



***CAMWORKS WIRE EDM
POST PROCESSOR
WRITER'S REFERENCE***

Disclaimer

Geometric Technologies, Inc. makes no warranties, either express or implied with respect to this manual. Geometric Technologies, Inc. reserves the right to revise and improve products as it sees fit, and to revise the specifications and information contained herein without prior notice. Due to continuing product development, specifications and capabilities described in this manual are subject to change without notice.

Copyright ©2008, 2009 Geometric Technologies, Inc. All Rights Reserved.



Geometric

Geometric Technologies, Inc. is a subsidiary of Geometric Ltd.

May 2009

Table of Contents

| | |
|-----------------------------------------------|------------|
| CHAPTER 1 COMMANDS AND VARIABLES | 1-1 |
| Commands | 1-2 |
| Variables | 1-4 |
| Defining Custom Post Variables | 1-19 |
| CHAPTER 2 POST SCRIPTING API FUNCTIONS | 2-1 |
| API Functions | 2-2 |

Table of Contents

Chapter 1 Commands and Variables

This chapter contains commands and variables for customizing CAMWorks Wire EDM post processing.

Commands

| Command | Purpose |
|-------------------------|----------------------------------------------------------------------------------------|
| oldvars | use old position (previous move) variables and misc. variables. |
| newvars | use new position (current move) variables and misc. variables. |
| force_x | force the X value to be output next time even if modal. |
| force_y | force the Y value to be output next time even if modal. |
| force_z | force the Z value to be output next time even if modal. |
| force_xl | force the X value of lower contour of 4Axis part to be output next time even if modal. |
| force_yl | force the Y value of lower contour of 4Axis part to be output next time even if modal. |
| force_xu | force the X value of upper contour of 4Axis part to be output next time even if modal. |
| force_yu | force the Y value of upper contour of 4Axis part to be output next time even if modal. |
| force_u | force the U value of upper contour of 4Axis part to be output next time even if modal. |
| force_v | force the V value of upper contour of 4Axis part to be output next time even if modal. |
| memo_xmove | Memorize the current X movement for use later. |
| memo_ymove | Memorize the current Y movement for use later. |
| memo_umove | Memorize the current U movement for use later. |
| memo_vmove | Memorize the current V movement for use later. |
| memorize_rethread_point | Output code to memorize current location for auto rethread (Sodick). |
| memorize_line_number | Memorize sequence number for use later. |
| memorized_line_number | Use memorize sequence number. |
| debug_on | Turns on automatic debug comments in post output. |
| debug_off | Turns off automatic debug comments in post output. |
| exit_if_glue_stop | Ignore the rest of the current contour if on glue stop. Used for Agie. |
| start_add_block_delete | Add block delete to all lines until stop_add_block_delete is used |
| stop_add_block_delete | Stop adding block delete that started from using start_add_block_delete |
| force_no_add_spaces | Force no spaces even when default is set to add spaces. |

| | |
|---------------------------------|------------------------------------------------------------------------------------------|
| <code>default_add_spaces</code> | Set add spaces back to condition before <code>force_no_add_spaces</code> was called. |
| <code>out_ctc_variables</code> | Output cutting conditions variables. Usually at the beginnig of the program. |
| <code>pass_specific</code> | Output start of pass block (Rough block for rough cut, 1st skim block for 1st skim etc). |

