**: the positioning is on the midpoint of an arc or a linear segment. It can also be activated by the **[CTRL+M]** key combination.
- **Centre of arc**: the positioning is at the centre of an arc, circle or ellipse. It can also be activated by the **[CTRL+C]** key combination.
- **Intersection point**: the positioning is at the intersection of segments (segments of Path L24 excluded). It can also be activated by the **[CTRL+I]** key combination.
- **Point on perpendicular**: the positioning is along a linear segment, an arc, a circle or an ellipse from the previous application point toward the perpendicular to the same segment. It can also be activated by the **[CTRL+O]** key combination.
- **Point on tangent**: the positioning is along a linear segment, an arc, a circle or an ellipse from the previous application point toward the tangent to same segment. It can also be activated by the **[CTRL+T]** key combination.
- **Point on quadrant change**: the positioning is along an arc, a circle or an ellipse, on the quadrant change point nearest to the cursor position. It can also be activated by the **[CTRL+Q]** key combination.
- **Face edge**: the positioning is on the face edge nearest to the cursor position. More specifically, the positioning exactly falls on one of the face edges, when the pointer position is outside the overall rectangle of the face or very close to it. It can be activated also by the **[CTRL+E]** key combination.

On the status bar the sort of snap is displayed during the drawing. The activation of the snap on programmed entity is restricted to the current insertion and, if necessary, it shall be recalled for the next insertion.

'F' = Snap between faces. If the snap is active on the programmed entity, entity search feature is extended to all the displayed workings, even if they are assigned on another face. Snap between faces can be activated/deactivated also by pressing the **'F'** key and the command selection is shown in the command field by **[F On]** or **[F Off]** writings. The snap between faces is truly operational, if a valid snap entity is found "around" the mouse position or at least if the mouse position "drops" within the representation area of a face. We mean, that the searching is initially limited to a anyway defined graphic area, centred on the click point (see later to display the area of graphic display): if any entity is not identified within this area, but if the mouse position drops within the representation area of a face, we go to look for all the workings in the program list of the same face. The positive outcome of this searching leads to the calculation of the snap selection between faces, otherwise the searching calculates on the remaining modes of selected snaps.
If a point snap typology on the perpendicular, tangent or face edge is selected a snap of **Programmed point** is anyway applied. The selection of snap between faces arranges the activation of the snap on depth, as well.
In the piece-face, you must select the Snap between faces to activate the searching among all the programmed processes in piece-face, even on a face different from the current one: in this case, if it is necessary to restrict the searching only to the workings of the piece-face, you can disable the view of the lists programmed on the other faces.
The **Snap between faces** is always available in Professional operating mode.

'D' = Snap on depth in which snap on programmed entity is active, the required positioning is defined also in the component part of the depth. Snap between faces can be activated/deactivated also by pressing the **'D'** key and the command selection is shown in the command field by the **[ZETA On]** or **[ZETA Off]** writings. The snap on depth is truly operational if permitted by the active snap entity. More specifically, the snaps on point are excluded: intersection, perpendicular or tangent

'G' = Snap to grid the required positioning is defined as vertex of the grid nearest to the click point of the mouse. Snap to grid can be also activated/deactivated also by pressing the **'G'** key and the command selection

is shown in the command fields by the **[G On]** or **[G Off]** writings. Snap to grid can be activated even if the grid is not displayed, otherwise the snap is applied to the displayed grid (grids, points).
In the case of snap applied to the lattice grid, it is possible to apply another one option, working on one of the three selections proposed on the menu of the snap voice:

- Snap to grid: it magnetizes on the points of the grid (default functioning)
- Point on horizontal straight line: it magnetizes on the points of the grid in vertical snap and then it moves discretely on the horizontal straight line thus identified.
- Point on vertical straight line: it magnetizes on the points of the grid in horizontal snap and then it moves discretely on the vertical straight line thus identified.

'X' = Stop x axis and **'Y' = Stop y axis** prevent the cursor from moving toward X or Y. Activation or deactivation can also be performed by pressing the **[X]** or **[Y]** keys. The movement blockage is employed in the current working or it is added to the selected snap entity, if the snap is enabled. The stop in one direction automatically unlocks that in the other one.

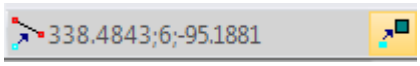
T = Tangential segment is available in case of insertion of a segment that continues a profile and forces the segment (linear or arc) to exit as a tangent line from the profile itself. Assignment modes of the geometric segment are as follows:

- an arc is determined by positioning the final point
- a linear segment is determined by positioning the final point, in any case redirected on the tangency direction, or by giving the length of the segment itself.

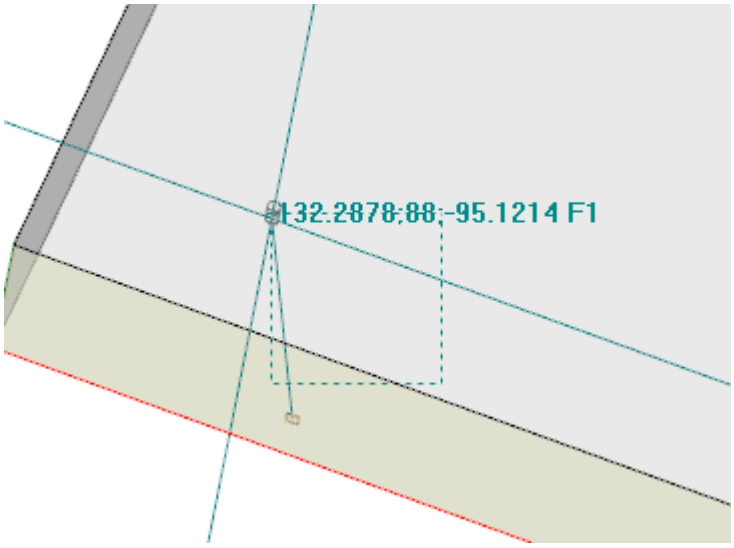
Activation or deactivation can also be performed by pressing the **'T'** key.

'Z' = Cancel last segment is available in case of insertion of a polyline and cancels the last inserted segment. Activation or deactivation can also be performed by pressing the **'Z'** key.

When inserting from the drawing menu on anyway in interactive mode in the status bar following information about the acquisitions made is shown:



position deducted for the mouse further to snap procedures. More specifically: the figure on the left shows the snap typology that has been calculated (for example: programmed point, midpoint...)  the selection of the box shows that a snap on faces has been calculated.
In interactive procedure, in an entity snap is required, it is possible to view the cross-hair used for the snap solution, by selecting the  box in status bar. The picture clarifies how it works:



the hatched area shows the dimension according to which the graphic searching area of the snap entity has been enlarged: we may indicate it as snap cross-hair. As already told, the searching graphic area is restricted and starts from a minimum dimension of some pixels and it is enlarged - always to a maximum area - until a correspondence is found (see also: Customize TpaCAD -> Views -> Mouse).
The figure can correspond to the case of a programmed hole translation in face 3 on a hole in face 1: the graphic highlights how the searching area has been enlarged to include the hole in face 1. The mouse position corresponds to the central point of the snap cross-hair.

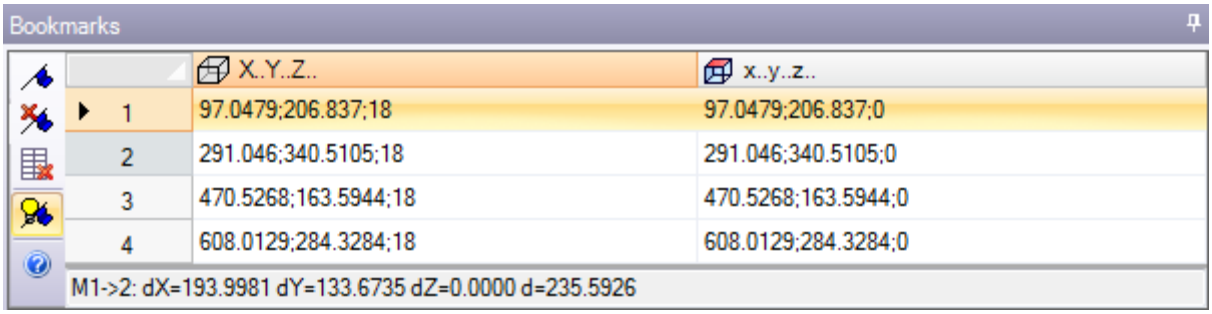
Much has been said on the selectable snap modes: of entity, depth, face.

Anyway, it is obvious that the most elementary condition for an interactive placement works without any other active snap. In his condition, the two-dimensional position of the graphic pointer is brought back to the three-dimensional "world position", by applying a conversion from the coordinates of the screen at the xy definition plane of the current face to the z coordinate with null depth.

9.8 Inserting Bookmarks

PROFESSIONAL

Bookmarks are auxiliary entities that can be added to the graphical display in order to mark important positions. For example, it is possible to assign bookmarks in snap mode on intersection points, centres, quadrant change point.
Bookmark management is located in a dedicated tag and it is enable during the TpaCAD configuration phase and always in Professional mode only.



The sidebar corresponds to the following commands:

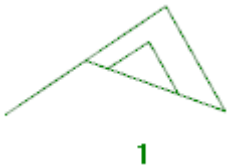
Assign a bookmark: the command is active in face view and starts the interactive procedure to assign a new bookmark position. Procedure management mode is similar to the drawing commands. More specifically:
- it does not manage the **Snap between faces**;
- the **Snap on depth** is forced to the active status, without possibility to deactivate the selection (a snap on entity always applies snap in depth).

To confirm the position a new row in the table is assigned, where the bookmark position is indicated in absolute piece coordinates (column: X..Y..Z..) and current face (column: x..y..z..). To quit the procedure, press ESCAPE.
You can assign up to 50 bookmarks.

Removes the bookmark in correspondence to the selected row in the table.

Removes all bookmarks in the table.

Show bookmarks: select the field to see the bookmarks represented.



A bookmark is displayed by a symbol similar to a flag and a number that corresponds to the progressive number of the row in the table.

If there are some bookmarks, the interactive procedures of acquiring coordinates can select the snap also on these entities from the selection menu of **Entities of snap**:
- **Bookmark**: the positioning is on the programmed point nearest to the cursor. It can be activated also by the **[CTRL+F]** key combination.

The bookmarks belong to the program as a whole and the positions are automatically adapted to the current face reference. The table of the bookmarks remains unchanged when a program closes: reset must be required by selecting the command from the command bar.
The tag of Bookmarks makes also the distance between two positions available:
- bring the selection on a row of the table (as in the figure: the line 1);
- then move the mouse to a different line.

In the area under the table are shown the distances calculated between the two positions for each of the three axes of the absolute reference (dX, dY, dZ) and as absolute distance (d). As in the figure: between the first two positions in the table (M1->2).

9.9 Change and Insertion

Selecting the insertion point in a program list

In face view the current working is highlighted both in ASCII text area and in face graphical view. The working assignment area shows the parameter settings of the working.

Scrolling and selecting the active working in graphical view
The face graphical view is made interactive by mouse clicking on the display frame.

Situations are managed as follows:
- **direct pointing to a working (click in the area)**: it moves the active working to that which is nearest to the position pointed by the mouse. More specifically, the searching is performed on the complete list of programs from the first to the last block, but only for the workings which are displayed at the moment. Logical (IF, ELSE, ENDIF) or comment workings or workings not displayed for view filter application are excluded from the searching. All face selections are reset to zero. We need to clarify if the 3D representation of the piece is active: in this case the working is searched at a graphic level by activating a procedure identifying the working represented within a specific search area around the position of the mouse. If the search is successful, the identification of the working is confirmed, otherwise the *click* is ignored: this means that the graphic identification of a working requires selecting one position *near* the working itself. To determine the video search area, please read the paragraph concerning ***Customize -> Views -> Mouse***.
- **scrolling the program**. The following keys are available:
  - **<Arrow up>**: moves the active line to the previous block in the list
  - **<Arrow down>**: moves the active line to the following block in the list
  - **<Previous page>**: moves upward the active line of a page (the dimension of a page is fixed at 10 lines)
  - **<Next page>**: moves downward the active line of a page
  - **<Home>**: moves the active line to the first block of the program
  - **<End>**: moves the active line to the last block of the program.

In the here listed cases, the current working cannot be displayed.
Every time the active working changes, the face selections are reset or remain unchanged according to the setting **Add selections** in **Customize -> Environment -> Activity**.

Scrolling and selecting the active working in ASCII text
It is possible to scroll the program directly on the ASCII text.
Situations are managed as follows:
- direct pointing to a working (click in the area): it moves the current line to the mouse pointed row. All face selections are reset.
- Scrolling the program. The available keys are the same as those used to scroll the program in graphic representation. Every time the active working changes all face selections are reset.

Scrolling and selecting the active working with menu commands
It is possible to select the current working also using the available commands in the group **Place at line** from the tag **Edit**:

	**First working**: It moves the active working to the first working in the list.
	**Last workingit moves the current working to the last working in the list.**
	**Previous working**: it moves the current working to the previous working. (See also the item: **Matching on profiles**).
	**Next working**: it moves the current working to the next working. (See also the item: **Matching on profiles**).
	**Matching on profiles**: if the item is active, the two previous commands consider the profiles as a single entity, assigned on the setup or, in the absence of setup on the working, on the working of profile start. If the item is not active, the two previous commands browse through the program list and apply a correspondence for each single line.
	**Profile start working**: it moves the current working to the working of initial profile: it works, if the current working belongs to a profile. This command can be available also in the ASCII Text and Graphic context local menu.
	**Profile end working**: it moves the current working to the profile end working: it works, if the current working belongs to a profile. This command can be available also in the ASCII Text and Graphic context local menu.

	**Go to line...**: it moves the current working to the working of the assigned progressive number. This command can be activated also from the status bar, by clicking the area that shows the progressive number of the current program line and the total number of the lines.
	**Next correspondence**: with active item, a click of the cursor in the graphic area makes a search on the program list starting from the current working, until the last block With non-active item, the searching is anyway performed on the whole program list and always from the beginning of the list. Being a match of graphic type, the workings that are not displayed in the graphic area, are excluded from the searching. In case of overlapped workings, this kind of selection allows the scrolling of all the working assigned in the same position. If the searching fails, the selection of **Next Correspondence** is automatically reset.

Inserting with respect to the current working

In the status bar we find the icon  :
- with active selection, the workings are inserted after the current working;
- with non-active selection, the workings are inserted before the current working.

If the working is directly entered in the middle of a profile, the entry point can be moved before or after the profile itself.

Selection

The selection of workings is enabled only in face view and with face program not empty. It is not possible to make partial selections of a complex working (subroutine or macro) or of a multiple profile segment.

How workings in graphical view are selected

The following key combinations are managed according to the priority order assigned as indicated by the sequence of points:

To draw the window, press the left mouse key and drag it until you get the window required.
The workings enclosed in the window are selected. More specifically:
- the only workings shown in the graphical view are taken into account (it applies active views and view filters);
- workings conditioned by logical statements or commented are therefore excluded from the search;
- the selected area does not change the active line;

[Shift+(left mouse key pressed)]: starts the area selection
[Shift+(left mouse key pressed)]+[ALT]: the selection is extended to complete the profiles which are even partially enclosed in the window;
[Shift+(left mouse key pressed)]+[CTRL]: selects the area with the workings keeping the previous selections

In piece-face the area selection affects the current face only.

[CTRL + (left mouse key pressed)]: select or deselect the working closest to the pointed mouse position.
- the only workings shown in the graphical view are taken into account (it applies active views and view filters);
- it also the **[ALT]** button is hold down and if the selection affects a profile element, the selection is extended to the whole profile.
- previous selections are maintained;
- the active line does not change.

In the graphic representation the selected workings are coloured according to the settings specified in the TpaCAD customization.

How to select the workings in ASCII test

The following listed key combinations are managed according to the priority order assigned as indicated by the sequence of points:

[Shift+(left mouse key pressed)]: selects from the active line to the program line pointed to with the mouse.
- previous selections are lost.
- the active line does not change

[CTRL+(left mouse key pressed)]: selects or deselects the program line pointed to with the mouse.
- if the **[ALT]** key is also pressed and the selection affects a profile element, the selection is extended to the whole profile.
- previous selections are maintained.
- the active line does not change.

In the ASCII text the workings that are not represented in the view can also be selected.

General Selection Commands

These commands are located in the group **Modify** of the **Edit** tag.

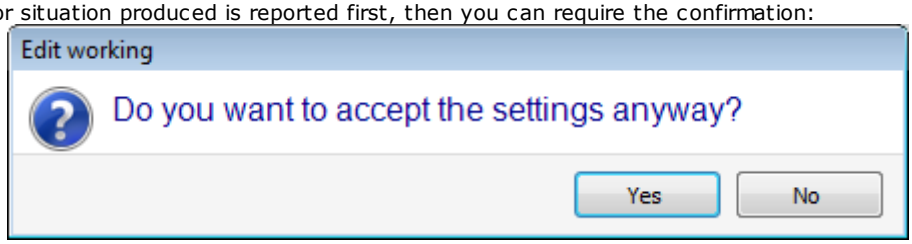
	**Select all (CTRL+A)**: select all workings of the face. In case of piece-face two situations have been recognized: • when the 3D view or box is active, the selection concerns the whole working list; • when the 2D view is active, the selection concerns the workings applied on the face in current view only
	**Delete all selections**: it cancels all face selections. In Face view the command is kept enabled, but it is almost obsolete: the face selections are automatically cancelled by a "click" on the working list, in graphic representation or ASCII text. The interest results from the fact that the command is enabled also in Overall view, cancelling the selections of each face.
	**Find and Select**: it opens a window in which you can define the criteria research to be applied to the programmed workings and where the selection can be applied. This command is fully analog to the ***Replace*** command, to which reference is made for more details. It is about Select instead of Replace. This command is enabled both in overall view and in face view in this case with non-empty face program.
	**Group together**: this command moves all the selections of the face to the bottom of the list and makes them consecutive. At the end of the grouping, the selections remain unchanged. This command is enabled in face view.

These commands can be available in the **ASCII text** and **Graphic context** local menu:

	**Select from here to the beginning of the profile**: this option selects the part of the profile between the current working and the beginning of the profile to which the current working belongs.
	**Select from here to the end of the profile**: this option selects the part of the profile between the current working and the end of the profile to which the current working belongs.

Change of the active working

The direct change of a working consists in changing parameter settings and/or working properties.
The direct change is always applied to the active working and cannot be applied to workings in locked status (it is about an induced call or it has Level, Construct or "O" field locked) or with invalid operating code.
The active working is changed only if no error messages concerning the settings of the working itself are signalled. In this case, if an error situation is reported, you need to solve the wrong situations or to cancel the modifications.
You can also ask to apply the change of working even in the event the operation has reported an error. In this case, the error situation produced is reported first, then you can require the confirmation:



The enabling power to validate error situation in editing and insertion is available in **Customize** (section **Environment**, page **Activity**, entry: **Allow confirmation of working compiled with an error**).
Once the modification has been confirmed, the working of the program list which follows that which has been changed becomes the current working.

General commands of Change in face program

Commands allowing the change of an established group of workings are available. An example of established set of workings is represented by the application of special views and/or view filters. As for general edit commands, it is necessary to create the selected set of workings before choosing the edit command.
Examples of established set of workings are the following ones:

- the selected workings which verify logical conditions
- the selected workings which are assigned on a given level.

Many edit commands consider as selected privileged group of workings, the group which is composed of selected workings. In this case if no selection is active, the current working is edited.

The general editing commands are examined in the next paragraphs.

Change of Properties

The change of properties is possible only in face view and with face program not empty.
The commands are located in the group **Assign property** of the **Edit tag**.
In the window shown it is possible to enable some criteria to apply the assignment:

- **View match**: if enabled, it takes into account the only workings represented in the current view (it applies active views and view filters). For more details on possible situations see the examination of the **Find and Replace commands**.
- **Apply to selected workings**: if enabled, it considers the selected workings only. The selection is available, if there are selected workings.

The application of the assignment is always conditioned also by possible lock filters in editing.

In case of piece-face two situations have been recognized:

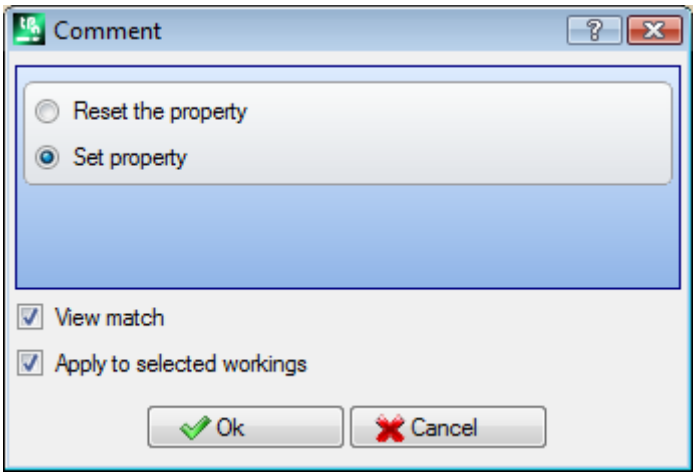
- when the 3D view or box view is active, the change concerns the whole working list;
- with the 2D face view active, the only workings applied to the face in current view are subject to editing.

The property fields may not be changeable (L, B, K, K1, K2 always and possibly M and O) if they are not configured to be directly edited in profile in the following cases:

- profile workings (arcs and lines) if the working opens a profile (open profile) the value is still 0, otherwise they take the value from the profile start working;
- in case of setup or complex working, if point hook is required, the non-editable properties in the profiles are propagated from the starting working of the profile.

It is possible to start the command of the group **Assign property** of the tab **Edit** also from the ASCII text area, by clicking the header cell of the column corresponding to the property.

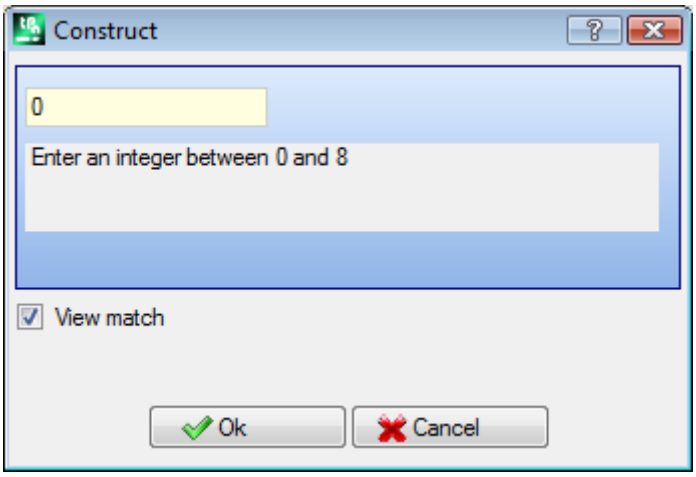
C Property or Comment
is an optional property.



It sets the (C) comment field.
The selection of the option **Reset the property** takes away the setting of the C properties from the workings that concern the modification. In this case the assignment does not verify the active view filters, because the possible commented program lines are not represented on the screen. The workings that had a set field now become irrelevant for the program. The selection of the option **Set property** makes the workings concerning the modification become comment workings: they remain in the list, but they do not affect the program. If a working has the "C" property active it can be edited only after disabling the comment. The C property can be activated also on case of assigned working with invalid operating code.
As far as the Comment property is concerned, we remind the reader that, when we refer to the previous or next working, with respect to another, it has always to be meant "commented workings excluded".

L Property or Layer
is an optional property.
The meaning and the assigning mode of the L field depend on the TpaCAD configuration in **Customize -> Environment -> Working edit** of the Application menu.
The window shows a case of selection in the value list to be attributed to the workings. As an alternative, the selection can occur in a direct edit field (as for the next case of: *B property or Construct*).

B property or Construct
is an optional property.



The meaning and the assigning mode of the B field depend on the TpaCAD configuration in **Customize -> Environment -> Working edit** of the Application menu.
The window shows a case of selection in the direct edit field of the value to be attributed to the workings. As an alternative, the selection can occur in a direct edit field (as for the previous case of: *L property or Layer*).
A working indicated as a construct is compiled but not executed.

O Property
is an optional property.
The meaning and the assigning mode of the O field depend on the TpaCAD configuration in **Customize -> Environment -> Working edit** of the Application menu. The window can show a selection in the list of a direct edit field of the value to be attributed to the workings.

M Property
K Property
K1 Property
K2 Property
They are optional properties.
It is possible to assign the value of the property only by direct edit.

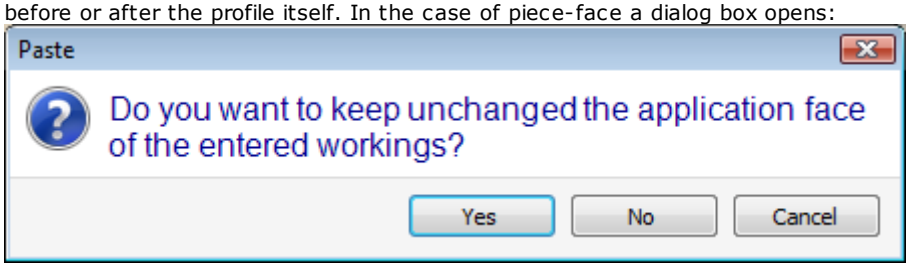
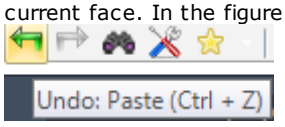
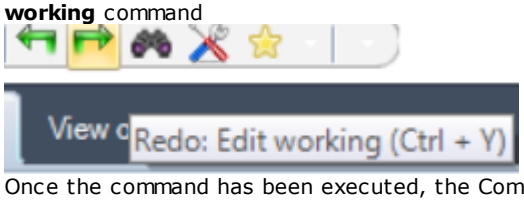
N Property or Name
is an optional property.
To the N property is assigned a string containing no more than 16 alphanumeric characters and the first character must be alphabetic. After confirming the window whose assignment field is empty, a confirmation is required deleting the Name of the concerned workings. For example, the property is used to apply programmed tools or for particular functions of parametric programming.

Face Property
it is available in Piece Face only and assigns the application face to the workings that concern the modification. The selection of the face occur in the list and the listed items correspond to the available real and fictive faces.

General Purpose Change Commands

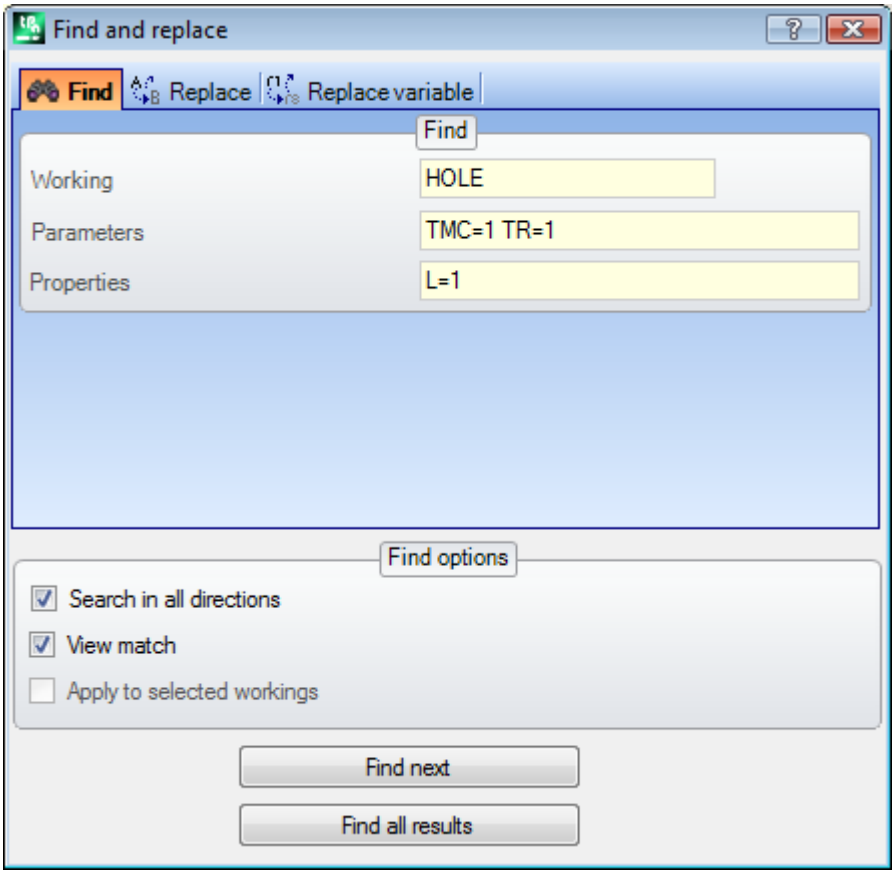
It is possible to apply the Copy, Paste, Cut, Delete, Delete All, Undo. These are command that are enabled in Face View only.
These commands operate on selected workings, if any, otherwise on the active working. They affect only the workings that verify the active view filters: selections, logical conditions, layers, special filters.
In the case of macro-program text, selecting "apply to a profile", this is evaluated by including the logical workings that may break up the construction of the profile.
The commands are available in the group **Clipboard** of the **Edit** tab.

	**Copy (CTRL+C)**: it copies the selected workings in the local Clipboard of the application. If the copy concerns one only working that belongs to a profile, the copy can be confirmed within the whole profile. In the case of piece-face two situations have been recognized: • when the 3D view or box is active, the copy concerns the complete working list; • when the 2D view is active, the copy concerns the workings applied on the face in current view only. This command can be available also in the ASCII Text and Graphic context local menu.
	**Paste (CTRL+V)**: it pastes the Clipboard content at the insertion point (before or after the current working).

	In the case a working is directly entered in the middle of a profile, the entry point can be moved before or after the profile itself. In the case of piece-face a dialog box opens:  • **[Yes]**: field programming remains unchanged **Application face** of the entered workings • **[No]**: it copies in the Application face of the entered workings the value of the current face This command can be available also in the ASCII Text and Graphic context local menu.
	**Cut (CTRL+X)**: it deletes the selected workings with a copy into the Clipboard of the application. If the deletion concerns one only working that belongs to a profile, the deletion can be confirmed within the whole profile. The command can be applied to the workings locked (layer, construct or locked O field). In the case of piece-face two situations have been recognized: • when the 3D view or box is active, the deletion concerns the complete working list; • when the 2D view is active, the deletion concerns the workings applied on the face in current view only. This command can be available also in the ASCII Text and Graphic context local menu.
	**Delete**: it deletes the selected workings, in the same way as the those of the previous command, but without copying them in the local Clipboard of the application. **(DEL)**: with active focus on the graphics of the piece or the ASCII text list, selection of the DEL key executes the **Delete** command.
	**Delete All**: it deletes all the workings without copying them into the local Clipboard of the application and without evaluation of the active view filters. In the case of piece-face two situations have been recognized: • when the 3D view or box is active, the deletion concerns the complete working list; • when the 2D view is active, the deletion concerns the workings applied on the face in current view only.
	**Undo (CTRL+Z)**: it cancels the last edit command of the face program. Moving the cursor to the command icon, a tooltip shows which is the first command in the list that can be cancelled for the current face. In the figure below a **Paste** command  Once the command has been executed, information about the cancelled action is provided in the Commands bar area. The list of actions (commands) which can be cancelled is cleared when the active program is closed. The command is available also in the Application menu toolbar. In Face View it is possible to cancel also some edit commands that have been executed in Overall view: for example, a comprehensive substitution of parameters (see later **Replace** command).
	**Redo (CTRL+Y)**: it redoes the last cancellation of face program modification, required by **Undo (CTRL+Z)**. By moving the cursor over the icon of the command, a tooltip indicates which is the command first in the list that may be restored to the current face. As shown in the figure: an **Edit working** command  Once the command has been executed, the Command area shows the indication of the restored command. The list of commands that you can restore corresponds to cancellations that have been required without a subsequent restoring. The list is always cleared when the active program is closed. This command is available also in the menu toolbar of the **Application** menu.

Find

The **Find** command  is enabled in face view only and with the face program not empty. The setting window for the data searching is recalled from the **Modify** group of the **Edit** tab.



The window opens and shows the settings as they have been assigned to the previous command recall. Furthermore, the tag stops are available within the commands, as follows: Replace, Replace variable.
The first selection of the button **[Find next]** allows the user to start searching for the first working of the face that verifies the assigned criteria:

- **Working**: it is the ASCII code of the working to be searched (in the example: "HOLE"). If the field is not assigned, the searching is not applied to the ASCII code. The  button nearby allows to set the field to the current working code.
- **Parameters**: assignment of parameters to be assigned (in the example: "TMC=1 TR=1"). If the field is not assigned, the searching is not applied to the parameters. The field must assign items split by space, where each item is called with the ASCII name of the parameter followed by the value as programmed; for names containing decimal figures or for a parametric assignment the form "name=.." is necessary.
  Valid example: "TMC=1 TD=r5 P1=12" where:
  "TMC=1" associates the value 1 (numerical) to the "TMC" parameter, "TD=r5" associates the value r5 (parametric) to the "TD" field, "P1=12" associates the value 12 (numerical) to the "P1" field. To shown the searching for a non-set parameter (that is: empty field), provide the name only (followed by "=" if the same contains decimal figures). Examples: "TMC=" and "TMC" are equivalent forms "P1=" is the obligatory form for the "P1" parameter. A change in the field can determine automatic changes due to automatic checks. More specifically, the parts recognized as name of parameters are assigned with capital letters.
- **Properties**: property assignments to be searched: (in the example: "L=1"). If the field is not assigned, the searching is not applied to the properties. The field must assign the items separated by space, where each item is called with the property name (L for Layer, B for construct, then: O, M, K, K1, K2) followed by the associated value (for the K1 and K2 fields or for a parametric assignment the "K1=.." form is obligatory).
  Valid examples:
  - "L4 M5000 K12 K1=5" where: "K12" associates the value 12 to the K field, "K1=5" associates the 5 value to the K1 field.
  - "L4 M=r5 K=12 K1=5" where: "M=r5" associates the "r5" parametric setting to the M field, "K=12" associates the value 12 to the K field with a form equivalent to "K12".

Unlike the parameters, a property is always considered set with default setting = "0". A field change can determine automatic changes due to automatic checks. More specifically, the parts recognized as name of properties are assigned with capital letters.

Following options can be selected in the Find options field:
- **Search in all directions**: if enabled, it starts the research throughout the list, otherwise after the current working only.
- **View match**: if enabled, it takes into account the only workings represented in the current view (it applies active views and view filters). Let us see in details the views and the applied filters:
  - the search excludes the workings: logical, with active C field or with operational invalid code (read: the working has no correspondence in the working database);
  - if the View of Selections is active: it considers the only selected workings;
  - if the View of logical Conditions is active: it considers the only workings that verify the logical conditions, including the exclusions;
  - if the View of the Filters of the layers is active: it considers the only workings assigned with displayed layer;
  - if the View of the special Filters is active: it considers the only workings verified by the special view filters (fields: B, O, K, K1; technology).

The **View match** field can be changed, if the program display field (in the status bar) is active. Otherwise, the item cannot be selected.

- **Apply to selected workings**: if enabled, it considers the selected workings only. The selection is available, if there are selected workings. The activation of the option is considered only if the item **View match**, by which it is already included, is not enabled.

In the case of piece-face and if the **View match** field is selected, two different situations are recognized:
- when the 3D view is active, the search is applied to the complete list of workings;
- when the box view is active, the search is applied to the workings applied to the real face of the piece only;
- when the 2D view is active, the search is applied to the workings applied on the face in current view only.

When the **View match** is not selected: the search always is applied to the entire list of workings.

The **[Find next]** button allows the user to start or to keep on searching:
- Button selection is not available if none of the searching fields (**Working, Parameters, Property**) is assigned;
- a message notifies the negative result of the search;
- otherwise, the working found becomes the current working.

The search is performed here without the application of possible conditions that prevents from changing the workings (example: L field locked).

The button **[Find all results]** allows to fine all the correspondences.
The search result is shown in the window of *Commands*.

Replace

The **Replace** command  is enabled both in overall view and in face view and in this case with not empty face program. The setting window for the data searching is invoked from the **Modify** group of the **Edit** tab.
The window opens and shows the settings as they have been assigned to the previous command recall.
Furthermore, the managed tabs are available within the commands, as follows: Find, Replace variable.

In field of Find are set the data that define the search criteria (see **Find** command):
- **Working**: it is the ASCII code of the working to be searched.
- **Name**: name of the working to be searched.
- **Parameters**: assignment of parameters to be searched.
- **Property**: assignments of the property to be searched.

In the Replace with field are set the new data that must be assigned:
- **Working**: Working ASCII code (it must correspond to a valid working and be applicable in the program list).
- **Parameters**: Parameter settings. To assign the field we refer to what has been provided for the corresponding field in the Find area. More specifically, to show that the parameter assignment must be deleted (let the field empty), the name only must be provided (followed by "=" if the same contains decimal figures). Examples: "TMC=" or "TMC" are equivalent forms "P1=" is the obligatory form for the "P1" parameter.
- **Properties**: property settings. To assign the field we refer to what has been provided for the corresponding field in the Find area.

The selection box flanking the fields enables the assignment to be performed of Working code or Parameters or Properties.

If the **Replace** command is activated in Overall view:
- Find options is not available;

- It is available only the button **[Replace all]**: replaces all the workings of the entire program that verify the search criteria with the new data set. The button selection has no effects, if the selected replacement field (**Working**, **Parameters**, **Properties**) is not set.

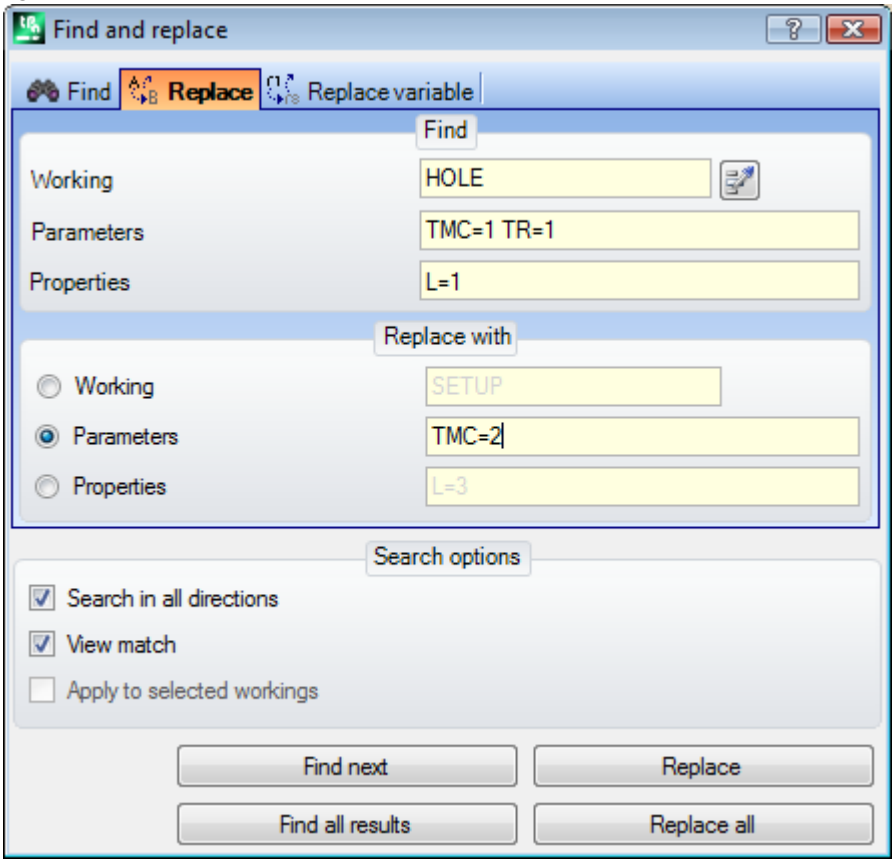
If the **Replace** command is activated in face view:
- the Search options command is available (see command: Find);
- the following buttons are available:
  [Find next] allows the user to start or to keep on searching without any replacement.
  A message notifies the negative result of the search.
  In case of match found: the working found becomes the current working.
  Unlike the information in the *Replace* tab search here is performed by applying possible conditions that prevent working changes (example: locked L field): the workings that have been found valid for the replacement can be less than that verify the simple search.
  If all field of the Find are empty, the search only applies the Search options and the general change working conditions.

  [Replace] applies the replacements to the match working found with the data set;
  [Replace all] replaces all workings of the face that verify the search criteria with the new data set.
  The selection of the replacement buttons has no effects, if the selected replacement field (**Working**, **Parameters**, **Properties**) is not assigned.
  [Find all results] finds all the correspondences and shows the result in the window of Commands.

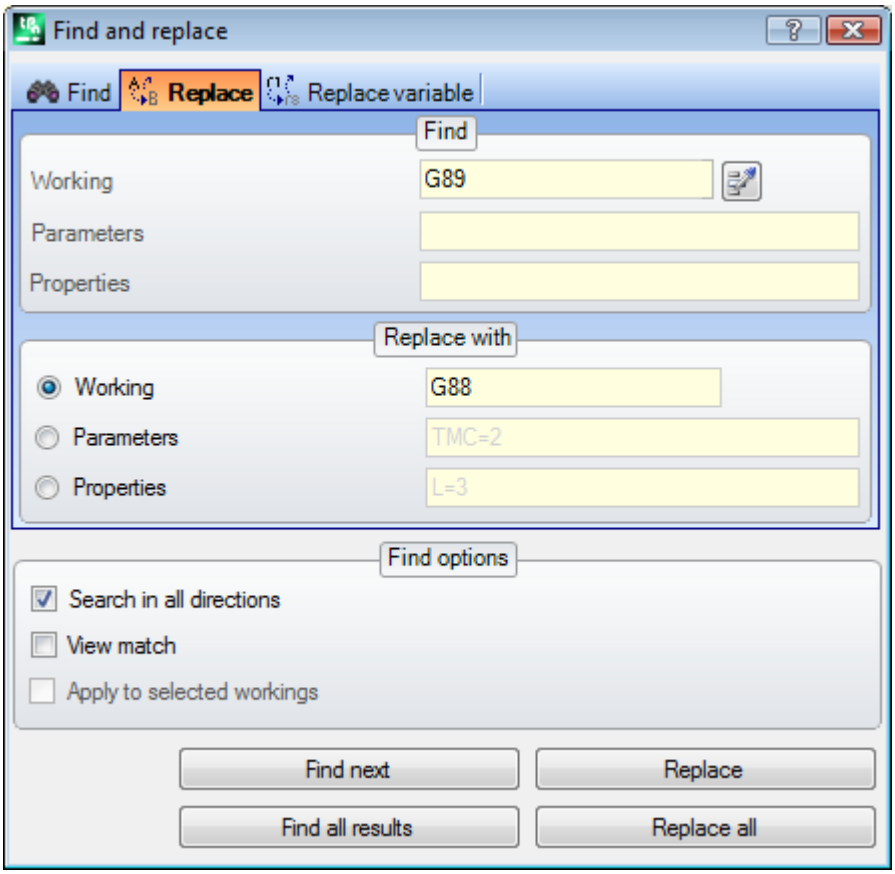
In the case of piece-face and if the **View match** field is selected, following situations are recognized:
- when the 3D view is active, the search is applied to the complete list of workings
- when the box view is active, the search is applied to the workings applied to the real face of the piece only
- when the 2D view is active, the search is applied to the workings applied on the face in current view only.

As already told, possible replacements also performed by Overall View, can be later cancelled by selecting Face View.

The setting parameters shown in the Figure below can be replaced by specific working parameters: Only the "HOLE" workings, where the settings for the parameters ("TMC=1 TR=1") and property ("L=1") are indicated, are affected by the replacement. For the processes that verify the matching criteria, the "TMC=1" assignment is replaced by "TMC=2 ".



The setting parameters shown in this second figure require an operating code substitution. The "G89" workings are the only ones to be replaced by workings called "G88" without any other setting.



Suppose to open a program which uses invalid working codes in the current program configuration (in the example: "G89" does not match any working operating code).
A solution to make valid workings is to replace "G89" workings with a valid working (in the example above: "G88").
However, the search function shall deactivate the **View match** option, since "G89" workings cannot be represented in graphical view.
The button **[Find next]** allows the user to keep on searching without any further replacements, the button **[Replace]** performs replacements for the current working, the button **[Replace all]** performs replacements in all face workings that match the assigned criteria.

Replace variable

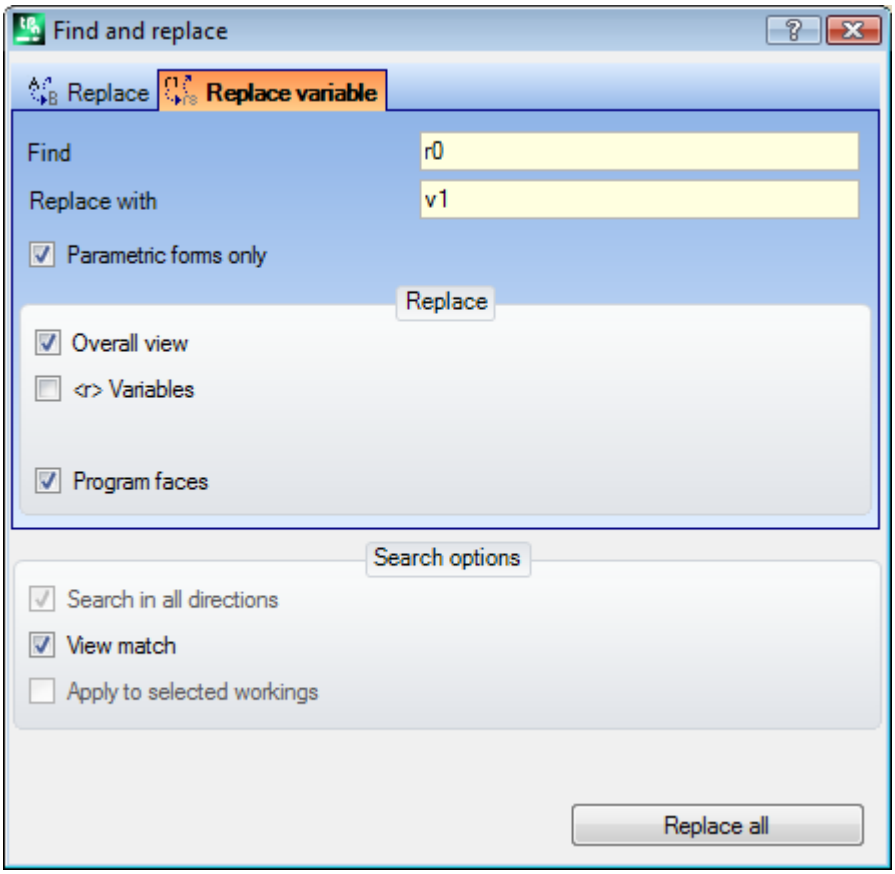
The window opens and shows the settings as they have been assigned to the previous command recall.
Furthermore, the tag stops are available within the commands, as follows: Find, Replace.

It finds the occurrence of the assigned parametric form and replaces it. The **Replace variable** command is enabled both in Overall View and in Face View (with non empty face program) and it is different in both the cases.
The setting window for the data searching is invoked from the **Modify** group of the **Edit** tab.
The command allows the assignment of a search with possible replacement of:

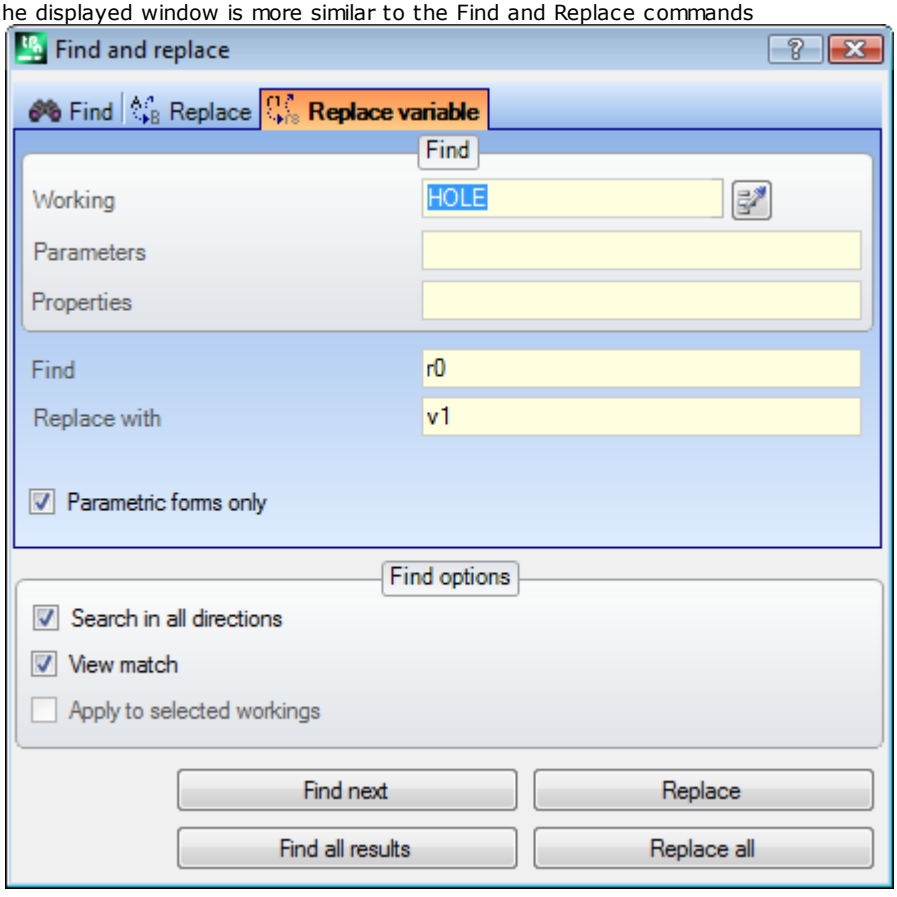
- o, v, r, j variables (in $ macro-program text only). For example, it allows the operator to replace the use of a <v> variable with a <r> variable (it replaces "v5" with "r15")
- variables and/or variable arguments. For example, it allows the user to replace "r5" with "lf", "lf" with "r\dim", "r\dim" with "100.6"
- generic substrings. For example, it allows the replacement of "geo[lface;" with "geo[isface;", "r5" with "abs[r5]", "r5" with "-100.6".

In Overall View is possible to require the replacement also in the general assignments of the program (variable, variable geometries) as well as in the face programs.
In face view is possible to assign some additional search criteria in a manner totally similar to the Find/Replace commands.

In Overall View the window is as follows:



In face view, the displayed window is more similar to the Find and Replace commands



In overall view, the Replace field allows the user to choose the search ranges where the replacement should be performed:
- General View or Current Section as in the picture: <r> Variable,
- Face program.

In face view, in the Find field the data defining the search criteria are set (see Find command: Workings, Parameters, Property).

The settings for the variable assignments are in the two fields:
- **Find**: variable parametric form to be replaced
- **Replace with**: parametric form to be replaced.

Both field must be assigned.

The option **Parametric forms only** select the form declared for the **Find and Replace** with:
- if selected, it declares that both the fields assign a parametric form of variable or of variable argument. Valid forms are, for example: "r5", "r\dim", "o7", "lf";
- if not selected, it declares that both the fields assign generic sub-strings.

In the first case, search and replacements are performed upon syntax check of the set fields and the strings are replaced if they are not preceded by figures or letters. For example, consider to replace "r1" with "r\abc":
- the "lf+r1/4" string is modified as "lf+r\abc/4"
- the "lf+pr1/4", "lf+r12/4" are not modified.

An error report shows if the field are not correctly set.

In the second case, search and replacement are not subject to a syntax check.

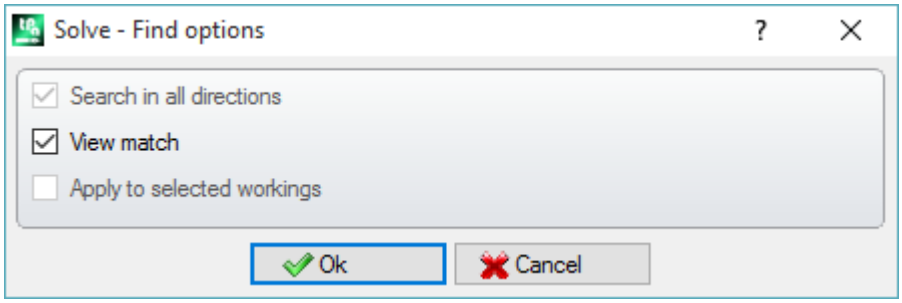
The **[Find next]** button allows the user to start or to keep on searching without replacements, the **[Replace]** button performs the replacements for the (current) working found. Both the buttons are enabled in face view only.
The **[Replace all]** button performs the replacements in all occurrences matching the assigned criteria:
- in overall view: the selected section are affected by the replacement in the Replace area: <o,v> variables, <r> variables, variable geometries and face programs.
- In Face View: this command replaces the parametric form shown in the face workings.

The button **[Find all results]** is active in face view and it finds all the correspondences and the result of the search is shown in the window of Commands.
As already told, possible replacements also performed by Overall View, can be later cancelled by selecting Face View.

Solve

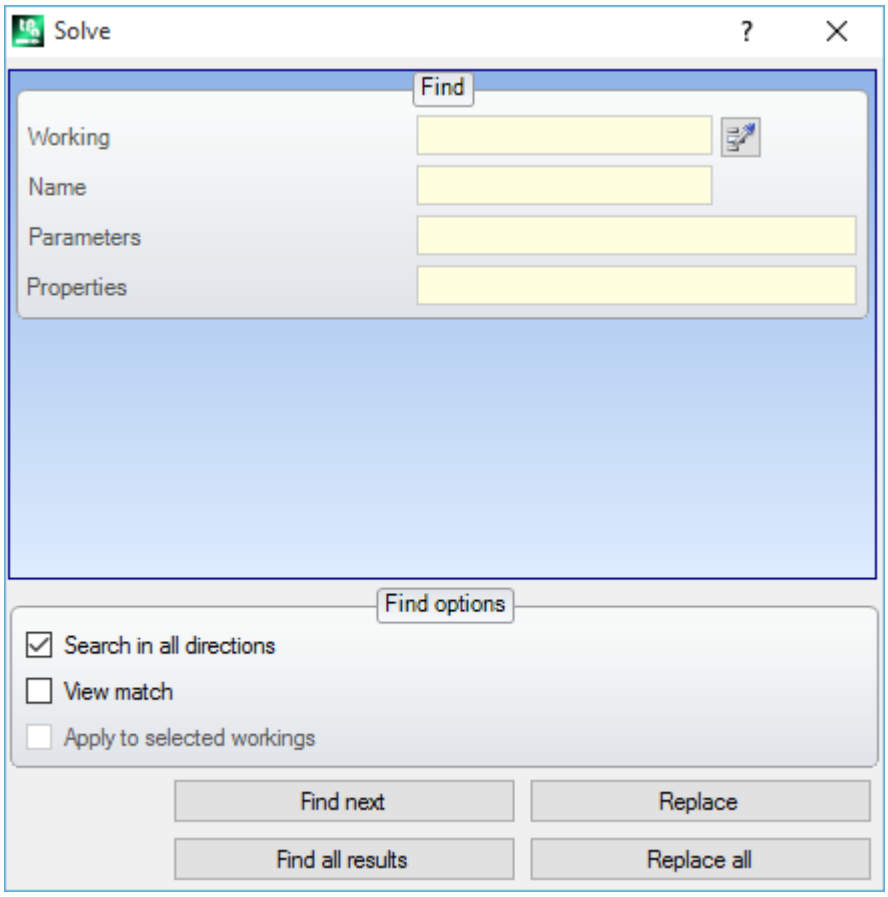
The  **Solve** command is enabled both in overall view and in face view and in this case with not empty face program.
The setting window for the data searching is invoked from the **Modify** group of the **Edit** tab.
This command finds each parametric form of numerical typology used in the working assignments and replaces it with the corresponding value calculated in accordance to the current status of the parametrizations (dimensions, variables). Parameter assignments of non-numerical typology, such as, for example, a writing or a subroutine name, if assigned in parametric format, remain unchanged.

In Overall View, the window shown is as follows:



Assigning the general correspondence criteria (see the command Find).
Confirming the window, the command is applied to all the programmed faces.

In face view, the window shown is more similar to the Find command.



In the Find area the data defining the search criteria (see Find command: Workings, Parameters, Properties).
The general correspondence criteria are set in the Find options area (see Find command).

The replacement of the parametric forms can be controlled by means of the buttons **[Find next]** and **[Replace]**.
The **[Replace all]** button performs the replacements in all occurrences corresponding to the assigned criteria.

10 Tools

10.1 Introduction

By the term Tools we mean all those commands which are specifically dedicated to edit workings mainly by making modifications of geometric nature. They are associated with the group of tools that are also some commands that essentially change the technology of the workings.
The windows that are opened show the settings as assigned to the previous recall of the tool.
The tools are applied to the workings that verify the active view filters: selections, logical conditions, layers, special filters. If the tool is applied directly to original workings, (selected or current) the modification cannot be applied to workings in a locked status (layer, construct, locked O field).
For the general tools only, in the setting window, the option **Apply to a copy of the workings** is automatically applied according to the settings of the status bar.

If the tools generate new profiles, these are opened with:
- a copy of the original setup, if available;
- with a copy of the reference setup (as assigned **Customize -> Technology -> Default codes** of the Application menu) otherwise.

In Piece-Face, most of Tools can be disabled if the Box view is active and if the current working is assigned on a non-real face.

The application of a tool can substantially alter the structure of the workings that are changed in their direct assignment (codes of workings, assignment of the parameters in parametric form) and/or geometric resolution. Where possible, the original structure of the work processes is maintained, with particular regard to the assignment in parametric format, but this is not always guaranteed.

10.2 General tools

Centring and alignment

A group of tools moves selected or current workings with centring or alignment to the face. These commands are present in the **General** group of the **Tools** tab and differ from the remaining commands of the group as they are directly applied, without further assignment, except when the option *Apply to a copy of the workings* is selected in the status bar of TpaCAD: in this case a direct confirmation is required.



Centre in X on the face: this option moves the working(s) by centring along the X axis of the face. Position along the Y axis of the face does not change.



Centre in Y on the face: this option moves the working(s) by centring along the Y axis of the face. Position along the X axis of the face does not change.



Centre in X+Y on the face: this option moves the workings by centring on the face (it couples both the previous commands).



Align to X=0 of the face: this option moves the working(s) by aligning the minimum overall dimension to the X=0 position of the face. Position along the Y axis of the face does not change.



Align to X=lf of the face: this option moves the working(s) by aligning the maximum overall dimensions to the X=lf position of the face. Position along the Y axis of the face does not change.

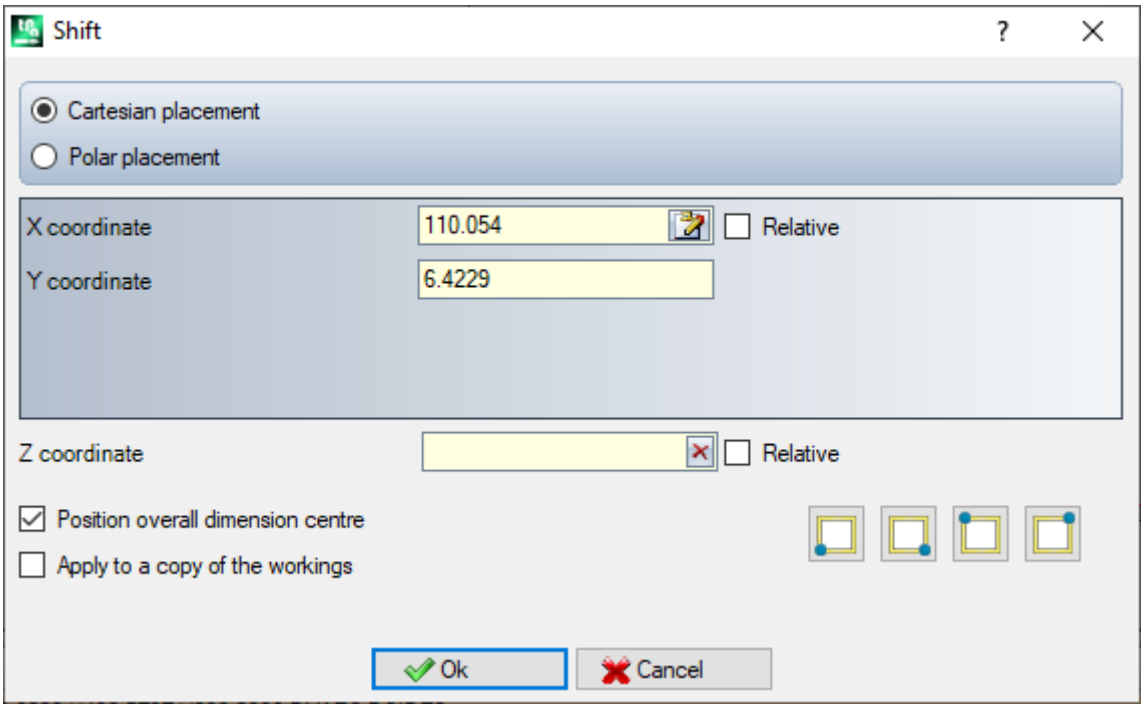


Align to Y=0 of the face: this option moves the working(s) by aligning the minimum overall dimensions to the Y=0 position of the face. Positioning along the X axis of the face does not change.

Align to Y=hf of the face: this option moves the working(s) by aligning the minimum overall dimensions to the Y=hf position of the face. Positioning along the X axis of the face does not change.

Translation

This tool translates the selected or the current workings to the assigned position. The translation of a working which belongs to a profile involves always the translation of the entire profile. The command **Shift** is available in the group **General** of the **Tools** tab.



The window displays all modes and options that can be activated. First of all, the choice of the positioning system on the xy plane of face, for the programming of the placement point:
- **cartesian**: assigns the x and y coordinates of the placement point in absolute or relative mode; if
- **polar**: assigns the x and y coordinates of the centre of the polar system (in absolute or relative mode), of the module and of the angle.

In case of absolute positioning, to the indicated position:
- the current working

If the option **Position overall dimension centre** is enabled, the centre of the overall rectangle for the translated workings is translated to the indicated position.
Alternatively, you can translate one corner of the overall rectangle by selecting one of the following four buttons:

shifts the corner to minimum positions in X and Y;

shifts the corner to maximum position in X and minimum in Y;

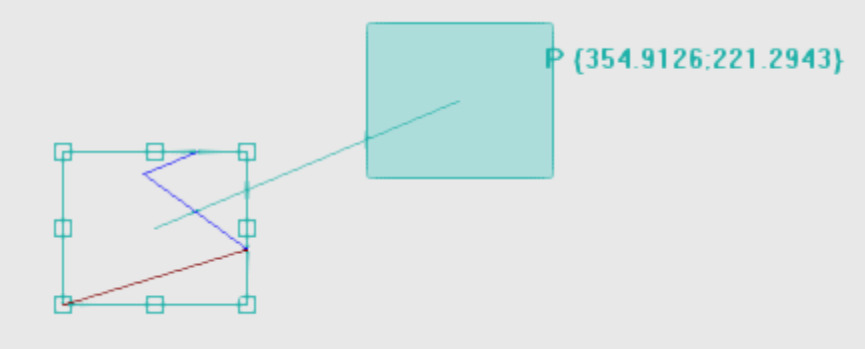
shifts the corner to maximum positions in X and Y;

shifts the corner to minimum position in X and maximum in Y.

Positioning coordinates can be entered as follows:
- in the edit fields the positioning can be expressed in absolute or relative mode, with numerical or parametric values set;

- by the mouse in the graphic area by clicking the icon (only if the program view is active). In this case the X and Y positioning coordinates are automatically set as absolute. In the case of selection of polar Positioning, the interactive acquisition may relate to the position of the centre, just as the module and angle values. The messages provided in the Command area lead through the interactive mode;

As for the interactive assignment mode of the position reference is made to **Insertion of Geometric Entities from Drawing Menu**:



- The overall rectangle is drawn matching the original dimensions of working, to which apply the translation together with the indication of the reference point for the translation (centre of the overall dimension, rather than the current working);
- The movement of the mouse corresponds to a graphic update of the positioning of the overall rectangle and of the reference point for the translation, matching the current mouse position.

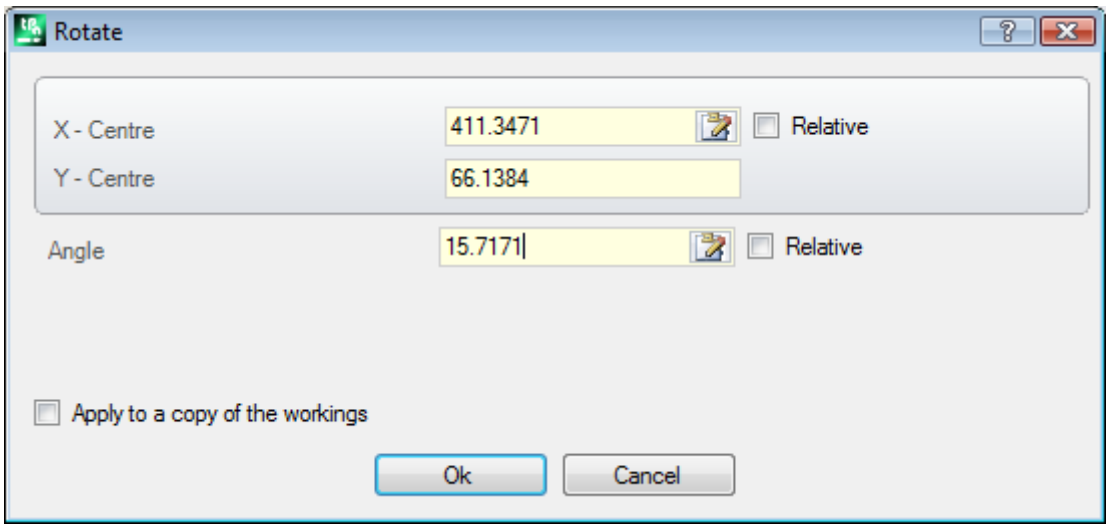
Automatic Snap
For the middle and vertex points on the sides of the overall rectangle an automatic snap is activated, that takes the mouse position within the panes that mark off the points. Anyway, you can deactivate this automatic snap by deleting the selection on the corresponding entry of the local menu. Disabling the automatic snap can allow for example a precise positioning to be made, for example, by selecting the direction keys.

Quitting the interactive procedure leads again to the window of direct assignment, in which you can integrate with the assignments and selections required.

The selection of **Apply to a copy of the workings** applies the tool to a copy of the workings and does not change the original lines.
A positioning in polar mode or to an absolute position causes the loss of each form of parametric programming.

Rotation

This option rotates the selected or current workings. The command **Rotate** is available in the group **General tools** of the **Tools** tab.



The rotation data can be entered as follows:

- in the edit fields the positioning can be expressed in absolute or relative mode, with numerical or parametric values set;
- by the mouse in the graphic area by clicking the icon  . In this case the X and Y positioning coordinates of the rotation Centre are automatically set as absolute. The interactive acquisition can also apply to the angle. The messages provided in the Command area lead through the interactive mode;
- by selecting a button from the command bar in the window:
  - the first 5 buttons assign the position of the centre on one of the remarkable points of the overall rectangle for the workings concerning the rotation (centre or corner). The position of the points is recorded on the tooltip message displayed for each button;
  - the right button assigns the centre and the rotation angle to the values in order to minimize the overall rectangle.

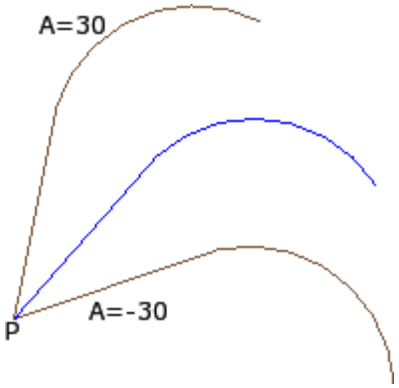
In case of positioning concerning the rotation Centre, the centre itself is place in relative with respect to:
- the current working

If activated from TpaCAD configuration, the application of the tool to an oriented setup can apply the transform to the orientation axes (only if the current face is plane, i.e. it is not curved or is assigned as a surface).

The rotation of a working which belongs to a profile involves
- The rotation of the entire profile, if the rotation is applied to the workings copied in the local Clipboard or to the selected workings, or if the **Apply from current working until the end of the profile** option is not selected;
- otherwise, the rotation of the profile portion from the current working and the profile end: the centre of the rotation now coincides with the start point of the current working. If the selection of the **Apply to a copy of the workings** option is active, a copy of the entire profile is anyway inserted.

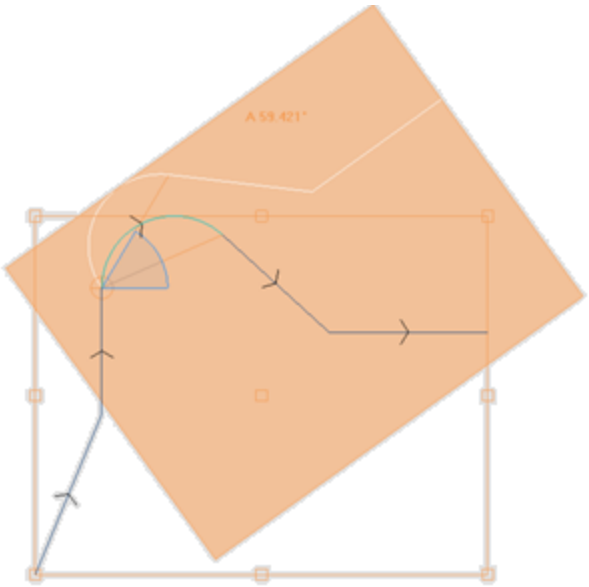
The application of the tool causes the loss of each form of parametric programming previously available for positioning on the face plane.

Let us see a particular example of rotation:



current working in **P**;
centre in relative to (0;0) coordinates. So, the centre is positioned in **P**;
rotation angle in relative positioning, of value:
- 30° (for upward rotation)
- -30° (for downward rotation).

In the picture, an example of rotation applied to a part of the profile, with interactive acquisition active:



The rotation tool cannot be applied to all workings. For example, all complex workings which verify one or both the following conditions are excluded:

- they recall a subroutine or a macro to which no rotation can be applied as established in the workings database.
- they are configured in the workings database as workings to which no rotation can be applied.

Typical examples are sawing works undertaken with tool that cannot be oriented.

Modify (menu of Graphics)

This command is available in the context-sensitive menu recalled in the graphic view area by pressing the right mouse button. This command is not available in the case of the *Essential* functionality.

This command allows to apply interactive simple transformations of *Translation* and *Rotation*.
The transformations are applied to the selected or to the current workings. If the working belongs to a profile, the transformations are always applied to the entire profile.

As already mentioned for the **Shift** tool, when the command is activated, the overall rectangle corresponds to the original overall dimension of the working, the centre, the points of the vertex and the middle points on the rectangle sides. On these points an automatic snap can be activated that takes the mouse position within the frames encircling the points.
The reference point for the translation and/or for the rotation is now the centre of the area.
The movement of the mouse corresponds to a graphic update of the positioning of the overall rectangle and of the reference point for the translation and/or rotation, at the current mouse position.

Switching between the two possible transforms occurs via contextual-sensitive menu, by selecting

'R'= Rotate switches to rotation tool;
'M'=Shift switches to shift tool.

To confirm a transform, click by the left mouse button.
[Enter] ends the command confirming the acquisitions made, **[Escape]**ends and cancels.

Symmetries

Symmetry tools mirror the selected workings with respect to a specified axis.
The commands of **Symmetries** are available in the group **General tools** of the **Tools** tab, all open the same window where it is possible to change the kind of symmetry required.

4 typologies of symmetries can be selected, as follows:

	Symmetry around a vertical axis
	Symmetry around a horizontal axis
	Horizontal+Vertical symmetry (around a point)
	Generic Symmetry

In the case of **Symmetry around a vertical axis** or **Symmetry around a horizontal axis** in the window one only value for the coordinate of the symmetry axis is displayed.
- A vertical axis is parallel to the face Y axis;
- A horizontal axis is parallel to the face X axis.

In the case of **Horizontal+Vertical axis**, in the window the X and Y coordinates of the symmetry point are displayed.
In the case of **Generic Symmetry**, in the window the X and Y coordinates of the two points are displayed. Furthermore, the selection causes the loss of each form of parametric programming previously available for positioning of the face plane.

If the tool is applied to a profile, also the tool compensation settings (right or left) and the selections of entry/exit segments to profiles are inverted, in case of right or left arc setting. If activated from TpaCAD configuration, the application of the tool to a profile can apply the mirror technology.
If activated from TpaCAD configuration, the application of the tool to an oriented setup can apply the transform to the orientation axes (only if the current face is plane i.e. is not curved or is assigned as a surface).

The symmetry of a working which belongs to a profile involves
- the symmetry of the entire profile, if the symmetry is applied to the workings in the local Clipboard or to the selected workings or if you select the **Symmetry around a generic axis** mode; or if the option **Apply from the current working until the end of the profile** is not selected;
- otherwise, the symmetry of the profile portion from the current working and the end of the profile: the symmetry axis is positioned at the start point of the current working. Even if the selection of the option **Apply to a copy of the workings** is active, a copy of the entire profile is anyway inserted.

The symmetry tool cannot be applied to all workings. For example, all complex workings which verify one or both the following conditions are excluded:
- they recall a subroutine or a macro to which the selected mirror cannot be applied as established in the workings database;
- they are configured in the workings database as workings to which the selected mirror cannot be applied.

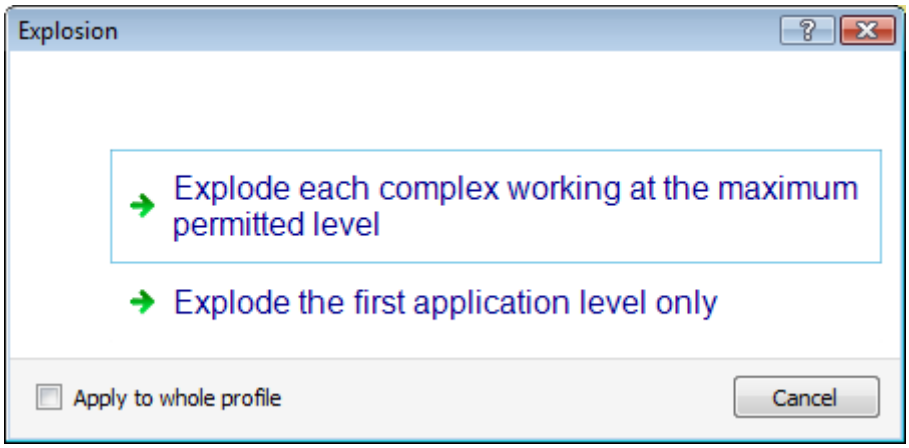
Typical examples are sawing works undertaken with tool that cannot be oriented.

Explosion

This option expands the complex workings or the profile multiple segments in the simple workings of which they are made.

The **Explosion**  is available in the **General tools** group of the **Tools** tab.
The tool is applied to the selected or current workings.
The Explosion tool cannot always be applied. The user must exclude the complex workings that:
- are configured in the workings database as workings to which the transform cannot be applied.
- in piece-face calculate codes of programmed induced call (SSIDE).

A selection window between two options can be displayed, when the command is selected and only if the window is enabled in TpaCAD configuration;



- By the first option the maximum possible explosion for the program lines involved is required;
- by the second option the minimum explosion of each program line involved in the command application is required.

If the selection window is not enabled in TpaCAD configuration, the first option is always applied.

If, for example, there is a ONE subroutine call, that on hi turn calls a multiple drilling working ("FITTING X"), to which no explosion limit is assigned:

1. in the first case the explosion replaces the subroutine call by the list of the individual drilling of the "FITTING X" working;
2. in the second case the explosion maintains the "FITTING X" working.

If for the "FITTING X" working an explosion limit is assigned, the "FITTING X" working cannot anyway be referred to a list of individual drillings.

The option **Apply to whole profile** can also be suggested in the window shown above or in a separated window: select in order to apply the tool to the current profile as a whole or to those profiles that match the selected workings.

When the **Explosion** command is applied, possible developments of induced calls get lost. In this case, a message informs about the situation.

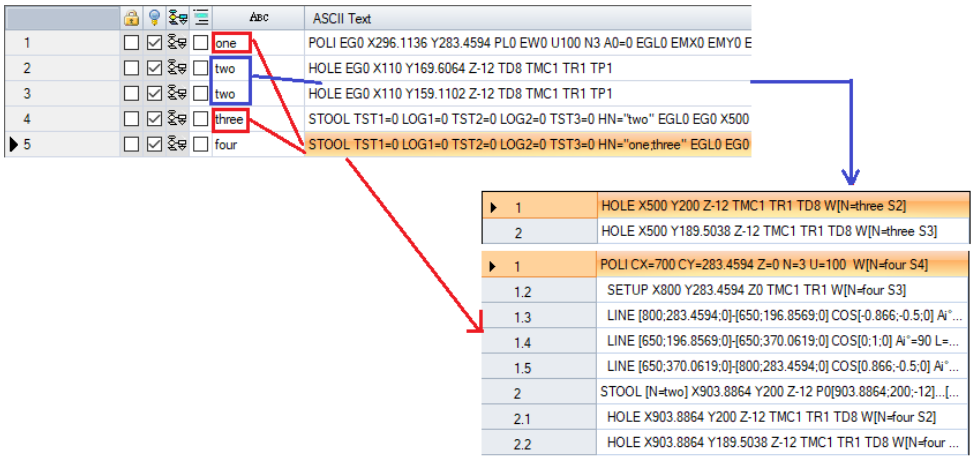
If the application of the Explosion command has led the concerned workings to simple workings only, the total geometric match of the program modified by the original version is guaranteed.
But if the command application has left unexploded macro or subroutine recalls, a message informs that the modified program may not totally match the original version. This may result from the fact that it is not always possible to apply the required transforms (for example: translation, rotation, mirroring, resizing, inversion, ...) to all branches of internal working development. Another critical case is the large number of Rotation and Symmetry transforms, as the application order of these transforms changes the final result.

If the application of the command does not contain at the level of programmed list any codes of programmed Tools, all the additional lines have the same *Name* assigned to the original program line.
Otherwise, the *Name* field of the additional lines can be assigned according to different rules, which have the purpose to preserve the original development of the program.

Advanced considerations

It is emphasized that we are now dealing with very particular problems, that can be found only with a specific TpaCAD configuration and that are considered advanced programming problems: the possibility to require a minimum explosion.

Let us examine now two particular cases.
In the figure an example of *Programmed tool applied recursively*:



The line 4 is a STOOL code that is applied to the workings called "two" (HOLE workings of line 2 and 3):
- alongside is the window that displays the list of the workings that match the STOOL code
- the line 4 is called "three".

The line 5 is a STOOL code that is applied to the workings called "one" (POLI workings of 1 line in the example is a macro developing a profile: SETUP + linear) and "one;three" (STOOL working of line 4):
- the window alongside displays the list of the workings that match the STOOL code;
- the line 5 is called "four".

We now see what happens expanding the line 5 at the only first application level:

		ASCII Text
1	one	POLI EG0 X296.1136 Y283.4594 PL0 EW0 U100 N3 A
2	two	HOLE EG0 X110 Y169.6064 Z-12 TD8 TMC1 TR1 TP
3	two	HOLE EG0 X110 Y159.1102 Z-12 TD8 TMC1 TR1 TP
4	three	STOOL TST1=0 LOG1=0 TST2=0 LOG2=0 TST3=0 HI
5	four	POLI EG0 X700 Y283.4594 PL0 EW0 U100 N3 A0=0 E
6	four	STOOL TST1=0 LOG1=0 TST2=0 LOG2=0 TST3=0 HI

Line 5 is exploded in 2 lines:
- [5] called "four": it results from the application of the POLI working of line 1
- [6] called "four": it results from the application of the STOOL working of line 4.

The names of the two lines correspond to the name of the original line.

Let us try to recall the program examined just now (that we call: PRG1) with SUB code, then let explode the call line to the only first application level:

		ASCII Text
1	xone	POLI EG0 X296.1136 Y283.4594 PL0 EW0 U100 N3 A0=0 EGL0 EMX0 EMY0 EIN
2	xtwo	HOLE X110 Y169.6064 Z-12 TD8 TMC1 TR1 TP1
3	xtwo	HOLE X110 Y159.1102 Z-12 TD8 TMC1 TR1 TP1
4	xthree	STOOL LOG1=0 TST2=0 LOG2=0 TST3=0 HN="xtwo" EGL0 EG0 X500 Y200 Z-12
5	xfour	STOOL TST1=0 LOG1=0 TST2=0 LOG2=0 TST3=0 HN="xone;xthree" EGL0 EG0 X

The working list corresponds to the original program test, to the names only a 'x' has been added. If the SUB code line had an assigned name (example: "yyy"), to the names it would have been added the name of the SUB (in the example: "yyyone" instead of "xone", ...). The names have been changed also in the assignment field of the STOOL codes (at the line 4: HN="xtwo"; to the line 5: HN="xone;xthree").
The new nomenclature generated for the additional workings tends to reduce as much as we can the possibility that:
- a listed STOOL code from a partial explosion can be applied downwards also to previously existing workings, assigned with the same name used in the subroutine (for example: "one")

- a previously existing STOOL code upwards can be also applied to workings inserted by the explosion and assigned with the same used name in the program itself (for example: "one");
- In this example it would be wrong to assign to the exploded lines the name of the original working (empty field or for example "yyy"), because the two inserted STOOL codes would no longer have found the workings of the application, originally assigned to an application level of the SUB code.

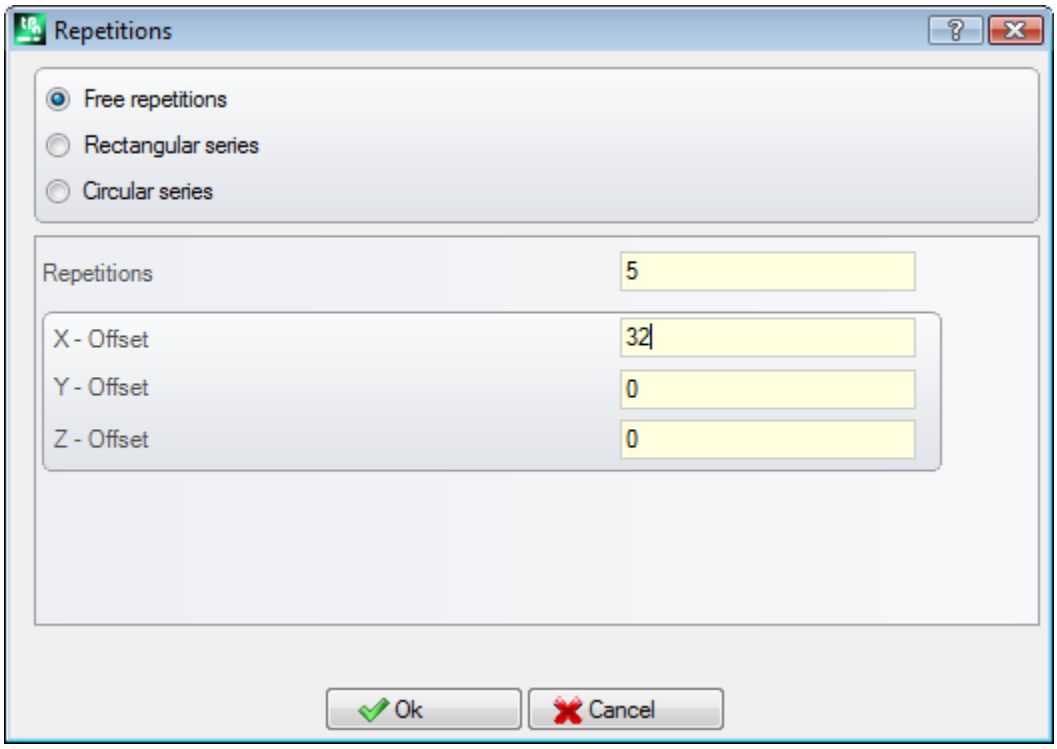
It is very clear that there is no absolute certainty that no opportunities are created for conflict between the names generated by a partial explosion and the original ones of the program list. However, it is possible to exploit the here described mechanism of changing names just to avoid these conflicts. The mechanism to change the names must obviously comply with the maximum length of a name (16 characters: the exceeding characters are deleted). For this reason we recommend the user not to use long names, so that no automatic truncations occur in the queued names.

Repetitions

Free repetitions

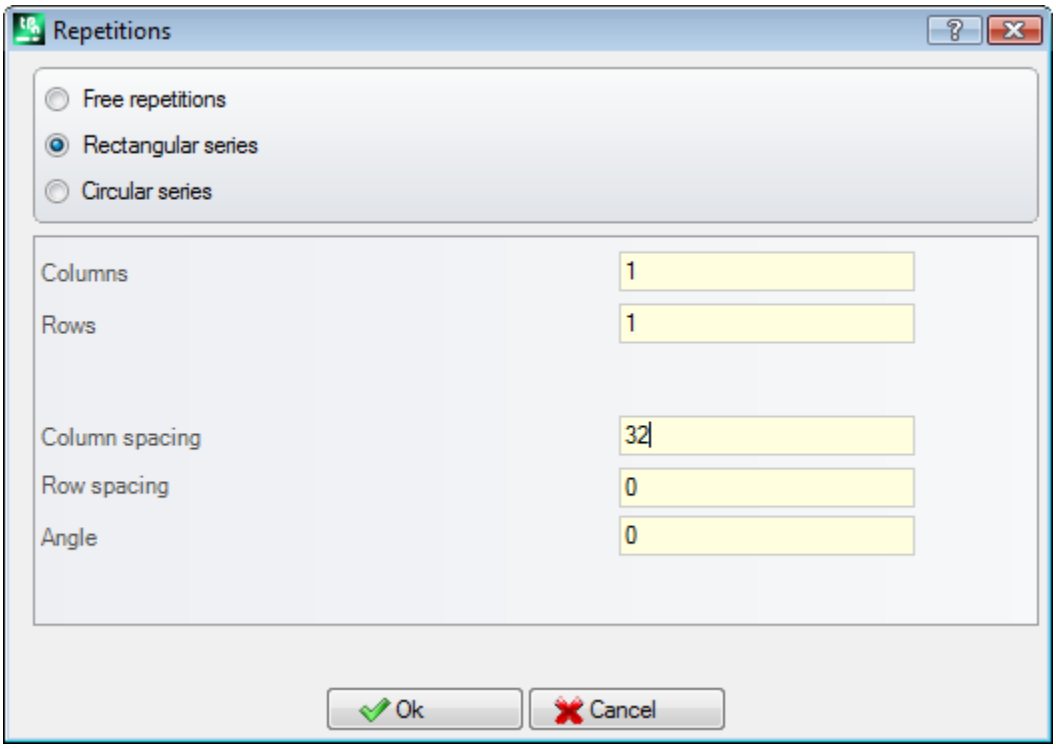
This option performs as many copies of the working(s) as specified in the **Repetitions** option and places them in a scheme, where for each coordinated axis a placement offset in set.
Each field of the window can set a numeric or a parametric value.
The repetition of a working which belongs to a profile involves:

- The repetition of the entire profile, if the tool is applied to the workings copied in the local Clipboard or to the selected workings, or if the **Apply from current working until the end of the profile** option is not selected;
- Otherwise: it performs the repetition of the profile portion between the current working and the end of the profile: the offsets of placement are now determined automatically, on the basis of the overall dimensions of the profile portion that concerns the tool.



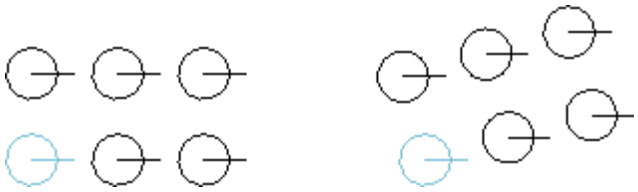
Rectangular series

This option performs a copy of the working(s) with positioning in accordance to a matrix scheme.
Each field of the window can set a numeric or a parametric value.



- **Columns**, **Rows**: both fields cannot have value 1 and the total number of the repetitions cannot exceed 100,000. Development on the rows is always associated with the Y axis of the face and that on the columns with the X axis of the face.
- **Column spacing** and **Row spacing**: they are significant values with sign.
- **Angle**: rotation angle (with respect to Face X axis, positive for CW rotation).

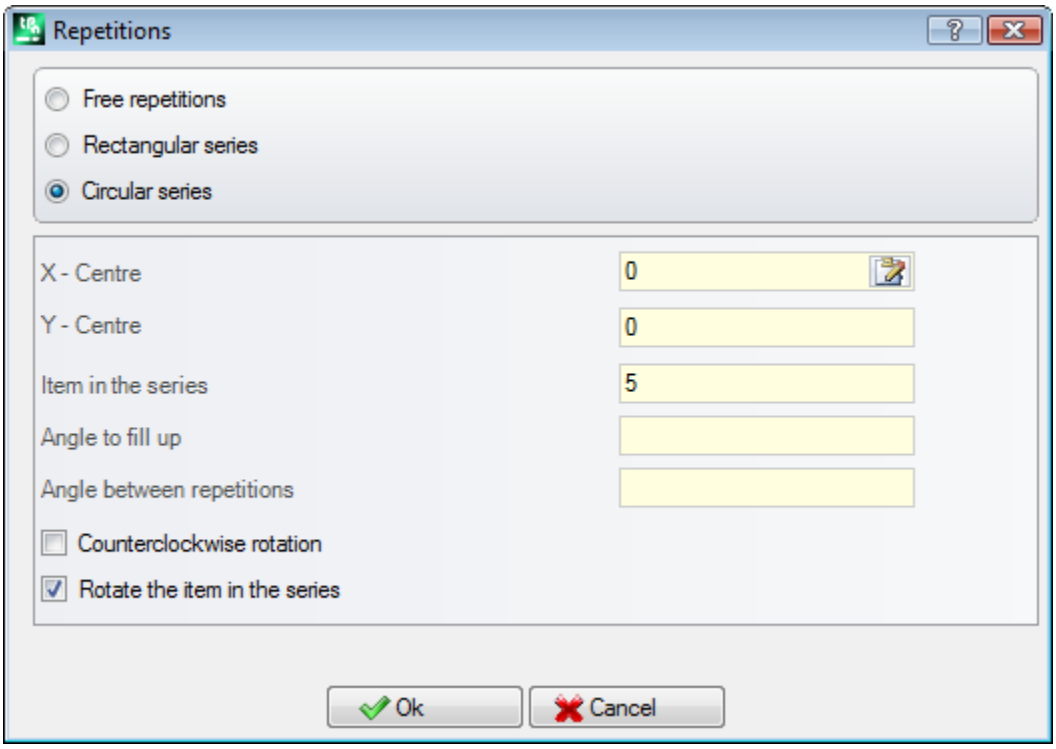
Example:



Repetition of the working marked with a bold line for **3** columns and **2** rows.
The left figure shows a series with **A**=0.
The right series show the same series with **A**#0.

Circular series

This option performs a copy of the working(s) a circular scheme.
Each field of the window can set a numeric or a parametric value.



- **X - Centre, Y - Centre**: centre of the arc along which the scheme is developed. Select the icon to acquire the position of the centre by the mouse in the graphic area.
- **Item in the series**: number of the elements of the series, original included. The value set, as well as the parametric one, must be greater than 1.
- **Angle to fill up**, **Angle between repetitions**: the values set, also the parametric ones, must be included between 0.001° and 360°.

Of the three last parameters two must be set, the third one is calculated automatically.

- **Item in the series**: number of the elements of the series, original included.
- **Angle to fill up**: angle that must be filled up by the repetitions (original elements included).
- **Angle between repetitions**: angle between consecutive repetitions.

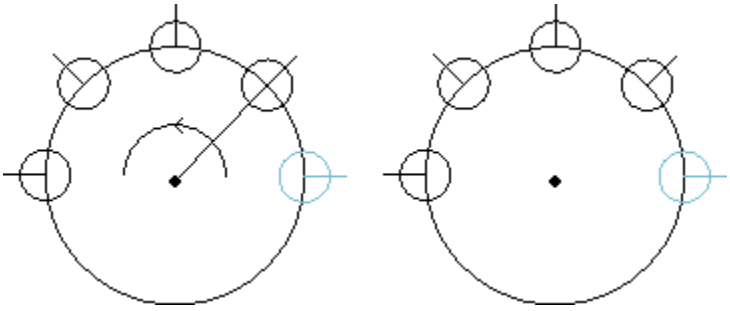
Let us see now what the priorities are to evaluate the settings:

- **Items in the series** and **Angle to fill up** set: the **Angle between repetitions** field is ignored and the angle between elements is calculated automatically.
- **Item in the series** is not set: both the field of the angles must be set. The number of the elements of the series is calculated automatically.
- **Angle to fill up** is not set. Both the other fields must be set.

- **Counterclockwise rotation**: select to require a development in the counterclockwise rotation of the repetitions.
- **Rotate the items in the series**: select to rotate the elements matching each single repetition, so that the development of the same is kept unchanged with respect to the rotation centre.

The application of the tool causes the loss of each form of parametric programming previously available for positioning of the face plane.

Example:



Repetition of the working marked with a bold line for **5** elements, angle to fill up: **180°**, counterclockwise rotation.
The left figure shows the series whose **Rotate item in the series** field is not selected.
The right figure shows the series whose **Rotate item in the series** field is selected.

Repetitions on a profile

This option makes a copy of the workings as many times as specified by the command **Repetitions** and distributes them along an already programmed profile.
The repetition of a working which belongs to a profile involves the repetition of the whole profile.
The original workings are not changed.

