SolidCAM GPPTool 2010

What's New

Content

1. General

1.1 Trace statement

Format:

trace <procedures>:<trace-level>

Description:

GPP will produce trace information while generating G-Code. The trace information will be generated only for those **SolidCAM** and user procedures which are defined in <procedures> list (see examples below). The <trace-level> determines how much information will be generated. The <trace-level> value must be in the range of 0 to 5, where 0 means no trace information at all, and 5 gives the maximum trace information available.

This information is of great interest in the development phase of a new post-processor; it can significantly shorten the time required to develop such a post-processor.

Examples:

trace “all”: 1	-	All **SolidCAM** and user procedures will be traced with the minimum trace information available.
trace "@proc":level	-	only procedure @proc will be traced, either it is **SolidCAM**'s or user's one**.**

Level definition:

level=0: no trace information
level=1: procedure routine name will be printed
level>1: trace data in produced.

Notes:

- @init_post is the only procedure that will NOT be traced. This is because @init_post initiates many internal variables, include the trace mode. Only after @init_post is executed **GPPTOOL** knows what procedures should be traced, and at what level.

- The trace printout contains the nesting level of the procedure, followed by the procedure name, and followed by a ">" symbol. After that symbol comes the trace data and the generated G-Code. Data.

Examples:

- if @line calls @user1 which calls @user2 (@line ==> @user1 ==> @user2)

  (1)@line > ....

  (2)@user1 > ....

  (3)@user2 > ....

- if @line calls @user1 and then calls @user2 (@line ==> @user1, @user2)

  (1)@line > ....

  (2)@user1 > ....

  (2)@user2 > ....

- If **SolidCAM** called to @proc and this proc was not in the GPP file, the trace would still trace it.

Note:

1. Nesting level for these procs is (0) give you extra information that the proc is missing.

2. If a proc is called from other proc - error message is displayed.

Example:

if @rapid_move does not exist, the trace printout prints it as nesting 0, but if that proc is called from @start_of_file for example, an error message is displayed.

1.2 GPPL Variables

Variable definition

A GPPL variable should adhere to the following rules:

- The first character should be a letter.
- All the other characters should be either a letter, digit or '_'.
- A variable name should not exceed **60** characters.

Note that **GPPL** does not distinguish between lower case and upper case (CAPITAL) letters.

Examples:

xpos , gcode_f , x0

1.3 Display Affix format

This is a new feature of a format string of a numeric variable or expression. GPPL will insert before or append after the number the required prefix of postfix (herein: affix) The <prefix> or <postfix> located at the very beginning or end of the format string, surrounded by '<' and '>' characters.

<prefix><sign><leading-zeroes><integer>.<fraction><trailing-zeroes><options><postfix>

- The affixes are optional.
- Affix might contain no string inside: '<>5.2'. This is the same as not write the <> at all.
- Affix length is limited by 31 characters. If a longer string is found, GPP will use the first 31 chars and will ignore the rest. No error message is produced.
- String variables do not have any format. Only numeric variables.
- The character '>' cannot be used inside the affixes strings..

Examples:

- '<X>5.2(p)' will print "X" just before the number.
- '5.2(p)<abc>' will append "abc" to the number.
- '<X>5.2<abc>' will produce both prefix and suffix.
- Old method: The below example will generate the correct number of spaces.

  xpos_f = <X>5.3

  ypos_f = <Y>5.3

  zpos_f = <Z>5.3

  {nb, [xpos' '], [ypos' '], [zpos' ']}

  Affix method: The below example will generate the correct number of spaces.

  xpos_f = <X>5.3< >

  ypos_f = <Y>5.3< >

  zpos_f = <Z>5.3< >

  {nb, [xpos], [ypos], [zpos]}

2. Postprocessor's writing style

2.1 The old Postprocessor's writing style

One of the most complicated tasks for the postprocessor developer is to set the home in the right place. The difficulties start with the need to set the **MAC** parameters correctly. The following **MAC** parameters have to be defined:

_4th_axes_around	=	Z/Y/X
first_rotation_angle	=	Z/X
_5th_axes_around	=	Z/Y/X
_5x_rotary_axes	=	YZ/ZX/XY/YX/ZYX
tilt_axis_dir	=	CCW
tilt_axis_dir_CW_CCW	=	Y/ N

This set of parameters influent the angles calculation for GPP subroutines: @tmatrix and @home_data. The postprocessor developer has to decide which set of angles that exist in @tmatrix and @home_data he needs to use:

```
..> rotate_angle_x:0.000T rotate_angle_y:0.000T rotate_angle_z:180.000T
..> opposite_rotate_angle_x:180.000T opposite_rotate_angle_y:180.000T
..> opposite_rotate_angle_z:0.000T
 .> rotate_angle_x_dir:cw rotate_angle_y_dir:cw rotate_angle_z_dir:cw
..> x_angle_const_z:0.000T y_angle_const_z:0.000T dev_angle_z:-180.000T
..> opposite_x_angle_const_z:-180.000T opposite_y_angle_const_z:-180.000T
    opposite_dev_angle_z:0.000T
..> x_angle_const_z_dir:cw y_angle_const_z_dir:cw dev_angle_z_dir:cw
..> x_angle_const_y:0.000T z_angle_const_y:0.000T dev_angle_y:-180.000T
..> opposite_x_angle_const_y:0.000T opposite_z_angle_const_y:0.000T
..> opposite_dev_angle_y:-180.000T
..> x_angle_const_y_dir:cw z_angle_const_y_dir:cw dev_angle_y_dir:cw
..> y_angle_const_x:0.000T z_angle_const_x:0.000T dev_angle_x:-180.000T
..> opposite_y_angle_const_x:0.000T opposite_z_angle_const_x:0.000T \
..> opposite_dev_angle_x:180.000T
..> y_angle_const_x_dir:cw z_angle_const_x_dir:cw dev_angle_x_dir:cw
..> angle_4x_around_x:0.000T angle_4x_around_y:0.000T
..> angle_4x_around_x_dir:cw angle_4x_around_y_dir:cw
```

Sometimes some angles have to be taken with opposite signs.
Sometimes no set of angles are correct for the kind of a machine. In this case the postprocessor developer needs to make his own calculation by using transformation matrix.

```
..> tmatrix_I_1:-1.000T tmatrix_I_2:0.000T tmatrix_I_3:0.000T tmatrix_I_4:0.000T
. .> tmatrix_I_5:0.000T tmatrix_I_6:-1.000T tmatrix_I_7:0.000T tmatrix_I_8:0.000T
..> tmatrix_I_9:0.000T tmatrix_I_10:0.000T tmatrix_I_11:1.000T tmatrix_I_12:0.000T
..> tmatrix_I_13:0.000T tmatrix_I_14:0.000T tmatrix_I_15:0.000T tmatrix_I_16:1.000T
```

- Some controllers have build-in new home calculation, as CYCLE 19 exist in the Hermle machine:

```
if save_x_angle_const_y <> opposite_x_angle_const_y
 or save_z_angle_const_y <>   opposite_z_angle_const_y ;
 or change_tool_flag == true
      {nb, ';'}
      {nb, 'CALL LBL 100'}
      {nb, 'L  Z+0 R0 FMAX M92'}
      {nb, 'CYCL DEF 7.0 NULLPUNKT'}
      {nb, 'CYCL DEF 7.1  X', shift_x:xpos_f}
      {nb, 'CYCL DEF 7.2  Y', shift_y:xpos_f}
      {nb, 'CYCL DEF 7.3  Z', shift_z:xpos_f}
      {nb, 'L  A', (-opposite_x_angle_const_y):xpos_f, '  C',
               (-opposite_z_angle_const_y):xpos_f, ' R0 F99999'}
      {nb, 'CYCL DEF 19.0 BEARBEITUNGSEBENE'}
      {nb, 'CYCL DEF 19.1  A', (-opposite_x_angle_const_y):xpos_f, '  C',
               (-opposite_z_angle_const_y):xpos_f}
endif

save_x_angle_const_y = opposite_x_angle_const_y

save_z_angle_const_y = opposite_z_angle_const_y
```

This option make the work of the postprocessor developer easier, but he still needs to decide what to do with the deviation angle depend with the possibilities of the machine.
If the machine doesn't support the CYCLE 19 the developer needs a lot of effort to write the post.