Variables

*Part Setup

| Command | Purpose |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <code>absolute_coord</code> | Output code to set coordinates to absolute mode (G90). Code defined in "G Codes/Absolute Coord:". |
| <code>incremental_coord</code> | Output code to set coordinates to incremental mode (G91). Code defined in "G Codes/Incremental Coord:". |
| <code>absolute_coord_no_code</code> | Set output mode to absolute, do not output G Code. |
| <code>incremental_coord_no_code</code> | Set output mode to incremental do no output G Code. |
| <code>work_coord</code> | Output work coordinates when it changing work pieces. (G54 etc) Code defined in "G Codes/Work offsets". |
| <code>pattern_work_coord</code> | Output work coordinates for pattern contours. (G54 etc) Code defined in "G Codes/Work offsets". |
| <code>zero_set_xp_yp</code> | Output code to set current work coordinate to programmed zero (G92X___Y___). Code defined in "G Codes/Coord zero set:". |
| <code>zero_set_xr_yr</code> | Output code to set current work coordinate to current cycle start point (G92X___Y___). Code defined in "G Codes/Coord zero set:". |
| <code>zero_set</code> | Output set zero command with out XY values (G92). Code defined in "G Codes/Coord zero set:". |
| <code>metric_mode</code> | Set output mode to metric. |
| <code>inch_mode</code> | Set output mode to inch. |
| <code>metric_mode_no_code</code> | Set output mode to metric without G code. |
| <code>inch_mode_no_code</code> | Set output mode to inch without G code. |
| <code>ref_plane</code> | Output reference plane value with prefix (Charmille). Prefix defined in "Prefixes/Reference plane". |
| <code>part_height</code> | Output workpiece height with prefix. Prefix defined in "Prefixes/Part height". |
| <code>machining_mode</code> | Output code for machining mode (no taper, taper, 4axis) |
| <code>work_plane_height</code> | Output work plane height. Determined by feature settings (Primary shape or Stock height, and use of land and taper). |
| <code>incr_taper_height</code> | Output incremental taper height from land, or primary feature shape. |
| <code>taper_height</code> | Output taper height. |
| <code>two_axis_with_taper_start</code> | Output code for two axis taper start. |
| <code>two_axis_with_taper_end</code> | Output code for two axis taper end. |

***Positioning**

| Command | Purpose |
|------------------------------|----------------------------------------------------------------------------------------|
| <code>xr</code> | Output rapid position x with prefix (X1.25) (X prefix hard coded). |
| <code>yr</code> | Output rapid position y with prefix (Y1.25) (Y prefix hard coded). |
| <code>xr_no_output</code> | X rapid value to previous move without outputting code. |
| <code>yr_no_output</code> | X rapid value to previous move without outputting code. |
| <code>xr_upper</code> | Output rapid position x upper guide with prefix (X1.25) (X prefix hard coded). |
| <code>yr_upper</code> | Output rapid position y upper guide with prefix (Y1.25) (Y prefix hard coded). |
| <code>zr_upper</code> | Output rapid position z upper guide with prefix (Z1.25) (Z prefix hard coded). |
| <code>xr_lower</code> | Output rapid position x upper guide with prefix (X1.25) (X prefix hard coded). |
| <code>yr_lower</code> | Output rapid position y upper guide with prefix (Y1.25) (Y prefix hard coded). |
| <code>zr_lower</code> | Output rapid position z lower guide with prefix (Z1.25) (Z prefix hard coded). |
| <code>xh</code> | Output home position x with prefix (X1.25) (X prefix hard coded). |
| <code>yh</code> | Output home position y with prefix (Y1.25) (Y prefix hard coded). |
| <code>xp</code> | Output programmed zero x (X___). (X prefix hard coded). |
| <code>yp</code> | Output programmed zero y (Y___). (Y prefix hard coded). |
| <code>prev_x</code> | Output previous x (X___). (X prefix hard coded). |
| <code>prev_y</code> | Output previous y (Y___). (X prefix hard coded). |
| <code>z_position_up</code> | Output code for programmable z position up. |
| <code>z_position_down</code> | Output code for programmable z position down. |
| <code>x_f</code> | Output code for x feed value with prefix. (X prefix hard coded). |
| <code>y_f</code> | Output code for y feed value with prefix. (Y prefix hard coded). |
| <code>u_f</code> | Output code for u feed value with prefix. (U prefix hard coded). |
| <code>v_f</code> | Output code for v feed value with prefix. (V prefix hard coded). |
| <code>u_f_incremental</code> | Output code for incremental u feed value with prefix. (U prefix hard coded). |
| <code>v_f_incremental</code> | Output code for change in incremental v feed value with prefix. (V prefix hard coded). |
| <code>u_f_incr_change</code> | Output code for change in incremental u feed value with prefix. (U prefix hard coded). |
| <code>v_f_incr_change</code> | Output code for incremental v feed value with prefix. (V prefix hard |