- **Repetitions**: number of element of the series.
- **Profile working**: progressive number of a profile programming where make the distribution (any segment)
  Select the icon to capture the profile interactively. The profile must be simple with arcs only in the xy plane and the tool is applied to selections, the distribution profile cannot itself be selected.

If the option **Position overall dimension centre** is enabled, the centre of the overall rectangle for the translated workings is translated to the position required.
Alternatively, a corner of the overall rectangle can be translated by selecting one of the four buttons, as follows:

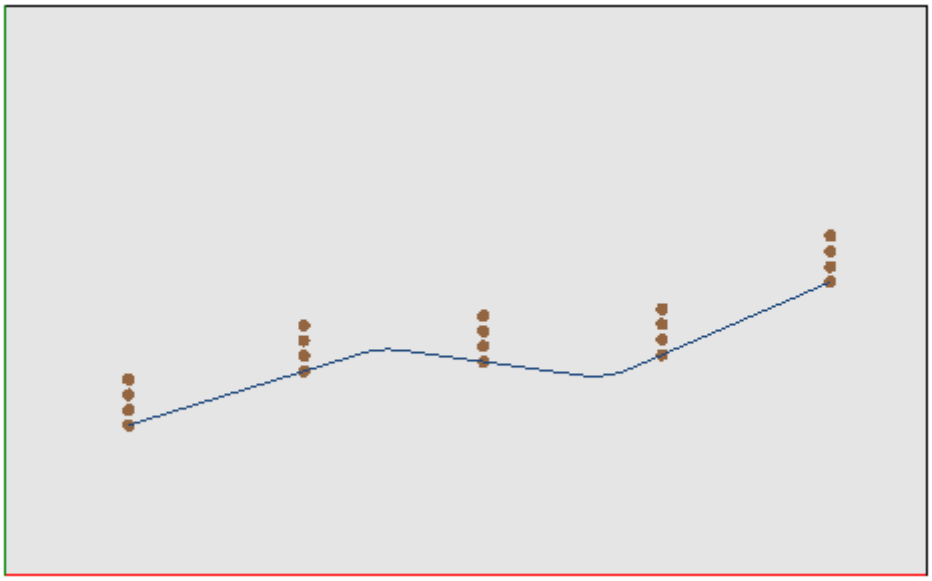
translates the corner to minimum positions in X and Y;

translates the corner to maximum position in X and minimum position in Y;

translates the corner to maximum positions in X and Y;

translates the corner to minimum in X and maximum position in Y.

The figure shows the repetition of 4 holes distributed vertically: the number of repetitions is 5.



10.3 Profile Tools

Change a profile segment

The **Change** is available in the group **Change profiles** of the **Tools** tab. It changes the current profile by changing the geometry lengthening or shortening the length by changing the ending point. The current segment must belong to a profile, to be simple and of arc or line typology. A linear portion may not have a null length.
In Piece-Face the tool is disabled if it is active the View-box, with current processing on a face not real.

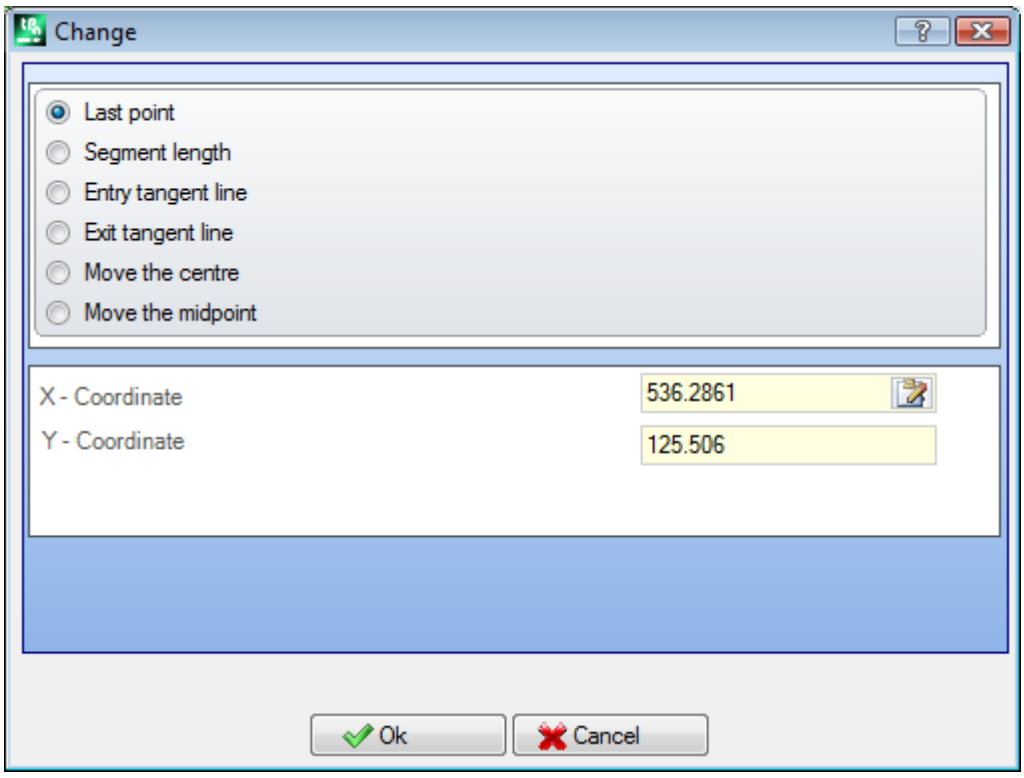
If the segment is **linear**, you can set:
- **Last point**: it moves the end point of the segment to the programmed coordinates in the X-Coordinates, Y-Coordinates fields (Click the icon to get the coordinates in interactive way). The displacement of the end point changes the direction of the segment. You can change also the coordinate of final depth of the segment (Z coordinate).
- **Segment length**: It defines the linear length of the stretch in the face plane.

- **Entry tangent line**: The section is modified by imposing the tangent with the previous section, or by assigning the value of the slope.

If the segment is a path segment (L24), you can set:
- **Last point**: it moves the end point of the segment to the programmed coordinates in the X-Coordinates, Y-Coordinates fields (Click the icon to get the coordinates in interactive way). The displacement of the end point maintains unchanged the directions of departure and arrival of the curve.
- **Segment length**: it defines the distance between the edge points of the curve in the face plane.
- **Entry tangent line**: The curve is modified by imposing the tangent with the previous section, or by assigning the value directly.
- **Exit tangent line**: The curve is changed by assigning the value of the tangent at the end point.

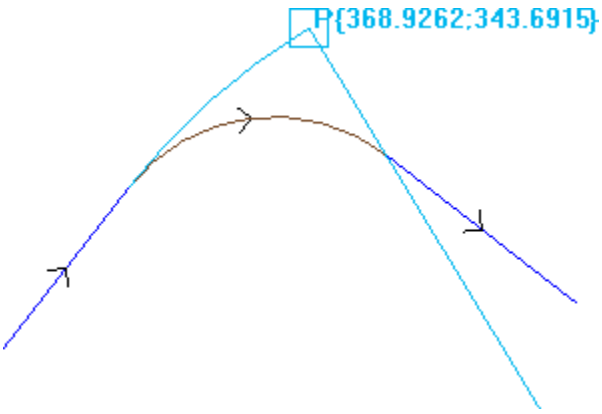
If the selected segment is an **arc** assigned in the face plane, it possible to set:
- **Last point**: it moves the end point of the segment to the programmed coordinates in the X-Coordinates, Y-Coordinates fields (Click the icon to get the coordinates in interactive way). The displacement of the end point maintains unchanged *Tangent in entry* of the arc and the point cannot coincide with the start point of the arc. You can change also the coordinate of final depth of the segment (Z coordinate).
- **Segment length**: it defines the length of the arc in the plan of allocation of the arc (the set value is limited to a maximum length of the circle) or the size of the angle in degrees (the value set is reduced to values between 0 and 360 °). The Size of the angle can also be determined in interactive mode.
- **Entry tangent line**: The section is modified by imposing the tangent with the previous section, or by assigning the value of the angle of departure of the arc.
- **Exit tangent line**: The section is modified by assigning the value of the tangent on the end point of the arc.
- **Move centre**: it moves the centre of the arc to the programmed coordinates in the X-Coordinates, Y-Coordinates fields (also in interactive mode).
- **Move midpoint**: it moves the midpoint of the arc to the programmed coordinates in the X-Coordinates, Y-Coordinates fields (also in interactive mode).

If the selected segment is an **circle** assigned in the face plane, it possible to set:
- **Segment length**: it defines the length of the arc in the plan of allocation of the arc (the set value is limited to a maximum length of the circle) or the size of the angle in degrees (the value set is reduced to values between 0 and 360°). The size of the angle can also be determined in interactive mode.

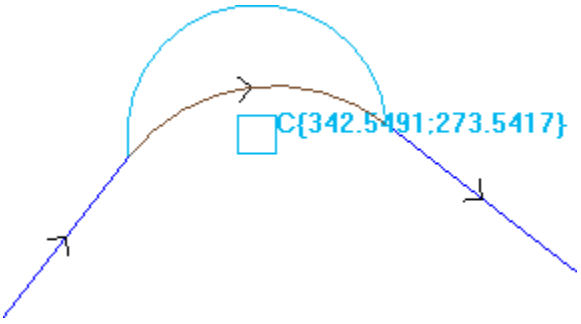


By moving the mouse in the graphic area, you can see how it changes the arc.
Hereunder three situations to modify the segment of arc typology:

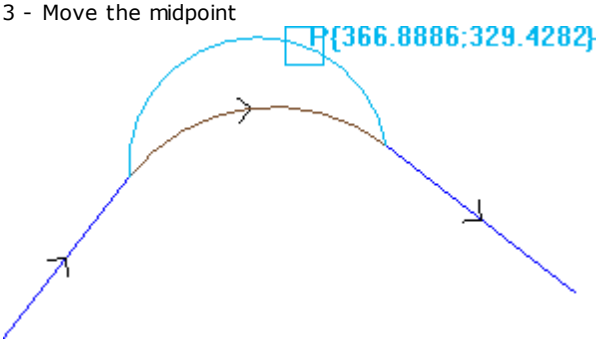
1 - Move the end point



2 - Move the centre



3 - Move the midpoint

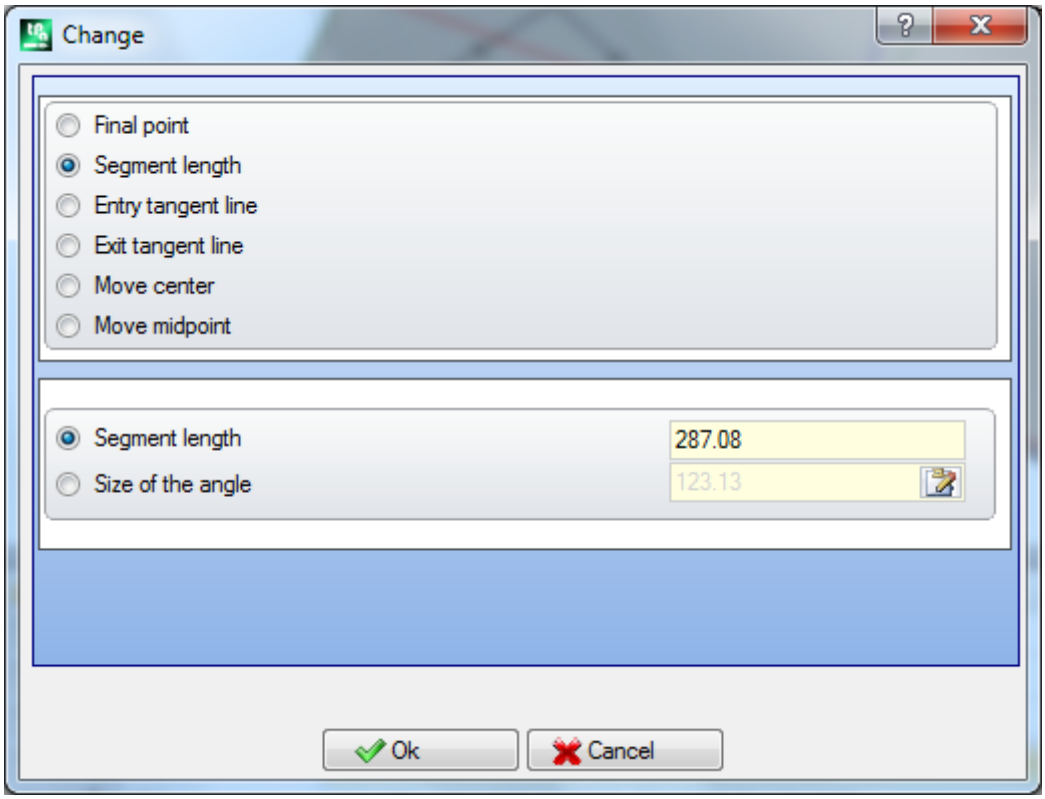


If the segment is a **conic arc (ellipse)** it is possible to change the length of the segment, expressed as linear or angular length:

- **Segment length**: Defines the linear length of the arc in the face plane. The value set is limited to a maximum length of the complete conic;
- **Size of angle**: in degrees (also in interactive way). The value set is reduced to values between 0 and 360.

If the selected segment is an **arc** or a **circle** in a different plane from the face plane, you can set:

- **Segment length**: Defines the linear length of the arc in the assignment face plane. The value set is limited to a maximum length of circle;
- **Size of angle**: in degrees, can be directly set. The value set is reduced to values between 0 and 360.



The change of the tract may cause changes to the operating code of the current working.
The tool does not work if the current working does not verify the active sight filters (selections, levels logical conditions, special filters) of if the profile is in a lock state (it is an induced call or his level, construct or O field are locked).
The tool is disabled if:
- there are no programmed workings
- the current working has no valid typology.

Change corner into arc

The **Change corner into arc**  is available in the group **Change profiles** of the **Tools** tab. It changes a corner into an arc.
In Piece-Face the tool is disabled, if it is active the Box-view, with current processing on a non real face.
The tool operates on extended profiles, but its application is possible only on a corner identified by two simple straight lines. The tool works directly on the current profile. In the window is shown the plane which the three points delimiting the corner lie on, as a plane on which to calculate the arc. If on the selected plane the geometrical conditions defining an arc do not exist, the transformation is not carried out.
The instrument is disabled, if:
- there are no programmed workings;
- the current working or the following one are not straight lines;
- the three vertices of the corner are not marked out or are aligned;
- in Piece-Face the tool is disabled, if the Box-view is active, with current working on a non real face.

Change the line in the path

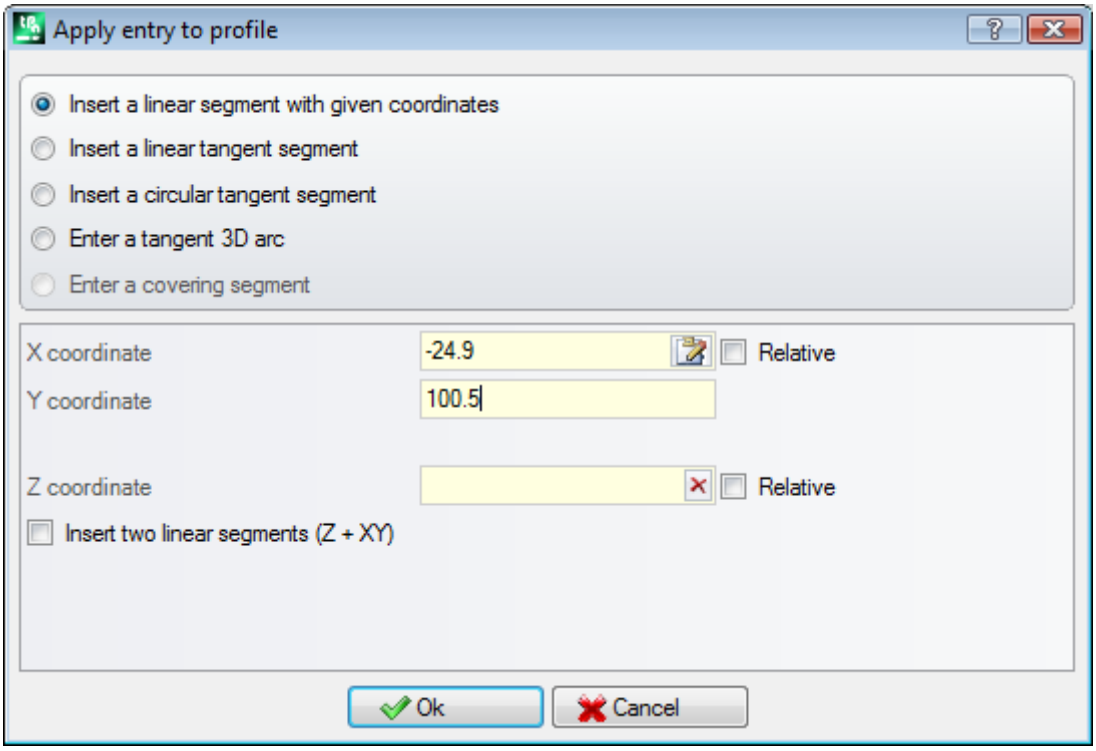


The **Change the line in the path**  is available in the group **Change profiles** of the **Tools** tab. This option changes the current profile in the working L24, that corresponds to an element called **Path**. The segment of the current profile must be linear.

This working is described in the paragraph **Workings -> Profile ->** [Path](#).

Apply entry to profile

The **Apply entry to profile**  is available in the group **Change profiles** of the **Tools** tab. In Piece-Face the tool is disabled, if it is active the Box-view, with current processing on a non real face. This tool adds a linear or circular entry segment to the current profile. If the current profile is an open profile a setup working is added as starting point of the added segment, otherwise the setup is moved to the new profile starting point.
In Piece-Face the tool is disabled, if it is active the Box-view, with current processing on a non real face.



Different insertion options can be chosen:

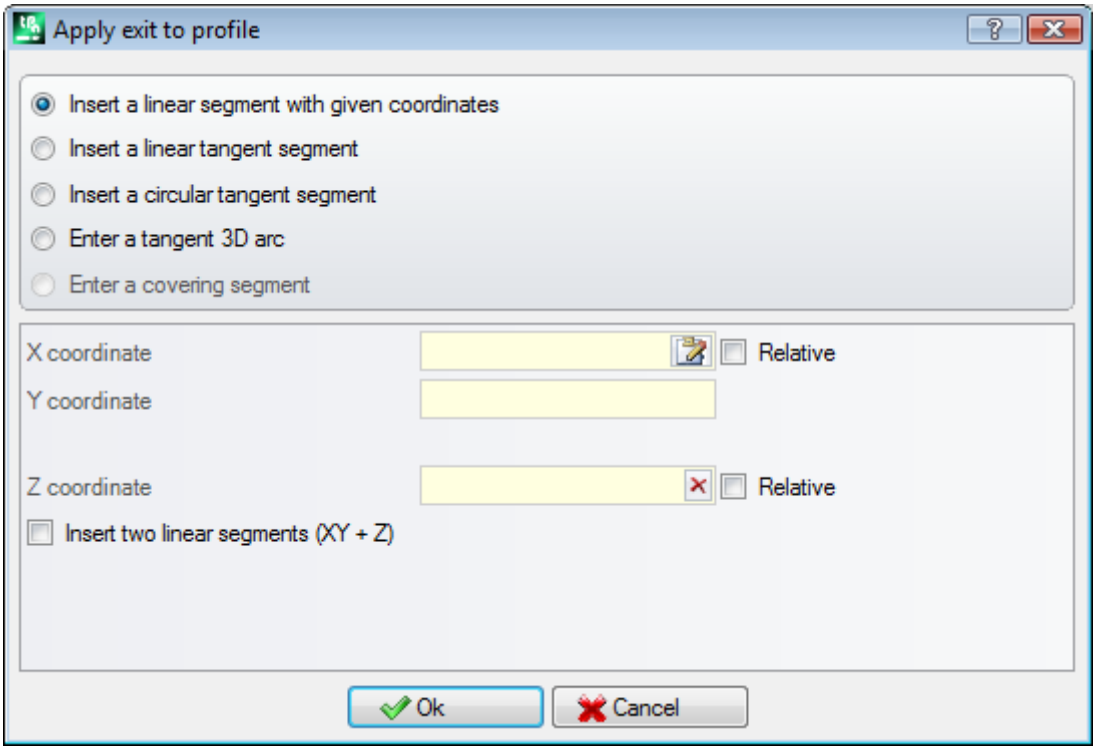
- **Insert a linear segment with given coordinates**: it inserts a linear segment starting from the point defined by the x and y coordinates (also in interactive mode) and ends on the start point of the original profile. If the relative mode is selected, the values are summed to the starting position of the original profile. If the entry **Insert two linear segments (Z+XY)** is selected, the programmed movement is divided into two linear segments:
  - the setup of the profile is place to the programmed position
  - the first linear segment carries out a movement in Z to the starting Z position of the original profile
  - the second linear segment carries out a movement to XY until it reaches the position of the original setup. The initial depth of the additional segment can be set in the **Z coordinate** field: If the relative mode is selected, the value and applied with respect to the initial depth of the original profile.
- **Insert a linear tangent segment**: it inserts a linear segment whose length is defined by **Module**, while the linear segment direction is assigned, so that the start direction of the original profile is maintained. When the **Apply in 3D** option is selected, this option set the tangent continuity in the space, when the profile is started up: the profile direction and the starting depth of the additional segment are automatically determined from the first segment of the original profile. If the option **Apply in 3D** is not selected, the initial depth of the additional segment can be set in the **Z coordinate field** (if it is selected in relative mode, the value is applied with respect to the initial depth of the original profile). If the entry **Insert two linear segments (Z + XY)** is selected, the programmed movement is divided into two linear segments, like in the previous case.
- **Insert a circular tangent segment**: it inserts an arc into the xy face plane, with direction of the assigned segment assigned in order to maintain the initial direction of the original profile. The X Coordinate and Y Coordinate parameters represent the absolute or relative coordinate of the start point of the arc (also in interactive mode). The initial depth of the segment is expressed by the Z coordinate parameter (if set in relative mode the value is applied with respect to the initial depth of the original profile).
- **Enter a tangent 3D arc**: it inserts a circular segment, that is defined by the **Radius** and by width of the angle (in degrees), while its direction is so assigned in the space, that the start direction of the original profile is maintained. This value of the angle width must be between 1.0° and 90°. If an arc cannot be determined, a

linear segment whose length is equal to the **Radius** is defined with continuous tangent at the profile entry. The segment solution is analogue to what applied into setup workings as far as the segment of the profile entry is concerned.

- **Enter a covering segment**: it inserts a portion of a set length that duplicates the geometry of the last and /or of the first segment of the original profile. It is possible to select a coverage line only if the original profile is a closed profile and if it closes with a profile line. Parameters to be defined are:
  - **Apply coverage at the beginning**: if enabled, it requires the insertion at the beginning of the profile a (total or partial) segment, covering the last profile segment. Set the fields:
    - **Module**: length of the additional segment. The field is initialized according the length value of the last profile segment. If a value null or higher than the initialized value is set, a total coverage will be obtained.
    - **Z - Coordinate**: it sets the initial depth of the segment. If the coordinate is in relative mode, it is applied with respect to the initial depth of the original profile. The depth coordinate is ignored, when the covering segment is an arc, developing on a different plane from xy.
  - **Apply coverage at the end**: if enabled, it requires the insertion at the end of a profile of a covering profile (total or partial) segment, of the first segment of the original profile. This option may not be selectable, if the geometry of the profile does not allow the insertion at the end of the covering segment. Set the fields:
    - **Module**: length of the additional segment. The field is initialized according to the value of the fist profile segment. If a value null or higher than the initialized value is set, a total coverage will be obtained.
    - **Z - Coordinate**: final depth of the segment. If the coordinate is in relative mode, it is applied with respect to the final depth of the original profile. The depth coordinate is ignored, when the covering segment is an arc, developing on a different plane from xy.

Apply exit to profile

The **Apply exit to profile**  is available in the group **Change profiles** of the **Tools** tab. This tool applies a linear or circular exit segment to the current profile. In Piece-Face the tool is disabled, if it is active the Box-view, with current processing on a non real face.



Different insertion options can be chosen:

- **Insert a linear segment with given coordinates**: it inserts a linear segment from the profile end point to the point defined by the x and y programmed coordinates (also in interactive mode). If the relative mode is selected, the values are summed to the final position of the original profile. If the entry **Insert two linear segments (XY + Z)** is selected, the programmed movement is divided into two linear segments:
  - the end point of the profile is placed to the programmed position;
  - the first linear segment carries out a movement to XY starting from the final position of the original profile until it reaches the programmed position;
  - the second linear segment carries out a movement to Z to the programmed final Z position.

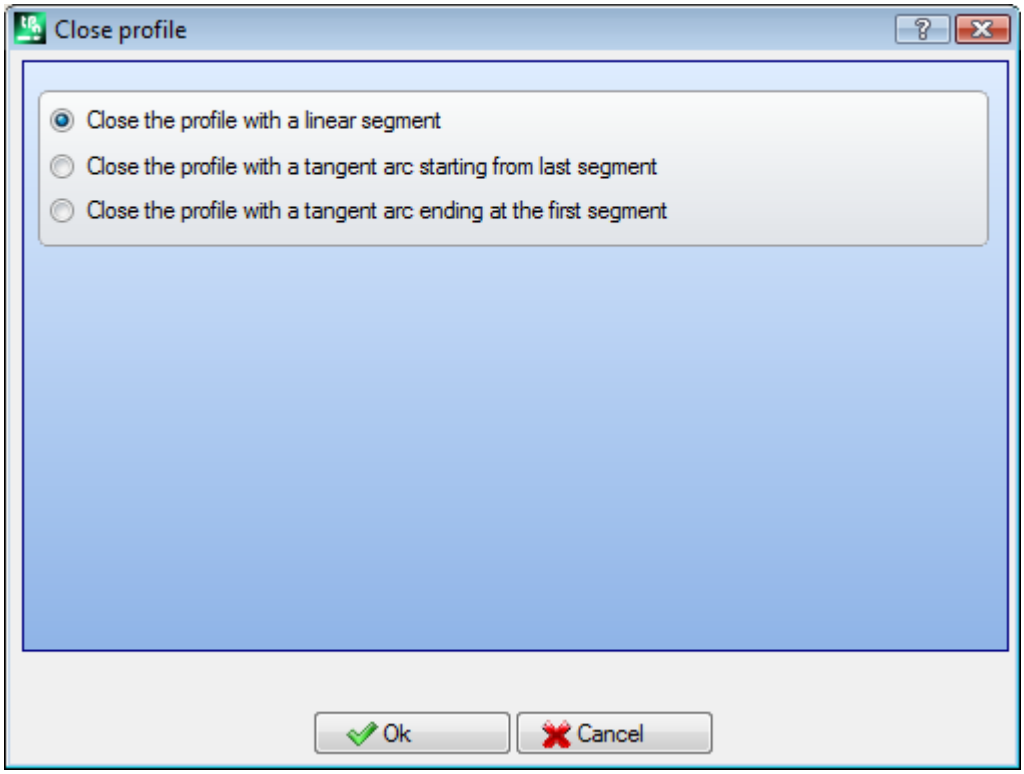
  The final depth of the additional segment can be set in the **Z coordinate** field: if the relative mode is selected, the value is applied with respect to the initial depth of the original profile.

- **Insert a linear tangent segment**: it inserts a linear segment whose length is defined by **Module**, while the linear segment direction is assigned, so that the closing direction of the original profile is maintained. When the **Apply in 3D** option is selected, this option sets the tangent continuity in the space, when the profile is being closed: the profile direction and the starting depth of the additional segment are automatically determined from the last segment of the original profile. If the option **Apply in 3D** is not selected, the final depth of the additional segment can be set in the **Z coordinate field** (if it is selected in relative mode, the value is applied with respect to the final depth of the original profile). If the entry **Insert two linear segments (XY + Z)** is selected, the programmed movement is divided in two linear segments, like in the previous case.
- **Insert a circular tangent segment**: it inserts a circular segment into the xy face plane, with direction of the assigned segment assigned in order to maintain the closing direction of the original profile. The X Coordinate and Y Coordinate parameters represent the absolute or relative coordinate of the end point of the arc (also in interactive mode). The final depth of the segment is given by the Z coordinate parameter (if set in relative mode the value is applied with respect to the final depth of the original profile).
- **Enter a tangent 3D arc**: it inserts a circular segment, that is defined by the **Radius** and by the width of the size of the angle (in degrees) while its direction is so assigned in the space, that the closing direction of the original profile is maintained. This value of the angle width must be between 1.0° and 90°. If an arc cannot be determined, a linear segment whose length is equal to the **Radius** is defined with continuous tangent while closing the original profile. The segment solution is analogue to what applied into setup workings as far as the segment of the profile exit is concerned.
- **Enter a covering segment**: it inserts a portion of a set length that duplicates the geometry of the first segment of the original profile. It is possible to select a coverage line only if the original profile is a closed profile and if it closes with a profile line. Parameters to be defined are:
  - **Module:** length of the additional segment. The field is initialized according to the value of the fist profile segment. If a value null or higher than the initialized value is set, a total coverage will be obtained.
  - **Z - Coordinate**: final depth of the segment. If set in relative mode, the value is applied at the starting depth of the profile. The depth coordinate is not considered, when the covering segment is an arc, developing on a different plane from xy.

Close profile

This tool applies a linear or circular closing segment to the current profile. The original profile cannot be closed.

The **Close profile**  command is available in the group **Change profiles** of the **Tools** tab.

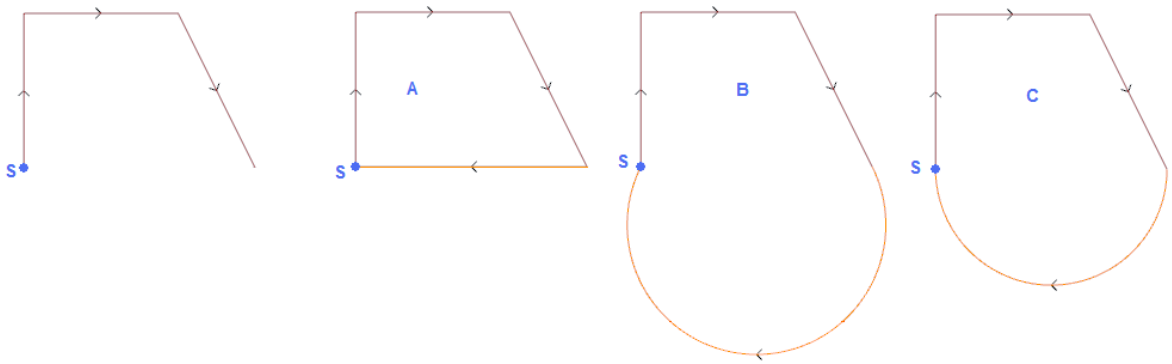


The kind of segment to be inserted can be selected from three options:

- **Close profile with a linear segment**: closes the profile by inserting a linear segment which connects the last profile point to the setup point.
- **Close profile with a tangent arc starting from last segment**: closes the profile by inserting an arc in tangent continuity with the last original segment of profile.

- **Close profile with a tangent arc ending at the first segment**: closes the profile by inserting an arc in tangent continuity with the first segment of the original profile.

Let us see now en example of closure of a simple profile, with three possible options:
- On the left we can see the original profile, not closed: 'S' is the start point (setup point), with counter-clockwise direction;
- figure A: the profile is closed with a linear segment;
- figure B: the profile is closed with an arc tangential to the last segment of the original profile;
- figure C: the profile is closed with an arc tangential to the first segment of the original profile.



Invert profile

This tool reverses the direction of selected or current profiles. The **Invert profile**  commands available in the group **Change profiles** of the **Tools** tab.
The tool is applied to:
- all profiles which have at least a selected element
- the current profile.

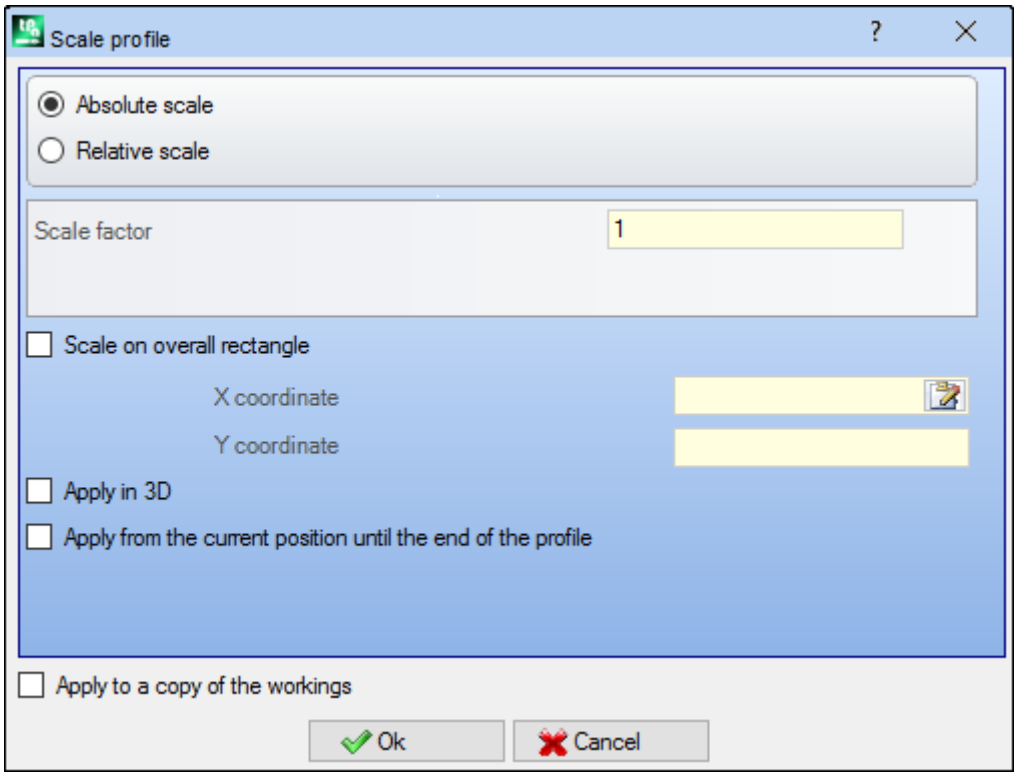
While executing the command, changes can be directly applied to a copy or to the original profiles.
The tool also inverts the settings of
- tool compensation (right or left);
- selection of entry/exit segments, in case of right or left arc settings.

If activated from TpaCAD configuration, the application of the tool to a profile can apply the mirror technology.
If activated from TpaCAD configuration, the application of the tool to an oriented setup can apply the transform to the orientation axes (only if the current face is plane, i.e is not curved or is assigned as a surface).

The tool resets tool offset changes in the path (interruptions, suspensions and shooting and/or side changes).

Scale profile

This tool applies a scale factor to one or more profiles. The **Scale profile**  command is available in the group **Change profiles** of the **Tools** tab.
It is applied to:
- all profiles which have at least a selected element
- the current profile



Select the scale assignment mode:
- **Absolute scale**: assigns directly the **scale factor**. A value greater the 1.0 increase the profile, a value between 0.01 and 1.0 (strictly less) decrease the profile.
- **Relative scale**: the factor scale is determined by setting two values:
  - **reference length**: current reference value;
  - **new length**: value changed after the scale.

For example: setting for the two values respectively 5.0 and 10.0, a segment 5.0 long is increased by the application of a scale factor equal to 2.0, obtained as the ratio of 10.0 to 5.0.

- **Scale on overall rectangle**: select to assign the base point to the centre of the overall rectangle of the profile concerned by the transform. If the option is not selected, we can set the base point in:
- **X - Coordinate, Y - Coordinate**: the coordinates are programmed in the X Coordinate, Y Coordinate fields (also in interactive mode, by clicking the icon).

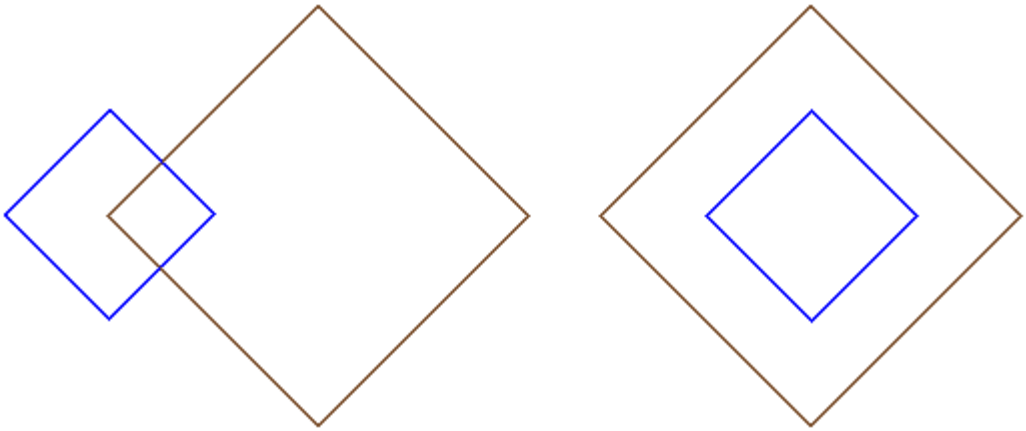
- **Apply in 3D**: enable to apply the scale also in face depth (otherwise in the xy plane only). The selection is obligatory if the concerned profiles execute arcs in #xy planes.

For the field **Scale factor** it is possible to activate the interactive mode (by clicking the icon): the graphic representation shows the overall rectangle of the profiles concerning the geometric transform, scaled with respect to the base point. The scale factor changes by means of the mouse wheel or by selecting the addition (+) or subtraction (-) keys. In interactive mode it is possible to modify the scale factor within the highest and the lowest values (0.2; 2.0);

The application to profiles requires:
- The change of the entire profile, if the tool is applied to the selected workings, or if the option **Apply from current position until the end of the profile** option is not selected;
- otherwise: the modification of the profile part between the current working and the end of the profile: the base point now coincides with the start point of the current working. If the selection of the option **Apply to a copy of the workings** is active, a copy of the entire profile is anyway inserted.

The selection of *Apply to a copy of workings* applies the tool to a copy of the workings and does not change the original lines.

Let's see an instance:

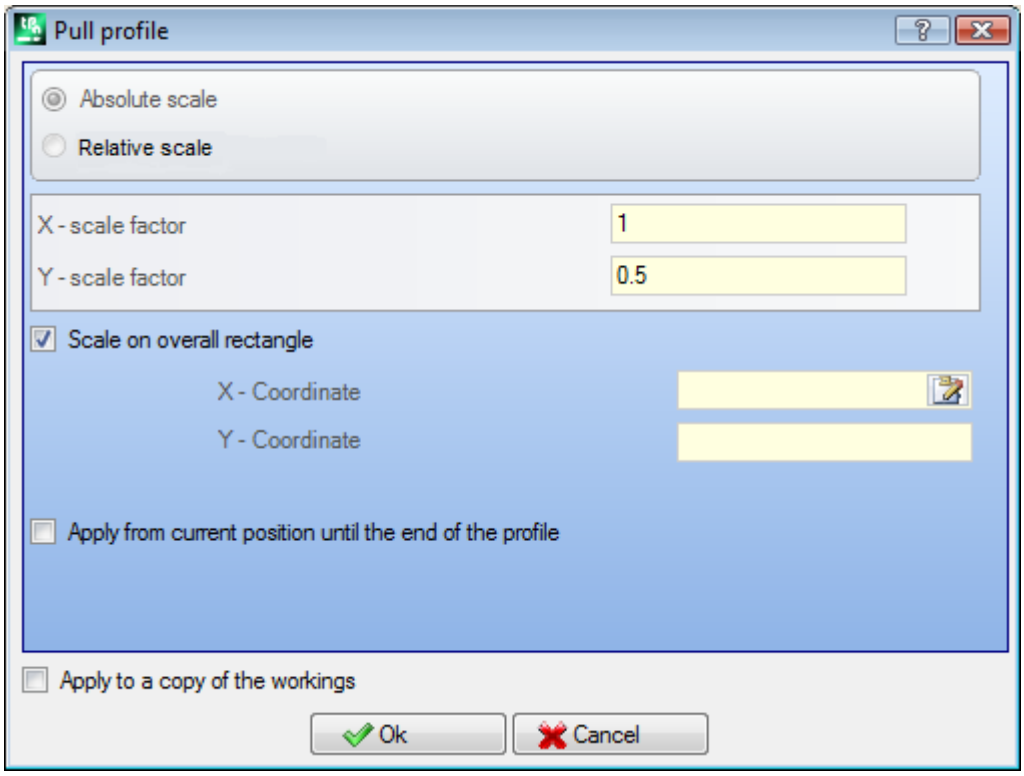


Required scale factor: **0.5**
Change the assignment of the **Base Point**:
- Left figure: (X, Y...) point shown on the left of the profiles;
- Right figure: centred on the overall rectangle.

The tool execution halves the dimension of each segment of profile and each profile halves its own distance from the specified base point. If also an entry and/or exit trait is assigned to the profiles, the scale factor is also applied to themselves.

Pull profile

This option applies a factor scale to one or more profiles, where the scale applied in x and in y can differ. The **Pull profile** command is available in the group **Change profiles** of the **Tools** tab.



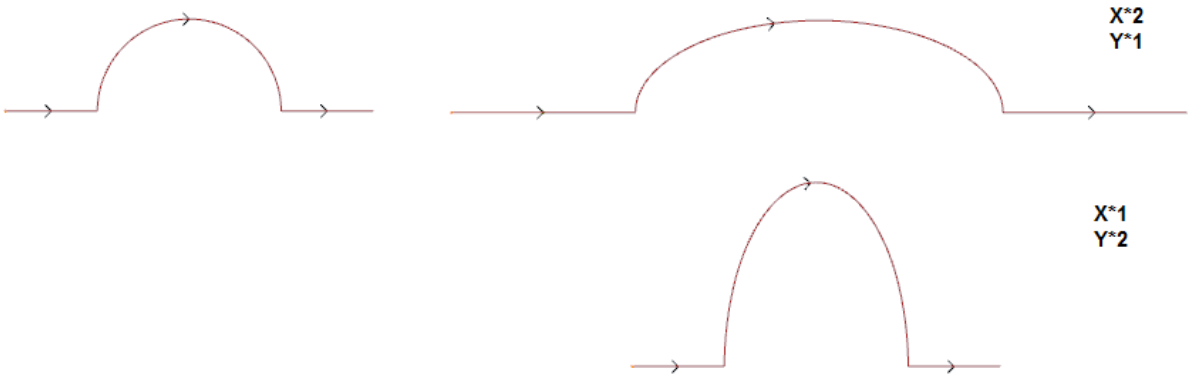
- **X-scale factor, Y-scale factor**: the scale factor is set with absolute values, by X and Y-direction.
- **Scale on overall rectangle**: select to assign automatically the base point to the centre of the overall rectangle of the profile concerned by the transform. If the option is not selected, we can set the base point in:
- **X-Coordinate, Y-Coordinate**: the coordinates are programmed in the X-Coordinate, Y-Coordinate fields (also in interactive mode, by clicking the icon).

For the fields **X/Y-scale factor** it is possible to activate the interactive mode (by clicking the icon)): the graphic representation shows the overall rectangle of the profiles concerning the transform, scaled with respect to the base point and the scale set. The individual scale factor changes by means of the mouse wheel or by selecting the addition (+) or subtraction (-) keys. In interactive mode it is possible to modify the scale factor within the highest and the lowest values (0.2; 2.0);

The application to profiles requires:
- The change of the entire profile, if the tool is applied to the selected workings, or if the option **Apply from current position until the end of the profile** option is not selected;
- otherwise: the modification of the profile part between the current working and the end of the profile: the base point now coincides with the start point of the current working. Even if the selection of the option **Apply to a copy of the workings** is active, a copy of the entire profile is anyway inserted.

The selection of *Apply to a copy of the workings* applies the tool to a copy of the workings and does not change the original lines.

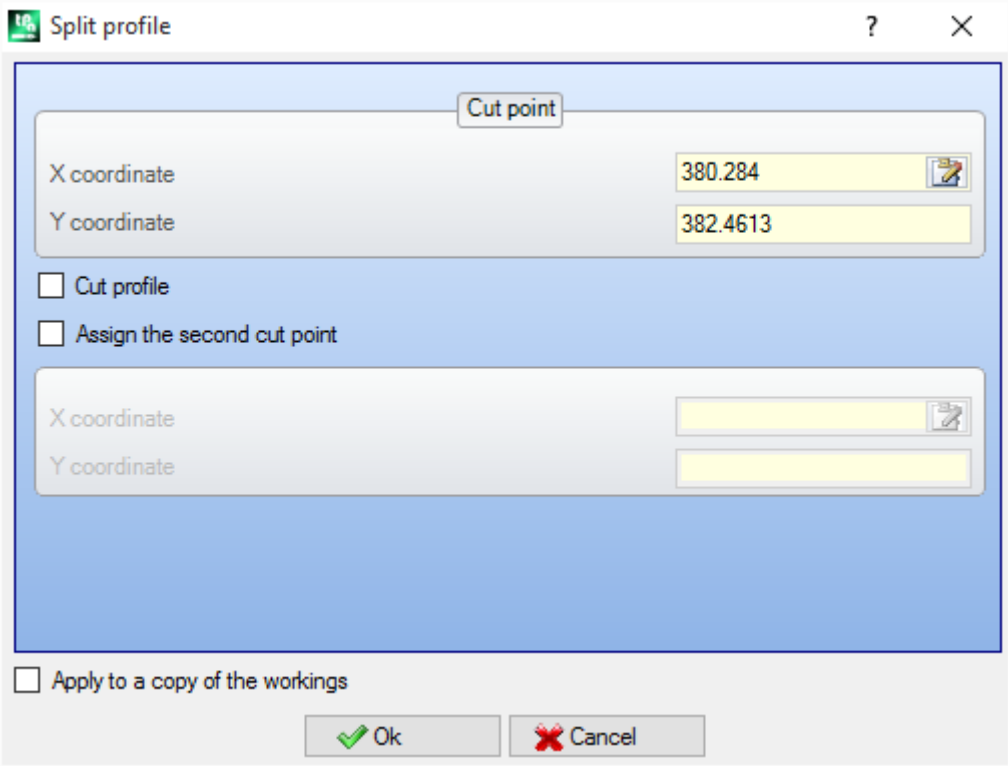
The arcs stretching generates automatically an arc of ellipse. In case of profiles with assigned entry and/or exit traits, the scale is also applied to the geometry of the traits also if the scale factors are the same.



- on the left, the original profile;
- on the right the resulting profile, where the scale factors are applied as indicated: in both cases, the original semi-circle has been modified with a semi-ellipse.

Split profile

The **Split profile** command available in the group **Change profiles** of the **Tools** tab. The tool allows to deletion of a portion of the current profile or the fragmentation of a profile into two separate segments.



- **Cut profile**: assigns the (X, Y) position of the cut point on the current profile (click the icon to position with the mouse on the graphic area);
- **Assign the second cut point**: select to assign a second point along the profile and delete the portion of profile between the two points.

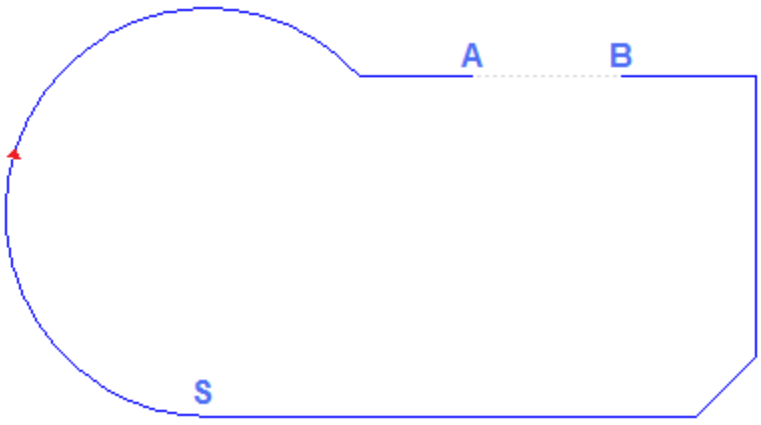
If the option is not selected, the tool split the profile in the first assigned cut point. In this case there are two distinct behaviours according to the status of the option **Cut the profile**:

- if selected, the profile is cut on the point and two profiles appears: the first one ends on the cut point, while the second one starts from the cut point and includes the final part of the original profile;
- if not selected: the profile segment on which the cut point lies, is divided in two parts, but the profile remains single.

If the option **Assign the second cut point** is selected, assign the second cut point in direct or interactive way.

Acquiring with the mouse the position of the cut point, the coordinates are assigned so that they correspond to a profile along the profile. If the same coordinates have to be directly assigned or modified, the nearest point to the point set along the profile is searched.
In the figure an example of profile:

- (S) indicates the profile starting point;
- the arrows indicates a counterclockwise direction;
- the example profile is closed



On the profile 2 cut points have been marked and namely: (A) and (B) (the two points can lie on the same segment or on different segments). The part of profile between the two points is deleted (**ATTENTION**: in the direction of the original direction).
The original profile is then broken by the tool into 2 profiles:
- the 1st profile starts from (S) to (A);
- the 2nd profile starts from (B) to (S).

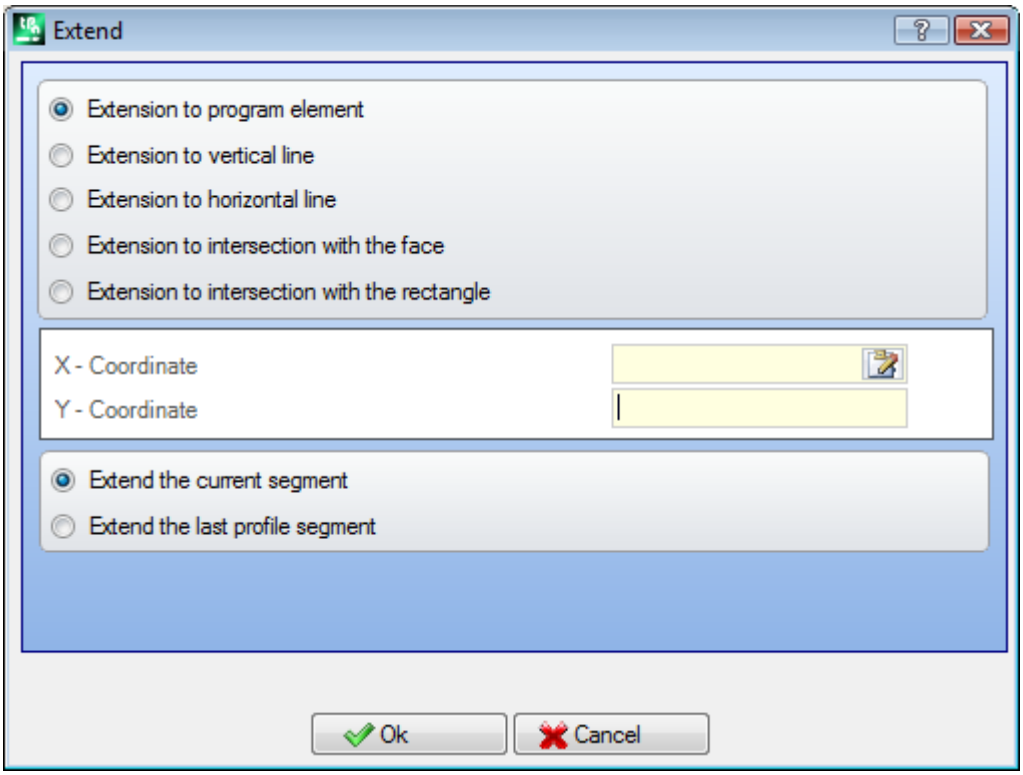
To break the concerned linear segment in only one point (example: (A), it is enough not to mark the second cut point.
In this case we have a single profile with one more segment.
To break in two marked out profiles, on (A) point it is enough not to mark the second cut point and select the option ***Cut profile***.

Take off each profile segment

Take off each profile segment is available in the group **Change profiles** of the **Tools** tab. This tool modifies the current profile or one of its copies by taking off each single segment and defining many distinguished profiles.
If the new profile starts by a setup, the user can start each new removed profile with a copy of this one or, if possible, he can apply the coordinates of the start point directly on each separated segment.

Extend

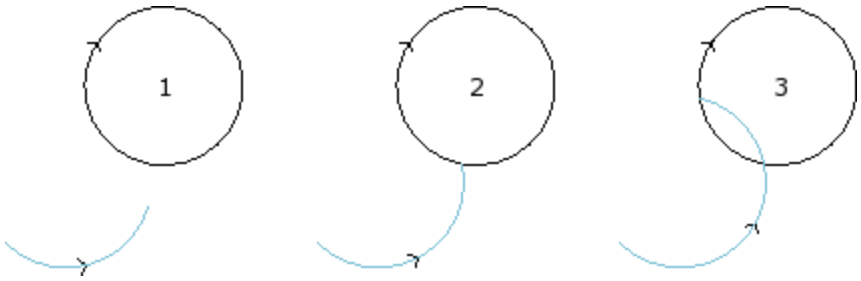
The **Extend** command is available in the group **Change profiles** of the **Tools** tab. The tool extents a segment of the current profile (current segment or last profile segment) until the intersection with a selected bounding element.
The current segment must belong to a profile and have one the following type:
- arc, but it cannot be a circle;
- a line with non-null length;
- if element of a path (L24); this is considered as a linear segment.



The possible selections of the bounding element are:
- **Extension to program element**: the delimitation element is set up by a profile programmed working, excluding the point and setup workings. The element is found by assigning a point near the segment: the X and Y coordinates are programmed in the edit fields (also in the interactive mode, clicking the icon). Section is extended to intersect the profile closest to selection point. If more intersection solutions exist, the closest to the point of origin is the valid one.

- **Extension to vertical line**: the delimitation element is set up by a vertical line. The coordinate of the vertical axis is programmed in the edit field (also in interactive mode, clicking the icon).
- **Extension to horizontal line**: the delimitation element is set up by a horizontal line. The coordinate of the horizontal axis is programmed in the edit field (also in interactive mode, clicking the icon).
- **Extension to intersection with the face**: the delimitation element is set up by overall rectangle of the face. The segment is extended until the intersection with a face side.
- **Extension to intersection with the rectangle**: bound element is set up by a rectangle. The coordinate of the horizontal axis is programmed in the edit fields (also in interactive mode, clicking the icon).

The figure shows the case of two applications of the tool.

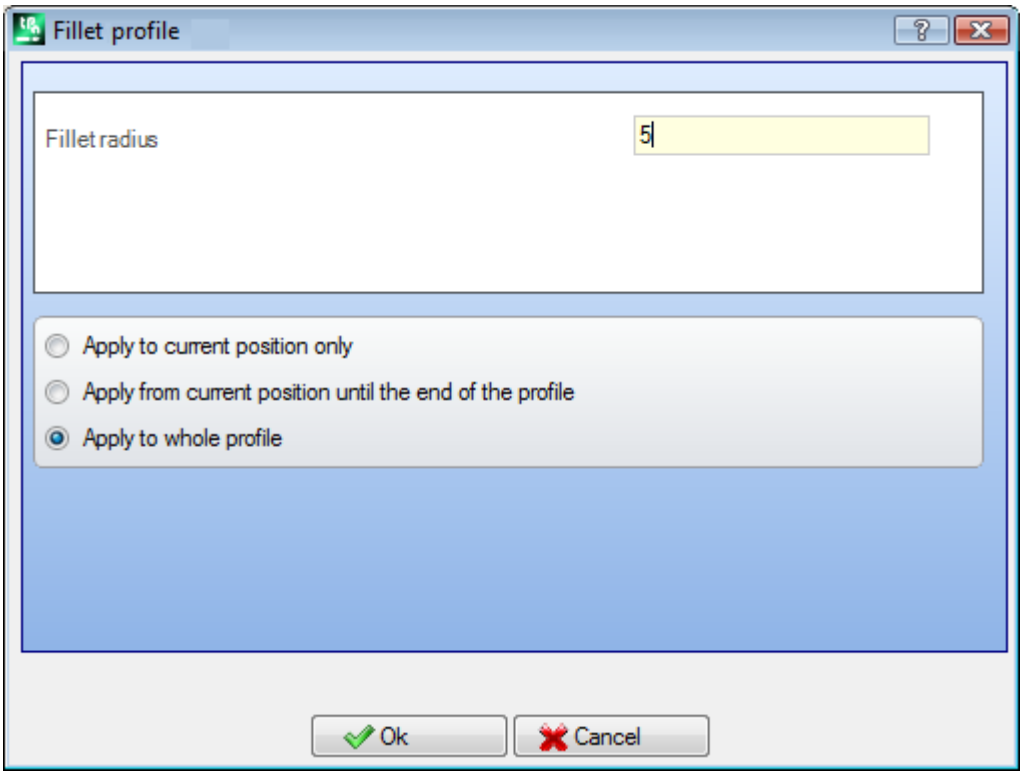


The arc represents the segment which must be extended up to the point of intersection with the circle

- figure **1** corresponds to the starting situation;
- figure **2** corresponds to the first extension application;
- figure **3** corresponds to the second extension application.

Fillet profile

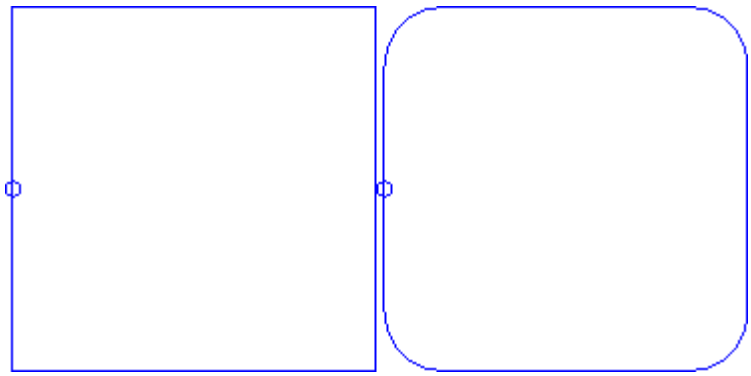
The **Fillet profile** command is available in the **Change profiles** group of the **Tools** tab. The tool inserts fillet arcs at the corners of a profile: each fillet is set up to guarantee the tangent continuity with the original segments of the corners.
The original corners can be delimited between two linear segments, a linear segment and an arc, two arcs.
The tool does not expand possible complex workings or multiple segments of a profile: if needed, apply first the tool **Explosion** (**General tools** group of the **Tools** tab).
This tool is directly applied to the current profile only.



- **Fillet radius**: radius of the fillet arc inserted on the corner.

It applies to:

- **Apply to current position only**: it inserts a fillet on the corner formed by the active working with the following working
- **Apply from current position until the end of the profile**: it inserts some fillets on each profile corner starting from the current working.
- **Apply to whole profile**: it inserts some fillets at all profile corners



On the left a rectangle with sharp corners.
On the right the final result after the tool application to the whole profile.

The application of the tool can affect only the corners that have such a size that it can insert the fillet within (and not outside) the original segments.

PROFESSIONAL

It is possible to generate profiles by inserting some junction arcs into the corner points of the profile, also in the form of complex working, by recalling the Programmed tools in the list of the workings. In the group of TOOLS select the STOOL: FILLET PROFILE:

- the **Workings** field sets the names assigned to workings programmed before that correspond to the original profiles.

The profiles can also be the result of the application of other complex codes; the development of the working corresponds only to the modified profile and it does not include the original profiles. Possible workings that cannot be used for the functionality required (for example, logical workings or workings of the points or complex workings that cannot be exploded) are ignored.

In the working following data are set:

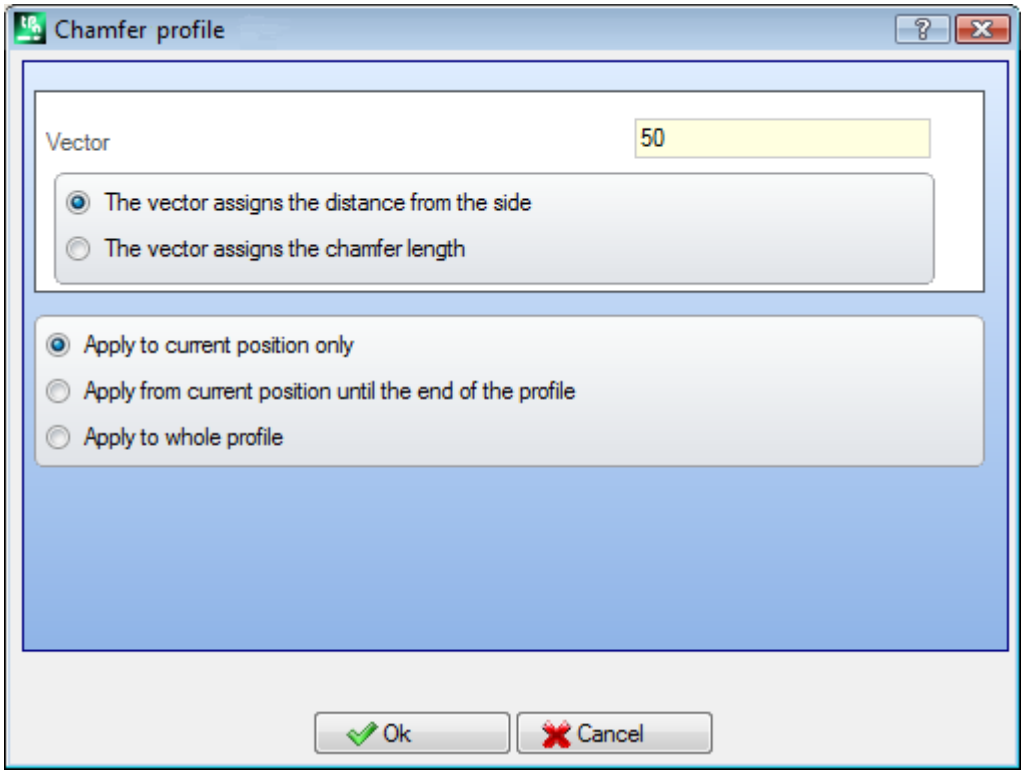
- Typical parameters of a complex working (see the preceding discussion of a generic Subroutine code):
  - **Qx,Qy Zp**: initial positioning coordinates of the developed workings;
  - **...**
  - **Working properties**: sets the properties given to the working.

- Specific parameters of the working functionality, with a meaning analogue to the fields defined in the window of the tool.
  - **Fillet radius**: radius of the junction radius inserted into the corners.
  - **Apply to acute angles**: if selected, this option enables the application of the fillet only to the corners within a right angle (<90°).
  - **Apply only to vertices with arc** if selected, this option enables the application of the fillet only to the corners assigned between line-arc, arc-line, arc-arc. So, the selection excludes the situations assigned between line-line.

The main advantage offered by the use of the working STOOL: FILLET PROFILE where the profiles generated adapt to the modifications of the original profiles, besides the fact that they can totally work on more than one profile, and also on the complex ones.

Chamfer profile

The **Chamfer profile** command is available in the **Change profiles** group of the **Tools** tab**.** The tool chamfers the profile at its corners; only the corners that are defined between two linear segments can be changed.
The tool does not expand possible complex workings or multiple segments of a profile: if needed, apply first the tool **Explosion** (**General tools** of the **Tools** tab).
This tool is directly applied to the current profile only.



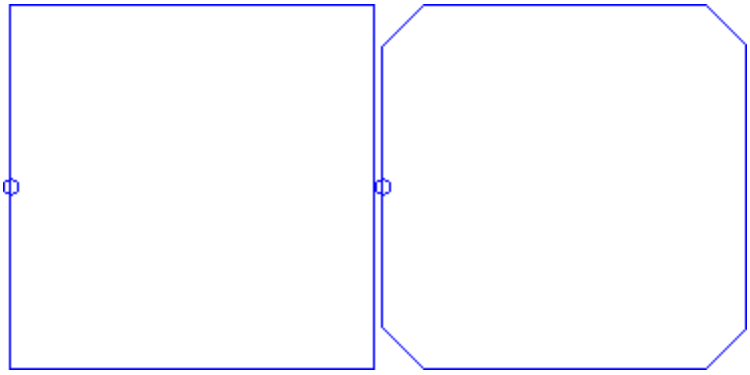
- **Vector:** the assigned length to define the chamfer.

Two options can be marked out:

- **The vector assigns the distance from the corner**: the *Vector* field sets the distance from the extreme points of the chamfer to the original corner (figure on the left U shows the value set).
- **The vector assigns the chamfer length:** The *Vector* field sets the length of the chamfer (figure on the right).



It applies to:
- **Apply to current position only**: it inserts a chamfer on the corner formed by the active working with the following working.
- **Apply from current position until the end of the profile**: it inserts some chamfers on each profile corner starting from the current working.
- **Apply to whole profile**: it inserts some chamfers at all profile corners.



On the left a rectangle with corners.
On the right the final result after the tool application to the whole profile.

The application of the tool can affect only the corners that have such a size that it can insert the chamfer within (and not outside) the original segments.

PROFESSIONAL

It is possible to generate profiles by inserting some chamfers into the corners of the profile, also as complex working, by recalling the Programmed tools in the list of the workings. In the group of TOOLS select the STOOL: CHAMFER PROFILE:

- the **Workings** field sets the names assigned to workings programmed before that correspond to the original profiles.

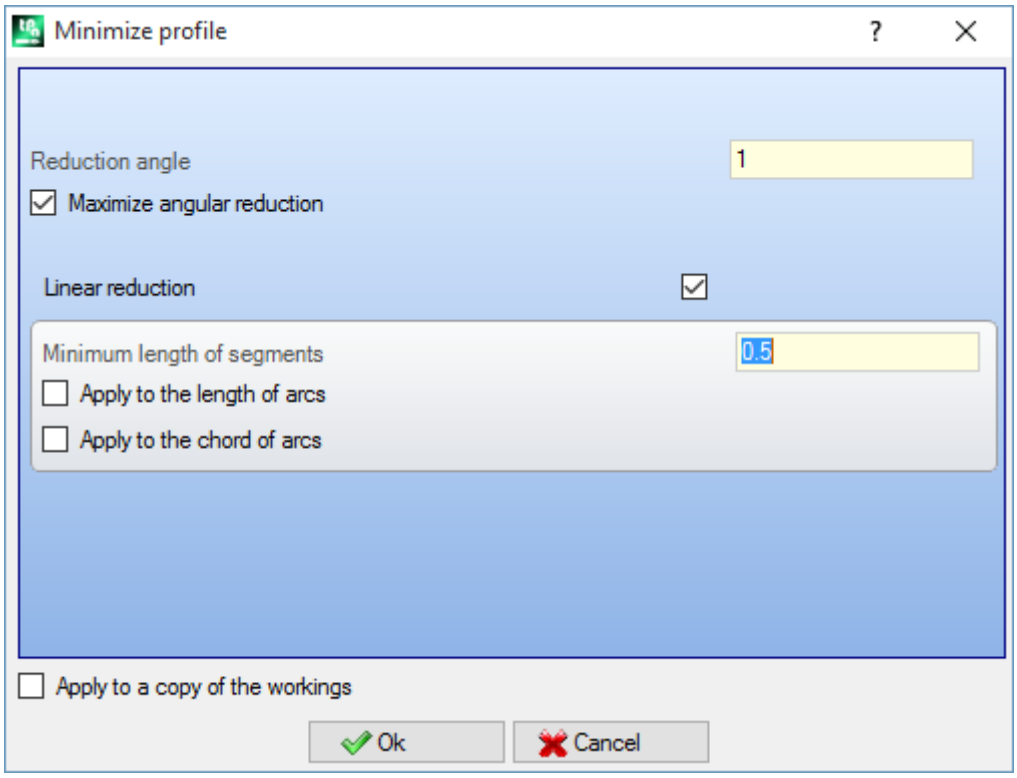
The profiles can also be the result of the application of other complex codes; the development of the working corresponds only to the modified profile and it does not include the original profiles. Possible workings that cannot be used for the functionality required (for example, logical workings or workings of the points or complex workings that cannot be exploded) are ignored.

In the working following data are set:
- Typical parameters of a complex working (see the preceding discussion of a generic Subroutine code):
  - **Qx,Qy Zp**: initial positioning coordinates of the developed workings
  - **...**
  - **Working properties**: sets the properties given to the working.
- Specific parameters of the working functionality, with a meaning analogue to the fields defined in the window of the tool.
  - **Chamfer:** length assigned to define the chamfer;
  - **Typology:** it assigns the kind of chamfer to be applied.
    - **Chamfer**= the value assigned to the **Chamfer** parameter is the length of the chamfering segment.
    - **Lines to vertex=** the value assigned to the **Chamfer** parameter is the length of the linear segments available on the two lines from the corner on which the chamfering is required.
  - **Apply to acute angles:** if selected, this option enables the application of the fillet only to the corners within a right angle (< 90°).

The main advantage offered by the use of the working STOOL: CHAMFER PROFILE, where the profiles generated adapt to the modifications of the original profiles, besides the fact that they can totally work on more than one profile, and also on the complex ones.

Minimize profile

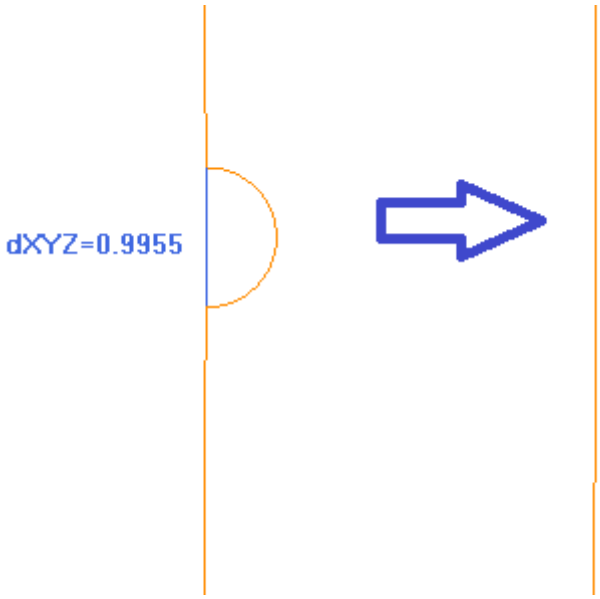
The **Minimize profile** command is available in the **Change profiles** group of the **Tools** tab. This tool allows the reduction of the segment numbers of which a profile is made. They are reduced by unifying the consecutive linear and or curved segments for which a condition of geometric continuity is verified, by selecting the criteria, as follows:



- **Reduction angle**: in degrees, assigns the maximum angular cone inside which the consecutive linear segments are unified. The value set must be included between 0.0º and 90°: the value 0.0º means that no angular reduction is applied.
- **Maximize angular reduction**: if selected, this option requires an iteration of the procedure for the angular reduction, until its application is reset.
- **Linear reduction**: if selected, this option requires a reduction that calculates the length of the single segments. The selection of the field enables the application of the following settings.
- **Minimum length of segments**: minimum length of the segments, calculated as a linear distance.
- **Apply to the length of arcs**: if selected, this option applies the minimum length also the arcs and calculates the linear length of the arc
- **Apply to the chord of arcs**: if selected, this option applies the minimum length also the arcs by calculating the linear length of the arc.

The tool applies
- a preliminary reduction of the profile by unifying the linear segments of the minimum length lower than epsilon;
- if required, the angular reduction of the linear segments with a possible iteration to maximize its effects;
- a reduction of the consecutive arcs related to one only split arc;
- if required, the reduction calculated on the linear distance required.

It is clear that the more the required phases are, the more the consequent processing difficult is also in relation to the dimension of the profiles that must be worked.
The picture shows an example of linear reduction, applied to a profile generated with path imperfections. The applied zoom is very high:

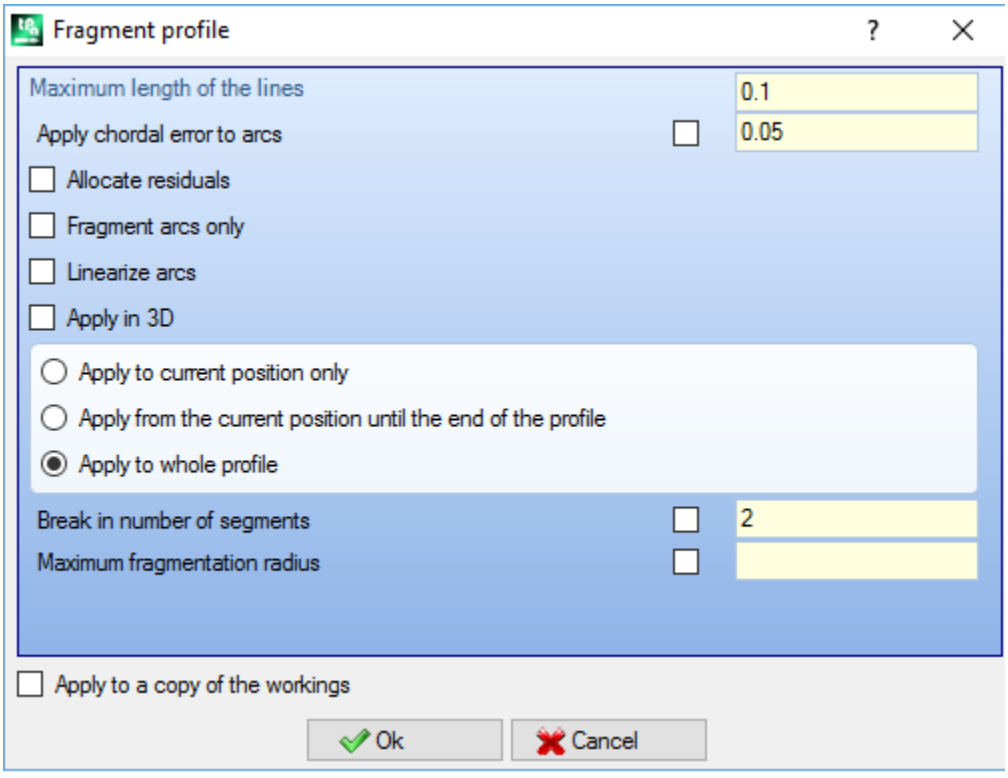


On the left the original path is displayed, light-coloured; it is clear that the indented area and the blue segment highlight its scale that is less than 1 mm;
on the right, it is displayed the result that can be obtained with a linear reduction, for example, of 0,3 mm.

The selection of *Apply to a copy of workings* applies the tool to a copy of the workings and does not change the original lines.
This tool is also available as a general program tool.

Fragment profile

The **Fragment profile**  command is available in the **Change profiles** group of the **Tools** tab. This tool breaks up the profile segment portions into several segments, selecting different criteria of fragmentation.



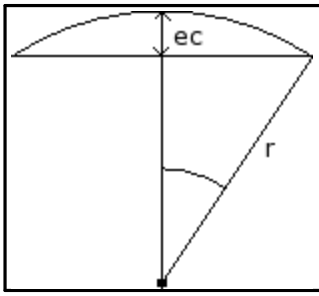
- **Maximum length of the segments**: the maximum length of the profile fragmentation length. The minimum value is positive and equal to 10.0*epsilon of coordinate resolution.
- **Apply chordal error to arcs**: if enabled, this option breaks up the arcs and assigns the Chordal error set (see later). In this case the value of **Maximum length of the segments** is applied to the fragmentation only of the linear segments.
- **Allocate residuals**: if selected, on each original individual segment the tool calculates the number of fragments into which the segment is broken up and on these the residual is distributed.
  For example, if a linear segment is 52 mm long and a segment length of 10 mm is assigned:
  - if the option is not selected, the linear segment is broken up into 6 segments: 5 10 mm long and 1 (the last one) 2 mm long;
  - If the option is not selected the linear segment is broken up into 6 segments of the same length. The length is recalculated and it will be equal to (52/6) =8.6666.
- **Fragment arcs only**: this tool is applied to the only curved segments. The selection is not applied if the option **Apply to the current position only** is selected.
- **Linearize arcs**: it breaks up the arcs into sections that are converted in linear segments.
- **Apply in 3D**: the maximum length of segments is also applied to the depth component.

It applies to: three options are available:

- **Apply to current position only**: it breaks the current segment only
- **Apply from current position until the end of the profile**: it breaks up from the current segment to the end of the profile.
- **Apply to whole profile**: it breaks up all profile.
- **Break in number of segments**: selecting **Apply to current position only** the user is enabled to break up the element into an assigned number of segment. In this case, the setting of **Maximum length of the segments** is ignored and automatically calculated. The field accepts a numeric input ranging from 2 to 99.
  The option is enabled, only if:
  - the tool is applied to the current working;
  - the current working has a line or an arc typology and performs a single geometric segment.
- **Maximum fragmentation radius**: allows the fragmentation of only the arcs with maximum radius assigned. The field accepts a positive numeric value. For example, select the check box and set 4.0 value: the tool breaks up the arcs whose radius is less or equal to 4.0 mm. The selection is not applied, if the option **Apply to current position only** is selected.

The selection of *Apply to a copy of workings* applies the tool to a copy of the workings and does not change the original lines.
If in the setup of the original profile the entry and/or exit segments are set, these remain directly assigned in the setup and are not subject to fragmentation.
This tool is also available as a general program tool.

Splitting up an arc



The figure shows the geometric meaning that has been given to the chordal error set (a set value of chordal error can be 0,05 mm).
Splitting up an arc according to the criteria of the chordal error determines samplings whose length changes according to the radius of the arc. If the radius increases, the length of the segments increases accordingly.
Splitting up an arc according to the chordal error criteria maintains the same sampling precision to all the arcs, because it depends on the arc curvature.

Furthermore

- for each splitting a maximum chordal error equal to 50% of the radius of the arc is accepted
- for an arc a sampling number is resolved anyway and it is not lower than the entire fractions of 45° (of the arc dimension);
- for each splitting precise limits are accepted for the sampling angle, that is calculated. Its minimum value is 1° and its maximum one is 45°.

PROFESSIONAL

It is possible to generate profiles by inserting profile fragmentation and linearization, also in the form of complex working, by recalling the Programmed tools in the list of the workings. In the group of TOOLS select the STOOL: FRAGMENT AND LINEARIZE:

- the **Workings** field sets the names assigned to workings programmed before that correspond to the original profiles.

The profiles can also be the result of the application of other complex codes; the development of the working corresponds only to the modified profile and it does not include the original profiles. Possible workings that cannot be used for the functionality required (for example, logical workings or workings of the points or complex workings that cannot be exploded) are ignored.

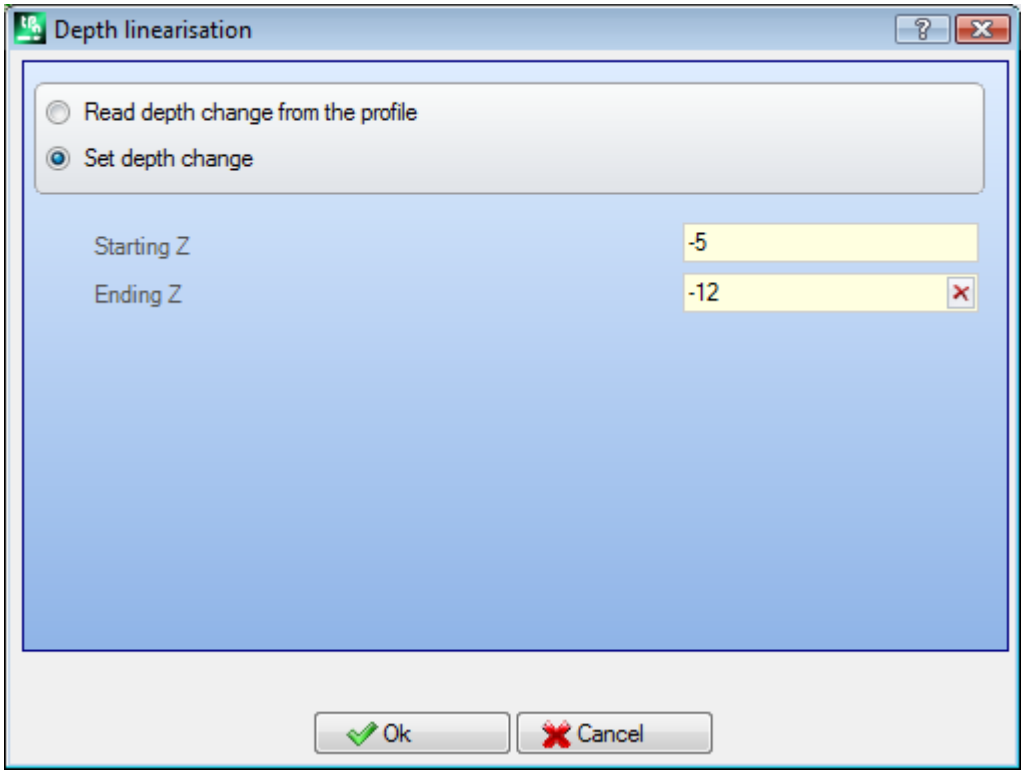
In the working following data are set:

- Typical parameters of a complex working (see the preceding discussion of a generic Subroutine code):
  - **Qx,Qy Zp**: initial positioning coordinates of the developed workings.
  - **...**
  - **Working properties**: sets the properties given to the working.
- Specific parameters of the working functionality, with a meaning analogue to the fields defined in the window of the tool.
  - **Maximum length of the segments**: it sets the maximum length of the segments into which the profile is broken.
  - **Apply chordal error to the arcs**: if enabled, this command divides the arcs into fractions, by assigning the Chordal error set in the next. field. In this case the value of **Maximum length of the segments** is applied only to the fragmentation of linear segments.
  - **Chordal error** it sets the chordal error.
  - **Allocate residuals**: if selected, it calculates on each single segment the whole number of the segments into which to fragment and along those it distributes the residual part.
  - **Fragment arcs only**: this tool is applied to the only curved segments.
  - **Linearize arcs**: this tool breaks the arcs into segments that are converted in linear segments.
  - **Apply in 3D:** the maximum length of the segments is also applied on the depth component.

The main advantage offered by the use of the working STOOL: FRAGMENT AND LINEARIZE, where the profiles generated adapt to the modifications of the original profiles, besides the fact that they can totally work on more than one profile, and also on the complex ones.

Linearize Z

The **Depth linearization** command is available in the group **Change profiles** in the **Tools** tab. This tool linearizes the profile variation depth. This tool works on simple profiles with only arcs on xy planes.



Select one of the two suggested options:

- **Read depth change from the profile**: it reads the extreme (starting and ending) Z coordinates as assigned on the profile and makes the data-entry fields inaccessible.
- **Set depth change**: sets the initial Z coordinate and the final Z coordinate directly in the fields. If the final Z is not assigned (empty field), it takes the same value of the initial Z, adjusting the depth on the whole profile.

With both the Z coordinates, as assigned in the figure, this option changes the final application depth of each profile segment in such a way that a gradual variation of the depth along the whole profile is obtained from the Z=5 coordinate to the final coordinate of Z=12.
This tool is directly applied to the current profile only.

PROFESSIONAL

It is possible to generate profiles with depth linearization also in the form of complex working, by recalling the working Programmed tools in the working list. In the TOOLS group select STOOL: LINEARIZE IN Z working.
- The field **Workings** sets the assigned names to before programmed workings corresponding to the original profiles.

The profiles can also be the result of the application of other complex codes and the development of the working corresponds only to the modified profile(s) and does not include the original profiles. Possible workings that cannot be used for the function required (for example: point or logical workings or complex workings that are not expandable) are ignored.

The working sets:
- Typical parameters of a complex working (see what has been said about a generic code of Subroutine):
  - **Qx,Qy Zp**: initial positioning coordinates of the developed workings;
  - **...**
  - **Working properties**: it sets the properties attributed to the workings.

- Specific parameters of the working function with a meaning analog to the fields in the tool window:
  - **Set the depth**: it enables the setting of the extreme coordinates.
  - **Starting Z, Ending Z**: they assign the initial and the final Z coordinates, where the **Set depth change** option is selected.

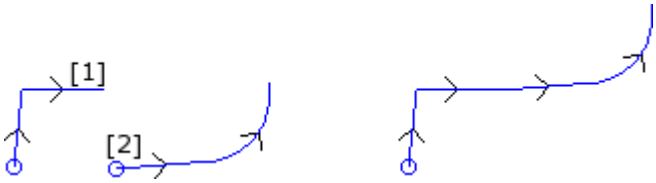
The main advantage of using the STOOL: LINEARIZE IN Z working consists in the fact that the generated profiles fit changes of the original profiles, besides the fact that it is possible to work on more than one profile, also on complex ones.

Profile Unions

With translation

The **Joint profiles by translation**  command is available in the **Change profiles** group of the **Tools** tab.
The tool joins two or more profiles, by translating them so that the starting point of the second profile coincides with the end point of the first profile. The tool works also on extended profiles.
In piece-face the tool is only enabled with the Box or the 2D face view active and operates only on the profiles applied to the face in current view.
Selecting the command, the option requires if apply the tool to a working copy.

The single profiles are set, if they are selected with the mouse in the graphic area; they are numbered to highlight in which order the profiles have been jointed. Related instruction are given in the Commands area and it is possible to select up to maximum 99 profiles.
To cancel the last selection, press the "Z" button. Invalid or duplicated selection are notified by messages.
Close the selection with **[Enter]** to confirm the tool application, **[Escape]** to cancel the selections.



On the left: the starting situation displaying two separate profiles. The profile **(1)** is the first selected and the profile **(2)** is the second one selected.
On the right: the final situation after the tool application. The profile **(1)** remains in the original position and the profile **(2)** is translated to the end point of the segment **(1)**. As a result we obtain one only profile with setup in the start point of the profile **(1)**.

With connection segment

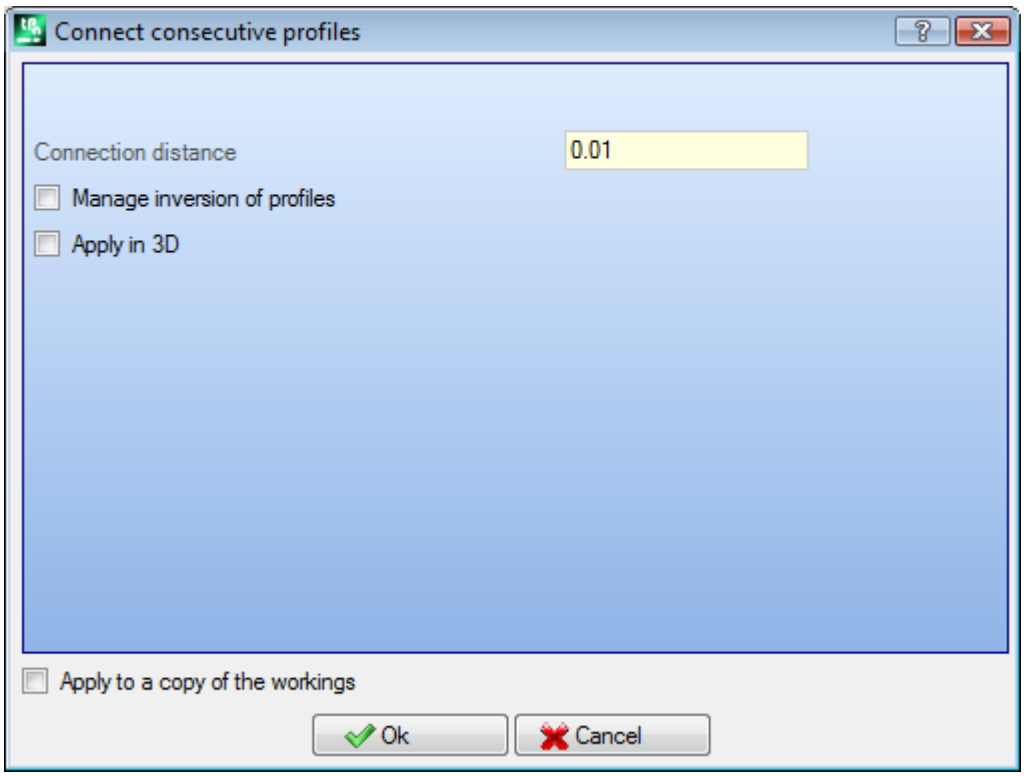
The **Connect profiles with segment**  command is available in the **Change profiles** group of the **Tools** tab.
This option connects the profiles without changing their position and inserting two connection linear segments.
See the same information on the previous tool.



On the left: starting situation displaying two separate profiles. The profile **(1)** is the first selected and the profile **(2)** is the second one selected.
On the right: the final situation after the tool application. The profiles remain in their original position and a linear connection segment is inserted between the end point of the segment **(1)** and the start point **(2)**. As a result we obtain one only profile with setup in the start point of the profile **(1)**.

Connect consecutive profiles

It connects the profiles whose final or start point coincides with the initial or end point of the next one. The **Connect consecutive profiles** command is available in the **Change profiles** group of the **Tools** tab.



- **Connection distance**: this is the largest distance allowed for two profiles, calculated between the end point of the first one and the start point of the second one to enable the automatic connection. A value between system *epsilon* and the value (100 * *epsilon*) is accepted. If the value set here is greater than the system *epsilon*, the connection moves the second profile in such a way that the geometric continuity with the previous one is made.
- **Manage inversion of profiles**: select to enable the inversion of the profile after the first one, in order to calculate the possibility of connection in the best way.
- **Apply in 3D**: select to calculate the distance of profiles also on the depth component (Z axis).

The selection of *Apply to a copy of workings* applies the tool to a copy of the workings and does not change the original lines.

Close the window on OK to confirm the settings and to continue with the tool. Then, it is required **showing the first profile** by means of the mouse pointer.
After performing the selection, the user is required **enabling the connection automatic search**:
- if the answer is positive, all consecutive profiles are connected automatically;
- if the answer is negative, keep on selecting the profile to be connected (up to maximum 99 profiles), as already seen for the two previous tools.

PROFESSIONAL

It is possible to generate profiles by means of connection also in the form of complex working, by recalling the working *Programmed tools* in the working list. In the TOOLS group select the STOOL:CONNECT PROFILES:
- The field **Workings** sets the assigned names to before programmed workings corresponding to the original profiles.

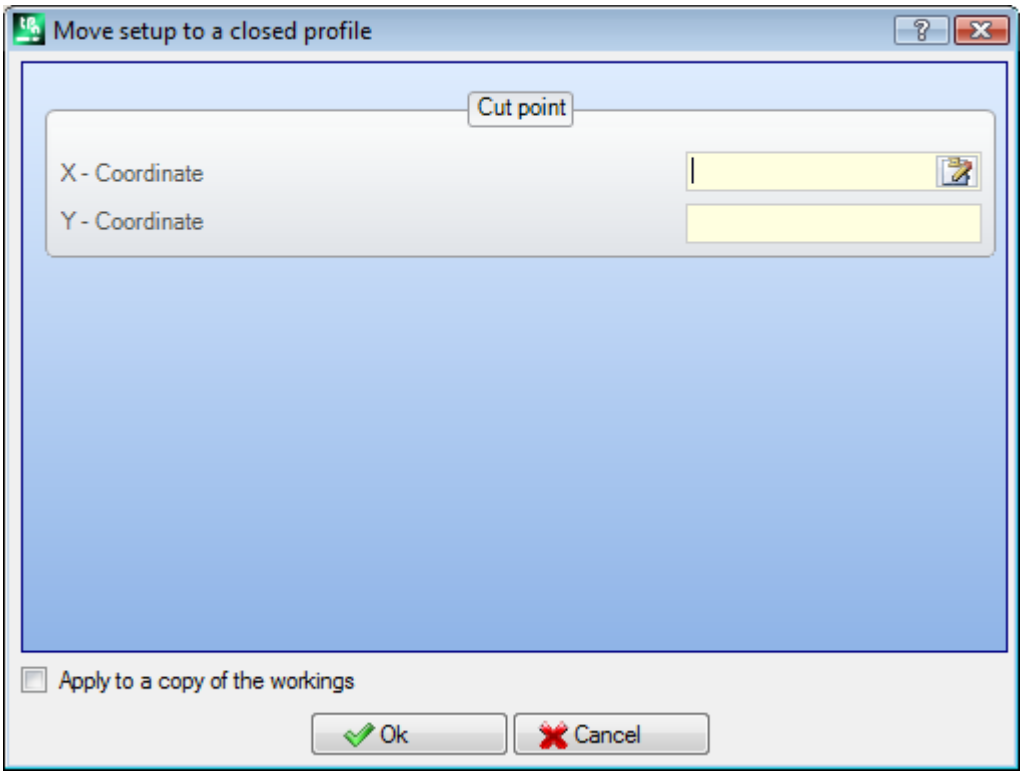
The profiles can also be the result of the application of complex codes and the development of the working corresponds only to the modified profile(s) and does not include the original profiles. Possible workings that cannot be used for the function required (for example: point or logical workings or complex workings that are not expandable) are ignored.

The working sets:
- Typical parameters of a complex working (see what has been said about a generic code of Subroutine):
  - **Qx,Qy Zp**: initial positioning coordinates of the developed workings;
  - **...**
  - **Working properties**: it sets the properties attributed to the workings;
- Specific parameters of the working function with a meaning analog to the fields in the tool window:
  - **Translate profiles**: select to enable the profile translation (see the tool: *Joint profiles by translation*). If the option is not selected, profiles are connected by inserting linear connecting segments (see the tool: *Connect profiles with connection segments*).

Profiles are connected without changing the order or the original direction of the profiles.
The main advantage of using the STOOL: CONNECT PROFILES consists in the fact that the inserted profiles fit changes of the original profiles.

Move setup to closed profile

The **Move setup to closed profile**  is available in the **Change profiles** group of the **Tools** tab. This tool moves the current profile setup to a different point of the profile itself. The profile must be closed, the initial and end points must be coincide in all the coordinates (x, y, z).