Example:

The calculation for Makino controller for **TABLE-TABLE** machine ZX kinematic:

```
(9999)

(HOMES TRANSLATION FOR VERTICAL 5AXES MACHINE)
(5AXE AROUND X)
(PLEASE USE FORMAT  "G65 P9999 I..J..K..C..A..U..D..)
(EXPLAIN: I = X ADRESS IN NEW HOME )
(-------------- J = Y ADRESS IN NEW HOME )
(-------------- K = Z ADRESS IN NEW HOME )
(-------------- C = C ANGLE DISPLACEMENT RELATIVE FROM EXISTING HOME,TABLE)
(-------------- A = A ANGLE DISPLACEMENT RELATIVE FROM EXISTING)
(--------------        HOME,AROUND X +30_-120)
(-------------- U = NUMBER OF OLD HOME: 54,55,.....,59)
(-------------- D = NUMBER OF NEW HOMELIKE U)

(#[5201+#22] = X MAC HOME)
(#[5202+#22] = Y MAC HOME)
(#[5203+#22] = Z MAC HOME)
(#[5204+#22] = A MAC HOME)
(#[5205+#22] = C MAC HOME)


(#1 - SHIFT Angle A POS)
(#3 - SHIFT Angle C POS)
```

```
(#4 - SHIFT X = I POS)
(#5 - SHIFT Y = J POS)
(#6 - SHIFT Z = K POS)

(OLD ROTATION CENTER)
(15.06.06)
(#13=-154.993                          TABLE ROTATION CENTER Y-AXIS)
(#14=-255.056                          TABLE ROTATION CENTER X-AXIS)
(#16=-600.036                          ROTATION CENTER Z-AXIS)
(#17=-154.935                          ROTATION CENTER Y-AXIS)

#13=-155.008                           ( TABLE ROTATION CENTER Y-AXIS)
#14=-255.067                           ( TABLE ROTATION CENTER X-AXIS)
#16=-600.020                           ( ROTATION CENTER Z-AXIS)
#17=-154.956                           ( ROTATION CENTER Y-AXIS)
#22=[#21-53.]*20.                      ( U )
#23=#7-53.                             ( D )
#124=-#1
#125=-#3
#126=#[5204+#22]                       ( A of MAC rotation)
#127=#[5205+#22]                       ( C of MAC rotation)
#130=#126                              ( A of MAC with opposite sign for back rotation)
#131=#[5202+#22]-#17+#5                ( Y distance from POS to A )
#132=#[5203+#22]-#16+#6                ( Z distance from POS to A)
#133=#131*COS[#130]-#132*SIN[#130]     ( Y distance from POS to A after rot -A MAC)
#134=#131*SIN[#130]+#132*COS[#130]     ( Z distance from POS to A after rot -A MAC)
#135=#133+#17                          ( Y distance from POS to abs after rot -A MAC)
#136=#134+#16                          ( Z distance from POS to abs after rot -A MAC)
#100=#[5201+#22]-#14+#4            ( X distance from POS to Rot Axis C before rot -C MAC)
#101=#135-#13                      ( Y distance from POS to Rot Axis C before rot -C MAC)
#103=#100*COS[#127]-#101*SIN[#127]     ( X distance from POS to Rot Axis C after rot -C MAC)
#104=#100*SIN[#127]+#101*COS[#127]     ( Y distance from POS to Rot Axis C after rot -C MAC)
#150=#103                              ( X distance from POS to Rot Axis C)
#151=#104                              ( Y distance from POS to Rot Axis C)
#103=#150*COS[-#127+#125]-#151*SIN[-#127+#125]  ( X distance from POS to Rot Axis C after rot C)
#104=#150*SIN[-#127+#125]+#151*COS[-#127+#125] ( Y distance from POS to Rot Axis C after rot C)
#105=#103+#14                          ( X distance from POS to abs after rot C)
#106=#104+#13                          ( Y distance from POS to abs after rot C)

#107=#136                              ( Z distance from POS to abs after rot C)
#117=#106-#17                          ( Y distance from POS to A after rot C)
#118=#107-#16                          ( Z distance from POS to A after rot C)
#137=-#[5204+#22]+#124                 ( A of POS)
#119=#117*COS[#137]-#118*SIN[#137]     ( Y distance from POS to A after rot C & rot A)
#120=#117*SIN[#137]+#118*COS[#137]     ( Z distance from POS to A after rot C & rot A)
#121=#105                              ( X distance from POS to abs after rot C & rot A)
#122=#119+#17                          ( Y distance from POS to abs after rot C & rot A)
#123=#120+#16                          ( Z distance from POS to abs after rot C & rot A)

#128=#126+#1
#129=#127+#3
G90 G10 L2 P#23 X#121 Y#122 Z#123 C#129 A#128

M99
```