Variables

| Command | Purpose |
|------------------------|--------------------------------------------------------------------------------------------------------|
| | coded). |
| u_f_angle_perpend | Output code for change in angle perpendicular to xy move (Pos angle to right) |
| v_f_angle_parallel | Output code for change in angle parallel to xy move (Pos angle forward) |
| x_lf | Output x value with prefix for lower shape in 4 Axis entity to entity programming. |
| y_lf | Output y value with prefix for lower shape in 4 Axis entity to entity programming. |
| x_uf | Output x value with prefix for upper shape in 4 Axis entity to entity programming. |
| y_uf | Output y value with prefix for upper shape in 4 Axis entity to entity programming. |
| thread_point_x | Output x value with prefix for thread point. |
| thread_point_y | Output y value with prefix for thread point. |
| x_start_hole_rapid | Output x value with prefix for rapid feed to edge of start hole. |
| y_start_hole_rapid | Output y value with prefix for rapid feed to edge of start hole. |
| first_rapid_x | Output x value with prefix memorized first rapid point. |
| first_rapid_y | Output y value with prefix memorized first rapid point. |
| first_rapid_z | Output z value with prefix memorized first rapid point. |
| x_first_cut | Output x value with prefix for entrance cut when special cutting conditions are used to enter contour. |
| y_first_cut | Output y value with prefix for entrance cut when special cutting conditions are used to enter contour. |
| skip_next_move | Skip next movement in order to output later. Used for line swapping in Agie. |
| output_xmemo | Output the x value that was memorized with memo_xmove. |
| output_ymemo | Output the y value that was memorized with memo_ymove. |
| output_umemo | Output the u value that was memorized with memo_xmove. |
| output_vmemo | Output the v value that was memorized with memo_xmove. |
| u0_on_no_taper | Output U0.0 on 2axis part when no taper is used. |
| v0_on_no_taper | Output V0.0 on 2axis part when no taper is used. |
| return_to_thread_point | Output block to return to thread point. Used at the end of a program. |

***Arcs**

| Command | Purpose |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| xcenter | Output x center of arc with prefix ("Format/Arc center type"). Prefix define "Prefixes/Arc x center:". |
| ycenter | Output y center of arc with prefix ("Format/Arc center type"). Prefix define "Prefixes/Arc y center:". |
| rcenter | Output Radius value of arc with prefix ("Format/Arc center type"). Prefix define "Prefixes/Radius value:". |
| arc_center | Output x, and y center of arc with prefixes ("Format/Arc center type"). Prefix define "Prefixes/Arc x center:" & "Prefixes/Arc y center:". |
| xcenter_lf | Output x arc center value with prefix for lower shape in 4 Axis entity to entity programming. |
| ycenter_lf | Output y arc center value with prefix for lower shape in 4 Axis entity to entity programming. |
| xcenter_uf | Output x arc center value with prefix for upper shape in 4 Axis entity to entity programming. |
| ycenter_uf | Output y arc center value with prefix for upper shape in 4 Axis entity to entity programming. |
| iso_radius | Output code setting an Iso radius value with prefix. Code define in "G Codes/Iso radius:". |

***Line Numbers**

| Command | Purpose |
|-------------------|--------------------------------------------------------------------------------------------------------------|
| n | Output next sequence number with prefix. Prefix hard coded to (N). |
| n_forced | Force output of next sequence number with prefix (N). Sequence number for set in "Format/sequence numbers:". |
| agie_100_n | Used to set the first line number prefix to : for agie 100. |
| seq_numbers_start | Set the current N sequence number to the sequence number start value. |

***Sub Programs**

| Command | Purpose |
|-------------------|-----------------------------------------------------------------------|
| sub_call | Output subprogram call. Prefix defined in "Prefixes/Arc x center:". |
| sub_return | Output subprogram return. Prefix defined in "Prefixes/Arc y center:". |
| sub_num | Output subprogram number without prefix. |
| sub_num_no_prefix | Output subprogram number without prefix. |

Variables

| | |
|---------------------------|---------------------------------------------------------------------------------|
| sub_num_with_prefix | Output subprogram number with prefix. Prefix defined in "Prefixes/Subprogram:". |
| pattern_contour_sub_start | Output pattern contour code for subprogram start. |