X - Coordinate, Y - Coordinate: they set the setup new position. Click the icon  to select with the mouse in the graphic area. Acquiring with the mouse the position of the point, the coordinates are assigned so that they correspond to a point along the profile. If the same coordinates have to be directly assigned or modified, the nearest point to the point set along the profile is searched.

The profile is modified while the sense of the direction is kept unchanged. If the original profile does not start with a setup working, the new setup point is assigned with a copy of the reference setup (as assigned **Customize -> Technology -> Default codes** of the Application menu).

By selecting the option *Apply to a copy of the workings* the tool is applied to a copy of the workings and does not modify the original lines.

Apply setup to profile

The **Apply setup** command is available in the **Change profiles** group of the **Tools** tab. This tool applies a technological setup to the selected profiles (profiles that have almost one selected element) or to the current profile.
To know the tool application mode see the paragraph **How to assign the technology to a profile**.
This tool id directly applied to original profiles or to a copy.

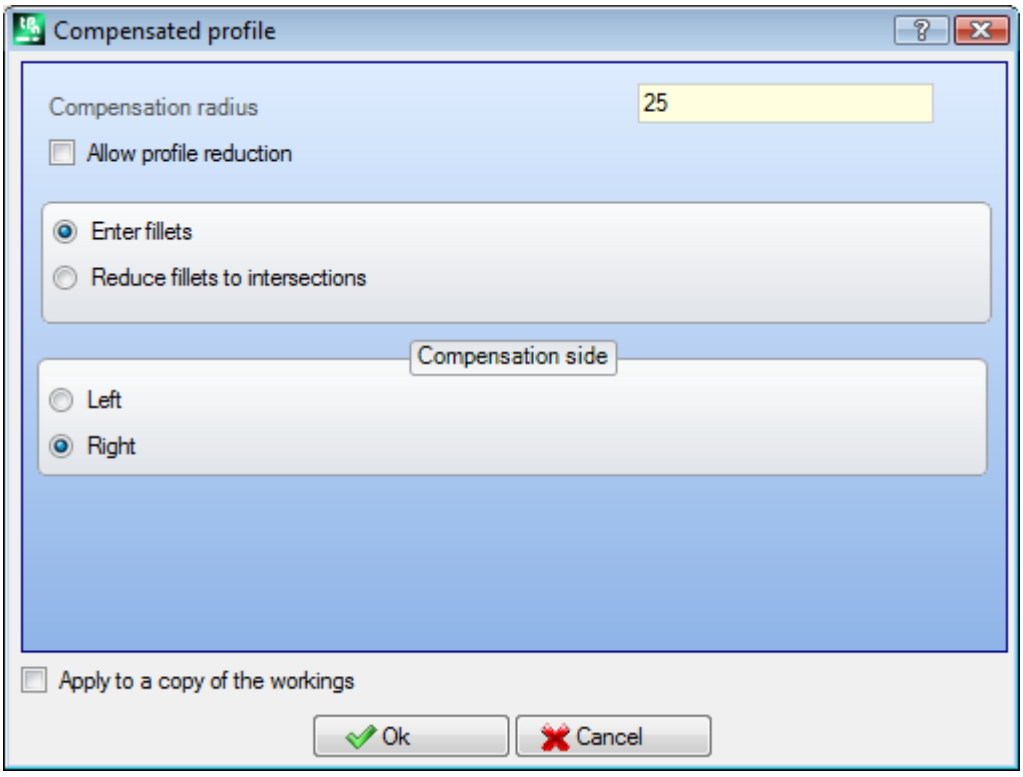
Apply multiple setup

The **Apply multiple setup** command is available in the **Change profiles** group of the **Tools** tab. This tool applies multiple setups to the selected profiles (profiles with almost one selected element) or to the current profile.
To know the tool application mode see the paragraph **How to assign the technology to a profile**.
This tool is directly applied to original profiles or to a copy.

10.4 Constructions

Compensated profile

This option inserts a new profile obtained by tool compensation from the current profile. The **Compensated profile** command is available in the **Constructions** of the **Tools** tab. The tool is enabled, if the current working belongs to a profile.
This command can be also available for the *Essential* functionality.

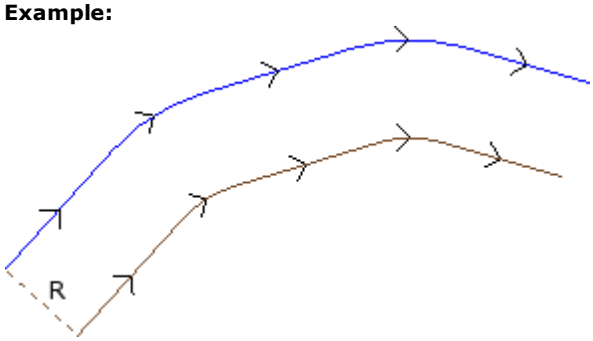


- **Compensation radius**: compensation value
- **Allow profile reduction**: enables the removal of segments in the compensated profile, with respect to the original one on the basis of geometric overall dimensions that exceed the compensation.

In correspondence with the external compensation of a corner:

- **Enter fillets**: select to insert a fillet;
- **Reduce fillets to intersections**: select to calculate a corner.
- **Compensation side**: selects the (left or right) compensation side.

Example:



The original profile is outside. The compensated profile is inside, so the required **Compensation side** is **Right**.
R represents the compensation radius.
If an entry and/or exit segment is set in the setup of the original profile, this remains assigned also in the setup of the compensated profile.

By selecting the option *Apply to a copy of the workings* the tool is applied to a copy of the workings and does not modify the original lines.
The compensated profile is generated with reduction to elementary profile codes and numeric only assignments (that is, it calculates in numeric format each parametrization used while assigning the original profile).
ATTENTION: path elements (L24) are expanded into the micro-segments that assign the curve.

PROFESSIONAL

It is possible to generate compensated profiles also in the form of complex working, by recalling the working Programmed tools in the working list. In the TOOLS group select the working STOOL: COMPENSATED PROFILE.
- The field **Workings** sets the assigned names to before programmed workings corresponding to the original profiles.

The profiles can also be the result of the application of complex codes and the development of the working corresponds only to the modified profile(s) and does not include the original profiles. Possible workings that cannot be used for the function required (for example: point or logical workings or complex workings that are not expandable) are ignored.

The working sets:
- Typical parameters of a complex working (see what has been said about a generic code of Subroutine):
  - **Qx,Qy Zp**: initial positioning coordinates of the developed workings
  - …
  - **Working properties**: it sets the properties attributed to the workings

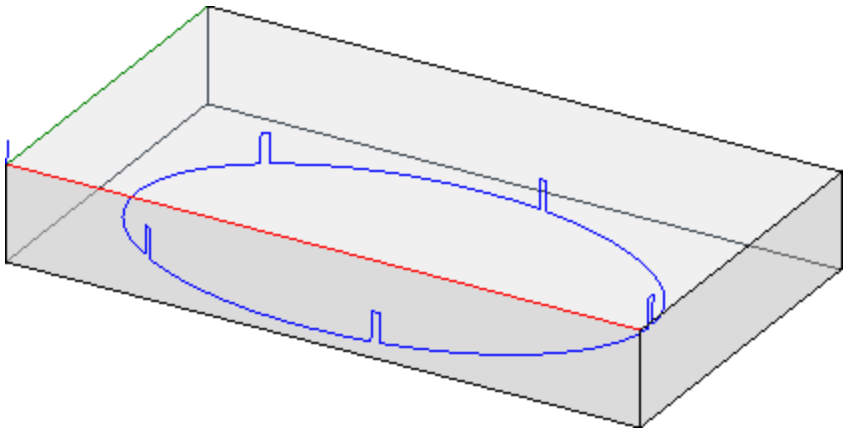
- Specific parameters of the working function with a meaning analogue to the fields in the tool window:
  - **Compensation radius**: compensation value.
  - **Compensation**: selects the (left or right) compensation side.
  - **Contouring**: it selects the compensation mode on the corners. The listed items are two:
    - **Fillets**: it enables the compensation with insertion of fillets.
    - **Corners**: it enables the compensation with searching of the intersections.
  - **Reduce the profile**: it enables the removal of the segments in the correct profile, with respect to the original one, on the basis of geometric clearance restrictions exceeding compensation.

The main advantage of using the STOOL: COMPENSATED PROFILE consists in the fact that the inserted profiles fit changes of the original profiles, besides the fact that it is possible to work on more than one profile.

Apply bridges to profile

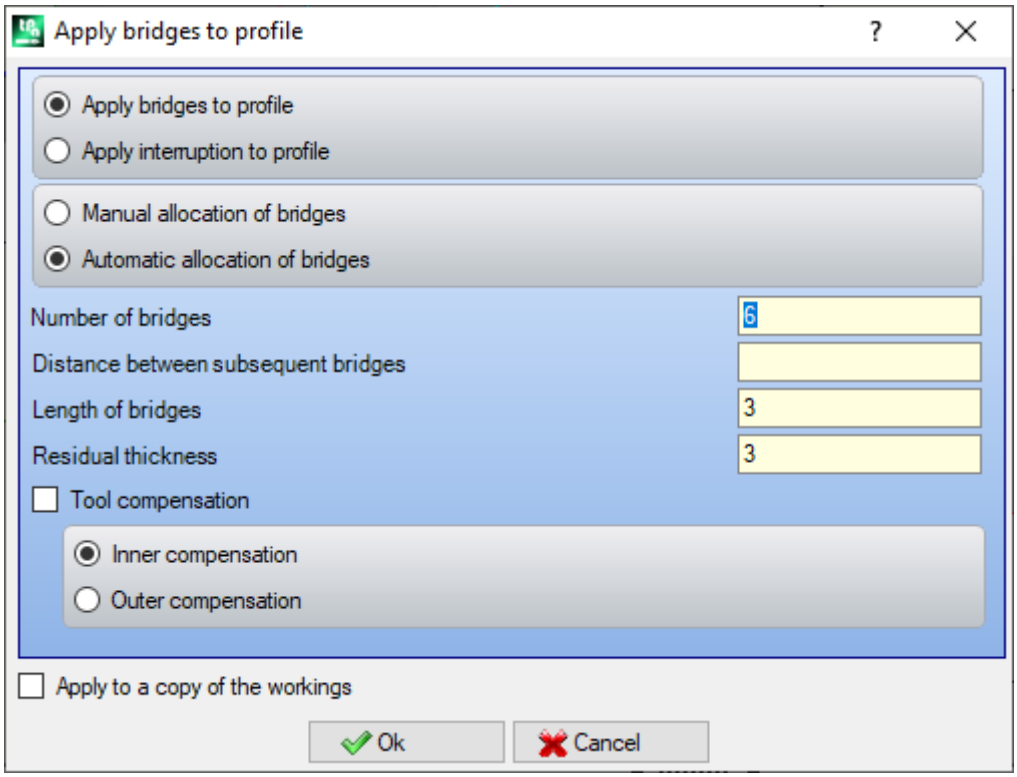
The **Apply bridges to profile**  is available in the group **Constructions** of the **Tools** tab. This tool has a double functionality, allowing you to apply bridges or interruptions.
This command can be also available for the *Essential* functionality.

A bridge results from the fragmentation of an original segment of profile into a short segment, executed at such depth that a residual thickness is left in the piece.
An interruption is to break the original profile segment in a short segment that it is then deleted, breaking the original profile in several profiles.
The tool is enabled, if the current working belongs to a profile.
Applying the bridges, this tool is typically used in case of closed profiles, where milling depth exceeds piece thickness (feed-through profile). In these cases, the direct execution of the profile would cause the detachment and eventual fall of a part of the piece (either the part inside or outside the milled area), during the piece working.
The application of bridges allows the user to leave hook points along the piece thus avoiding the above-mentioned detachment. The figure shows an example of closed feed-through ellipse, with 5 bridges applied.



By applying the interruptions, it is typical to use the tool with cut technologies that differ from a traditional tool: plasma/laser cutting or similar.
The allocation of bridges (or interruptions) on the profile can be executed with the aid of:

- **Manual allocation of bridges**: select with the mouse the profile points on the profile where to apply the bridge. The acquisition procedure of the positions is started up after the closure with window confirmation.
- **Automatic allocation of bridges**: the bridges to the given number are distributed on the profile in the most homogeneous way.

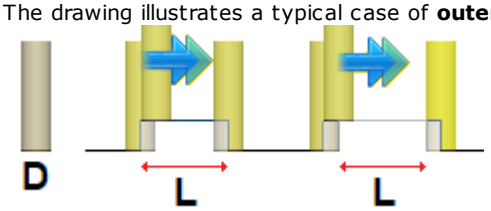


- **Number of bridges**: it sets the number of bridges to be automatically distributed. The field accepts a whole numeric input ranging from 1 to 255.
- **Distance between subsequent bridges**: sets the linear distance between subsequent bridges and it is significant if greater than (epsilon*10.0).
  This setting is an alternative to **Number of bridges**, if the number of the bridges set is lower than 1. If the **Number of bridges** is greater than 1, it supplements its use: the distance set here can be recalculated in order to distribute at least the number of the bridges required. In any case the minimum number of bridges allocated is 1.
- **Length of bridges**: it sets the length of the bridge (in the xy face plane). The value cannot be less than the resolution epsilon value set by the machine manufacturer in the configuration of the application.
- **Residual thickness**: it sets the thickness which the tool leaves in the piece while executing the bridge. It accepts positive values at least equal to the resolution epsilon value (set in the configuration of the application by the machine manufacturer). The parameter is not significant in case of application of interruptions.

- **Tool compensation**: select to modify the real bridge length, taking into account the overall dimensions of the tool. With active selection: the bridge is made reduced or enlarged, to allow the tool to realize it of the required length. The criterion of changing the bridge is determined by the selection:
  - **Inner Compensation**: the bridge is generated narrower than the tool diameter
  - **Outer Compensation**: the bridge is generated enlarged than the tool diameter.

In the case of inner Compensation: the length set for the bridges must be at least equal to the tool diameter.

The drawing illustrates a typical case of **outer** compensation:

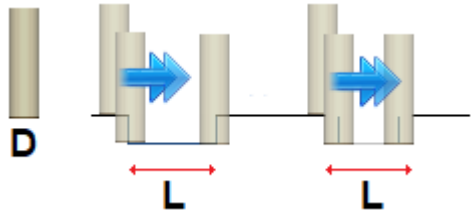


- **L** is the programmed **Length of bridges**
- **D** is the diameter of the tool

On the left there is a bridge executed without Tool compensation and on the right there is the same bridge with outer Compensation:

- without compensation the real length of the bridge is reduced by the value **D**
- with compensation the actual length of the bridge is **L**.

The drawing illustrates a typical case of **inner** compensation:



On the left there is a bridge executed without Tool compensation and on the right there is the same bridge with inner Compensation:

- without compensation the actual length of the connector is increased by the **D** value;
- with the compensation the actual length of the connector is **L**.

In the example of the ellipse in the figure, suppose:
- thickness of the piece of 65 mm;
- programmed depth for the ellipse: -70 mm
- milling cutter diameter: 9 mm

Set the window:
- number of bridges: 5;
- length of bridges: 20 mm
- residual thickness: 2 mm

Along the ellipse the 5 bridges are distributed and are made with:
rising from Z=-70 mm to Z=-(65-2)=-63 mm
execution of an ellipse arc long (20+/-9)=29/11 mm
descent to Z=-70 mm, continuing the programmed trajectory to the next bridge.

If in the setup of the original profile the entry and/or exit segments are set, these remain directly assigned in the setup.
By selecting the option *Apply to a copy of the workings* the tool is applied to a copy of the workings and does not modify the original lines.
The profile with bridges is generated with only numeric assignments: it calculates in numeric format each parametrization used in the assignment of the original profile.

PROFESSIONAL

It is possible to generate profiles modified by the application of bridges also in the form of complex working, by recalling a *Programmed Tools* kind of working in the list of the workings. In the group of TOOLS select the STOOL: APPLY BRIDGES working:

- The field **Workings** sets the names assigned to workings, which have been previously programmed, corresponding to the original profiles.

The profiles may also result from the application of complex codes and development of the working STOOL: APPLY BRIDGES is only for the compensated profile(s) generated and does not include the original profiles.

Possible workings that cannot be used for the function required (for example: point or logical workings or complex workings that are not expandable) are ignored.

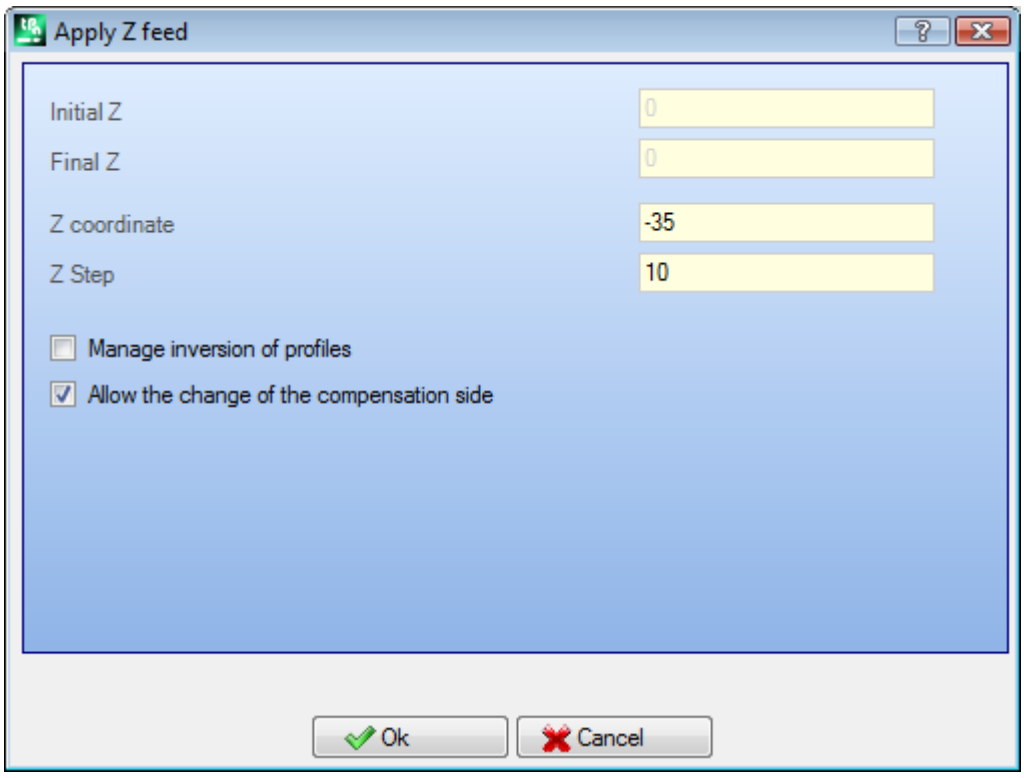
The working sets:
- Typical parameters of a complex working (see what has been said about a generic code of Subroutine):
  - **Qx,Qy Zp**: initial positioning coordinates of the developed workings
  - **..**
  - **Working properties**: it sets the properties attributed to the working

- Specific parameters of the working function with a meaning analogue to the fields in the tool window:
  - **Apply interruptions**: select to apply interruptions.
  - **Number of bridges**: number of bridges to be distributed. The field interprets values between 1 and 255.
  - **Distance between subsequent bridges**: sets the linear distance between subsequent bridges and it is significant if greater than (epsilon*10.0). This setting is an alternative to the **Number of bridges**, if the number set of the bridges is less than 1. If the **Number of bridges** is greater than 1, it supplements its use: the distance set here can be recalculated in order to distribute at least the number of the bridges required. In any case, the minimum number of bridges distributed is 1.
  - **Length of bridges**: length of bridge (in the xy face plane).
  - **Residual thickness**: thickness that the tool leaves in the piece while performing the bridge.
  - **Tool compensation**: if selected, it modifies the actual length of the bridge so as to keep into account the overall dimensions of the tool. The selection is on 3 entries: not required; inner; outer.

The actual number of bridges distributed on each profile also depends on the total development of the profile itself (total length and its fragmentation) and may therefore be lower than the set value.
The main advantage of using the STOOL: APPLY BRIDGES is that the compensated profiles fit changes of the original profiles, besides the fact that it is possible to work on more than one profile.

Apply Z feed

This option modifies the current profile, by inserting consecutive steps until an assigned final depth. The **Apply Z feed**  command is available in the group **Constructions** of the **Tools** tab.
This command can be also available for the *Essential* functionality.
This tool is enabled, if the current working belongs to a profile and works on single profiles only. It is typically used in profiles, that should be executed at a depth measure, that cannot be obtained with one only passage.



- **Initial Z**: shows the initial depth read by the profile. The field is not editable.
- **Final Z**: shows the depth read by the profile corresponding to the next programmed segment (final depth). The field cannot be edited.

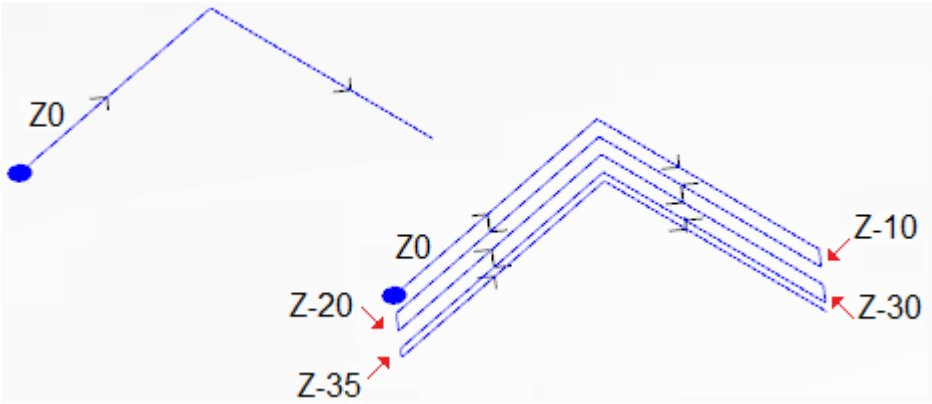
- **Z coordinate**: sets the final depth required in the application of the recursive profile development. It is a position that should be reached according to the step set later. The value must be outside the range between the **initial Z** and the **final Z**.
- **Z Step**: sets the depth step feed applied at each development. This setting is significant without any sign: the procedure applies the feed required (+/-) to reach the final depth. If the value set is null or greater than the maximum value allowed, the value that allows to reach the **Z coordinate** required with a single additional execution, is automatically determined.
- **Manage inversion of profiles**: this option is activated, only if the command is applied to a closed profile. If selected, the option inverts the execution of the profile at every depth change. If the option is not selected, the profile is always executed in accordance to the original direction. If the profile is not closed, the execution is obligatorily inverted at each depth change.
- PROFESSIONAL **Allow compensation side to be changed**: If enabled, to each additional pass an inversion of compensation side (from the right to the left or vice versa) is added. Enabling it is not significant in case of:
  - closed profile where no inversion profile is required;
  - profile where not tool compensation is required.

  This option is managed only if it is enabled by the manufacturer of the machine during the configuration of TpaCAD. This option is available in **Professional** mode only.

If the original profile has not any depth changes, the **Z coordinate** set is also the depth reached on the entire development of the profile. If, on the other hand, the original profile has some depth changes, the control on the achievement of the **Z coordinate** is made on the development of each advancement, calculated as an excursion between **initial Z** and **final Z**.
The value of the last advancement step can be rounded down so that the **Z coordinate** set is not exceeded in the excursion between the **initial Z** and the **final Z**.
In the application of the tool we do not consider intermediate programmed depth changes of the profile.

If in the setup of the original profile the entry and/or exit segments are set, these remain directly assigned in the setup and they are not taken into account in the application of the tool.
On the added passes, all variations of tool offset in the path are reset (interruptions, suspensions and shooting, and/or changes to the side).

The figure below shows an example of tool application:



On the left you will see the original profile, assigned with two linear segments carried out at a depth measure of Z = 0.
On the right you will see the profile after being modified by the tool according to the following assignments:

- Z coordinate = 35.0
- Step Z = 10.0

The original profile is not closed, therefore each repetition inverts the execution.
In the figure there are the different depth positions, as calculated for each passes: the last pass reaches the Z coordinate that has been set by the reduction of the advancement step from 10.0 to 5.0.
In the example, the maximum value of the **Z step** is 35.0.

PROFESSIONAL

It is possible to generate profiles modified by the depth feed also in the form of complex working, by recalling the *Programmed Tools* kind of working in the list of the workings. In the TOOLS group select STOOL: FEEDS IN Z working.

- The field **Workings** sets the assigned names to before programmed workings corresponding to the original profiles.

The profiles may be the result of the application of other complex codes and development of working STOOL: FEEDS IN Z is only for the modified profiles and it does not include the original profiles. Possible workings that cannot be used for the function required (for example: point or logical workings or complex workings that are not expandable) are ignored.
In addition to the tool, the working allows to assign the development axis on one of the three coordinate axes of the face.

The working sets:

- Typical parameters of a complex working (see what has been said about a generic code of Subroutine):
  - **Qx,Qy Zp**: initial positioning coordinates of the developed workings;
  - **..**
  - **Working properties**: it sets the properties attributed to the workings;

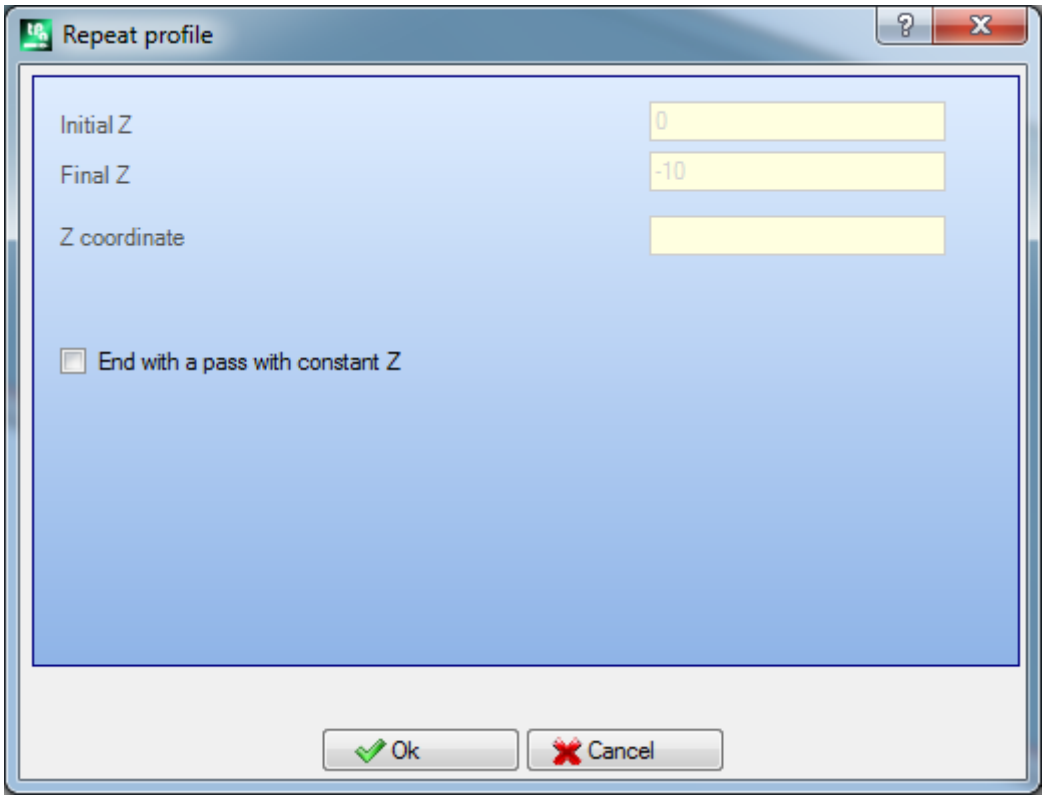
- Specific parameters of the working function with a meaning analog to the fields in the tool window:
  - **Development axis**: selects the development axis on one of the three coordinated axis of the face (Z, X, Y).
  - **Assign the number of passes**: if selected, the option requires the assignment of the number of passes. Otherwise, it requires the assignment of the final depth at which the last feed is performed.
  - **Final position**: it sets the final required depth along the development axis. The assignment is ignored, if the previous option is selected;
  - **Number of passes**: required number of passes (valid setting: 1-1000). The assignment is significant if the option *Assign the number of passes* is selected;
  - **Z Step**: it sets the depth step feed applied at each development. Assigning the final depth, the setting is significant without sign: the procedure applies the feed required (+/-) to reach the final depth.

  Instead the setting is significant without sign in case of assignment of the number or passes.
  - **Manage inversion of profiles**: it manages the inversion of the execution (see the tool).
  - **Allow the change of the compensation side**: it manages the tool compensation side change (see tool). This setting is significant in case of development along the Z axis.

The main advantage of using the working STOOL: FEEDS IN Z working consists in the fact that the compensated profiles fit changes of the original profiles, besides the fact that it is possible to work on more than one profile, also on complex ones.

Apply profile repetition

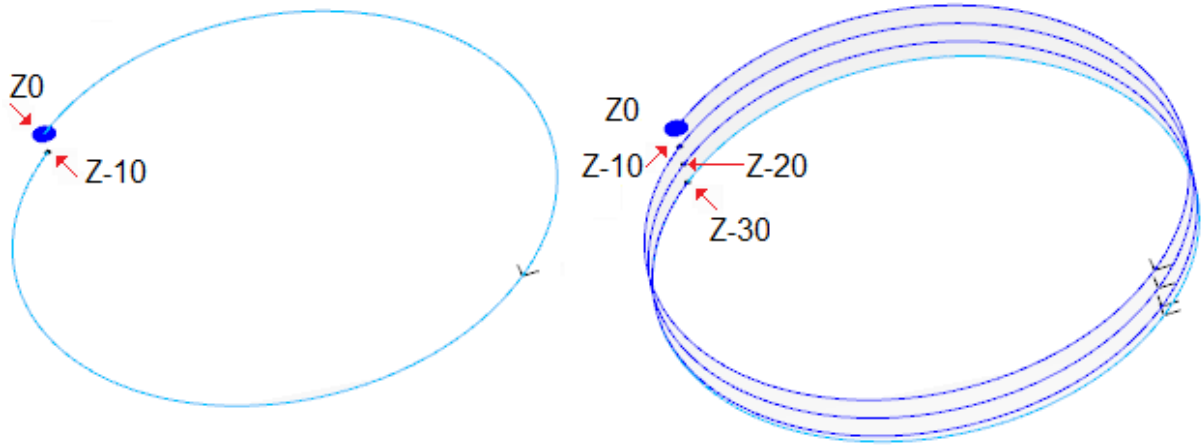
This option modifies the current profile by inserting later repetitions until a final assigned depth is reached or approached with possible execution of a final pass at a constant depth. The **Repeat profile**  is available in the **Construction**group of the< text style="font-weight:bold;">Tools tab.
This command can be also available for the *Essential* functionality.
The tool is enabled if the current working belongs to a profile and works on simple profiles only.
Furthermore, the profile must be closed on XY plane and perform a depth variation between its beginning and its end.
The main difference with **Apply Z feed** is that the profile repetition are not determined by the insertion of vertical segments, but by the structure itself of the original profile.



- **Initial Z**: shows the initial depth read by the profile. The field cannot be edited.
- **Final Z**: shows the depth read by the profile in correspondence with the last programmed segment (final depth). The field cannot be edited
- **Z coordinate**: sets the final depth required in an application of the profile repetitions: It is a limit value that cannot actually be achieved, if the change in depth of the original profile does not allow it. The value must be outside the range between the **initial Z** and **final Z**.
- **End with a constant Z pass**: the option is significant, if the profile does not perform arcs in a plane different from xy. If selected, it ends the modification of the profile by adding a pass with a constant depth. If the value of **Z coordinate** is equal to **Final Z**: the tool modifies the by adding the only pass with a constant depth.

In the application of the tool any changes of depths, programmed in the middle of the profile are not considered.

If the setup of the original profile some segments of entry and/or exit are set, these remain directly assigned in the setup and are not considered in the application of the tool.
On the added passes, all the changes of tool compensation in the path (breaks, suspensions and resumption and/or changes of side) are reset.
In the figure below you will see an example of typical application of the tool:



On the left: the original profile, assigned with a circle carried out with a Z depth from 0.0 to - 10.0.
On the right: the profile as modified by the tool by the application of the following assignments:

- Z coordinate= 30.0

In the figure the various depth positions are shown, calculated for each of the repetitions:
- the first additional repetition begins from Z=10.0 and ends to Z=20.0.
- the second additional repetition begins from Z=20.0 and ends to Z=30.0.
- In the figure also a last repetition is added, carried out with a constant depth Z=30.0.

PROFESSIONAL

It is possible to generate profiles modified with repetition also in the form of complex working, by recalling the **Programmed tools** in the list of the workings. In the group of TOOLS select the STOOL: PROFILE REPETITION;
- the **Workings** field sets the names assigned to workings programmed before that correspond to the original profiles.

The profiles may also result from the application of other complex codes and from the development of the working STOOL: PROFILE REPETITION is only for the modified profiles and does not include the original profiles. Possible workings that cannot be used for the functionality required (for example logical workings or workings of the points or complex workings that cannot be exploded) are ignored.
Beyond the tool, the working allows to assign the development axis on one of the three coordinated axes of the face.

The working sets:
- Typical parameters of a complex working (see the preceding discussion of a generic Subroutine code):
  - **Qx,Qy Zp**: initial positioning coordinates of the developed workings;
  - **..**
  - **Working properties**: sets the properties given to the working.

- Specific parameters of the working functionality, with a meaning analogue to the fields defined in the window of the tool.
  - **Development axis**: selects the development axis on one of the three coordinated axes of the face (Z, X, Y)
  - **Assign the number of passes**: if selected, the option requires assigning the number of passes otherwise, it requires assigning the final depth at which the last advancement has been executed.
  - **Final position**: sets the final depth required along the development axis. The assignment is ignored if the previous option is selected.
  - **Number of passes**: number of the required passes (valid setting: 0-1000). The assignment is significant, if the option **Assign the number of passes** is selected
  - **End with a constant Z pass**: if selected, it ends the modification of the profile by adding a constant depth pass. If you set a null **Number of passes** or a **Final position** that is not distinguished from the position of the final development of the original profile, only the constant depth pass is added to the profile.

  The execution of the final pass always depends on the check on the original profile, with relation to the selected **Development axis**:
  - if Z axis: it must not develop arcs in a plane different from xy;
  - if X axis: it must not develop arcs in a plane different from yz;
  - if Y axis: it must not develop arcs in a plane different from xz.

  The application of the programmed tool with a modification of the original profile depends of the check on the original profile, with relation to the selected **Development axis**:
  - If Z axis: the profile must be closed on the XY plane and develop a modification along Z between the beginning and the end.
  - If X axis: the profile must be closed on the YZ plane and develop a modification along X between the beginning and the end.
  - If Y axis: the profile must be closed on the XZ plane and develop a modification along Y between the beginning and the end.

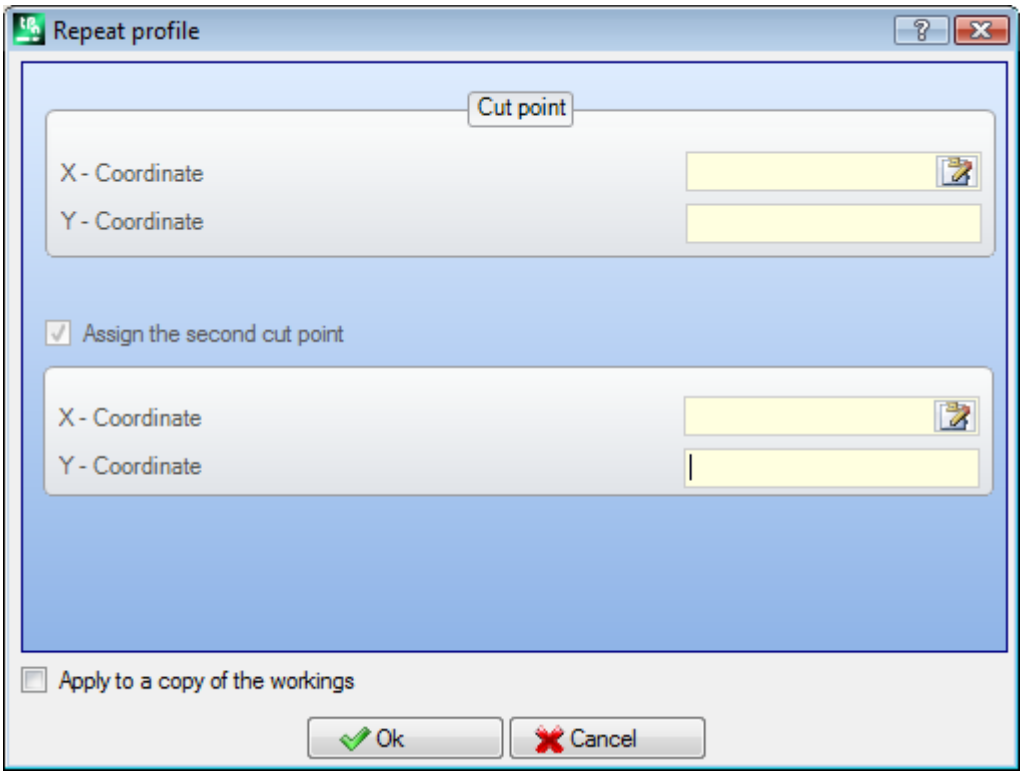
The main advantage offered by the use of the working STOOL: PROFILE REPETITION consists on the fact that the compensated profiles adapt to the modifications of the original profiles, besides the fact that they can totally work on more than one profile, and also on the complex ones.

Duplicate profile

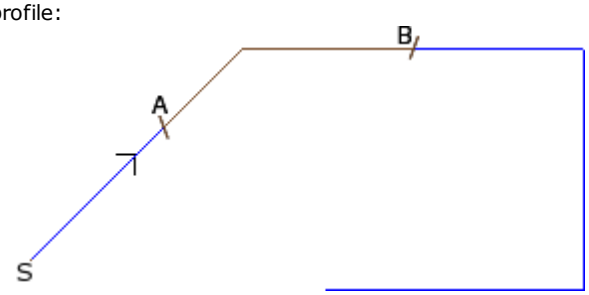
It repeats part of the current profile. The **Duplicate profile**  command is available in the group **Constructions** of the **Tools** tab. The tool is enabled, if the current working belongs to a profile.

The portion of the profile to be repeated is defined between two cut points.

The profile is obtained as a geometric repetition of the segments between the two cut points and it is opened with a copy of the original setup or of the reference setup (as assigned in **Customize -> Technology -> Default Technology** of the **Application** menu).



The coordinates or the cut points can be inserted in the edit fields or in the graphic area with the mouse (clicking the icon).
By selecting the option *Apply to a copy of the workings* the tool is applied to a copy of the workings and does not modify the original lines.
In the figure an example of profile:



- (S) indicates the profile starting point;
- the arrow indicates a counterclockwise direction
- the profile is not closed.

On the profile the two cut points (A) and (B) are indicated. The two points can lie on the same segment or on different segments.
The part of profile between the two points, always following the sense of the original direction, is extracted from the profile.
If the tool needs to work on the current profile, the parts of the profile are eliminated
- from (S) to the point (A);
- from (B) to the end of the profile.

If the tool needs to work on a copy of the current profile, a new profile is added and assigned from the point (A) to the point (B).
The compensated profile is generated with reduction to elementary profile codes and numeric only assignments; it calculates in numeric format each parametrization used while assigning the original profile.

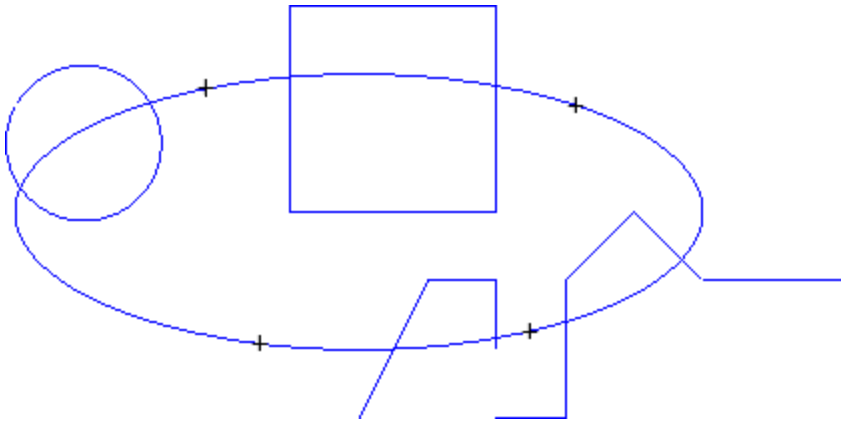
Cut profiles

PROFESSIONAL

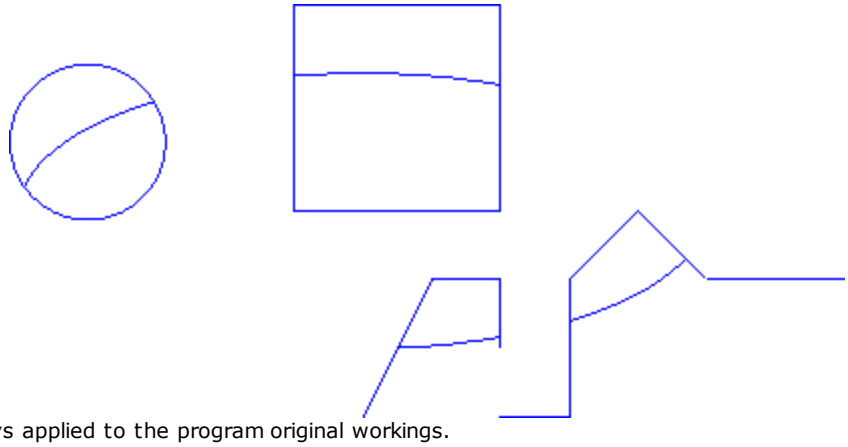
This tool allows the user to cut profile parts called *cutting edge*, selected on the intersection of profiles.

The **Cut profiles** command  is available in the group **Constructions** of the **Tools** tab.
In the case of piece-face the tool is only enabled if the 2D or the Box-View face is active and it works on profiles applied to the face in current view only.

The profiles that identify the cutting edges are selected profiles (if any), or all profiles selected from the face.
The cutting edges are assigned with the mouse, by following the instructions provided in the Commands area.
ATTENTION: arcs on non-XY plane and (L24) path elements are excluded from the calculation of the intersection points of profiles.
The Figure shows a program made of intersecting profiles. The 4 crosses identify the cutting edges:



After the application of the tool the program appears like in the Figure

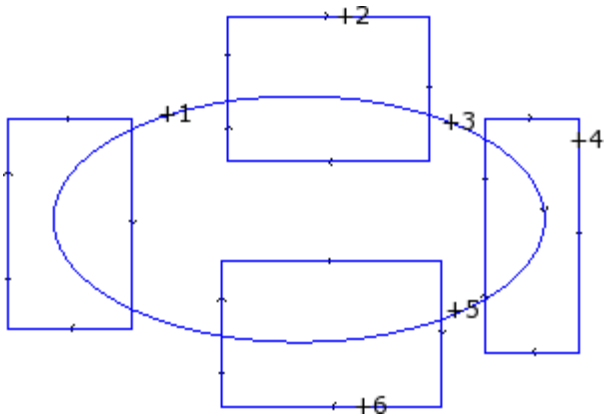


The tool is always applied to the program original workings.

Profile Building

PROFESSIONAL

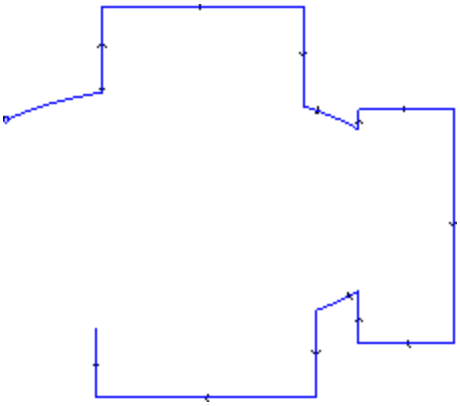
This tool allows building a new profile by selecting one or more programmed segments of profile. The selected segment must have a point of intersection with the previous segment of profile. The **Profile building**  commandis available in the **Constructions** group of the **Tools** tab.
In the case of piece-face the tool is only enabled if the 2D or the Box-View face is active and it works on profiles applied to the face in current view only.
The command selection requires the new profile technology data to be set: setup working and related technological assignments. The segments belonging to the new profile are assigned by the mouse, following the instructions provided in the Commands area.
ATTENTION: arcs on non-XY plane and (L24) path elements are excluded from the calculation of the intersection points of profiles.
Example:



The Figure shows a program made of intersecting profiles. The 6 crosses identify the indicated parts for the construction of a new profile. Crosses are numbered and indicate the order by which the selected parts are added to the new profile:

- cross 1 marks the starting point. The segment of profile which is nearest to the position clicked with the mouse is searched;
- cross 2 chooses how to continue the profile. The segment of profile which is nearest to the position clicked with the mouse and which continues geometrically the segment already selected as segment (1) is searched. The geometric continuity can also determine the inversion of segment (1) and/or segment (2), with respect to the direction of execution of original profiles;
- cross 3 chooses how to continue the profile. The segment of profile which is nearest to the position clicked with the mouse and which continues geometrically the segment already selected as segment (2) is searched. Now the geometric continuity can cause the inversion of the only segment (3), with respect to the direction of execution of original profiles;
- ..
- up to cross 6.

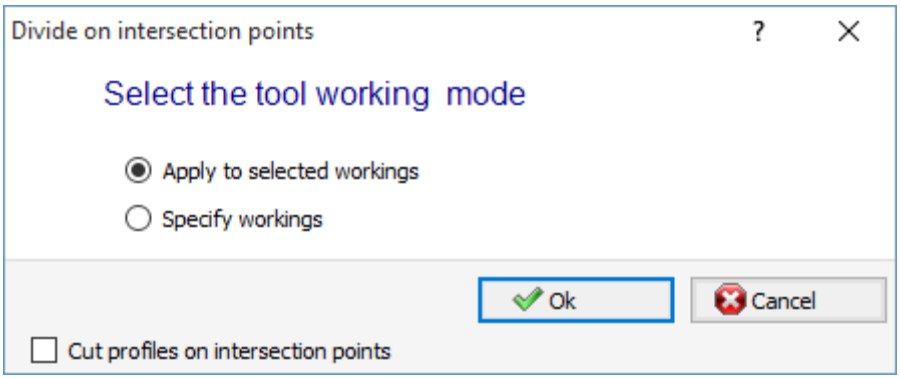
After the selections have been finished, confirm the command by means of the **[Enter]** button or selecting the option with the mouse from the local menu (opened with the right mouse button). At this point the acquired positions are processed and a new profile is added to the face program, without modifying the original profiles.
In next Figure we can see the profile built by following the above-mentioned instructions:



Divide on intersection points

PROFESSIONAL

This tool allows finding the intersection points of the profiles and dividing the individual segments in the same points. The command **Divide on intersection points** is available in the group **Constructions** of the **Tools** tab.
In the case of piece-face the tool works only on the profiles applied to the face in current view.
If some profiles are selected, this window appears, as follows:



Select one of the two suggested working modes:
- **Apply to selected workings**: this tool is applied to the selections.
- **Specify workings**: closing the window and confirming, the user should provide indication interactively by the mouse, in accordance to the instructions in the Command area.

- **Cut profiles on intersection points**: select to divide each single segment in distinguished profiles. Otherwise, the profile segment on which the cut point falls is divided in two, but the profile remains one.

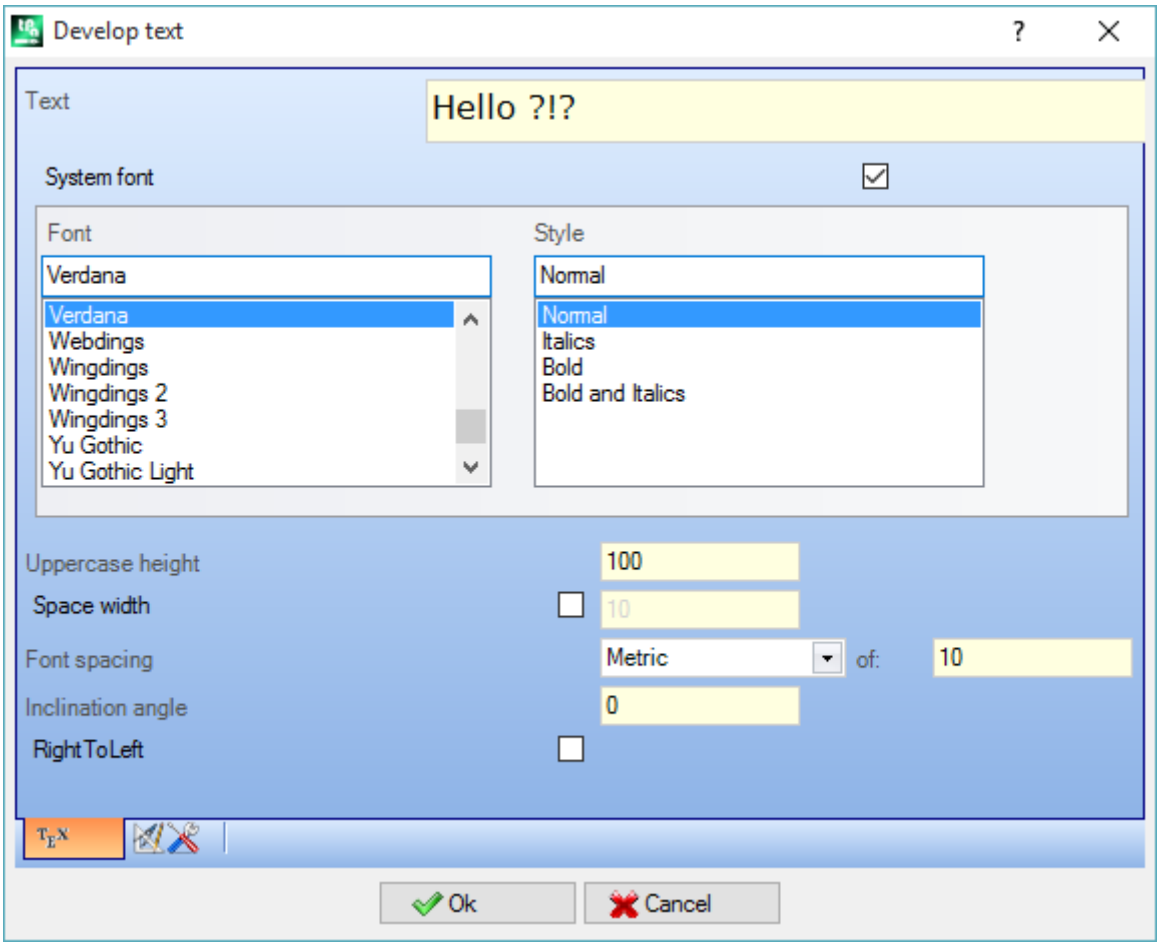
If there are no selected profiles, a selection window of the cut options appears and the **Specify workings selection is automatically selected**.

The tool is always applied to the original workings of the program and can also work on one only profile.

Text generation

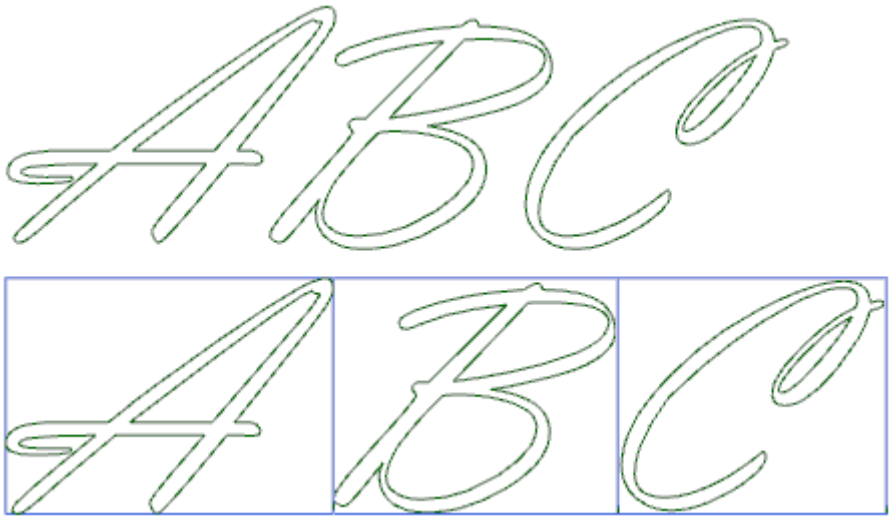
PROFESSIONAL

This tool allows the user to enter text into the face program, directly in form of profiles. The **Develop text** command is available in the **Constructions** of the **Tools** tab.
Opening a window can take several seconds to search and create the list of the available fonts.



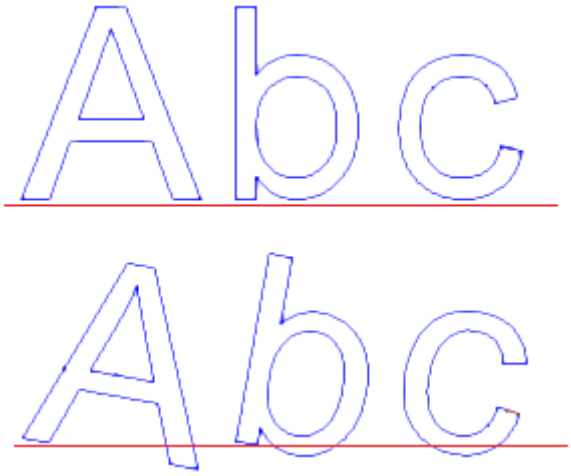
- **Text**: text to be entered
- **Font**: kind of Font. The list makes all the installed characters available for which it is possible to select at least one of the styles (Regular, Italics, Bold, Bold and Italics).
- **Style**: available style for text formatting (Regular, Italics, Bold, Bold and Italics).
- **Upper-case height**: it sets the letter A height in the piece unit of measure.
- **Space width**: it sets the width of the spaces, if available in Text.
  - Select the check box to make the field editable. The value set here is assigned as a space character width; it is also possible to reset.
  - If the check box is not selected, the width used for the space character is that defined for the selected font (it corresponds to the width of the character (–)).
- **Font spacing**: assigns the spacing of the single characters of the writing. This setting is used if the **Automatic distribution** option is not selected, available on the second page. The list shows two entries to select the mode for the assignment of spacing between two next characters in the list:
  - **Metric**: the space is determined by the rules defined for each single character of the font.
  - **Geometric**: the space is determined by the overall rectangle of each single character.

In the figure below you will see an example of a text developed for the two cases of spacing cases, with a value set for the font spacing =0.0:



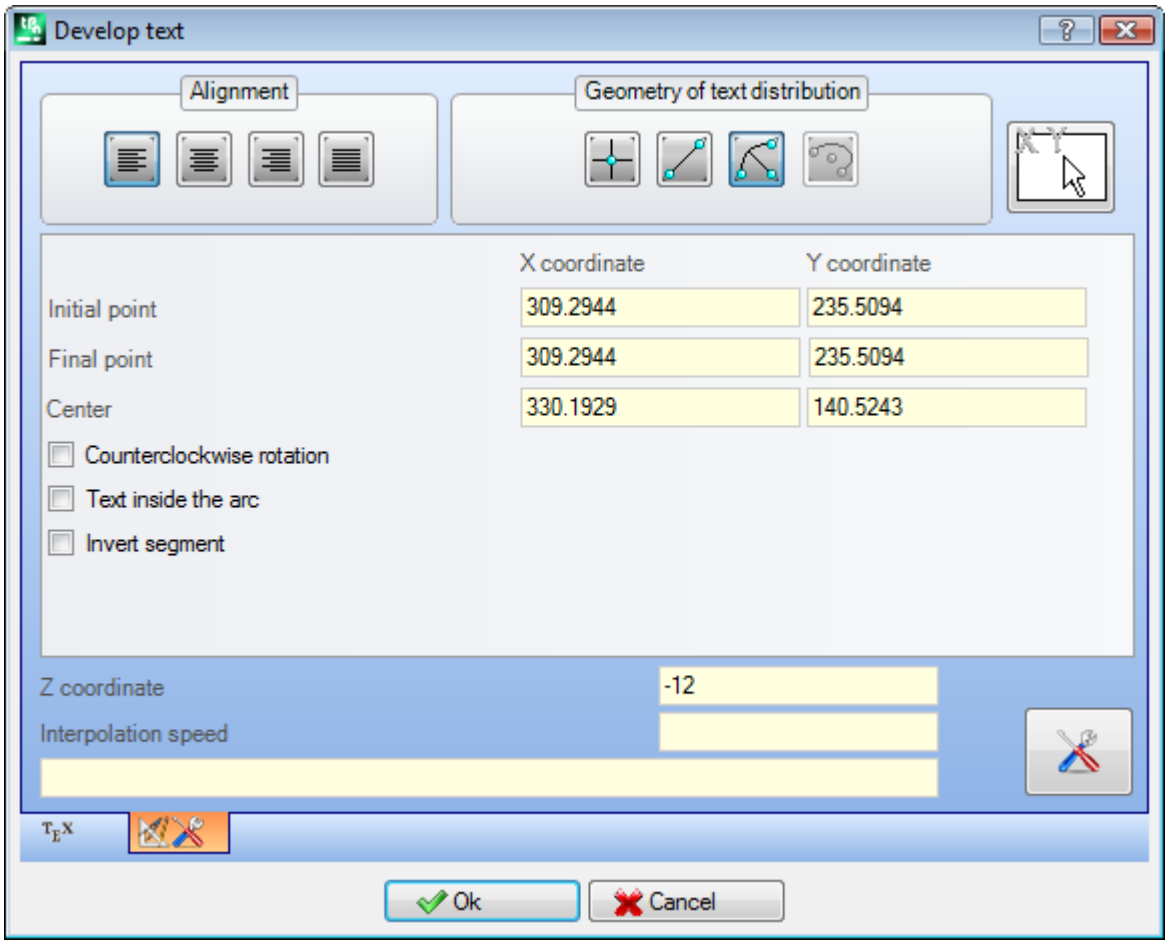
The text developed on the top applies a *metric* spacing. The text developed below applies a *geometric* spacing: around each character its own overall rectangle is shown.

- **Inclination angle**: sets the inclination of each character with respect to the base development line of the text. The field is set in degrees (°) and tenths of degrees; the value by default is 0. A positive value bends the characters to the right, in accordance to the cursive style. In the following picture the same text is developed with a different inclination (0.0 and 10.0):

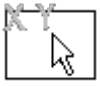


- **RightToLeft**: the selection is available for the compose layouts from rights to left, for example for the Arabic or the Jewish language. The selection inverts the order of the character in the text. If you had applied the selection to the example in the figure, the developed text would have been "CBA".

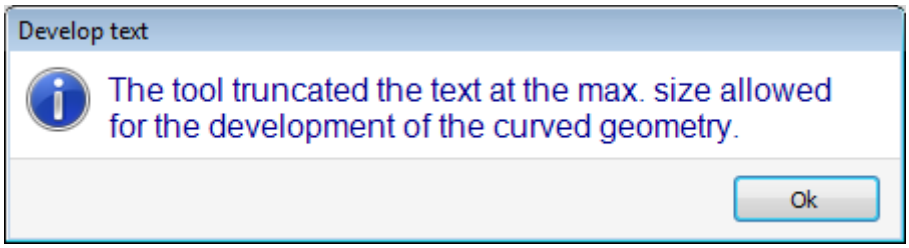
Click the icon  to activate the tab of the text distribution and of the technological assignments.



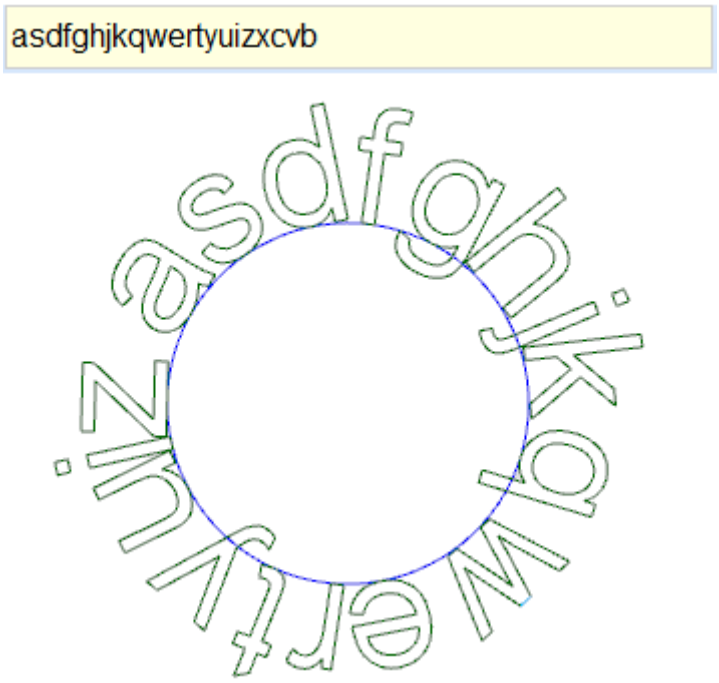
- **Alignment**: four selections are available:
  - **Left align**: is the selection by default, always employable.
    The remaining three selections are applied only if the text is distributed along a geometric segment of the line, arc or conic:
  - **Centred alignment**: the text is centred along the segment
  - **Right align**: the text is aligned starting from the final part of the segment
  - **Text automatic distribution**: aligns the text both at the initial and at the end point of the segment and adds the necessary space among the characters in order to obtain an equal distribution of the text. The selection is not significant, if the text is made by only one character.
- **Geometry of text distribution**: the text distribution can be assigned with reference to:
  - **point**: with the (X, Y) writing start position and the inclination angle of the writing. This selection always determines the application of **Left align**.
  - **linear geometry**: with an initial and a final position (X, Y) of the segment. The **Invert the segment** allows the user to apply the segment geometry inverted in the initial and end points.
  - **arc geometry**: with an initial and a final position (X, Y) of the arc, centre and rotation. The element can find an arc or a circle. Following options are available:
    - **Text inside the arc**: if selected, this option allows the application of the text inside the arc.
    - **Invert the segment**: if selected, this option allows the user to apply the segment geometry inverted
    - **Arc of conic**: With start and end position (X, Y) of the arc of conic, centre, extreme point along one of the axes, and rotation. The element can identify a conic of arc or a full conic, with elliptic or oval development. The following options are available:
      - **Text inside the arc**: if selected, this option allows the application of the text inside the arc
      - **Invert segment**: if selected, this option allows the user to apply the segment geometry inverted


The bitmap allow the mouse acquisition of the text distribution programmed element: point, linear segment or arc. The fields in the *Geometry of text distribution* area are updated according to the typology and the geometry of the selected segment. These settings can be changed according to the requirements, except for the geometric element of the conic, for which the interactive discovery only is possible.

In the case of the distribution on an arc of a circle or of a conic, the text that is really developed cannot exceed the length of the closed figure. A message reports if the text has been truncated:



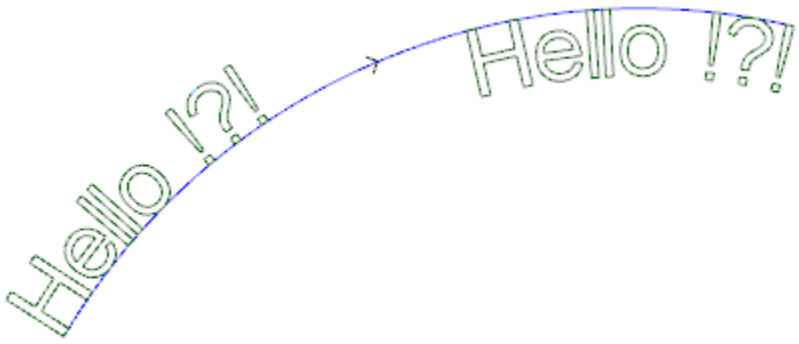
in the following figure there is an example where the tool has truncated the text:



The technological attributions of the inserted profiles are set by:

- Technology button assigning the setup of the profiles (working code, technology, working property).
- **Z coordinate**: execution depth of the profiles
- **Interpolation speed**: assigned speed for the execution of the profile themselves.

For example:



The Figure above shows an example of generation of 2 equal writings, with selection of no automatic text distribution along a circular segment in clockwise direction.

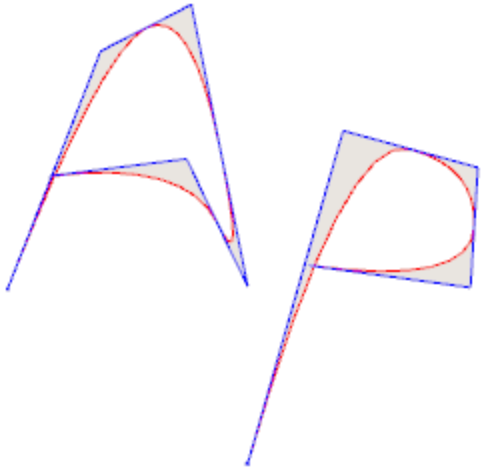
- The left writing is generated without any additional selection.
- The right writing is generated with selection of alignment on the right and text inside the arc.

If some custom assigned font are available, the font selection page to use shows also the **System font** selection option:
- select to use one of the installed system fonts (as above),
- keep the option unselected to use one of the custom fonts. In this case, the selection of the font *Style* is not available; it is possible to select the option of B-quadratic B-spline curve processing for each of the inserted profiles (see: Generate spline from polyline tool)

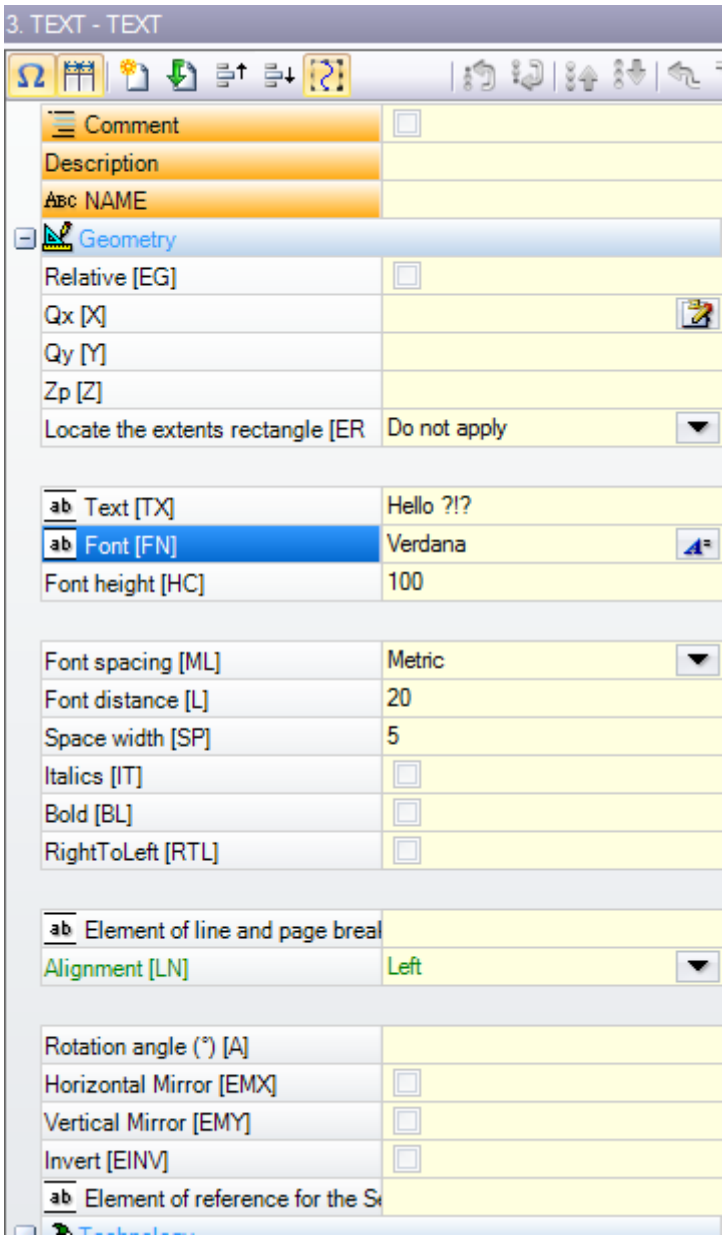
All the other settings are interpreted with the same meaning for the generic case of font system usage.

Files assigning the custom fonts are stored in the *TPACADCFG\CUSTOM\DBFONTS* folder with *fcad* extension. A custom font defines a more or less complete set of characters (upper/lower cases, digits, punctuation marks, …)
A character is described by one or more profile, each marked as a polyline.
For the detailed description of a custom font file format see dedicated documentation.
Custom font is selected in the list of the name of the files.
The figure shows the example of writing by means of a hypothetical custom font for the "Ap" writing.
- The external contour lines can match the original profiles
- The internal contour lines can match the profiles processed with a spline curve.

The actual modification of the profiles according the spline curve logic depends on the original assignment of the profiles for each font character.
The grey fields among the curves of each letter are displayed in the figure only highlight the distance between the two curves.



It is also possible to enter texts by recalling dedicated macros in the list of workings.
Example:
in the group of the SPECIAL MILLING CUTTERS select the TEXT working:



It is a complex working carried out by means of a macro. It allows the assignment of:

- typical parameters of a complex working (see what has been said about a generic code of Subroutine):
  - **Qx,Qy Zp**: initial coordinates of the text and depth
  - **Locate the extents rectangle**: selection field to place the overall rectangle of the writing
  - **Horizontal Mirror** and **Vertical Mirror**: it enables the symmetry required
  - **Rotation angle (°)**: it sets the inclination angle of the text
  - **Invert**: it enables the inversion during the execution of the profiles
  - **Working properties**: it sets the properties attributed to the workings
- Assignment of the technology:
  - **Element of reference for the Setup**: sets the Name of a setup working that assigns the profile technology generated by the working (the field is available only if the Name of workings property is managed).
    The working is searched before the current working, it must correspond to a Setup in Cartesian programming mode and the compilation must not have generated errors. Furthermore, the **Comment** field of the working must not be selected and, if in the face-piece, the working must be applied to the same face. In the event of multiple correspondences (more than a Setup programmed with the same name) the last found is selected, that is the nearest to the TEXT working.
    As an alternative the field can assign the name (parameter) of a Global technology (see: **Customize -> Technology -> Default codes**). In this case no accessory programming is required.
    To the property settings of the workings are applied the same criteria used in the programming of all of the complex codes, which normally correspond to the propagation of non-null values of the properties set for the complex code (in our case: the TEXT working).
    Let us see in practice: the external Setup has the L level = 2:

- if the TEXT working has L level = 0, the setup of all the profiles keeps the value of the L level = 2
- if the TEXT working has L level = 1, the setup of all the profiles shall have the value of the L level = 1.

An exception is made for the B field (construct), in view of the fact that it is usual to assign the external Setup of a construct, in such a way as to exclude it from the execution of the piece. In this case: the setup of the profiles generated by the TEXT working can be a construct setup only if TEXT is programmed as a construct setup.

As an alternative, it is possible to assign the technology of the profiles by setting the parameters grouped in the nodes:

- **Technology, Advanced technology data**: the two nodes show the parameters to choose the tool, the tool compensation, the speed, …

ATTENTION: if it is not possible to use an external setup (no correspondence is found valid for the entry **Element of reference for the Setup**), a *Warning* appears and the technology of the profiles is always assigned by means of the settings for the **Technology, Advanced technology data** nodes.

- Specific parameters of the working function:
  - **Text**: text to be entered
  - **Font**: type of characters that must be applied to the text (the list makes available all the installed characters for which the regular, italics and bold styles can be selected).
  - **Font height**: it sets the height of the A character (in unit of measure of the piece)
  - **Font spacing**: assigns the spacing mode between the single characters of the text (selections: Metric, Geometric)
  - **Font distance**: distance of following characters
  - **Space width**: it sets the width given to spaces (if available in Text) available in the text writing. Set a negative value to use the font width.
  - **Inclination angle**: sets the inclination of each character with respect to the base development line of the text. The field is set in degrees (°) and tenth of degrees.
  - **Italic**: it enables the Italics style
  - **Bold**: it enables the Bold style
  - **RightToLeft**: select for the cases of text from right to left (inverts the position of the text characters)
  - **Element of line and page breaks**: set the Name of the working assigning the geometry for the text distribution. The processing is searched before the current working and must correspond to a linear segment or an arc of a circle or an arc of a conic: furthermore, the compilation of the element must not have generated any error; also: it cannot have selected the **Comment** field and, if in face-piece, it must be applied to the same face of the current working (in our case: TEXT). In the event of multiple correspondences (more workings with the same name) the last found is selected, that is the nearest to the TEXT working.
  - **Alignment**: select the alignment mode of the text in the four entries of the list. If a geometric distribution element is not assigned (linear segment or arc), the alignment of the Left will be always applied.

In a similar way the GEOMETRIC TEXT working is available, in which the custom fonts assigned can be selected.

Generate spline from polyline

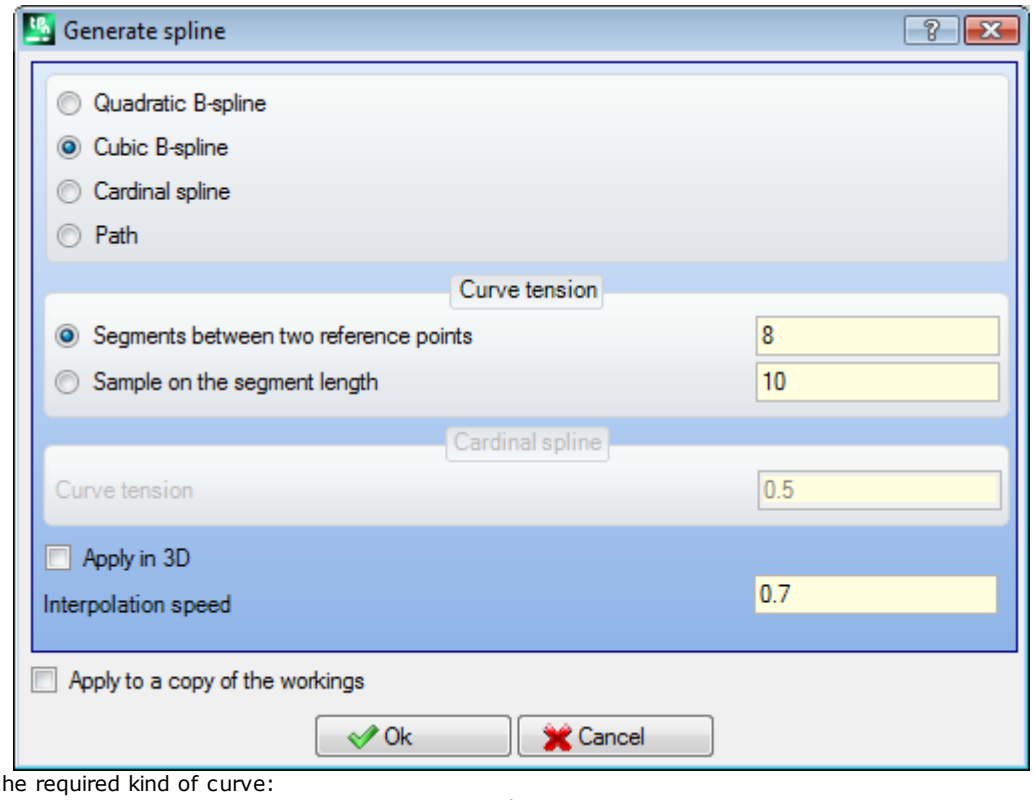


The command **Generate spline** from polyline  is available in the **Constructions** group of the **Tool** tab.

This tool consists of a profile construction in correspondence of a programmed polyline.
For each identified profile the tool uses its vertices as reference points (control vertices or reference point) in order to generate a curve that interpolates the control vertices. The theoretical calculated curve is then sampled in linear segments; as a result we obtain a polyline with the following general features:

- the first point coincides with the start point of the original profile
- the last point coincides with the start point of the original profile
- the passage of the curve from the other intermediate points of the original profile depends on the kind selected curve
- the theoretical curve is always continuous, without cusps.

Arcs and L24 Path element of the original profile are carried out as line segments.
Possible circles are deleted from the profile for the evaluation of reference points. Moreover the original profile cannot assign arcs in a plane different from the xy plane.



Select the required kind of curve:

- **Quadratic B-spline**: the curves are calculated by Bézier quadratic curve solution (at least 3 reference points are needed).
- **Cubic B-spline**: the curves are calculated by cubic Bézier curve solution (at least 4 reference points are needed).
- **Cardinal spline**: the curves are calculated by a particular cubic Hermite curve solution, called cardinal Spline (at least 3 reference points are needed).
- **Path**: the curves are calculated by a solution of each single curve segment by means of an operational code called Path, otherwise called as: L24. This option cannot be available in the window. Selecting the *Path* option, only the field **Interpolation speed** can be set.

Selecting the kind of sampling

- **Segments between two reference points**: number of linear segments between two reference points (values from 8 to 100 are accepted). The value assigns the curve sampling criterion. Assuming that the original profile is assigned with 5 linear segments and that the field value is 8, the generated curve will have up to 8 * (5-2) = 24 linear segments (segments whose length is less than epsilon * 5.0, or less than the length set in the next field are not generated);
- **Sample on the segment length**: select to apply a sampling based on the length of the segments in which the longest segment of the original profile has to be split. In this case, the value set in the previous field is ignored. Assuming the longest segment of the original profile is 70 mm and that the value here set has value 0.5, be automatically calculated a number of samplings equal to 70/0.5=140 sampled segments per each segment, where the minimum length of the generated segments is not less than *epsilon* * 10.0.
  Although no sampling on the segment length is required, the value set (in the example: 0.5) is always used as minimum length of the actually sampled segments, to which the minimum applied value is *epsilon* * 5.0 and the maximum one is *epsilon* * 100.0.
- **Curve tension**: value of the curve tension in case of *Cardinal Spline curve*. The field recognize value between 0.0 and 1.0 (an invalid setting is taken back to the interval):
  - 1.0 corresponds to the maximum tension: the calculated curve corresponds to the original profile, broken on the indicated segments;
  - 0.0 corresponds to the minimum tension: the calculated curve corresponds to the situation of maximum deviation from the original profile;
- **Apply in 3D**: select to enable the curve solution according to the depth coordinate, so a curve in the space is generated. If the field is not selected, the generated curve set the Z coordinate on the setup only;
- **Interpolation speed**: it sets the execution speed of the spline curve.

The generated profiles are opened with a copy of the original setup, if available, or with a copy of the reference setup (as assigned in **Customize -> Technology -> Default codes** from the Application menu).
By selecting the option *Apply to a copy of the workings* the tool is applied to a copy of the workings and does not modify the original lines.

Let us see more specifically the features of the produced curves.
The first two cases - Bézier curves - have some common features:

- as already told: the extreme point of the curve coincide with those of the original profile.
- the beginning of the curve is tangent to the first side of the original profile.
- the end of the curve is tangent to the last side of the original profile.
- the curve never passes through the intermediate points of the original profile.

In the computer graphs, the Bézier curves are always used as an examples to model chamfered curves.
A typical example are the system TrueType fonts that are made by quadratic Bézier curves.
In the case of *Cardinal spline*:

- as already told: the extreme point of the curve coincide with those of the original profile.
- the curve passes through all the intermediate points of the original profile.
- the portion of the curve between two original points can be found outside the domain of the original segment.
- In the case of original closed polyline, the development is invariant with respect to the initial/end point of the polyline.

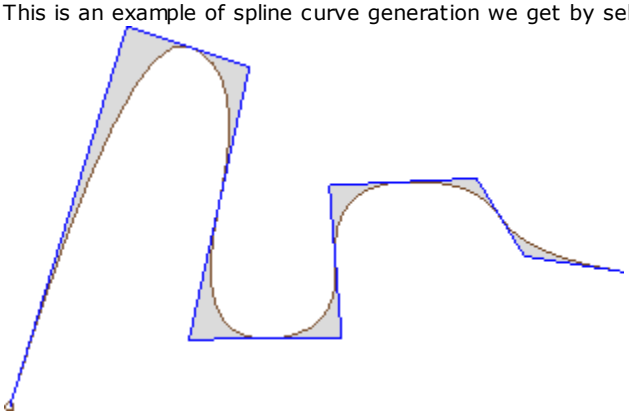
In the case of *Path*:

- the extreme point of the curve coincide with those of the original profile.
- the curve passes through all the intermediate points of the original profile.
- The portion of the curve between two original points can be found outside the domain of the original segment.
- the here generated curve is continuous, without cusps, but only because of specific choice.

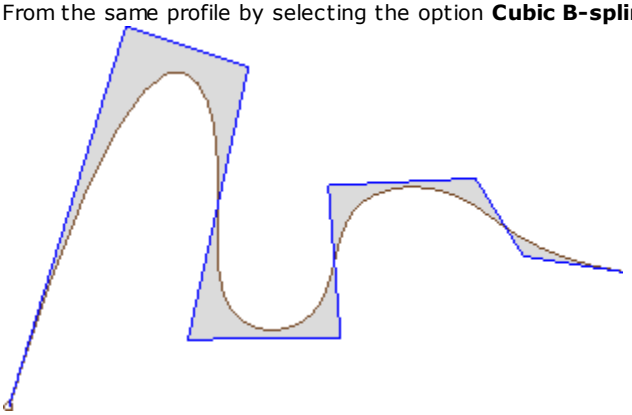
The Path has a different meaning from the other selection of spline curves, because it is associated to a specific working (L24) that can be used apart from the application of the here examined tool.
In the *Path* curve that is automatically generated by a reference polyline:

- the first element begins with a tangent line from the first polyline segment.
- ends each L24 element with tangent line assigned on the next polyline segment.
- begins each following element in tangent continuity with the previous one.

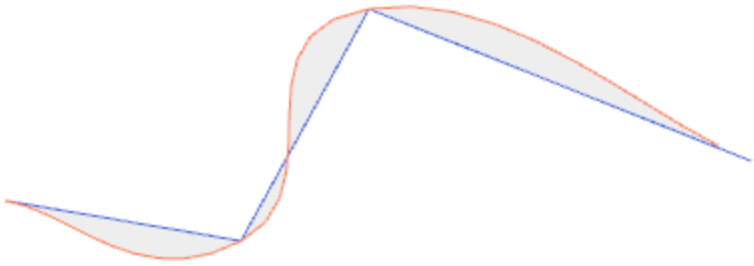
The generated path can be modified according to the requirements.
This is an example of spline curve generation we get by selecting the **Quadratic B-spline** option:



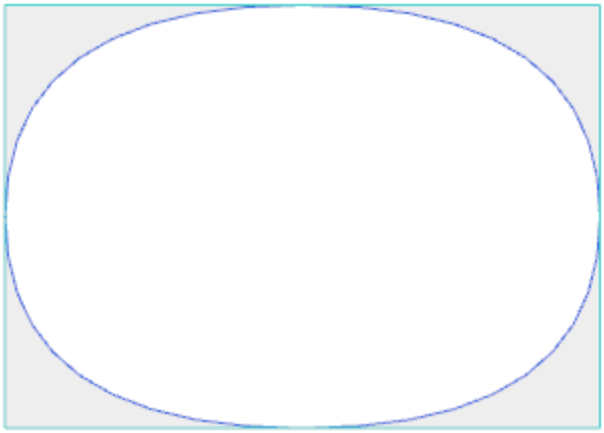
From the same profile by selecting the option **Cubic B-spline** we get the curve:



This is an example of curve generation we get by selecting the option **Cardinal Spline**:



As last example let us see the generation of a **B-spline quadratic** curve applied to a rectangle/square, inserted for example by a command from the **Draw** menu. The built curve corresponds to a generally elliptic path:



The grey fields between the original curve and the spline curve are displayed in the figure only to highlight the distance between the two curves.

It is possible to generate spline curves also in the form of complex working, recalling the *Programmed tools* working in the list of workings. In the group of TOOLS select the STOOL working, SPLINE:
- the **Workings** field set the names assigned to previously programmed working that correspond to the original profiles.

The profiles may be the result of the application of complex codes and development of working STOOL: SPLINE is only for the modified profiles and it does not include the original profiles. Possible workings that cannot be used for the function required (for example: point or logical workings or complex workings that are not expandable) are ignored.

The working sets:
- Typical parameters of the complex working (see what has been said about a generic code of Subroutine):
  - **Qx**, **Qy Zp**: initial positioning coordinates of the developed workings.
  - **..**
  - **Working properties**: it sets the properties attributed to the working.

- Specific parameters of the working function with a meaning analogue to the fields in the tool window:
  - **Curve typology**: selection of the quadratic B-spline curve, cubic B-spline curve or cardinal Spline curve (**ATTENTION**: no Path curves).
  - **Segments between two reference points**: number of linear segments generated between two reference points (set a value between 8 and 50).
  - **Curve tension**: set the curve tension (value from 0.0 to 1.0), used in case of Cardinal Spline curve.
  - **Apply in 3D**: select to enable the curve solution also in accordance with the depth coordinate.
  - **Interpolation speed**: it sets the spline curve execution speed.

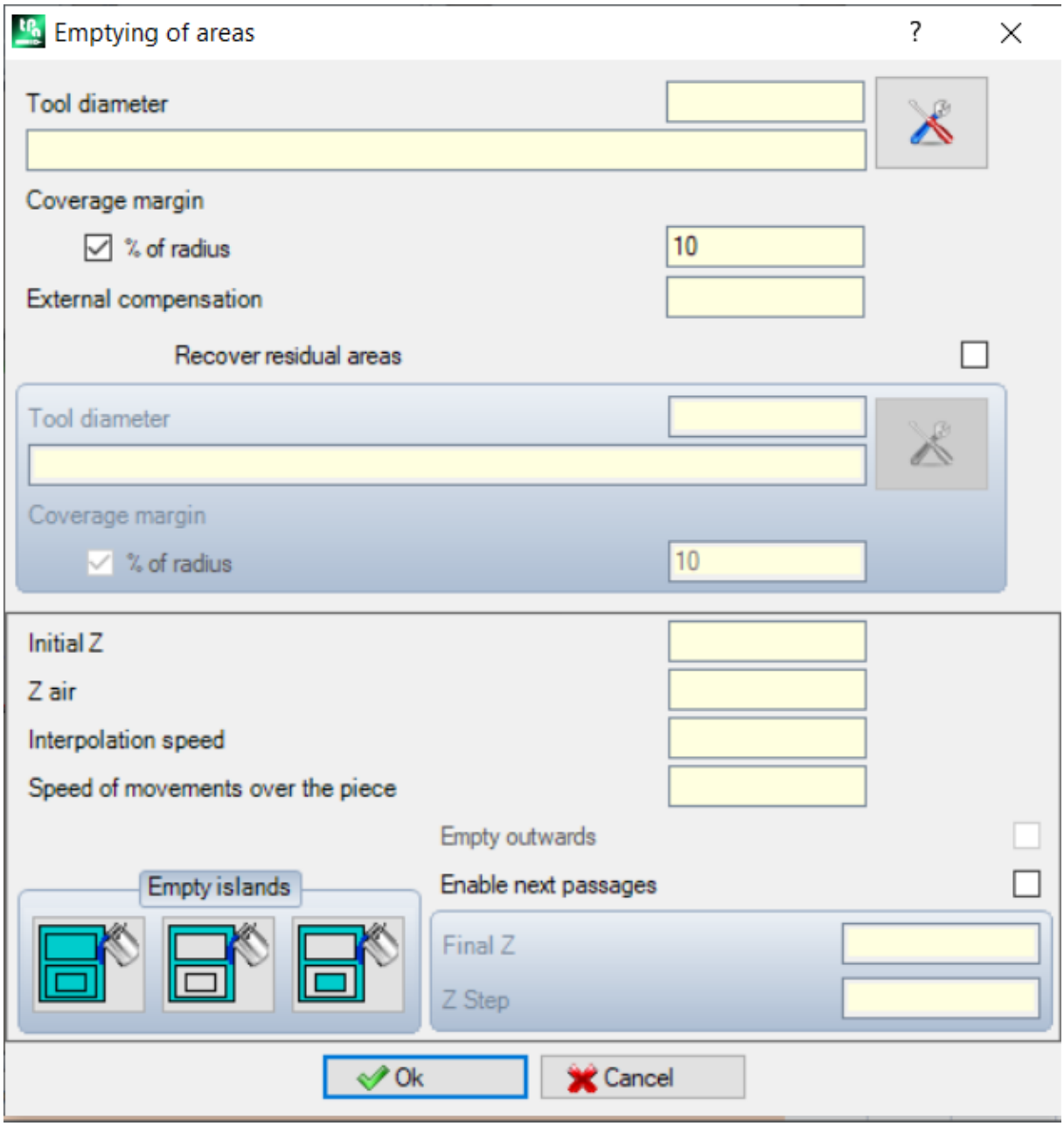
- Assignment of the technology:
  - **Element of reference for the Setup**: set the Name of a setup working or of a Global technology that assigns the profile technology generated by the working (the field is available only if the Name of workings property is managed or if there are assignments of global Technologies). Interpretation and application correspond to the information in **Text generation**, with regard to the TEXT working (please, read).

The main advantage of using the STOOL: SPLINE consists in the fact that the inserted profiles fit changes of the original profiles.

Emptying of areas

PROFESSIONAL

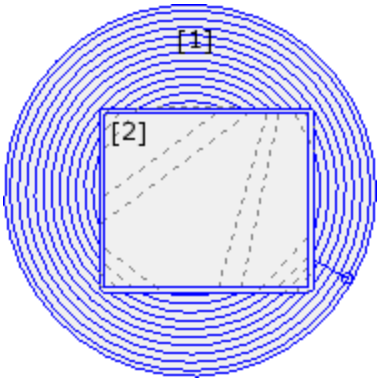
This tool allows the emptying of an area defined by a closed profile, by directly inserting emptying profiles into the face. The **Emptying of areas** command is available in the group **Constructions** of the **Tools** tab.
In the case of piece-face the tool is only enabled if the 2D or the Box-View face is active and it works on profiles applied to the face in current view only. This tool does not take into account the profiles where the **Emptying profile** parameter is active. This parameter is managed in setup workings to mark profiles generated during an emptying process.
Furthermore, the emptying process takes into account no more than 300 profiles.



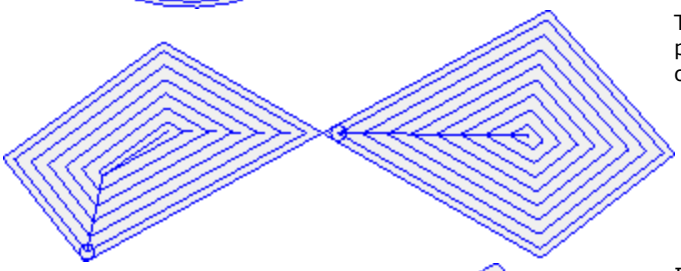
A closed area is emptied when inside an area is generated a profile that is made up by successive passes obtained with progressive deviation from the original profile, until the internal area is fully covered. If required, the emptying process guarantees the observance of internal closed areas (islands) and it tries to recover the not fully emptied areas, because of the original observance of the original area limits.
Move the current working on the profile you wish to empty and select the command from the menu.
Let us examine more carefully the fields in the window:

- **Tool diameter**: it assigns the tool diameter. By means of the Technology button  the user can choose the code and the setup working technology to be used for the emptying path: the corresponding diameter is shown in the field.
- **Coverage margin**: it shows how much the successive tool passes overlap. The field can be expressed in absolute (mm) or in **% of tool radius** set. The tool interprets a positive value as:
  - minimum value equal to 10% of the tool radius
  - minimum value equal to the tool radius
- **External compensation**: it shows how much the exit is from the programmed profile executing the first passage. The value is expressed in unit of measure (mm or inch) and subtracted from the value of the tool radius; so, a positive value greater or equal to epsilon*10 and less or equal to the compensation radius is

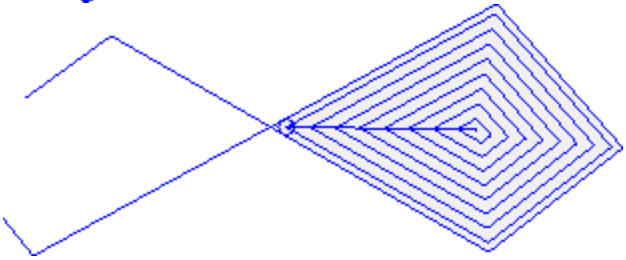
interpreted as significant (read: non-null) For example: with epsilon = 0.001 mm and the program unit of measure in [mm]:
  - the minimum value for the field is 0.01;
  - In a lower value is set, an *External Compensation* is not applied.
- **Recover residual areas**: if selected, it enables the options of the next box. It allows the setting of a second emptying technology to be used if the external areas are proved not to be totally emptied by the main technology. The technology of the recovery tool is assigned in the same way as the default one:
  - the working remains the same as that of the default technology;
  - It is always possible to set the technology fields: the tool now assigned must have an overall dimension (diameter) lower than the default one, because it must operate in areas with lower overall dimensions.
- **Initial Z**: it sets the depth coordinate executing the emptying profile(s). If any execution of passes at different depths is not required (see later), it corresponds to the depth of the first passage.
- **Z air**: it sets the safe clearance height coordinate of the tool for the additional movements over the piece.
- **Interpolation speed**: it sets the speed of the movements during the emptying process.
- **Speed of movements over the piece**: it sets the speed of the movements at the coordinates over the piece. The lowering segments from Clearance Z to the working depth are executed at the same tool entry speed, as previously set up (Technology button). If no lowering tool speed is set, the lowering segments are executed at the same movement speed over the piece.
- **Enable next passages**: it enables the repetition of the emptying cycle in several passes executed at different depths.
  - **Final Z**: this is the final depth to be reached while executing the last passage.
  - **Z Step**: this is the depth variation to be applied to the next passes.
- **Empty islands**: three selection (graphical buttons) are available from left to right:
  - **Ignore islands**: it empties the inner part of the area defined by the profile and it ignores the closed boundaries inside it.
  - **Empty outside only**: it empties the inner part of the area and it stops at the closed boundaries inside it
  - **Alternate emptying**: it empties the inner part of the area defined by the profile. If this process finds a closed outline in the inner part of it, the emptying process is broken until another closed outline in the inner part of the previous one is found and from it the emptying process starts to work again and continues in the same way.
- **Empty outwards**: if this option is active, the execution of the emptying process from the inner part of the area is required. This option can be selected only if the selection of **Ignore islands** is active.