2.2 The New Postprocessor's writing style

The postprocessor developer does not have to take care about defining of the **MAC** parameters and the GPP @home_data and @tmatrix subroutines.

2.2.1 Type of the supported coordinate systems SolidCAM provide

GCode output for a number of coordinate systems:

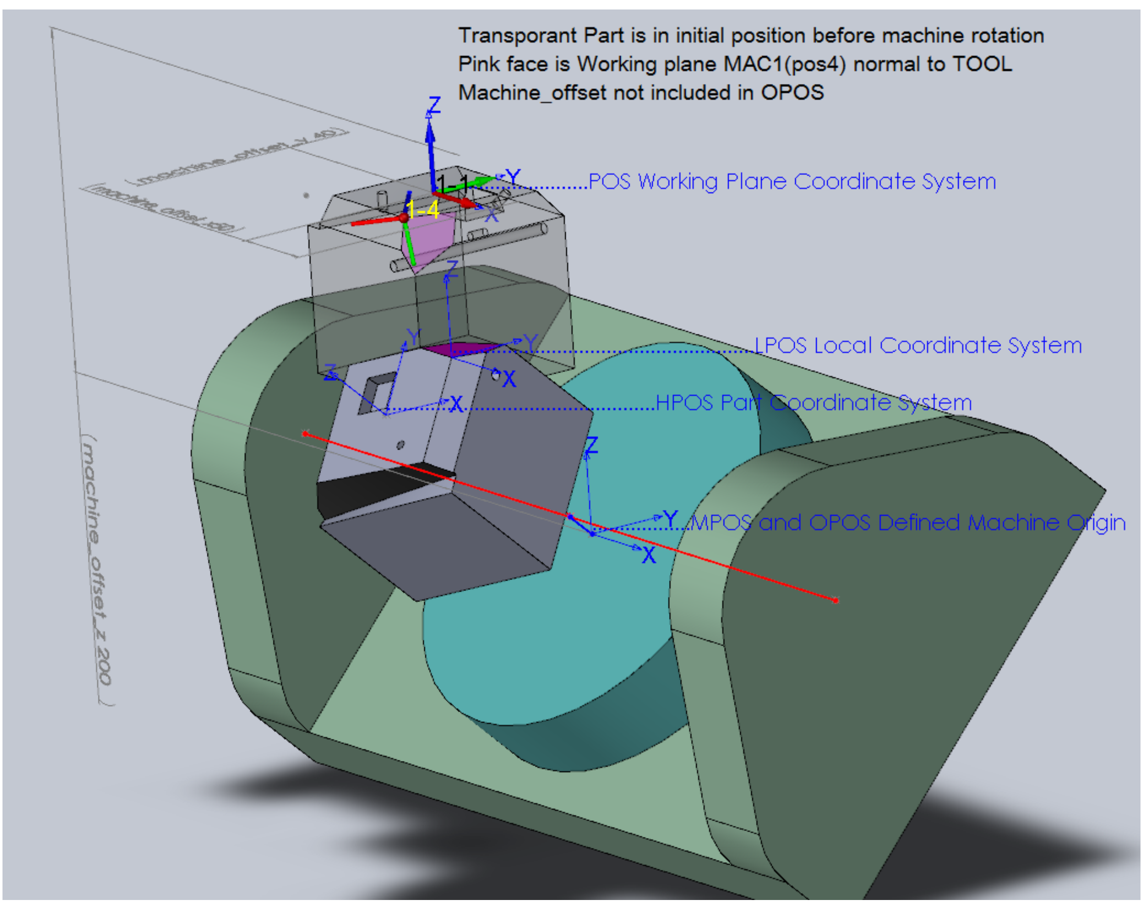
1. Part coordinate system MAC N Pos 1.
   Located in hpos set.