In the figure: the emptying process of a circle (1) with rectangular island (2).
The area between the two profiles is emptied according a profile that carry on towards the inner part with successive reductions.
The emptying profile is broken in correspondence to the rectangular area: the tool goes up and moves over the piece over the island (dashed lines), going down to the working coordinate when it comes back in the area to be emptied.



The figure represents the emptying process of a profile that generates more closed areas and each of those is emptied independently.



In the figure is represented the emptying of a profile that is not closed: the check for the emptying process is performed on the existing closed areas.

It is possible to perform emptying processes also in the form of complex working, recalling the dedicated macros in the working list:

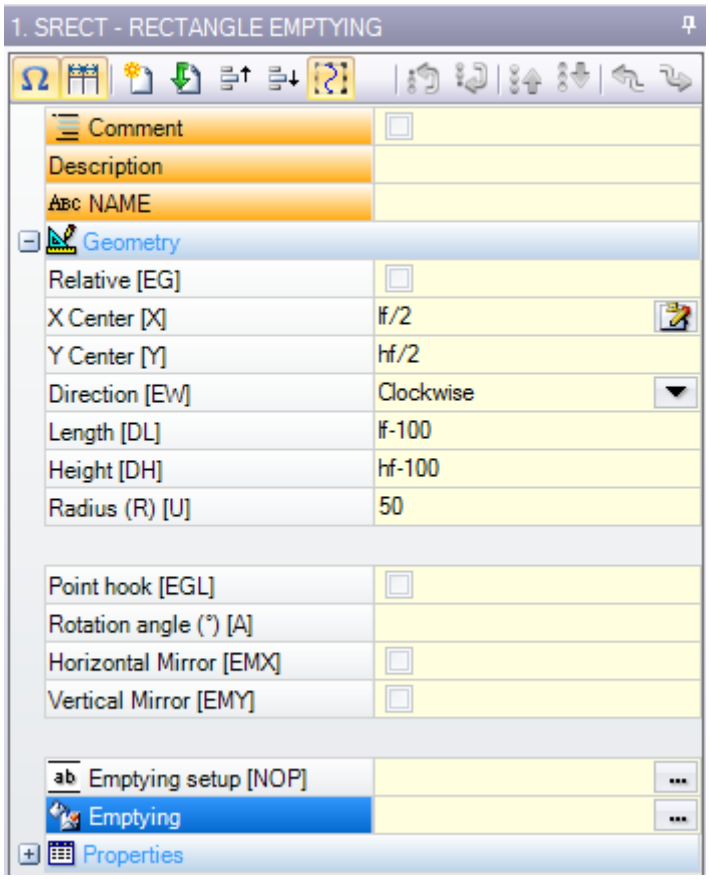
- in the group of SPECIAL MILLING CUTTER following workings are selected: RECTANGLE EMPTYING, POLYGON EMPTYING, ELLIPSE EMPTYING...; they use a complex working that assigns a particular closed geometry (rectangle, polygon, ellipse...) on the basis of the geometric parameters set and of the criteria of the emptying process of the same;

- in group of the SUBROUTINES select the EMPTY working: use an application subroutine code, conveniently arranged to assign the criteria of the emptying process for the profiles resulting from the subroutine application;

- in the TOOLS group select the STOOL: EMPTY working: the **Workings** field sets the names that are assigned to workings programmed before and that correspond to the original profiles. The profiles can also be the result of the application of complex codes and the development of the STOOL: EMPTY working is only for the modified profile(s) and does not include the original profiles. Possible workings that cannot be used for the function required (for example: point or logical workings or complex workings that are not expandable) are ignored.

In all indicated cases, the parameters assigning the emptying process are managed by a dedicated box, that derives from that examined for the **Emptying of areas** tool.

In the group of SPECIAL MILLING CUTTER, select the working RECTANGLE EMPTYING:



this is a complex working made with the help of a macro; it allows to assign:
- Typical parameters of a complex working (see the preceding discussion of a generic Subroutine code):
  - **Point hook**: this option requires hooking to a part of the profile assigned before;
  - **Horizontal Mirror and Vertical Mirror**: activates the symmetry required;
  - **Rotation angle (°)**: sets the angle of the text inclination;
  - **Working properties**: sets the properties given to the working.

- Specific parameters for the geometry of the working. In our case:
  - **X Centre, Y Centre**: centre of the rectangle;
  - **Length, Height**: dimensions of the rectangle;
  - **Radius**: radius on the corners.

- Specific parameters of the emptying functionality.
  - **Emptying setup**: it is possible to set a number or a string.

- A number assigns the code of the setup working to be assigned to the emptying profile; it can be set by direct edit or by selecting the code of the setup list shown in the window;
- a string set the Name of the setup working assigning the technology to be assigned to the emptying profile (example: "aa"). The working is searched before the current working, it must correspond to a Setup in Cartesian programming mode and the compilation of the element must not have generated any error; furthermore, it cannot have selected the **Comment** field and, if in the face-piece, it must be applied to the same face of the current working (in our case: RECTANGLE EMPTYING).
  To the property settings of the workings are applied the same criteria used in the programming of all of the complex codes, which normally correspond to the propagation of non-null values of the properties set for the complex code (in our case: the RECTANGLE EMPTYING working).
  Let us see in practice: the external Setup has the L level = 2:
  - if the RECTANGLE EMPTYING working has an L level = 0, the setup of all the profiles keeps the value of the L level = 2;
  - if the RECTANGLE EMPTYING working has an L level = 1, the setup of all the profiles shall have the value of the L level = 1.

  An exception is made for the B field (construct), in view of the fact that it is usual to assign the external Setup of a construct, in such a way as to exclude it from the execution of the piece. In this case: the setup of the profiles generated by the RECTANGLE EMPTYING working can be of construct only if RECTANGLE EMPTYING is programmed of construct.
- A string can also assign the name (parameter) of a Global technology (see: **TpaCAD Customization -> Technology -> Default codes**). In this case no accessory programming is required.
- If the field is not assigned, the setup code by default is used.

As an alternative, it is possible to assign the technology of the profiles by setting the parameters in:
- **Emptying**: The field opens a window similar to the one managed by the instrument, for the allocation of the parameters relating to the procedure of emptying: criteria of emptying (coverage margin, direction of emptying, control of the passes and of the islands) and technology.

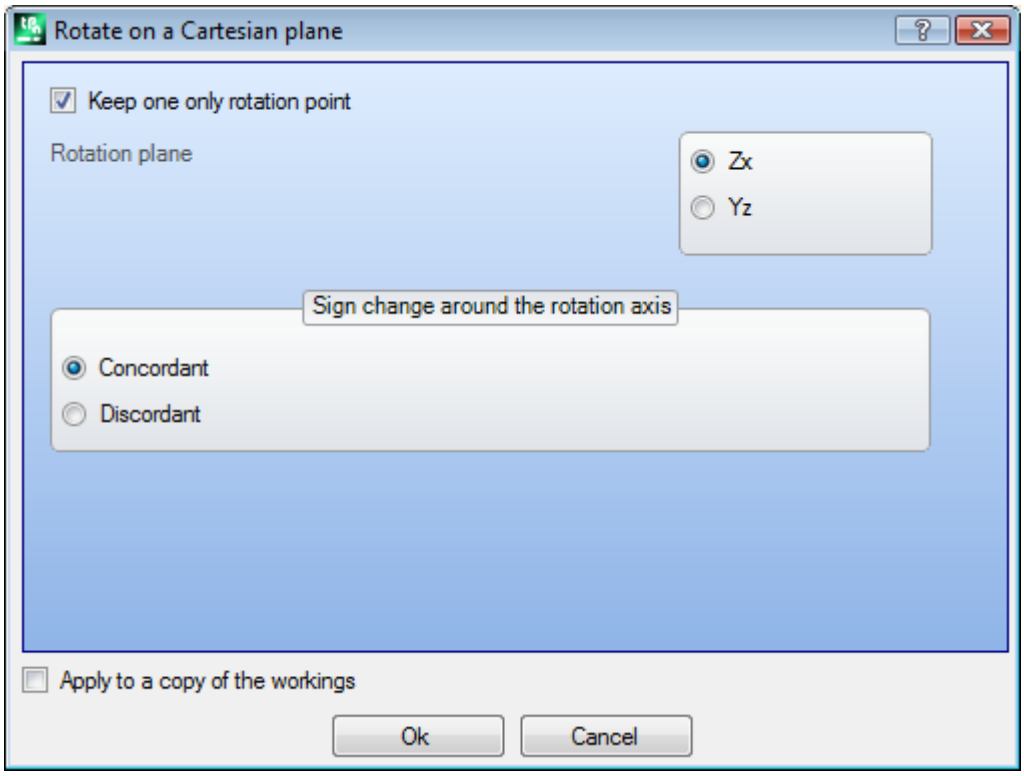
ATTENTION: if the complex working foresees the possibility to generate some profiles for **Recover residual areas**, the **Emptying setup** parameter can assign:
- the code of the setup working also for the generated recovery profiles, in case of a numerical value. In this case: the setup technology is set in **Emptying**.
- In case of string, it can set also a second Name of setup working. The working is searched before the current working with the same criteria as for the Setup of the primary emptying. Examples: "aa;bb": "aa" is the name used to search the setup to be assigned to the primary emptying. "bb" is the name used to search the setup to be assigned to the recovery emptying.

ATTENTION: if it is not possible to use one or both the external setups (no correspondence is found valid for the entry **Emptying setup**), a *Warning* appears and the technology of the profiles is always assigned by means of the settings for the **Emptying** window.

Rotating profiles on a Cartesian plane

This tool allows the rotation of one or more profiles around one of the two coordinated axis of the plane that assigns the face. The **Rotate profiles on a Cartesian plane**  command is available in the group **Constructions** of the **Tools** tab.
In the case of piece-face the tool is only enabled if the 2D or the Box-View face is active and it works on profiles applied to the face in current view only.
It is applied to:
- all profiles which have at least a selected element;
- the current profile.

This tool fails, if any workings for the execution of an arc on the xz, yz, xyz planes are not available.



This tool rotates the profiles(s) of 90° around one of the two coordinated axes of the face plane. The possible2 options are:

- **Keep one only rotation point**: if applied, this selection affects more than one profile. Select the option to keep one only rotation centre for all profiles: the centre coincides with the start point of the first rotated profile. If the option is not selected, each profile is rotated around his own start point;

- **Rotation plane**: two values are selected:
  - **Zx**: rotation is around the X axis of the face;
  - **Yz**: rotation is around the Y axis of the face.

  Now let us see how the coordinates of each segments are changed:

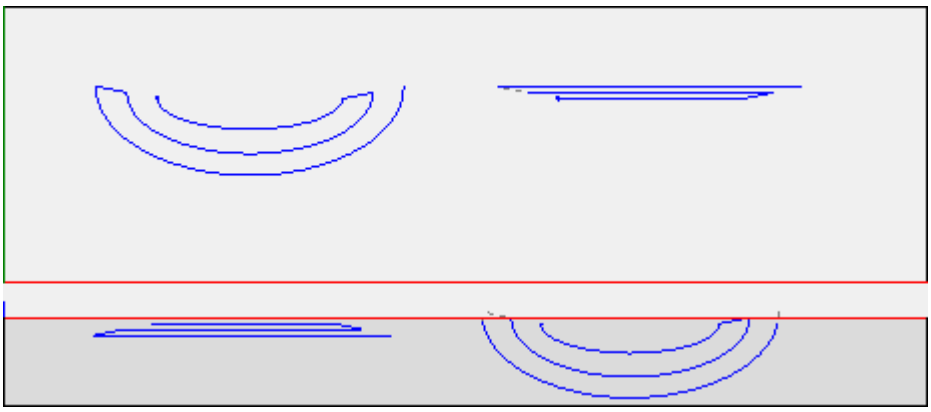
the coordinates along the axis	if Zx plane	if Yz plane
X	they remain assigned in X	they appear in Z
Y	they appear in Z	they remain assigned in Y
Z	they appear in Y	they appear in X

- **Sign change around the rotation axis**: select between the two proposed options, to indicate the exchange modes of the variation of coordinates that concern the exchange
  - **Concordant**: the variations have been exchanged holding the sign;
  - **Discordant**: the variations are exchanged inverting the sign.

  The application of the selection takes into account the depth axis programming modes. For example, with Zx rotation plane: positive variations along Y are given in the variations along the Z axis.

This tool for example can rotate an emptying profile on two planes. In the figure below:
- a start of elliptical emptying programmed on the face plane (left profile);
- on the right, the profile is rotated on the Zx plane.

The 2D view of the face appears above and below appears the front side of the face (the depth axis is represented vertically).



By selecting the option *Apply to a copy of the workings* the tool is applied to a copy of the workings and it does not modify the original lines.
ATTENTION: path elements (L24) are expanded into the micro-segments that assign the curve.
ATTENTION: the tool resets the tool offset in the path.

PROFESSIONAL

It is possible to recall the rotation tool on a Cartesian plane also under form of complex working by recalling the working *Programmed tools* in the list of the workings. In the group of TOOLS select the STOOL working: In the group of TOOLS select the STOOL: STPLANE working.

- the **Workings** field set the names assigned to previously programmed working that correspond to the original profiles.

The profiles may be the result of the application of complex codes and development of working STOOL: STPLANE is only for the modified profiles and it does not include the original profiles. Possible workings that cannot be used for the function required (for example: point or logical workings or complex workings that are not expandable) are ignored.

The working sets:

- Typical parameters of the complex working (see what has been said about a generic code of Subroutine):
  - **Qx,Qy Zp**: initial positioning coordinates of the developed workings.
  - **..**
  - **Working properties**: it sets the properties attributed to the working.

  - Specific parameters of the working function with a meaning analog to the fields in the tool window:

The main advantage of using the working STOOL: STPLANE consists in the fact that the curves fit changes of the original profiles.

10.5 Nesting construction of profiles

Nesting

PROFESSIONAL

This tool allows placing one or more paths within a rectangular or variable outline in order to occupy less space as possible and possibly to repeat the application several times. Individual paths are placed by evaluating the overall rectangle or the shape of the paths themselves which can be brought back to the position of minimum extent by the transform of the path rotation.

The functionality of the instrument is independent of the specific activation from HW key, with two possible levels of operation:

- Rectangular nesting
- Nesting True Shape.

If the operation of Nesting True Shape is available, you can choose the level you want to apply.

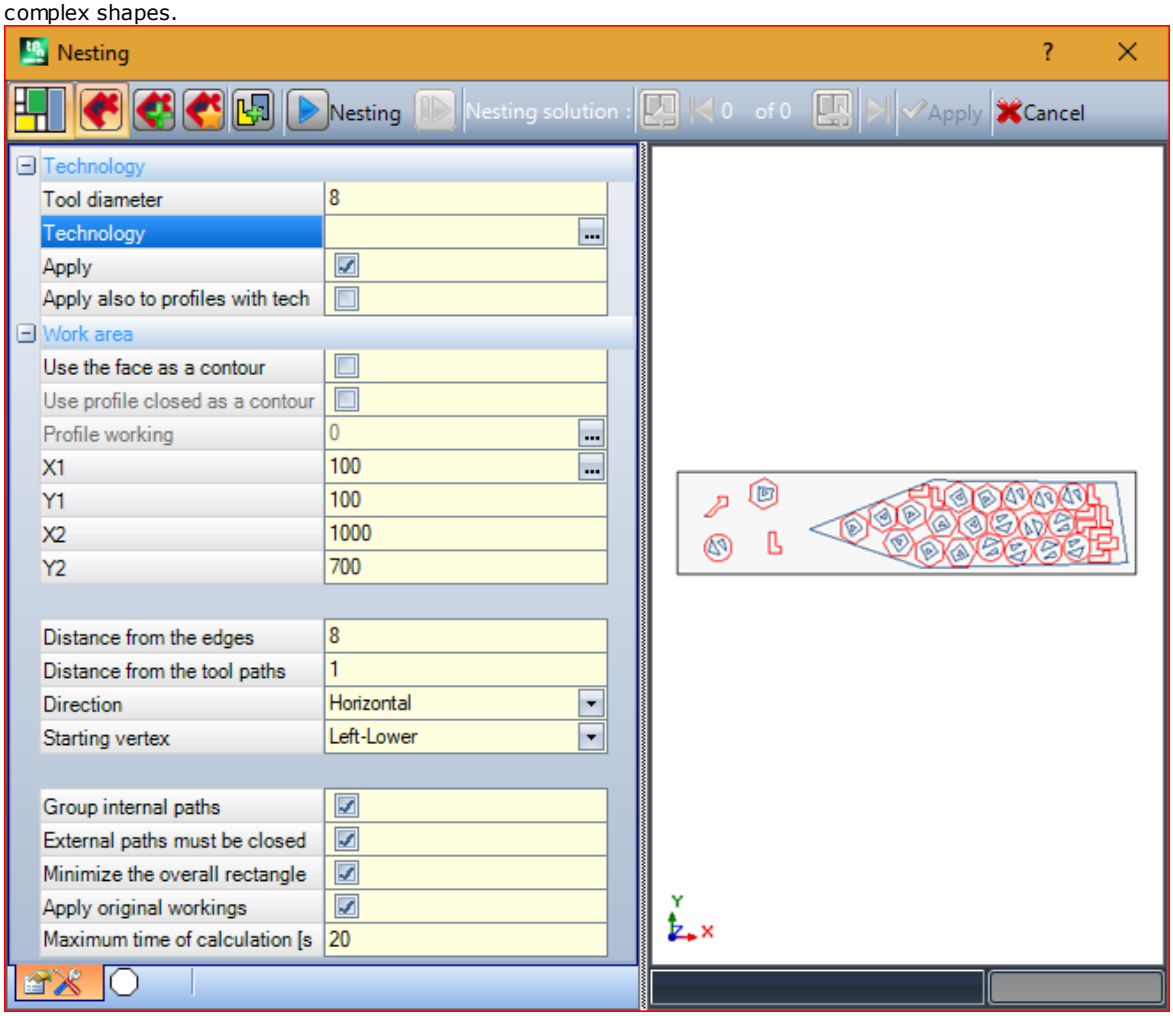
The **Nesting**  is available in the group **Profile Nesting** of the **Tools** tab.

In the case of face-piece the tool only works on the profiles applied to the face in the current view.

The tool works either on the selected profiles or on the current one. The profiles must be simple and not necessarily closed.

The evaluation of the overall dimensions of the profiles does not take into account any possible tool correction request.

It is possible to create some automatic and/or manual groups of the same profiles, in order to place more complex shapes.



A window divided into two areas appears as follows:
- on the left the settings divided into two pages
- on the right the graphic preview of the tool application.

The window can be sized and the vertical division bar allows the resizing of the two areas.

The top bar groups the total available commands:

 The button is managed only if *True Shape* operation is available and changes the image according to the selection status:

- inactive selection: corresponds to *Nesting True Shape operation* and the image that appears is the one above
- active selection: corresponds to the operation of *Rectangular nesting* and the image that appears is  . The selection corresponds to the only possible status, when the operation of *True Shape* is not available and, in this case, the button is made invisible.

Rectangular Nesting manages the placement of the pieces by applying dispositions, which respect the bounding-box rectangles of each single piece. Placements are made within a rectangular contour.

Nesting True Shape manages the placement of the pieces by applying rules, which comply with the actual dimensions of each piece. Placements are made within any closed contour.

The following options are available and applicable only in the event of *Nesting True Shape*:

- **Allow placements in the holes**: select to allow placing within the internal profiles, but only for the groups whose corresponding entry in the column is selected. When the selection is not active, no placement can be applied within the internal profiles, regardless of the group assignments
- **Place in the holes recursively**: select to allow recursive placements within the internal profiles
- **Prioritize placements in the holes**: the active selection privileges the placements within the internal profiles.
- **Automatic clusters**: this selection enables the application of automatic profile clusters with respect to single placements, but only for profiles that have the corresponding entry in the selected column. For each profile in the list, the efficiency that can derive from an automatic cluster is checked: a cluster that assigns an efficiency greater or equal to the value set in the Nesting function configuration (see Nesting function manual, chapter: **Nesting configuration -> Nesting options**) determines a privileged application of the group with respect to the placement of the single piece. Profiles of geometry corresponding to: regular circles, conics or polygons (inscribed in a circle).
- **Grid placements**: this selection enables the application of placements according to a matrix layout, but only for pieces that have the corresponding item in the selected column. This option can be used to create uniform placements according to a grid layout.
  Those pieces for which a grid placement is required, are used before the others and are placed with row* column arrangement, based on the space available on the panel. In order to determine the placement method, each piece can also be analyzed by applying autonomous cluster strategies, to optimize the placement grid. Grid placement performs piece placements with repetition of a unit that can correspond to a single piece, always repeated with the same rotation, or two pieces, with mutual cluster defined with a 180° rotation. The repeating unit, of single or double piece, can then be positioned evaluating a rotation variation of 0° or 90°.
- **Nesting**: the button starts the verification of the settings and consequent nesting optimization. The command can be cancelled if the settings require corrections.

Let us look at the settings pages in detail:

- **Technology**: select the technology to be assigned to the profiles.
  - **Apply**: select to apply the technology
  - **Apply also to profiles with technological setup**: this option applies the technology also to profiles already opened by a setup working. If the option is not selected, the tool is applied only to the open profiles or to those beginning with a GEOMETRIC SETUP working;

  The selection of the technology is unnecessary: if it is not carried out, each profile will be applied without modification in regard to the original technological programming. More specifically, you can apply profiles with different diameter of technology: in this case the overall geometric rectangle of each profile is increased by the diameter assigned.

- **Work area**: the node groups the fields useful for assigning the outline useful for placements

  - **Use the face as a contour**: select to use the whole current face as an overall rectangle for the positioning. As an alternative, you can assign an outline indicating the two extreme points:
    - **X1, Y1**: coordinates of the point of the minimum dimensions
    - **X2, Y2**: coordinates of the point of the maximum dimensions.

    Select the icon of the **X1** field to acquire a profile in an interactive way: the overall rectangle of the profile assigns automatically the fields of the 4 coordinates.
    A minimum value is set for the overall dimension of the outline of the placement, applied to both dimensions, and equal to the maximum size of the indicated profiles.
  - **Use profile closed as a contour**: this selection is available and can be applied only in case of *Nesting True Shape*. Select to enable the use of a contour assigned by a closed profile:
    - **Profile working**: select this icon to select the profile interactively. Profiles used for placements are excluded and the profile must be closed. The fields (**X1, Y1, X2, Y2**) are automatically updated with the values corresponding to the dimensions of the profile itself

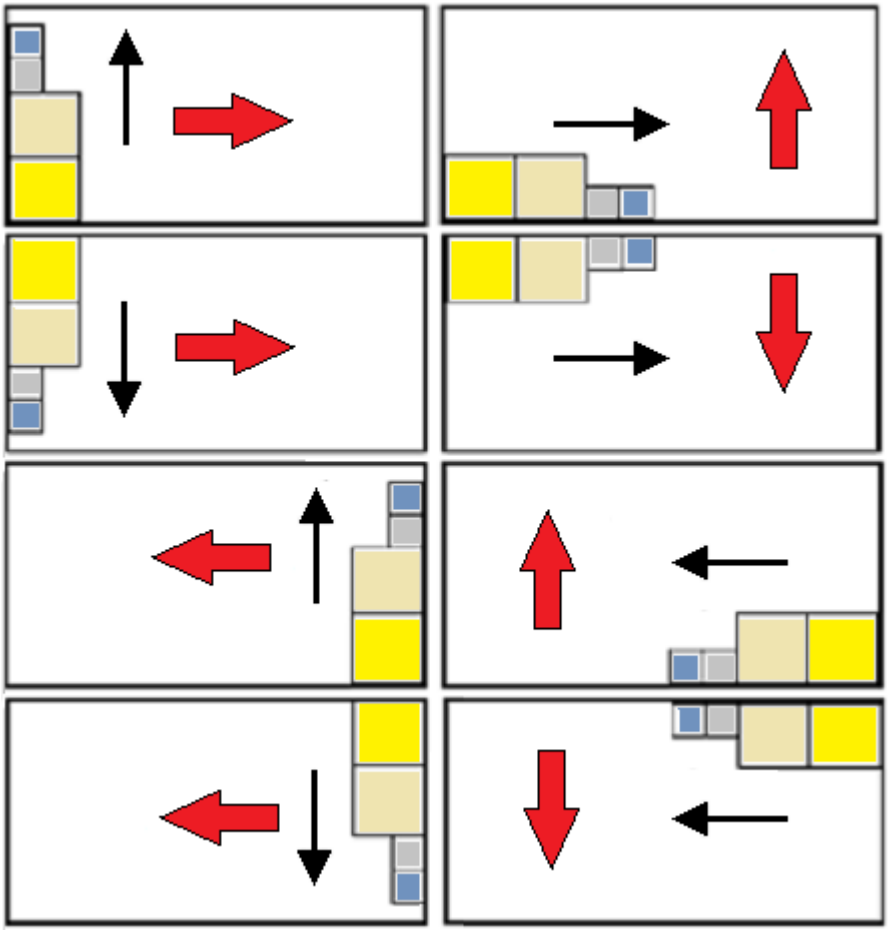
- **Distance from the edges**: margin from the edges of the filling rectangle

- **Distance from the tool paths**: distance that is added to the diameter of the technology, to determine the actual distance of the pieces placed.

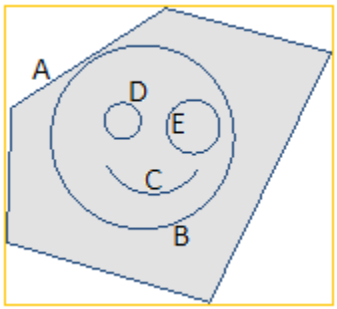
- **Direction**: select the direction of the feed for the placements between the two following available options:
  - Horizontal (in the figure: to the right, horizontal red arrow)
  - Vertical (in the figure: to the left, vertical red arrow)

- **Starting vertex**: it selects the starting vertex for the placements according to the following four options:
  - Left-Lower (in the figure: on the first row)
  - Left-Upper (in the figure: cases on the second row)
  - Right-Lower (in the figure: on the third row)
  - Right-Upper (in the figure: on the fourth and last row)

  The **Direction** and **Starting vertex** settings can be ignored when using a variable contour.



- **Group internal paths**: select to group into a single entity paths that have their own internal dimension in another path. An example of a group automatically recognized corresponds to the figure.

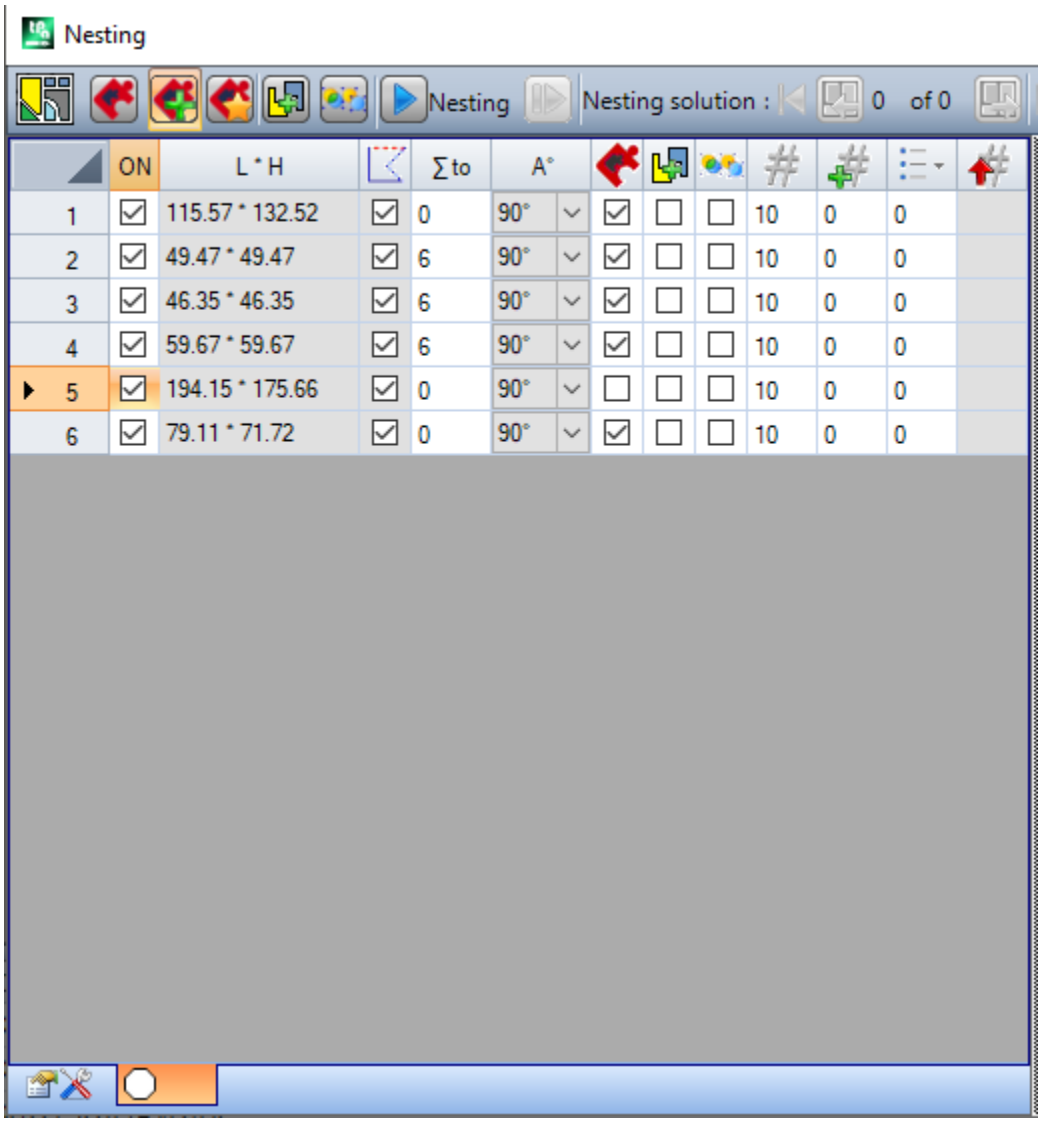


A is the external profile

B, C, D, E are profiles inside **A**: the overall rectangle of each of them is located inside to the overall rectangle of **A**.
The set of 5 profiles constitute a single group and the possible positions will keep unchanged the mutual positions of the original paths. A minimum group is made by one only path.

- **External paths must be closed**: select if the external paths must be geometrically closed. In this case and in the example above: **A** would not be detected as external profile, if it were not closed.
  ATTENTION: in the evaluation of the extreme points of a profile, entry/exit segments programmed on the setup working are excluded.
- **Minimize the overall rectangle**: select to enable the search of the rotation that corresponds to the minimum overall dimension of each group of paths, for which the rotation is enabled (see: second page of settings). This new position replaces the original one, for the subsequent application of the possible rotations.
  The setting is automatically assigned to all groups.
- **Apply original workings**: select to place also the original paths. If this selection is not active, the original workings remain unchanged and their placement does not contribute in any way in the definition procedure of nesting.
- **Maximum time of calculation [s]**: it sets the maximum amount of time (in seconds) within which to terminate the calculation of the nesting. The minimum significant value is 20 seconds, the value 0 deactivates each time limit.

The second page of the window lists the groups of profiles and allows the assignment and placement modes to be changed.



The assignments in the table may undergo more or less general changes after the applications of the assignment of the previous page. A confirmation of your settings in the first displayed page is not directly applied, but leads to the activation of the page itself, so as to allow a review of any automatic changes.

A row of the table assigns a group and each column assigns a setting of the group

- **Row header**: progressive number automatically assigned and used as an univocal identifier (ID) of the group.
- **ON**: the selected case enables the use of the group. Select the header cell of the column to change the box of all rows in the table (if there are selected rows, the change is limited to these)

- **L*H**: dimensions of the overall rectangle (the fields cannot be edited)
- : the box is checked, if the external path is geometrically closed (the field cannot be edited). The column may not be invisible if external paths are required to be closed (see: setting on previous page)

- **Σ to**: this box can assign a progressive number of the group to which the current group can be associated in manual grouping mode. As the illustrated settings, the groups (2,3,4) are associated to the group 6.
  - So, the group 6 will be made of 4 profiles,
  - this overall rectangle of the composed group will correspond to the union of the dimensions of each original path
  - the possible placements will keep unchanged the mutual positions of the original paths.

ATTENTION: if the option **External paths must be closed** and if the group 6 is made by a single profile, the manual grouping will apply only if the profile itself is geometrically closed. If the group 6 is assigned with more profile, the condition is already verified.

It is possible to activate quickly a manual assignment. In our example:

- select the rows of the groups (2, 3, 4)
- right-click the header cell of the group 6

the setting of the case for the column **Σ to** will be automatic:

-  : check the box to enable the placement of the group even in case of rotation of 90° with regard to the original or minimum dimensions automatically determined; If the box is not checked, the piece can be only placed like in the original version.
  In the event of available *Nesting True Shape* the selection occurs on a list of 3 items, where you can select also the rotation *any*, which can be applied only in the *True Shape* functionality (the value corresponding to the rotation *any* is assigned in *Nesting configuration*).

ATTENTION: if more profiles are grouped, also a single profile with limitations inherent in its rotation can disable the field of the group.
Select the header cell of the column to change the box of all rows in the table (if there are selected rows, the change is limited to these).

-  : select to allow placements within the internal profiles of the group. This selection is ignored, if the corresponding global activation on the command bar is disabled. The column may not be visible and the selection is significant only in case of *True Shape* placements.

-  : Check the box to enable the application of profile automatic cluster with respect to the single placement. The selection is ignored if the corresponding global activation on the command bar is disabled. The column may not be visible and the selection is significant only in case of *True Shape* placements.

-  : Check the box to request the placement according to a grid development. The selection is ignored, if the corresponding overall enablement on the command bar is disabled. The column may not be visible and the selection is meaningful only in the case of *True Shape* placements and with placements within a rectangular work area.

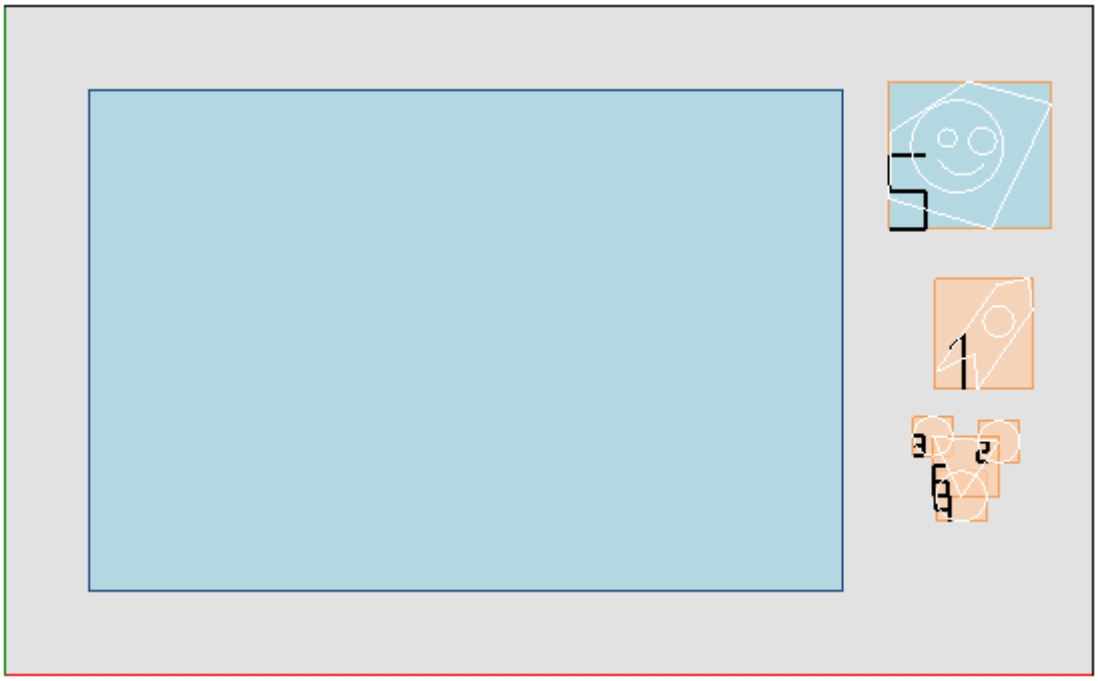
-  : quantity to place. Set a positive value (>=0) not greater than 999. The field is initialized by the value **Repetitions** assigned in the previous page. It is possible to change the setting by differentiating the quantity for each group. The setting of a group manually associated to another one is ignored and the value of the reference group is applied.
- In the case of a single group and value 0, the nesting procedure will try to place the highest number of possible repetitions for the group itself.
-  : a value greater than the previous one assigns the maximum usable quantity (not greater than 999): the difference between both values is the quantity that can be used for filling the fill area, only after placing the quantities of all groups. The setting of a group manually associated to another one is ignored and the value of the reference group is applied.

-  : the column is automatically managed and shows the quantity actually used downstream of the Nesting solution (the field cannot be edited).

-  **Priority**: groups with assigned priority take precedence in nesting solution (default value: 0: maximum value: 100). How the priority value is interpreted is defined in the option **Lower priority with increasing value** in Nesting configuration (see manual about nesting functions, chapter **Nesting configuration->Nesting options**).

The maximum number of profiles that can be requested is 10000.

Overall assignments window correspond to the situation as below:

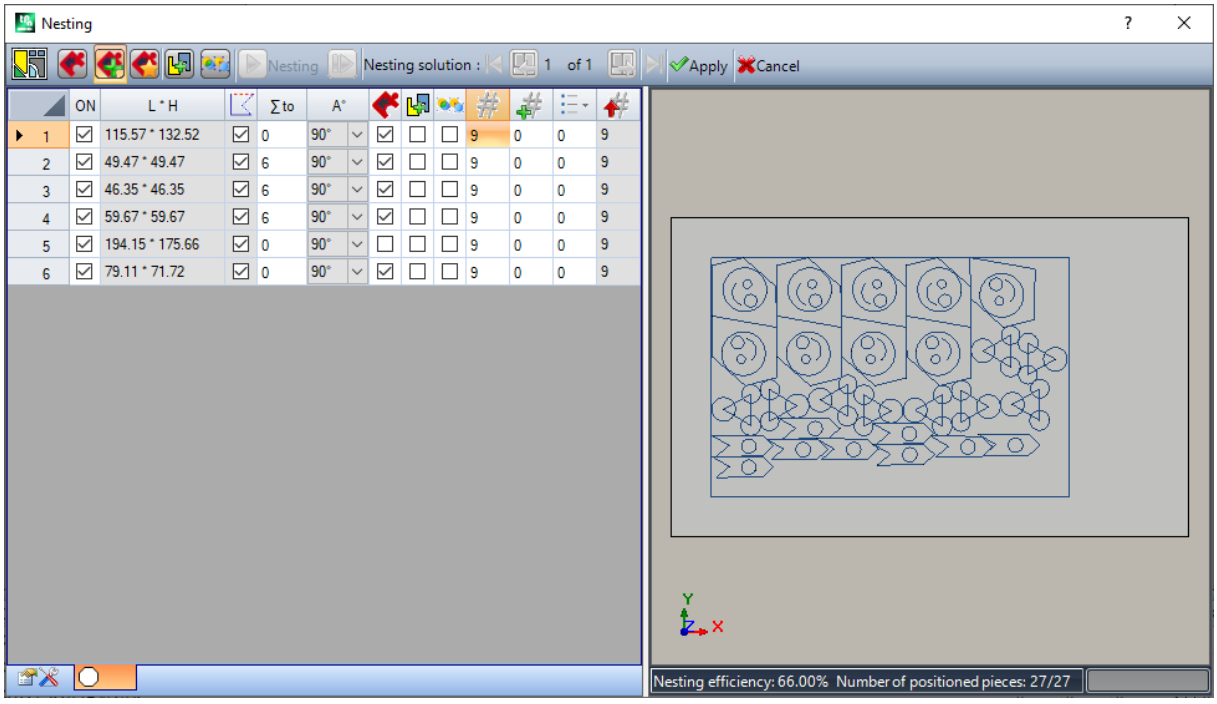


On the left, there is the placement rectangle.
On the right there are the groups of profiles, each enclosed in the overall rectangle and identified by a progressive number:

- the groups 1 and 5 show the application of the option **Group the internal paths**
- the remaining groups (2, 3, 4, 6) are not automatically grouped, because they do not verify the criterion of internal paths. As stated before, here they can be manually grouped in a single group (in the example 6).

Changing the current row in the table, the corresponding group is graphically shown (in the figure: the Group 5).

In the event of *Rectangular nesting*, the confirmation of the settings leads to the result shown in the figure:



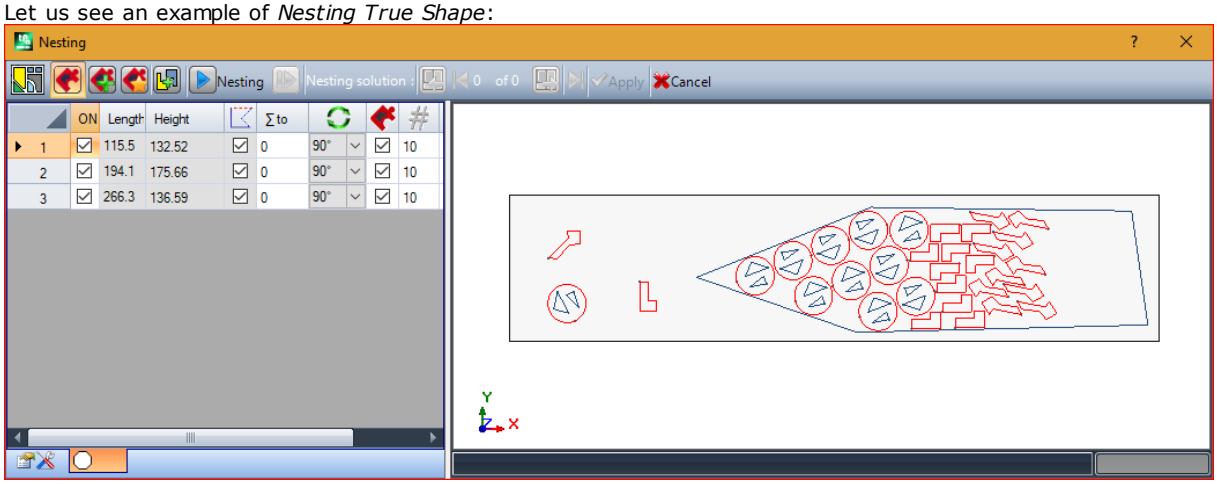
The figure shows the rectangle assigned for the placement of the profiles.
For each group we have performed the placement of 9 copies for a total amount of 27/27 placements required.

It is clear how each group can be positioned differently with respect to the original workings: enabling the rotation has activated the search for the position corresponding to the minimum size. The placements of group 1 (arrow-like silhouette with inner circle) show that 90° rotations are possible.

By modifying the settings it will be possible to request a new nesting application.

Apply: select to apply the results to the current program
Cancel: select to close the window by cancelling the command.

Let us see an example of *Nesting True Shape*:



The placement of the profiles is now possible within a non-rectangular area.
Restart: the button requests to start the *True Shape* nesting procedure, taking the last solution calculated as a starting point. Then you can proceed to determine more solutions, up to a maximum of 10, and scroll through them choosing the one that you consider to be the best:
Go to the next solution:

Go to the previous solution:

these two buttons allow to scroll and activate one of the calculated solutions.

Apply: select to apply the results of the current solution to the present program

The request for the application of the Nesting procedure leads to various checks of the validity of the assignments, by which it is possible to report and /or modify the settings made. More specifically:

- the extension of the placement outline does not verify the minimum dimensions
- at least one group in the list must be active.

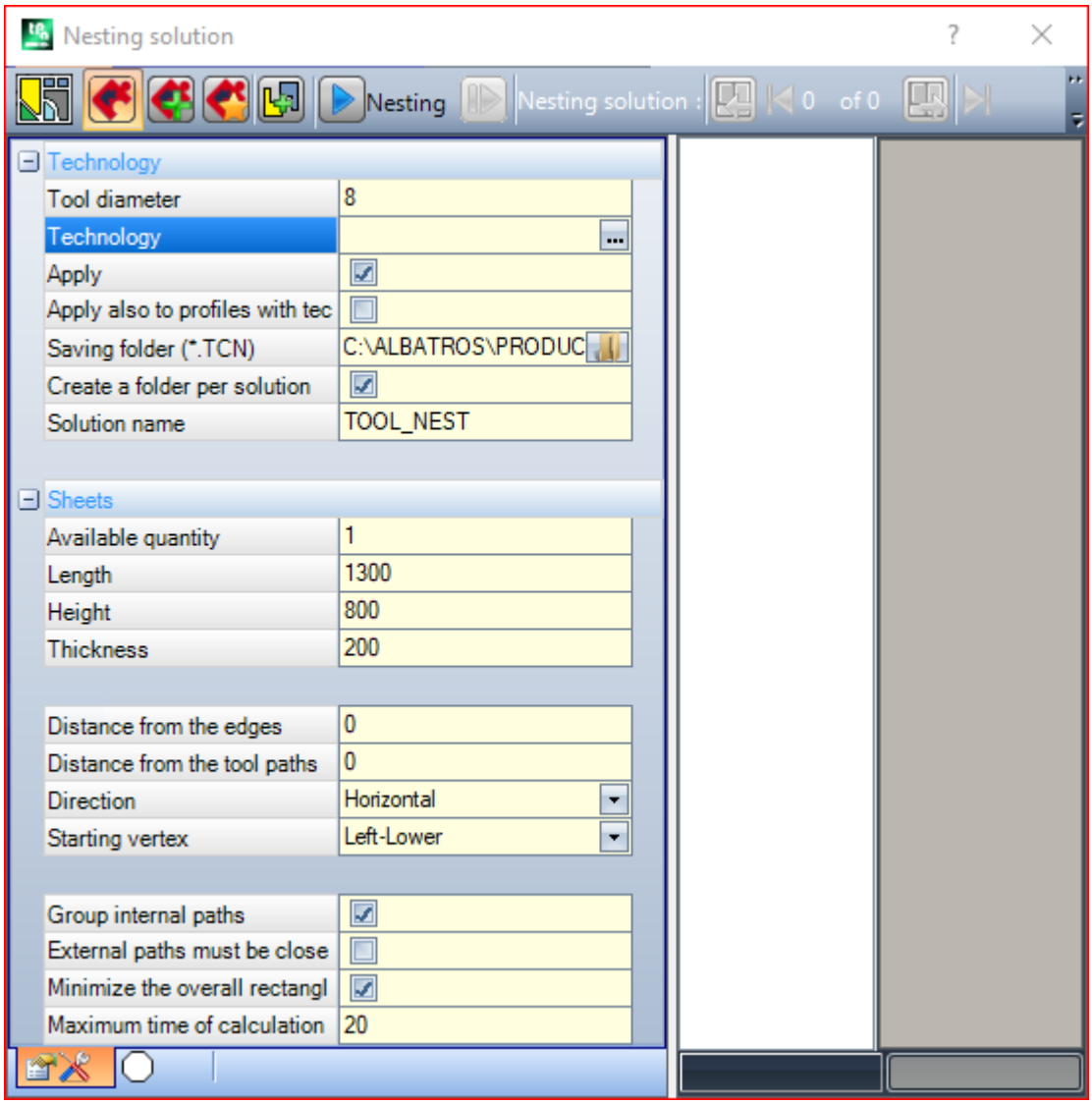
Nesting solution

PROFESSIONAL

The **Nesting solution** is available in the group **Profile Nesting** of the **Tools** tab.
This command is not available in Demo mode or if the view of a face different from 1 is not active.
The general presentation of this tool is similar to that of the tool in the previous chapter and we refer to it for a preliminary analysis.

The feature of the tool is that the placement of the profile groups occurs now not on the program under edit, but through the creation of new programs (*.TCN), otherwise called sheets.

The first page of the settings shows some variations with respect to the first page of **Nesting** command settings

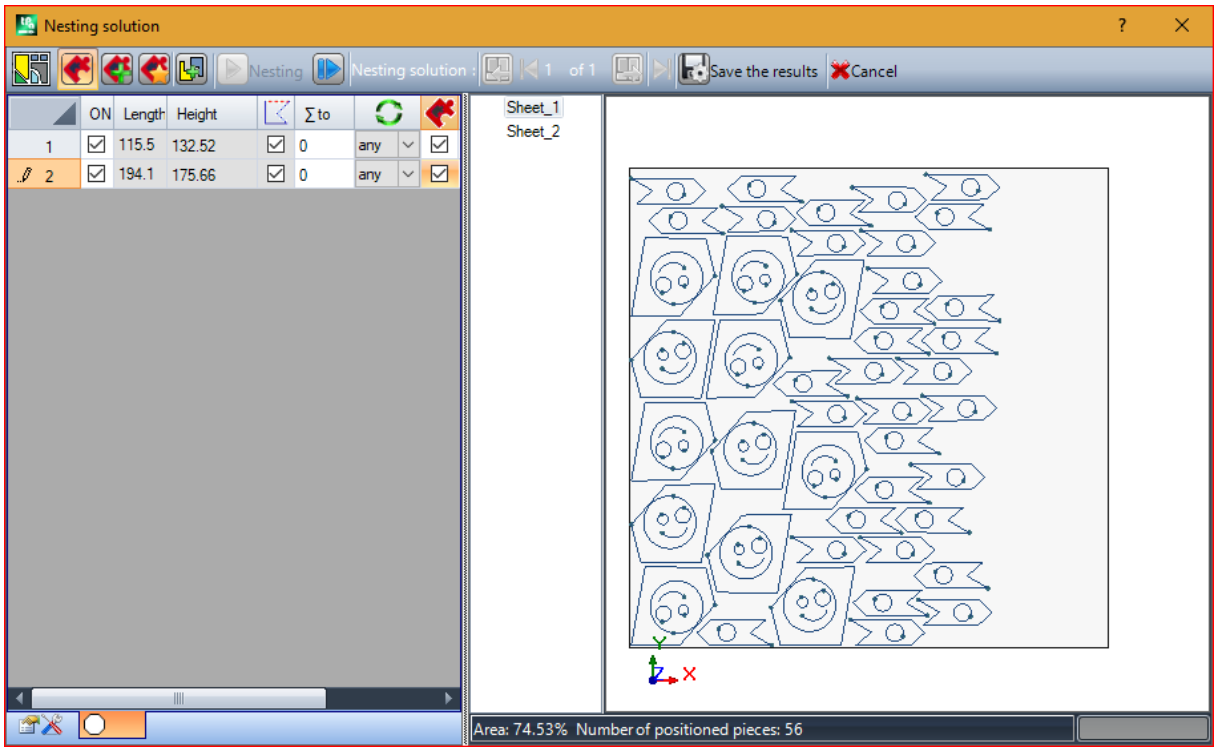


- **Saving folder (.TCN)**: this is the path for the *solution*
- **Create a folder per solution**: this option creates with the box a storage folder for the solution. This box is selected and cannot be changed.
- **Solution name**: this is the name assigned to the solution.

This folder, created to record the programs of the solution, must be reported under **Saving folder**. The names of the programs have a common matrix given by the name defined in the **Solution name**, followed by '_' (underscore), a letter, if needed, to differentiate from previous saving ('a', 'b',...) and a progressive number. Examples of composite names: "tool_nest_a1", "tool_nest_a2", "tool_nest_b1".

Infos on the sheets:
- **Available quantity**: set a positive value (>=0) not greater than 100. If the value is 0, this procedure calculates the number of the panels needed to place the total amount of the profile groups.
- **Length, Height, Thickness**: dimensions of the sheets. Each (*TCN) program will be created according to the dimensions set here; the unit of measure is the same as the edited program.

This figure is an example of application of *Nesting True Shape*:



The solution can lead to more than one TCN program being generated: in the figure two.
You can generate new solutions, as reported for the previous tool, and choose the solution to be applied.

Once the procedure has come to end, some messages are displayed that describe the outcome and indicate the number of the placements and the number of the programs that were recorded.

TCN programs are created using the prototype program file or the nesting panel file as a starting point, according to what is assigned in the nesting functionality configuration (see Nesting functionality manual, chapter **Nesting configuration -> Nesting options**).

Depending on the saving options for *.TCN programs and the results of the Nesting functionality, a window might open to also select the file saving with format conversion, among those made available by the configuration.

At the end of the execution of the tool, the path assigned to save the *.TCN programs is set as the last open for the next time you open the program.

10.6 Advanced tools in face program

Create fictive face from geometry

PROFESSIONAL

The command is enabled in face view with face program not empty. The **Create fictive face from geometry** tool is called in the **Advanced** group of the **Apply** tab.

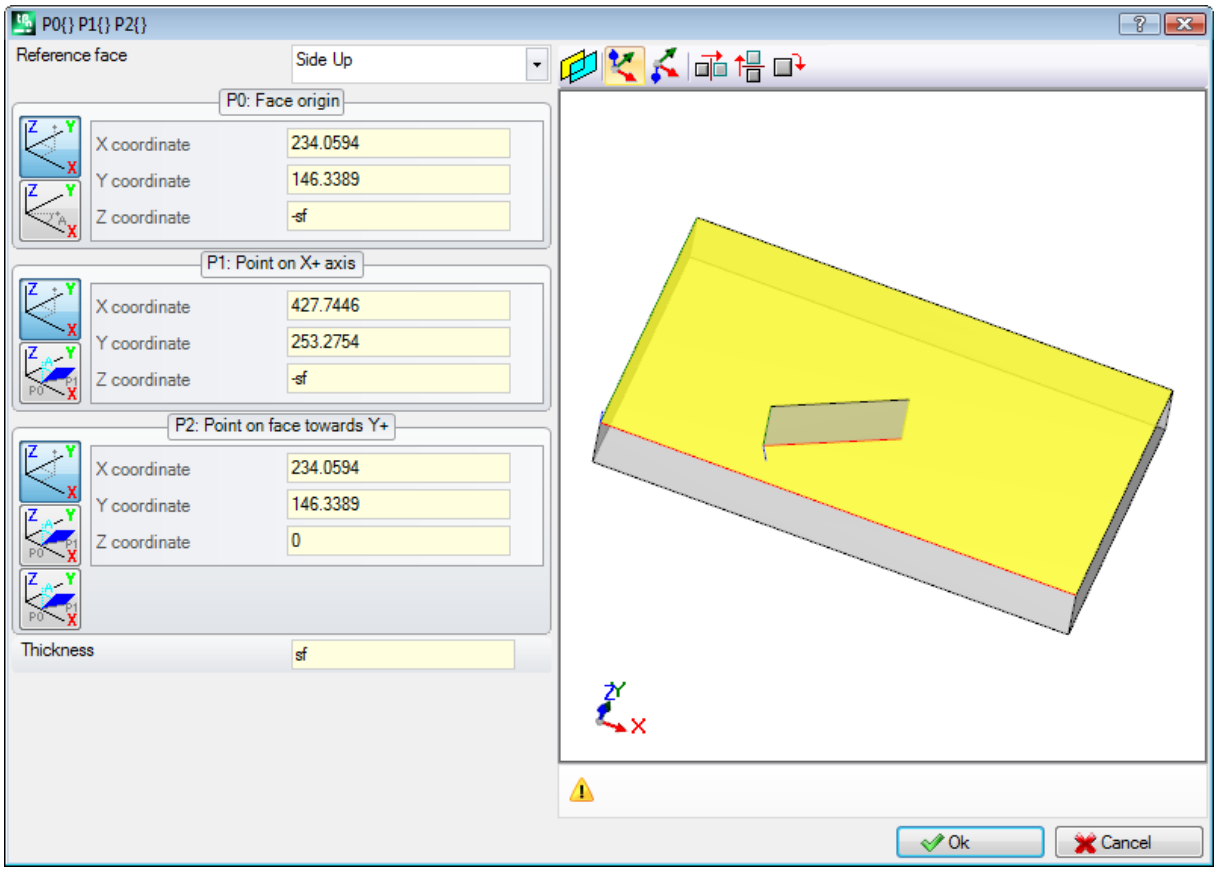
In the case of piece-face:
- the command is not active if the current working is applied in automatic face
- This command does not work, if the current working is applied to an automatic or to a fictive face of curved typology or surface.

- the command is only enabled if the 2D face view is active, with face view corresponding to the application face of the current working.

It is about a tool for simplified creation of fictive faces, on the basis of linear or curved segments already programmed on the face.
If a linear segment belonging to an oriented profile is found, TpaCAD requires if you want to **orient the face according to the profile**.

- an affirmative answer creates an inclined face with respect to the vertical direction to the face;
- a negative response creates a vertical face to face current.

The possibility of creating a face from a curved element is due to the fact that the curved faces can be managed.
Once the segment has been found (by the mouse), the window to set the fictive faces opens:



Fields are set on the basis of the positions read by the identified segment.
The insertion of the fictive face into the list of program faces takes place after confirming the exit from the window and assessing the geometric correctness of the face.
Once the fictive face has been entered, it is possible to carry on with this command by indicating another linear segment or exit the command window by pressing the **[ESCAPE]** key.

ATTENTION: If the management of the Curved faces, you can select also a curved geometric element (arc in the xy face plane).

ATTENTION: this command is not available in view of curved fictive face or assigned as a surface.

A face inserted with this procedure does not remain in no way constrained to the linear portion used for its setting. The segment may change or be cancelled and this will not, in any way, a subsequent modification or automatic removal of the same face.

Create surface from geometry

PROFESSIONAL

This command is enabled in face view and with a not empty face program. The **Create surface from geometry** tool is recalled in the **Advanced** group of the **Apply** tab, when the functionality is enabled and active

The description of the associated functionality is included in the specific documentation, that can be recalled by the "Help to modelling" available in the menu.

Create modelling from geometry

PROFESSIONAL

This command is enabled in face view and with a not empty face program. The **Create modelling from geometry** tool  is recalled in the **Advanced** *group* of the **Apply** tab, when the Modelling section is active and enabled.

The description of the associated functionality is included in the specific documentation, that can be recalled by the "Help to modelling" available in the menu  .

Create font from geometry

PROFESSIONAL

The command is enabled in face view with face program not empty and cannot be available, depending on the configuration of TpaCAD. The **Create font from geometry** tool is called in the group **Advanced** of **Apply** tab. The command allows to assign a character custom font from programmed profiles.

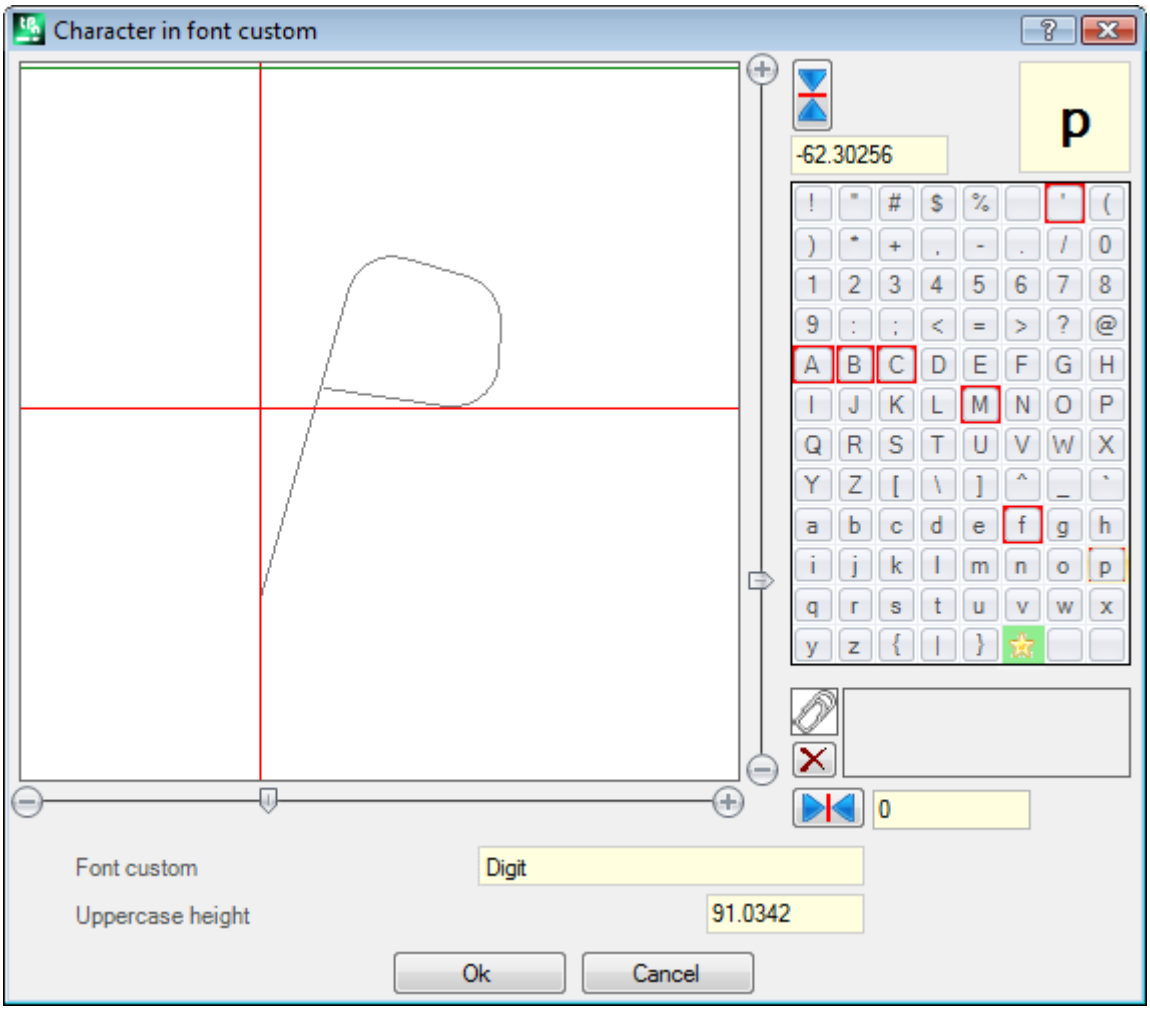
We will say from the start that the aim of TpaCAD is being able to use the custom fonts, and not managing their generation in a complete and multi-step way: the tool described here seems to be limited, but it is provided as an additional help and not as an incomplete feature.

The format of the custom font is simple and documented and the basic installation of TpaCAD provides an example.
Anyone can "enjoy" to invent others, at will, perhaps using a background image to current face and "drawing" polylines following the design background.
The tool *Create font from geometry* can be useful in this case to create a first draft of the characters.
Starting from TpaCAD programs, perhaps imported from an external format - DXF or other -, for example, it can predict the development of a form of *Export to the font file*.

As already told: a custom font in the font file is described with one or more profiles, each characterized as a polyline.
It is initially proposed the list of installed files as custom font, with the ability to assign a new font.
Then starts a procedure for interactive selection of profiles to be considered:

- You can select only simple profiles assigned entirely linear segments, arc or paths;
- curved segments on not face plane are considered as linear segments.

When we close and confirm the interactive procedure, a window open to complete the command.



- **Font custom**: the font name that has been select (It is not editable).
- **Upper-case height**: the font height above the baseline. The value indicates how much a character in the font may rise above the line of writing. The field cannot be modified and It is assigned on the creation of font (read: if you assign the first character of a font) of the vertical overall dimension of profiles that have been selected. In creating a font is required starting with the allocation of the **A** capital letter. If you add a font already assigned, the value is not editable.

The value **Upper-case height** has a corresponding even in the system font and is used by the procedures of application of the font for scaling and positioning the characters along the writing line.

The left picture represent to scale the selected profiles.
The area is dimensioned so as to include the entire character, positioned on the upper quadrant as identified by the cursor represented with the two red lines:

- the point where the two lines cross represents the zero placement of the character, horizontally and vertically;
- the area above the horizontal red line is dimensioned at least equal to the maximum between the height of the font and the vertical dimension of the profiles represented;
- the area under the horizontal red line is dimensioned at least equal to the height of the font.

The horizontal green line shown on the top of the image corresponds to the height of the font.

If the first character of the font is assigned, you can place the character on the single horizontal line of the cursor: otherwise the character can be positioned along both lines.
The positioning of the cursor on the horizontal line assigns a fixed offset added to horizontal overall dimension of the character: the offset will have a positive sign, when it moves to the right and a negative sign, if the character moves to the left.
The positioning on the vertical line of the cursor assigns a deviation from the base line of application of the character. The deviation will be of positive sign with upward displacement, negative with downward displacement.
The value of the deviations is reported in the two boxes provided on the side of the scroll bars, with possibility of direct assignment of values. As in the figure, the character represented is assigned:

- a deviation downwards equal to -60.51295 mm

- one horizontal deviation zero.

The positioning of the character takes place with the two scroll bars located on the right side (vertical bar) and below (horizontal bar) in the image, where the two red lines are placed:
- move the slider to non-precision positioning;
- otherwise select the extreme buttons (**+** and **-**) for precision positioning.

The two buttons shown in line with the scroll bars show the respective deviations at the initial zero value:
 resets the offset along the horizontal axis
 resets the offset along the vertical axis
Considering for example the word **A g l q `** in the font of your system, have a value of extension:
- below the baseline the characters **g** and **q**;
- above the baseline the character **`**. The character **`** can be assigned as a horizontal offset (moving the character to the right side of the red vertical cursor) in order to increase its real size.

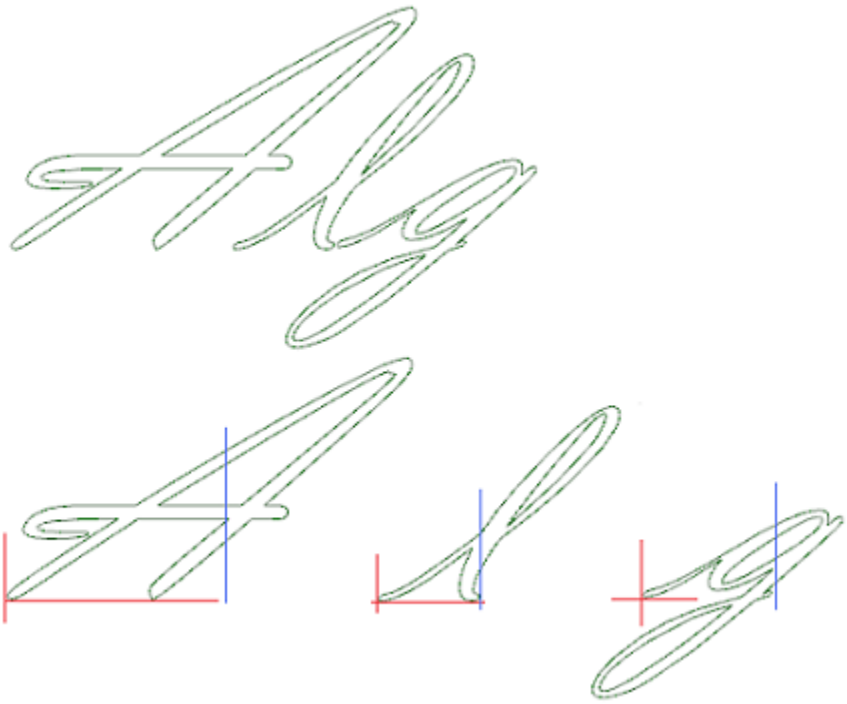
The horizontal scroll bar on the top moves the blue vertical line and allows to assign the hook position of the following character. The positioning is significant, if it is placed on the right side of the vertical red line (see the figure).

ATTENTION: the horizontal offset (vertical red line) and the hook position (vertical blue line) are applied in the metric spacing mode. In the geometric spacing mode, the considered overall dimension corresponds to the real overall rectangle of the single character.

On the right side of the window shows:
- the character is assigned (in the figure: **p**);
- the character map that you can assign. The characters, already assigned in the font, have a red border: it is only a visual signal, which however does not block the change.

The character to be assigned can be changed directly in the box or selecting it on the map.
The graphic character in the shape of a star indicates 'wild card' character: represents the character that is used to write a character that has not been assigned.
 The associated text field allows to assign the characters that "use" the same character (in our example in the figure p is represented with the same profile of the **p** letter). The text field cannot be directly modified.
- select the CTRL key and a character in the map to add or to remove the character from the field;
- select this button  to reset the field of the text.

As previously mentioned, in the creation of a font is required starting with the allocation of the capital letter **A** and the character map is not interactive.
In the figure a custom font has been assigned from a system font installed:
- at the top of the figure, the text "Alg" is developed with the Metric placement;
- at the bottom of the figure there are the positions of the cursors, as they have been assigned in the Custom font window.



10.7 Useful tools

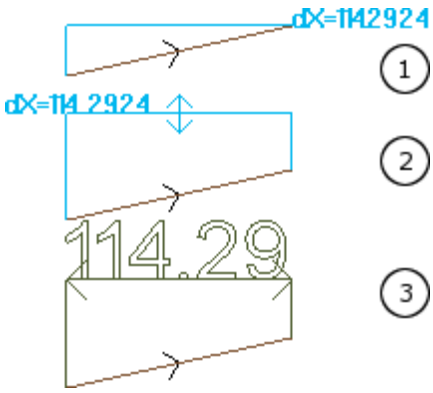
Dimensioning

PROFESSIONAL

The commands are available in the group **Dimensioning** of the **Apply** tab.
This tool is available only if in the database of the workings a specific working code for measuring is assigned and if working properties has been enabled. Construct ("B" field) and Name ("N" field).
The measuring tool allows to add *lines* to the measuring program; these are special processes that do not determine any running but only Views of Construct, as an aid to direct documentation of a program. The segment to measure is chosen with the mouse, directly in the graph area. Indications about the sessions sequence are given in the Commands area.
You can choose from four different types of measuring:

- **Horizontal**: inserts a horizontal line measurement and the dimension value.
- **Vertical**: inserts a vertical line measurement and the dimension value.
- **Horizontal+Vertical**: inserts a vertical measuring line and its position. inserts a horizontal measuring line and its position.
- **Diagonal**: inserts a diagonal line measurement and the dimension value.

Example of a horizontal measuring



The figures show the sequence of necessary steps that complete the insertion:

1) two extreme points of the linear segment are identified to calculate the horizontal dimensioning (dX=114.2924)

2) then, the vertical position of the dimensioning writing is set

3) After confirming the command, a horizontal segment with arrows at its edge points and the value of the segment (114.29 in this case) is inserted.

The measuring information included here is not in any way dependent on the factors on which the measurement was performed (snap points, or other); the elements themselves may change or be eliminated, but neither change nor automatic removal follows.

Measures

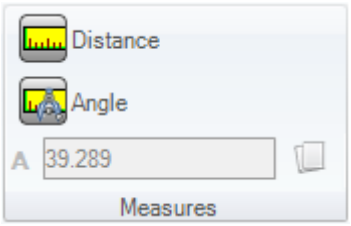
The commands are available in the group **Measures** of the **Apply** tab. There are two specific commands:

- **Distance**: Measure the linear distance between two points, however, identified in the graphic area. The distance between the two points is represented with a linear segment, shown with the actual distance calculated.
- **Angle**: Measure the angle between a vertex and two linear segments.

The position of the points is assigned with the mouse, directly in the graph area. The Indications are given in the Commands area. From the context menu, you can activate the snap to grid or entity.
It may be of particular interest to recall the possibility to use the snap:

- on the Faces;
- in Depth;
- on Bookmark entities;

With confirmation of the command, the corresponding size is shown in the menu, the group measures the Apply tab, as shown in the figure. The copy command reported alongside performs the copy of the value, for a possible future use.



In case of linear measure (**Distance**), it is possible to choose from the measure found for both following positions:

- 3D: distance 3D
- dX, dY, dZ: distance along the X/ Y/ Z coordinate. If the measure has been acquired by the snap between different faces, the distances refer to the absolute reference system of the piece.

10.8 Overall Program Tools

These are commands available in overall view, called in **Apply to piece** group of **Apply** tab.
All these commands can be applied to the whole program and they normally have an equivalent command in the Tools applicable in Face view.
On confirmation of Overall program tool, a setting window of the *Search options* appear:

- **View match**: if enabled, it takes into account the only workings displayed (it applies active views and view filters).

Let us see in details the views and the applied filters:
- the search excludes the workings: logical, with active C field or with locked property (L, B field) or with operational invalid code (read: the working has no correspondence in the working database);
- if the View of Selections is active: it considers just the selected workings;
- if the View of logical Conditions is active: it considers the only workings that verify the logical conditions, including the exclusions;
- if the View of the Filters of the layers is active: it considers the only workings assigned with displayed layer;
- if the View of the special Filters is active: it considers the only workings verified by the special view filters (fields: B, O, K, K1; technology).

- **Apply to selected workings**: if enabled, it considers the selected workings only (the field is enabled only if there are selected workings). The activation of the option is considered only if the item **View match**, by which it is already included, is not enabled.

Apply technology

The command **Technology**  is similar to Apply Setup To Profile tool. It applies technology to point workings and open profiles, that is without opening setup or headed with geometric setup.
The type of setup or point code to be assigned is selected in a window where all workings of selected typology, made available by the software, are displayed.
All technological parameters are set in the standard window of Technological assignment. No technology can be assigned to open profiles or profiles headed with geometric setup or geometric points, if defined within a complex working.

In a point working the parameter Diameter is assigned according to the following rules:
- If the working of point geometric code **has not** any diameter value set, the substitution **is** made.
- If the working of point geometric code **has** a diameter value set, the substitution **is not** made.

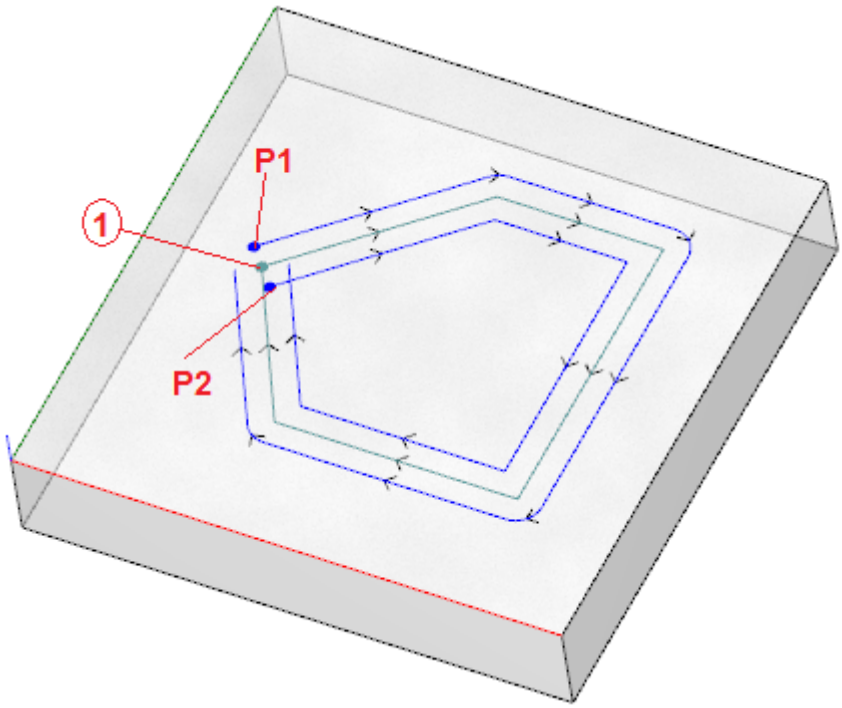
Convert [mm]-[inch]

This command **[mm]-[inch]**  converts the program from [mm] to [inch] or vice versa. The conversion is applied only to the information on position or on speed.
This tool is mainly used to convert programs imported from different formats (for example, from DXF) and originally written with a different unit of measure.
The command may not be available and, if its use is required, the database of the workings must be properly drawn up, so that all the information about the working to be modified can be correctly found.
There is no equivalent of this command in the applicable Tools in Face View.

This tool warns if its application may concern parametric programming; in this case the user can carry out the numerical forms only and exclude the parametric ones.
The application of the conversion also concerns the program units: dimensions and unit of measure, execution modes, "o" and/or "v" variables with assigned dimension, modelling geometry, fictive faces (geometries and additional parameters).

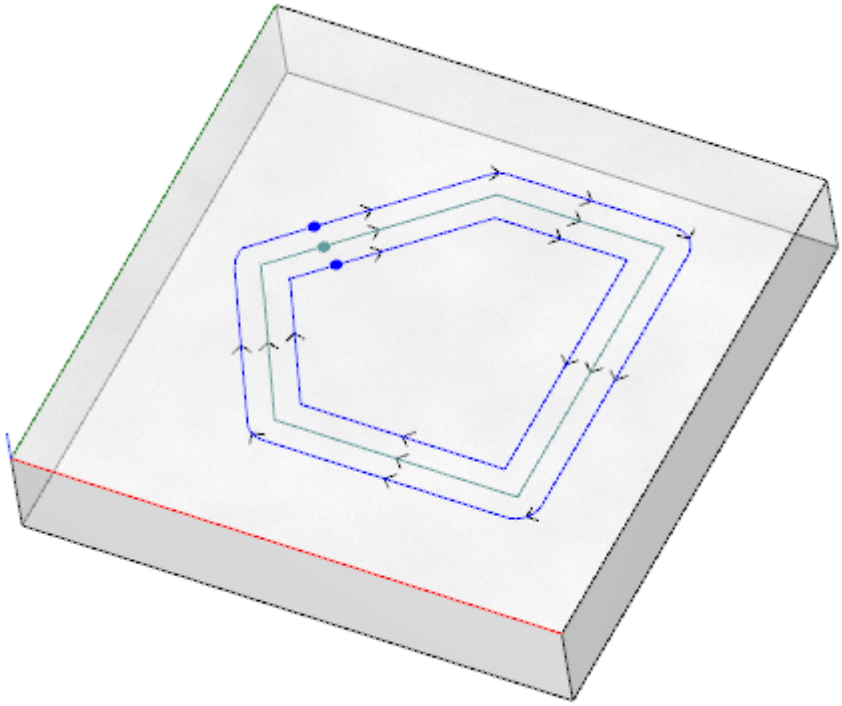
Validate profiles

The **Validate profile**  command examines the setup point of the closed profiles and, if necessary, it moves it in such a way as to neutralize any application of compensation tool or even only the entry of the tool in an "uncomfortable" point.
The examined situations correspond to start points of an edge profile. Let us consider the example in the figure:

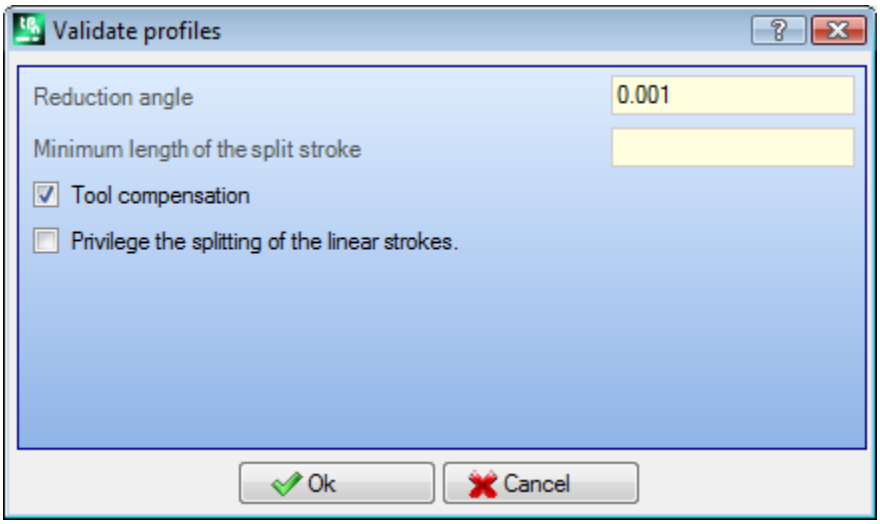


The central profile corresponds to the programmed profile: the setup is indicated as a point (1);
(P1) shows the setup of the profile taken through left Compensation;
(P2) shows the setup of the profile taken through right Compensation.

It is realized that both the compensated profiles present a problem:
(P1) is no longer a closed profile;
(P2) shows an intersection of the segments outside the profile.
In this case the tool would move the setup on the first linear segment, breaking it in two segments, now in tangent continuity. The picture shows the new situation:



The criteria for the application of the tool are shown in the window:



- **Reduction angle**: sets the angular value of tolerance to recognize the condition of continuity on the original setup. Sets a value not exceeding 45° (with null value set, currency the continuity of tangent with a tolerance of 0.001 °).
- **Minimum length of the split stroke**: sets a minimum length for the choice of the profile segment on which the setup should be moved.
- **Tool compensation**: to be selected, if you need to take into account the overall dimension of the tool, when you calculate the minimum length of the single segment of the profile. In this case, the minimum length required is calculated equal to the diameter *3.0.
- **Privilege the splitting of the linear strokes**: to be selected, if you need to move the point of setup on a linear segment, if it is possible. If this is not possible, the setup is moved on the first segment, line or arc, that have the minimum length.

The minimum length of the segments is calculated on the face plane (xy plane) and also takes into account both the set value (**Minimum length of the split stroke**) and the **Tool compensation**, with a minimum value set at 50.0* epsilon.
This tool precludes the calculation of portions of the profile assigned with complex codes or with arcs on the #xy plane.
Once the segment on which the setup will be moved has been identified, the segment itself is broken into two sections of equal length.

It is possible that a profile assigned on all very small segments does not allow the application of the tool.
The application of the tool can bring about the loss of parameter settings.

Apply reduction to profiles

This command **Reduce profiles** is similar to Minimize the profile tool. It reduces the number of segments by assigning criteria of angular and/or linear reduction. Profiles defined within a complex working cannot be minimized.

Apply fragmentation to profiles

The **Fragment profiles** command is similar to Fragment profile tool. It fragments the segments of a profile into maximum assigned length segments. Fragmentation concerns the arcs only, where fragmented segments can be linearized. The command does not fragment the profiles defined within a complex working.
In the dialog box the parameters must be set that are already considered for the Fragment profile tool, to which reference is made.

Apply connection to profiles

The command **Connect profiles** is similar to Connect consecutive profiles tool.
For each concerned face, profiles are connected with check of geometric continuity between the starting and the end points by also assessing the **profile inversion**. The tool identifies the starting connection working with the first (isolated or not) setup or profile segment working. To ensure that the distance of connection is also valued on a component of depth (Z axis) you need to select the option in **Apply in 3D**. Profile connections cannot be created within a complex working.

The procedure for connection can take a long time in the case of a large number of profiles because of the recursion of the procedure itself: for each programmed profile the program searches any possible connections to all the other profiles; this search ends when no possible connection is found. The above cases can correspond to programs with thousands of profiles.
The selection of the option **Reduce data matching search** allows to reduce the search scope of connection: for each profile you can only examine the profiles programmed downstream.

10.9 Overall program transforms

PROFESSIONAL

They are commands available in General view, recalled in the **Apply to piece** group of the **Apply** tab.
All these commands are applied to the whole program and have no equivalent in the Tools that can be applied to Face view. With regards to these programs, we are talking about overall transforms, because they lead to a change of the piece as a whole. The transforms also change the piece-face, if needed.

General conditions to use the commands:
- enabled real faces;
- if only one face is enabled, it must be the 1 or the 2;
- the faces 1 and 2, if both enabled, must have the XY face plane placed in the same way;
- all the enabled side faces must have the Y axis along the absolute vertical direction with the same assigned direction (Z+ axis or absolute Z-).

Specific conditions to use the commands for the current piece:
- the piece should not assign variable geometries (fictive or automatic faces);
- if you need to transfer the working to another face, this one should be enabled;
- if you need to apply a mirror or rotation transform to the workings of only one face, the transform should be applicable to all the workings, also by exploding them, if needed.

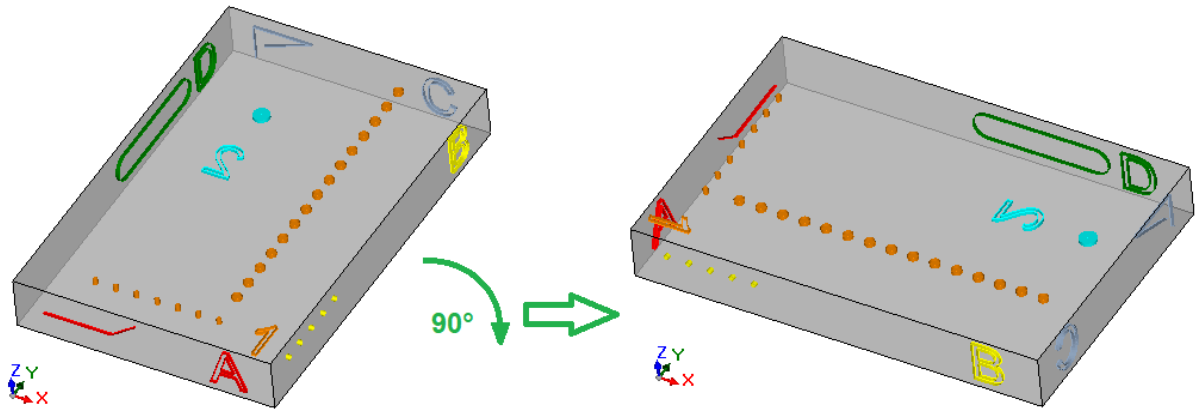
The execution of the commands will result in the cancellation of the command list that can be cancelled or restored, also if the commands themselves are cancelled.
If the commands are cancelled, the piece is restored to the previous status.

Rotate the piece

 This option applies a counterclockwise rotation of the piece of 90°.

Specific conditions to use the command:
- the working that programs an arc in the space must be available (code= 2110).

Below an example of transform application:



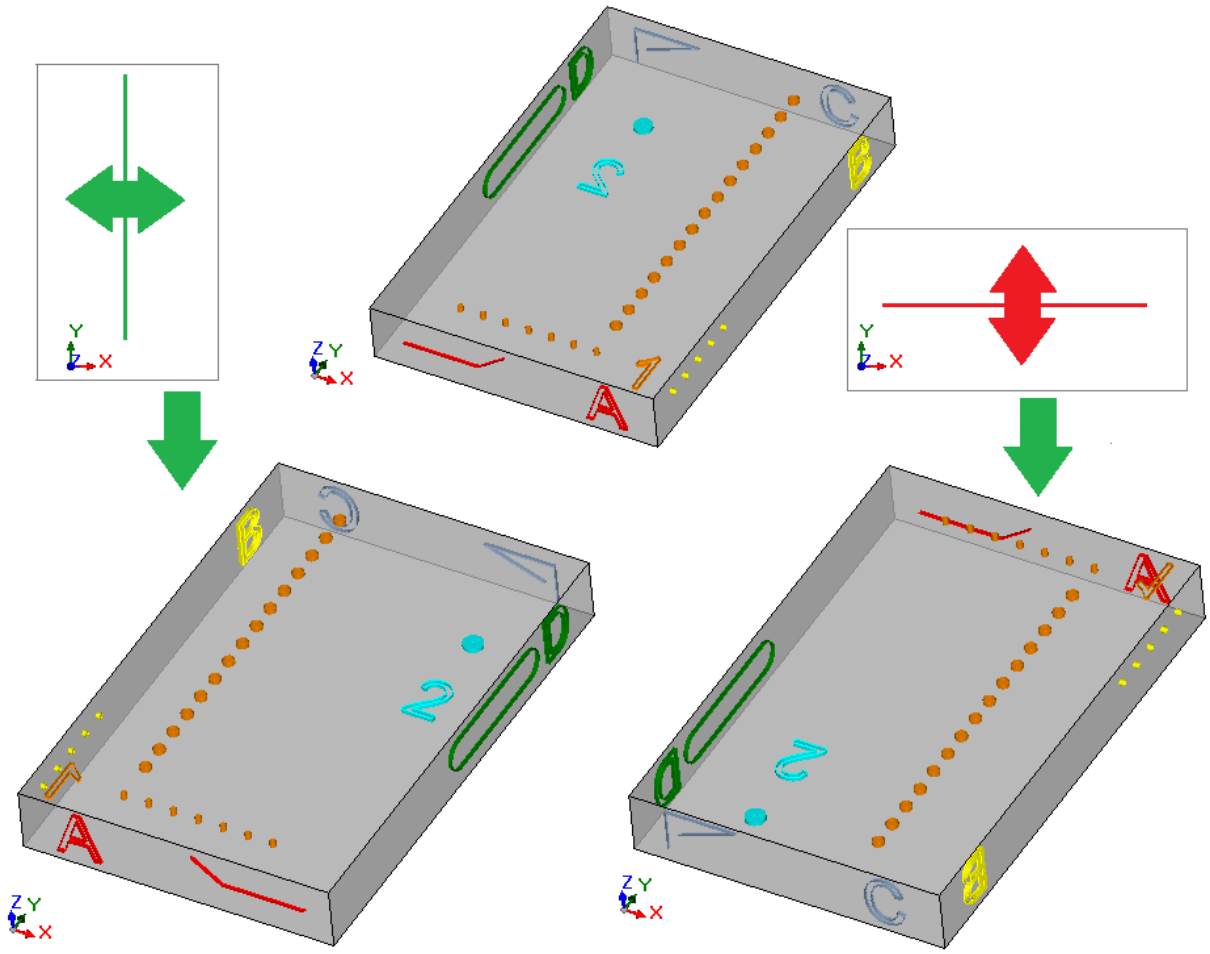
Effects of the transform:
- the workings of the faces 1 and 2 are rotated in such a way as to comply with the counterclockwise overall rotation of 90°
- the workings of the face 4 pass to the face 3;
- the workings of the face 5 pass to the face 4;
- the workings of the face 6 pass to the face 5;

- the workings of the face 3 pass to the face 6;
- where needed, a mirrored overall transform is applied to the face length;
- the dimensions of the piece exchange length and height.

Mirror the piece

 This option applies a mirroring of the piece around a vertical or a horizontal symmetry axis.

Below an example of transform application:

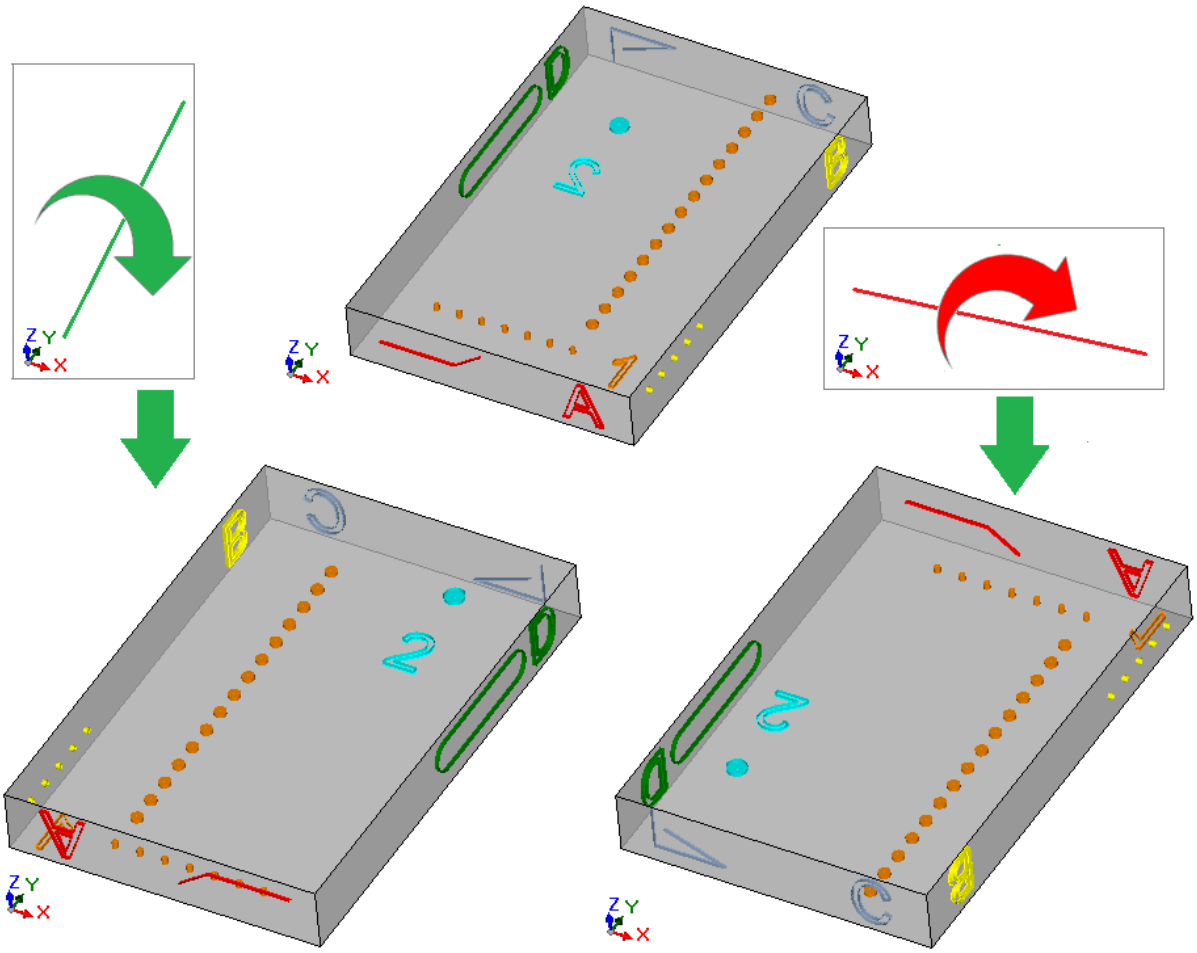


Effects of the transform:
- This option is applied to the workings of the faces 1 and 2 an overall mirrored transform with respect to a vertical or horizontal symmetry axis, as required;
- the workings are exchanged between the pairs of side faces (4 and 6) or (3 and 5), as needed;
- where needed, a mirrored overall transform is applied to the face length.

Overturn the piece

 This option applies an overturning of the piece around a vertical or a horizontal symmetry axis.

Below an example of transform application:



Effects of the transform:

- the workings are exchanged between the pairs 1 and 2;
- Then, this option applies to the working of the faces 1 and 2 an mirrored overall transform with respect to a vertical of horizontal symmetry axis, as required;
- the workings are exchanged between the pairs of side faces (4 and 6) or (3 and 5), as needed;
- where needed, a mirrored overall transform is applied to the length and/or on the height of the faces.

11 Parametric Programming

11.1 Introduction

Program assignments usually enable a parametric setting.
Let us consider for example program variables of "o" and "v" type. They are numeric variables whose setting field can generally assign a number or a numeric expression.
A greater specialization is required for "r" variables. The "r" variable type is not fixed but it can be assigned between two numeric-type variables (Double and Integer) and one non-numeric-type variable (String).
The Double type configures the "r" variable in the same way as an "o" or "v" variable (for these last the variable type assignment is automatic). The setting field can generally assign a number or a numeric expression and the calculated value holds the decimal part.
In case of Integer type the setting field can generally assign a number or a numeric expression but the calculated value truncates the decimal part.
Let us set, for example, for an "r" variable, the expression: "1000/3":

- in case of Double type variable, the calculated value will be = 333.333333;
- in case of Integer type variable, the calculated value will be = 333.

In case of String type variable the setting field generally assigns an alphanumeric expression and also the value assigned to the variable is a string. The String-type variable is typically used for subroutine assignment, as we can see in the following configuration examples: "doors\prg1.abc", "Ciao".
The same considerations outlined for "r" variables are applied to working parameters: they apply to both the numeric (double and integer) and non-numeric types (string).
In any case the variable type selection is transparent during programming, since it is set in workings database assignment.

11.2 Variables and Numeric Parameters

A numeric expression is any expression which can be evaluated as a number. The elements of the expression can include any combination of keywords (the functions which can be used in parametric programming), variables (example: piece dimensions), constants (example: Greek pi) and operators (example: +, -, *, /,|) whose result is a number.

A numeric expression must be assigned:

- with lower-case characters;
- the use of spaces is limited to string functions or variable arguments;
- maximum allowed number of characters: 100.

Examples of numeric expression are the following:

- 20: the expression can be directly solved. It directly assigns the numerical value;
- (100+32)/2: it uses numbers, mathematical operators, brackets;
- r27+100: it uses numbers, variables, mathematical operators;
- sqrt[r27+r15]-r5: it uses variables, mathematical operators, single-argument mathematical function.

The meaning of the above-listed expressions is intuitive. Let us follow step-by-step how each expression is evaluated:

- (100+32)/2=(132)/2=132/2=66;
- (value of r27=50) -**>** r27+100=50+100=150;
- (value of: r27=50, r15=30, r5=-5) -**>** sqrt[r27+r15]-r5=sqrt[50+30]-(-5)=sqrt[80]-(-5)=9.944271-(-5)= 9.944271+5=14.944271.

Precedence of Operators
When an expression includes several operations, each part is evaluated and solved according to a pre-established order, defined "precedence among operators".
Mathematical and logical operators are evaluated on the basis of the order of precedence specified in the following list:

- Multiplication (*), divisions (/, #), module (%), step adjustment (?) and logical operators (&, |);
- Addition and subtraction (+, –).

When there are operators with the same order of precedence in an expression (example: multiplication and one division), each operation is evaluated in the order in which it appears, from left to right. Same for an addition and a subtraction within the same expression.
Using round brackets, it is possible to ignore the order of precedence and let some parts of an expression be evaluated before others; the maximum limit of nested brackets depends only on the maximum allowed string length (100 characters). Expressions in round brackets are evaluated first.
Inside the round brackets the normal operator precedence is respected.

Examples of expressions:
"2+3*4": runs before the multiplication and after the sum. It is as follows: "2+12"=14
"(2+3)*4": the use of round bracket changes the result. It is as follows: "(5)*4"=20.

11.3 Functions

The use of functions allows to make more complex computations that those allowed by operators. An example of function is "*sqrt[r27+r15]-r5*" which uses the *sqrt* mathematical function which calculates argument's square root.
Functions are divided into two categories:
- single-argument functions: an example is the *sqrt* function;
- multi-argument functions: an example is the *pown* function.

Single-argument functions can be used with two formalisms:
- numerical formalism: the argument is a positive number. Example "*sqrt25*": the argument (25) is written directly after the function name;
- non-numeric formalism: the argument is a negative number (example: -25) or it is expressed in parametric form (examples: "r25", "100-32"). Example "*sqrt[r25]*": now the argument is written in square brackets.

The non-numeric formalism is compulsory also for a few special single-argument functions, which belong to the References to piece variables.

Multi-argument functions can only use the non-numeric formalism, with *name[op1;op2;…;opn]* syntax:
- *name* is the function name. Example: **pown**;
- *[…]* they delimit the function operands
- *op1* first argument
- *;* separator between two arguments
- *op2* second argument
- …
- *opn* last argument.

The number of arguments of a multi-argument function can be fixed or variable: in the following paragraph we will examine in detail each single function, by giving particular attention to the required number of arguments and to which arguments it is necessary to assign and which not.
The way the syntax of a function is written is important to interpret the number and the use of arguments and reflects a general formalism. Let us see a few examples:
- *pown[nb;ne]* 2-argument function: both of them shall be assigned
- *min[n1;…;n30]* function with variable number of arguments: the allowable number ranges between 1 and 30;
- *case[nc;nc1:nv1;nc2:nv2;...;nvdef]* function with variable number of arguments: the first 3 (*nc;nc1:nv1;nc2:nv2*) shall be assigned, then follows a number of optional arguments (…;) and the last assigned (*nvdef*) has a particular interpretation;
- *prmac[(nm); nkind;(vdef)]* the 1st and the 3rd parameters are in round brackets (*nm*), (*vdef*): this means that the argument can be assigned empty (in this case: the function applies a default value). Since *vdef* is the last argument of the function it is also possible not to assign it at all.

There is no limit in terms of function nesting: it depends only on the maximum allowable string length (100 characters).

11.4 Variables and String Parameters

Examples of alphanumeric expression are the following:
"doors\prg1.abc": the expression can be directly solved, with a value coinciding with (string)
"doors*r1.abc": it uses variable (r1)
"qx=*r1.*r2": it uses variables (r1, r2)
"*pr[r45+5]": it uses variable (r45), variable reference function.

The meaning of the expressions is less intuitive than the case of numeric expressions. Let us examine step-by-step how each expression is evaluated:
- (r1 is a string variable, with value="prg1") -> "doors*r1.abc"="doors\prg1.abc"
- (r1 and r2 are numeric variables, with value = 123 and 45) -> "qx=*r1.*r2" = "qx=123.45"
- (r45 is a numeric variable, with value = 2) -> "*pr[r45+5]" = "*pr[2+5]" = "*pr[7]" -> (r7 is a string variable, with value = "prg1") -> = "prg1".

An alphanumeric expression can be assigned:
- also with upper-case characters;
- the use of spaces is allowed (except for start and end spaces);
- characters between ' ' (space) and '}' (decimal values between 32 and 125) can always be used. They are characters of general visualization, regardless of the international settings of the operating system: digits (0-9), lower-case letters (a-z), upper-case letters (A-Z), punctuation marks (ex: .,;:?!), arithmetic operators (ex:+ - * /<>#%), brackets (ex: []{}())

- there are no limitations on the characters which can be entered, including the possibility to use all the specific characters of the various system settings in addition to the Unicode characters (see: Japanese, Chinese, Arabic characters, …).

While the formalism of a numeric expression fully satisfies the general solution criteria of an expression, an alphanumeric expression is interpreted according to some predefined formalisms (partially described above and resolved), that must be observed:

- **"doors*r1.abc"**
  in this formalism the "*rn" expressions have a parametric interpretation, where "n" specifies the "r" variable to use (n=0-299).
  In the example:
  - if r1 is a string-type variable, in this case the value (string) of r1 is replaced at the "*r1" expression, as stated above;
  - but if r1 is a numeric-type variable, in this case the string corresponding to the entire part of the r1 value is replaced at the "*r1" expression;
  - in case of r1 unassigned variable, in this case the "0" string is replaced at the "*r1" expression.

  There is no limit in the number of replacements. Thus, for example the following assignments are valid:
  "doors*r1.*r3"
  "abc*r5*r1.*r3":
  The string "abc*r500" recognizes no parametric form.

  It is also possible to extract a part of the addressed string by a "*rn" expression.
  Syntax: "…*rn[ni;nc]…" where:
  - n = r variable index (example: 5 for r5). It can only be numeric;
  - ni = start position from which the string assigned for r5 is read (significant from 1). It can be assigned:
    - numeric (example: ni=3);
    - with numeric-type r variable (example: ni=r2);
    - with variable j (example: ni=j5);
    - with $ variable -if in macro text- (example: ni=$0);
  - nc = number of read characters, from ni (optional). It can be assigned:
    - numeric (example: ni=3);
    - with numeric-type r variable (example: ni=r2);
    - with j variable (example: ni=j5);
    - with $ variable, if in macro text, (example: ni=$0).

    Furthermore, it is also possible to handle the use of symbolic names for r variables, in the two following forms:
  - "……*r\name\…" **ATTENTION**: the symbolic name must end at the '\' character
  - "……*r\name[ni;nc]…" **ATTENTION**: here the symbolic name must end at the '[' character

  Example: "door leaves\ *r5[3;1].cnc"
  let r5 be the assigned string variable = "abcdef";
  ni=3: it reads r5 from the third character;
  nc=1: it reads 1 character;
  -> the solution is "door leaves\c.cnc".

  Example: "door leaves\ *r5[3].cnc"
  let r5 be the assigned string variable = "abcdef";
  ni=3: it reads r5 from the third character;
  nc is unassigned: it does not truncate the string;
  -> the solution is "door leaves\cdef.cnc".

  Example: "door leaves\ *r5.cnc"
  let r5 be the assigned string variable = "abcdef";
  -> the solution is "door leaves\abcdef.cnc".

  Example: "door leaves\ *r\str1\.cnc"
  let r5 be the assigned string variable = "abcdef", with name ="str1";
  -> the solution is "door leaves\abcdef.cnc".

  Example: "door leaves\ *r\pippo[3].cnc"
  let r5 be the assigned string variable = "abcdef", with name ="pippo";
  ni=3: it reads r5 from the third character;
  nc is unassigned: it does not truncate the string;
  -> the solution is "door leaves\cdef.cnc".

  When using a r numeric variable, it is also possible to request a formatting on the decimal part.
  Syntax: "... * rn[**d**;nc]..." where:
  - n = index (numerical only) of the variable r (example: 5 for r5);
  - d = assign letter 'd'
  - nc = number of decimal figures (non-significant figures are deleted)

Example: " *r5[d;4]"
be r5 numeric variable with a value = 123.4006;
nc=4: assigns the first 4 decimal figures -> as a result the solution is "123.4006".
if (nc=3): assigns the first 3 decimal figures -> as a result the solution is "123.4 (the first 2 figures are deleted because thy are not significant).

- **"*pr[r45]"**
  this second formalism is more rigorous than the previous. In fact, it interprets only the "*pr[.....]" form, where the pr[..] function argument can assign any numeric expression.
  The solution of the pr[..] function argument is a numerical value of Integer type (n), which identifies in its turn a rn variable (n is the index of the variable).
  rn is normally a string type variable; it follows that the rn's value (string) assigns the string value of the alphanumeric expression.
  But if rn is a numeric type variable, it follows that the string corresponding to the entire part of the rn's value assigns the string value of the alphanumeric expression. Let us consider the following example:
  r3 variable of numeric format = 250.8
  r5 string variable = *pr[3]= "250";
  If rn is not assigned, the "0" string is replaced at the *pr[……]" expression.

- **"*p[...]"**
  formalism similar to the previous one, where the argument of the function p[..] can assign any numerical expression.
  The solution of the argument has as a result a numerical value (n): the string that corresponds to the integer part of the value (n) assigns the string value of the expression.
  Example: "*p[1024/6]"
  1024/6=170.6666 -> the result is the string "170".

- **" *j1.*j2 "**
- **" *$1.*$2 "**
  in this formalism the expressions "*jn" (and "*$n"), have a parametric interpretation, where n specifies the variable "j" (or "$") to be used. The considerations already made for the analog formalism already examined for the "r" variables remains operative: in this case, the variables used are numerical only; so, the string that corresponds to the integer part.
  In a string expression the user can use at the same time the syntaxes concerning all the variable he can manage. Example: "doors\ *r5*j1.cnc"
  be r5 an assigned variable string = "abcdef";
  be j1=4
  -> as a result: solution "doors\abcdef4.cnc".

"*geo[sub;..]", "*geo[param;..]", "*geo[lparam;..]"
These formalisms interpret only the "*geo[.....]" form, where the argument of the geo[..] function can assign any numeric expression. The result of the solution of the geo[..] function used is the value of an information or of a working parameter. More specifically:
- in the event of information or string parameter, it corresponds to the result of the function
- in the event of numeric typology: the result is a string that corresponds to the integer portion of the value.

For the details concerning the use of these functions, please read the corresponding paragraph.

11.5 Numerical Formats of Special Use

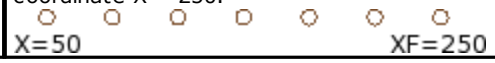
Let us examine here a form of special parameter prefix which, even if not directly used in programming, can be generated in tool application (rotation, mirroring, …).
It is about the "a;…" form of programming which can be set by working numeric parameters with meaning of coordinates.
Examples of valid assignments are the following:
"a;500" the parameter value is numeric;
"a;l/2" the parameter value itself is parametric.

The "a;…" form means that the corresponding coordinate is in absolute programming.
Let us consider for example an arc working: the centre coordinates are interpreted in relative positioning with respect to the arc starting point. It is possible to force an interpretation of coordinates in absolute by using the "a;…" form.

11.6 Expression Terms

Operators

Arithmetic

+	Addition. Example: 100.6 + 7 = 107.6
-	Subtraction. Example: 100.6 - 7 = 93.6
*	Multiplication. Example: 100 * 7 = 700
/	Division. The denominator cannot be zero. Error conditions: 125: null denominator; Example: 100/7 = 14.285714 Division between two operands.
%	Division modulus between two operands. Modulus (division reminder). The denominator cannot be zero. Example: 100 % 7 = 2 The computational procedure is as follows: • it does division (100/7) = 14.285714 • it separates the decimal part of the result: (14.285714 – 14) = 0.285714 • it multiplies by the divisor: 0.285714 * 7 = 2
#	Integer division. The denominator cannot be zero. Example: 100 # 7 = 14 The computational procedure is as follows: • it does division: 100/7 = 14.285714 • it separates the integer part of the result: integer (14.2857) = 14
?	Step adjustment between two operands. The denominator cannot be zero. Example: 100 ? 7 = 7.14285 The computational procedure is as follows: • it does division (100/7) = 14.285714 • it separates the decimal part of the result: (14.285714 – 14) = 0.285714 • it multiplies by the divisor: 0.285714 * 7 = 2 (= 100 % 7) • it divides the module by the integer of the division: 2/14 = 0.14285 • it adds to the divisor: 7 + 0.14285 = 7.14285. The first three points calculate the module. Then, the ? operator returns the divisor, modified so as to obtain an integer division result. In the particular case of a division result lower than 1, the operation returns the dividend. Example: 10 ? 15 = 10 {the solution is: 10/15= 0.6666} Example: programming a X fitting in the particular case, in which the final X coordinate coincides with the drilling position and it is necessary adjust the pitch between the two holes. Example initial x = 50, final, final x = 250, Pitch = 200 ? 32. The resulting pitch is 33.33 and the last hole is made at the coordinate X = 250. 

Logical

Operators of advanced programming have to be considered.

&	Bit-to-bit AND of the integral part of the operands, with truncation of the decimal part of the result. Example: 10.456 & 3.56 = 10 & 3 = 2 In fact, the bit representations of 10 and 3 are considered: 10 = 1 0 **1** 0 and 3 = 0 0 1 1 0 0 1 0 = 2 decimal value The representation in bits of the result keeps the bits that are at 1 set at 1 in both operands.
|	Bit-to-bit OR of the integral part of the operands, with truncation of the decimal part of the result. Example: 10.456 | 3.56 = 10 | 3 = 11 In fact, the bit representations of 10 and 3 are considered: 10 = **1** 0 **1** 0

	or 3 = 0 0 1 1 1 0 1 1 = 11 decimal value The representation in bits of the result keeps the bits that are at 1 set at 1 in one or both of the operands.

Brackets, Separators

(..)	Level of brackets: they can be nested without limits. The expression is solved starting from the inner brackets. Example: "12*((r0+r3)*sqrt[12])" • 1st operation: (r0+r3) {e.g. = 10} • 2nd operation: (10*sqrt[12])=34.64 • 3rd operation: 12*34.64=415.6921.
[..]	Delimiters for the arguments of a function can be used for: • parametric or negative function argument; • assignment of multi-operand function. Examples: sqrt[r12] single-argument function with parametric argument sin[-45] single-argument function with negative argument min[r12;1;67] multi-argument function
. ,	Separator between integer and decimal part of a numeric argument. The separator to use is only one and is indicated in TpaCAD configuration Examples: 128.6 .965
;	Separator between arguments of multi-argument function. Example: pown[5;2]
"..."	Direct string assignment (e.g., for function : strcmp). The space character is also allowed. Example: strcmp[5; "pippo"] evaluates r5 and compares with string "pippo"; strcmp[5; "ciao ..."] evaluates r5 and compares with string "ciao ..."

Variable Arguments

General arguments

pi	Greek pi (π = 3.1415...). Usable: always.
eps	Threshold (epsilon) for linear position, takes a value conforming to the current unit of measure of the program: • 0.001 for mm • 0.001/25.4 for inch multiplied by the Epsilon Multiplier factor (assigned in TpaCAD configuration) Usable: always. Example: • with program in [mm], multiplication factor = 10 -> eps is equal to: 0.001*10 = 0.01 • with program in [in], multiplication factor = 10 -> eps is equal to: 0.001*10/25.4 = 0,0003937
cnq	Conversion factor for linear position, takes value conforming to the current unit of measure of the program: • 1 for mm • 1/25.4 for in Usable: always. The cnq variable argument is to be used in writing subroutines and/or macro-programs, for comparisons and/or direct assignments with coordinate (or speed) values, when it is predictable that the subroutine (or macro-program) can be used in a program indifferently written in [mm] or [in] units. Example: • a subroutine (ONE) is written in [mm] units • the subroutine executes drilling works, spaced each other by an offset in x • the distance between holes is assigned in an r variable (re-assignable): example r3

<table>
<tr><td></td><td>• a minimum 20 mm distance is wished to be applied: then, r3 is compared with number 20.
There are no problems if also the program which applies ONE is written in [mm].
If, on the contrary, the program which applies ONE is written in [in]: r3 is now set to [in]. In this case it is no longer possible to compare r3 directly with 20. Both cases are valid if the comparison is made with "20*cnq", which is:
• 20 if the program is written in [mm] units
• 20/25.4 = 0.7874 if the program is written in [in] units</td></tr>
<tr><td>cnf</td><td>Linear speed conversion factor, it assumes a value depending on the unit of measurement of the program:
• 1 for mm
• 0.6561 for inch, with speeds in: m/min or in/sec
• $6.561*10^{-4}$ for inch, with speeds in: mm/min or in/sec
• 1000/25.4 for inch, with speeds in: m/min or in/min
• 1/25.4 for inch, with speeds in: mm/min or in/min

Usable: always.
The use of the cnf argument is analogous to cnq, only here the conversion concerns a linear speed value.</td></tr>
<tr><td>l
h
s</td><td>Piece dimensions: "l" is the length, "h" is the height, "s" is the thickness.
The values of the arguments always correspond to the dimensions of the piece in programming, even if, for example, used within call to the subroutine.
Usable: always.</td></tr>
<tr><td>face</td><td>Number of the current face.
We distinguish between the situations of use:
• in working parameter: it returns the number of the face the working is applied to (value from 1 to 6, if real face; from 7 to 99, if fictive face; from 101 to 500, if automatic face):
  • in case of real face: it is about the face custom number.
  • in case of piece-face: it matches the F field (in case of application in automatic face: it returns the number assigned to the automatic face – example: 120)
• in case of variable list (o, v, r), assignment of fictive face or custom section: it returns the value -1.
The argument value always corresponds to the current face of the piece in programming, even if, for example, used within call to the subroutine.
Usable: always.</td></tr>
<tr><td>face0</td><td>The argument shows if it is face-piece schedule:
• 1 checked condition
• 0 otherwise.
Usable: always.</td></tr>
<tr><td>faceauto</td><td>This argument returns the number of the automatic face.
The value is significant in face-piece programming only; in case of an automatic face previously before, a value between 101 and 500 is returned.
Usable: always.</td></tr>
<tr><td>lf
hf
sf</td><td>Current face dimensions: "lf" is the length, "hf" is the height, "sf" is the thickness.

Alternative one-letter options can also be used: "x" is length, "y" is height, "z" is thickness.

There are several conditions of use:
• in working parameter: they return the dimensions of the face the working is applied to: they are the dimensions of the face that corresponds to the value of "face".
• in case of variable list (o, v, r), assignment of fictive face or custom section: They return the corresponding dimension of the piece ("lf" is equal to "l", "hf" is equal to "h", "sf" is equal to "s").
The argument value always corresponds to the current face of the piece in programming, even if, for example, used within call to the subroutine.
Usable: always.
Example:
• the dimensions of the piece are assigned as l*h*s=1000*450*18
• let us see how much the three variable arguments for the six faces of the piece are worth

<table>
<tr><th></th><th>Face 1</th><th>Face 2</th><th>Face 3</th><th>Face 4</th><th>Face 5</th><th>Face 6</th></tr>
<tr><td>lf</td><td>1000</td><td>1000</td><td>1000</td><td>450</td><td>1000</td><td>450</td></tr>
<tr><td>hf</td><td>450</td><td>450</td><td>18</td><td>18</td><td>18</td><td>18</td></tr>
<tr><td>sf</td><td>18</td><td>18</td><td>450</td><td>1000</td><td>450</td><td>1000</td></tr>
</table>
</td></tr>
<tr><td>prgt</td><td>Piece type: 0=program, 1=subroutine, 2=macro.</td></tr>
</table>

	The argument value always corresponds to the access level of the piece in programming, even if, for example, used within call to the subroutine. Usable: always.
prgrd	Access piece level: 0=Operator level, 1=Installer level, 2=Manufacturer level. The argument value always corresponds to the access level of the piece in programming, even if, for example, used within call to the subroutine. Usable: always.
prgwr	Edit piece level: 0=Operator level, 1=Installer level, 2=Manufacturer level. The argument value always corresponds to the level of change to the piece in programming, even if, for example, used within call to the subroutine. Usable: always.
prgnum	Progressive number of the last programmed working in the list of the current face, not of the comment. In the list of variables (o, v, r), assignment of virtual faces or custom section: is 0. Usable: always.
'ch'	Replaces a numeric decimal value corresponding to the ASCII coding of the ch. character. Valid codes: from 32 (' ') to 125 ('}'). The upper-case letters are converted into lower-case letters. Usable: always. Example: 120+'a' = 120+97=217 'a'-' ' = 97-32=65

Execution mode

prgrun	Program execution environment: 0 = edit (TpaCAD environment); 1 = running. The prgrun argument can be useful to differentiate custom error messages: • it is possible to implement error messages relative to technological malfunctions only in Execution mode; • lastly, completely differentiate the development of a program (typically: a macro) in the two environments. Usable: always.
prgdraw	Active work environment: in the case of prgrun=1 it allows to distinguish between different situations of execution. More specifically, a positive value (1) shows that a mode of "graphic preview" is active. A typical application of the argument allows distinguishing the program compilation, taking into account operational or graphic requirements. Usable: always.
prgn	Flag of normal execution: 1 = normal execution; 0 = different execution. It corresponds to the field *Execution,* as assigned in Execution mode. The prgn argument allows differentiating the execution of a program on the basis of the way the program itself is deployed. Examples: to execute or not some workings, to assign the direction of execution of a sawing work. Usable: always.
prgx	Flag of mirrored execution X: 1 = mirrored execution X; 0 = different execution. It corresponds to the field *Execution,* as assigned in Execution mode. The prgx argument allows differentiating the program assignment based on the basis of how the program itself is deployed. Usable: always.
prgy	Flag of mirrored execution Y: 1 = mirrored execution Y; 0 = different execution. It corresponds to the field *Execution,* as assigned in Execution mode. The prgy argument allows differentiating the program assignment based on the basis of how the program itself is deployed. Usable: always.
prgxy	Flag of mirrored execution XY: 1 = mirrored execution XY; 0 = different execution. It corresponds to the field *Execution*, as assigned in Execution mode. The prgxy argument allows differentiating the program assignment based on the basis of how the program itself is deployed. Usable: always.
prarea	Execution area. It corresponds to the field *Work area*, as assigned in Execution mode. The prarea argument allows differentiating the program assignment based on the basis of how the program itself is deployed.

	Usable: always.
prqx prqy prqz	X, Y, Z Step in execution area. The values correspond with the *Locator offsets in work area*, as assigned in Execution mode. The prqx/y/z arguments allow to differentiate the program assignment based on the basis of how the program itself is deployed. Usable: always.
prun1 prun2 prun3 prun4 prun5 prun6 prun7 prun8	Additional parameters in execution. In edit mode (prgrun = 0), the parameters have always the same value 0. In Execution mode (prgrun = 1): they can be assigned with significant value for processing and execution of the program. Usable: always.

Environment Settings

The arguments can be used during the writing of subroutines and/or macro-programs, to compare and/or direct assignments of values (coordinates, rotation axes), where it is expected that the subroutine (or macro-program) can be used in a non-predefined configuration.

These should be considered as advanced programming arguments.

sysface	Face geometry. 0= faces in transparency 1= custom systems Usable: always.
sysquad	Operating quadrant: value from 1 to 4. Usable: always.
sysz	Indicates the direction of the z axis applied to face: 0 = negative values enter in the piece. 1 = positive values enter in the piece. Usable: always.
sysxz	Indicates the arc typology on the xz-plane of face: 0 = in case of the X-axis is orthogonal to the plane (XZ plane); 1 = in case of the Z-axis is orthogonal to the plane (XZ plane). Usable: always.
sysbeta	Indicates the managed rotation for the rotation axis around the Y axis: 0 = in case positive towards semi-axis X+ of the absolute reference system 1 = in case positive towards semi-axis X- of the absolute reference system 2 = if the rotation occurs around the X axis (positive toward the Y-semi-axis of the absolute reference system). Usable: always.
sysfeed	Indicates the programming unit of the linear speed rates: • in a program with [mm] unit, possible selections are: 0=[mt/min] 1=[mm/min] • in a program with [in] unit, possible selections are: 0=[inch/sec] 1=[inch/min].
sysline syschord	The sysline arguments indicate the criterion assigned for the fragmentation of the arcs; 0 = the arcs are fragmented according to the assigned length 1 = application of the criterion of the chordal error. The syschord argument indicates: • the length of the arc fragmentation (first case) • the chordal error (second case) the value of syschord is always converted in unit of measure of the program. Usable: always.
syskey	The argument indicates the presence of any professional mode key: • 1 condition is fulfilled • 0 otherwise (in basic mode or function key demo)

	Usable: always.
sysname	This argument shows if the management of the working name field is enabled: • 1 condition verified • 0 otherwise (field not available) Usable: always.

Piece Variables

o0-o15 o\name	"o" variables of the piece. There are several conditions of use: • in program text or subroutine: usable are the only variables assigned managed (there may be a number of variables lower than 16, or at worst no variables at all); • in macro text: there are 16 variables and they can always be used; The second above-mentioned form corresponds to the symbolic formalism, with: • fixed part "o\"; • variable part: the symbolic name assigned to the variable. The values returned always correspond to the variables of the piece in programming, even if, for example, used within call to the subroutine. They cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions. Error conditions: • 114: use in invalid context; • 121: invalid index to "o" variable.
v0-v15 v\name	"v" variables of the piece. There are several conditions of use: • in program text or subroutine: the only assigned and manipulated variables can be used (there may be a number of variables lower than 16, or at worst no variables at all); • in macro text: there are 16 variables and they can always be used; The second above-mentioned form corresponds to the symbolic formalism, with: • fixed part "v\"; • variable part ("name"): the symbolic name assigned to the variable. The values returned always correspond to the variables of the piece in programming, even if, for example, used within call to the subroutine. It cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions. Error conditions: • 113: use in invalid context; • 120: invalid index to "v" variable.

r0-r299 r\name	"r" variables of the piece. They are always used, always 300 of them: an unassigned variable still has a numerical value of 0.0. The second form shown corresponds with the symbolic formalism, recognized variable assignment or numeric parameter: • fixed part "r\"; • variable part ("name"): the symbolic name assigned to the variable. They cannot be used: • in assignment of 'o', 'v' variables • in assignment of custom functions. Error conditions: • 112: use of r variable in invalid context; • 117: invalid index to "r" variable • 102: in case of "**...*rn[ni;nc]...**" format, with ni or nc assigned with invalid syntax (see also: Variables and numerical-Type Parameters). The interpretation of return values requires a distinction between different situations of use (the considerations made here apply to each function that uses the variables r): • in case of r variable list: they are read from the list; • in case of application of a machining program directly in the face of the piece in programming: they are read from r variable of the piece itself;

	• in case of internal developing of a complex machining (see: interpreted machining performed to a subroutine or macro): • to assign a re-assignable variable to subroutine or machining parameter: research variables on levels of expansion upstream (at most: up to the list of r variables of the main program); • to assign a non re-assignable variable to subroutine: search variables on levels of expansion upstream (at most: up to the list of r variables of the main program), but starting from its level. Example of assignment of string r variable: "doors*r28.*r29" where: r28 is a string variable with ="p007" (string) value r29 is a numeric variable, with =12.5 (numeric) value the computation of the expression will result in the following (string) value: ="doors\p007.12".
j0-j99	Program global variables. There are several conditions of use: • in working parameter: it applies the real values of j variables; • in the list of program variables of "r" type: it applies always null values. They cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges). Error conditions: • 115: use in invalid context; • 118: invalid index to "j" variable.
$0-$299	Auxiliary variables: they can be used only in writing a macro-program. They cannot be used: • in 'o', 'v', 'r' variable assignment; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges); • in program text. Error conditions: • 111: use in invalid context; • 119: invalid index to "$" variable.

References to Piece Variables

They are parametric forms that allow to synthesize the reading of program variables.
They are normally used to write macros. They should be considered as advanced programming forms.

pr[.]	Reference to r variable. The parametric form (use of [.] brackets) is compulsory and the expression is evaluated by taking the rn variable value, with n=value calculated in square brackets. In variable or numeric parameter assignment: it returns the numeric value of the rn variable (0.0 in case of non-numeric variable); In variable or non-numeric parameter (string) assignment, if the "*pr[..]" remarkable form is recognized: • if rn identifies a string variable: it returns the corresponding "$" variable calculated; • if rn identifies a numeric variable: it returns the "$" variable corresponding to the integer part of the variable value; • if rn identifies an unassigned variable: it returns the string "0". It cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions. Error conditions: • 112: use of r variable in invalid context; • 117: invalid index to "r" variable. Examples: pr[12]: it returns the r12 value pr[10+5]: it returns the r15 value pr[r1], with r1=7: it returns the r7 value.
pj[.]	Reference to a program j variable. The parametric form (use of [.] brackets) is compulsory and the expression is evaluated by taking the jn variable value, with n=value calculated in [.].

	There are several conditions of use: • in working parameter: it applies the real values of j variables; • in the list of program variables of "r" type: it applies always null values. It cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges). Error conditions: • 115: use in invalid context; • 118: invalid index to "j" variable. Examples: pj[12]: it returns the j12 value pj[10+5]: it returns the j15 value pj[r1], with r1=7: it returns the j7 value.
p$[.]	Reference to auxiliary variable in macro-program text. The parametric form (use of [.] brackets) is compulsory and the expression is evaluated by taking the $n variable value, with n=value calculated in [.]. The function can only be used in writing a macro-program. It cannot be used: • in 'r', 'o', 'v' variable assignment; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges); • in program text. Error conditions: • 111: use in invalid context; • 119: invalid index to "$" variable. Examples: p$[12]: it returns the $12 value p$[10+5]: it returns the $15 value p$[r1], with r1=7: it returns the $7 value.

Assignments relative to the application of subroutine or macro

They are arguments returning information about subroutine or macros application and whose use has to be made in the text of the same subroutine or macro.

subx suby subz	Return the x, y, z positioning coordinates: • in cycle application (subroutine or macro-program), they take the value of the application point with resolved point hook and relative programming; • in 'r' variable assignment: they take value 0.0. The subx/y/z arguments allow the user to know the application point set in a subroutine call, within the subroutine itself. They cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges). Error conditions: • 109: use in invalid context;
subang subang0	Return the rotation angle: • in cycle application, they take the rotation angle value; • in 'r' variable assignment: they take value 0.0. The subang argument (suba synthetic form is permitted) returns the rotation value set in a subroutine call, within the subroutine itself. The subang0 argument returns the rotation value set in the subroutine call within the subroutine itself, but including all the possible nested calls. Let us consider the example of a subroutine call with a 20.0° rotation, which in turn develops a second call with a -5.0° rotation: at the innermost level the resulting rotation (returned by subang0) will be equal to 20.0°+(-5.0°)=15.0°. They cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions;

	• in assignment of variable geometries (fictive face edges). Error conditions: • 109: use in invalid context;
subinv subinv0	Return the assignment of inversion: • in cycle application, they take the inversion parameter value (1 if required); • in 'r' variable assignment: they take value 0. The subinv argument (subi synthetic form is permitted) allows to know if the subroutine call has required the inversion of execution, within the subroutine itself. The subinv0 argument returns the value of the inversion set in the subroutine call, within the subroutine itself, but including all the possible nested calls. They cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges). Error conditions: • 109: use in invalid context.
submir submir0	Return the assignment of mirror: • in cycle application, they take the value corresponding to the required mirrored execution (0=inactive; 1= mirror x; 2= mirror y; 3= mirror x+y); • in 'r' variable assignment: they take value 0. The submir argument (subm synthetic form is permitted) allows to know if a subroutine call has required a mirrored execution, within the subroutine itself. The argument submir0 allows the user to know if subroutine call has required a symmetry of the execution, within the subroutine itself, but including all the possible previous calls. Let us consider the example of a subroutine call where x mirror is active, which in turn develops a second call where x+y mirror is active: at the innermost level the resulting mirror enquiry will be an y mirror only. They cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges). Error conditions: • 109: use in invalid context.
sublink	Returns the assignment of point hook: • in cycle application, they take value 1 if a point hook is required; • in 'r' variable assignment: they take value 0. The sublink argument allows the user to know if a subroutine call has required a point hook of the execution, within the subroutine itself. The synthetic form is allowed: subl. It cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges). Error conditions: • 109: use in invalid context.
substr substr0	Return the assignment of stretch: • in cycle application, it takes the required stretch value; • in 'r' variable assignment: they take value 1.0. The substr (subs synthetic form is permitted) argument allows to know if a subroutine call has required a resizing of the execution, within the subroutine itself. The substr0 argument allows the user to know if the subroutine call has required a resizing of the execution, within the subroutine itself, but including all the nested calls. Let us consider the example of a subroutine call with a stretch factor equal to 2.0, which in turn develops a second call with stretch factor equal to 0.5: at the innermost level the resulting stretch factor will be equal to 2.0*0.5=1.0. They cannot be used:

	• in assignment of 'o', 'v' variables; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges). Error conditions: • 109: use in invalid context.
subemp	Returns the assignment of emptying: • in cycle application, they take value 1 if emptying is required; • in 'r' variable assignment: they take value 0. The argument allows to know if a subroutine call has required applying emptying operations, within the subroutine itself. The synthetic form is allowed: sube. It cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges). Error conditions: • 109: use in invalid context.
subface	Returns the applied face. • in cycle application, they take the number value of the applied face (in case of real face number, it is the face custom number); • in 'r' variable assignment: they take value -1. The synthetic form is allowed: subf. They cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges). Error conditions: • 109: use in invalid context.
submaster	Returns multiple call information: • in cycle application, it takes the value: • -2: if it corresponds to a master call; • >0: if it corresponds to an induced call and returns the master face number (if it is a real face number, it is the face custom number); • -1 in any other case; • in "r" variable assignment: it takes value -1. It cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges). Error conditions: • 109: use in invalid context.
subxm subym subzm	Return the placement coordinate (x,y,z) of the application point of the **master call**, relative to the case of multiple case. As also for the arguments (subx, suby, subz): • in cycle application (subprogram or macroprogram), they take the value of the application point, with hook point and relative programming resolved; • in assignment of 'r' variable: they take value 0.0. If the **submaster** argument returns: • -2: the returned value corresponds to (subx, suby, subz); • >0 if it corresponds to induced call and the returned value return the field (subx, suby, subz) as applied on the master face. They cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges). Error conditions: • 109: use in invalid context.
subvl subvb	Return the value assigned to the property field: L (subvl)

subvo subvm subvk subvk1 subvk2	B (subvb) O (subvo) M (subvm) K (subvk) K1 (subvk1) K2 (subvk2) • in cycle application, they take the property value; • in 'r' variable assignment they take value 0. They cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges). Error conditions: • 109: use in invalid context.
rempty[nn]	Verifies that the r variable, whose index is indicated by nn, is assigned. It returns: • 0 if the variable is assigned; • 1, in any other case. The parametric form is obligatory (use of the [] brackets). If the argument of the function is parametric and has a "rn" or "r\name" remarkable format, the function works directly on the rn variable. The following situations may occur: • in case of subroutine or macro, it returns value 0, if the r variable of the cycle is assigned in the call or if it corresponds to a variable of the subroutine or of the non-re-assignable macro; • in the assignation of r variable or not in the cycle application, it takes value 0, if the r variable of the program is assigned. It cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions; Error conditions • 112: use of r variable in invalid context; • 117: invalid index to "r" variable. Examples (uses in subroutine development): rempty[12], rempty[r12]: it returns 0 if variable r12 is assigned, otherwise 1
rempty[nn;valdef;(nmin);(nmax)]	Verifies that the r variable, whose index is indicated by nn is assigned, returning a value for the variable: • if the variable is assigned: the assigned value may be linked back to a maximum range of values; • in any other case: a default value. If the argument of the function is parametric and has a "rn" or "r\name" remarkable format the function operates directly on the rn variable. Arguments: nn = index or name of r variable. The value of nn must be included between 0 and 299; valdef = default value to be returned if the variable is not assigned nmin = minimum value allowed (apply if assigned) nmax = maximum value allowed (apply if assigned) • no check is made on the correctness of the extreme values: it can be nmin > nmax; • (nmin, nmax) are only used if the variable is assigned The function is intended to operate for variables of numerical type, otherwise: • if assigned, the associated value is null (= 0.0) The same consideration of use as for the previous prototype are applied. It cannot be used: • in the assignment of a 'o', 'v' • in the assignment of custom functions. Error situations: • 112: use of r variable in an invalid context

	• 117: index of variable r is not valid • 124: number of operands =4 • 130: valdef argument omitted (empty assignment). Example (uses in subroutine development): rempty[r12; 32;1;96]: below are the different cases • if variable r12 is not assigned -> 32. Otherwise • with value of r12 in range (1; 96) -> r12 • with value of r12 lower than 1 -> 1 • with value of r12 higher than 96 -> 96.
rvalue[cnd;nn;value]	Requests a value to be assigned to r variable, whose index is determined by nn: • the function must be **alone** in the assignment field • the function result is actually applied only when the subroutine (or the macro-program) is used • the function can be used during the assignment of a private and numeric r variable of subroutine. The function return is not significant • the variable for which the value assignment is requested (determined by argument: nn) must be in the subroutine itself, must correspond to an index lower than that of the variable in question and must be: public, numeric, not assigned (i.e.: return of function rempty[nn] has value 1) Arguments: cnd: assignment flag (0 = off = does not apply; 1 = on = applies) nn = r variable index or name. The value of nn must be within 0 and 299. As for the allowed formats, see function (rempty); value = value to be assigned to variable (nn). It cannot be used: • assigning 'o', 'v' variables • assigning custom functions Error situations: • 112: use of r variable in an invalid context; • 117: invalid index of r variable • 124: number of operands # 3. Example (use in subroutine development): rvalue[r5; 10; 96]: the following cases can occur • if the variable r5 has value 0 -> the function has no effect. Equivalent situation if: • the function is used assigning a variable whose index is less than 10 (e.g.: r8), or is public or a string • the subroutine does not assign the variable r10, or r10 is private or a string, or has a valid assignment. Otherwise: • the variable r10 will be assigned the numeric value 96 • any use of the function "rempty[10]" always return value 1. The function shall be used only when strictly necessary.

Setting of custom sections

The arguments allow a direct access to the data assigned in the custom sections, limited to the numerical items (whole numbers or numbers with comma, selection from the list).
If the section or the option indicated is not assigned, the argument takes a null value (0).
Its use is in subroutines and/or macro-programs writings, that must be considered as advanced programming forms.

The arguments cannot be used in:
- variable assignment (‘o’, ‘v’, ‘r’)
- variable geometry assignment
- in assignment of custom section
- in assignment of custom functions.

szs\name	Returns a numeric item in the Special settings section: • fixed part "szs\"; • variable part ("name"): the symbolic name assigned for the item in the section. Example: "szs\aaa": returns the field value named "aaa".
szi\name	Returns a numeric item in the section of Additional info
szo\name	Returns a numeric item in the section of Optimization settings
szl\name	Returns a numeric item in the section of Constraint settings

Global Variables

The use of the Global Variables must be specifically enabled in the configuration of TpaCAD.
They are strictly numerical variables, they are no more than 300 and can only be recalled by name.
Also the list of the Global variables is assigned in TpaCAD configuration: this is an information that cannot be modified in the program.

glb\name	returns the assigned value to the corresponding Global Variable. • fixed part "glb\"; • variable part ("name"): symbolic name of the variable. Usable: always. If the variable is not assigned, a generic syntax error appears.

Auxiliary functions

They are normally used to write subroutines and/or macro-programs.
They should be considered as forms of advanced programming.

nfa	Returns the real number of the (custom) face specified as argument. The function operates in case of argument of value included between 1 and 6: the argument is interpreted as face custom number and the function returns the face real number. Otherwise: it returns the integer part of the argument in any case. Usable: always. Error conditions: none. Examples: assign a face custom numbering as follows: face 1 -> custom number: 5 face 2 -> custom number: 6 face 3 -> custom number: 1 face 4 -> custom number: 4 face 5 -> custom number: 2 face 6 -> custom number: 3 nfa5=nfa[5]=1 nfa2=nfa[2]=5
nfc	Returns the custom number of the (real) face specified as argument. The function operates in case of argument of value included between 1 and 6: the argument is interpreted as face real number and the function returns the face custom number. Otherwise: it returns the integer part of the argument in any case. Usable: always. Error conditions: none. Examples: (assign the face custom numbering as indicated for function: nfa) nfc1=nfc[1]=5 nfa2=nfc[2]=6

Mathematical functions

abs	Returns the absolute value of the argument.

	The synthetic form is allowed: a. Usable: always. Error conditions: none. Examples: abs5 = 5 abs[-5] = 5
sqrt	Extracts the square root of the argument. The value of the argument must be positive (>=0.0). The synthetic form is allowed: q. Usable: always. Error conditions: • 127: negative argument. Examples: sqrt[25] = 5 sqrt[-25] <- it causes error (127)
int	Returns the integer part of the argument obtained by truncation (remove the decimal part). The synthetic form is allowed: i. Usable: always. Error conditions: none. Examples: int[-12.8] = -12.0 int[12.9] = 12
inv	Returns the reciprocal of the argument (1/x). The argument cannot be null. The synthetic form is allowed: v. Usable: always. Error conditions: • 125: null argument. Examples: inv2 = 0.5 inv 0 <- it causes error (125)
pow	Squares the argument. The synthetic form is allowed: p. *Usable: always.* Error conditions: none. Examples: pow3 = 9 pow[-3] = 9 pow0 = 0
pown[nb;ne] pown[nb;ne1;ne2]	Raises the first argument (base) to the power resulting from the second argument (exponent): • nb (1° argument) = base • ne (2° argument) = exponent. It is applied to the integer part. • ne1,ne2 (2° argument and 3° argument) = the exponent is calculated as ne=(ne1/ne2) and used without truncation of the entire part. The arguments of this function can be 2 or 3. Particular cases: • nb#0.0, ne=0.0, it returns 1 • nb=0.0, ne=0.0, it returns 1 • ne2=0.0, uses the 2-argument form Usable: always. Error conditions: 123: number of operands =0; 124: number of operands #2,3; 128: nb=0.0, ne <0.0 (negative). Examples: pown[5;2] = 5 * 5 =25 pown[5;3.5] = pown[5;3.5;0] = 5 * 5 * 5 = 125 pown[5; 0] = 1 pown[0; 5] = 1 pown[5;1;2] = $5^{1/2}$ = 2.236 (mathematical equivalence: square root of (5^1)) pown[5;2;3] = $5^{2/3}$ = 2.924 (mathematical equivalence: square root of (5^2))
round	Rounds the argument to the nearest integer. Usable: always. Error conditions: none.

	Examples: round[12.8] = 13 round[12.3] = 12 **round[12.5] = 12 <- up to 0.5, rounded down** **round[12.501] = 13 <- over 0.5, rounded up** round[-10.3] = -10 round[-10.7] = -11
range[nval;(nmin);(nmax)]	Validates a value within a maximum interval of values. • nval (1° argument) = value to be validated • nmin (2° argument) = min. value allowed (apply, if assigned) • nmax (3° argument) = maximum value allowed (apply, if assigned) No check on the correctness of the extreme values is performed: it can be nmin > nmaximum Usable: always. Error situations: • 123: operand number = 0 • 124: operand number > 3; • 130: first argument omitted. Examples: range[5;0;10] returns 5 range[5;0;4] returns 4 (it limits to the maximum value) range[-5;0;4] returns 0 (it limits to the min. value) range[-5;;4] returns -5 (it does not apply the minimum value) range[5;0] returns 5 (it does not apply the maximum value)

odd	Returns 1 if the integer part of the argument is odd; otherwise 0. Usable: always. Error conditions: none. Examples: odd12.8 = odd12 = 0 odd13.8 = odd13 = 1
sign[nval]	Returns: • 1 if the argument is positive or null (≥ 0) • -1 if the argument is negative. Usable: always Error situations: • 123: number of operands =0; • 124: number or operands #1; Examples: sign[12.8] = 1 sign[-13.8] = -1
hypot[c1;c2] hypot[c1;c2; c3]	Returns the hypotenuse of the right triangle which has assigned legs. The arguments of the function can be 2 or 3. If the arguments are 3, the triangle is assigned in the space. Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands #2 and #3; Example: hypot[5;2;1] = sqrt[5*5+2*2+1*1] = 5.477 hypot[5;2;1] = sqrt[5*5+2*2+1*1] = 5.477
min[n1;..;n30] max[n1;..;n30] ave[n1;..;n30] sum[n1;..;n30]	Return the minimum, maximum, average value between the arguments or the sum of the arguments. The maximum number of arguments is 30. Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands >30. Examples: min[5;12;3;25] return 3 max[5;12;3;25] return 25 sum[5;12;3;25] return 5 + 12 + 3 + 25 = 45 ave[5;12;3;25] return (5 + 12 + 3 + 25)/4 = 11.25 <- divides the sum by the number of arguments

minr[n1;n2] maxr[n1;n2] aver[n1;n2] sumr[n1;n2]	Returns the minimum, maximum, average value or the sum between the values assigned to r variables in the (n1, n2) range. The arguments of this function must be 2. The concerned variables may be of any type: • numeric (double, integer): the function reads the value; • string: the function takes value 0.0. The n1 and n2 arguments must identify a range of variables included between r0 and r299. They cannot be used: • in assignment of 'o', 'v' variables • in assignment of custom functions. Error conditions: • 112: use of r variable in invalid context; • 123: number of operands =0; • 124: number of operands #2; • 117: invalid index to "r" variable. Example r10=minr[2;5], with values r2=5; r3=12; r4=3; r5=25: the function return 3; It is equivalent of: min[5;12;3;25]
minj[n1;n2] maxj[n1;n2] avej[n1;n2] sumj[n1;n2]	Return the minimum, maximum, average value or the sum between the values assigned to j variables in the (n1, n2) range. The arguments of this function must be 2. The n1 and n2 arguments must identify a range of variables included between j0 and j99. There are several conditions of use: • in working parameter: it applies the real values of j variables; • in the list of program variables of r type: it applies always null values; They cannot be used: • in assignment of 'o', 'v' variables • in assignment of custom functions • in assignment of variable geometries (fictive face edges). Error conditions: • 123: number of operands =0; • 124: number of operands #2; • 115: use in invalid context; • 118: invalid index to "j" variable. Example maxj[2;5] (with values j2=5; j3=12; j4=3; j5=25) -> the function returns 25.
min$[n1;n2] max$[n1;n2] ave$[n1;n2] sum$[n1;n2]	Return the minimum, maximum, average value or the sum between the values assigned to $ variables in the (n1, n2) range. The arguments of this function must be 2. The n1 and n2 arguments must identify a range of variables included between $0 and $299. They cannot be used: • in 'r', 'o', 'v' variable assignment; • in assignment of custom functions; • in assignment of variable geometries (fictive face edges); • in program text. Error conditions: • 123: number of operands =0; • 124: number of operands #2; • 111: use in invalid context; • 119: invalid index to "$" variable. Example ave$[2;5] (with values $2=5; $3=12; $4=3; $5=25) -> the function returns 11.25

Trigonometric Functions

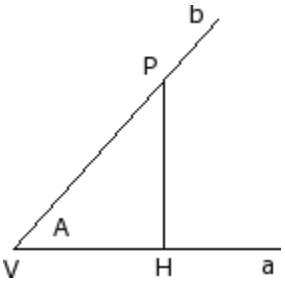
Outlines of trigonometry

The brief outlines of trigonometry given below provide a frame of reference to solve geometric problems, which recur in the programming.

The main unit of measurement of the plane angles are: The centesimal degree and the radiant angle.

In Mathematics the linear measure of the angles, whose unit of measurement is the radiant, is normally used; however, the most widely used unit of measurement of the angles is definitely the degree. For this reason, the following trigonometric functions require or return angular values expressed in degrees.

It is useful to remember: 1 radiant = (180/p)°, with (p = 3.1415..) known as pi (π).



The user should remember that an angle is defined as positive when it rotates in counterclockwise direction.
Let us consider an angle **(A)**: (in radiants) of vertex **V** and sides **a** and **b**.
Let us take on the half-line **b** any point **P** distinct from the vertex **V**. Let us project it on the half-line **a**: be **H** the point of the perpendicular described by **P** on **a**.

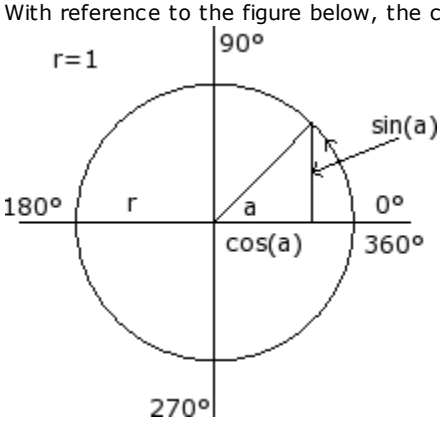
Now, let us consider the right triangle **VHP** and the ratio between the oriented segments:
HP/VP; VH/VP; HP/ VH
It is shown that these ratios only depend on the angle A and not on the point P chosen on the half-line b.

The three written ratios define the three functions of the angle **A** called:

sine of **A**	$\frac{HP}{VP} = sin\ A$ more specifically with **VP**=1, is: **HP**=sin**A**
cosine of **A**	$\frac{VH}{VP} = cos\ A$ more specifically with **VP**=1, is: **HP**=cos**A**
tangent of **A**	$\frac{HP}{VH} = tg\ A$

Furthermore, it is demonstrated the significant relationship, as follows: (sin**A**)^2 + (cos**A**)^2 = 1.0.

With reference to the figure below, the correspondences are as follows:



A°	sinA	cosA	tgA=sinA/cosA
0	0.0	1.0	0.0
0÷90	0.0÷1.0	1.0÷0.0	0.0÷ +(infinite)
90	1	0.0	+(infinite)
90÷180	1.0÷0.0	0.0÷(-1.0)	-(infinite) ÷ 0.0
180	0.0	-1.0	0.0
180÷270	0.0÷(-1.0)	(-1.0)÷0.0	0.0 ÷ +(infinite)
270	-1.0	0.0	-(infinite)
270÷360	(-1.0)÷0.0	0.0÷1.0	-(infinite) ÷ 0.0
360	0.0	1.0	0.0

Functions

sin	Computes the sine of the argument (in °). The value of the function is included in the range (-1.0 ÷ 1.0). The synthetic form is allowed: s. Usable: always. Error conditions: none. Example sin[90]= 1 sin[-90]= -1
cos	Computes the cosine of the argument (in °). The value of the function is included in the range (-1.0 ÷ 1.0).

	The synthetic form is allowed: c. Usable: always. Error conditions: none. Example cos[90]= 0 cos[gr[pi]]= -1
tan	Computes the tangent of the argument (in °). The synthetic form is allowed: t. Usable: always. Error conditions: • 132: invalid angle for tangent calculation. Example tan[45]= 1 tan[90]= causes error 132 tan[-90]= causes error 132
asin,as	Computes the arc-sine of the argument. The value returned by the function is in ° (degrees), included between 0 and 180°. The value of the argument must be included between - 1 and 1. The synthetic form is allowed: d. Usable: always. Error conditions: • 126: argument outside the range of values (- 1; 1). Example asin1= 90 asin[-1]= -90
acos,ac	Computes the arc-cosine of the argument. The value returned by the function is in ° (degrees), included between 0 and 180°. The value of the argument must be included between: - 1 and 1. The synthetic form is allowed: e. Usable: always. Error conditions: • 126: argument outside the range of values (- 1; 1). Example acos0= 90 acos[-1] = 180
atan,at	Computes the arc-tangent of the argument. The value returned by the function is in ° (degrees), included between -90° and 90°. The synthetic form is allowed: f. Usable: always. Error conditions: none. Example atan1= 45 atan[-1] = -45
gr	Converts the argument from radians into degrees (°): 1 radian=(180/Greekpi) °. The synthetic form is allowed: g. Usable: always. Error conditions: none. Example gr[pi] = 180
atan2[y,x]	It calculates the arc-tangent of (y/x). The value returned by the function is in ° (degrees), included between -180° and 180°. If both arguments are null, it returns value 0. Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands #2; Example atan2[1;0]=90 atan2[0;0]=0 atan2[1;1]=45

Functions which operate on strings

They should be considered as forms of advanced programming.

strlen[nn]	Returns the number of characters set for the r variable addressed by the argument. The examined string is: • setup string: if numeric variable • value string: if string variable In case of unassigned variable, the function returns 0. Remarkable format If the function argument is a parameter and has "rn" or "r\name" remarkable format (in this case the parametric form is compulsory: use of parentheses [.]): the function works directly on rn variable. It cannot be used: • in assignment of 'o', 'v' variables; • in assignment of custom functions Error conditions: • 112: use of r variable in invalid context; • 117: invalid index to "r" variable. Example 1 Let us assign the variable of type string r5="submio*r4", con r4="pippo" Solve the parametrization of r5: "submio\pippo" strlen5 returns the value 12 that is the number of character of "submio\pippo"; strlen[r5] returns the value 12 that is the number of character in r5="submio\pippo"; Example 2 Let us assign the numeric variable r5="r4/12" strlen5 returns the value 5 that is the number of characters in "r4/12".
getat[nn;np]	Returns the decimal value corresponding to a character string extracted from: • setup string: if numeric variable • value string: if string variable Returns a value of 0, in cases of: • unassigned variable • character position invalid (lower than 1 or greater than the length of the string). Arguments: nn = r variable index. The value of nn must be included between 0 and 299; np = character position in the string value (significant from 1). Remarkable format If the first function argument is a parameter and has "rn" or "r\name" remarkable format: the function operates directly on the rn variable. Remarkable format The first argument can assign directly a string between double quotation marks (ex: "foo"). In this case, the function can always be used. It cannot be used: • in assignment of 'o', 'v' variables • in assignment of custom functions Error conditions: • 112: use of r variable in invalid context; • 123: number of operands =0; • 124: number of operands #2; • 117: invalid index to "r" variable. Example 1 the selected variable (of string type) r5 is assigned as: "s2*r4", (where r4="pippo") if the parametric expression should be resolved r5: "s2\pippo" getat[5;2] returns 50 which corresponds to the decimal value of the '2' character getat[5;6] returns 112 which corresponds to the decimal value of the 'p' character Example 2 the numeric variable is assigned r5="r4/12" strlen[5,2] returns 52 which corresponds to the decimal value of the '4' character.
strcmp[n1;n2]	Returns the value of comparison between two strings: • 0 if the two strings are the same, • <0 if the first string is lower than the second, • >0 if the first string is greater than the second. The comparison does not take into account the differences between upper-case and lower-case characters. The comparison terminates when an inequality is discovered or when both strings have been compared. If two strings are equal to the end of one of the two, but still remain in the other characters, is considered greater. The return value is the result of the last comparison performed. Arguments: n1= index to first r variable n2= index to second r variable.

	Arguments may have parametric setting. The function operates on r variable of any type: • if of string type: the function applies to the resolved $ variable, • if of numeric type: the function applies to the data-entry $ variable, • in case of unassigned variable: the function applies to an empty string. One of the two arguments (or both) can assign directly a string, enclosed in quotation marks. Remarkable Format If a function argument is a parameter and has "rn" or "r\name" remarkable format: the function operates directly on the rn variable. It cannot be used: • in assignment of 'o', 'v' variables • in assignment of custom functions. Error conditions: • 112: use of r variable in invalid context; • 123: number of operands =0; • 124: number of operands #2; • 117: invalid index to "r" variable. Examples: strcmp[5; "pippo"] evaluates r5 and compares with "pippo" string strcmp[r5; "pippo"] evaluates r5 and compares with "pippo" string strcmp["pippo";r6] evaluates r6 and compares with "pippo" string strcmp[r5;r6] evaluates and compares r5 and r6.
strfind[n1;n2(;np)]	Returns the occurrence position of a string into another. The arguments (n1, n2) have a meaning and a syntax analog to the function strcmp. Argument np = start position for the search (significant from 1). It cannot be used: • in the assignment of a 'o', 'v' • in the assignment of custom functions. Error situations: • 112: use of r variable in a non valid context; • 123: number of operands =0; • 124: number of operands #2, 3; • 117: index of r variable non valid. Examples: (with: r5="pippo", r6="i") strfind[5;"p"] calculates r5 and searches the first occurrence of "p" (return: 1) strfind[r5;"p";4] calculates r5 and searches the first occurrence of "p" starting from the 4° character (return: 4) strfind["pippo";r6] searches r6 in the string "pippo" (return: 2) strfind[r5;r6] searches r6 in r5 (return: 2) strfind[r5;r6;3] searches r6 in r5 starting from the 3° character (return: 0) strfind["pippo";"i"] searches "i" in the string "pippo" (return: 2)
toolex[nn;nfield] tooltip[nn;nfield]	Apply to r variable of string type and interpret the value string. Arguments: nn = index to r variable nfield = field index (see later). Arguments may have parametric setting. The two functions return: toolex: the nfield field value resulting in the string value tooltip: 1 if the nfield field is numeric, 0 in case the variable is unassigned or of numeric type or nfield is invalid. A field is recognized: numeric (unsigned and integer numbers) or non-numeric field. For both functions the particular (nfield = 0) case is handled: they return the number of fields recognized in value string. Remarkable format If the first function argument is a parameter and has "rn" or "r\name" remarkable format: the function operates directly on the rn variable. Remarkable format The first argument can assign directly a string between double quotation marks (ex: "pippo"). In this case, this function can always be used. It cannot be used: • in assignment of 'o', 'v' variables • in assignment of custom functions. Error conditions:

	• 112: use of r variable in invalid context; • 123: number of operands =0; • 124: number of operands #2; • 117: invalid index to "r" variable. Example: r5="12|25;64" **toolex[5]=5** Number of recognized fields = 5 **tooltip[5;1]=1** **toolex[5;1]=12** 1st field value = 12 numeric field **tooltip[5;2]=0** **toolex[5;2]=124** 2nd field value = decimal of '|' = 124 non-numeric field ... 3rd field value = 25 numeric field 4th field value = decimal of ';' = 59 non-numeric field 5th field value = 64 numeric field nth field (n > 5) value = 0 non-numeric field. The example above demonstrates how the function can be used to interpret a programming corresponding to a tool mask, in this case, the example may require the selection of the tools from position 12 to 25, plus 64.

Logical Functions

ifelse[nc;n1;n2;(eps)]	Minimum ternary operator: • it returns n1 if nc is different from zero, • otherwise it returns n2. The arguments of this function must be 3 or 4. The equality comparison between nc and the zero value (0) is assessed on the epsilon value: • in the 3 argument version, the applied epsilon is a value between 0.0 and 0.001, as assigned in the TpaCAD configuration (Epsilon used in the logical comparisons). • in the 4 argument version, epsilon is assigned on the last argument (it is significant in absolute value). Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands #3; The function always evaluates both n1 and n2, although it returns only one of the two values. Nevertheless, a few particular situations are filtered corresponding to mathematical errors which may occur in the evaluation of the not returned term. Specifically, the following errors are not reported: • 125: null denominator, executing divisions or inv function; • 127: negative argument, in sqrt function; • 126: argument outside the range of values (-1; 1), in asin, acos functions. • 128: ne argument <0 or >10, in pown function. Example: ifelse[50;100;l/2] = 100 ifelse[0;100;l/2] = l/2
ifcase[nc1;nesp;nc2;n1;n2;(eps)]	Full ternary operator: it evaluates the (nc1 ? nc2) condition, with '?' assigned by nesp: • if the condition results true, it returns n1, • otherwise n2. The function returns n2 even if *nesp* is assigned an invalid value. The arguments of this function must be 5 or 6. The *nesp* argument is interpreted to assign the condition between nc1 and nc2: • Value 0 corresponds to < (less than) • Value 1 corresponds to <= (less than or equal to) • Value 2 corresponds to > (greater than) • Value 3 corresponds to >= (greater than or equal to) • Value 4 corresponds to = (equal to) • Value 5 corresponds to <> (different from). It is also possible to assign the nesp argument the corresponding symbolic forms, instead of the numerical value. Example: the "greater than or equal to" condition can be set as value 3 or ">=".

	The inequality relation can be expressed as "<>" or as "#". The comparison condition between nc1 and nc2 is assessed on epsilon value: • in the 5 argument version, the applied epsilon is a value between 0.0 and 0.001, as assigned in the TpaCAD configuration (Epsilon used in the logical comparisons) • in the 6 argument version, epsilon is assigned on the last argument (it is significant in absolute value). Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands #5; The function always evaluates both n1 and n2, although it returns only one of the two values. Nevertheless, the same particular conditions listed above for the ifcase function are filtered, corresponding to mathematical errors which may occur in the evaluation of the not returned term. Examples ifcase[5; >=;12;3;25] = 25 ifcase[5;<;12;3;25] = 3 ifcase[5;<>;12;3;25] = 3
case[nc;nc1:nv1;nc2:nv2;..;nvdef]	Condition test operator: it evaluates the (nc = nc1), (nc = nc2) conditions, returning the assigned "nv" value of the first verified condition. It tests if a condition, among those assigned, is verified, returning the value assigned to the condition verified as true. The arguments are: • nc: value to evaluate • nc1: first value of comparison with nc • nv: value returned by the function if nc = nc1 is true • nvdef: default value returned if no match has been found. The argument *nvdef* is not compulsory (if it is not assigned: it is 0); if assigned, it must be entered as last argument. The separator character between nc and nc1 is ";" (semicolon), while that between nc1 and nv1 is compulsorily ":" (colon). The maximum number of managed cases is 10, nvdef included. All arguments can be numeric or parametric. The comparison condition between nc and the assigned nc* values is assessed on epsilon value between 0.0 and 0.001, as assigned in TpaCAD configuration (Epsilon used in the logical comparisons). Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands <2 or >11. The function always evaluates nv1, nv2… nvdef, even though it returns only one of the two values. Nevertheless, the same particular conditions listed above for the ifcase function are filtered, corresponding to mathematical errors which may occur in the evaluation of the not returned terms. Example case[h;100:r0;200:h-100;l:l/2;h] • if h=100 the value of (r0) variable is returned • if h=200 the value resulting from (h-100) is returned • if h=l the value resulting from (l/2) is returned • if no equality has been detected, the (h) value is returned.
not[nc]	Argument negation operator: • if nc=0 returns 1; • if nc#0 (different from 0), returns 0. The equality comparison between nc and the zero (0) value is assessed on epsilon value between 0.0 and 0.001, as assigned in TpaCAD configuration (Epsilon used in the logical comparisons). Examples not[5] = 0 not[0] = 1

Technological Functions

Generally, you can access all the information (parameter) for the plant technology. As already seen, the actual allocation of the implant technology depends on the individual application: it is, however, provided a group of general functions, which allow access to this information.

Each parameter is:
- generally accessible with a numeric identifier (type) or, optionally, of type string. See for example the function prtool, in which nkind specifies the type of the parameter in numerical form or as a name. For parameters that are judged to be of considerable interest, nkind may indicate a symbolic name, with the following formalism:
  - fix part "p\";
  - variable part, however, non-customizable (see table below), the assigned symbolic name for the parameter. The combination of parameter's symbolic name and numerical reference (typology) occurs automatically.
- can be in some cases entered through a features indication. See function prfi that read directly the parameter assigning the diameter of a tool.
- can be addressable in an absolute way as a matrix cell, with indication of (row, column). This addressing mode requires that the parametric plant may be interpreted in a matrix organization: each has its own identification information (numerical and possibly literal) and also a position, in a matrix, in fact. The functions (prmxmac, prmxgru, …) access to each individual parameter in this way. The use of these functions requires a good knowledge of how the parametric drive is structured so it is reserved for developers.

Technological parameters assigned with symbolic formalism

p\gron	Type of parameter enabling a head group (numeric: 6)
p\face	Type of parameter assigning working face/faces of a spindle (numeric: 6)
p\ofx p\ofy p\ofz	Parameter types: • offset (x/y/z) of a group • offset (x/y/z) of a spindle (numeric: 100, 101, 102)
p\xmax p\xmin	Types of parameters of the min. and maximum positioning of a head group on the X axis. (numeric: 150, 151)
p\ymax p\ymin	Types of parameters of the min. and maximum positioning of a head group on the Y axis. (numeric: 152, 153)
p\zmax p\zmin	Types of parameters of the min. and maximum positioning of a head group on the Z axis. (numeric: 154, 155)
p\cmax p\cmin	Types of parameters of the min. and maximum positioning of a head group on the C axis. (numeric: 158, 157)
p\betamax p\betamin	Types of parameters of the min. and maximum positioning of a head group on the B axis. (numeric: 160, 159)
p\attr	Type of outfit parameter of a spindle or of a toolholder position (numeric: 220)
p\fitool	Type of parameter of a tool diameter (numeric: 1002)
p\tiptool	Type of parameter of a tool (numeric: 1001)
p\lltool	Type of working length of a tool (numeric: 109)
p\ltottool	Type of total length of a tool (numeric: 111)
p\lauxtool	Type of auxiliary length of a tool (numeric: 112)
p\ariatool	Type of parameter of tool clearing position (numerical: 121)
p\feedmin	Type of parameter of min. working speed of a tool (numeric: 2004)
p\feedmax	Type of parameter of maximum working speed of a tool (numeric: 2006)
p\feed	Type of parameter of default working speed of a tool (numeric: 2005)
p\rpmmin	Type of parameter of min. rotation speed of a tool (numeric: 2001)
p\rpmmax	Type of parameter of maximum rotation speed of a tool (numeric: 2002)
p\rpm	Type of parameter of default rotation speed of a tool (numeric: 2003)
p\invtool	Type of mirror tool parameter (numeric: 124)

Access functions to a generic plant group

They should be considered as functions of advanced programming.

primp[nkind; (vdef)]	Returns a generic Plant group parameter: • nkind = parameter type (this item is compulsory) • vdef = default value (in case of parameter not found). If unset or empty, the default value is 0. If nkind is =0 (null), the function returns the machine number set up in the plant. It also recognized the form primp["nameKind";(vdef)], where the parameter is indicated by name. The meaning of "nameKind" must be agreed in phase specific technologies plant. Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands >2.; • 130: nkind argument omitted (empty assignment = . Example: primp[1100]: returns the parameter value 1100 (0 if parameter not found) Example: primp [1100]: returns the value of parameter 1100 (0 if parameter not found)

Access functions to a machine level for the Configuration Of Head Groups

They should be considered as functions of advanced programming.

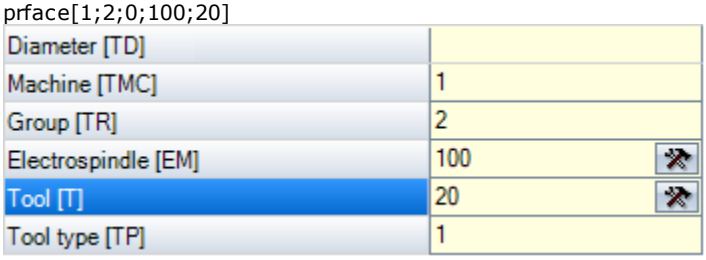
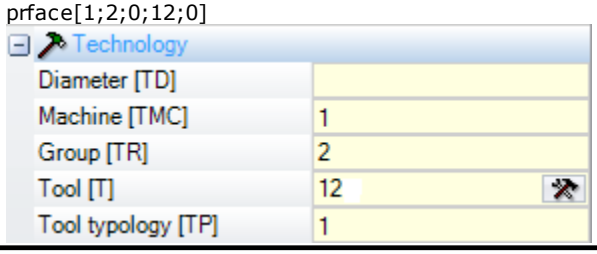
prmac[(nm);nkind;(vdef)]	Returns a generic Machine parameter: • nm = machine (this item is compulsory) (in case of empty assignment: the default value is 1) • nkind = parameter type (this item is compulsory) • vdef = default value (in case of not found parameter). The maximum value that can be set for nm is given by the configuration of the plant technology. If it is nkind=0 (null), the function returns 1 (different from 0), if the machine is configured and present in the plant. It also recognized the form *prmac[(nm);"nameKind";(vdef)]*, where the parameter is indicated by name. The meaning of nameKind must be agreed during the specification of the technologies of the plant. Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands <2 or >3; • 130: nkind argument omitted (empty assignment). Examples • prmac[2;1100]: this function returns the parameter value (1100) of machine 2 {0 in case of not found parameter} • prmac[;1100;100]: this function returns the parameter value (1100) of machine 1 (default machine) {100 in case of not found parameter) • prmac[2;0]: returns the value of the parameter called "areas" of machine 2 (0 if the parameter is not assigned)
prgr[(nm);(ng);nkind;(vdef)]	Returns a generic head group parameter: • nm = machine (this item is compulsory) (in case of empty assignment: the default value is 1) • ng = head group (this item is compulsory) (in case of empty assignment: the default value is 1) • nkind = parameter type (this item is compulsory) • vdef = default value (in case of not found parameter). If unset or empty, the default value is 0. The maximum usable values for nm and ng are given by the configuration of the plant technologies and machine. If it is nkind =0 (null), the function is brought back to 1 (different from 0) if the group is configured and present. It also recognized the form *prgr[(nm);(ng);"nameKind";(vdef)]*, where the parameter is indicated by name. The meaning of "nameKind" must be agreed on during the specification of the technologies of the plant.

	Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands <3 or >4; • 130: nkind argument omitted (empty assignment). Examples • prgr[2;3;1100]: this function returns the parameter value (1100) of group 3 of machine 2 {0 if parameter not found} • prgr[2;;1100;100]: this function returns the parameter value (1100) of group 1 (default group) of machine 2 {100 if parameter not found) • prgr[2;3;p\ofx]: returns the value (x offset of the group) of the parameter (p\ofx) of group 3 of machine 2.

Tool access functions

If the universal tool management is enabled, it is possible to assign value 0 to both machine, group and spindle. However, the complete prototype of the function must always be used.

prface[(nm);(ng);(np);(ns);nt;(side)] prface[(nm);(ng);(nt);side] prface[ng;nt;nside] prface[nt;nside]	Tests if the tool can work on the inside face: • nm = machine (this item is compulsory) (in case of empty assignment: the default value is 1) • ng = head group (this item is compulsory) (in case of empty assignment: the default value is 1) • np = if the outfit found by (ns;nt) identifies a toolholder, the parameter np shows the position of the tool fitted out in the toolholder • ns = spindle (compulsory) • nt = tool/toolholder to be used on the ns spindle or spindle. (if nt is not assigned, value 0 is imposed) • nside = face (if left out or assigned as empty sets off nside = current face; if assigned, it interprets the face custom number) The function returns value 1 if the test is verified, otherwise it returns 0. Some particular cases can be examined, such as: (ns<=0; nt=0) In this case any valid technology is not shown. The function returns value 0; (ns>0; nt=0) The ns spindle is fitted out as configured in the head group. In particular: • if ns is not fitted out (this is a position of electrospindle with tool change): verification is made according with the information on the working face assigned to the electrospindle • if ns is fitted out with a toolholder, np displays the tool position on the toolholder (by default if no=0: first point). Verification is only made on ns configuration. (ns<=0; nt#0) The spindle is now displayed in nt and it is fitted out as shown in the head group configuration (see: (ns>0; nt=0), with spindle now in nt). (ns>0; nt#0) If nt has a significant value (in a valid range tool or toolholder range), the spindle ns is considered fitted out with nt. In particular, if nt displays a toolholder, np displays the tool position on the toolholder. In case of outfit on a toolholder: • if np is not assigned: it is verified on the first fitted out tool. • if np is assigned as not valid (<=0 or as well as the maximum value allowed), the function is brought back to the value 0.
	The maximum usable values for (nm, ng, np,ns, nt) are given by the technologies configurations of plant, machine, group, tools and toolholders catalogue. Manner of fitting out a tool rather than a toolholder also depends on technologies configuration. The nside face number assigns the z-axis plane and orientation. The equivalent forms of the reduced formats are as follows: prface[nm;ng;nt;side] is equal to -> prface[nm;ng;0;-1;nt;side] prface[nm;ng;nt;side] is equal to -> prface[1;ng;0;-1;nt;side] prface[nt; nside] is equal to -> prface[1;1;0;-1;nt;side].

	Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands <3 or >6 • 130: arguments ns and nt both left out (empty assignation) Examples prface[;;;;90]: head if the spindle identified as (nm=1; ng=1; np=0; ns=1; nt=90) can work on the active face. With reference to the technological program of a working, following correspondences to the parameters of the *prface* function are shown in order to check the work face in case of: • programming of technology for spindle and tool: prface[1;2;0;100;20]  • programming technology for spindle and tool: prface[1;2;0;100;20] • programming technology for spindle where both the equivalent forms are available: prface[1;2;0;0;12] prface[1;2;0;12;0] 
prfi[(nm);(ng);(np);(ns);nt] prfi[(nm);(ng);(ns);nt] prfi[(nm);(ng);nt] prfi[ng;nt] prfi[nt]	Returns the diameter of the tool: • nm = machine (this item is compulsory) (in case of empty assignment: the default value is 1) • ng = head group (this item is compulsory) (in case of empty assignment: the default value is 1) • np= if the tooling found by (ns;nt) specifies a toolholder, the parameter np displays the position of the tool as fitted out on the toolholder • ns = spindle (compulsory) • nt = tool/toolholder to fit out on ns spindle or spindle. (If nt is not assigned a value 0 is assigned) We can consider the same cases seen for the function: *prface*. The equivalent forms of the reduced formats are as follows: prfi[nm;ng;nt] is equal to -> prfi[nm;ng;0;-1;nt] prfi [ng; nt] is equal to -> prfi[1;ng;0;-1;nt] prfi [nt] is equal to -> prfi[1;1;0;-1;nt] Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands <3 or >5; • 130: arguments nt and ns both left out (empty assigment). Examples prfi[1;1;90]: returns the diameter of the spindle identified as (nm=1; ng=1; np=0; ns=1; nt=90) prfi[0;0;;0;1015]: returns the diameter of the universal tool identified as (nm=0; ng=0; np=0; ns=0; nt=1015)
prrot[(nm);(ng);(np);(ns);nt] prrot[(nm);(ng);(ns);nt]	Returns the rotation direction of the tool:

prrot[(nm);(ng);nt] prrot[ng;nt] prrot[nt]	• nm = machine (this item is compulsory) (if assignment empty: uses = 1) • ng = head group (this item is compulsory) (if assignment empty: uses = 1) • np = if the tooling identified by (ns;nt) specifies a toolholder, the parameter np indicates the position of the tool equipped in the toolholder • ns = spindle (this item is compulsory) • nt = tool/toolholder to be equipped on the spindle ns or spindle. (If nt is not assigned, 0 value is set) The function returns 1 if the direction is counterclockwise, otherwise it returns 0. We can consider the same particular cases as for the function: *prface*. Below are the equivalent forms of the reduced formats: prrot[nm;ng;nt] is equivalent to -> prrot[nm;ng;0;-1;nt] prrot[ng;nt] is equivalent to -> prrot[1;ng;0;-1;nt] prrot[nt] is equivalent to -> prrot[1;1;0;-1;nt] Usable: always. Error situations: • 123: number of operands =0; • 124: number of operands <3 or >5; • 130: arguments nt and ns both omitted (empty assignment).
prtool[(nm);(ng);(np);ns;(nt);nkind;(vdef)] prtool[(nm);(ng);nt;nkind;(vdef)] prtool[nm;ng;nt;nkind] prtool[ng;nt;nkind] prtool[nt;nkind]	Returns a generic tool parameter: • nm = machine (this item is compulsory) (in case of empty assignment: the default value is 1) • ng = head group (this item is compulsory) (in case of empty assignment: the default value is 1) • if the tooling found by (ns;nt) specifies a toolholder, the np parameter displays the position of the tool as fitted out on the toolholder. • ns = spindle (compulsory) • nt = tool/toolholder to fit out on the ns spindle or spindle. (If nt is not assigned, the value 0 is assigned). • nkind = parameter type (this item is compulsory) • vdef = default value (returned if parameter not found). (If vdef is not assigned, the value 0 is assigned). We can consider the same cases seen for the function: *prface*. The maximum usable values for (nm, ng, np, ns, nt) are given by the technologies configurations of plant, machine, group, tools and toolholders catalogue. Manner of fitting out a tool rather than a toolholder also depends on technologies configuration. Forms are also recognized with *nKind* replaced by "nameKind", where the parameter is indicated by name. The meaning of "nameKind" must be agreed on in phase of specific technologies plant.
	The equivalent forms of the reduced formats are as follows: prtool[nm;ng;nt;nkind;vdef] is equal to -> prtool[nm;ng;0;-1;nt;nkind;vdef] prtool[nm;ng;nt;nkind] is equal to -> prtool[nm;ng;0;-1;nt;nkind;0.0] prtool[ng;nt; nkind] is equal to -> prtool[1;ng;0;-1;nt;nkind;0.0] prtool[nt;nkind] is equal to -> prtool[1;1;0;-1;nt;nkind;0.0]. Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands <2 or >7; • 130: nt and ns arguments both left out or nkind left out (empty assignment). Examples: prtool[1;2;;100;3;100]: returns the value of the tool parameter (100) identified as (nm=1; ng=2; np=0;ns=100, nr=3). It returns 0 if the parameter required is not found.

	prtool[1;1;90;p\fitool]: reads the spindle diameter in position (nm=1; ng=1; nt=90). The p\fitool parameter can be replaced by the value 1002. prtool[1;1;90;p\tiptool]: reads the value of spindle type in position (nm=1; ng=1; nt=90).
prtip[(nm);(ng);(np);ns;(nt)] prtip[(nm);(ng);(ns);nt] prtip[nm;ng;nt] prtip[ng;nt] prtip[nt]	Returns the type of a tool: • nm = machine (compulsory) (in case of empty assignment: it uses =1) • ng = head group (compulsory) (in case of empty assignment: it uses =1) • np = if the outfit resulting from (ns;nt) defines a toolholder, it shows the tool fitted out on the toolholder. • ns = spindle (compulsory) • nt = tool/toolholder to fit out on the ns spindle or spindle (in case of empty assignment: it uses =0) We can consider the same cases seen for the function: *prface.* The equivalent forms of the reduced formats are as follows: prtip[nm;ng;nt] is equal to -> prtip[nm;ng;0;-1;nt] prtip[ng;nt] is equal to -> prtip[1;ng;0;-1;nt] prtip[nt] is equal to -> prtip[1;1;0;-1;nt]. Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands <1 or >5; • 130: nt and ns arguments both left out.
prfulcrox[(nm);(ng);(np);ns;(nt)] prfulcroy[(nm);(ng);(np);ns;(nt)] prfulcroz[(nm);(ng);(np);ns;(nt)]	Return the (x, y, z) position of the selected tool pivot point, in condition of machine set to 0: • nm = machine (this item is compulsory) (in case of empty assignment: the default value is 1) • ng = head group (this item is compulsory) (in case of empty assignment: the default value is 1) • np = if the outfit found by (ns;nt) specifies a toolholder, the np parameter shows the position of the tool fitted out on the toolholder • ns = spindle (compulsory) • nt = tool/toolholder to fit out on the ns spindle or spindle (in case of empty assignment: is uses =0). We can consider the same cases seen for the function: *prface*. The maximum usable values for (nm, ng, np, ns, nt) are given by the technologies configurations of plant, machine, group, tools and toolholders catalogue. Manner of fitting out a tool rather than a toolholder also depends on technologies configuration. Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands #5; • 130: nt and ns arguments both left out (empty assignment).
pngru[(nm);(ng);(ns);nt;(deltafi)] pnstool[(nm);(ng);(ns);nt;(deltafi)] pntool[(nm);(ng);(ns);nt;(deltafi)]	Return information related to the technology replacement: • nm = machine (compulsory) (if empty assignment: use = 1) • ng = head group (compulsory) (if empty assignment: use = 1) • ns = spindle (compulsory) • nt = tool/tool holder to be equipped on the ns spindle or on the spindle • deltafi = permitted deviation on the diameter ▪ 0.0 value (or <epsilon) shows the replacement is only possible with the same diameter ▪ positive value (ex: 3.0) indicates that the replacement admits a maximum difference between the diameters of +/- 3.0 (mm / inch) ▪ negative value (ex: -1.0) indicates that the substitution allows a maximum difference of 1.0 (mm / inch) but only when it decreases compared to the original diameter

	The functions return: • pngru: the replacing Group number • pnstool: the replacing Spindle number • pntool: the replacing Tool number The functions return a negative value (-1), if the technology requires a replacement that is not possible. Usable: always. Error situations: • 123: number of operands =0; • 124: number of operands <4 or>5; • 130: arguments nt and ns both omitted (empty assignment).

Function of direct access to matrix plants

As it has already been said, the addressing mode matrix states that the parametric plant can be interpreted with this organization: each has its own identification information (numeric and possibly literal) as well as a position in a matrix, in fact.
Functions must be considered advanced programming.

prmxmac[(nm);nrow;ncol,(vdef)	Returns the value of a matrix generic cell of machine configuration: • nm = machine (compulsory) (in case of empty assignment: it uses =1) • nrow = matrix row (compulsory) (significant from value 1) • ncol = matrix column (compulsory) (compulsory) (significant from value 1) • vdef = default value (returned if the parameter is not found) If not set or set up as empty: it uses =0 Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands different from 3,4; • 130: nrow argument or left out ncol (empty assignment).
prmxgru[(nm);(ng);nrow;ncol,(vdef)]	Returns the value of a matrix generic cell of groups configuration: • nm = machine (compulsory) (in case of empty assignment: it uses =1) • ng = head group (compulsory) (significant from 1) • nrow = matrix row corresponding to ng group (compulsory) (significant from 1) • ncol = matrix column (compulsory) (significant from 1) • vdef = (returned if the parameter is not found). If not set or set as empty: it uses =0 Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands different from 4 and 5; • 130: nrow argument or left out ncol (empty assignment).
prmxtool[(nm);ntool;(nkind);(ncol);(vdef)]	Returns the value of a generic information from the matrix of the tool catalogue: • nm = machine (compulsory) (in case of empty assignment: it uses =1) • ntool = tool number (compulsory) • nkind = parameter type • ncol = matrix column (significant from 1) (returned if the parameter is not found). If not set or set as empty: it uses =0

	If ntool is null (=0): the function returns the maximum identification number of the configured tools in the tool catalogue. If ntool has assigned a valid worth, it selects a tool (the absolute value is taken). The following cases can be distinguished: • nkind =0, ncol =0 (null): the function returns 1 (different from 0) if the ntool tool is configured; • nkind different from 0: it reads the parameter with nkind typology (nkind: considered in absolute value), assigned for the ntool tool; • nkind =0, ncol different from 0: it reads the parameter in the column ncol (ncol: considered in absolute value), assigned for the ntool tool. Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands < 2 or >5; • 130: nrow argument or left out ncol (empty assignment). Examples: prmxtool[1;0;0;0]: returns the maximum identification number of the configured tools in the tool catalogue of the machine 1 prmxtool[1;3;p\fitool]: Returns the diameter of the tool 3 of the tool catalogue of the machine 1 prmxtool[1;3;0;6]: returns the parameter of column 6 of the tool 3 of the tool catalogue of the machine 1
prmxhtool[(nm);ntool;(nrow);(nkind);(ncol);(vdef)]	Returns the value of a generic information from toolholder matrix (catalog): • nm = machine (compulsory) (in case of empty assignment: it uses =1) • ntool = toolholder number (compulsory) (significant from 1) • nrow = toolholder row (compulsory) (significant from 0) • nkind = parameter type • ncol = matrix column (significant from 1) • vdef = value by default (in case of not accessible cell) – (if not set or set as empty: it uses =0) If ntool is null, the function returns the toolholders number, that are configured in the toolholder catalogue. If ntool is not null: it selects a toolholder. The following cases can be distinguished: • nkind =0, ncol =0 (null): the function returns 1 (different from 0) if the toolholder ntool is configured; • nkind different from 0: it reads the parameter of type nkind (nkind: considered in absolute value), assigned or the ntool tool to the configuration row nrow; • nkind =0, ncol different from 0: it reads the parameter in the ncol column (ncol: considered in absolute value), assigned for the ntool tool in the row configuration nrow. If it is nrow=0: shows the configuration row of the toolholder. If it is nrow #0 (the absolute value is taken): shows the configuration row of the nrow-th fitted out tool (significant for the maximum number of the tools, that can be fitted out). Usable: always. Error conditions:

	• 123: number of operands =0; • 124: number of operands < 2 o > 6; • 130: ntool, nrow argument or left out ncol (empty assignment). Examples: prmxhtool[1;0]: returns the number of the tool-holders configured for the machine 1 prmxhtool[1;3]: returns 1 if the tool-holder 3 of machine 1 is configured prmxhtool[1;3;0;6]: returns the parameter defined in the column 6 of the configuration row of the tool-holder 3 of the machine 1 prmxhtool[1;3;4;p\ofx]: returns the parameter correction x of the point 4 of the tool-holder 3 of machine 1 prmxhtool[1;3;4;;37]: returns the column parameter 37 of the point 4 of the tool-holder 3 of machine 1
prmxstore[(nm);nstore;(nrow);(nkind);(ncol),(vdef)]	Returns the value contained in a generic matrix configuration cell of a carousel: • nm = machine (compulsory) (in case of empty assignment: it uses =1) • nstore = carousel number (significant from 1) • nrow = matrix row of the carousel defined by nstore (compulsory) (significant from 0) • nkind = parameter type • ncol = matrix column (significant from 1) • vdef = value by default (in case of not accessible cell) - (if not set or set as empty: it uses =0) If the *nstore* parameter is null, the function returns the number of the configured magazines. If the *nstore* parameter is different from 0: select a tool magazine (the absolute value is taken). The following cases can be distinguished: • nkind =0, ncol =0: the function returns 1 (different from 0) if the position defined by nrow is fitted out. • nkind different from 0: returns the parameter with nkind typology (the value of nkind is considered in absolute value) assigned for the nrow position. • nkind=0, ncol different from 0: it reads the parameter in the ncol column (ncol: considered in absolute value), and nrow row Usable: always. Error conditions: • 123: number of operands =0; • 124: number of operands < 2 o > 6; • 130: nstore argument left out (empty assignment) Examples: prmxstore[1;0]: returns the number of the tool collectors configured for the machine 1 prmxstore [1;3]: returns the maximum position equipped for the tool collector 3 of machine 1 prmxstore [1;3;4]: returns 1 if the position 4 of the tool collector 3 of machine 1 is equipped prmxstore [1;3;4;1200]: returns the parameter of typology 1200 from the position 4 of the tool-holder 3 of machine 1 prmxstore [1;3;4;0;12]: returns the parameter given in the column 12 from the position 4 of the tool-holder 3 of machine 1.

Multi-Purpose Geometry Library Functions

This is a function that implements a rich set of functionalities, mainly geometric: the first argument of the function indicates the name of remarkable selection of functionality. The following is indicated in bold and each case is documented separately. For all cases are subject to the following notes:

Usable:

- always, in the versions that do not use working names
- The versions using working names cannot be used, when 'o and 'v' variables, variable geometries (fictive faces edges), custom functions are assigned.

Error conditions:
- 123: number of operands =0;
- 124: wrong number of operands.
- 116: invalid context in those versions that use working names.

No geometric error conditions have been detected; in any case a default condition is assumed.

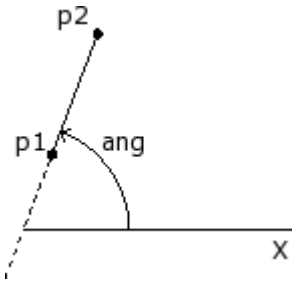
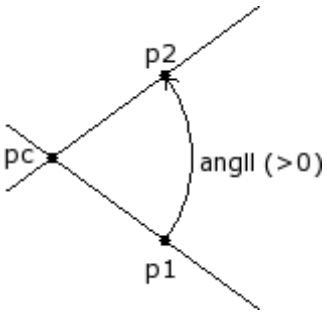
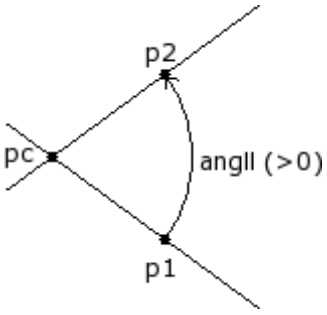
The geometric context used for the solution of geometric situations corresponds to a system of XYZ Cartesian triad pure. The correspondence of the topics specified in correspondence with an axis (X, rather than Y or Z) with real axes of face or piece is in general completely abstract. A topic indicated for the X axis may actually correspond to a Z axis of the inclined face or of the piece: the geometry features have a valence of geometric library, to be used adapting arguments and results to specific items.

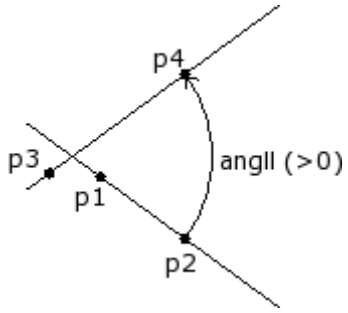
Function versions that use working names search the version indicated before the current working. Search is broken at the first correspondence. The working name is indicated with the formalism "wname" and must be placed between double quotation marks.
A term can be summed to the search. Following syntax types are recognized:
- "wname+2": shows that the searched working is located two lines after "wname";
- "wname-2": shows that the searched working is located two lines before "wname";
- "wname+";nn: where the displacement is assigned in an added argument (in parametric form, as well). "wname-" can also be used.

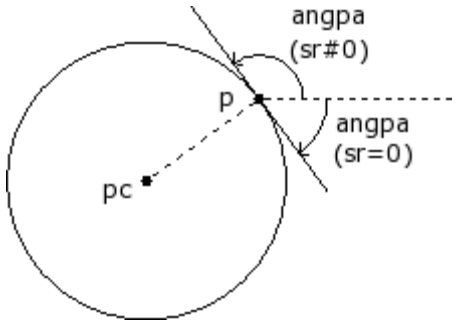
The functions should be partly considered as functions of advanced programming.