2. Absolute coordinate system according to absolute defined machine zero.
   Located in mpos set.

3. Absolute coordinate system according to absolute defined machine zero without machine offset calculation.
   Located in Opos set.

4. Working plane coordinate system with zero point in the Part home position.
   Located in pos set.

5. Local coordinate system with zero point shifted to the position location.
   Located in lpos set.



2.2.2 New MAC parameter

- For working in the new way you have to add the following new parameter to the **MAC** file:

  pos_to_machine = Y

- The machine kinematic is described in the section Sim Five axis:

  ;Sim Five axis

```
kinematic_type        = HEAD_TABLE
spindle_direction     = 0.0000 0.0000 1.0000
rotate_axis_dir1      = 0.0000 0.0000 1.0000
rotate_axis_dir2      = 0.0000 1.0000 0.0000
rot_axis_base_pnt1    = 0.0000 0.0000 0.0000, 0.0000 0.0000 0.0000
rot_axis_base_pnt2    = 0.0000 0.0000 230.0000, 0.0000 0.0000 0.0000
rot_axis_min_limit0   = -10000.0000
rot_axis_min_limit1   = -15.0000
rot_axis_max_limit0   = 10000.0000
rot_axis_max_limit1   = 195.0000
trans_axis_min_limit  = -100000.0000 -100000.0000 -100000.0000
trans_axis_max_limit  = 100000.0000 100000.0000 100000.0000
```

These **MAC** parameters are used in **SolidCAM2010** in addition for 5 axis simultaneous operations and Machine Simulation also for definition of machine .kinematic.

- **tilt_axis_min_limit** type :numeric

  This parameter defines the default min limit for the tilt rotational axis.

- **tilt_axis_max_limit** type :numeric

  This parameter defines the default max limit for the tilt rotational axis.

Example:
In **TABLE-TABLE** machine these **MAC** parameters are used to control the default angle pair and set as follows:

tilt_axis_min_limit = 0

tilt_axis_max_limit = 90

2.2.3 New GPP subroutine

@rotate_to_plane

SolidCAM develops the GPP subroutine @rotate_to_plane each time the machine has to rotate its rotation axes and/or change the working plane:

@rotate_to_plane ==>

```
first_axis_angle:0.000 second_axis_angle:0.000
opposite_first_axis_angle:0.000 opposite_second_axis_angle:0.000
change_tool_follows:1
rotate_angle_x:0.000T rotate_angle_y:90.000T rotate_angle_z:0.000T
opposite_rotate_angle_x:0.000T opposite_rotate_angle_y:90.000T
opposite_rotate_angle_z:0.000T
euler_angle_z:180.000T euler_angle_x:-90.000T euler_angle_dev_z:-90.000T

opposite_euler_angle_z:0.000T opposite_euler_angle_x:90.000T
opposite_euler_angle_dev_z:90.000T
shift_x:0.000T shift_y:0.000T shift_z:0.000T
shift_x_after_rot:0.000T shift_y_after_rot:0.000T shift_z_after_rot:0.000T
machine_offset_x:0.000 machine_offset_y:0.000 machine_offset_z:0.000
normal_to_plane_x:0.000T normal_to_plane_y:1.000T normal_to_plane_z:0.000T
tool_z_level:10.000 tool_start_plane:2.502
radial_start_tool_level:3.339 rear_start_tool_level:-2.502
xhnext_tool_z_level:0.389T yhnext_tool_z_level:10.000T zhnext_tool_z_level:0.001T
xmnext_tool_z_level:10.000T ymnext_tool_z_level:-0.389T zmnext_tool_z_level:0.001T
xhnext_start_tool_level:0.389T yhnext_start_tool_level:2.502T
zhnext_start_tool_level:0.001T
xmnext_start_tool_level:2.502T ymnext_start_tool_level:-0.389T
zmnext_start_tool_level:0.001T
xonext_start_tool_level:-0.389T yonext_start_tool_level:0.000T
zonext_start_tool_level:0.001T
```