Functions for calculating angles

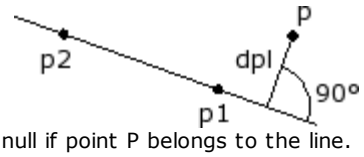
geo[**angc**;ang;(sgn)]	Returns the angle ang reduced to: 0° - 360°. ang = angle sgn = if value different from 0, preserves the sign of the original ang (if the field is empty or not given: use = 0). Examples: geo[angc;4500] = 180 geo[angc;-450] = 270 geo[angc;-450;1] = -90
geo[**angm**;ang;angr;(sgn)]	Evaluates if (ang) is multiple of (angr) and returns the number of multiples. ang = angle (used reduced to +/-(0° – 360°)) angr = multiple angle (used in absolute value) sgn = if the value is different from 0, it preserves the original sign of ang (if the field is empty or not assigned, it uses=0). Returns 0, if: • |ang| <= 0.001 • |angr| <= 0.001 • (ang) is not a multiple of (angr). Otherwise, it returns an integer value #0. The returned value is negative if (ang<0) and (sgn#0). Examples: geo[angm;180;90;0] = 2 geo[angm;-180;90;0] = 2 geo[angm;-180;90;1] = -2 geo[angm;181;90;0] = 0 geo[angm;540;90;0] = 2 <- (ang) is reduced to (0° - 360°)
geo[**ang**;x1;y1;x2;y2]	Returns the angle between the x axis and the line-oriented P1-P2: x1;y1 = abscissa and ordinate of point P1 x2;y2 = abscissa and ordinate of point P2

	 The angle value is included in the range [0 <= ang < 360], in units of degrees [°]. If the two points P1 and P2 coincide: the function returns 0. Example: geo[ang;100;100;400;400] = 45
geo[**ang**;"wname"] geo[**ang**;"wname+nn"] geo[**ang**;"wname",nn"]	The function is similar to the function geo[**ang**;x1;y1;x2;y2]. The oriented line P1-P2 is defined by the working name. Working "wname+nn" must correspond to a linear segment, that is assigned with one only line. If the working is not correctly defined, the function returns the value 0.0.
geo[**angll**;xc;yc;x1;y1;x2;y2]	Returns the angle between the oriented lines Pc-P1 and Pc-P2 (Pc=centre) xc;yc = abscissa and ordinate of point Pc x1;y1 = abscissa and ordinate of point P1 x2;y2 = abscissa and ordinate of point P2. The angle value is included in the range [−180 < angll <= +180], in units of degrees [°]: • zero (= 0): if the points Pc, P1 and P2 are not all distinct, • positive: if the line Pc-P1 closes on the line Pc-P2 in counterclockwise; • negative: otherwise.  Examples: geo[angll;100;100;200;0;400;100]= 45 geo[angll;100;100;400;100;200;0]= -45 geo[angll;100;100;400;100; 400;100]= 0
geo[**angll**;xc;yc;zc;x1;y1;z1;x2;y2;z2]	Returns the angle between the oriented lines Pc-P1 and Pc-P2 (Pc=centre) in the space: 

	xc;yc;zc = coordinates of point Pc x1;y1;z1 = coordinates of point P1 x2;y2;z2 = coordinates of point P2 The value of the angle is included in the range [−180 < angll <= +180], in units of degrees [°]. If the points P1 and P2 coincide, the value of the angle is 0. Examples: geo[angll;400;0;-100;400;0;0;450;-20;0] = 28.3 geo[angll;0;0;0;100;0;0;0;100;0] = 90 <- angle between two coordinate axes (X axis and Y axis) geo[angll;0;0;0;100;0;0;0;0;100] = 90 <- angle between two coordinate axes (X axis and Y axis)
geo[**angll**;x1;y1;x2;y2;x3;y3;x4;y4]	Returns the angle between the oriented lines P1-P2 and P3-P4 x1;y1 = abscissa and ordinate of point P1 x2;y2 = abscissa and ordinate of point P2 x3;y3 = abscissa and ordinate of point P3 x4;y4 = abscissa and ordinate of point P4. The angle value is included in the range [−180 < angll <= +180], in units of degrees [°]: • null (=0): if the two lines are parallel or coincident • positive: if the line P1-P2 closes on the line P3-P4 in counterclockwise direction; • negative: otherwise;  Example: geo[angll;100;100;200;0;0;100;400;100] = 45
geo[**angll**;"wname1+nn";x3;y3;x4;y4] geo[**angll**;"wname1+nn";"wname2+nn"]	This function is similar to the function geo[**angll**;x1;y1;x2;y2;x3;y3;x4;y4]. It returns the angle between the oriented lines, where the first or both the lines are defined by a working name. "wname1"= name of the working assigning the first line "wname2"= name of the working assigning the second line "wname1" and "wname2" workings must define the linear segments made of one only line. If the workings are not correctly defined, the function returns the value 0.0.
geo[**angpc**;x;y;xc;yc;(sr)]	Returns the tangent angle of a point P on circle: x;y = abscissa and ordinate of point P xc;yc = abscissa and ordinate of the circle centre Pc sr = direction of rotation on circle (#0 counterclockwise; 0=clockwise= default if unassigned). The angle value is included in the range [0 <= ang <= 360], in units of degrees [°]. In case the two points P and Pc coincide: returns the value 0.

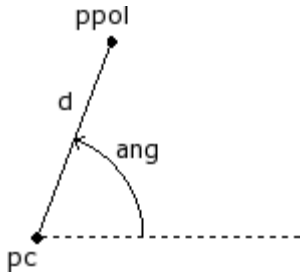
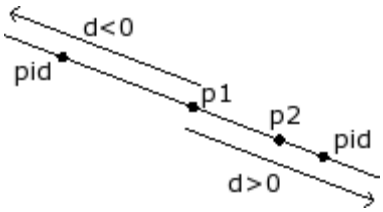
	 Examples: geo[angpc;0;100;0;0;1] = 180 geo[angpc;0;100;0;0] = 0
geo[**angpc**;"wname+nn"]	This function is similar to the function geo[**angpc**;x;y;xc;yc;(sr)]. It returns the tangent angle of a Point P on a circle arc, assigned with a working name: • working "wname" must define an arc on xy plane made of one only line; • the point P is the end point of the arc. If the working is not correctly defined, the function returns the value 0.0.

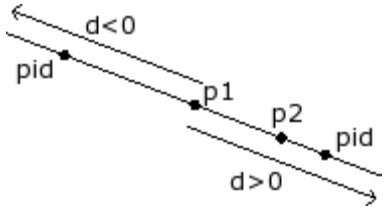
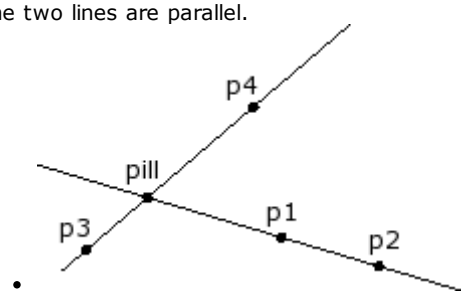
Functions for calculating distances

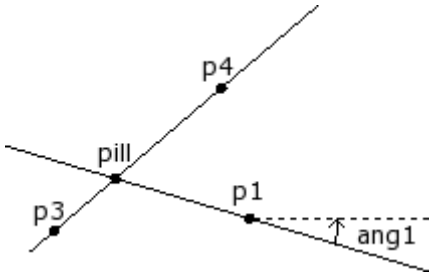
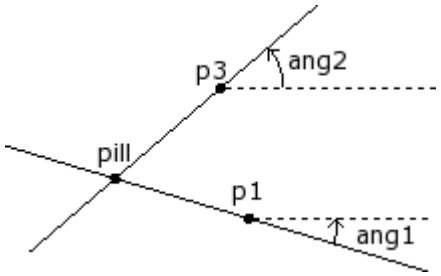
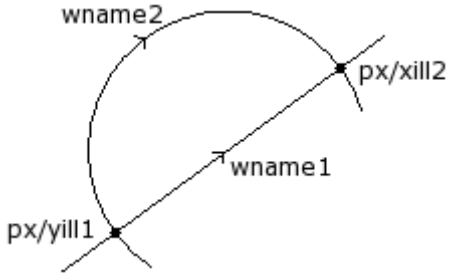
geo[**dist**;x1;y1;x2;y2]	Returns the distance between the two points P1 and P2 (in plane): x1;y1 = abscissa and ordinate of point P1 x2;y2 = abscissa and ordinate of point P2. Examples geo[dist;0;100;100;-200] = 316.2278
geo[**dist**;x1;y1;z1;x2;y2;z2]	Returns the distance between the two points P1 and P2 (in space): x1;y1;z1 = abscissa, ordinate and height of point P1 x2;y2;z2 = abscissa, ordinate and height of point P2. Examples geo[dist;0;100;10;100;-200;-10] = 316.8596
geo[**dpl**;x;y;x1;y1;x2;y2]	Returns the distance of point P from the line P1-P2: x;y = abscissa and ordinate of point P x1;y1 = abscissa and ordinate of point P1 x2;y2 = abscissa and ordinate of point P2.  The distance is null if point P belongs to the line. Examples geo[dpl;0;200;0.;0;100;100] = 141.4214 geo[dpl;50;50;0.;0;100;100] = 0 <- the point (50;50) is on the straight line
geo[**dpl**;"wname1+nn";x1;y1;x2;y2] geo[**dpl**;x;y;"wname2+nn"] geo[**dpl**;"wname1+nn";"wname2+nn"]"	This function is similar to the function geo[dpl;x;y;x1;y1;x2;y2]. It returns the distance of the point P from the line P1-P2. The point and/or the line can be determined by a working name. "wname1"= name of the working assigning the point P. It may correspond to: • point or to a setup, • In a profile line (line or arc) the point P is the end point of the line. "wname2"= name of the working assigning the linear segment. It must define a linear segment made of one only line. If the working is not correctly defined, the function returns the value 0.0.
geo[dim;"wname+nn";(nmodo)]	Returns the length of a profile or of a profile element:

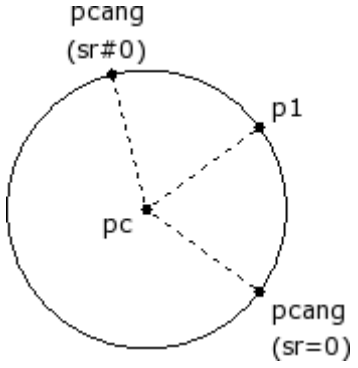
	• nModo: 0=(default) returns the length of the profile from the "wname" working to the end (or anyway before the current working) – including possible entry/exit segments programmed on the setup • nModo: 1 (#0) = returns the length of the element "wname" (if setup=0)

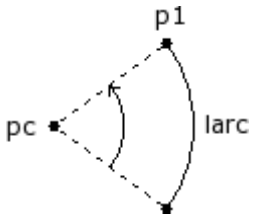
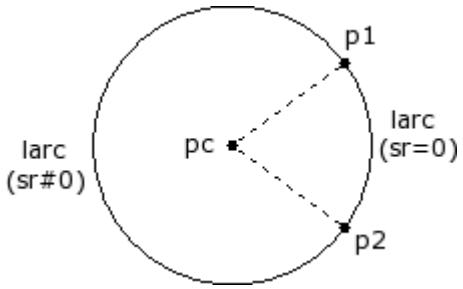
Function of detection points on geometric elements

geo[**pxpol**;xc;yc;ang;d] geo[**pypol**;xc;yc;ang;d]	Return the abscissa (x) or the ordinate (y) of the point assigned with polar modes: xc;yc = abscissa and ordinate of point Pc (origin of the polar system) ang = angle d = vector (it is applied in absolute value).  If the vector d is null: the function returns the initial coordinate of point Pc. Examples geo[pxpol;0;0;30;100] = 86.6025 geo[pypol;0;0;30;100] = 50
geo[**pxld**;x1;y1;x2;y2;d] geo[**pyld**;x1;y1;x2;y2;d]	Return the abscissa (x) or the ordinate (y) of the point on the line P1-P2 (assigned on the plane), at distance d from P1: x1;y1 = abscissa and ordinate of point P1 x2;y2 = abscissa and ordinate of point P2 d = distance. If d>0 (positive): the point is calculated from P1 to P2; If d<0 (negative): the point is calculated from P1 in direction opposite to P2; If d=0 (positive): the point coincides with P1.  Examples geo[pxld;0;0;100;0;200] = 200 geo[pyld;0;0;100;0;200] = 0
geo[**pxld**;x1;y1;z1;x2;y2;z2;d] geo[**pyld**;x1;y1;z1;x2;y2;z2;d] geo[**pzld**;x1;y1;z1;x2;y2;z2;d]	Return the abscissa (x), the ordinate (y) or the coordinate z of the point on the line P1-P2 (assigned in the space), at a distance d from P1.

	 x1;y1;z1 = abscissa and ordinate of the point P1 x2;y2;z2 = abscissa and ordinate of the point P2 d = distance. If d>0 (positive): the point is calculated from P1 to P2; If d<0 (negative): the point is calculated from P1 in direction opposite to P2; If d=0 (positive): the point coincides with P1. If the points P1 and P2 coincide, the coordinate of P1 is returned.
geo[**pxld**;"wname+nn";d] geo[**pyld**;"wname+nn";d] geo[**pzld**;"wname+nn";d]	The functions are similar to the functions: geo[**pxld**;x1;y1;z1;x2;y2;z2;d], geo[**pyld**;x1;y1;z1;x2;y2;z2;d] and geo[**pzld**;x1;y1;z1;x2;y2;z2;d]. They return the abscissa (x) or the ordinate (y) of the point on the line, that is defined by a working name, at a distance d from P1. The working "wname1" must define a linear segment made of one only line. If the working is not correctly defined, the function returns the value 0.0.
geo[**pxill**;x1;y1;x2;y2;x3;y3;x4;y4] geo[**pyill**;x1;y1;x2;y2;x3;y3;x4;y4]	Return the abscissa (x) or the ordinate (y) of the intersection point between the two lines P1-P2 and P3-P4: x1;y1 = abscissa and ordinate of point P1 x2;y2 = abscissa and ordinate of point P2 x3;y3= abscissa and ordinate of point P3 x4;y4 = abscissa and ordinate of point P4. Return, in any case, the coordinate of point P1 if: • one or both lines are assigned with a null segment (P1=P2 and/or P3=P4); • the two lines are coincident; • the two lines are parallel.  <u>Examples</u> geo[pxill;0;0;45;0;300;300;0] = 150
geo[**pxill**;x1;y1;ang1;x3;y3;x4;y4] geo[**pyill**;x1;y1;ang1;x3;y3;x4;y4]	Return the abscissa (x) or the ordinate (y) of the intersection point between the two lines P1-ang1 and P3-P4: x1;y1 = abscissa and ordinate of point P1 ang1 = inclination angle of the first line x3;y3 = abscissa and ordinate of point P3 x4;y4 = abscissa and ordinate of point P4. Return, in any case, the coordinate of point P1 if: • the second line is assigned with a null segment (P3=P4); • the two lines are coincident;

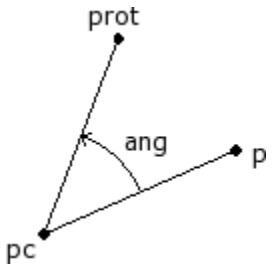
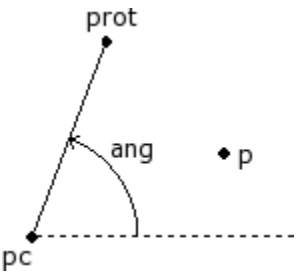
	• the two lines are parallel.  Examples: geo[pxill;0;0;45;0;300;300;0] = 150
geo[**pxill**;x1;y1;ang1;x3;y3;ang2] geo[**pyill**;x1;y1;ang1;x3;y3;ang2]	Return the abscissa (x) or the ordinate (y) of the intersection point between the two lines P1-ang1 and P3-ang2: x1;y1 = abscissa and ordinate of point P1 ang1 = inclination angle of the first line x3;y3= abscissa and ordinate of point P3 ang2 = inclination angle of the second line.  Return, in any case, the coordinate of point P1 if: • the two lines are coincident; • the two lines are parallel.
geo[**pxill**;"wname1+nn";x3;y3;x4;y4] geo[**pxill**;"wname1+nn";x3;y3;ang2] geo[**pxill**;"wname1+nn";"wname2+nn"] geo[**pxill**;"wname1+nn";"wname2+nn"(;nsol)] geo[**pyill**;"wname1+nn";x3;y3;x4;y4] geo[**pyill**;"wname1+nn";x3;y3;ang2] geo[**pyill**;"wname1+nn";"wname2+nn"] geo[**pxill**;"wname1+nn";"wname2+nn"(;nsol)]	The functions are similar to the functions: geo[**pxill**;x1;y1;ang1;x3;y3;x4;y4] and geo[**pyill**;x1;y1;ang1;x3;y3;x4;y4]. They return the abscissa (x) or the ordinate (y) of the intersection point between the two lines respectively defined by a "wname1" and "wname2" working name. The workings must define a linear segment or an arc – consisting of only one segment – or an arc of conic (ellipse/oval) or a tracked element. nsol = in the form with "wname1" and "wname2", if two intersection points are detected, it is possible to indicate which one of the two should be returned: • nsol=1, the first solution is returned (it is the default, if nsol is not assigned), • nsol=2 the second solution is returned, • nsol=3, 4 is returned to the third/fourth solution (the case may be significant in cases of intersection between two conical or an arc and a conic). 

	In the example in the figure: • "wname1" is a linear segment. • "wname2" is an arc. Both segments have 2 intersection points: • the intersection which is returned as a default is the one nearest to the point of beginning of the "wname1" segment (corresponding to nsol=1 or not assigned nsol). • The second intersection is returned, if a 2 value nsol is assigned. If the workings are not correctly defined, the function returns the value 0.0.
geo[**pxme**;x1;y1;x2;y2] geo[**pyme**;x1;y1;x2;y2]	Return the abscissa (x) or the ordinate (y) of the intermediate point of the segment P1-P2 (in plane): x1;y1 = abscissa and ordinate of point P1 x2;y2 = abscissa and ordinate of point P2. Examples geo[pxme;0;0;400;300] = 200 geo[pyme;0;0;400;300] = 150
geo[**pxme**;x1;y1; z1;x2;y2; z2] geo[**pyme**;x1;y1; z1;x2;y2; z2] geo[**pzme**;x1;y1; z1;x2;y2;z2]	Return the abscissa (x), the ordinate (y) or the height (z) of the intermediate point of the segment P1-P2 (in space): x1;y1;z1 = abscissa, ordinate and height of point P1 x2;y2;z2 = abscissa, ordinate and height of point P2. Examples geo[pxme;0;0;0;400;300;50] = 200 geo[pyme;0;0;0;400;300;50] = 150 geo[pzme;0;0;0;400;300;50] = 25
geo[**pxme**;"wname+nn"] geo[**pyme**;"wname+nn"] geo[**pzme**;"wname+nn"]	The functions are similar to the functions: geo[**pxme**;x1;y1; z1;x2;y2; z2], geo[**pyme**;x1;y1; z1;x2;y2; z2], geo[**pzme**;x1;y1; z1;x2;y2; z2]. They return the abscissa (x), the ordinate (y) or the depth (z) of the middle point of the line defined by a working name. The working must define a linear segment or an arc made of one only line. The function calculates the linear distance between the extreme points of the lines. The length of an arc is calculated by the function geo [**larc**;"wname"]. If the working is not correctly defined, the function returns the value 0.0.
geo[**pxcang**;x1;y1;xc;yc;ang;(sr)] geo[**pycang**;x1;y1;xc;yc;ang;(sr)]	Return the abscissa (x) or the ordinate (y) of a point P of a circle, determined from point P1 and with clockwise or counterclockwise angle of rotation: x;y = abscissa and ordinate of point P1 xc;yc = abscissa and ordinate of the circle centre Pc ang = angle of rotation sr = direction of rotation on circle (0=clockwise, #0 anticlockwise). In case of empty or unassigned field, the default angle value is 0. If points P1 and Pc coincide: the function returns the initial coordinate of P1. 

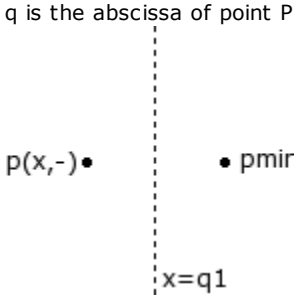
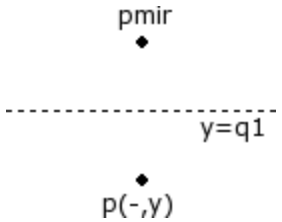
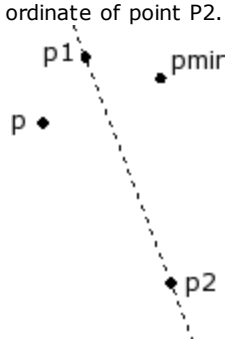
	Examples geo[pxcang;0;100;0;0;45] = 70.7107 geo[pxcang;0;100;0;0;45;1] = -70.7107
geo[**larc**;rad;ang]	It returns the length of an arc of circle, determined from the radius rad and with angular amplitude ang: If ang=0.0, it returns a null value. Examples: geo[larc;100;90] = 157.0796 geo[larc;100;360]=628.3185 ← corresponds to the whole circumference
geo[**larc**;x1;y1;xc;yc;ang]	Returns the length of the arc of circle, defined from the point P1 and with the size of an angle ang:  x1;y1 = abscissa and ordinate of the point P1 xc;yc = abscissa and ordinate of the centre Pc of the circle ang = size angle If the points P1 and Pc coincide, or if ang=0.0: the value is returned null. Examples: geo[larc;0;0;100;0;90] = 157.0796 geo[larc;0;0;100;0;360] = 628.3185 ← It is the entire circumference
geo[**larc**;x1;y1;x2;y2;xc;yc;(sr)]	Returns the length of the circle arc, defined from the point P1 until the point P2, clockwise or anticlockwise:  x1;y1= abscissa and ordinate of the point P1 x2;y2= abscissa and ordinate of the point P2 xc;yc= abscissa and ordinate of the centre Pc point of the circle sr = direction of rotation of circle (#0 counterclockwise; 0=clockwise= default, if not assigned) If the points P1 and Pc, or P2 and Pc coincide, the value is returned null. Examples: geo[larc;0;0;100;100;100;0] = 157.0796 geo[larc;0;0;100;100;100;0;1] = 471.238
geo[**larc**;"wname"]	The function is similar to the function: geo[**larc**;x1;y1;x2;y2;xc;yc;(sr)].

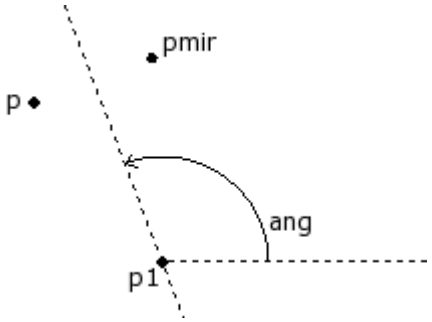
	It returns the length of the arc of the circle defined by a working name: • The working can now locate a circular arc lying on any plane, or a conic arc; • P1 and P2 are respectively the starting point and the end point of the arc. The length is calculated on the plane of the arc development. If the working is not correctly defined, the function returns the value 0.0.

Point rotation functions

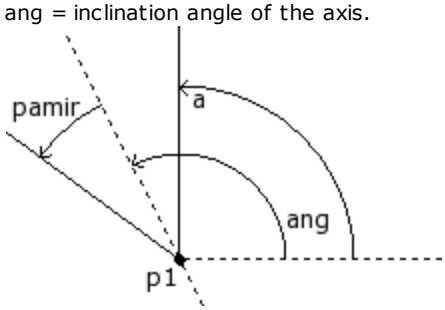
geo[**pxrot**;x;y;xc;yc;ang] geo[**pyrot**;x;y;xc;yc;ang]	Return the abscissa (x) or the ordinate (y) of a point P rotated by the angle ang with centre Pc (incremental rotation): x;y = abscissa and ordinate of point P xc;yc = abscissa and ordinate of point Pc ang = incremental rotation angle: • If ang>0 (positive): the point rotates in counterclockwise direction; • If ang<0 (negative): the point rotates in clockwise direction. If point P coincides with the centre or if ang=0: the function returns the initial coordinate.  Examples: geo[pxrot;70.7107;70.7107;0;0;-45] = 100 geo[pyrot;70.7107;70.7107;0;0;-45] = 0 geo[pxrot;100;0;0;0;90]=0 geo[pyrot;100;0;0;0;90]=100
geo[**pxrota**;x;y;xc;yc;ang] geo[**pyrota**;x;y;xc;yc;ang]	Return the abscissa (x) or the ordinate (y) of a point P rotated by the angle ang with centre Pc (absolute rotation): x;y = abscissa and ordinate of point P xc;yc = abscissa and ordinate of point Pc ang = final rotation angle. If point P coincides with the centre: the function returns the initial coordinate.  Examples: geo[pxrota;70.7107;70.7107;0;0;90] = 0 geo[pyrota;70.7107;70.7107;0;0;90] = 100

Mirror functions

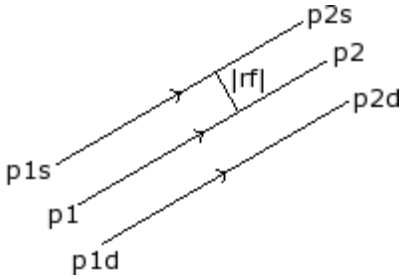
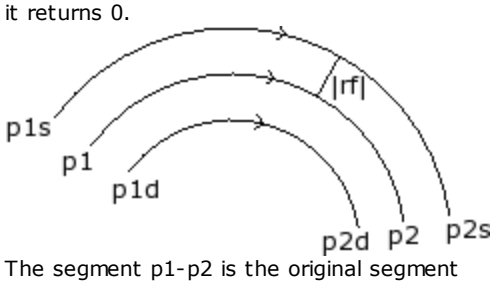
geo[**pmir**;q;q1]	Returns the coordinate of point P mirrored around a vertical or horizontal axis: • if q stands for x coordinate: the mirror axis is vertical and its equation is X = q1; the mirror axis is vertical: its equation is x = q1; q is the abscissa of point P (x=q)  • if q stands for y coordinate: the mirror axis is horizontal and its equation is Y = q1; the mirror axis is horizontal: its equation is y = q1; q is the ordinate of point P (y=q)  Example: geo[pmir;100;500] = 900
geo[**pxmir**;x;y;x1;y1;x2;y2] geo[**pymir**;x;y;x1;y1;x2;y2]	Return the abscissa (x) or the ordinate (y) of point P mirrored around the P1-P2 axis: x;y = abscissa and ordinate of point P x1;y1 = abscissa and ordinate of point P1 x2;y2 = abscissa and ordinate of point P2.  Examples: geo[pxmir;0;0;0;500;500;0]=500 geo[pymir;0;0;0;500;500;0]=500
geo[**pxmir**;x;y;x1;y1;ang] geo[**pymir**;x;y;x1;y1;ang]	Return the abscissa (x) or the ordinate (y) of a point P mirrored around the P1- ang axis: x;y = abscissa and ordinate of point P x1;y1 = abscissa and ordinate of point P1 ang = tilt angle of the line.

	 Examples: geo[pxmir;0;0;0;500;-45]=500 geo[pymir;0;0;0;500;-45]=500
geo[**pxmir**;"wname1+nn",x1,y1,x2,y2] geo[**pxmir**;"wname1+nn",x1,y1,ang] geo[**pxmir**;x;y;"wname2+nn"] geo[**pxmir**;"wname1+nn";"wname2+nn"] geo[**pymir**;"wname1+nn",x1,y1,x2,y2] geo[**pymir**;"wname1+nn",x1,y1,ang] geo[**pymir**;x;y;"wname2+nn"] geo[**pymir**;"wname1+nn";"wname2+nn"]	The functions are analogue to the functions: • geo[**pxmir**;x;y;x1;y1;x2;y2], geo[**pymir**;x;y;x1;y1;x2;y2] • geo[**pxmir**;x;y;x1;y1;ang], geo[**pymir**;x;y;x1;y1;ang]. They return the abscissa (x) or the ordinate (y) of the point P, mirrored around an axis. The point P and/or the axis are assigned by the name of a working: • "wname1" = name of the working assigning point P. It may correspond to a point or a setup or a profile segment (line or arc: point P is the end point of the segment); • "wname2" = name of the working assigning the linear segment: it must define a linear segment consisting of only one segment. If the "wname1" working is not correctly defined, the function returns value 0.0. If the "wname2" working is not correctly defined, the function returns the coordinate of point P.

Angle rotation functions

geo[**pamir**;a;ang]	Returns the angle (a) mirrored around the ang axis: a = angle to mirror ang = inclination angle of the axis.  Example: geo[pamir;30;90] = 150
geo[**pamir**;a;x1;y1;x2;y2]	Returns the angle (a) mirrored around the P1-P2 axis: a = angle to mirror x1;y1 = abscissa and ordinate of point P1 x2;y2 = abscissa and ordinate of point P2. Example: geo[pamir;30;0;0;0;100] = 150
geo[**pamir**;a;"wname+nn"]	This function is analogue to the function geo[**pamir**;a;x1;y1;x2;y2]. It returns the angle (a) mirrored around the axis defined by a working name. The working should define a linear segment made of one only segment. If the working "wname" is not correctly defined, the function returns the a value.

Segment correction functions with offset

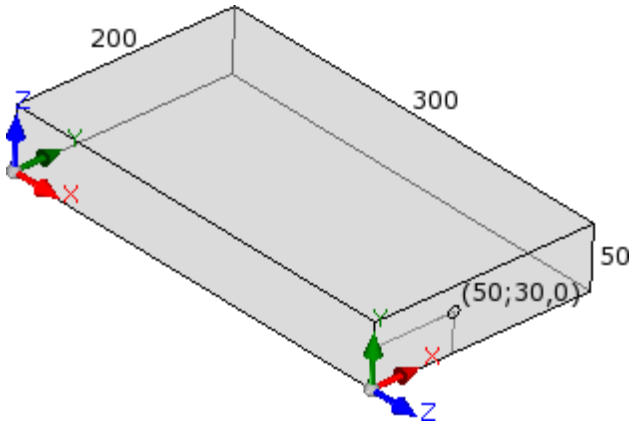
geo[**px1rf**;x1;y1;x2;y2;rf(;nret)] geo[**py1rf**;x1;y1;x2;y2;rf(;nret)] geo[**px2rf**;x1;y1;x2;y2;rf(;nret)] geo[**py2rf**;x1;y1;x2;y2;rf(;nret)]	They return the abscissa (x) or the ordinate (y) of the segment starting (1) or end point (2), after correcting a linear segment: x1,y1= abscissa and ordinate of the starting point of the segment P1 x2,y2= abscissa and ordinate of the starting point of the segment P2 rf=correction value to apply on the segment: • If the value is positive, the correction will be carried out to the left side of the segment. • if the value is negative, the correction will be carried out to the right side of the segment. nret= flag of outcome request. If a positive value is set, the function returns 1, if the geometric data are correct; otherwise it returns 0.  The segment p1-p2 is the original segment. If the value of rf is positive, the correct segment is p1s-p2s. If the value of rf is negative, the correct segment is p1d-p2D. In both cases the correct segment has a distance from the original one equal to the absolute value of rf. If the coordinates of the points P1 and P2 are the same, the function returns the coordinate of the starting point without applying the **correction**. Examples geo[px1rf;100;100;300;300;50] = 64.6447 geo[py1rf;100;100;300;300;50] = 135.3553 geo[px2rf;100;100;300;300;50] = 264.6447 geo[py2rf;100;100;300;300;50] = 335.3553
geo[**px1rf**;x1;y1;x2;y2;xc;yc;sr;rf;(nret)] geo[**py1rf**;x1;y1;x2;y2;xc;yc;sr;rf;(nret)] geo[**px2rf**;x1;y1;x2;y2;xc;yc;sr;rf;(nret)] geo[**py2rf**;x1;y1;x2;y2;xc;yc;sr;rf;(nret)]	They return the abscissa (x) or the ordinate (y) of the segment starting (1) or end point (2), after carrying out the correction to an arc. x1,y1= abscissa and ordinate of the starting point of the segment P1 x2,y2= abscissa and ordinate of the starting point of the segment P2 xc,yc= abscissa and ordinate of the centre of the arc. sr= direction of circle rotation. If sr=0, it is a clockwise direction, otherwise it is considered anti-clockwise rf= correction value to apply on the segment: • If the value is positive, the correction will be carried out to the left side of the segment • if the value is negative, the correction will be carried out to the right side of the segment nret=flag of outcome request. If a positive value is set, the function returns 1, if the geometric data are correct; otherwise it returns 0.  The segment p1-p2 is the original segment

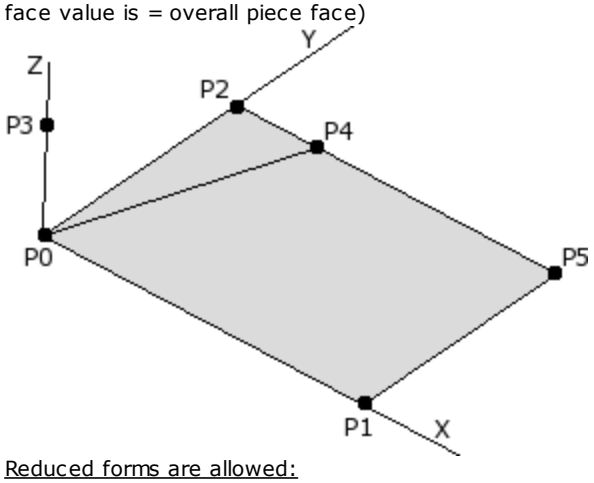
	If the value of rf is positive, the correct segment is p1s-p2s. If the value of rf is negative, the correct segment is p1d-p2D In both cases the correct segment has a distance from the original one equal to the absolute value of rf. If the arc is not valid (different initial and final radius) or the internal required correction has a greater value than the radius of the arc, the function returns the point coordinate without applying the correction. Examples geo[px1rf;0;0;200;0;100;0;0;50] = -50 geo[py1rf;0;0;200;0;100;0;0;50] = 0 geo[px1rf;0;0;200;0;100;0;0;-50] = 50 geo[py1rf;0;0;200;0;100;0;0;-50] = 0
geo[**px1rf**;"wname+nn";rf;(nret)] geo[**py1rf**;"wname+nn";rf;(nret)] geo[**px2rf**;"wname+nn",rf;(nret)] geo[**py2rf**;"wname+nn";rf;(nret)]	They return the abscissa (x) or the ordinate (y) of the segment starting (1) or end point (2), after applying the correction to it. Before the current working, the line is defined by a working name and can be a linear segment or an arc. "wname"=name of the working assigning the line to which the correction needs be applied. rf=correction value to apply on the segment. If the value is positive, the correction will be carried out to the left side of the segment: • If the value is positive, the correction will be carried out to the left side of the segment, • if the value is negative, the correction will be carried out to the right side of the segment. nret= flag of outcome request. If a positive value is set, the function returns 1, if the geometric data are correct; otherwise it returns 0. The selected working must define an arc on a plane xy or a linear segment made of one only line.

Functions of coordinate conversion and of faces information reading

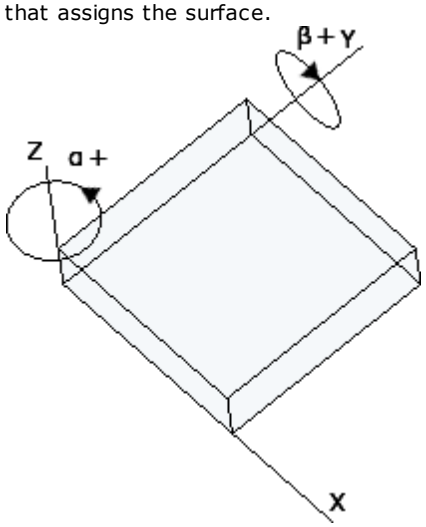
For all these functions, if the programming of machining in the face-piece, with the possibility of assignment of automatic faces and field name, for defining parameters of the face (example: *nside*) are permitted syntax:
- if *nside* = 100 interprets the automatic face last assigned, upstream of the current working;
- It is also recognized forms with nside replaced by "*nameFace*" = name of automatic face, always last assigned upstream of the current working with the specified name.

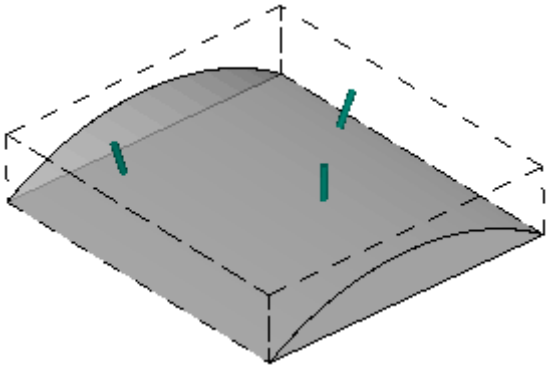
geo[**pxp**;x;y;z;(nside)] geo[**pyp**;x;y;z;(nside)] geo[**pzp**;x;y;z;(nside)]	Returns the coordinate (x; y; z) of the indicated point, that has been converted from the local face system (nside) to the absolute coordinates of the piece: x;y;z = point coordinates in the local system of the face nside = custom face number (if real face: it is custom number). The face number is optional: if unassigned, it takes the active face. The function returns the initial coordinate if: • nside assigns an invalid face value; • nside is not specified and the active face is the overall piece face; • in assignment of (o, v, r) variables or variable geometry and nside indicates a fictive face. If *nside* indicates a real face: the function operates although the face is not assigned on the piece. If the working is programmed in face-piece and nside=100, it is considered the last automatic face, that has been assigned, before the selected working. Let this figure be considered:

	 • length of the piece=300; height of the piece=200; thickness of the piece=50 • on face 4 a point is shown at the positions 50;30;0 • in absolute coordinates of piece, the point has coordinates 300;50;30 Examples: geo[pxp;100;100;-5;1]: this function returns the abscissa of the point (100;100;-5) from the local system of face 1 to the absolute coordinates of the piece; geo[pxp;100;100;-5]: this function returns the abscissa of the point (100;100;-5) from the local system of the active face to the absolute coordinates of the piece
geo[**pxf**;x;y;z;(nside);(nsorg)] geo[**pyf**;x;y;z;(nside);(nsorg)] geo[**pzf**;x;y;z;(nside);(nsorg)]	Returns the coordinate (x; y; z) of the indicated point, that has been converted from the local face system (nsorg) to the local system of the face (nside): x;y;z = point coordinates in the local system of the face nsorg nside = number of the target face (custom number) (in case of empty assignment: the default face value is = current face) nsorg = number of the source face (custom number) (in case of empty assignment: the default face value is = overall piece) Reduced forms are allowed: • with 4 arguments. Example: geo[pxf;x;y;z]: • nside: it takes the active face; • nsorg: it takes the overall piece face value (-1); • with 5 arguments. Example: geo[pxf;x;y;z; nside]: • nsorg: it takes the overall piece face value (-1). In case of unassigned or invalid nsorg: the function converts from the absolute coordinates of the piece to the local system of the face (nside). The function returns the initial coordinate if: • nside assigns an invalid face value; • nside is not specified and the active face is the overall piece face; • in assignment of (o, v, r) variables or variable geometry and nside indicates a fictive face. If nside and/or nsorg indicates a real face: the function operates although the face is not assigned on the piece. If the working is programmed in face-piece and nside=100 or nsorg=100, it is considered the last automatic face that has been assigned, before the selected working. Examples: geo[pxf;100;100;-5;1;7]: this function returns the abscissa of the point (100;100;-5) from the local system of face 7 to the local system of face 1; geo[pxf;100;100;-5;;7]: this function returns the abscissa of the point (100;100;-5) from the local system of face 7 to the local system of the active face; geo[pxf;100;100;-5;1]: this function returns the abscissa of the point (100;100;-5) from the absolute coordinates of the piece to the local system of face 1;

	geo[pxf;100;100;-5]: this function returns the abscissa of the point (100;100;-5) from the absolute coordinates of the piece to the local system of the active face;
geo[**px**;(np);(nside);(dd);(ndest)] geo[**py**;(np);(nside);(dd);(ndest)] geo[**pz**;(np);(nside);(dd);(ndest)]	Returns the coordinate (x; y; z) of the face edge (nside) in coordinates of the local system of the face (ndest): np = face edge identification number (in case of empty assignment: it uses =0) • 0: face origin (point P0) • 1: point along the x+ axis (point P1) • 2: point along the y+ axis (point P2 recalculated) • 3: point along the z axis, toward z clearance (point P3) • 4: start point P2 • 5: fourth point of the face rectangle (point P5) nside = number of the source face from which the edge is read (custom number) (in case of empty assignment: the default face value is = current face) dd = assignment of the required point from P0 (it is not significant if np=0) (in case of empty assignment: no value is applied) ndest = number of the target face on which the edge is read (custom number) (in case of empty assignment: the default face value is = overall piece face)  <u>Reduced forms are allowed:</u> • with 2 arguments. Example: geo[px;np]: • nside: it takes the active face; • dd: the distance is the default distance (face dimension along the x-axis); • ndest: it takes the overall piece face value (-1); • with 3 arguments. Example: geo[px;np;nside]: • dd: the distance is the default distance (face dimension along the x-axis); • ndest: it takes the overall piece face value (-1); • with 4 arguments. Example: geo[px;np;nside;dd]: • ndest: it takes the overall piece face value (-1). If np has an invalid value: the function operates as with np=0. If np=4: the function does not interpret dd. The function returns the null coordinate (=0.0) if: • nside assigns an invalid face value; • nside is not specified and the active face is the overall piece face; • in assignment of (o, v, r) variables or variable geometry and nside indicates a fictive face. If nside indicates a real face, the function operates although the face is not assigned on the piece. If nside shows a face assigned as a surface: • for the (P0; P2; P4; P3) points: the function returns the values calculated for the first geometric element that assigns the surface • for the (P1; P5) points: the function returns the values calculated for the last significant geometric element that assigns the surface.

<table>
<tr><td></td><td><u>Examples:</u>
geo[px;0]: this function returns the abscissa of a point P0 of the active face, converted to the absolute coordinates of the piece.
geo[px;0;7;;1]: this function returns the abscissa of a point P0 of face 7, converted to the local system of face 1</td></tr>
<tr><td>geo[alfa;(nside)]
geo[beta;(nside)]</td><td>Return respectively the rotation angle (alfa) and the rotation axis around Y axis (beta) to assign to a tool in order to make it work perpendicular to the nside face:
nside = face number (custom number) (in case of empty assignment: the default face value is = current face).
In any case the function returns 0, if:
nside assigns an invalid face value;
nside corresponds to a face which is not assigned on the piece;
the face is assigned with invalid geometry;
in assignment of (o, v, r) variables or variable geometry and nside indicates a fictive face.
If nside indicates a real face: the function operates although the face is not assigned on the piece.
If nside shows a curved face or a surface, the return is determined, if you consider the plane face of the construction.

If nside reports a face assigned as a surface, the function returns some values calculated for the first geometric element that assigns the surface.
 
With reference to the figure above:
beta rotates around the y axis
alfa rotates around the z axis.
With reference to the six real faces of the parallelepiped, alfa and beta values are assigned as follows: alfa and beta are assigned as follows:<table><tr><td>Face</td><td>Beta; Alfa</td><td>Note</td></tr><tr><td>1</td><td>(0;0)</td><td>Alfa: any</td></tr><tr><td>2</td><td>(180;0)</td><td>Alfa: any</td></tr><tr><td>3</td><td>(-90;90); (90;-90)</td><td></td></tr><tr><td>4</td><td>(-90;180); (90;0)</td><td></td></tr><tr><td>5</td><td>(-90;-90); (90;90)</td><td></td></tr><tr><td>6</td><td>(-90;0); (90;180)</td><td></td></tr></table>If in the configuration of TpaCAD, the inversion of the rotation axis around Y axis is set, the values of the rotation axis around Y axis in the table must have to be inverted signs.</td></tr>
<tr><td>geo[alfa;x;y;z;(nside)]
geo[beta;x:y;z;(nside)]</td><td>These functions are similar to the previous ones that can indicate the point of face against which to determine the</td></tr>
</table>

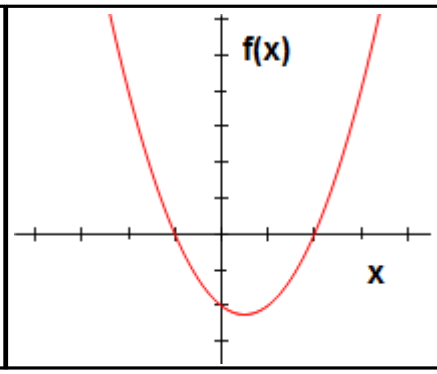
	values of the both the rotary axes (alpha, beta) to be assigned to a tool that works perpendicularly to the nside face: x;y;z= coordinates of the point in the local face system (nside) nside = number of the face (if the assignment is empty: use = current face). The function returns 0, if: • nside assigns an invalid value of face; • nside indicates a face not assigned in the piece; • the face is assigned with an invalid geometry; • (o, v, r) variables, variable geometry or custom section are assigned; nside indicates a non-real face. If *nside* indicates a real face or anyway a flat face, the functions correspond exactly to the formats without point coordinates and, in the case of a real face, they work even if the face is not assigned in the piece. If *nside* indicates a curved face or a surface, the returned value is determined by considering the position assigned with the three coordinates. With reference to the figure it is clear that the vertical direction changes, when the position of the represented curved face changes. 
geo[**lface**;(nside)] geo[**hface**;(nside)] geo[**sface**;(nside)] geo[**zface**;(nside)]	Return respectively: length (lface), height (hface), thickness (sface) and z-axis orientation (zface) of the nside face: nside = face number (custom number) (in case of empty assignment: uses = active face). In any case the function returns 0, if: • nside assigns an invalid face value; • nside corresponds to a face which is not assigned on the piece; • in assignment of (o, v, r) of variables, of variable geometry or custom setion; *nside* indicates a non-real face. If *nside* indicates a real face: the function operates although the face is not assigned on the piece.
geo[**rface**;(nside)] geo[**cxface**;(nside)] geo[**cyface**;(nside)] geo[**czface**;(nside)]	They return specific information on curved face. Respectively: radius of curvature (rface), coordinate of the curvature centre (cxface; cyface; czface) of the face nside: nside = number of the face (if the assignment is empty: use = empty face). Anyway, the function returns 0, if: • nside assigns a face invalid value; • nside shows a non curved face or assigned as a surface; • in assignment of variables (o, v, r), of variable geometry or custom section; *nside* shows a non-real face.
geo[**plface**;(nside)]	Return specific data of curved face or surface: • curvature plane (0=xz; 1=yz), if curved face • development axis (0=x; 1=y), if surface. The function returns 0 anyway if: • nside assigns an invalid face value; • nside indicates a non-curved face or surface.
geo[**isface**;(nside)]	Returns the existence flag for the face in the piece (1 = face exists, 0 = face does not exist):

	nside = face number (custom number) (in case of empty assignment: the default face value is = current face). The function returns 0 if: • nside assigns an invalid face value; • nside corresponds to a face which is not assigned on the piece; • in assignment of (o, v,r) variables or variable geometry and nside indicates a fictive face. If *nside* shows a curved face, the function returns 2. If *nside* reports a face assigned as a surface, the functions returns 3.
geo[**simil**;(nModo);(nside)]	Returns the number of the real face verifying criteria similarity of the face that is defined by nside: nMode = similarity criterion. If the parameter is not assigned, the default is 0: • 0: verifies that the tool has the same entry direction on the face. The face (nside) is generated by translation of one or more axes and possible rotation on the plane xy. • 1 (different from 0): the face (nside) is generated by simply translation of one or more axes. nside = number of the face (custom number). If the parameter is not assigned, the number of the current face is used. In any case the function returns 0, if: • nside assigns a non valid face value • nside shows a fictive face not assigned in the piece • nside shows a curved face or assigned as a surface • in assignment of (o, v,r) variables or variable geometry and nside indicates a fictive face. If the returned value is significant (different from 0), it corresponds to a face custom number. If *nside* assigns a value of real face: the function becomes nside again.
geo[**pr1**;(nside)] geo[**pr2**;(nside)] geo[**pr3**;(nside)] geo[**pr4**;(nside)] geo[**pr5**;(nside)]	Return the additional parameters of fictive or automatic face: nside = number of the face (custom number). If the parameter is not assigned, the default is the number of the current face. The function returns 0, if: • nside assign an invalid value of face; • nside corresponds to a face which is not assigned on the piece; • in assignment of (o, v,r) variables, or variable geometry, or custom section.

Algebraic functions

The algebraic functions are functions of advanced programming

geo[**equ**;a;b;c;(nret)]	Solve an algebraic equation of the second degree $\mathbf{ax^2+bx+c=0}$ a, b, c=coefficients of the equation nret = flag to outcome request (if not set is 0): • set with a null value returns the number of solutions (roots) of the equation (0 if no solution; 1 if it allows a solution (case of: a=0); otherwise 2); • set 1 to obtain the first solution, 2 for the second. In the case of any viable options, the value assigned to the solutions is 0.0. In the case of a single viable solution, the value assigned to the solutions is identical. From a geometric point of view: • the chart of the second degree function $\mathbf{f(x)= ax^2+bx+c}$ in the Cartesian plane is a parabola; • the solutions of the function are the values of x in which the chart of the function touches the X-axis. Examples geo[equ;1;-1;-2] = 2 <- the equation has 2 solutions geo[equ; 1;-1;-2;1] = 2 geo[equ; 1;-1;-2;2] = -1 the figure shows the geometric meaning of the example



Access functions to programmed working information

The function should be considered as function of advanced programming.

geo[**param**;"wname+nn";"pname";(nret)] geo[**param**;"wname+nn";pID;(nret)]
It returns the value of an information or the parameter of programmed working: • "wname" = name of the working searched before the current working. • "pname" = parameter name (ASCII). According to the syntax, the name should be placed between double quotation marks, in lower-case letters. • pID =parameter's identification number. pID use form is recommended to advanced users only. • nret = flag required outcome (that is 0 if it is not assigned). If it is positive (1), the function returns: • 1 if the parameter and working search is correct   • 0 otherwise. For the "pname" argument (ASCII parameter name) some remarkable cases are managed, as follows: • working generic information: • "#cop" reads the operative code of the working   • "#tip" reads the working typology; (0 = punctual; 1 = setup; 2 = arc segment; 3 = linear segment; 4 = logical; 5 = complex);   • "#tips" reads the sub-typology of the working;   • "#tipt" reads the technological typology of the working;   • "#prog" reads the consecutive sequence number of workings.   • "#list" reads the number of the workings in the list of the workings. The value can be significant (i.e.: >0), if the working is complex (subroutine, macro, STOOL) or if it is a multiple segment of a profile (examples: fillet, multiple arcs, conic section)   • "#vl" reads the value of the L property   • "#vb" reads the value of the B property   • "#vo" reads the value of the O property   • "#vm" reads the value of the M property   • "#vk" reads the value of the K property   • "#vk1" reads the value of the K1 property   • "#vk2" reads the value of the K2 property • specific information of complex working (subroutine or macro-program)   • "#subxi", "#subyi", "#subzi": read the corresponding coordinate of the first position worked;   • "#subxe", "#subye", "#subze": read the corresponding coordinate of the 'last position worked;   • "#subxn", "#subxp": read the position corresponding to the x-coordinate of minimum/maximum overall dimension of the working;   • "#subyn", "#subyp": read the position corresponding to the y-coordinate of minimum/maximum overall dimension of the working;   • "#subzn", "#subzp": read the position corresponding to the z-coordinate of minimum/maximum overall dimension of the working. If the function is used in the "*geo[param;..]") remarkable string formalism: • "#name" reads the name of the working. Examples geo[param;"w1", "x"]: returns the value of the X-coordinate of the "w1" working, according to the programming and to the assignment (absolute/relative,...) geo[param;"w1";31]: returns the value of the parameter of ID=31 of the "w1" working: it is the X position of the centre of an arc. If "w1" corresponds to an arc programmed through 3 points, the function reads the X coordinates calculated for the centre (use 32 for the Y coordinate).
geo[**lparam**;"wname+nn";"pname";nlist1;(nlist2);(nlist3);(nlist4);(nlist5);(nlist6);(nret)]

geo[**lparam**;"wname+nn";pID;nlist1;(nlist2);(nlist3);(nlist4);(nlist5);(nlist6);(nret)]

It returns the value of a data or the parameter of a working found in an expanded list of a given working:

- "wname" = name of the working searched before the current working.
- "pname" = (ASCII) name of the parameter. The syntax requires that the name must be placed between double quotation marks and in lowercase.
- pID = numeric identifier of the parameter. We recommend that only advanced user should employ the form with pID.
- nlist1 = progressive number of working in the "wname" expanded list. Set a strictly positive value (>0) and not greater than the value returned by the function:
  geo[param;"wname";"#list"]
  The argument is compulsory and its value must be valid.

- nlist2=increasing ID number of the working in the expanded working list identified by:
  geo[lparam;"wname";"#cop";nlist1]
  Set a strictly positive value (>0), not greater than the value returned by the function:
  geo[lparam;"wname";"#list";nlist1]

- …
- nlist6=increasing ID number of the working in the expanded working list identified by:
  geo[lparam;"wname";#cop";nlist1;nlist2;nlist3;nlist4;nlist5]
  Set a strictly positive value (>0), not greater than the value returned by the function:
  geo[lparam;"wname";"#list";nlist1;nlist2;nlist3;nlist4;nlist5]
  The evaluation of the arguments (nlist2,…,nlist6) shall be interrpted at the first non-strictly-positive value, stopping the browsing in the expanded lists at the previous level.
- nret = flag of request for result (it is 0, if it is not assigned). If the value is positive (1), the function returns:
  - 1 if the search of the working and of the parameter is correct
  - otherwise 0.

As for the argument "pname" (ASCII name of te parameter), all the significant cases provided for the geo[**param**;…] function are managed.
The main use of the function concerns the **Advanced use of the programmed Tools**.

Examples
geo[lparam;"w1";"x";1]: it returns the value of the X position of the first working in the expanded list of "w1"
geo[lparam;"w1";"x";1;2]: the first working in the expanded list of "w1" must have its own expanded list, with at least two workings in the list, and the function returns the value of the X position of the second working in the expanded list.

geo[**sub**;"pname";(nret)]
geo[**sub**;pID;(nret)]

It returns the value of a parameter or data of a working regarding the application of a subroutine or macro, and whose use is considered in the text of that subroutine or macro:
- "pname" = parameter name (ASCII). According to the syntax, the name should be placed between double quotation marks, in lower-case letters.
- pID = identification parameter number
- nret = flag of outcome requested (it is 0 if it is not assigned). If it has positive value (1), the function returns:
  - 1 if the working and parameter search are corrected;
  - 0 otherwise.

As gor the argument "pname" (ASCII name of the parameter), some remarkable cases are managed:
- "#cop" reads the operating code of the working;
- "#tips" reads the sup-type of the working;
- "#tipt" reads the technological type of the working;
- "#prog" reads the progressive number in the working list.

The same arguments that end by '0' (e.g.: "#cop0") return the data regarding the main application (i.e.: in the program list) of subroutine or macro.
In these cases, the argument does not correspond to a parameter ASCII name, but to a piece of information of a working.

If the working is used in a chain-type remarkable formalism ("*geo[sub;…]"):
- "#name" reads the name of the working
- "#name0" reads the name of the working corresponding to the main application.

Example:

- The SUB code is programmed with recall of the SUB1 subroutine and (X=100; Y=200) application point
- in SUB1 let us program a HOLE working applied in:
  - X= 100+geo[sub"x"]
  - Y= 50+geo[sub;"y"]

The hole of SUB1 will be in (X=200; Y=250).

Example:
- In the previous example, we assign the SUB code a name ="aa", then we look at some programming in the subroutine SUB1 (e.g. for private r variables):
  - geo[sub;"#cop"] -> returns the operating code of the working that calls SUB (=2010)
  - "*geo[sub;!#name"]" -> returns the name of the working that calls SUB (="aa"). The assignment is significant for the string r variable.

geo[**sub**;"wname";"pname";(nret)]
geo[**sub**;"wname";pID;nret]

As already seen with the previous formats, it returns the value of a parameter or data of a working regarding the application of a subroutine or macro, and whose use is considered in the text of that subroutine or macro.

The difference of these formats is the presence of the argument "wname". The interpretation of the argument is assigned on remarkable strings:
- "prcopsetup1", "prcopsetup2": they indicate, respectively, the first and the second working indicated by name in the parameter PRCOPSETUP of the complex code
- "prcoppoint1", "prcoppoint2": they indicate, respectively, the first and the second working indicated by name in the parameter PRCOPPOINT of the complex code.

The parameters (PRCOPSETUP, PRCOPPOINT) can be used when defining complex codes in order to implement automatic technology substitutions, respectively for setup or point workings.
Using the parameter PRCOPSETUP offers many applications in the base database, and normally corresponds to the parameter *Element of reference for the Setup*: using "prcopsetup2" (or "prcoppoint2") corresponds to those cases in which the complex working wants for the distinct assignment of two technologies.

ATTENTION: the automatic technology substitution with the parameters PRCOPSETUP (PRCOPPOINT) implies that the macro-program (or subroutine) follows some writing rules. For further details, please see the dedicated documentation.

As for the remaining arguments, what was said earlier for the previous formats is valid:
- "pname" = parameter name (ASCII). The syntax wants the name between double quotes, written in lowercase.
- pID = ID number of the parameter
- nret = request flag for the result (is 0 if not assigned). If it is a positive value (1), the function returns:
  - 1 if the query on working and parameter is correct;
  - 0 otherwise.

Please note that the argument *nret* is not optional in the second format. The lack of the argument interpret the formalism *geo[**sub**;"pname";(nret)]*, corresponding to the previous formats.

For the argument "pname" (ASCII name of the parameter) some remarkable cases are managed:

- "#cop" reads the operating code of the working;
- "#tips" reads the subtype of the working;
- "#tipt" reads the technology type of the working;
- "#prog" reads the progressive number in the working list.

In these cases, the argument does not correspond to a parameter ASCII name, but to a working data.

If the function is used in a string-type remarkable formalism ("*geo[sub;…]"):
- "#name" reads the name of the working.

The function works if used in the development of a complex code.

Example:
Let us define a complex code that calls the subroutine SUB1 and manages the parameter PRCOPSETUP in string format. In SUB1, we use the programming

"geo[sub;"prcopsetup1";"#cop"]"

for instance, in private r variable, and we evaluate different assignments for the parameter PRCOPSETUP:

- "aa" -> the function "geo[sub,…" returns the operating code of the working assigned earlier with name "aa", but only if it corresponds to a setup working

- "tec\aa" -> the function "geo[sub,..." returns the operating code of the working corresponding to the global technology indicated (0 if not assigned)
- "aa;bb" -> it allows using "aa" through "prcopsetup1" and "bb" through "prcopsetup2".

Example:
Continuing with the example, and supposing we have PRCOPSETUP="aa", we will see how to read the working parameters:
- geo[sub;"prcopsetup1";205] -> reads the tool programmed in the working called "aa"
- geo[sub;"prcopsetup1";"f"] -> reads the entering speed programmed in the working called "aa".

Custom functions

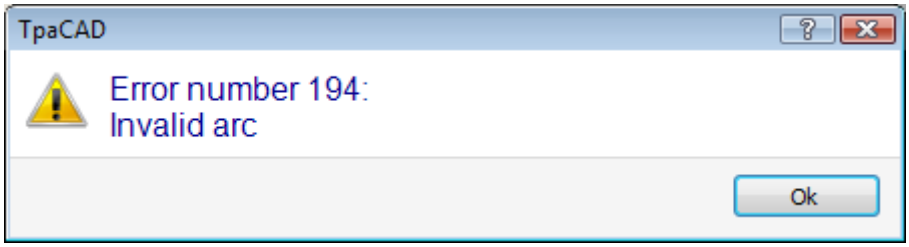


Custom functions are only available in Professional mode.

funxxxxx[n1;..;n30]	They run the corresponding custom function, assigned with the default name The maximum arguments number is 30. Error conditions: • 123: operands number = 0; • 124: operands number > 30; • 134: too many calls of custom functions (maximum 5) • 135: use of custom function in a not valid context (it signals that a custom function call already stored in stack or an illicit use of a private custom function is programmed).

12 Error Messages

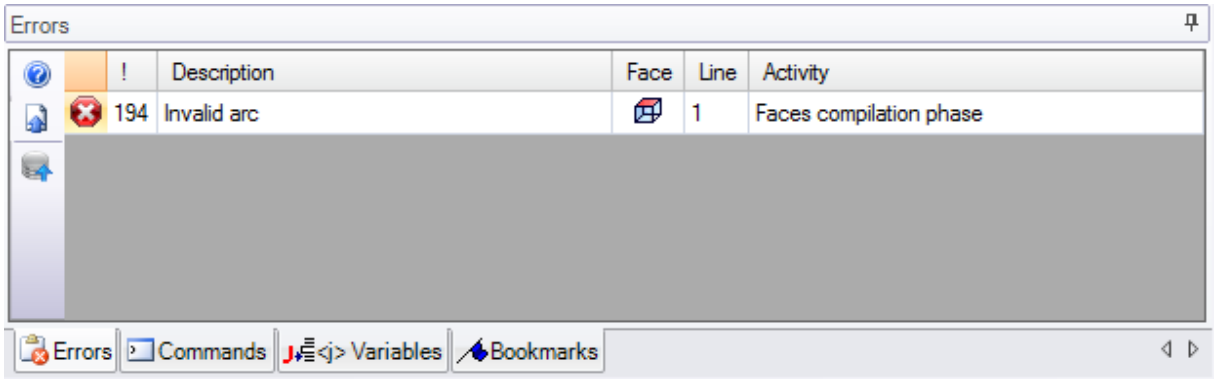
The message of the figure is displayed, for example in case of an error in entering or editing a working of type of arc:



Pressing the **[?]** button calls up the help window that describes the error.

In the display of errors, errors are reported overall diagnosed in the program, showing in addition to the description of the error, including the face, line and program activities in progress when the error occurred.

If you select the icon **Go to the corresponding position**, it jumps to the program line where the error occurred selected.



12.1 General Errors

It is about messages, directly displayed in the window, which are strictly connected to the execution of program commands. They can identify:

- actual errors resulting from a failed procedure (example: failure to load a program from a file),
- simple warnings: messages which inform about a specific situation (example: the request for a tool applied to unsuitable workings).

1 - Error in procedure

Explanation:
It is about a general error which is not otherwise identified.

Context:
Any context is fine. However, it is necessary to specify that, while each different error condition is usually precisely identified by a detailed message (for example: the reporting of a detailed error). The documentation provided in this section is only applicable to general cases, although the number of situations which generate this error is limited.

2 - Error in memory allocation

Explanation:
there is not enough system memory to execute the required procedure. It is a critical error: you are recommended to close the program and perform the necessary system checks.

Context:
any

5 - Error in file access

Explanation:
An error in file access (in reading or saving) has occurred. It can identify such situations: improper file addressing, access to file not allowed, empty file, invalid file format.
The signal can denote a problem in accessing the folder or a single file.
Trying to read or to record a program, the signal can show a lock situation since another application program is using that program.

Context:
This error message may result from the request for any of the following:
- program loading or saving;
- piece matrix loading or saving;
- copy of workings to Clipboard (the error can be due to the creation of an auxiliary temporary file);
- custom function file loading or saving (in loading this error can be caused by the identification of an invalid file format).

6 - Error accessing the Clipboard

Explanation:
An error while accessing the Clipboard has made impossible to save or retrieve data. This kind of error is strictly connected to an improper system functioning.

Context:
This error message may result from the request for any of the following:
- copy of workings to Clipboard (working edit commands: Copy, Delete);
- clipboard data retrieval (working edit commands: Paste; general tools: Translation, Rotation, Mirrors, Repetitions, Exploded view).

7 - Error accessing an Undo temp file

Explanation:
An error in accessing one of the temporary files created to support the UNDO function has occurred. This error can be due to external temp-file corruption or to error conditions resulting from the access to the storage peripheral device. It is about an error strictly connected to an improper system functioning.

Context:
The error message can result from the execution of each program edit command which it is possible to cancel:
- working edit commands: Edit, Insert, Paste, Delete, Selective replacements (parameters and/or properties);
- tools.

13 - System level doesn't allow execution of this operation

Explanation:
The activated command cannot be executed because the user access level is lower than that required by the command.

Context:
This error message may result from the request for any of the following:
- macro-program loading (required access level: Constructor)
- loading of a program which has a read access level higher than the level set
- loading of a program which has applied Professional level tools in a system working with a standard level key
- storage of a program that has a write access level higher than the level set

18 - Current working is invalid

Explanation:
General error related to the application of a command to the current working.

Context:
This error message may result from the request for any of the following:
- working edit commands: Edit, Insert, Paste, Delete, Selective replacements (parameters and/or properties)
- tools

36 - Maximum number of workings per face was overcome

Explanation:
It is no longer possible to enter workings into the current face, because the maximum allowed number has been reached (1000000).

Context:
This error message may result from the request for any of the following:
- program loading
- subroutine (or macro) application because of excessive number of read lines or excessive number of lines resulting from the application of repetitions or emptying operations
- working insert commands: Insert, Paste
- application of tools requiring the insertion of workings because of excessive number of lines as a result of the tool application.

While reading the program, the error can be retrieved: the exceeding lines are ignored.

38 - Can't insert this working on the current face

Explanation:
It has been required inserting a working on a face where the working itself is disabled.

Context:
This error message may result from the request for any of the following:
- insertion of a few types of workings on Piece-Face. In fact, while for each face the workings which are not handled are usually disabled, in Piece-Face all workings are always enabled and an error message appears if it is tried to enter a disabled working.

39 - The tool can't use a required working

Explanation:
The tool cannot be activated because, among the workings which have been configured for the application, a working essential for the tool to function is not available. It is always about a basic work on profile for a type of segment. Basic profile codes:
- L01 [code = 2201] for linear segment;
- A01 [code = 2101] for arc assigned in xy plane;
- A05 [code = 2105] for arc assigned in xz plane;
- A06 [code = 2106] for arc assigned in yz plane;
- A10 [code = 2110] for arc assigned in xyz plane.

Context:
Tools which edit or generate profiles:
- all profile tools (Break profile, Displace setup point, …)
- Advanced profile tools (Text generation, Emptying of area, Profiles cut, Profile building)

41 - Errors assigning working properties

Explanation:
you entered an incorrect value for a property of processing (level, construct, Fields M, O, K, K1, K2). For example, a parameter setting does not recognize, a value outside the range of minimum and maximum assignable.

Context:
You have entered an incorrect value being edited or added during working or total allocation of property.

42 - No modifications or replacements have been affected

Explanation:
The activated command has not generated any changes.

Context:
This error message may result from the request for any of the following:
- working edit commands: Edit, Insert, Selective replacements (parameters and/or properties);
- Tools (general, profile, advanced profile tools).

49 - This tool may only be applied to profiles

Explanation:
A profile tool has been activated for executions which are not works on profile.

Context:
This error message may result from the request for a profile tool, with selected workings which do not belong to a profile.

281 - File reading: unexpected file end

Explanation:
an error occurred during the file read. The report shows that the file format is valid, but his syntax is wrong.

Context:
You have reached the end of the file with a section that was being read. The report can indicate that the file has been tampered or that its generation does not observe the syntax required.
This is a situation that you can retrieve by forcing the closure of each open section and by closing the file read.
An example of the final part of a TCN program is reported as follows: on the left the end of the file (marked by: EOF) occurs with the section of open face; in the central column the section ends correctly.

…	…	Opens the face
SIDE#1{	SIDE#1{	section
…	…	..
#2201{ ::WTI #1=532.89 … }W	#2201{ ::WTI #1=532.89 … }W	Last working
EOF	**}SIDE**	
	EOF	

282 - File reading: section closure not found

Explanation:
an error occurred during the file read. The report shows that the file format is valid, but his syntax is wrong.

Context:
The correspondence between header and closure lines of a file section is missing. The report can indicate that the file has been tampered or that its generation does not observe the syntax required.

283 - File reading: invalid face identifier

Explanation:
an error occurred during the file read. The report shows that the file format is valid, but one of his syntax is wrong.

Context:
More specifically: it corresponds to the assignment of a face section reported with an invalid numbering.
In the message the wrong number as read by the file can be displayed. This is a situation you can retrieve: the corresponding section is ignored.

284 - File reading: working identification number not valid

Explanation:
an error occurred during the file read. The report shows that the file format is valid, but its syntax is wrong.

Context:
This error occurs when the file read in ASCII encoding meets a working that does not exist in the working database, or when you encounter an invalid identifier for the file read in the internal encoding.
In the first case a working is recalled by the ASCII name (examples: HOLE, SETUP) while in the second case a working is recalled by the operating code (examples: 81,88).
In the error message the name of the working can be displayed as read by the file.
This is a situation you can retrieve: if in the database the NOP instruction (Not Operation) is available, this one is assigned for this line of program, otherwise the line is deleted.

286 - File reading: error file decoding

Explanation:
An error occurred while reading the program. The report indicates that the file has been recognized as an Encrypted format, but an error occurred while decoding.

287 - File reading: the program is not compatible with environment assignments

Explanation:
an error occurred while reading the program. The warning indicates the presence of unmanaged programs in the file, based on the current configuration, and corresponds to situations that may substantially change the assignment of the program itself. In particular:

- there are programmed surfaces
- curved faces

with no specific enabling.

12.2 Specific errors in applying tools

It is about messages, directly displayed during the activation of the application tools. The message indicates that the tool has not worked.

50 - Tool did not interpret transforms

Explanation:
No changes have been produced by the tool activated according to the parameters set.

Context:
This error message may result from the request for any of the following:

- general or profile tools;
- tool of emptying of area: no closed areas have been identified.

51 - This tool may be applied only to a simple profile

Explanation:
A profile tool has been activated for a complex profile

Context:
This error message may result from the request for any of the following tools:

- Linearize z (profile depth)
- Apply bridges to profile
- Disconnect profile
- Close open profile
- Generate path for tool radius compensation
- Generate spline curves

53 - Minimize the profile: reduction angle exceeds the value of 90.0°

Explanation:
A value higher than 90° was assigned to the reduction angle parameter.

Context:
The event can be signalled after calling the tool:

- Minimize profile.

54 - Fragment profile: maximum length of traits is null

Explanation:
The maximum fragmentation length parameter is set to an invalid value (< 5.0 * epsilon)

Context:
This error message may result from the request for the following profile tool: Fragment profile.

55 - Apply bridges to profile: invalid number of bridges [minimum 2; maximum 255]

Explanation:
The number of assigned bridges is outside the range of values from 2 to 255.

Context:
This error message may result from the request for the profile tool: Apply bridges to profile, in automatic distribution mode or programmed tools (codes STOOL) corresponding.

56 - Apply bridges to profile: invalid length of bridges or compensation exceeding tool diameter

Explanation:
The length of bridges is set to null value (< epsilon); or with tool compensation active flag, the length of bridges is set to a value lower than the tool diameter.

Context:
This error message may result from the request for the following tool: Apply bridges to profile.

59 - Apply bridges to profile: invalid or unassigned thickness of bridges

Explanation:
The residual thickness of bridges is set to null value (< epsilon).

Context:
This error message may result from the request for the tool: Apply bridges to profile.

60 - Apply bridges to profile: cannot distribute bridges on profile (reduce the number of bridges)

Explanation:
The profile length is not enough to distribute all bridges required. To solve this problem, it is necessary to set a lower number of bridges.

Context:
This error message may result from the request for the tool *Apply bridges to profile* with request for automatic distribution of bridges.

61 - Profile inversion: complex codes encountered that cannot be inverted

Explanation:
The current profile that is wished to be inverted is assigned with complex codes (subroutines and/or macros) which:

- cannot be assimilated to a profile and do not handle the inversion parameter; or
- which cannot be inverted since they must respect restrictions set during the assignment of the working database for the application (they themselves apply workings for which the possibility to invert the execution has been excluded).

Context:
This error message may result from the request for the following profile tool: *Invert profiles*.

62 - Apply tool: complex code of profile end does not terminate with a profile trait

Explanation:
The current profile to be inverted terminates with a complex code (subroutines and/or macros) whose development does not end with a segment of profile.

Context:

This error message may result from the request for the following profile tool: *Invert profiles*, *Apply entry to profile* (selecting a joined segment as a coverage in exit).

63 - Displace setup in profile: the position coincides with the current setup

Explanation:
The position where the setup point should be moved corresponds to a point on profile which coincides with the current setup position (the difference between values is significant if greater than epsilon).

Context:
This error message may result from the request for the following profile tool: Displace setup in closed profile.

64 - The tool can be applied to a closed profile

Description:
The current profile is not closed. The starting point must coincide with the end point (the difference between values is significant if greater than epsilon).

Context:
This error message may result from the request for the following profile tool: Displace setup in closed profile, Apply entry to profile (selecting a joined segment as a coverage in exit).

67 - Fillet or chamfer profile: radius assigned is null

Explanation:
The radius assigned to fillet or chamfering has null value (< epsilon).

Context:
This error message may result from the request for the following profile tools: Apply fillets to profile, Apply chamfering to profile, Generate path for tool radius compensation or corresponding programmed tools (codes STOOL).

68 - Cut profile: shown position already coincides with setup

Explanation:
The position where the profile should be cut corresponds to a point on profile which coincides with the current setup position (the difference between values is significant if greater than epsilon).

Context:
This error message may result from the request for the following profile tool: Cut profile.

69 - Cut profile: shown position already terminates a profile

Explanation:
The position where the profile should be cut corresponds to a point on profile which coincides with the profile end position (the difference between values is significant if greater than epsilon).

Context:
This error message may result from the request for the following profile tool: Cut profile.

70 - Enter / Exit profile: reference working is unassigned

Explanation:
The tool cannot be activated because, among the workings which have been configured for the application, a working essential for the tool to function is not available. It is always about a basic work on profile for a type of segment. The necessary basic codes are the following:

- COPL01 for linear segment;
- COPA17 for circular segment at the beginning of profile;
- COPA16 for circular segment at the end of profile.

Context:
This error message may result from the request for the following profile tools: Apply entry to profile, Apply exit to profile.

71 - Apply tool: can't link an entry before profile

Explanation:
It is not possible to assign a profile opening or, in any case, a point hook before the current profile since:
- the profile starts with a complex code (subroutine or macro) which, in any case, cannot be assimilated to a profile, at the beginning of its own development or
- the profile starts with a complex code (subroutine or macro) which does not handle the hook parameter, or which cannot be hooked because it must respect restrictions set during the assignment of the working database for the application.

Context:
This error message may result from the request for the following profile tools: Apply setup, Apply multiple setups, Apply entry to profile.

72 - Enter profile: displacement unassigned for initial point of profile

Explanation:
The position where the setup point should be moved coincides with the current setup position (the difference between values is significant if greater than epsilon).

Context:
This error message may result from the request for the following profile tool: Apply entry to profile.

73 - Exit profile: displacement unassigned for profile final point

Explanation:
The position where the end point of profile should be moved coincides with the current end point of profile (the difference between values is significant if greater than epsilon).

Context:
This error message may result from the request for the following profile tool: Apply exit to profile.

75 - Join profiles: second profile not properly identified

Explanation:
No profile with geometric continuity with respect to the first selected profile has been identified.

Context:
This error message may result from the request for the following profile: *Connection between consecutive profiles*.

78 - Join profiles: profiles are separated

Explanation:
The selected profiles have no geometric continuity such as to allow to join separate profiles into a single profile (the difference between values is significant if greater than epsilon).

Context:
This error message may result from the request for the following profile tool: Join profiles.

79 - Stretch profile: non-modifiable complex codes were encountered

Explanation:
The current profile is assigned with complex codes (subroutines and/or macros) which cannot be modified by the selected tool:
- they cannot be assimilated to a profile and do not handle the scale parameters; or
- they cannot be reduced or amplified since they must respect restrictions set during the assignment of the working database for the application (they themselves apply workings for which the possibility to scale the execution has been excluded)
- they execute arcs in planes different from xy and a scale limited to the only xy plane is required.

Context:
This error message may result from the request for the following profile tool: Scale the profile.

80 - Stretch profile: amplification or reduction factor unassigned or equal to 1.0

Explanation:
The assigned scale factor is equal to 1.0 or is unassigned.

Context:
This error message may result from the request for the following profile tool: Scale the profile.

82 - The tool required too many repetitions

Explanation:
An excessive number of repetitions has been required: for example, the total number of repetitions to enter cannot be greater than 1000.

Context:
This error message may result from the request for the following tools:
- Repetition of workings, Rectangular series of workings, Circular series of workings with several repetitions greater than 100000.

- Repetitions along a profile with a number of repetitions greater than 100000 or if the distance calculated among following repetitions is too little (< 10.0 epsilon quote).

and of programmed tools (codes STOOL):
- Apply Z feed with excessive number of passes (maximum: 1000).

85 - Apply tool: the profile assigns arcs in a plane different from xy

Explanation:
The activated tool cannot operate on one or more selected profiles because the profiles themselves have circular elements (arcs) laying on a plane different from xy.

Context:
This error message may result from the request for any of the following tools:
- Scale the profile, with request for a scale limited to the only xy plane
- Linearize z (profile depth)
- Generate path for tool radius compensation
- Generate spline curves

86 - Profile exit: can’t hook on a downward exit

Explanation:
It is not possible to assign a hook point after the profile itself since the profile terminates with a complex code (subroutine or macro) which:
- cannot be assimilated to a profile, at the end of its own development or
- which cannot handle the hook parameter or cannot be hooked since it must respect restrictions set during the assignment of the working database for the application;

Context:
This error message may result from the request for the following profile tool: Apply exit to profile.

88 - Apply tool: unable to apply setup, reference code missing

Explanation:
The activated tool has not worked because of the impossibility to assign a reference setup code to profile.

Context:
This error message may result from the request for profile tools. The assignment of a reference setup code is performed during the Configuration of TpaCAD.

92 - The tool didn't introduce displacements on any axis

Explanation:
the parameters do not give effect to any translation.

Context:
the message can be a result of the demand for Translation tool.

93 - The tool introduced a null rotation

Explanation:
The parameters do not give effect to any rotation.
No rotation is generated by the parameter setting. It could have been assigned a relative rotation angle of null value or an absolute rotation angle and a relative centre with zero value.

Context:
This error message may result from the request for the Rotation tool.

94 - The tool didn't introduce repeated applications

Explanation:
The total number of repetitions to enter is not enough.

Context:
This error message may result from the request for the following tools:
Repetition of workings with number of required repetitions lower than 1
Rectangular series of workings with total number of repetitions (rows*columns) lower than 1
Circular series of workings with number of elements to execute lower than 2

The resulting error can be a consequence of the use of programmed tools (codes STOOL): Apply Z feed with lower step forward to (epsilon*10) or number of passes zero.

95 - Develop text: the text was truncated at the maximum size allowed for the development of the curved geometry

Explanation:
Case of distribution of a text along a geometric curved element (arc of circle or of conic section), with total length of the original string exceeding the length of the whole curve, circle or conic section.
The text has been developed up to the maximum development allowed for the geometric element.
The report is managed as a Warning.

Context:
The report can be the result of the request of the advanced tool Generation of texts or of the programming for text development working.

96 - Develop text: the conic section of the development is not valid

Explanation:
The arc of the conic section (oval or ellipse) on which to distribute the text is assigned in a wrong way.

Context:
The report can be the result of the request of advanced tool Text generation.

98 - Create text: insufficient font height (min = eps * 100)

Explanation:
Too small font size has been assigned. The value cannot be lower than (epsilon*100).

Context:
This error message may result from the request for the advanced Text Generation tool.

99 - Develop text: invalid arc of development

Explanation:
The arc on which the text is distributed is incorrectly assigned: the radius is null, or the initial radius is different from the final radius.

Context:
This error message may result from the request for the advanced Text Generation tool.

294 - Emptying of areas: profile is not closed

Explanation:
The emptying of open area/s was requested.

Context:
This error message may result from the request for the advanced Emptying of areas tool.

295 - Emptying of areas: profile unsuitable for the assigned tool

Explanation:
It was requested emptying an inconsistent closed area (empty area) or such that not even a first partial emptying pass, with the assigned technology, is allowed.

Context:
This error message may result from the request for the advanced Emptying of areas tool.

296 - Emptying of areas: tool radius assigned is null [min: 10*epsilon]

Explanation:
During the emptying of an area, an invalid radius compensation value (< 10.0*epsilon) has been assigned.

Context:
This error message may result from the following request:
- advanced Emptying of areas tool;
- applying an emptying cycle (application of subroutine or macro).

297 - Emptying of areas: coverage exceed the tool radius

Explanation:
The assigned coverage value exceeds the tool radius compensation.

Context:
This error message may result from the following request:
- advanced Emptying of areas tool
- applying a surface emptying cycle (application of subroutine or macro).

298 - Emptying of areas: depth range includes Z=0.0

Explanation:
The initial and final depth values are assigned with opposite sign in surface emptying with execution of subsequent passes.

Context:
This error message may result from the following request:
- advanced Emptying of areas tool
- applying a surface emptying cycle (application of subroutine or macro).

299 - Emptying of areas: invalid Z air coordinate

Explanation:
The values for the initial depth and the coordinate over the workpiece are assigned with the same sign.

Context:
This error message may result from the request for any of the following:
- advanced Emptying of areas tool
- applying a surface emptying cycle (application of subroutine or macro).

300 - Emptying of areas: excessive profile number to evaluate (max: 300)

Explanation:
It has been requested executing a surface emptying, which has generated the evaluation of an excessive number of closed areas (maximum allowed value = 300).

Context:
This error message may result from the request for any of the following:
- advanced Emptying of areas tool
- applying a surface emptying cycle (application of subroutine or macro).

12.3 Errors In Parametric Programming

These are signals in correspondence to incorrect settings of parameter setting.
In case of an error of this type, the relative expression is resolved by assigning:
- 0.0, in the case of variable or parameter number;
- string resolved coincident with the string programmed, in the case of variable or parameter does not numeric.

For a discussion of situations in which the various errors are reported, refer to the chapter about the Parametric programming.

101 - Parametric programming: string too long

Explanation:
An expression consisting of a too high number of characters has been typed. The maximum allowed number of characters is 100.

102 - Parametric programming: invalid syntax

Explanation:
The syntax used in parametric programming is invalid.
Here are a few programming rules which can be useful to interpret a syntax error:
- characters whose value is enclosed between the space (' ') and the closed curly bracket ('}') are valid;
- the space can only be used in assignment of a string (string-type variable or parameter or string-type argument). Examples:
  - "**strcmp[r5;"pippo 1"]**" is valid
  - "**120+ 12**" causes syntax errors
- the " character is interpreted to assign direct messages (it heads and closes a message). Example: strcmp[r5;"pippo"];
- the ' character is interpreted to assign a character value (heading and closure). Example: 120+'a';
- the use of the relative syntax must assign symbolic names with:
  - a non-empty symbolic name: "o\" causes error;
  - name length must not exceed 16 characters: "o\abracadabraaaaaaaaaaaa" causes error;
  - a variable of the type indicated by the specified name must be assigned: "o\aaa" causes syntax error if the "o" variable is not assigned with the "aaa" symbolic name.
- the use of variable "o" functions relative to the development of custom functions causes syntax error, if they are used in a program

Example:
Examples of invalid syntax are the following:
"100+16-" -> becomes valid if changed to "100+16"
"32*(r0+r3" -> becomes valid if changed to "32*(r0+r3)"
"abs(r5)"-> becomes valid if changed to "abs[r5]"
"o\aaa" -> if the "o" variable is not assigned with the "aaa" symbolic name

103 - Parametric programming: <r> variable recalled by name not found

Explanation:
You are using an <r> variable with a symbolic name that has not been assigned.
According to the configuration of TpaCAD, this can be a serious or a non-serious report (a warning): in case of warning, a null numerical value (0.0) is assigned to the use of the variable.

105 - Parametric programming: value exceeds range allowed (-3.4E+30; 3.4E+30)

Explanation:
The value calculated for the numeric expression exceeds the range allowed for a value with decimal point.

106 - Parametric programming: solution of string parameter too long (max = 260 chars)

Explanation:
The string which results from a parametric programming exceeds the maximum allowed number of characters for a string-type parameter which is 260.

109 - Parametric programming: invalid context for subroutine arguments

Explanation:
Error relative to the use of variable arguments concerning the application of subroutine or macro: subx, suby, …, subface.

Context:
Invalid contexts of use are the following:
- assignment of 'o', 'v' variables;
- assignment of custom functions;
- assignment of variable geometries (fictive face edges).

111 - Parametric programming: invalid context for use of variables <$>

Explanation:
Error relative to the use of <$> variables or functions with <$> variables: $0-$299, p$[.], min$[.], max$[.], ave$[.], sum$[.].

Context:
Invalid contexts of use are the following:
- assignment of 'r', 'o', 'v' variables;
- assignment of custom functions;
- assignment of variable geometries (fictive face edges);
- program text.

112 - Parametric programming: invalid context for use of variables <r>

Explanation:
Error relative to the use of <r> variables or functions with <r> variables: r0-r299, pr[.], minr[.], maxr[.], aver[.], sumr[.], strlen, getat[.],strcmp[.], toolex[.]. tooltip[.].

Context:
Invalid contexts of use are the following:
- assignment of 'o', 'v' variables;
- assignment of custom functions.

113 - Parametric programming: invalid context for use of variables <v>

Context:
Invalid contexts of use are the following:
- assignment of 'o', 'v' variables;
- assignment of custom functions.

114 - Parametric programming: invalid context for use of variables <o>

Context:
Invalid contexts of use are the following:
- assignment of 'o', 'v' variables;
- assignment of custom functions.

115 - Parametric programming: invalid context for use of variables <j>

Explanation:
Error relative to the use of < j > variables or functions with <j> variables: j0-j99, pj[.], minj[.], maxj[.], avej[.], sumj[.].

Context:
Invalid contexts of use are the following:
- assignment of 'o', 'v' variables;
- assignment of custom functions;
- assignment of variable geometries (fictive face edges).

116 - Parametric programming: invalid context for use of working name

Explanation:
error concerning the use of a geometric library function whose syntax uses the name of a working.

Context:
Invalid contexts of use are the following:
- assignment of 'o', 'v' variables;
- assignment of custom functions;
- assignment of variable geometries (fictive face edges).

117 - Parametric programming: invalid index to a variable <r>

Explanation:
An invalid index to an <r> variable is indicated or calculated: valid values are included between 0 and 299. However, in <r> variable programming, the range of values is more limited: an <r> variable can only use lower index variables; thus, for example: r10 can use r9, but not r11.

Examples:
Examples of wrong programming are the following:
"r400", "pr[400]", "*pr[400]" the maximum index to an "r" variable is 299;
"r20" used for example in r10 variable assignment.

118 - Parametric programming: invalid index to a variable <j>

Explanation:
An invalid index to a <j> variable is indicated or calculated: valid values are included between 0 and 99.

119 - Parametric programming: invalid index to a variable <$>

Explanation:
An invalid index to a <j> variable is indicated or calculated: valid values are included between 0 and 99.

120 - Parametric programming: invalid index to a variable <v>

Explanation:
An invalid index to a <v> variable is indicated or calculated: valid values are included between 0 and 15, with the maximum allowed value which depends on the number of <v> variables which the software is enabled to handle:
- the maximum value is equal to 15 with 16 handled variables;
- the maximum value is equal to 6 with 7 handled variables;
- …
- no value is valid, with any <v> variable handled.

121 - Parametric programming: invalid index to a variable <o>

Explanation:
An invalid index to an <o> variable is indicated or calculated: valid values are included between 0 and 15, with the maximum allowed value which depends on the number of <o> variables which the software is enabled to handle:
- the maximum value is equal to 15 with 16 handled variables;
- the maximum value is equal to 6 with 7 handled variables;
- …
- no value is valid, with no handled <o> variable.

122 - Parametric programming: function with too many operands (max 30)

Explanation:
A function has been called with several operands greater than 30, which is the maximum allowed limit for the number of operands of a function.

123 - Parametric programming: function without operands

Explanation:
A multi-operand function without operands has been called. An example of wrong programming is the following: "ifcase[]".

124 - Parametric programming: function with wrong operands number

Explanation:
A multi-operand function with a wrong number of operands has been called. An example of wrong programming is the following: "ifelse[r0;l/2].

125 - Parametric programming: division by zero

Explanation:
In a mathematical operation a division by zero has been executed. The error results from the use of division mathematical operators(/, %, #, ?).

126 - Parametric programming: value of trigonometric function (sin, cos) beyond range -1 +1

Explanation:
If the operand is not included between –1.0 and +1.0, an inverted trigonometric function (asin, acos) has been executed.

127 - Parametric programming: square root of negative value

Explanation:
A *sqrt* function which returns the square root of a number has been executed with negative operand.

128 - Parametric programming: exponentation with invalid exponent [min: 0; max: 10]

Explanation:
The *pown* involution function with the 2nd operand (exponent) not included between 0 and 10 has been executed.

129 - Parametric programming: invalid geometrical library function

Explanation:
The multi-purpose geometry library function *geo[..]* has been executed with the 1st operand which does not match a valid name. An example of wrong programming is "geo[aaa;l/2]": "aaa" does not match a valid name.

130 - Parametric programming: function missing required argument

Explanation:
A compulsory function argument is missing. Example of wrong programming is "primp[;100.0]": the 1st argument, which is considered compulsory, is missing.

132 - Parametric programming: invalid angle for tangent calculation

Explanation:
The error is due to the execution of the trigonometric function *tan*, when an invalid value for tangent calculation is assigned to the operand (angle). The assigned angle, reduced to the numeric range (0° - 360°) cannot be 90° or 270°, because in these cases calculating the tangent loses its meaning.

134 - Parametric programming: too many nested calls of custom function (max: 5)

Explanation:
The number of the nested custom calls is greater than 5. This error may only result from a wrong assignment of custom functions, as performed during the software configuration.

135 - Parametric programming: invalid use of custom function

Explanation:
This error message signals that an already stacked custom function is programmed or an unauthorized use of private custom function.

136 - Parametric programming: invalid use of arguments arg# res#

Explanation:
Error relative to the use of arguments reserved for program custom functions.

137 - Parametric programming: arg# argument invalid index or name

Explanation:
An invalid index to an <arg> variable is indicated or calculated, or no variable is assigned with the specified symbolic name. The error message can appear only in writing custom functions.

138 - Parametric programming: res# argument invalid index or name

Explanation:
An invalid index to a <res> variable has been indicated or calculated, or no variable is assigned with the specified symbolic name. This error message can only appear in writing custom functions.

139 - Parametric programming: error calling custom function

Explanation:
An error has been detected at custom function level. For more information refer to the specific documentation on the custom functions assigned during the software configuration.

140 - Parametric programming: error while using functions reserved for custom functions

Explanation:
Functions reserved for program custom functions have been used.

141 - Parametric programming: invalid index to var# argument

Explanation:
An invalid index to a <var> variable is indicated or calculated, or no variable is assigned with the specified symbolic name. This error message can only appear in writing custom functions.

12.4 Errors in processing variable geometries

It is about error messages concerning the assignment of *variable Geometries*.

22 - It's impossible the deletion of a face which has workings assigned

Explanation:
A fictive face, on which workings have been programmed, cannot be cleared.

Actions:
Eliminate all the programmed workings on the fictive face, then clear the fictive face itself.

144 - Variable geometries: invalid or not assigned reference face

Explanation:
An invalid number has been assigned to the reference face. It is always about a fictive face number (greater than 6) and it may specify that the selected face:
- is unassigned;
- has an identification number greater than or equal to the current fictive face;
- has invalid geometry (no distinct or aligned points).

145 - Variable geometries: not all the face vertices are distinguished

Explanation:
The fictive face or the automatic face which is being defined does not have all three vertices distinguished.

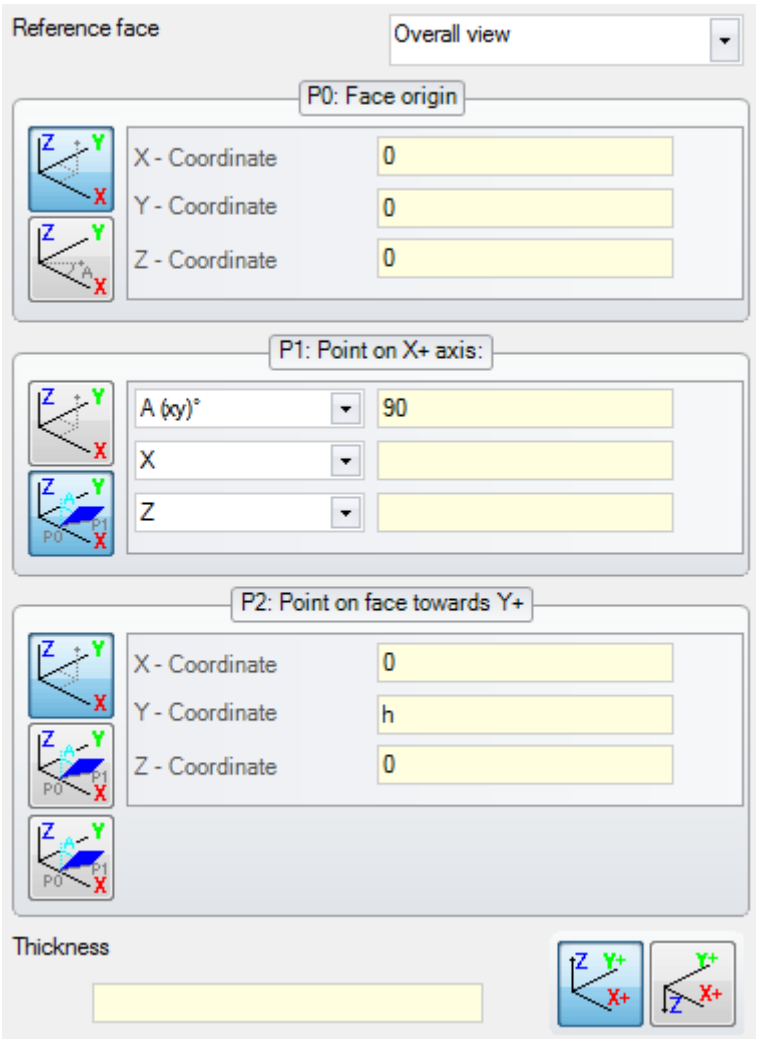
146 - Variable geometries: face vertices are aligned

Explanation:
The three vertices of the fictive or the automatic face which is being defined are assigned aligned.

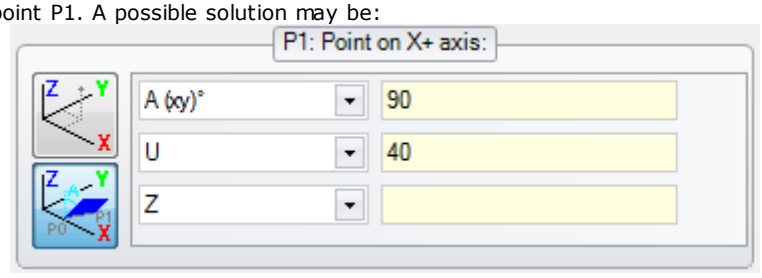
147 - Variable geometries: invalid face polar geometry

Explanation:
A geometric error has occurred during the assignment of a face edge in polar coordinates. In particular, the error message appears in case of computation of infinite polar coordinates vector.

Example:
An example of wrong programming is shown in the Figure below:



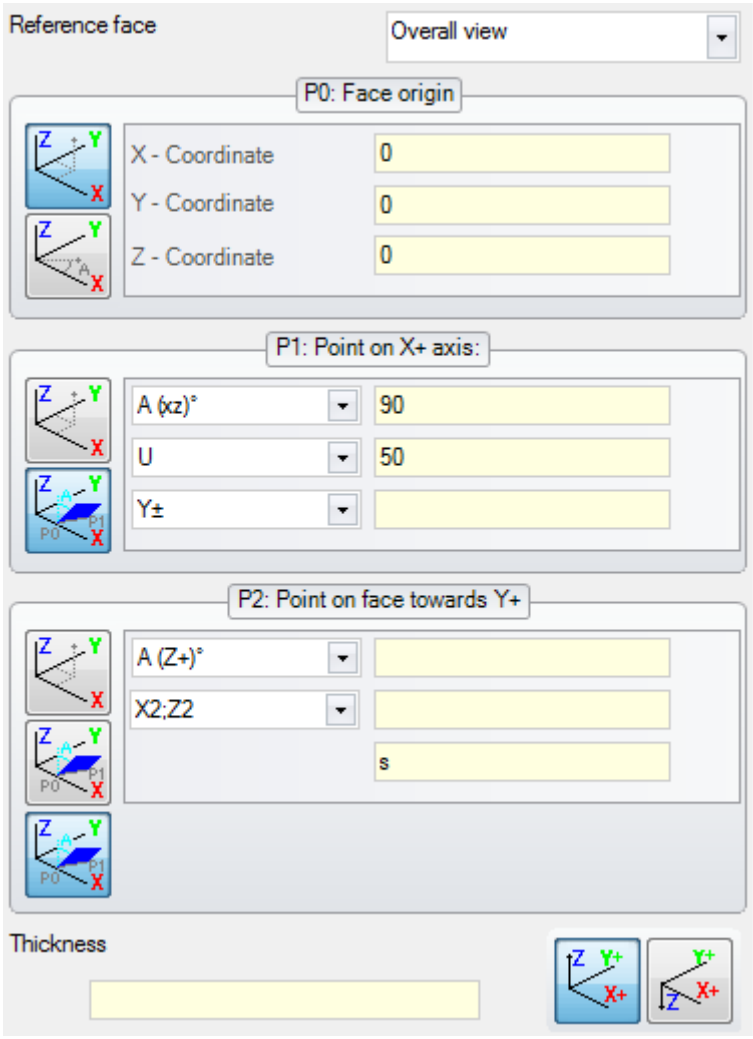
The error concerns point P1. A possible solution may be:



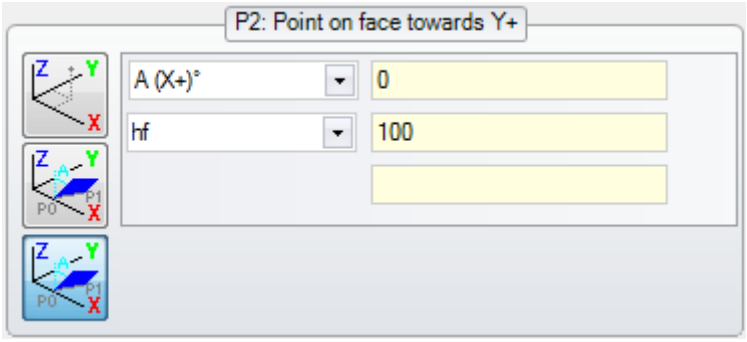
148 - Variable geometries: invalid rotation plane

Explanation:
An error has occurred in the assignment of the face rotation plane (no distinguished or aligned points).

Example:
An example of wrong programming is shown in the Figure below:



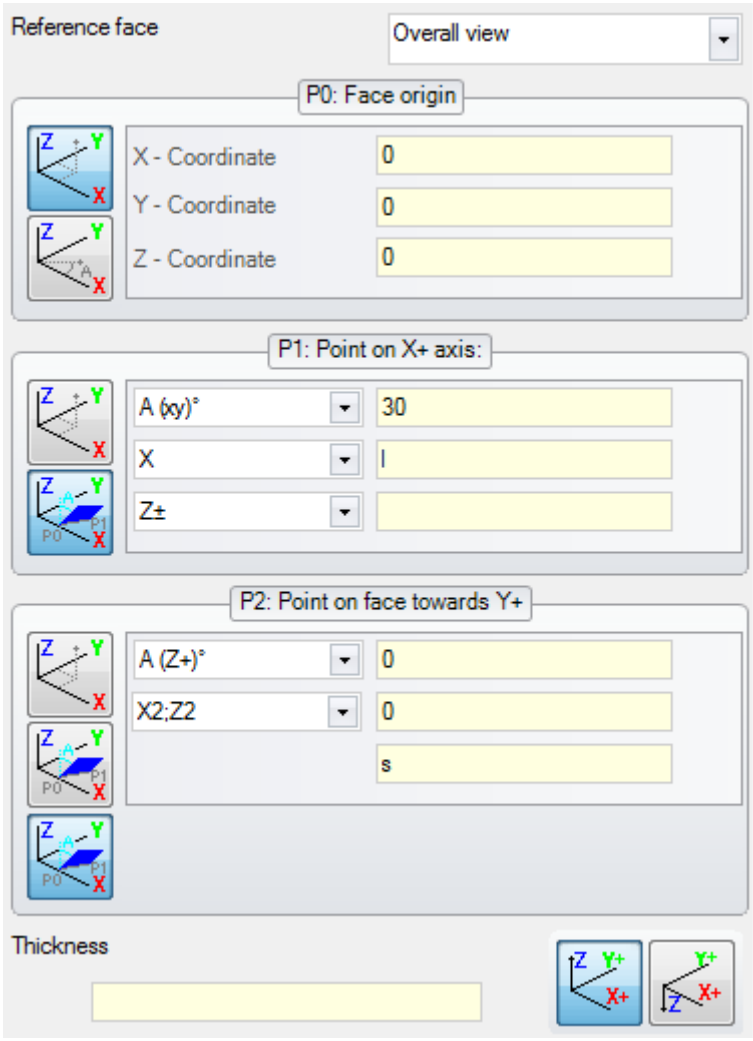
The error concerns point P2. A possible solution may be



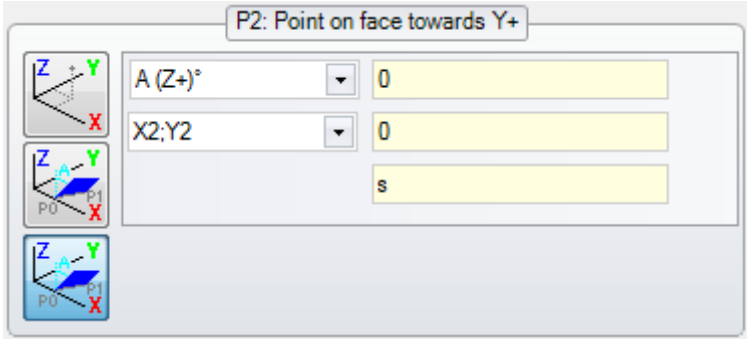
149 - Variable geometries: impossible to assign the face third point

Explanation:
An error has occurred in the assignment of the third corner by face rotation plane.

Example:
A case of incorrect programming corresponds to the assignments as shown below:



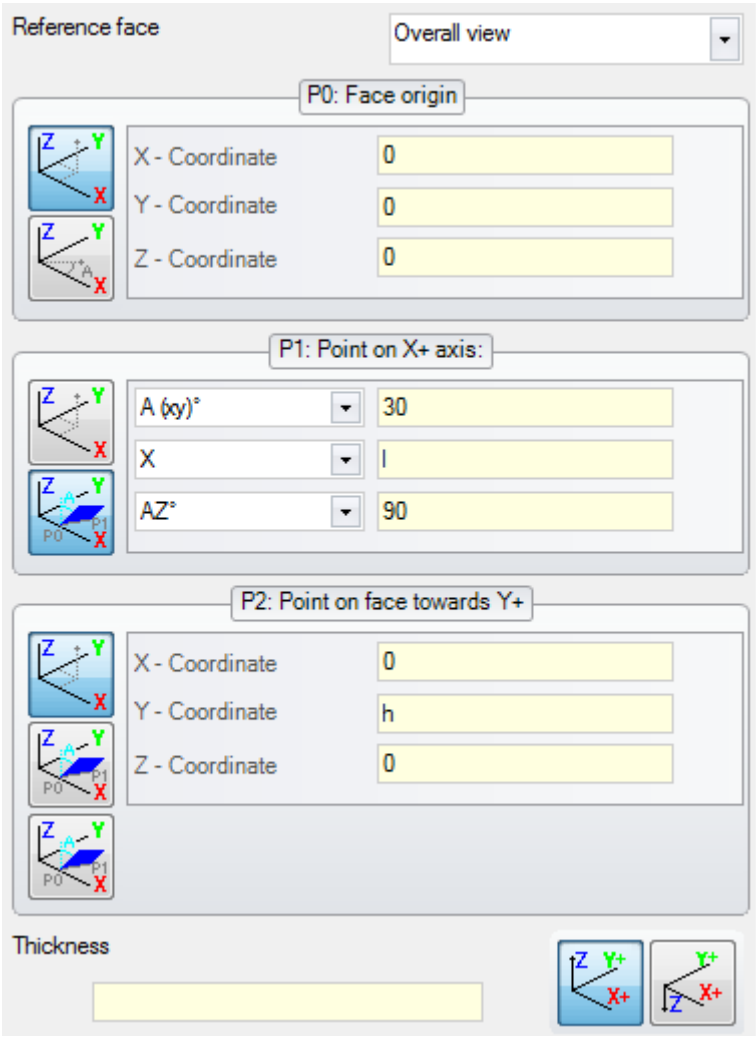
The error concerns point P2. A possible solution may be:



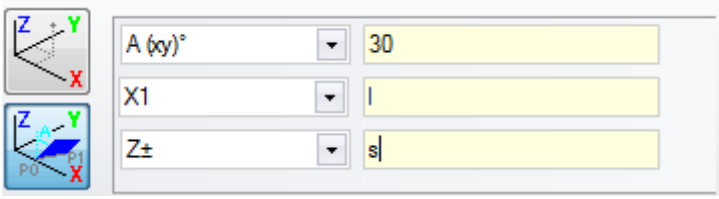
150 - Variable geometries: invalid depth point

Explanation:
An error has occurred in the assignment of the point third coordinate axis in polar coordinates. In particular: this error may be generated in case of coordinate assignment with angular increment mode, if the angle has absolute value equal to 90°.

Example:
An example of wrong programming is shown in the Figure below:



The error concerns point P1. A possible solution may be:



165 - Variable geometries: invalid face curvature radius

Explanation:
A non-null radius of curvature of the face has been assigned (the radius is taken null, if <= epsilon*2.0), but less than the distance between P0 and P1.

166 - Variable geometries: geometric solution error of the surface

Explanation:
It is impossible to solve a geometric continuity between fictive faces.

167 - Variable geometries: max. number of elements in the surface

Explanation:
It is not possible to insert any further element in the surface: the maximum managed number (300) has been reached.

12.5 Errors compiling face program

These are errors that occur during compilation of the program face. Only in some it is a fatal error, that does not make it possible to run the program.
A solution of default is always used (please refer to the documentation of processes, for a detailed examination of each situation).

151 - The <operative code name> working code is invalid

Explanation:
The specified operative code (in place of <operative code name>) is not assigned in the working database implemented for the application or it is assigned with a different typology.

Actions:
- The corresponding working is solved by propagation of work coordinates from the previous working;
- it is about a **critical error**, which makes the program execution impossible:
- to solve this error, it is necessary to replace the working with a valid working (for example, by replacing the operative code).

Context:
This error may appear while reading a program from a file.

152 - <parameter name> parameter: invalid value

Explanation:
A value outside the range defined in configuring the application has been assigned to the selected parameter (in place of <parameter name>)

Actions:
- it is about a **critical error**, which makes the program execution impossible.

153 - <parameter name> parameter: set format $nn

Explanation:
In a FOR instruction the first term of one of the three expressions has not been correctly set.

Context:
This error may only appear in programming a macro-program.

Example:
An example of correct assignment is the following:

FOR (***$0*** = r1 to ***$0*** <= r2; ***$0*** = $0+r3)
{
.
} ENDFOR
where the first term of each expression is in bold. For each term the $0 form is used (it is not necessary that the same variable is always indicated), as required.

Here is an example of wrong assignment:

FOR (***$0*** = r1 to ***$0+5*** <= r2; ***$0*** = $0+r3)
{
.
} ENDFOR
where the wrong assignment is underlined.

155 - <field name> property: invalid value

Explanation:
the indicated property is assigned a value outside the range defined in the configuration. The <field name> may be replaced with:
"L" level;
"B" to construct;
"O", "M", "K", "K1", "K2" for the corresponding field.

Actions:
if the value has been assigned negative, the property is imposed value 0. If the value is greater than the maximum value set in the configuration, the property is imposed the maximum value.

156 - <field name> field: the value doesn't comply with the minimum set

Explanation:
The signalling refers to a field in custom section: it is shown in group of generic programming errors due to the similarities of the signalling. At the indicated field is assigned a value that is less than the minimum value defined in configuration. However, this is not a serious error (it is a warning).

Actions:
However, the value that has been assigned, is maintained. Given the peculiarities of the custom sections, you delegate any decision on the fields set to a subsequent interpretation (read: being optimized).

157 - <field name> field: the value exceeds the maximum set

Explanation:
the report refers to a field in custom section, just like the above error. At the indicated field is assigned a value that is greater than the maximum value defined in configuration. However, this is not a serious error (it is a warning).

Actions:
however, the value that has been assigned, is maintained. Given the peculiarities of the custom sections, you delegate any decision on the fields set to a subsequent interpretation (read: being optimized).

158 - Modelling: invalid code or sequence code

Explanation:
The signal regards a line in modelling section, and it shows an invalid code or an incorrect sequence codes.

Action:
However, the value that has been assigned, is maintained. It is needed the deletion or modification of the section, in order to detect the error message.

161 - Too many or not available automatic faces

Explanation:
It is not possible to assign a reference to an automatic face for an excessive number of created faces (maximum allowed number: 400) or because there are no assigned faces at all.

Actions:
- The corresponding working is executed without assignment of automatic face;
- it is about a **critical error**, which makes the program execution impossible;
- to solve this error, it is necessary to modify programming instructions (by deleting the relevant working or reducing the total number of assigned automatic faces or using a non-automatic reference face).

Context:
The error message may appear while processing a working code which assigns an automatic face.

162 - Field F: invalid value

Explanation:
An invalid value has been set for the (F field) working property Application face.

Actions:
- it is about a **critical error**, which makes the program execution impossible;
- to solve this error, it is necessary to modify programming instructions.

Context:
This error message may only appear in programming the piece-face and may be caused by one of the following cases of assignment:
- unauthorized assignment of automatic face;
- failed assignment of automatic face;
- failed assignment of non-automatic real or fictive face.

190 - Working exceeding the application limits (axis <axis name>)

Explanation:
the working development exceeds an area or an application length. The message shows the axis or the axes for which the report has been activated.

To do:
- If the alert corresponds to an **error**, the execution of the program is not made possible.

Context:
The alert may occur while applying the SUBNEST working, if the development of the workings is outside the programmed dimension area.

12.6 Errors in works on profile

192 - Tool radius computed as infinite

Explanation:
An infinite value has been computed for the vector radius of a polar system.

Context:
Here are the concerned workings:
- L04 [code = 2204];
- L05 [code = 2205];
- L06 [code = 2206];
- L07 [code = 2207].

193 - Tool radius null

Explanation:
A null value has been computed for the polar radius coordinate or the arc radius. This message should be viewed as a **warning** rather than as an error.

194 - Invalid arc

Explanation:
An arc has been incorrectly or insufficiently assigned (unassigned centre, the initial radius is different from the final radius).

195 - Invalid intersection line

Explanation:
An intercept-line is not correctly assigned (unassigned or no distinct points, or geometrically invalid). An intercept-line must be assigned with:
- two distinct points, or
- a point and an angle.

196 - Invalid entry tangent

Explanation:
An entry Tangent line has not been correctly defined (unassigned or geometrically invalid). An intercept-line must be assigned with:
- an angle, or
- two distinct points.

This message should be viewed as a **warning** rather than as an error.

197 - Invalid exit tangent

Explanation:
An exit Tangent line has not been correctly defined (unassigned or geometrically invalid). Remember that an intercept-line must be assigned with:
- an angle, or
- two distinct points.

This message should be viewed as a **warning** rather than as an error.

198 - Point computed external to traits

Explanation:
In a chamfering or in a fillet, the point computed is external to the original programmed segment.

199 - Intersection non-existent

Explanation:
Error message which may appear in case of double arcs, if no solution is found.

200 - Invalid arc (points are not distinguished)

Explanation:
An arc is incorrectly assigned, because of the coincidence between arc points and/or points with the centre.
Error conditions:
- arc assigned to three points: the three points are not distinct;
- arc assigned to two points and the centre: the centre coincides with a point on the arc.

201 - Invalid arc (points aligned)

Explanation:
In arc assignment by points, these last have been assigned aligned. If the arc takes place in the space, the error reports also the cases of circle or of aligned initial and end points and centre.

202 - Oval: invalid radius

Explanation:
In constructing an oval profile, the minor radius is assigned greater than or equal to the minor half-axis.
This message should be viewed as a **warning** rather than as an error.

203 - Oval reduced to a circle

Explanation:
In constructing an oval profile, the two half-axes are defined equal. This message should be viewed as a **warning** rather than as an error.

204 - Oval: null or invalid axis / axes

Explanation:
In constructing an oval profile, one or both semi-axes are null (the difference between values is significant if greater than epsilon).

205 - Ellipse/Oval: start point exterior conic extents

Explanation:
In constructing an oval or an ellipse the starting point falls outside the assigned conic extents.

206 - Rectangle: invalid axis/axes or radius

Explanation:
In constructing a rectangle one or both axes are defined null (the difference between values is significant if greater than epsilon) or the set fillet radius is such that it exceeds the rectangle extents.

207 - Polygon: invalide number of sides

Explanation:
In the definition of the polygon working an invalid sides number has been assigned: a value between 3 and 99 is accepted. It is not a serious warning: an invalid value is brought back to the shown range.

12.7 Errors in subroutine or macro

208 - Programmed tool: no correspondence found

Explanation:
case of corresponding STOOL type code (programmed tool), indicating that no valid working has been found. The warning is managed as an **alert** and not as an error.
Error situations are:
- nothing has been assigned in the programmed tool **Name** field
- the name/s indicated in the **Name** field have no match in the workings programmed earlier
- no working is valid for the applied tool.

209 - Application Invalid encrypted program

Explanation:
the subroutine (or macro) that is assigned does not meet the criteria set for encryption. Error conditions:
- the file does not match a macro program;
- the application code is SUB generic;
- the file does not match the signature database of custom work;
- the file has attributes that do not match the setting in the database of workings.

Context:
The message can display one of the following situations:
- encryption subroutine occurred after the use of the file itself;
- the current structure of the application has not released the signature corresponding to the database of custom work;
- the encrypted file for the program has been tampered with manually.

210 - Invalid subroutine name

Explanation:
The subroutine (or macro) name is incorrectly assigned. Error conditions:
- it is assigned with invalid characters: "#%;/\;"
- it is assigned with more than one character "."

211 - Subroutine doesn't exist

Explanation:
the subroutine (or macro) does not exist or cannot be read.

212 - Shown file hasn't a valid format for subroutine

Explanation:
The specified subroutine (or macro) has an invalid format. This error message may also appear as a result of an attempt to apply a macro with a generic subroutine code without being enabled to do it.

213 - Face number not valid

Explanation:
It has been required applying an invalid identification number (number lower than 1 or greater than 99) to a face.

Context:
The message can display one of the following situations:
- A face number lower than 1 or higher than 99 has been assigned;
- in the program of face- piece:
  - a subroutine induced call is applied on an automatic face;
  - a SSIDE working (programmed induced call working) has defined an invalid face of the subroutine to apply;
  - a SSIDE working (programmed induced call working) has defined an invalid application face.

214 - Element of technological reference not applied

Explanation:
Case of complex code with assignment parameter of the technological working by name, with not managed working. The report can indicate that any setup or point working, named in that way, has not been found or that the working is not valid by compilation of operating code.
The report is managed as a Warning.

216 - Subroutine read failure

Explanation:
An error has been detected in reading the subroutine (or macro).

217 - Subroutine name unassigned

Explanation:
No name has been assigned to the subroutine (or macro).

218 - Curve creation can't be applied

Explanation:
It has not been possible to generate a Spline curve, since no profiles to which to apply the transform have been recognized. If it has not been required deleting original workings, this message should be viewed as a **warning** rather than as an error.

219 - Emptying cannot be applied

Explanation:
It has not been possible to apply emptying, since no profiles to which to apply it have been recognized. If it has not been required deleting original workings, this message should be viewed as a **warning** rather than as an error.

220 - Rotation cannot be applied

Explanation:
It has not been possible to apply the required rotation for the following reasons:
- inherent limitations to the development of the subroutine (or macro): the development applies, on its turn, subroutine (or macro) to which the transform cannot be applied (in working database configuration);
- the transform cannot be applied to the application itself (in configuration);
- the subroutine (or macro) development includes circular elements (arcs) assigned in planes different from xy, but the auxiliary working A10 [code = 2110] is not configured.

221 - Inversion cannot be applied

Explanation:
It has not been possible to apply the required inversion for the following reasons:
- inherent limitations to the development of the subroutine (or macro): the development applies, on its turn, subroutine (or macro) to which the transform cannot be applied (in working database configuration);
- the transform cannot be applied to the application itself (in configuration).

222 - Mirror x cannot be applied

Explanation:
It has not been possible to apply the required mirror for the following reasons:
- inherent limitations to the development of the subroutine (or macro): the development applies, on its turn, subroutine (or macro) to which the transform cannot be applied (in working database configuration);
- the transform cannot be applied to the application itself (in configuration).

223 - Mirror y cannot be applied

Explanation:
It has not been possible to apply the required mirror for the following reasons:
- inherent limitations to the development of the subroutine (or macro): the development applies, on its turn, subroutine (or macro) to which the transform cannot be applied (in working database configuration);
- the transform cannot be applied to the application itself (in configuration).

224 - Stretch cannot be applied

Explanation:
It has not been possible to apply the required stretch for the following reasons:
- inherent limitations to the development of the subroutine (or macro): the development applies, on its turn, subroutine (or macro) to which the transform cannot be applied (in working database configuration);
- the transform cannot be applied to the application itself (in configuration);
- the subroutine (or macro) development includes circular elements (arcs) assigned in planes different from xy, but stretch is only required in xy plane.

225 - Programmed tool: one or more workings had been excluded

Explanation:
during the execution of a code programmed tool (example: STOOL) one or more workings among those listed by name have been excluded from the transformed, because not compatible with the tool itself.
This message should be viewed as a warning rather than as an error.

226 - Too many nested subroutine calls (max 5)

Explanation:
The subroutine (or macro) development has too many nested subroutine (or macro) function calls. The maximum allowed number of nested function calls is:
- 5: in case of program typology;
- 4: in case of subroutine or macro typology.

The message can also correspond to calls of programmable instruments (STOOL code) recursive. In this case the nesting procedure of the calls has a development that we can say *vertical*, along the same text of a program (subroutine or macro).

227 - Custom error number <custom error code>

Explanation:
The subroutine (or macro) development has detected a custom error, by interpreting a programmed error instruction:
- ERROR [code= 2009];
- BREAK [code= 2005], it can only be used in macro text.

The programmed error number replaces <custom error code> in the message.
This error message can also result from an ERROR instruction directly programmed in the program text. In this case, it is generated when the program is saved and means that the program cannot be executed.

228 - Impossible to assign font (invalid name)

Explanation:
The assigned font name is invalid. Error conditions:
- unassigned font name;

- the name is invalid because it consists of an excessive number of characters (maximum allowed number of chars: 32, characters);
- the name is invalid to assign a font handled by the Windows® system;
- the name is invalid to assign a TrueType font;
- the name is invalid to assign a custom font (file not found);
- the custom font does not assign a height value of the valid font (minimum: epsilon*100).

Context:
The error message may appear in case of application of text generation subroutine (or macro)

229 - Impossible to assign device for font creation

Explanation:
A system error has occurred in the assignment of the display device for font creation.

Context:
The error message may appear in case of application of text generation subroutine (or macro).

12.8 Errors in logical conditions

These errors related to the verification of the logical conditions applied to the program. They are always fatal errors, which make impossible to run the program.

230 - Number of unloaded ELSE or ENDIF exceeds the loaded IF

Explanation:
The error may indicate that it is not assigned an IF upstream, at the indicated instruction, or that ENDIF conditional statements are greater than the number of IF conditions defined. Check the correspondence among IF, ELSE and ENDIF. It is useful to remember that an open IF must not be closed by ENDIF.

Context:
The error can occur, not only in the definition of an IF statement, also in application of a subroutine or macro.

Example:
Here is an example of wrong programming:

```
IF …
  IF …
  …
  ENDIF
…
ENDIF
…
ENDIF <- Endif without the relevant If
```

231 - Number of unloaded ENDIF lower than loaded IF

Explanation:
Several IF conditional statements greater than the number of ENDIF conditions has been defined. Check correspondence between IF, ELSE and ENDIF.

Context:
The error can occur not only in the definition of an IF statement, in application of a subroutine or macro.

Example:
Here is an example of wrong programming:

```
IF … <- If without the relevant Endif
  IF …
  …
  ENDIF
…
```

232 - Invalid code after an open IF

Explanation:
The working after an open IF is invalid. An open IF may condition a working as follows:
- point working;

- custom logic working;
- complex working (subroutine or macro).

Only if in macro-program text:

- setup working;
- work on profile (circular or linear segment);
- non-custom logic working for ($, j) variable assignment.

Context:
The error can occur not only in the definition of an IF statement, in application of a subroutine or macro.

233 - Number of unloaded ENDFOR greater than loaded FOR

Explanation:
Several ENDFOR commands greater than the number of FOR conditions has been defined. Check correspondence between FOR and ENDFOR.

Context:
This error may only appear in programming a macro-program.

234 - Number of unloaded ENDFOR lower than loaded FOR

Explanation:
Several FOR conditions greater than the number of ENDFOR commands has been defined. Check correspondence between FOR and ENDFOR.

Context
This error may only appear in programming a macro-program.

235 - Number of FOR instructions exceeds the maximum admissible (max = 500)

Explanation:
More than 500 FOR cycles have been assigned. This value is the maximum number of instructions handled by a face program.

236 - Number of now running iterations of FOR cycles exceeds the maximum value (max = 100000)

Explanation:
the application of a macro has resulted in the execution of more than 100000 cycles FOR: this control is activated in order to exclude the possibility of inserting infinite loops, which block the application.

237 - An ENDIF instruction is used to close a FOR cycle

Explanation:
Check the correspondence between IF-ELSE and ENDIF, FOR and ENDFOR. Two simple general rules are recalled, which must be applied in contextual assignment of IF and FOR cycles:

- if inside an IF cycle:a FOR ..ENDFOR cycle must be completely defined within a single branch of IF… ELSE… ENDIF
- if inside a FOR cycle: an IF… ELSE… ENDIF cycle must be completely defined within the FOR cycle.

Example:
Here is an example of wrong programming:

```
FOR …
  IF …
  …
  ENDIF
…
ENDIF <- Endif that unloads FOR
```

238 - ENDFOR instruction used to close an IF cycle

Explanation:
Check the correspondence between IF-ELSE and ENDIF, FOR and ENDFOR. See warnings for the above error.

Example:
Here is an example of wrong programming:
IF …
  IF …
  …
  ENDIF
…
ENDFOR <- Endfor which unloads If

12.9 Errors in global function assignment

239 - An ELSEIF instruction is used in an IF cycle after an ELSE

Explanation:
In an IF loop an ELSEIF instruction was defined after an ELSE instruction: in an IF loop, if you use an ELSE instruction, it must be the last one of the loop, before ENDIF.

Context:
In addition to occurring in the definition of an IF instruction, it can also occur in applying a subroutine or a macro.

240 - Custom function name is unassigned

Explanation:
Function name unassigned.

Context:
This error message means that an incorrect working is assigned to the working database.

241 - Custom function name is invalid

Explanation:
Invalid function name.

Context:
This error message may identify one of the following situations:
- an incorrect working is assigned to the working database;
- an incorrect custom function is assigned during the program configuration.

242 - Error during the carrying out of custom function: returns are unassigned

Explanation:
the global function development is invalid.

Context
This error message identifies a parametric programming error situation: this error is detected during function interpretation and development.
It also means that no variable (j) is assigned.

12.10 Errors in multiple setups (profiles)

The reporting of these errors occurs only on request for program optimization or piece matrix creation.
They are always fatal errors, which make impossible to run the program.

245 - Development of multiple profiles exceeds maximum number of workings that may be assigned on a face

Explanation:
The procedure of development of multiple setups cannot be terminated in the selected face, since the maximum allowed number of workings has been reached.

12.11 Errors in the assignment of technological parameters for profile and point workings

The reporting of these errors occurs only on request for program optimization or piece matrix creation.
They are always fatal errors, which make impossible to run the program.

250 - Impossible to apply setup to an open profile as reference code is missing

Explanation:
The concerned operation cannot be terminated since the necessary setup working is missing. In particular, both the technological and geometric setups are missing. The technological setup is assigned at the software configuration item level.

251 - Impossible to apply a technological point as reference code is missing

Explanation:
The concerned operation cannot be terminated since the necessary technological working per point, for replacing geometric points, is missing. The technological point working is assigned at the software configuration level.

252 - Impossible to assign open profiles

Explanation:
Open profiles or profiles headed with geometric setup have been found, but their execution is not supported by the software configuration. In this case, it is necessary to assign a technological setup for each profile.

253 - Technology replacements were carried out

Explanation:
This report is not serious (it is a Warning) and indicates that one or more technological replacements were carried out due to the wear condition of programmed tools.

254 - Could not replace a technology

Explanation:
the report indicates that it was not possible to replace a technology, due to the wear condition of a programmed tool. In this case we need to solve by making the tool valid through a replacement or by allowing a valid replacement to be found.

12.12 Errors assigning Entry/Exit segments to profile

271 - Enter/Exit profile: cannot solve a 3D arc

Explanation:
This procedure cannot solve the required typology of segment, because the working "circular segment with development in the space" is not available in the workings configured for the application.
In the TpaCAD context, the procedure solves a linear segment and the report corresponds to a "Severe warning".
In an executive context: the reports correspond to an Error.

272 - Enter/Exit profile: programmed geometry is not compatible with the request for tool compensation

Explanation:
The procedure cannot solve the segment typology "**Approach**"/"**Removal**"/"**Coverage**", because it is not compatible with the simultaneous request for tool compensation. The report is generated in the event that the segment is not in continuity of tangent with the initial/final segment of the profile.
In the TpaCAD context, the procedure does not solve any enter/entry segment and the report corresponds to a "Severe warning".
The report corresponds to an Error, when the program is executed in the machine.

273 - Enter/Exit profile: cannot solve a covering segment, if the profile is not closed

Explanation:
The procedure cannot solve the typology of the final segment in "**Coverage**", because the original profile is not closed.
If the original profile starts with a linear segment or with an arc in yx plane, the closure needs only to be checked on the (x;y) coordinates. If the original profile starts with an arc in #xy plane, the closure needs to be checked also on the depth coordinate.
In the TpaCAD context, the procedure does not solve any enter/entry segment and the report corresponds to a "Severe warning".
The report corresponds to an Error, when the program is executed in the machine.

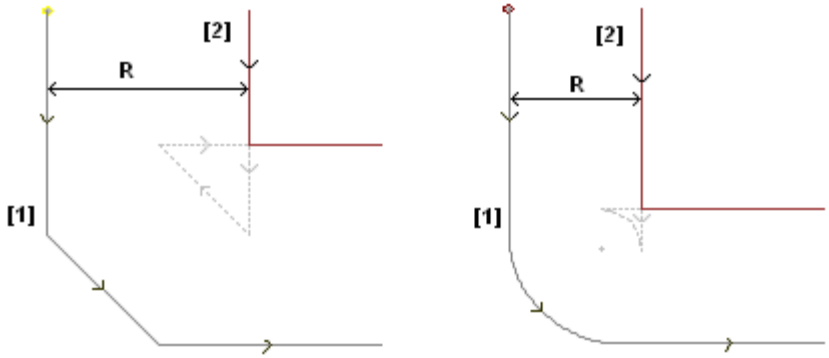
12.13 Errors in applying tool corrections

They are always fatal errors, which make impossible to run the program.

261 - Tool compensation: correction exceeds the arc radius

Explanation:
Internal correction is required for an arc, with compensation value greater than the arc radius. If it is necessary to apply compensation, this error can be solved by enabling the Reduce profile command.

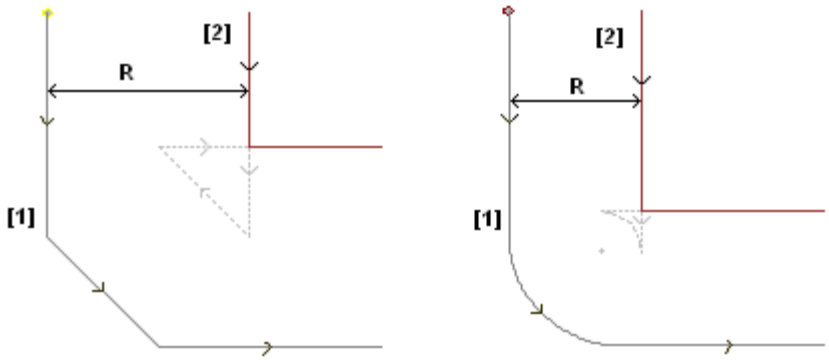
Restating what was said in the paragraph on the application of the *Tool compensation*, the situation corresponds with the case in the picture, on the right:



262 - Tool compensation: correction exceeds the line

Explanation:
Correction is required for a linear segment, with application of such compensation value that the segment direction is inverted. If it is necessary to apply compensation, this error can be solved by enabling the Reduce profile command.

Restating what was said in the paragraph on the application of the *Tool compensation*, the situation corresponds with the case in the picture, on the left:



263 - Tool compensation: applied reduction to profile

Explanation:
the tool compensation of the profile required the application of a reduction.
The report is managed as an **alert** and not as an error. The purpose of the report is to highlight the possible need for a check on the outcome of the correction.

265 - Tool compensation: error during correction on different xy-plane, with intersection solution of the line segments

Explanation:
Correction is required for a circular segment assigned on a plane different from xy, while the intersection condition is verified within the arc. This error can be solved by deleting the segment or disconnecting segment compensation.

266 - Tool compensation: error for correction in different xy-plane

Explanation:
Correction is required for a circular segment assigned in a plane different from xy, when the condition of original arc, inverting one of the two x or y-coordinates, while the execution of the segment, is verified. This error can be solved by deleting the segment or disconnecting segment compensation, or by modifying the geometry or the arc value.

267 - Tool compensation: inverting a correction should resolve an intersection or resume an interruption

Explanation:
It indicates that in a profile the inversion of a correction has been requested as a not inverted segment; or that a correction, started again after a suspension has not been set or that it is not possible to solve an intersection of the correct segments. The error can be solved by setting a segment with interruption of compensation.

268 - Tool compensation: suspension of correction without consecutive resumption had been required

Explanation:
In a profile a suspension of correction has been executed without consecutive resumption. This error can be solved by setting the resumption of correction in the selected point.

269 - Tool compensation: suspension and consecutive resumption of correction can't compute a connection

Explanation:
in a profile has been recognized as an erroneous suspension and resumption of correction: the correction solution between the two sections concerned does not resolve one intersection.

270 - Tool compensation: a suspension and consecutive resumption of correction must verify geometric continuity of line segments

Explanation:
if a profile has been recognized as an erroneous suspension and resumption of correction: the two contours concerned must have geometric continuity (the first ends at the starting point of the second).

12.14 Errors of fragmentation and linearization of arcs in planes different from xy

The reporting of these errors occurs only on request for program optimization or piece matrix creation.
They are always fatal errors, which make impossible to run the program.

255 - 3D arc linearization exceeds the maximum number of lines

Explanation:
It is not possible to terminate the concerned operation in the selected face since the maximum allowed number of workings has been reached.

256 - Impossible to linearize 3D arcs as reference linear code is missing

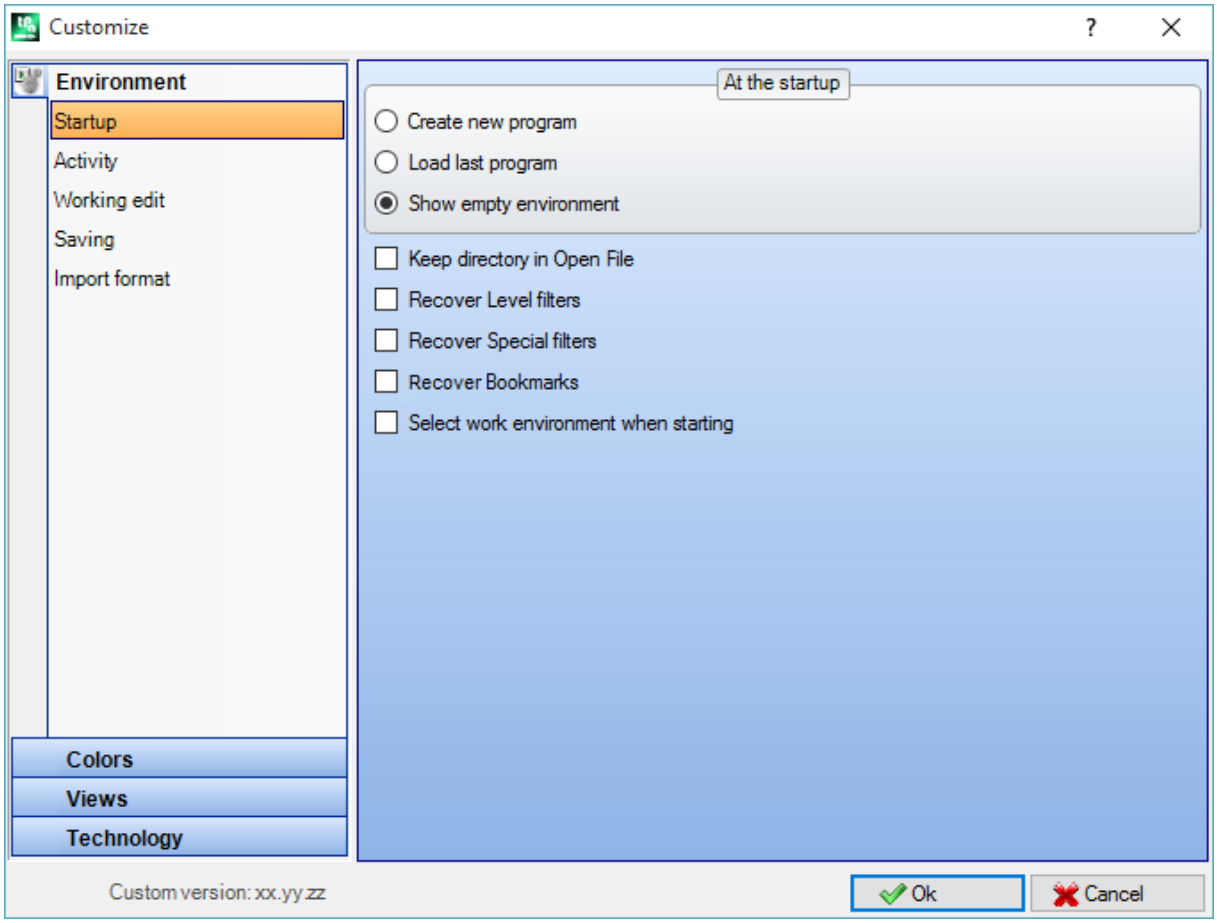
Explanation:
It is not possible to terminate the concerned operation since the working with operating code L01 [code= 2201] is unavailable.

13 TpaCAD Customization

In order to customize TpaCAD, the menu item **Customize** has to be selected from the Application menu.

13.1 Environment

Startup



At the start:
select TpaCAD behaviour at starting.

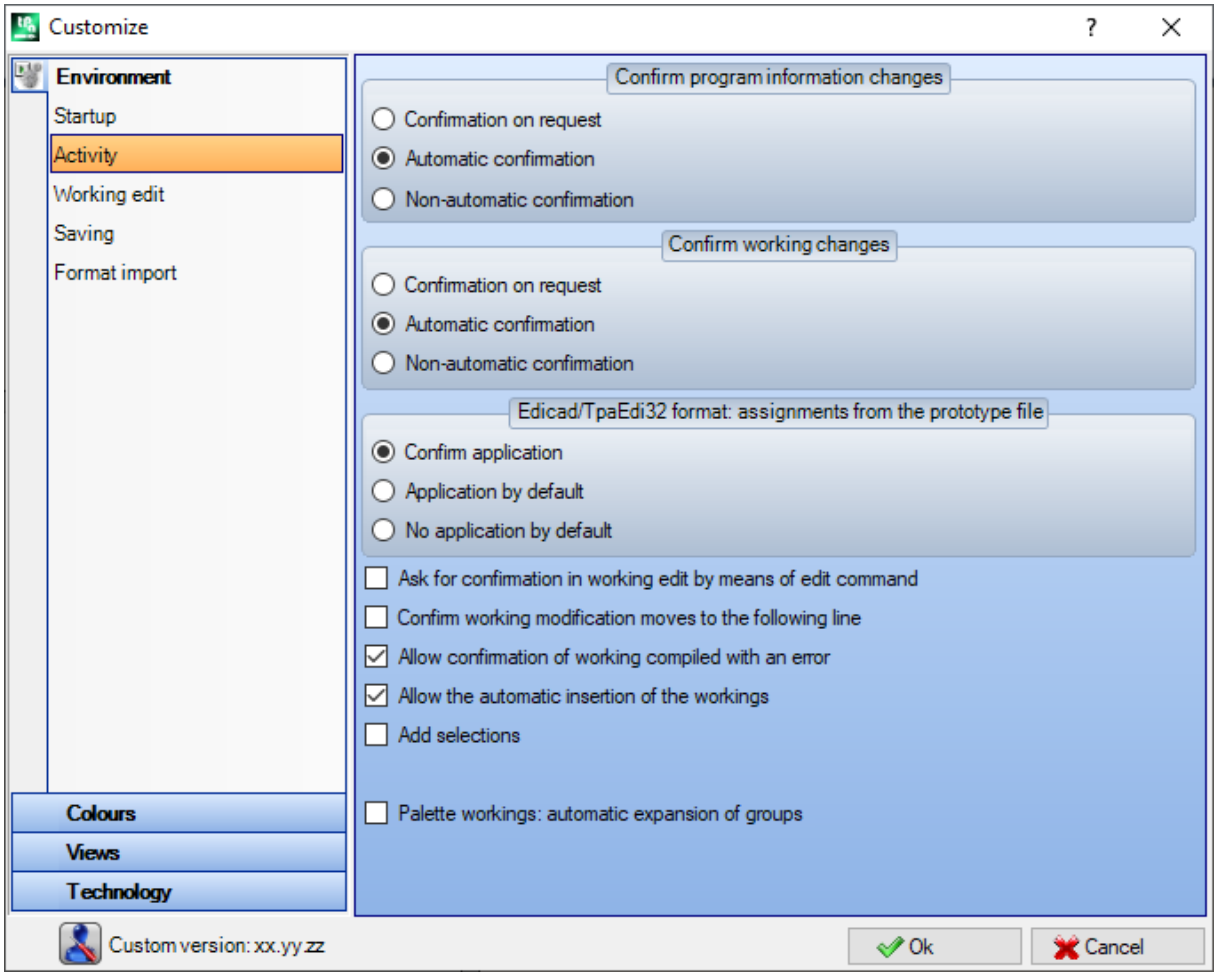
- **Create new program**: creates a new program based on prototype file.
- **Load last program**: loads the last opened program.
- **Show empty environment**: no program has been loaded.

- **Keep directory in Open file**: if it is enabled the Open Piece window shows the folder and the file typology of the last file opened. If the default is disabled, the Open Piece window proposes the programs storage folder and the programs-piece typology.
- **Recover Level filters**: if enabled, at the start-up of TpaCAD, the free or locked status of Levels are recovered as they were set the last time the program was started. Otherwise: neither visibility nor editing restriction on Levels is predefined. The default is disabled. This item is unavailable if the Level property is not managed.
- **Recover Special filters**: if enabled, at the start-up of TpaCAD, the free or locked status of Special Filters are recovered as they were set the last time the program was started. Otherwise: neither visibility nor editing restriction on Special Filters is predefined. The default is disabled. This item is unavailable if the Special Filters are not managed.
- **Recover Bookmarks**: if enabled, while starting TpaCAD the bookmarks are recovered like they were set in the last closure of the application program. The default value is disabled. This option is not available, if the section of the Bookmarks is not managed.

- **Select work environment when starting**: if enabled, when starting TpaCAD, it is possible to choose which environment should be selected, Machine or Draw. This option is not available, if the functionality of the double work environment is not managed.

Custom version: it gives a useful string to identify the installed version of custom configuration. The information is read from a file version that has to be managed by the customization responsible: it identifies database versions of custom workings.

Activity



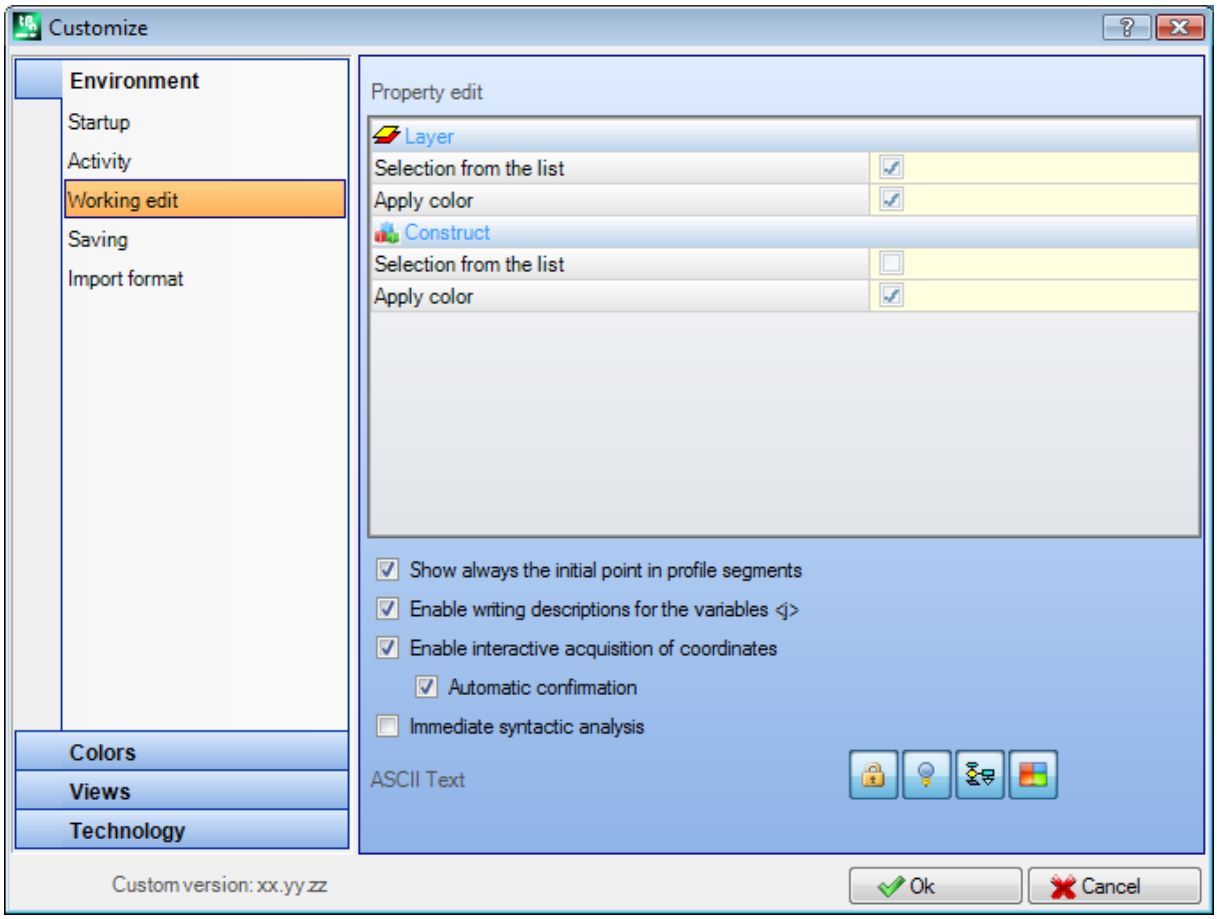
Confirm program information changes
It sets the way a general info change of the piece is managed (dimensions, variables, custom section…), even if it is not explicitly confirmed. Three options are listed:
- **Confirmation on request**: the system requires if it has to acquire changes.
- **Automatic confirmation**: changes are automatically saved.
- **Non-automatic confirmation**: changes that are not confirmed explicitly will be lost.

Confirm working changes
it sets the way a working change in program face is managed, even if it's not explicitly confirmed. Three options are listed:
- **Confirmation on request**: the system requires if it has to acquire changes.
- **Automatic confirmation**: changes are automatically saved.
- **Non-automatic confirmation**: changes that are not confirmed explicitly will be lost.

Edicad/TpaEdi32 formats: assignments from the prototype file
It sets how the assignments from prototype file are managed in the case of the program opening of Edicad or TpaEdi32 format. Three options are listed:
- **Confirm application**: the system asks if it has to acquire the assignments.
- **Application by default**: assignments are acquired in an automatic way.
- **No application by default**: assignments are not executed.

- **Ask for confirmation in working edit by means of edit command**: if it is enabled, each time a user select a Delete, Paste, Move or Working Duplication Command, the command confirmation is required.
- **Confirm working modification moves to the following line**: if enabled, the explicit confirmation of working change (for example, the confirmation by selecting the button in the assignment of the working) moves the current working to the following line of the list.
- **Allow confirmation of working compiled with an error**: if this option is enabled, it is possible to confirm the insertion or the direct modification also if the operation has reported an error. Situations of severe and not recoverable error are excluded: an extreme situation occurs when the available system memory is exhausted.
- **Allow the automatic insertion of the workings**: if enabled, the insertion of a specific working is automatically finished without the need to provide confirmation. The selection concerns the insertion of a working selected from the graphic palette or from the list of the favourite workings. Enabling the direct insertion is a specific feature of each single working.
- **Add selections**: if enabled, the selections in the face list are not reset, when you choose a working either through direct pointing in graphic area (click in the area) or in the ASCII text.
- **Palette workings: automatic expansion of groups**: if enabled, the buttons (groups) of working palette open bringing the mouse over the same button. Otherwise: to open a group, you must click on it.

Working edit



Property edit

- **Selection from the list**: select the entry to manage the selection in corresponding property list, otherwise an edit box is managed. The selection is possible only if the property assigned a maximum value up to 16.
- **Apply colour**: select the entry to apply the attributed colour to the corresponding property, in working graphs.

Here are the default settings:

- "L" field: select in list, apply colour;
- "B" field (construct): select in list, apply colour;
- "O" field: select in list, it does not apply colour;
- "K" field: the selection is not in list, it does not apply colour (it's not available);
- "K1" field: the selection is not in list, it does not apply colour (it's not available);
- "K2" field: the selection is not in list, it does not apply colour (it's not available).

If the list selection is enabled, the entries that compose the list can be assigned:
- in customization phase (in the colour tables) for "L" and "B" fields;
- In configuration phase for "O", "K", "K1" and "K2" fields. In this case messages are translated.

If the working graphic representation has a lot of requests of the Apply colour selection, these criteria are applied: "L" field prevails on "B" and "O" fields; "B" field prevails on "O" field.

- **Bitmaps of references for O-Field**: the option is available only if the O field can have a value up to 1. It's possible to choose among two options to display the meaning that is attributed to the significant values for the field (0/1):
  - Left side/Right side
  - Bottom side/Top side.

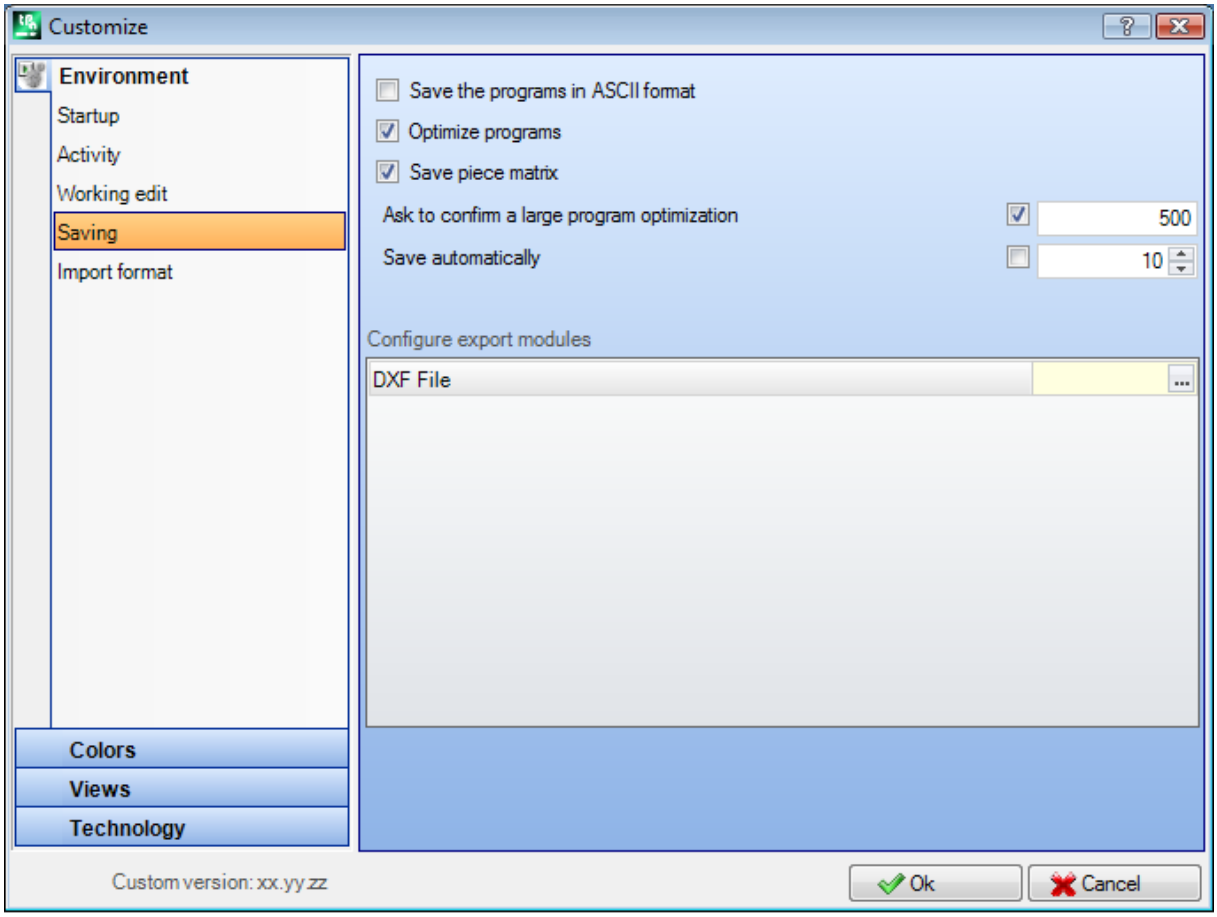
ATTENTION: to manage this item, you must enable specific options in the configuration of TpaCAD.

- **Show always the start point in profile segments**: the entry is about the parameters that correspond to the start point coordinates of a profile segment (arc or line). Select the entry to visualize and manage the fields; the entry has to be unselected to retain the coordinates invisible, but only if they are unset. If the option is not selected, the fields are still visible, even if not set, if the working (arc or line) actually starts a profile: it is the first programmed in the face list, or it is preceded by a working that does not belong to a profile.
  The status of the field can also be changed with the button in the command bar of the current machining, applying the same criteria.
  This modality leaves the interactive insert features of geometric elements unchanged, so they directly acquire the start application point, while it can only simulate the schedule to the end point of a profile segment that is inserted by direct selection from the working palette. Default setting: enabled.
- **Enable write descriptions for the variables <j>**: select the entry to allow the edit of the fields that correspond to the descriptive messages of the <j> variables, into inserts of the assignment statements of the variables. If the entry in not selected, the fields are visible, but they can't be modified: if there are already assigned settings, they will be displayed. Default setting: enabled.
- **Enable interactive acquisition of coordinates**: select the entry to allow the interactive acquisition of the coordinates from the working data-entry, as in the working database configuration (fields showing the  icon). Default setting: enabled.
  - **Automatic confirmation**: selects the item to confirm automatically the modification of the working, after an interactive capture of the coordinates from the working data-entry. Default is not active. Just when the interactive acquisition closes, you can refuse the here selected condition by pressing the Shift key.

WARNING: to manage these items, you must enable specific options in the configuration of TpaCAD.

- **Immediate syntactic analysis**: select the entry to execute the parameter setting valid control, when there is a change of a working parameter. In case of wrong setting, the error is immediately signalled. Default setting: disabled.
- **ASCII Text**: the group of the graphic selections assigns the activation state of the auxiliary columns in the table of the ASCII text respectively for:
  - **Edit status**: the column shows the edit status of the working
  - **Display status**: the column shows if a visualization in the area of the graphic representation corresponds to the working.
  - **Logical status**: the column shows the logical status of the working, if the option *View of logical conditions* is active.
  - **Colour**: the box shows the primary colour associated to the working, according to the kind of working (points, setup, profile segment) or to the operation code. According to the typology the colour is assigned in the following group of setting or in the database of the workings. If the code is complex (subroutine call or macro code) and any customized colour in the working database is not assigned and if the development corresponds to a profile, the colour set for the profile elements is assigned to the program line.
    The activation is applied also in the expanded working list and in the window of Sequences.

Saving



- **Save the programs in ASCII format**: select the entry to save the programs in ASCII format. In this context ASCII format is not the file type that is recorded, that is a text file of ASCII type, but the coding used to record the program info, principally the workings. The ASCII format can be used for intuitive reading and it can also be used to create programs for TpaCAD. Hereafter the row that corresponds to the registration of a drilling work (operation code: 81; name ASCII: HOLE) for the ASCII format and for the internal format respectively:
  - HOLE WS1 EG0 X100 Y100 Z-12 TD10 TMC1 TR1 TP1
  - W#81{::WTp WS=1 #8015=0 #1=100 #2=100 #3=-12 #1002=10 #201=1 #203=1 #1001=1 }W.

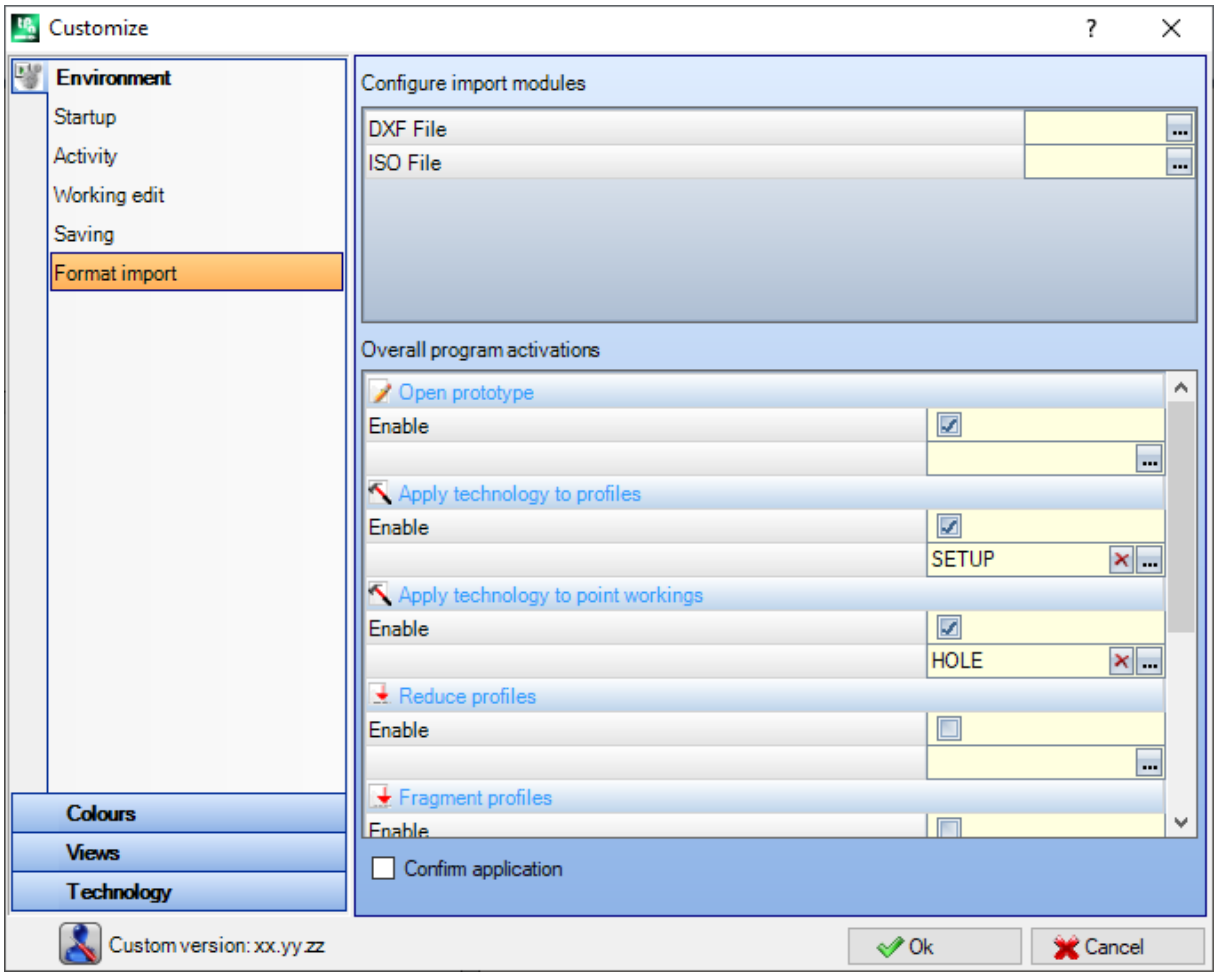
  When an ASCII format program is read, if there was no correspondence between the loaded workings and the defined workings of the workings database, source lines would be deleted. The default is disabled.
  The selection of this option is not applied to the saving of a macro-program.
- **Optimize programs**: if enabled, after the program has been stored, program optimization is performed. The default is disabled. If it is enabled:
  - if an optimization program is not set, the verification is limited to a general program execution, with the application of: logic conditions, technological default assignments, tool compensation). This analysis phase can be ended with diagnostic referrals;
  - if an optimization program is set, it checks piece for the execution (if necessary, with the application of path optimizations and/or stops and/or changes of tools...). The overall analysis of the previous point is executed in any case, before of the program optimization.
- **Save piece matrix**: if it is enabled the optimization of a program saves on disk a file that corresponds to the optimized version (otherwise called piece matrix; with file extension .TXN and\or .TXM). The entry is shown only if an optimization program is set and the possible selection is applied only if it is selected: **Optimize programs**. The default is disabled.
- **Ask to confirm a large program optimization**: if enabled, after the storage program, it asks the confirmation to execute the optimization or the format export, if the program dimension is bigger than the set value. The value to set is in KB units between 1 and 50000. The default is disabled.
  ATTENTION: this setting is used, even when programs are being opened:
  1. to manage the graphic preview.
     If there is a program that has a bigger dimension than the set one, the preview is temporarily disabled in an automatic way. In any case, to require the graphic preview of the program, it is necessary the activation of the entry **Preview** in the Open File window.

2. to manage the graph of the program. (See: **TpaCAD Customization -> Views -> Customize graphics**, setting **Deactivate the additional graphic elements for a large program**).

- **Save automatically**: if enabled, an automatic save copy is automatically created for the program in the specified time interval. The value to be set is between 1 and 60 in minute units. The automatic saving is carried out in the folder of the TpaCAD temporary data, in a temporary file with a fixed name and a TBK extension. Default is not active.
  In normal operating conditions, the automatic saving file is deleted when TpaCAD is closed.
  If this does not occur, for example when an error forces an unexpected closure of TpaCAD or if the supply of electric power is suspended, while the modification of a program is in progress, it is possible to recover the last data saved. Starting TpaCAD a message reports that a copy of the automatic saving file has been found and that it is possible to recover its contents: the same file is copied in a non-temporary backup file (same folder and name, but with SBK extension), to reopen it later.
  IMPORTANT: Automatic Saving cannot exactly respect the specified time, if an interactive procedure or any other command is being executed. In this case the data are saved as soon as possible at the end of the current functionality.
  IMPORTANT: Automatic Saving cannot replace the **Save** command. The user still needs to save the program at the end of the work.
  IMPORTANT: Automatic saving cannot be used to recover a program that has been closed without being saved.

  Configure export modules
  - the export modules, that are defined in the configuration phase by the machine constructor and that allow to select the Configuration at this level, are shown. Clicking on the icon [...] it is possible to define criteria and parameters that will be used in the export phase of the program. If it is not possible to set an export module, the area is not shown.

Import format



Configure import modules

- the import modules, that are defined in the configuration phase by the machine constructor and that allow the selection of the Configuration at this level, are shown. Clicking on the icon  it is possible to define criteria and parameters that will be used in the import phase of the program. If it's not possible to set an import module, the area is not shown.

Overall program activations

The table assigns the automatic assignment features that are executed at the program opening when the import format is activated. Most features that are displayed have a total correspondence in *Overall program tools* already considered:

- **Open prototype**: if it's enabled, it focuses the program initializing based on the default prototype file (the file PIECE.TCN, archived in the folder TPACADCFG\CUSTOM). If the file exists, the program initializes, reading from the same prototype file, the execution types, the variables "o" and "v", the custom sections that are managed (among special Settings, added Info, Constraints, Optimization Settings) and the face names. If the icon is selected  the window is shown, and it displays the full file path from where the import program is imported. When the face piece is managed, the following option to be activated can be used:
  Face-piece: Always update from prototype file: select the field to require also the assignment of Face piece from the prototype file.

ATTENTION: in any case, the initializing from prototype file are executed complying with possible section blocks, if they are assigned respecting the file created in the import process.

- **Apply technology to profiles**: if it's enabled, it assigns the setup working code opening the profiles that are imported as opened profiles (read: without setup) or that begin with a geometric setup code. By selecting the icon  it is possible to set the setup working and the technological assignments as they are required. Among the assignments, it is possible to set also the numeric properties (level, M field,...) aside from the field B (construct). No assignment is executed if the field is not enabled.
- **Apply technology to point workings**: if it's enabled, it assigns the point working codes that are imported with a geometric working code. Selecting the icon  it is possible to set the point working and the technological assignments as they are required. Among the assignments, it is possible to set also the numeric properties (level, M field...) aside from the field B (construct). No one assignment is executed if the field is not enabled.
- **Reduce profiles**: if it's enabled, it reduces the number of the follow segments of the profiles, according to the rules defines in the window opening, when the icon  is selected.
- **Fragment profiles**: if it's enabled, it executes the segment fragmentation of a profile, according to the defined rules of the window that is opened when the icon is selected  .
- **Connect profiles**: if it's enabled, it activates the merge profiles process according to the geometric continuous function of the profiles and to the defined rules of the window that is opened when the icon is selected  .
- **Validate profiles**: if enabled, this option activates the validation procedure of the profiles according to the programming of the setup in a non-neutral point, in accordance to the rules defined in the window that opens after having selected the icon  .

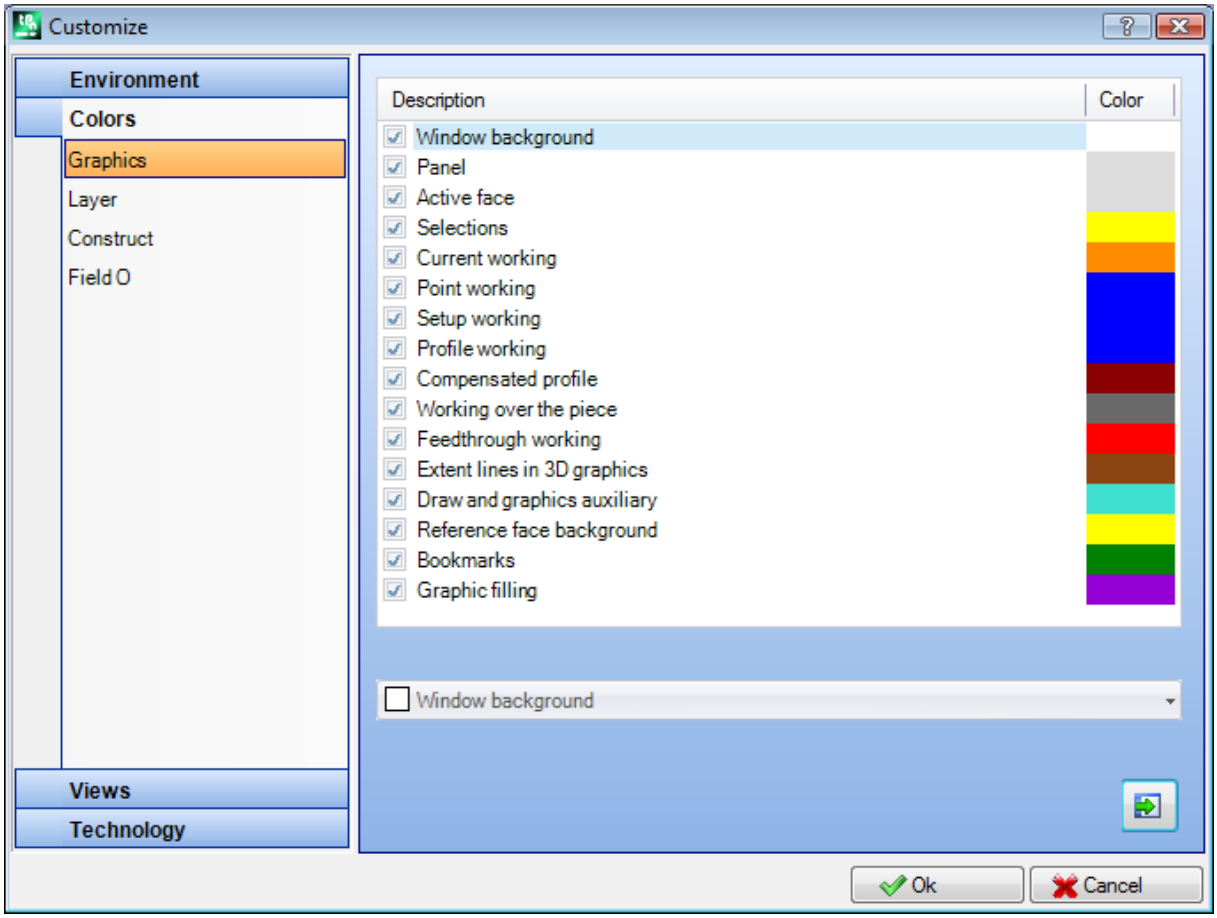
- **Miscellaneous procedures**: if it's enabled, it activates the procedures as they are assigned in the window that is opened when the icon is selected. More specifically:
- **Inverts depth axis**: if it's enabled, it inverts the schedule sign of the depth axis of a face.

Confirm application: it enables or disables, when a program is opened, a confirmation request before the enabled tools application in the previous table.

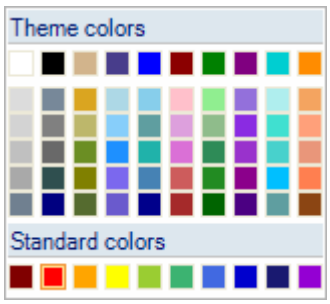
13.2 Colours

Graphics

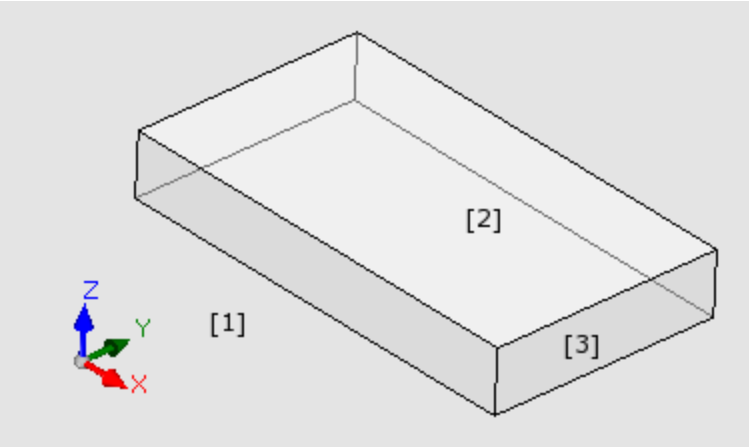
Graphic display colours



Moving the selection on the list, the below auditing place is updating with the writing and the colour that correspond to the selection. Clicking on the icon to set a colour



A check box is displayed for each entry and its status is editable only for some entries (see below):
- if the box is selected, the corresponding colour is normally applied;
- if the box is unselected, the colour is not managed.



- **Window background**: background colour for the graphic area (picture: colour **1**).
- **Panel**: background colour of the panel (picture: colour **3**). The check box can be inactive: in this case the panel fill is not executed with the set colour. **ATTENTION**: the fill with a graphic pattern has to be disabled to get a transparency effect of the panel with respect to the window background (see below);
- **Active face**: background colour of the current face (picture: colour **2**). The check box can be inactive: in this case the current face fill is not executed with the set colour. **ATTENTION**: the fill with a graphic pattern has to be disabled to get a transparency effect of the face with respect to the panel and, if necessary, to the window background (see below);
- **Selections**: graphic display colour of the selected workings
- **Current working**: graphic display colour of the current working. The check box can be disabled: in this case, the current working is represented using the own colour that comes from the schedule (from de level properties, construct or "O" field) or the colour that describes the working typology.

- **Point working**: graphic display colour of the point workings. The colour is used if one of the considerable situations is not recognized for the working (with established priority from the list order):
  - it is selected or it is current working;
  - it is executed over the piece or it is feeding-through;
  - it assigned a property colour (level, B field, O field);
  - it assigned a customized representation colour in working database.
- **Setup working**: graphic display colour for setup workings. The colour is used if one of the considerable situations is not recognized for the working as they are shown for the point workings
- **Profile working**: display colour of the segment profiles (linear segments, arcs). The colour is used if one of the considerable situations is not recognized for the working as they are shown for the point workings.
- **Compensated profile**: display colour of the profiles with applied tool compensation.

- **Working over the piece**: display colour of the workings over the piece: the evaluation is executed on the depth axis (Z coordinate) of the workings. if an arc or a linear segment is fully or partially executed over the piece, the concerned part is marked by this colour. The check box can be inactive: in this case, a planned working over the piece is represented as if it is planned in the piece.

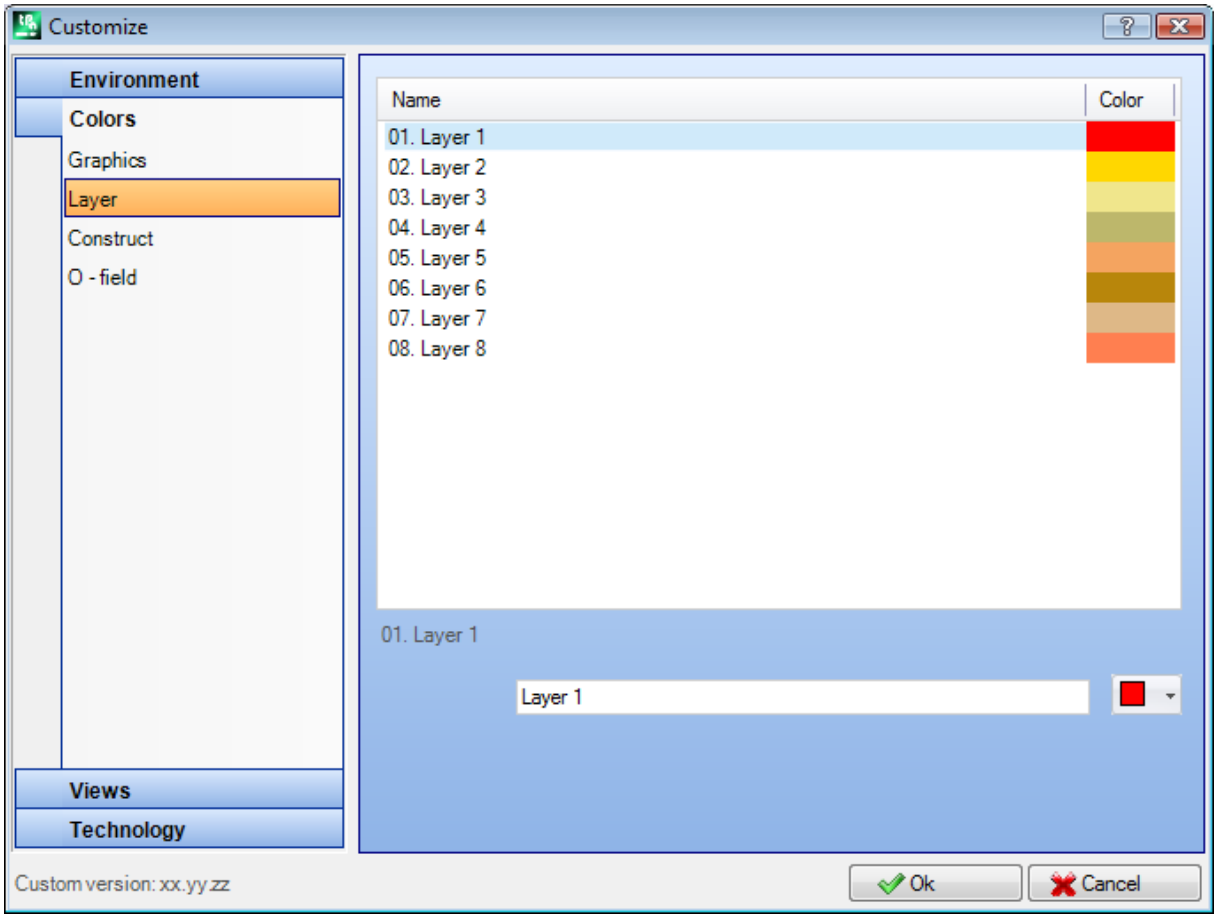
The entry may not be available, according to the TpaCAD configuration.
Workings over the piece examples:
- drilling or setup to the coordinate Z=10;
- line from Z=10 to Z=0;
- line from Z=10 to Z=-10 (it is partially executed over the piece).
- **Feedthrough working**: display colour of the workings executed over the face thickness (feed-through working). if an arc or a linear segment is fully or partially executed over the piece, the concerned part is marked by this colour. The check box can be inactive: in this case, a feed-through working is represented as if it is planned in the piece.
  The entry may not be available, according to the TpaCAD configuration.
  Feed-through workings examples (with piece thickness 18 mm):
  - drilling or setup to the coordinate Z=-25;
  - line from Z=10 to Z=-22 (from Z=10 to Z=0 is executed over the piece; from Z=-18 to Z=-22 is executed feed-through) graphic display colour of the workings executed over the face thickness (feed-through working): if an arc or a linear segment is fully or partially executed feed-through, the concerned part is marked by this colour. *Attention*: the recognition of feed-through working prevails on a possible colour of the priorities (level, B field, O field).
- **Extent lines in 3D graphics**: display colour for overall dimension of the workings in 3D graph The check box can be inactive: in this case, the overall dimension has the same colour associated to the working.
- **Draw and graph auxiliary**: colour used in:
  - drawing functions;
  - interaction of tools with the mouse (examples: translation point, mirror axis);

  - expanded list management in program ASCII text.
- **Reference face background**: reference face background colour, in form of fictive or automatic face edge assignment.
- **Bookmarks**: display colour of the graphic bookmark elements;
- **Graphic filling**: fill colour of the closed graphic elements, like the advanced schedule of macro-program. The check box can be inactive: in this case, the fill is realized using the own colour of the construct ("B" field).

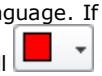
 Select to set the graphic colours to a default colour set.

Layer

This board is unavailable if the property is not managed.



The table dimension is based on several rows (8 in the picture) equal to the maximum number that can be assigned to levels and, in any case, for a maximum number of 16. Property values bigger than 16 apply the colour that is here assigned for the 16 value.

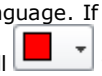
- **Header**: layer number, progressive number that starts from the value 1.
- **Name**: name to assign to the layer. If it is not set, a default name is assigned: in this case, the name is translated in current language. If the name is edited, it is unchanged from language translation.
- **Colour**: click on the cell  to set a colour to the layer.

All workings which have a "L" field different from 0 can be displayed with the colour here assigned to the level value. The colour field depends on the level default use and is an aid to the graphic representation.
It is also possible to assign a level value that excludes the working graphic (see below).

Construct

This board is unavailable if the property is not managed.
The table dimension is based on several rows equal to the maximum number that can be assigned to the constructs and, in any case, for a maximum number of 16. Property values bigger than 16 apply the colour that is here assigned for the 16 value.

- **Header**: construct number, progressive number that starts from the value 1.

- **Name**: name to assign to the construct. If it's not set, a default name is assigned: in this case, the name is translated in current language. If the name is edited, it's unchanged from language translation.
- **Colour**: click on the cell  to set a colour to the construct.

All workings which have a "B" (Construct) field value different from 0 can be displayed with the colour here assigned to the construct value.
It is also possible to assign a construct value that excludes the working graphic (see below).

Field O

This board is unavailable if the property is not managed.
The table dimension is based on several rows equal to the maximum number that can be assigned for the property and, in any case, for a maximum number of 16. Property values bigger than 16 apply the colour that is here assigned for the 16 value.

- **Header**: property number, progressive number that starts from the value 1.
- **Name**: name to be assigned to the property. The file cannot be modified.
- **Colour**: click on the cell  to set a colour to the property.

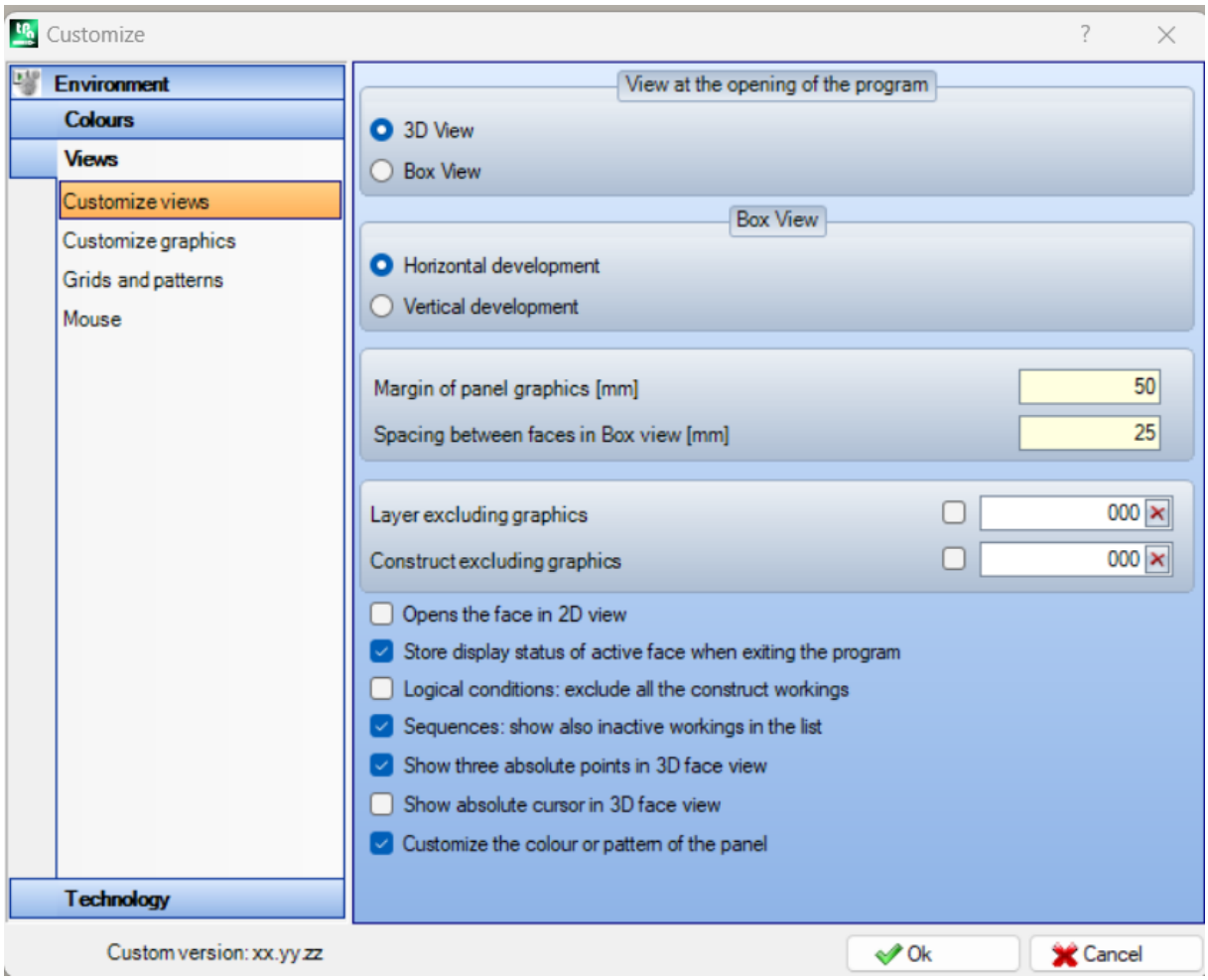
The names assigned to the values of O field are used if there is the selection in list of the property value.

Nesting

The board is not available if the *Nesting* functionality is not managed.
The table dimension is based on 25 rows, tagged from "ID1" to "ID25", associated to the first 25 allocatable pieces in a Nesting project. In case of allocation of consecutive rows, these same colours will be used, with adjustments on different shades.

13.3 Views

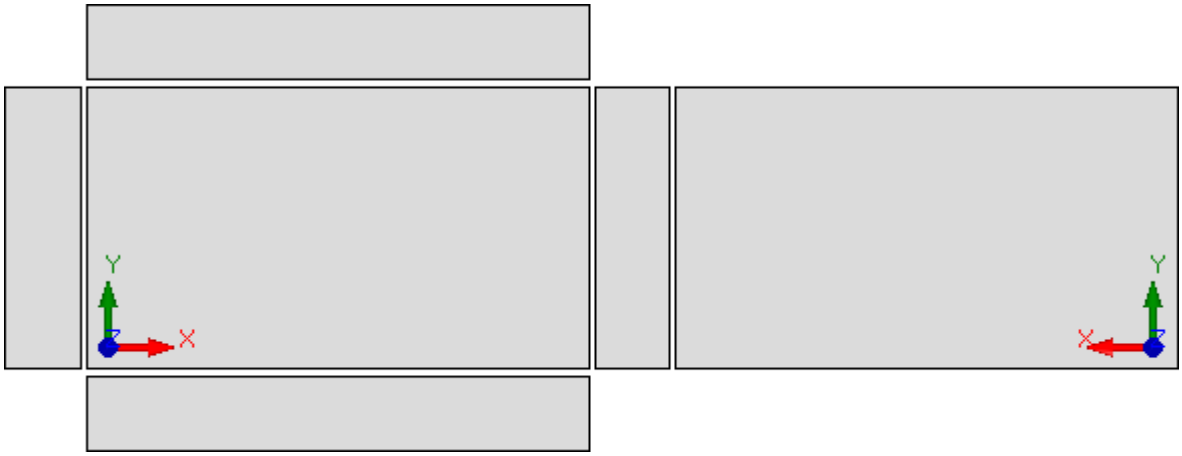
Customize views



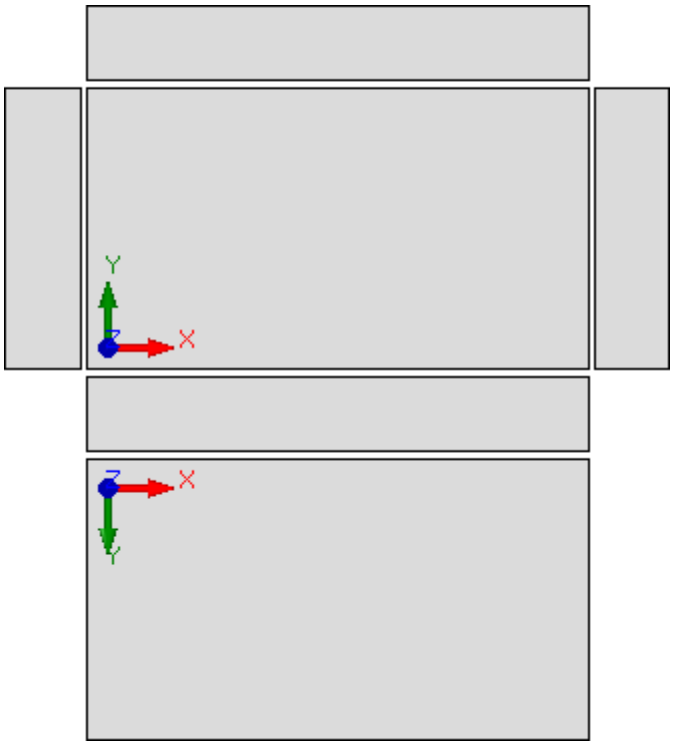
View at the opening of the program
select the display mode that is assigned atCouche the program opening. Two entries are available:
- **3D View**: The panel is shown in 3D.
- **Box View**: The panel is shown exploded (applying the selection that follows).

The selection may not be available, according to the TpaCAD configuration.

Box View
select the display mode that is assigned at the exploded view of the panel. Two entries are available:
- **Horizontal development**: the panel is exploded with horizontal development (if the face 2 is assigned, it's displayed on the right with the horizontal opening of the panel).



- **Vertical development**: the panel is exploded with vertical development (if the face 2 is assigned, it is displayed below with the vertical opening of the panel).



The selection is irrelevant in the cases of:
- face 2 no assigned;
- face 1 no assigned. In this case the face 2 is represented instead of the face 1.

The side faces are represented also if they are partially no assigned.
The selection may not be available, according to the TpaCAD configuration.
- **Margin of panel graphs**: margin left around about the piece graphic representation. To set in: [mm] or [inch] (units of measurement for setting parameters). The field values between 0 mm and 100 mm.

- **Spacing between faces in Box view**: margin left as spacing between the faces in case of graphic representation of the piece in Box View. To set in: [mm] or [inch] (units of measurement for setting parameters). The field values between 0 mm and 100 mm. The selection may not be available, according to the TpaCAD configuration.

- **Layer excluding graphs**: Level ("L" field) value which excludes the working graphs. The selection is useful, for example, when the working graph is fully based on the use of constructs. Values between 0 (in this case: it never operates) and the maximum usable value. Select the check mark box to apply the graph exclusion. To clear the set value, click on the icon . This option is unavailable if the Level property is not managed.
- **Construct excluding graphs**: construct value ("B" field) excluding the graphic display of the working. The selection is useful when the working graphics is used to create constructs. with the downstream usage of workings that apply geometric transforms (STOOL type codes). Values between 0 and 255 (in this case: it never operates). Default value: 225. To type the value to use, it is firstly required the enabling of the edit field (select the check mark box). To clear the set value, click on the icon . This item is unavailable if the Construct property is not managed.