first_axis_angle and second_axis_angle

first_axis_angle and second_axis_angle are the calculated angles of rotation around machine axes vectors.

- The angles first_axis_angle and second_axis_angle can be used for the rotation movements of rotation axes.
- One of the angles (which one depends of the kinematic) can be used for the working plane definition.

Example:
In the Integrex machine GCode function G68 is used for working plane definition.

rotate_angle_x   rotate_angle_y   rotate_angle_z

rotate_angle_x, rotate_angle_y and rotate_angle_z are the angles calculated in the ZYX order. Depending on the machine type these angles can be used to rotate its rotation axes and/or define the working plane.

euler_angle_z   euler_angle_x   euler_angle_dev_z

euler_angle_z, euler_angle_x and euler_angle_dev_z are the euler angles calculated in the ZXZ order. Depending on the machine these angles can be used to rotate its rotation axes and/or change the working plane.

shift_x   shift_y   shift_z

shift_x, shift_y and shift_z are the shifts in the part coordinate system. They can be used for the working plane defined coordinate system (lpos set) working style.

shift_x _after_rot   shift_y _after_rot   shift_z _after_rot

shift_x _after_rot, shift_y _after_rot and shift_z_after_rot are the shifts calculated after the machine devices TABLE and HEAD are rotated. They can be used for the working plane defined coordinate system (lpos set) working style.

machine_offset_x   machine_offset_y   machine_offset_z

machine_offset_x, machine_offset_y and machine_offset_z define the distance of the current home position (G54…) from absolute machine zero in X, Y, Z directions. They have to be set the same as defined in the home offset table on the machine itself.

normal_to_plane_x   normal_to_plane_y   normal_to_plane_z

normal_to_plane_x, normal_to_plane_y and normal_to_plane_z define a vector to the working plane.

2.2.4 GPP sets of coordinates in the existing GPP subroutines

@line and @rapid_move

The following sets of the coordinates have been developed:

xpos:-35.000T ypos:-29.000T zpos:100.000T
xhpos:-35.000T yhpos:-29.000T zhpos:0.000T
xmpos:-35.000T ympos:-29.000T zmpos:100.000T
xopos:-35.000T yopos:-29.000T zopos:100.000T
xlpos:-35.000T ylpos:-29.000T zlpos:100.000T

- **Part Coordinate System set:**

  xhpos, yhpos, zhpos

This set of coordinates calculation can be used for CL file generation.
It can be useful for a kind of machines that know to convert part coordinates to machine coordinates.

- **Machine Absolute Zero Coordinate System set:**

  xmpos, ympos, zmpos

- For this set coordinates calculation **SolidCAM** takes in account the kinematic parameters defined in **Sim Five axis** section of the **MAC** file.
- The home machine offset "Center of Rot. Origin based on Machine CoordSys (Sim 5 Axis)" that exist in **SolidCAM** home dialog is taken in account.
- If use_tool_h_length = Y the tool H length is taken in account.
- Limits of these coordinates are checked. If one or some limits have override a warning message is shown.

- **Machine Coordinate Axes with Current Reference Point set (G54, G55…):**

  xopos, yopos, zopos

- For this set coordinates calculation **SolidCAM** takes in account the kinematic parameters defined in **Sim Five axis** section of **MAC** file.
- For calculation this coordinate set the home machine offset parameter "Center of Rot. Origin based on Machine CoordSys (Sim 5 Axis)" that exist in **SolidCAM** CoordSys Data dialog is not taken in account.
- The tool H length value is not taken in account.
- For **HEAD_HEAD** machine this coordinate set has no influence since opos = hpos.
- For **HEAD_TABLE** machine rot_axis_base_pnt2 value is not taken in account.

- **Working Plane Coordinate System set:**

  xpos, ypos, zpos

- For this coordinates set calculation **SolidCAM** takes in account the following:
  - The kinematic parameters defined in **Sim Five axis** section of **MAC** file.
  - The angles that are used to define the working plane.

    - For calculation this coordinate set the home machine offset “Center of Rot. Origin based on Machine CoordSys (Sim 5 Axis)” that exist in **SolidCAM** CoordSys Data dialog is not taken in account.
    - The tool H length is not taken in account for **HEAD_HEAD** and **HEAD_TABLE** machines.
    - Rot_axis_base_pnt2 is not taken in account for **HEAD_TABLE** machine.
    - Rot_axis_base_pnt1 and rot_axis_base_pnt2 are not taken in account for **HEAD_HEAD** machine.
- These are the coordinates in the rotated plane where the tool axis is perpendicular to the plane.

  Example:

  These coordinates can be used after G68 that rotate the plane.
- Zero this coordinate system is in the Reference Point (G54, G55…)
- In case of **TABLE_TABLE** machine pos = opos.

➢ **Shifted Working Plane Coordinate System set:**

xlpos, ylpos, zlpos

This coordinate set is as pos coordinate set but shifted to the local coordinate system point.

@arc

Gpp subroutine @arc has the same sets of the coordinates as @line:

xpos:-35.000T ypos:-29.000T zpos:100.000T
xhpos:-35.000T yhpos:-29.000T zhpos:0.000T
xmpos:-35.000T ympos:-29.000T zmpos:100.000T
xopos:-35.000T yopos:-29.000T zopos:100.000T
xlpos:-35.000T ylpos:-29.000T zlpos:100.000T

Gpp subroutine @arc has the following additional data:

xcenter, ycenter, radius, zstart

The coordinates of the center in the Working Plane Coordinate System.

start_angle end_angle

The arc angles in the Working Plane Coordinate System.

arc_direction

The direction CW/CCW in the Working Plane Coordinate System.

xcenter_rel ycenter_rel

The coordinates of the arc center relative to the arc start point in the Working Plane Coordinate System.

xhcenter, yhcenter, zhcenter

The coordinates of the center in the Part Coordinate System.

Xhstart, yhstart, zhstart

The coordinates of the start point in the Part Coordinate System.

arc_plane_h, start_angle_h, end_angle_h

The plane start and end angles in the Part Coordinate System.

Xmcenter, ymcenter, zmcenter

The coordinates of the center in the Machine Absolute Zero Coordinate System.

Xmstart, ymstart, zmstart

The coordinates of the start point in the Machine Absolute Zero Coordinate System.

start_angle_m, end_angle_m

The start and the end angles in the Machine Absolute Zero Coordinate System.

xmcenter_rel ymcenter_rel

The coordinates of the arc center relative to the arc start point in the Machine Absolute Zero Coordinate System.

arc_plane_m

The plane in the Machine Absolute Zero System.

arc_plane_m:error

If arc plane is not one of main planes XY ZX YZ, the word “error” is shown in arc_plane_m:error.

xocenter, yocenter, zocenter

The coordinates of the center in the Machine Coordinate Axes with Current Reference Point.

xostart, yostart, zostart

The coordinates of the start point in the Machine Coordinate Axes with Current Reference Point.

arc_odirection

The CW/CCW direction in in the Machine Coordinate Axes System.

xocenter_rel yocenter_rel

The coordinates of the arc center relative to the arc start point in the Machine Coordinate Axes System.