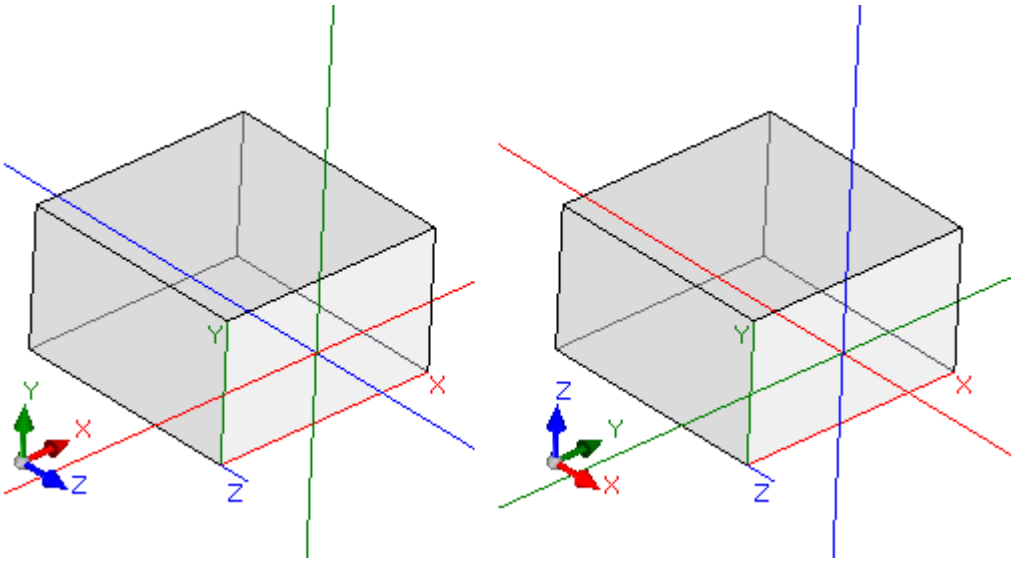
- **Open the face view in 2D view**: select to set the view on the XY plane, when the view of a face opens. If the item is not selected, the piece representation mode remains unchanged when the face view opens. Default setting is disabled. The graphic representation remains unchanged when the view of a face-piece opens, even if the item is selected.

- **Store display status of active face when exiting the program**: enables storage of display status of active face when closing TpaCAD. Next time TpaCAD is started, the stored face, if it exists, will be activated and the saved display statuses and the face representation mode (3D view, 2D view or box view) will be applied to it.

- **Logical conditions: exclude all the construct workings**: it enables the possibility to exclude from the *Logic Conditions View* all the construct workings. If the entry is unselected, in Logical conditions View, the construct workings (with positive "B" value) are considered with respect to direct Logical conditions (IF… ENDIF, EXIT). The default setting is disabled.
- **Sequences: show also inactive workings in the list**: it enables or disables the complete display in carousel view. If the entry is unselected, in carousel view are selected only the workings for those is possible the sequence assignment; so, the open profiles and the workings for those is directly disabled the sequence management are not visualized. The default setting is disabled.

- **Show three absolute points in 3D face view**: it enables or disables the absolute Cartesian coordinate system visualization in the 3D face view. If not selected, the represented three points is the one that reflects the face orientation. The default setting is disabled.
- **Show absolute cursor in 3D face view**: it enables or disables the absolute cursor visualization in the 3D face view. If not selected, the represented cursor is the one that reflects the face orientation. The default setting is disabled.

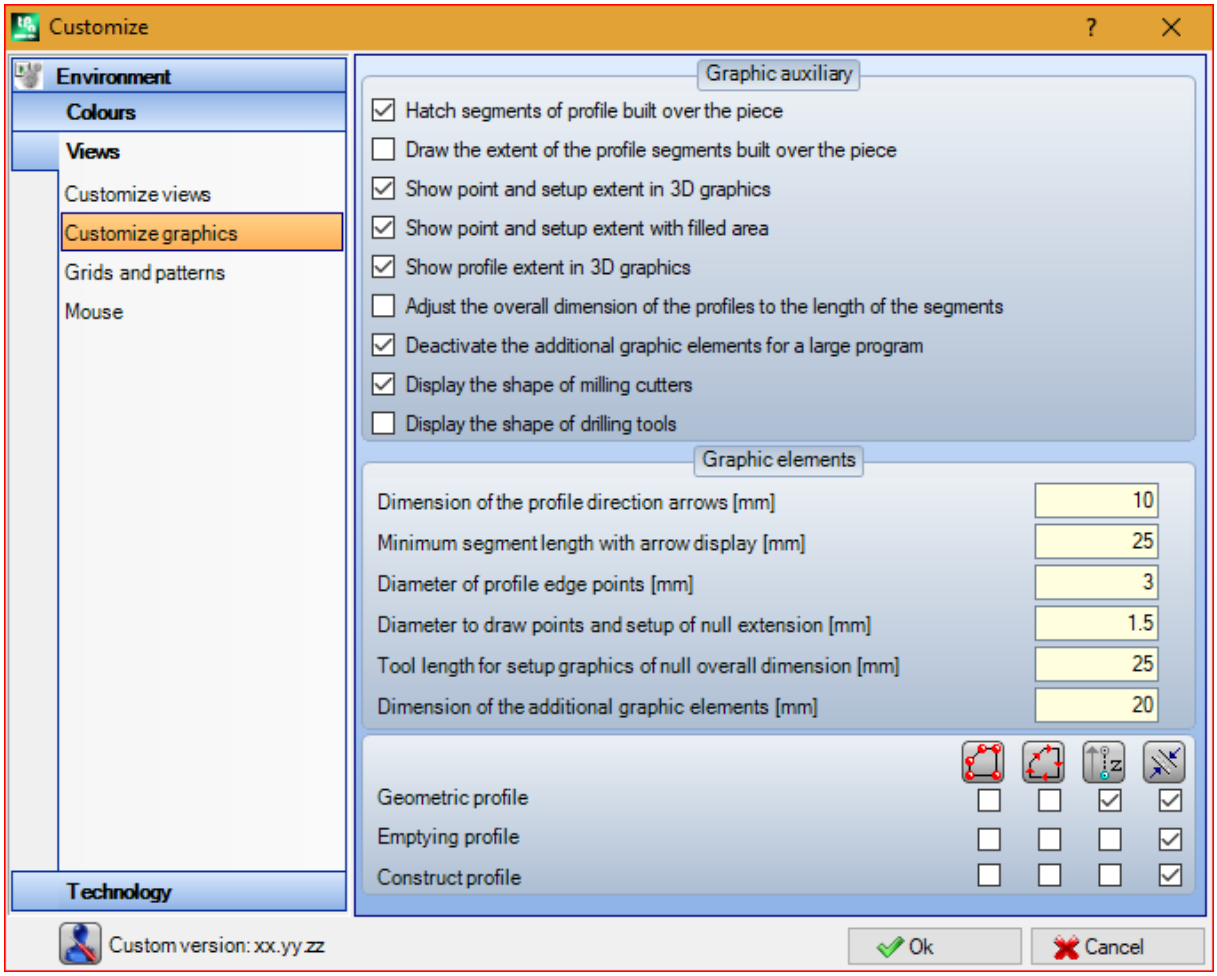
The activation of the 4-face view allows to see how the three points and the cursor representations change.



[on the left] both the entries are disabled: Show the local three points and cursor to the face;
[on the right] both the entries are enabled: Show the absolute three points and cursor on the face.

- **Customize the colour or pattern of the panel**: enables or disables the application of the current program of the colour or pattern, if assigned in *Special Settings* of the program. The default is not enabled, and the voice may not be visible to the assignment.

Customize graphics



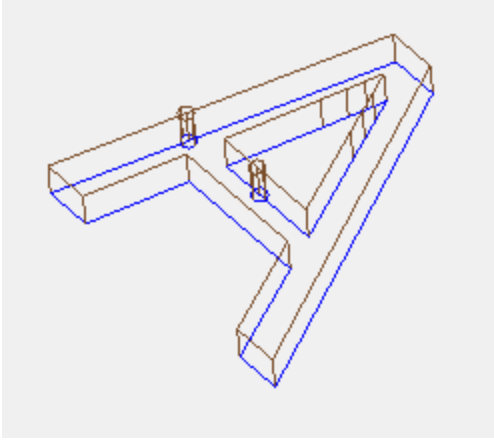
Graphic auxiliary

- **Hatch segments of profile built over the piece**: select to enable the profile segments visualization executed over the piece with dotted lines. The default setting is enabled. The selection may not be editable, according to the TpaCAD configuration.
- **Draw the extent of the profile segments built over the piece**: select to enable the visualization of the overall dimension tool even for the profile segments executed over the piece. The overall dimension tool on the profile segments is displayed in *tool compensation view*. The default is disabled. The selection may not be editable, according to the TpaCAD configuration.
- **Show point and setup extent in 3D graphs**: it enables or disables the overall dimensions visualization in the 3D representation of point or setup workings. The overall dimension indicates the usage of the tool in the piece, it is considered according to the programmed depth in relation to the effective entrance of the tool in the piece.



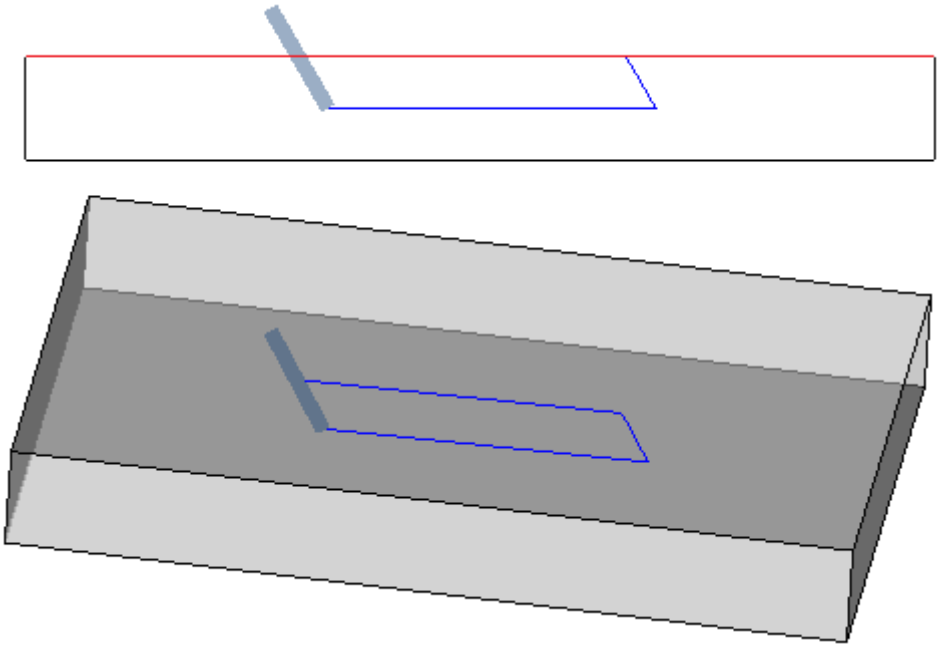
The drawing shows a 3D impersonation example of point workings (or isolated setups), that have an overall dimensions view: for each working is represented a little cylinder that is as high as the programmed depth. The little cylinder fill is determined by the selection **Show extent of points and setups with full segment**. The default setting is disabled.
The application of the selection is conditioned by the entry state **Overall dimensions in 3D graphs**, in View menu and of the local menu associated to this entry: if the entry in unselected, the option **Show point and setup extent in 3D graphs** is ignored.

- **Show profile extent in 3D graphs**: it enables or disables the overall dimensions view in 3D representation of profile workings. The overall dimension underlines the tool usage in the piece, considered on the programmed depth.

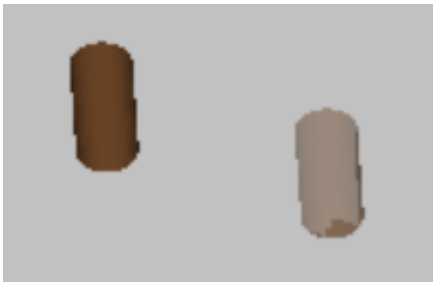


The drawing shows a 3D representation example of profiles, with the overall dimensions view enabled: each profile segment underlines the tool overall dimension in the piece. The default is disabled.
The application of the selection is conditioned by the entry state **Overall dimensions in 3D graphs**, in View on menu. The value of the selection can be changed from the local menu, managed on **Overall view in 3D graphs**, in the View on menu.
If the profile is programmed with an oriented setup, the setup is represented by a cylinder drawn oriented and the associated profile is displayed with the segments oriented according to the parameters assigned in the setup.

Both drawings are an example of thee-dimensional view of oriented profile, where the display of the overall dimensions is enabled. Setup is indicated on the left, with the tool point turned downward. The profile segment underlines:
- the tool overall dimension inside the piece (low segment);
- the contact segment with the face plan (on top segment, with front side view of the piece).



- **Show point and setup extent with filled area**: if the option is enabled, the overall dimensions display in the 3D representation of point and setup workings shows a little cylinder with full stroke (in the picture, on the left). If the option is disabled: the visualization shows the external contours of the little cylinder in transparency (in the picture, on the right).



- **Adjust the overall dimension of the profiles to the length of the segments**: if this option is enabled, the display of the overall dimension of the profiles is limited by the length of the segments. The selection allows to reduce the amount of graphic elements corresponding to profiles fragmented in many little strokes. More specifically:
  - the value set at the entry **Minimum segment length with arrow display** (see following fields in the window) is applied for the length of the segments;
  - a cumulative criterion is applied on consecutive segments;
  - for segments that are considered "short" the view of overall dimensions excludes the end part.

- **Deactivate the additional graphic elements for a large program**: if this option is valid, the view of the additional graphic elements (arrows, extreme points, 3D overall dimensions, profile overall dimension) is set as non-active, while opening of a large program. The evaluation of a large program takes two elements into account:
  1. KB dimension of the file, set in **Environment -> Saving** (field associated to **Ask to confirm a large program optimization**)
  2. the number of the workings that the program processes, if greater than 100000 (the value used is variable, with a range between 5000 and 100000, depending on the size set in file kB).

- **Display the shape of milling cutters**: select to display the milling cutter tools with their real shape, for example conical or generally shaped. The selection is managed both when programming by tool and by diameter, with real application compatible with the technology used. If this item is not selected, the milling cutter tools are displayed with a cylinder.

- **Display the shape of drilling tools**: select to display the drilling tools with their real shape, for example countersunk or generally shaped. The selection is managed both when programming by tool and by diameter, with real application compatible with the technology used. If this item is not selected, the drilling tools are displayed with a cylinder.

Graphical elements

- **Dimension of the profile direction arrows**: this option allows to set the length of the two profile direction arrow segments. To set in: [mm] or [inch] (units of measurement for setting parameters). The field values between 0.5 mm and 100 mm.
- **Minimum segment length with arrow display**: this option allows to set the minimum profile section length to show the direction arrow. To set in: [mm] or [inch] (units of measurement for setting parameters). The field accepts values between 0.5 mm and 100 mm.
- **Diameter of profile edge points**: this option allows to set the profile edge points diameter. The representation of the extreme points is subject to the minimum length that a profile segment must have, established at twice the value set here. To set in: [mm] or [inch] (units of measurement for setting parameters). The field accepts values between 0.5 mm and 20 mm.
- **Diameter to draw points and setup of null extension**: this option allows to set diameter for point graph and null extension setup operation. If we have a construct working, the value is moreover used. To set in: [mm] or [inch] (units of measurement for setting parameters). The field accepts values between 0.5 mm and 20 mm.
- **Tool length for setup graph of null overall dimension**: Length of cylinder representation for the setup tool, if an invalid length is assigned of the tool or if no technology is assigned. In case of construct working, the value is not used. To set in: [mm] or [inch] (unit of measure of the configurations). The field accepts values between 0.0 mm and 50 mm.
- **Dimension of the additional graphic elements**: added graphic elements dimensions during the interactive process (drawing, tools). The value is, for example, used in the representation of:
  - bookmarks;
  - profile selection elements;
  - profile bridges;
  - considerable element during the positioning;

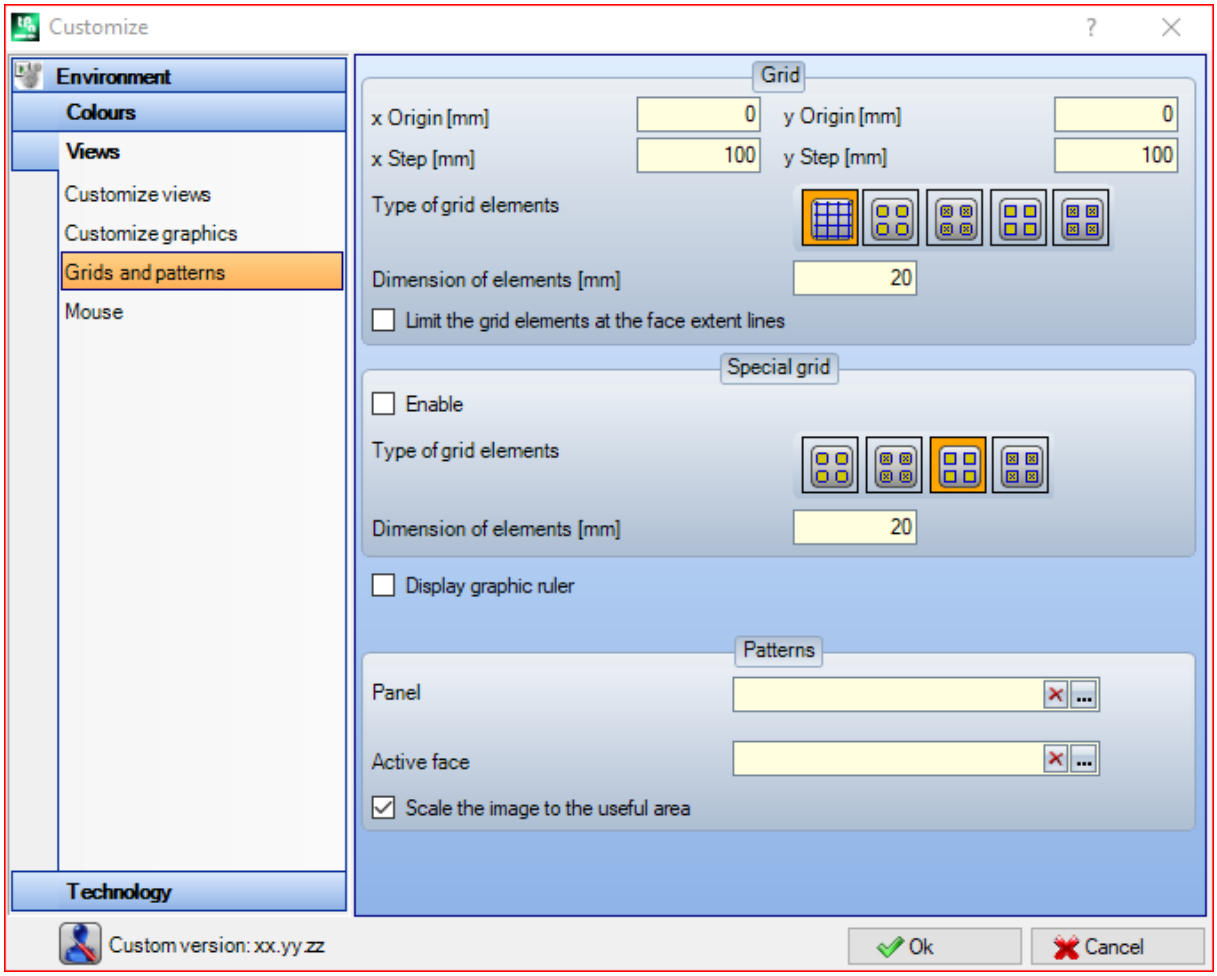
- axis movement arrows.
  To set in: [mm] or [inch] (units of measurement for setting parameters). The field accepts values between 5 mm and 100 mm.

The last group of selections allows the customization of the graphic representation of particular profiles:
- **Geometric profile**: it is a profile that has the **Geometric Profile** parameter selected, in the setup;
- **Emptying profile**: it is a profile that has the **Emptying Profile** parameter selected, in the setup;
- **Construct profile** in the setup, it is a profile that has set the **B field** with null value (closely positive).

For each of this profile typology, it is possible to enable the graphic elements representation:
- edge points,
- arrows,
- 3D graph overall dimension,
- overall dimensions of the profiles.

The default for all the selection corresponds to inactive entries.
A profile can be set of geometric typology to reduce the graphic process. The typical usage is in the development of an ISO curve. If the profile is added through an emptying procedure, it will be set of emptying.

Grids and patterns



Grid
Settings assign an orthogonal Cartesian grid with a grid development in the XY plan of the current face. The grid is not visualized in *Piece Overall view* or with *3D View* active.
- **x Origin**: x origin of the grid. To set in: [mm] or [inch] (unit of measure of the configurations). Default is 0.0.
- **y Origin**: y origin of the grid. To set in: [mm] or [inch] (unit of measure of the configurations). Default is 0.0.
- **x Step**: grid step along the face x axis. To set in: [mm] or [inch] (unit of measure of the configurations). The field accepts a minimum value corresponding to 1 mm.
- **y Step**: grid step along the face y axis. To set in: [mm] or [inch] (unit of measure of the configurations). The field accepts a minimum value corresponding to 1 mm.
- **Type of grid elements**: grid elements display options are listed below:

  - **Lines**: the grid is represented with horizontal and vertical lines, spaced out according to the set steps. The intersection points of the lines are the grid points. This is the default option.
  - **Empty circles**: the grid is represented with empty circles, centred on the grid points
  - **Crossed circles**: the grid is represented with crossed circles, centred on the grid points
  - **Empty squares**: the grid is represented with empty squares, centred on the grid points
  - **Crossed squares**: the grid is represented with crossed squares, centred on the grid points
- **Dimension of elements**: grid element dimension, in the case of circles it assigns the circle diameter; in the case of squares it assigns the square side. To set in: [mm] or [inch] (unit of measure of the configurations). The field accepts a minimum value corresponding to 1 mm.
- **Limit the grid elements at the face extent lines**: select to request the grid element representation inside of the face area. The selection is active if the grid is represented without lines. Even if the entry is unselected, the view over the face overall dimension is limited in any case.

Special Grid
Settings assign a grid directly assigned for single points, as defined by the constructor while configuring the machine. Even the special grid is defined in the xy plan of the current face, but only in the case of faces 1 or 2. Furthermore, the representation is always limited to the inside elements of the face.
This selection is not available in case of *Essential* mode.

- **Enable**: enables or disables the management of the special grid. Default setting is disabled.
- **Type of grid elements**: grid elements display options are listed below:
  - **Empty circles**: the grid is represented with empty circles, centred on the grid points. This is the default option.
  - **Crossed circles**: the grid is represented with crossed circles, centred on the grid points
  - **Empty squares**: the grid is represented with empty squares, centred on the grid points
  - **Crossed squares**: the grid is represented with crossed squares, centred on the grid points
- **Dimension of elements**: grid element dimension, in the case of circles it assigns the circle diameter; in the case of squares it assigns the square side. To set in: [mm] or [inch] (unit of measure of the configurations). The field accepts a minimum value corresponding to 1 mm. The value by default is 20 mm.

- **Display graphic ruler**: this option enables or disables the display of rulers next to the 2D representation or box view of the current face. The ruler is not displayed in the cases of *3D View*.

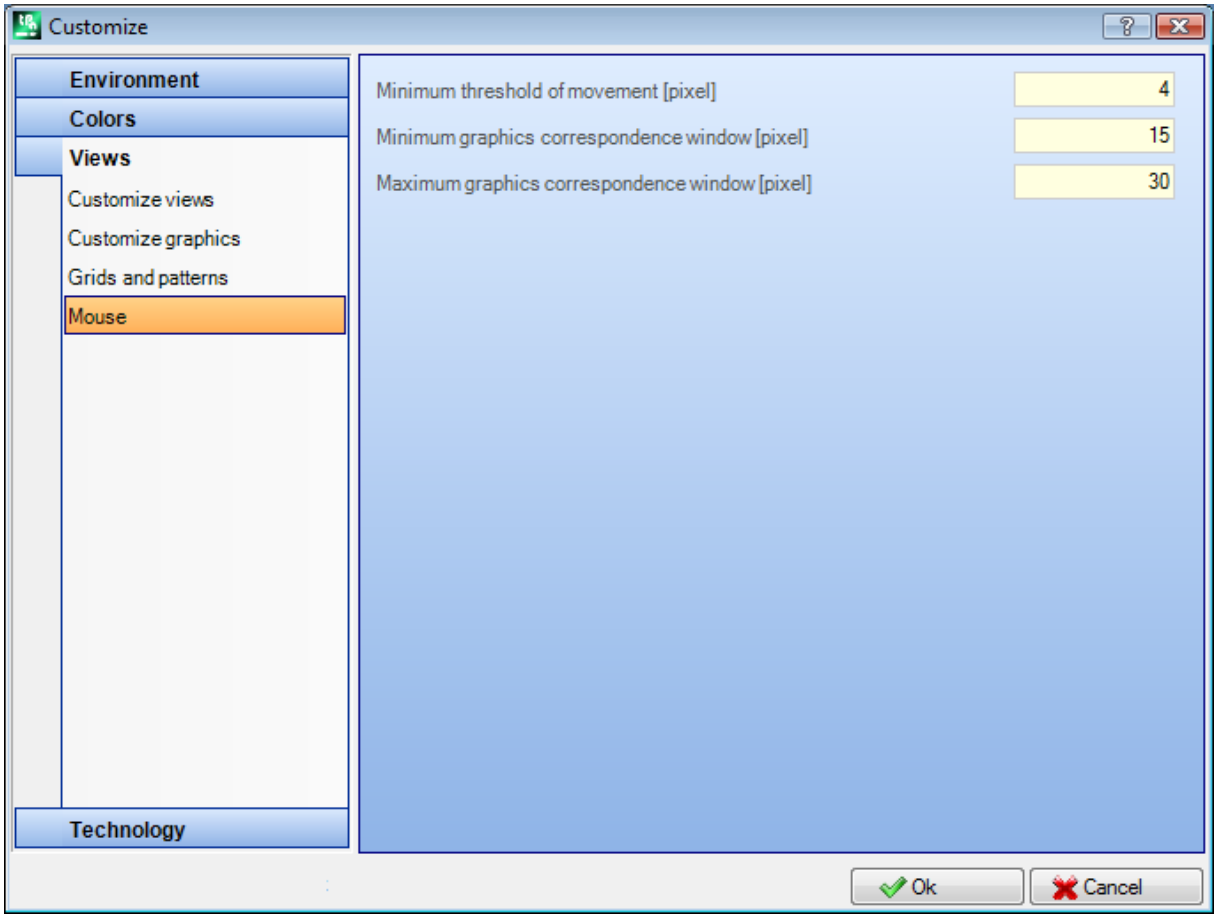
Patterns
A graphic pattern can be chosen to fill the **Panel** and the **Active face**. The pattern name can be edited in the edit box or by clicking on the icon  : a window opens in which the image files stored in the configuration folder (TPACADCFG\CUSTOM\DBPATTERN) appear: the recognized valid formats are *.PNG, *.JPG, *.BMP and you are required selecting a file in the assigned folder. To delete the name of a set theme, click on the icon  .

Some added considerations should be made when using a graphic pattern of panel in the program where a specific horizontal or vertical direction of grain is assigned. If a file with the same name + "_gx" or "_gy" has been found, the program automatically decides which file to load. Use, for example, the file *patternA.jpg*:
- ✓ with file *patternA_gx.jpg* available: *patternA_gx.jpg* will be used with horizontal grain, *patternA.jpg* for the other cases
- ✓ with *patternA_gy* file available: *patternA_gy.jpg* will be used with vertical grain, *patternA.jpg* for the other cases
- ✓ with both files *patternA_gx.jpg* and *patternA_gy.jpg* available: *patternA_gx.jpg* will be used with horizontal grain, *patternA_gy.jpg* will be used with vertical grain, *patternA.jpg* in case of no assigned grain.

Instead, if only *patternA.jpg* file is available:
- ✓ *patternA.jpg* will be used unchanged with horizontal grain or without assigned grain
- ✓ *patternA.jpg* will be used rotated 90° with vertical grain.

- **Scale the image to the useful area**: this option specifies how to position the graphic patterns. When the item is enabled, it adjusts the image in the usable area (panel or active face), otherwise, it reproduces the flanked image until the usable area is filled.

Mouse



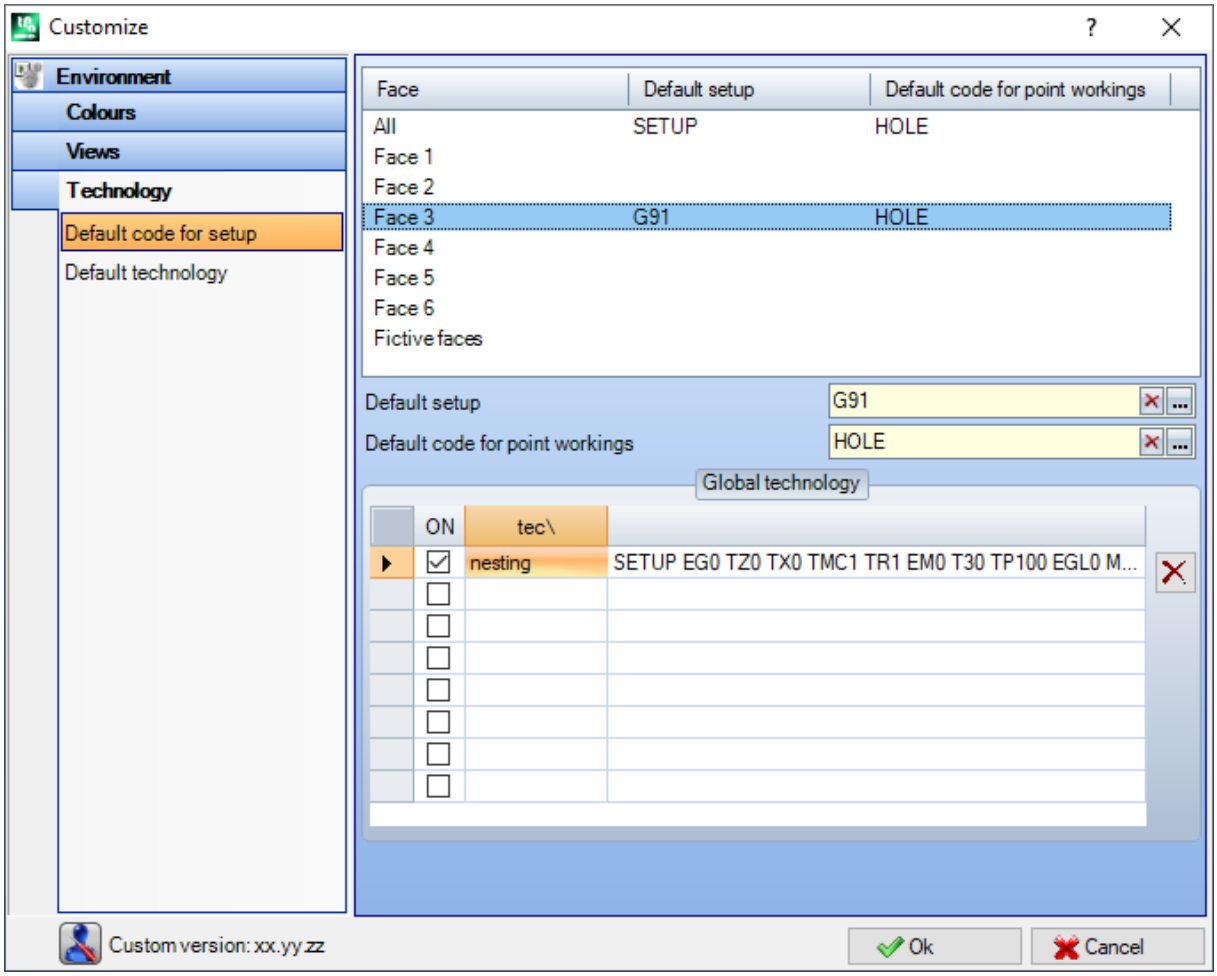
- **Minimum threshold of movement [pixel]**: distance in pixel to be covered by the pointer on the screen before intercepting a status change. This setting allows you, for example, to avoid unwilling rotations of the piece This setting is also used in the interactive acquisition (for example in the draw functionality) as a filter to activate the search of a snap entity. The default value is 2 and accepts values between [1; 10].
  - **Minimum graph correspondence window [pixel]**: minimum dimension of the window for the search of graphic correspondence. The default value is 10 and accepts values between [1; 50].
  - **Maximum graph correspondence window [pixel]**: maximum dimension of the window for the search of graphic correspondence. The default value is 20 and accepts values between [1; 50], but not less than the previous field.

The two values size the graphic search area, that is applied in the graphic acquiring (searching the snap entity or the current working). With reference to the example in the paragraph *Insertion of Geometric Entities From Drawing Menu*, setting 10 and 20 values for the two fields), no more than 2 attempts of graphic search are executed: the first one within a 10 pixel area, centred on the mouse position and the second one within a 20 pixel area. If the values (7 and 20) are set, no more than three attempts are executed in areas measuring respectively 7, 14, 20. The progression of areas occurs according multiples of the minimum dimension.

13.4 Technology

The data displayed in the group pages can be changed only with work program closed and in the *Machine* operating environment.

Default codes



In the grid the technologies used for point and setup workings are assigned when a default assignment is required.
Usage examples of these assignments are:

- the execution of tools that required the insertion of a setup when a new profile is opened;
- the application of complex workings as text generation or emptying;
- the insertion of geometric element *Point* from Drawing menu;
- the execution of opened or headed with setup geometric code profiles or point geometric code workings (that, for examples, come from an import process of external format).

How to proceed to set the technology:

- select the row of the face for which a technology must be set

Face	Code
All	It sets a common technology for each face. It is applied when a face has not his own technology set.
Face 1	set a technology for the face 1 (top) and, if it is necessary, for the fictive and automatic faces that check a similarity criterion
...	
Face 6	set a technology for the face 6 (queue face) and, if necessary, for the fictive and automatic faces that check a similarity criterion
Fictive faces	set a general technology for all the fictive and automatic faces or only for the faces that do not check a similarity criterion with one of the six real faces

- in the grid, choose the face line on which to work. Clicking on the icon ... of one of the two fields below the grid, a window is opened where it is possible to choose the workings among those that are available and set the technological data on it. From the list of the workings that can be selected, they are excluded:
  - those that are unavailable in the working palette;
  - those with polar schedule.

Furthermore:

- for a Setup: only the setup workings are listed;

- for a Point: the point and setup workings are listed.

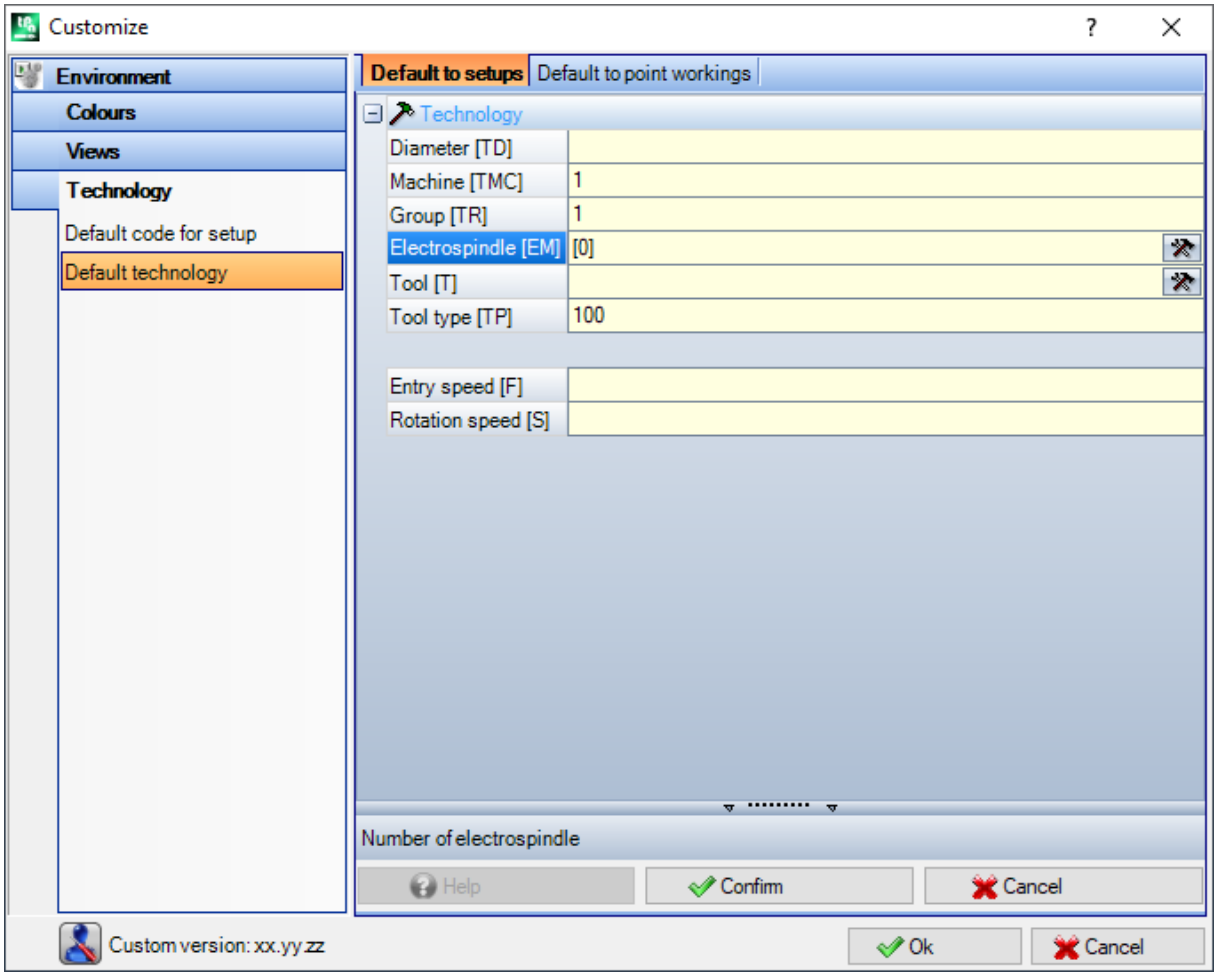
The dimensional technological parameters (coordinates and speed) must be assigned with particular care, because they must be set according to the measurement unit defined in the configuration ([mm] or [inch] for the coordinates; [m/min] - [mm/min] or [inch/sec] - [inch/min] for the speeds. Assignments can be set in numerical or parametric format: in each case a possible error of parametric schedule is reported. In a punctual working by default the parameter Diameter is assigned according to the following rules:

- if the working of point geometric code doesn't have a set diameter value, a replacement **is done**;
- if the working of point geometric code has a set diameter value, a replacement **is not done**;

Click the  icon to delete a setting in the main grid about the default code box.

The table of **Global technologies** assigns up to 8 significant technologies for setup workings that can be used in parametric programming form. In the table each row can assign a setup with the usual procedure considered for the application of technological setup.
To enable a setup, check the box in the ON column and assign a symbolic name to the setup that will be employed in the programming; as shown in the column heading, the type of the recognized parametric form is "tec\namesetup". For each enabled setup, the name is obligatory and must be univocal.
To change the setup technology assignments, a double click (or F2) the cell to the right of the affected row opens the assignment window.
To disable an already assigned setup, you must uncheck the box in the ON column.
A *global technology* can be used in all processes including the possibility of assigning a setup technology using a Name, intended as a field NAME of a programmed setup: now the setup is recalled by mean of a parametric name and does not require any programming.

Default technology



The tab is shown only if the working used to assign the default technology is in the workings database. The choice of the technology is not different for face and it is about the technological assignments of a general point or setup working.

It is possible to set the technological parameters for:

- **Default to setups**: it is about the setup workings and the complex codes of profile typology (codes that have a working Sub-typology set in the workings database that is equal to 1; for examples the workings: Pocket, Door).
- **Default to point workings**: it is about the point workings and the complex codes of profile typology (codes that have a working Sub-typology set in the workings database that is equal to 0; for examples the workings: Drilling Fitting, Distribution of the holes on the circle).

Window settings are changed when it is required changing the default assignment of a parameter with respect to the way as it was proposed during the insertion, perhaps making it not editable.
Set, for example, the value = 1 to the Machine parameter for the setup technology:
- for each new setup working insertion (or: Pocket, door), the machine field will be proposed set to 1 moreover editable;
- the setting doesn't change the schedule or the interpretation of the already inserted workings.

If it is required forcing the schedule of the parameter Machine = 1 (for example: because in the application is managed only the Machine 1), in the widow for the Machine, it has to be set:
- "(1)": value closed in round brackets or "v,1". This notation allows that the Machine value is always 1 and that the parameter can be seen in insertion/edit of the setup working (or: Pocket, door), but it's not editable (the "v" notation is for "view");
- "[1]": value closed in square brackets or "h,1". This notation allows that the Machine value is always 1 and that the parameter is not visible in insertion/edit of the setup working (or: Pocket, door), but it's not editable (the "h" notation is for "hide").

Both settings change the programming or the interpretation of already programmed workings; in the example: the Machine field is always forcedly assigned 1.

Other valid settings are:
- "()": no value in round brackets or "v". This notation allows the parameter to be seen into the window of working insertion; but it cannot be changed, and it cannot have any assigned setting.
- "[]": no value in square brackets or "h". This notation allows the parameter to be hidden into the window of working insertion; but it cannot be changed, and it cannot have any assigned setting.

This forced setting of the parameters has to be used to assign particulars of the plant technology, for example a plant
- composed by a single **Machine** and/or by a single **Group** and/or by a single Milling Cutter **Tool**;
- that excludes the schedule of the **Electrospindle**.

WARNING:
- Those parameters, already defined as not changeable parameters in the working database, cannot be changed.
- Only the parameters can be assigned: Machine, Group, Spindle, Tool, Tool type, Diameter, Speed and Rotation tools.
- A set value can be expressed in numerical or parametric format: in each case a possible error of parametric schedule is reported.
- The settings assigned here are not integrated with those relating to the previous page **Default codes**.

13.5 Customize the "prototype" file

As already told, the creation of a program uses as a starting point a default prototype file.

To open and modify the prototype file, select the command Open the prototype file  from the menu Application: a PIECE.TCN file opens, in the folder TPACADCFG\CUSTOM.

According to the configuration of TpaCAD the user can manage different prototypes for each assignable typology: program, subroutine or macro program. In this case, the selection of the command leads to a following selection of the concerned file, as is already the case for the creation of a new program.

More specifically, you can assign to a prototype file a non-minimum access and/or writing level, in order to avoid unauthorized modifications. Anyway, the access and/or writing level assigned to the new program appears to the minimum required for the type of the piece, that in the case of program, corresponds to the *Operator* level.

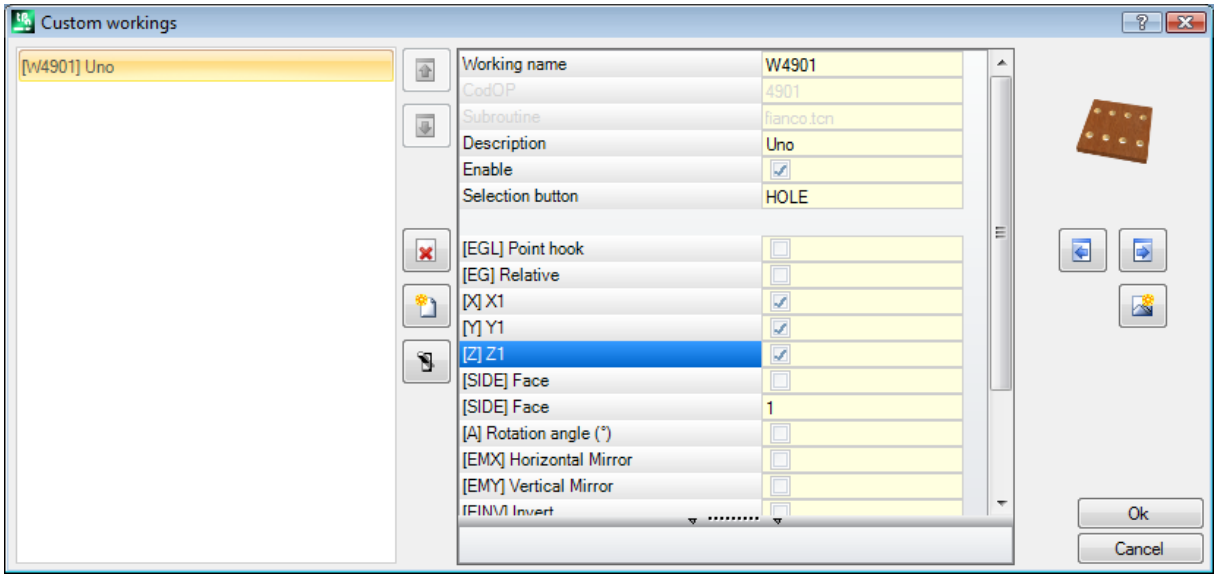
Furthermore, the prototype for the creation of a piece of program typology can assign a different typology, for example, a subroutine.

14 "Client" workings creation

A "client" working is always a complex working. It is an aggregation of workings that has the objective of hiding the complexity of the working to the operator, simplifying the selection and the assignment of parameters and properties.

The installed working database with the TpaCAD application program makes available a lot of codes of complex workings that are usually based on a macro.
The considered feature implements the possibility to assign complex codes even for the end-user, basing on subroutines written by the same user. It is possible to enable up to 100 "client" workings.

The command **Custom workings** for the definition of a complex working is selected by the menu with the program closed. The command is not available on the menu if the working database does not assign the reference working (sample working) that is needed to create custom workings.



To create a complex working, select the button : Open Piece to select the subroutine to recall.
By accessing the Manufacturer level, a macro-program type file can also be selected.

- **Name of the working**: name of the working. The default name is displayed in the form W + (opc), where "opc" stands for operating code. This last is automatically assigned to the working (the first free code among those available for this set of workings). It is possible to define a set of alphanumeric characters ranging between 2 and 10. The first character shall be a letter. The literal names of 2 characters that start with the "W" letter are considered reserved names and so they cannot be used (examples: "WC", "WB"...), because they are reserved for internal usage. The chosen name cannot be already defined for another working or for a parameter of the same working, even for those set for the variables which can be reassigned. The name ASCII of the working heads a program row in the ASCII Format list. The field setting is obligatory.

- **OpCode**: operative code of the working. The field value is automatically assigned and cannot be changed.

- **Subroutine**: Returns the subroutine that is applied by the processing, indicated here as an extension.name. In the text area of Help at the bottom of the control, appears the whole path of the file. The field is not editable.

- **Description**: description name of the working. The field is initialized with the name of the subroutine (e.g. W4901), it has a length of maximum 30 characters, and it is not inserted in the file language, thus it cannot be translated.

- **Enable**: if the field is selected, it inserts the working in the working palette. Even if it's not enabled, the working assignment has to be totally valid.

- **Selection button**: it shows the group name of the workings where the working is inserted. Clicking the buttons and the picture to select is updated, with Groups flip of the Workings tab. The custom working is added to the working group chosen.

 this button allows the user to assign a new group to insert the custom working. By selecting the button, you open the folder to search the image to be associated to the new group. The name that identifies the new group is automatically assigned.

WARNING: The working group selection buttons do not appear, if the palette of the graphic selection is assigned on one directly exploded group.

Typical parameters of a subroutine call code can also be set (as, for example: Point hook, Relative, Positioning coordinates, Application Face, Rotation Angle, Horizontal Mirror and Invert) in the cases in which:
- they are set by the machine constructor in the sample working;
- there are no more than 30 parameters;
- for each parameter is reported the ASCII name, inside the square brackets.

Select the field to enable the view and the following management of the parameter in the working window.
A particular parameter is that of the **Face** of the subroutine that has to be applied, for which is assigned a check box and an edit field:
- Select the field to enable the view and the following management of the parameter in the working window.
- the edit field assigns the default value of the parameter, that is directly editable only if the parameter is directly managed. Specific functioning cases are distinguished:
  - to force the application of a specific face of the subroutine: leave the parameter not enabled and assign the face number in the edit field (for example: 1);
  - to force the application of induced calls: leave the parameter not enabled and leave the edit field empty or with assigned value 0 or -1;
  - to leave the availability of both the previous cases: set the enabled parameter and leave the edit field empty or with a beginning default value, but moreover editable.

- **r Variables**: it shows the variables of the subroutine that can be reassigned, they become parameters of the complex working. For each variable it is assigned: ASCII name, description, enabling status, default value, entry typology of the field.
  The first 50 variables of the subroutine that can be reassigned are considered.
  - **Name**: ASCII name of the variable. From 1 to 10 alphanumeric characters can be set. The first character must be a letter. The literal names of 2 characters that start with the "W" letter are considered reserved names and so they cannot be used (examples: "WC", "WB"…), they are reserved for internal usages, besides the names of the parameters already assigned in the sample working (for example: Point hook, Relative…) and the ASCII name of the same working.
  - **Description**: descriptive name of the parameter (for example: "Offset x"). The field is initialized with the symbolic name of the variable or, if this last is not assigned, with the description of the variable or, if not assigned, as R+(nn), with nn = number of the variable (Example: "R0", "R27"). The field has a maximum length of 30 characters and it is not inserted in the file of language, therefore it cannot be translated.
  - **Enable**: if the field is selected, it enables the direct setting of the field. If it is not enabled, the field assignment corresponds to the field **Value**, without edit possibilities.
  - **Value**: default value proposed during the working insertion. The field is initialized according to the value set in the subroutine. If the option **Set check box** is enabled, a check box is shown instead of the edit field of the value. It is possible to assign a parametric value.
  - **Set check box**: it shows a check box instead of the edit field to assign the value.

If the **Name** of a r variable is assigned equal to the name used to identify a technological field of electrospindle or tool, the variable will be automatically associated with the possibility to start the opening of the technology window, with the possibility of interactive selection of the value.

If it is necessary to update the complex code assigned to a subroutine, select the command  . The same result can be obtained by selecting the command  for the concerned subroutine. In this way the application program:
- it checks that an already defined code is assigned for the subroutine;
- it recoveries and checks the already set info;
- it proposes the resulting settings.

To complete the management of a custom working it is required arranging:
- a picture file to load during the composition of the Working tab (the assignment of the file is needed). The file has to be stored in the folder TPACADCFG\CUSTOM\DBBMP, with the name composed by "W" + (operative code) or, alternatively, with the same name assigned to the working and in the formats of the known pictures (*.png; *.jpg; *.bmp);
- a picture file to load as contextual graphic help during the working assignment (the file assignment is not needed). The file has to be stored in the folder TPACADCFG\CUSTOM\DBBMPHLP, with name and format assigned as for the previous point.

To delete a custom working that is already in the list, bring the selection on the working and select the button  .

To visualize and check the data entry of the created working, select the button  .

The buttons  and  move the selected working, defining a different presentation order in the group of the corresponding workings.

15 Conversion Program

15.1 From DXF to TpaCAD format

The standard installation installs one import module from the DXF format: **TpaSpa.DxfCad.v2.dll**.

- **DxfCad**: available from the version of TpaCAD 1.4.2.

For the documentation, please read the manual of the importer.

15.2 From TpaCAD to DXF format

The standard installation installs one export module to DXF format: **TpaSpa.DxfCad.v2.dll**.

Program in DXF format only assigns workings that have verified the logical conditions, as assigned in the workpiece stored in TpaCAD environment. System logic workings are excluded from conversion (cycles IF…ELSE… ENDIF, ERROR, EXIT, assignments of J variables). Complex workings (of profile, application of subroutines or of macros) are exploded and each parametric assignment is solved and replaced with numerical setting.

Parameters

- **Piece layer**: Name of the layer assigned in the DXF file to identify the geometry for the overall dimension of the piece. In the DXF file a rectangular polyline is generated (length by piece height):
  ✓ In case of 3D export: the polyline is assigned a thickness corresponding to the piece thickness. The set name is assigned to the polyline layer.

- **Layers**: This group indicates the names of the layers assigned to the machining types, when no specific assignments are applied (see later: setting page **Workings and Layers**). The settings of the group cannot be changed.

- **Create individual geometric elements**: select to create geometric individual elements, not in polyline, each with height corresponding to the final Z dimension of the segment (linear segment or arc). It corresponds to the default operation. If not selected:
  ✓ A profile whose constant depth generates one only polyline.
  ✓ A profile whose variable depth creates a polyline for each element (line or arc), whose Z position is assigned at the final depth of the element itself

- **Field separator**: this option selects the character to be interpreted as a separator between the fields. You can select the characters ‘ ’ # _ % - + (no characters, hash, underscore, percent, minus, plus).

- **Calculate 3D view faces**: it selects the field to enable the export of a 3D piece. Default is active. If the field is selected:
  - The 6 basic faces and the piece-face are treated for the operations that are assigned to the basic faces. The obtained DXF file corresponds to a 3D drawing, whose three faces are converted into a Cartesian coordinate system and where the programming of the depth Z- axis is managed both for the positive and negative values.

  If the field is not selected: only face 1 is considered and the DXF file corresponds to a 2D drawing.
  - **Piece layer**: select to export the piece layer in a 2D drawing. If the field is not selected: it exports only the entities corresponding to workings (on face 1).

Workings and Layers

This page allows associating the layer of a DXF entity with an original working.

In the page you can assign up to 40 associations for setup or point workings.

In the DXF file the layer is typified by a name of 100 characters:

- The first characters, which we call prefix, are linked with a working of those classified point or setup that are available in the TpaCAD working database;
- the remaining characters are linked with the parameters and/or properties of the working.

Working prefixes

Prefix:

- It is made from 2 to 30 alphanumerical characters (the first character cannot be numerical);
- it cannot be repeated in the table.

You can link to each assigned row some indicators for the parameters and/or the TpaCAD working properties in the two following tables.

The list of the information considered here corresponds only in part to that directly managed in the assignment of the working in TpaCAD: fields deriving from the *compilation* of the workings are now added. One example is the technological information for a milling cutter Setup:

- in TpaCAD it is typical to program the work tool by assigning machine, group, electrospindle and tool;
- In the information now available there is also the tool diameter, as acquired from the plant technology.

Parameter prefixes

The indicators of the parameters and/or of the properties are assigned with a single alphabetical character. For each prefix up to 30 total indicators of parameters and properties are available.

In the event of a parameter that can be directly set in the assignment of the working in TpaCAD, the table shows also the message describing the parameter itself.

The assignment of a non-numerical parameter has no effect on the conversion.

Programmed Workings

The workings examined by the export module are listed below.

Point workings (operation code between: 1-1000)

For each point working we have:
✓ a geometric circle, if the tool diameter is non-null;
✓ a geometric point, otherwise.
The height of the geometric entity corresponds to the programmed Z coordinate.
In the event of a non-configured working in **Layers and workings**, the assigned layer is "BOR".

Setup workings (operation code between: 1-1000)

A polyline, whose height corresponds to the Z setup coordinate, corresponds to each non-isolated setup working.
A particular case is a profile that corresponds to a sawing work (blade setup followed by a linear stroke): you can and it is generally appropriate to configure a dedicated layer in order to differentiate the sawing work from a differently assigned profile.

In the event of a non-configured working in **Workings and Layers**, the assigned layer is "ROU".

If the setup is alone, it is converted:
✓ a geometric circle, if the tool diameter is non-null;
✓ a geometric point, otherwise.
In the event of a non-configured working **Workings and Layers**, the assigned layer is "SET".

Profile working of linear typology

For each linear working corresponds to a line in the polyline.
The assigned layer is the same that is interpreted for the Setup working ("ROU", in the event of a non-assigned linkage).

Profile working of arc typology (xy plan)

An arc in the polyline corresponds to each working of arc type.
If it is a circle: if it is alone, it generates a circle; otherwise, it converts in the polyline in two semi-circles.
The assigned layer is the same that is interpreted for the Setup working ("ROU", in the event of a non-assigned linkage).

Profile working of arc typology (no xy plan)

An arc, assigned in a plan different than the (xy) plan of face, has to arrive to the converter exploded in a series of linear segments.
Each segment of the series of linear segments is converted in line of polyline and the considerations described above are valid.
If an arc assigned in a plan different than the xy plan arrives to the converter, it does not translate the segment.

15.3 From ISO format to TpaCAD format

Here the functioning specifications of the importation module from ISO format which is included in the standard installation are described.
The conversion procedure must be obviously enabled by the machine constructor during the machine configuration.

The available settings for the conversion are assigned in a dialog. The manufacturer decides on the level access to the dialog.
Before examining in detail the available settings, let us see which criteria were adopted during a file conversion in a DXF Format.

Profile and drilling workings are converted and assigned to the only face 1.
Here it is displayed a valid ISO file fragment, where the fields that are interpreted are shown in boldface:

(FLAT 20MM 2F EC HSS)
G71
G0 X-627.857**Y**0**Z**312.249 **B**13.135 **A**0 **S**12000 **T**4;… (remark)…
G40
G1 **X**-2.272**Y**0**Z**-9.738 P0.22724Q0R0.97384 F6000 T1
G1 **X**888.346**Y**0**Z**-217.56 P0.22724Q0R0.97384 T1 B13.134
G1 **X**898.083**Y**0**Z**-219.832 P0.22722Q0R0.97384 T1 B13.134
…
M2

The file is considered valid if the first line starts with one of the characters: **%** (percentage), **(** (open bracket), **;** (semicolon), (colon), **[** (open square bracket), **/** (slash), **O** (letter "O"), **P** (letter "P"),**G** (letter "G"), **N** (letter "N"), **M** (letter "M"), **T** (letter "T"), **S** (letter "S").

The ISO format interpretation is not case-sensitive; for example, "g10" is the same as "G10".
- The first string between round brackets, read before than an important instruction, assigns the program comment (in the example: (FLAT 20MM 2F EC HSS);
- rows that start with the character **%** (percentage), **(** (open bracket), and **;** (semicolon) are not interpreted;
- the character **;** (semicolon), that is in a row of the file makes the comment the part of the row that follows;
- the default unit of the file ISO is [mm]. To directly assign the schedule unit and, if possible, the piece dimensions it is needed the assignment of the field **G70**/**G71** before of the first **G0** (and not in rows that are not interpreted):
  "**G70X**20**Y**12**Z**3.9" sets the file ISO unit in inches [inch] and the piece dimensions (length = 20 inches; height = 12 inches, thickness 3.9 inches);
  "**G71X**1300**Y**1300**Z**80" sets the ISO file unit in [mm] and the piece dimensions (length = 1300 mm; height = 1300 mm; thickness 80 mm)

In the example we find **G71** on the second row, without the setting of the piece dimensions. In this case the same are automatically assigned, including the positive overall dimension on all the coordinate axes.

The profile interpretation starts with a row **G0** (rapid movement. In our example it is the third row) and on this row interprets the fields:
- (**X**, **Y**, **Z**) as initial coordinates of the profile;
- (**B**, **A**) as initial values of the rotated axes (they are then displayed on the profile setup, if it is assigned with rotating axes)
- **G90**/**G91** to program absolute/incremental coordinates;
- **T4** tool selection
- **S12000** spindle rotation speed

On the same row of G0 or in the next one, it can be interpreted the specification regarding the mill-radius compensation:
G40 no correction (default);
G41 on the left of the profile;
G42 on the right of the profile.

Each row following the first one of the profile can assign:

1. a segment of **G1** linear interpolation and the fields are interpreted (**X**, **Y**, **Z**) as final coordinates of the linear segment and **G90**/**G91** to program absolute or incremental coordinates. A non-assigned coordinate is propagated from the previous segment. It can interpret an interpolation speed in the **F** field (unit: [mm/min] or [inch/min]) that is converted in programming unit as assigned in the configuration of TpaCAD.

2. a segment of **G2**/**3** circular interpolation (respectively: clockwise/anticlockwise) and the following fields are interpreted:
   (**X**, **Y**, **Z**) as initial coordinates of the curve segment;
   G17/**G18**/**G19** to program the arc development plane (respectively: XY (default), ZX, YZ)
   (**I**, **J**, **K**) as coordinates of the centre. The 2 coordinates that correspond to the plane of the arc are significant (in relative programming, or as a result of the programming of **G90**/**G91**)
   G90/**G91** to program absolute (default) or incremental coordinates.

   It can interpret an interpolation speed in **F** field.
   In the case of arc in the plan ZX (G18), if the TpaCAD application program solves the arcs on the Xz plane, the direction of rotation of the arc is inverted.

   In the case of arc in the XY plan (G17 = default), if both the coordinates of the centre (I, J) are not set and if the arc does not solve the circle, it solves an arc with radius schedule (**R.**). The radius has to have a value that is the same as (epsilon*10.0) and moreover it cannot be less than the distance among the edge points of the arc, otherwise the conversion is stopped for an error situation.
   If in the ZX plane or in the YZ plane both the coordinates of the centre are set, the conversion is stopped because this is an error situation.
   In the case of profile lines (G1, G2, G3) that do not have a correspondence at the beginning of the G0 working, the conversion is cancelled.

3. a new profile starts **G0**

4. A hole interpretation interprets a code **G81** and on this row it interprets the fields:
   - (**X**, **Y**, **Z**) as initial coordinates of the profile;
   - **G90**/**G91** to program incremental/absolute coordinates;
   - **T4** tool selection;
   - **S12000** spindle rotation speed;
   - **F100** tool entry speed.

5. possible lines with other codes **G** are ignored.

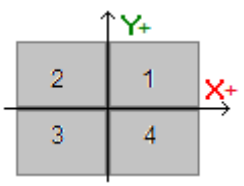
The interpretation of the program ends at the end of the file or if the **M2** field is interpreted.

Settings

Let us see now the settings available in the process of the import module:

- **Dimension G code**: set the G code to which the interpretation of the piece dimensions should correspond. We have already mentioned the codes G70/G71, whose interpretation is active to set units of measurement and dimensions. You can assign here a different code with a valid value between 100 and 10000.
- **Drilling G code**: set a value between (81-89) to interpret as a drilling working.

- **Quadrant of the machine**: set a value between 1 and 4 to interpret the XY coordinates read in the ISO file. A setting different from 1 corresponds to the interpretation of an ISO file in the machine coordinates. With reference to the drawing:



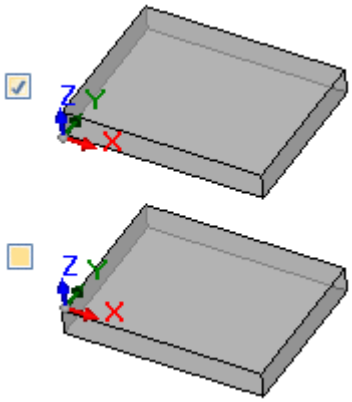
- **1** corresponds to the default situation for which the coordinates are not changed;

- **2** corresponds to the situation of X coordinates in negative area of the machine: the import changes the X positions by taking them back to the positive area;
- **3** corresponds to the situation of both (X,Y) coordinates in the negative area of the machine: the import changes the X and Y positions by taking them back to the positive area;
- **4** corresponds to the situation of Y coordinates in machine negative area: the import changes the Y positions by taking them back to the positive area;

Work positions are changed according to the dimensions read (or deduced) for the file.

- **Rotating axes assigning (B,A)**: this setting concerns reading ISO curves straight from the program and, specifically, the interpretation of the rotating axes. The selection is from a three-option list, and indicates the couple of rotating axes that are considered to assign the axes (B, A):
  - (B, A): the assignment uses the same names
  - (A, C): the A axis of the ISO curve assigns B, the C axis of the ISO curve assigns the A axis
  - (B, C): the B axis of the ISO curve assigns B, the C axis of the ISO curve assigns the A axis.

- **Absolute Z axis reference system**: select the field to interpret the Z coordinates in the absolute reference system. If not, the Z coordinates are directly interpreted in the 1 face system.
  When the selection is active, the import modifies the Z positions by taking them back to the 1 face.



- **The coordinates of the centres apply G90/G91**: select the field to enable the interpretation of the centres according to the codes G90/G91. Otherwise, the coordinates of the centres are always interpreted incremental with respect to the start point of the arc.

- **Delete the isolated G0 codes**: select the field to delete from the import those g0 codes that do not continue with movements on a linear/or curve trajectory. They are, in general, fast positioning processes carried out over the piece at the maximum speed permitted for the axes; they correspond to zero positions of the piece, disengagement, tool change and are not useful to interpret the machining operations on the piece.

- **Mill Setup**: settings regarding the conversion of a vertical Mill Setup. The case corresponds to a profile with no rotating axis assignments
  - The first field corresponds to a list of selections, corresponding to the workings of available Setups
  - The second field allows the direct assignment of parameters for the selected working.
    - an example of setting is "TMC = 1 TR = 2", corresponding to the assignment of the technological parameters of the Machine (1) and Group (2)
    - the setting can only use the ASCII names of the parameters and the value must be separated from the '=' and numeric character
    - the fields that correspond to the parameters already managed independently by the import module (application dimensions, rotary axes, RPM, cutter radius correction) are however ignored and excluded
  - it is possible to select a working of vertical or oriented Setup

- **Mill Setup (oriented)**: settings regarding the conversion of a vertical Mill Setup oriented. The case corresponds to a profile with assignments of rotating axes. The assignments are completely analogous to the previous case.

15.4 From TpaCAD format to ISO format

Below we will see the functioning specifications of the exportation module to ISO format, which is included in the standard installation (**TpaToIso**).
The conversion procedure must be obviously enabled by the machine constructor during its configuration. The conversion can only be applied to programs or subroutines.

The conversion is applied only to programmed workings on face 1 (directly or from face-piece) or, if the face 1 is not managed, on face 2 (directly or from face-piece), and to drilling workings and profiles.

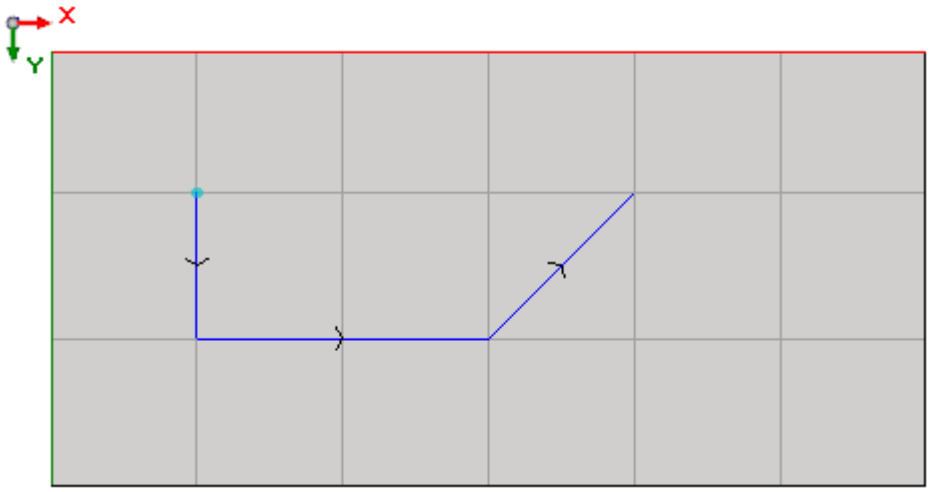
The program in ISO format only assigns workings with logical conditions checked, as assigned in the piece filed in TpaCAD environment. Moreover, the conversion does not include system logical workings (cycles IF.. ELSE.. ENDIF, ERROR, EXIT, assignments of J variables). Complex workings (of profile, application of subroutines or of macros) are exploded and each parametrical assignment is solved and substituted with a numerical setting.
The conversion module loads the technology parameters, in order to acquire information that can be useful to the composition of the final file. In case valid technologies are not verified, there will be no warning messages.

Settings

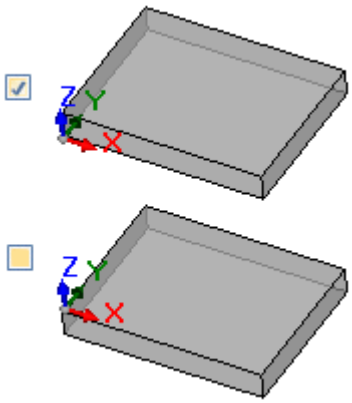
- **Absolute XY axis reference system**: select this field to bring the XY coordinates to an absolute frame of reference. The selection concerns only the coordinated XY axes and it applies to cases where TpaCAD manages a XY system of face 1 different from the standard. In this case:
  - with active selection, the XY coordinates are converted to the absolute frame, with origin of the axes on the bottom right. In the conversion, pieces of information such as: rotation direction of the arcs and the correction side of a profile are subject to modification as well
  - with selection not active, the XY coordinates stay unchanged.

With reference to the figure:

- the programming origin is the top-left corner
- the profile is programmed in:
  - SETUP X100 Y100
  - L01 Y200
  - X300
  - X400 Y100
- we can see how the profile is with conversion to the default XY frame (origin in bottom-left corner)
  - SETUP X100 Y200
  - L01 Y100
  - X300
  - X400 Y200



- **Absolute Z axis reference system**: select this field in order to transfer Z coordinates to an absolute frame of reference, instead of face 1. Selecting this option, all Z coordinates are added to the piece thickness. The picture shows the two frames of reference:
  - **absolute** (top of the picture)
  - **face** (bottom of the picture)



- **Drillings**: a drilling working can be translated either with a corresponding fixed-cycle G code or a milling cycle
  - **Perform milling cycle**: select the field to translate all drillings with milling cycles (together with the detail of the entry and exit movements into and from the piece)
  - **Drilling G code**: set a value within the interval (81-89) to be used as drilling cycle

- **Milling: solve entry/exit**: select this field to translate a milling detailing the entry and exit movements into and from the piece

- **The coordinates of the centres apply G90/G91**: select the field to translate the coordinates of the centres referring to the codes g90/g91. Otherwise, the coordinates of the centres are always translated as incremental towards the arc starting point.

- **Safety Z**: set the safety Z coordinate for movements over the piece. The programming unit is [mm], in range [10.0; 1000.0]. The value sets a position on the Z axis in an absolute frame of reference.

- **Measure over the workpiece**: set the default Z coordinate, to be used in case of non-assigned technological value. The programming unit is [mm], in range [2.0; 500.0]. The value sets a position of Z axis on a face 1 system: value 5.0 corresponds to a position on Z axis of 5.0 mm over the piece.

- **Header rows**: it is possible setting up to 5 rows on the program opening, before the lines related to the workings. The first row of the created file is in any case assigned as "%0".

  An example might be represented by the row assigning the system unit and the original piece size "**G7%u X%l Y%h Z%s**", where some parametrical forms are used (syntax: "%" + letter):
  - "%u" is substituted by the character corresponding to the unit of measurement: "0" if (inch), "1" if (mm)
  - "%l" is substituted by the piece length
  - "%h" is substituted by the piece height
  - "%s" is substituted by the piece thickness

  The other parametrical forms managed are:
  - "%o#" (with #= 0..7) is replaced by the variable (o0,..o7) of the piece
  - "%v#" (with #= 0..7) is replaced by the variable (v0,..v7) of the piece
  - "%n" is replaced by the value corresponding to the piece execution mode: 0=normal execution, 1=mirror x, 2=mirror y, 3=mirror xy
  - "%x", "%y", "%z": are replaced by the coordinate (X, Y, Z) of the work area stroke, as assigned in the piece.

- **Footer rows**: it is possible setting up to 5 rows on the program closing, after the lines related to the workings. In any case, the line "M2" is added at the end of the created file.

Syntax and Examples

Below you will see an example of created file:

%0
(TpaToIso By TPA Srl)
G71X800.0Y450.0Z80.0
...
G90G40
G0X100.0Y-65Z-12.5 A10B-60 T4M12S12000
G01X250 F4000
G02G17X...Y...I...J...
...

M02

- **%0**: fixed header row
- (TpaToIso By TPA Srl): unvarying comment row
- **G71**X800.0Y450.0Z80.0: unit of measurement and piece dimensions: G71 SI units (mm, mm/min), G70 imperial units of measurement (inch, inch/min)
- … (more header rows)
- **G0**…: instruction rows
- …
- … (footer rows)
- **M02**: program closing.

Milling

A profile translation is determined by the option
- **Milling: solve entry/exit**

Below we shall see how a setup working is translated in case the field is not selected:

- **G90G40**
  - G90 absolute programming (all is translated into G90)
  - G40: item related to the correction of the milling radius
    - G40 no correction (default)
    - G41 left of profile
    - G42 right of profile

- **G0**X100.0Y-65Z-12.5 A10B-60 T4M12S12000M3F3000
  - a setup working is translated with G0 code:
  - X… Y…Z… axis coordinates
  - A… axis rotating around X
  - B… axis rotating around Y
  - T… spindle or tool selection (if strictly set positive)
  - M… translates the M field of the working (if strictly set positive)
  - S… spindle rotation speed (if strictly set positive)
  - M3/M4 spindle rotation (M3=clockwise, M4=counter-clockwise)
  - F… entry speed ([mm/min], [inch/min]) – (if strictly set positive)

The **M** field may be used to enable an ancillary function.
The **S** and **F** fields present the value that was originally programmed, or assigned in the tool technological parameters.
The data in **M3**/**M4** on spindle rotation shows the value assigned in the tool technological parameters.

Let us see now how a setup working is translated if we select the option **Milling: solve entry/exit**:

• T4 M6	spindle selection and enabling
• S12000 M3	spindle settings (speed and rotation: M3 = clockwise, M4 = counter-clockwise)
• G90G40	absolute programming, tool compensation (G40/G41/G42)
• **G0** X100.0Y-65 Zout	fast at working (X,Y), Z at safety position (Zout)
• M12	ancillary function
• **G0 Z**air	fast Z movement over the piece
• **G1** Z-12.5 F3000	interpolated Z movement at working position, F entry speed

The first two rows concerning the spindles settings are not present if the execution of a profile with the same technology precedes them.
As for fields **S**, **F** and **M3**/**M4**, the same abovementioned considerations apply.

The coordinate **Zout** presents the value assigned in setting **safety Z**, possibly brought back to the face reference frame.
The coordinate over the piece **Z air** presents the value assigned in the tool technological parameters or, if not available, in the setting **Measure over the workpiece**.

The profile continues with linear and circular interpolation lines:

- **G01**X250 F4000 M55
  - a linear interpolation is translated with code G01
  - X…Y…Z… axis coordinates (non-assigned axes do not move)

  - F… interpolation speed: it shows the originally programmed value, or the one assigned in the tool technological parameters.
  - M… translates the M field of the working (if strictly set positive)

- **G02**G17X…Y…I…J…F… M55
- **G03**G17X…Y…I…J…F… M55
  - a circular interpolation is translated with code G02 (clockwise rotation) or G03 (counter-clockwise rotation)
  - X…Y…Z… axis coordinates (non-assigned axes do not move)
  - G17 circular interpolation plane: G17 if XY plane (default), G18 if ZX plane, G19 if YZ plane. If not assigned, it propagates the value of the last assignment
  - I…J…K… centre coordinates, respectively on axis X, Y, Z. The two coordinates corresponding to the assigned plane are significant (either in absolute or relative mode, as per the setting **The coordinates of the centres apply G90/G91**)
  - F… interpolation speed: it presents the originally programmed value or the one assigned in the tool technological parameters.
  - M… translates the M field of the working (if strictly set positive)

The **F** field is translated upon the first element of the profile (G1/G2/G3) and the following elements only if a variation is programmed.
The **M** field is translated only in case of variation along the profile.

Selecting the option **Milling: solve entry/exit**, the profile ends with the lines:

- **G0** Zout — fast Z movement at safety position (Zout)
- M5 — spindle stop

The function **M5**, spindle stop, is not available, if the execution of a profile with the same technology precedes them.

Drilling working

The translation of a drilling working is determined by the options

- **Drilling: perform milling cycle**
- **Drilling G code**

If the field **Drilling: perform milling cycle** is not enabled, the working is translated with a fixed cycle (example: G81)

- **G81**G90 X100.0Y-65Z-12.5 T4M12S12000F3000
  - a point working is translated with the code **G81** (corresponding with the setting **Drilling G code**)
  - G90 absolute schedule (is all translated into G90)
  - X…Y…Z… axis coordinates
  - T… spindle or tool selection (if strictly set positive)
  - M… translates the M field of the working (if strictly set positive)
  - S… spindle rotation speed (RPM = revolutions per minute) – (if strictly set positive)
  - F… entry speed ([mm/min], [inch/min]) – (if strictly set positive)
  - D… programmed diameter, if there is no spindle nor tool selection.

In particular cases, the **T** field may present a multiple tool selection, with syntax "Tv1/v2,v3,..,vn":

- ✓ v1 = first tool (reference for position)
- ✓ v2 = second tool
- ✓ …
- ✓ vn = last tool.

The fields **S** and **F** present the originally programmed value or the one assigned in the tool technological parameters.

If the field **Drilling: perform milling cycle** is enabled, the working is translated with a milling cycle.
We shall see now how the same abovementioned working is translated with code G81:

- T4 M6 — spindle selection and enabling
- S12000 M3 — spindle settings (speed and rotation: M2 = clockwise, M4 = counter-clockwise)

- G90G40 absolute programming, disables the tool compensation
- G0 X100.0Y-65 Zout fast at working (X,Y), Z at safety position (Zout)
- M12 ancillary function
- **G0 Z**air fast Z movement over the piece
- **G1** Z-12.5 F3000 interpolated Z movement at working position
- **G0** Zout fast Z movement at safety position (Zout)
- M5 spindle stop

For fields **T**, **S**, and **F** the same abovementioned considerations apply.
The data present in **M3**/**M4** on the spindle rotation shows the value assigned in the tool technological parameters.
The function **M5**, spindle stop, is not available when the execution of a profile with the same technology follows.

15.5 From TpaCAD format to Edicad format

The conversion procedure must be obviously enabled by the machine constructor during the machine configuration. Conversion only applies to pieces with program or subroutine types.

Translation Mode

General information on piece

The General piece assignments are converted in this way:
- **Measuring unit and dimensions**: are recovered in the format EdiCad. To recover a program in inches [inch], it is necessary to deactivate conversion on measuring unit, in TpaCAD configuration, during creation phase of piece matrix. Otherwise the program in the format EdiCad will be converted in [mm].
- **Comment**: is recovered the format EdiCad, up to 250 characters.
- **"o" Variables**: the first three variables "o" are displayed in the offsets of the piece in environment EdiCad, in numerical format. Each parametric form is calculated.
- **"v" Variables**: the first eight "v" variables are displayed in the system variables of the piece in environment EdiCad, in numerical format. Each parametric form is calculated.
- **"r" Variables**: the assignments regarding the "r" variables are lost.
- **Variable Geometries**: the set fictive faces are recovered. The assignment is brought on three edges, in numerical format. Each parametric form is calculated. The assignment of a reference face and the definition of a fictive face on a geometry that is different from the Cartesian assignment of the three edges: in the piece in the format EdiCad there are the coordinates of the three edges of the face, in each case they are connected to the absolute system of the piece. Also fictive faces are recovered when assigned empty or like construction auxiliary faces. In the variable geometries also the automatic faces assigned with program in face-piece are recovered: in this case the face numbering of the face is connected to that of the fictive faces, taking up the first available numbers. Thickness settings for fictive faces are lost, as well as z axis direction setting.
- **Custom sections**: each assignment is lost.
- **Sequences**: each assignment is lost.

Programmed workings

In each face, conversion is made by maximum for 32500 workings assigned in matrix; other workings are lost.
Conversion involves all faces assigned to TpaCAD, included face-piece. In this case:
- face-piece workings are sorted to the relevant assignment faces, before any workings directly assigned in faces.
- automatic faces are converted into fictive faces.

The program in the format EdiCad only assigns the workings where the logical conditions are checked, as assigned in the piece stored in TpaCAD environment.
Moreover, conversion does not include system logic workings (cycles IF.. ELSEIF.. ELSE.. ENDIF, ERROR, EXIT, assignments of J variables).
Complex workings (of profile, application of subroutines or of macros) are exploded and each parametric assignment is solved and substituted with numerical setting.

Point workings

Point workings have operating code between 1 and 1000.
The operative code [81] in TpaCAD environment makes the schedule of a hole for a tool and for diameter; in the EdiCad environment the code [81] makes the schedule of a hole for diameter, while the code [82] makes the schedule of a hole for tool.
Considering the above, code [81] is translated into:
- code [81]: if no tool is assigned (tool field with value: 0);
- code [82]: if a value different than zero is assigned.

For all the other cases of point workings, conversion refers to operating code as assigned in matrix.
For all point workings, the following conversion rules apply:

TpaCAD **Format**	EdiCad **Format**
L field	Sets in Layer field (value up to 8 only)
O Field	Sets in Origin field (value up to 3 only)
M Field	Sets in M field
X coordinate of application point	Sets in Qx field
Y coordinate of application point	Sets in Qy field
Z coordinate of application point	Sets in Zp field
Machine	Sets in corresponding field, only if value is different from 0
Group	Sets in corresponding field, only if value is different from 0
Tool	Sets in corresponding field, only if value is different from 0
Tool type	Sets in corresponding field, only if value is different from 0
Tool Diameter	Sets in corresponding field, only if value is different from 0
Rotation speed	Sets in corresponding field, only if value is different from 0
Operating speed	Sets in corresponding field, only if value is different from 0
Slowdown coordinate on entry	Sets in corresponding field, only if value is different from 0
Slowdown coordinate on exit	Sets in corresponding field, only if value is different from 0
Custom parameters	Sets the same custom parameter, only if value is different from 0

Logical workings

Logical workings have operating code between 1 and 1000.

Conversion shows an operating code as assigned in matrix.
For all custom logical workings, the following conversion rules apply:

TpaCAD **Format**	EdiCad **Format**
L field	Sets in Layer field (value up to 8 only)
O Field	Sets in Origin field (value up to 3 only)
M Field	Sets in M field
Machine	Sets in corresponding field, only if value is different from 0
Group	Sets in corresponding field, only if value is different from 0
Tool	Sets in corresponding field, only if value is different from 0
Tool type	Sets in corresponding field, only if value is different from 0
Tool Diameter	Sets in corresponding field, only if value is different from 0
Rotation speed	Sets in corresponding field, only if value is different from 0
Operating speed	Sets in corresponding field, only if value is different from 0
Slowdown coordinate on entry	Sets in corresponding field, only if value is different from 0
Slowdown coordinate on exit	Sets in corresponding field, only if value is different from 0
Custom parameters	Sets the same custom parameter, only if value is different from 0

Setup

Setup workings have operating code between 1 and 1000.
Conversion shows an operating code as assigned in matrix. The following conversion rules apply to all setup workings:

TpaCAD Format	**EdiCad Format**
L field	Sets in Layer field (value up to 8 only)
O Field	Sets in Origin field (value up to 3 only)

M Field	Sets in M field
X coordinate of application point	Sets in Qx field
Y coordinate of application point	Sets in Qy field
Z coordinate of application point	Sets in Zp field
Machine	Sets in corresponding field, only if value is different from 0
Group	Sets in corresponding field, only if value is different from 0
Tool	Sets in corresponding field, only if value is different from 0
Tool type	Sets in corresponding field, only if value is different from 0
Tool Diameter	Sets in corresponding field, only if value is different from 0
Rotation speed	Sets in corresponding field, only if value is different from 0
Operating speed	Sets in corresponding field, only if value is different from 0
Slowdown coordinate on entry	Sets in corresponding field, only if value is different from 0
Slowdown coordinate on exit	Sets in corresponding field, only if value is different from 0
C axis coordinate (rotation)	Sets in corresponding field
B axis coordinate (rotation axis around Y axis)	Sets in corresponding field
Custom parameters	Sets the same custom parameter, only if value is different from 0

If matrix is generated with tool compensation applied, the assignments relevant to milling machine setup compensation are not set up by conversion.
For blade setup cases, compensation parameters are always set up by conversion, as not applied in matrix.

The following conversion rules apply:

TpaCAD format	**EdiCad Format**
Tool compensation selected (off/Left/Right)	Sets in corresponding field
Compensation radius	Sets in corresponding field, only if value is different from 0

The result of the tool compensation in TpaCAD and in EdiCad can be different: TpaCAD improves the recovery of situations that are managed in a different way or that are not managed EdiCad and it adds new performances, that in the step towards EdiCad would be lost (choke of profiles, variations in the profile corrections).
The creation of the matrix with the application of the tool compensation cancels these differences: the programs imported in EdiCad are already validated.

Profile

Linear typology
Conversion refers to operating code L01 [2201].
The following conversion rules apply:

TpaCAD Format	**EdiCad Format**
X coordinate of application point	Sets in Xf field, only if different from previous segment
Y coordinate of application point	Sets in Yf field, only if different from previous segment
Z coordinate of application point	Sets in Zf field, only if different from previous segment
Interpolation speed	Sets in corresponding field, only if value is different from 0
Custom parameters	Sets the same custom parameter, only if value is different from 0

Arc typology (xy plane)
Conversion refers to operating code A01 [2101].
The following conversion rules apply:

TpaCAD Format	**EdiCad Format**
X coordinate of application point	Sets in Xf field, only if different from previous segment
Y coordinate of application point	Sets in Yf field, only if different from previous segment
Z coordinate of application point	Sets in Zf field, only if different from previous segment
Centre X coordinate	Sets in Cx field

Centre Y coordinate	Sets in Cy field
Rotation direction	Sets in rotation field (0=clockwise, 1=counter-clockwise)
Interpolation speed	Sets in corresponding field, only if value is different from 0
Custom parameters	Sets the same custom parameter, only if value is different from 0

Arc typology (xz plane)
Conversion refers to operating code A05 [2105].
The following conversion rules apply:

TpaCAD Format	EdiCad Format
X coordinate of application point	Sets in Xf field, only if different from previous segment
Y coordinate of application point	Sets in Yf field, only if different from previous segment
Z coordinate of application point	Sets in Zf field, only if different from previous segment
Centre X coordinate	Sets in Cx field
Centre Z coordinate	Sets in Cz field
Rotation direction	Sets in rotation field (0 = clockwise, 1 = counter-clockwise)
Interpolation speed	Sets in corresponding field, only if value is different from 0
Custom parameters	Sets the same custom parameter, only if value is different from 0

In TpaCAD configuration, a broken line with linear segments can be chosen for recording in piece matrix, instead of arc in xz plane. In this case, each linear segment of broken line is converted by operating code L01 [2201] and considerations referred to linear typology profile working apply.

Arc typology (yz plane)
Conversion refers to operating code A06 [2106].
The following conversion rules apply:

TpaCAD Format	EdiCad Format
X coordinate of application point	Sets in Xf field, only if different from previous segment
Y coordinate of application point	Sets in Yf field, only if different from previous segment
Z coordinate of application point	Sets in Zf field, only if different from previous segment
Centre Y coordinate	Sets in Cy field
Centre Z coordinate	Sets in Cz field
Rotation direction	Sets in rotation field (0 = clockwise, 1 = counter-clockwise)
Interpolation speed	Sets in corresponding field, only if value is different from 0
Custom parameters	Sets the same custom parameter, only if value is different from 0

In TpaCAD configuration, a broken line with linear segments can be chosen for recording in piece matrix, instead of arc in yz plane.
In this case, each linear segment of broken line is converted by operating code L01 [2201] and considerations referred to linear typology profile working apply.

Arc typology (xyz plane)
An arc assigned on a generic plane (xyz) is converted into a piece matrix in a broken line with linear segments.
Criteria defining the broken line generation mode are assigned into TpaCAD configuration.
Each linear segment of broken line is converted by operating code L01 [2201] and considerations referred to linear typology profile working apply.

15.6 TpaCAD Program

A program that can directly be opened in TpaCAD is a text file written with a special syntax system.
The default extension applied by TpaCAD is (.tcn).
Here below it is described the format for the basic workings managed in order to allow a simple interface if it is required creating a program externally.

Check the program structure:

```
TPA\ALBATROS\EDICAD\02.00
$=TpaCAD interface test
::UNm DL=1000 DH=800 DS=40
SIDE#1{
$=home cell
W#81{ ::WTp
#1002=10 #1=101 #2=102 #3=-15 #8015=0 #2005=1.5 #2002=3300 #9012=-5 #9013=-10 #1001=1 }W
}SIDE
SIDE#2{
$=under
}SIDE
SIDE#3{
$=front
}SIDE
SIDE#4{
$=queue
}SIDE
SIDE#5{
$=behind
}SIDE
SIDE#6{
$=head
}SIDE
```

The structure proposed here corresponds to a parallelepiped piece, with minimum general assignments (dimensions and comment).

Header lines

Let us now see the blocks that define the structure.

```
TPA\ALBATROS\EDICAD\02.00
$=TpaCAD interface test
::UNm DL=1000 DH=800 DS=40
```

The first line is obligatory for the preliminary launch at the program opening.
The second row headed with "$=" assigns the comment to the program and it is optional. If there is it, it has to respect the header with "$=", followed by the description.

The third line is obligatory, and it assigns a unit of measurement and dimensions:
- "::" is the header;
- "UNm" unit of measurement in [mm] (default); "UNi" unit of measure in [inch];
- "DL=1000 DH=800 DS=40" dimensions: DL= length, DH= height, DS= thickness.

Fields are divided by the spaces.

Advanced Tools In Face Program

```
SIDE#1{
$=home cell
W#81{ ::WTp
```

```
#1002=10 #1=101 #2=102 #3=-15 #8015=0 #2005=1.5 #2002=3300 #9012=-5 #9013=-10 #1001=1 }W
}SIDE
```

The first row is obligatory to open the face program section: "SIDE#1{" it opens the section of the face 1, …, "SIDE#6{" it opens the section of the face 6.
The second row headed with "$=" assigns the face name and it's optional. If there is it, it has to respect the header with "$=", followed by the name.
Headed blocks with "W#nn{ ::" follow and they are closed with "}W" to define the face workings.
The last row ("}SIDE") is obligatory to close the face section.

It is not obligatory to assign the face sections that do not have programmed workings.

Section of assigned working in face program

```
W#81{ ::WTp
#1002=10 #1=101 #2=102 #3=-15 #8015=0 #2005=1.5 #2002=3300 #9012=-5 #9013=-10 #1001=1 }W
```

A working can be assigned on one or more text lines. Similar workings are proposed here for example:

```
W#81{ ::WTp #1002=10 #1=101 #2=102 #3=-15 #8015=0 #2005=1.5 #2002=3300 #9012=-5 #9013=-10
#1001=1 }W
```

```
W#81{ ::WTp
#1002=10 #1=101 #2=102 #3=-15 #8015=0 #2005=1.5 #2002=3300 #9012=-5 #9013=-10 #1001=1
}W
```

```
W#81{ ::WTp #1002=10 #1=101 #2=102 #3=-15
#8015=0 #2005=1.5 #2002=3300 #9012=-5 #9013=-10 #1001=1
}W
```

```
W#81{ ::WTp
#1002=10 #1=101 #2=102 #3=-15
#8015=0 #2005=1.5 #2002=3300 #9012=-5 #9013=-10 #1001=1
}W
```

in which the different parts have been put on the same line or divided on more lines, maintaining unchanged some syntax rules:
- the fields in a line are divided by the space;
- the section header has fixed structure "W#*nn*{ ::WT*c*" (for example: "W#81{ ::WTp"), with:
- nn = operative code (numerical) of the working,
- c = character that assigns the working typology (‘p’ = point, ‘s’ = setup, ‘l’ = line, ‘a’ = arc);
- the remaining fields have a fixed structure "#*nn*=*st*" (for example: "#1002=10"), with:
- nn = numerical identifier of the parameter;
- st = value assigned to the parameter;
- the section closure has fixed structure "}W".

Working: Hole

The header results: "W#81{ ::WTp".

The geometric parameters:

#1=	X coordinate of application
#2=	Y coordinate of application
#3=	Z coordinate of depth
#8015=	Application coordinates expressed in Absolute = 0 (default)/Relative = 1

Technological parameters in the case of programmed drilling for diameter:

#1002=	Tool Diameter
#201=	Machine
#203=	Group
#1001=	Tool type

Technological parameters in the case of programmed drilling by tools:

#201=	Machine
#203=	Group
#205=	Tool
#1001=	Tool type

General technological parameters:

#2005=	Tool entry speed (m/min)
#2002=	Spindle rotation speed (rpm)
#9012=	Slowdown coordinate on entry
#9013=	Slowdown coordinate on exit

Working: Mill setup

The header results: "W#89{ ::WTs".

The geometric parameters:

#1=	X coordinate of application
#2=	Y coordinate of application
#3=	Z coordinate of depth
#8015=	Application coordinates expressed in Absolute = 0 (default)/Relative = 1

The technological parameters to select the tool:

#201=	Machine
#203=	Group
#205=	Tool
#1001=	Tool type

General technological parameters:

#2005=	Tool entry speed (m/min)
#2002=	Spindle rotation speed (rpm)
#40=	Application of tool compensation: 1 for left compensation, 2 for right compensation, 0 for no required compensation (default)

Working: Line

The header results: "W#2201{ ::WTl".

The geometric parameters:

#1=	X application coordinate (end point of the linear segment)
#2=	Y application coordinate
#3=	Z coordinate of depth
#8015=	Application coordinates expressed in Absolute = 0 (default)/Relative = 1

General technological parameters:

#2008=	Tool entry speed (m/min)

Working: Arc in face plan

The header results: "W#2101{ ::WTa".

The geometric parameters:

#1=	X coordinate of application (end point of the arc)
#2=	Y coordinate of application
#3=	Z-coordinate of application
#8015=	Application coordinates expressed in Absolute = 0 (default)/Relative = 1
#31=	Centre X coordinate, in relative regarding the X initial coordinate of the arc
#32=	Centre Y coordinate, in relative regarding the Y initial coordinate of the arc
#34=	Direction of arc rotation: 0 if clockwise (default), 1 if counter-clockwise

General technological parameters:

#2008=	Interpolation speed (m/min)

Tecnologie e Prodotti per l'Automazione S.r.l.

Via Carducci, 221
I - 20099 Sesto S.Giovanni (MI)
Ph. +393666507029

www.tpaspa.com

info@tpaspa.it

August2026