Xlcenter, ylcenter, zlcenter

The coordinates of the center in the Shifted Working Plane Coordinate System.

Xlstart, ylstart, zlstart

The coordinates of the start point in the Shifted Working Plane Coordinate System.

xocenter_rel yocenter_rel

The coordinates of the arc center relative to the arc start point in the Shifted Working Plane Coordinate System.

@drill

Gpp subroutine @drill has the following sets of the coordinates:
The same sets of the coordinates as exist in @line:

xpos:-35.000T ypos:-29.000T zpos:100.000T
xhpos:-35.000T yhpos:-29.000T zhpos:0.000T
xmpos:-35.000T ympos:-29.000T zmpos:100.000T
xopos:-35.000T yopos:-29.000T zopos:100.000T
xlpos:-35.000T ylpos:-29.000T zlpos:100.000T

The following additional data sets:

drill_clearance_z:94.000 drill_upper_z:69.334 drill_lower_z:67.334
drill_clearance_zm:94.000 drill_upper_zm:69.334 drill_lower_zm:67.334
drill_clearance_zm:94.000 drill_upper_zm:69.334 drill_lower_zm:67.334
drill_clearance_zl:94.000 drill_upper_zl:69.334 drill_lower_zl:67.334

@drill_point

Gpp subroutine @drill_point has the following sets of the coordinates:
The same sets of the coordinates as exist in @line:

> xpos:-35.000T ypos:-29.000T zpos:100.000T
> xhpos:-35.000T yhpos:-29.000T zhpos:0.000T
> xmpos:-35.000T ympos:-29.000T zmpos:100.000T
> xopos:-35.000T yopos:-29.000T zopos:100.000T

The following additional data sets:

- **Coordinates of the point on the upper level**

  The Current Machine Home Coordinate System:

  xmupos:15.000T ymupos:-10.500T zmupos:67.334F

  The Part Coordinate System:

  xhupos:15.000T yhupos:-10.500T zhupos:-6.666F

  The Machine Coordinate Axes with Current Reference Point:

  xoupos:15.000T youpos:-10.500T zoupos:-6.666F

  The Working Plane Defined and Shifted Coordinate System:

  xlupos:15.000T ylupos:-10.500T zlupos:-6.666F

- **coordinates of the point on the safety level**

  The Current Machine Home Coordinate System:

  xmspos:15.000T ymspos:-10.500T zmspos:67.334F

  The Part Coordinate System:

  xhspos:15.000T yhspos:-10.500T zhspos:-6.666F

  The Machine Coordinate Axes with Current Reference Point:

  xospos:15.000T yospos:-10.500T zospos:-6.666F

  The Machine Working Plane Defined and Shifted Coordinate System:

  xlspos:15.000T ylspos:-10.500T zlspos:-6.666F

- **Coordinates of the point on the safety level**

  The Current Machine Home Coordinate System:

  xmspos:15.000T ymspos:-10.500T zmspos:67.334F

  The Part Coordinate System:

  xhspos:15.000T yhspos:-10.500T zhspos:-6.666F

  The Machine Coordinate Axes with Current Reference Point:

  xospos:15.000T yospos:-10.500T zospos:-6.666F

  The MachineWorking Plane Defined and Shifted Coordinate System:

  xlspos:15.000T ylspos:-10.500T zlspos:-6.666F

@call_proc

@change_tool

@start_of_job

Gpp subroutines @ call_proc, @change_tool, @start_of_job have the coordinate's sets of the operation first point:

Working Plane Defined Coordinate System set:

xnext:0 ynext:0 znext:100.000

Part Coordinate System set:

xhnext:0 yhnext:0 zhnext:0

Machine Absolute Zero Coordinate System set:

xmnext:0 ymnext:0 zmnext:0

Coordinate Axes with Current Reference Point set:

xonext:0 yonext:0 zonext:0

Working Plane Defined and Shifted Coordinate System set:

xlnext:0 ylnext:0 zlnext:0

@compensation

Gpp subroutine @compensation has the additional data for opos set:

side_o