*Comments

| Command | Purpose |
|----------------------------|---------------------------------------------------------------------------------|
| comment_start | Output comment start. Code defined in "Misc parameters/Comment start:". |
| comment_end | Output comment end. Code defined in "Misc parameters/Comment end:". |
| system_comment | Output the automatic comment "FEATURE 1 CNT2X ROUGH CUT FORWARD" |
| feature_name_comment | Output Feature Name comment "2 AXIS CONTOUR1-RECTANGULAR DIE3" |
| pass_name_comment | Output pass name comment "ROUGH1" |
| feature_pass_names_comment | Output Feature and pass names "2 AXIS CONTOUR1-RECTANGULAR DIE3 ROUGH1" |
| sub_comment | Output subprogram description comment. |
| output_date | Output the date in the form "Tue. 03/05/2002" |
| output_time | Output the current time in the form "11:35AM" |
| prog_n | Output program number without prefix (O0001). Hard code prefix ("O",prog_n) |
| prog_name | Output program name (file name of saved program on disk) |
| machine_make | Output machine manufacturers name as string. |
| machine_model | Output machine model name as string. |
| pass_name | Output pass name. (Used in comment) |
| feature_name | Output feature name. (Used in comment) |
| workpiece_name | Output workpiece name. (Used in comment) |
| feature_pass_names_comment | Output the comment for the pass name of the current cut of the current feature. |
| user_comment_1 | Output the comment entered in part settings comment 1. |
| user_comment_2 | Output the comment entered in part settings comment 2. |
| user_comment_3 | Output the comment entered in part settings comment 3. |
| user_comment_4 | Output the comment entered in part settings comment 4. |
| user_comment_5 | Output the comment entered in part settings comment 5. |
| user_comment_6 | Output the comment entered in part settings comment 6. |

| | |
|-----------------|---------------------------------------------------------|
| user_comment_7 | Output the comment entered in part settings comment 7. |
| user_comment_8 | Output the comment entered in part settings comment 8. |
| user_comment_9 | Output the comment entered in part settings comment 9. |
| user_comment_10 | Output the comment entered in part settings comment 10. |
| user_comment_11 | Output the comment entered in part settings comment 11. |
| user_comment_12 | Output the comment entered in part settings comment 12. |
| user_comment_13 | Output the comment entered in part settings comment 13. |
| user_comment_14 | Output the comment entered in part settings comment 14. |
| user_comment_15 | Output the comment entered in part settings comment 15. |
| user_comment_16 | Output the comment entered in part settings comment 16. |

***G Codes**

| Command | Purpose |
|----------------------|-------------------------------------------------------------------------------------------------------|
| feed_move | Set movement to feed and output feed move command (G01). Value defined in "G Codes/Feed move:". |
| rapid_move | Set movement to rapid and output rapid move command (G00). Value defined in "G Codes/Rapid move:". |
| cc | Output code for Cutter compensation. Codes defined in "G Codes". |
| measurement | Output measurement mode (inch/metric). Inch defined in "G Codes/Inch mode:" & "G Codes/Metric mode:". |
| g_taper | Output Code for taper cutting on. (At present hard coded to G51,G52 will add questions). |
| g_arc_move | Output G code for arc moves. G02, G03 hard coded. |
| g_lower | Output G code for lower shape in 4 Axis entity to entity programming (G01, G02, G03). |
| g_upper | Output G code for upper shape in 4 Axis entity to entity programming (G01, G02, G03). |
| four_axis_xyuv_start | Output code to start 4 axis xyuv cutting. Code defined in "G Codes/4axis start". |
| four_axis_xyuv_end | Output code to end 4 axis xyuv cutting. Code defined in "G Codes/4axis end". |

***M Codes**

| Command | Purpose |
|-------------|-----------------------------------------------------------------------|
| end_of_file | Output end of file code (M02). Code define in "M Codes/End of file:". |
| stop | Output stop code (M00). Code define in "M |

Variables

| | Codes/Stop:". |
|-------------------------------------------|---------------------------------------------------------------------------------------------------|
| optional_stop | Output code for Optional Stop (M01). Code define in "M Codes/Optional stop:". |
| stop_on_rough | Allow a stop to output if on rough cut and no glue stop |
| stop_tab_no_submerged | Output stop code if contour is tab cut when not in submerged machining. |
| stop_no_submerged | Output stop code if in submerged machining. |
| stop_no_autothread | Output stop code if no autothreading. |
| stop_for_tab_forced | Force the output of stop code on tab cut. |
| optional_stop_last_skim | Allow a optional stop to output on last skim pass |
| stop_for_tab | Output stop code for tab cut. |
| optional_stop_for_tab | Output optional stop for tab cut. |
| stop_die_on_rough_no_tab | Output Stop code at the end of the rough cut on a Die when no tap cut is used. |
| stop_on_rough_no_coreless | Output Stop code at the end of the rough cut except for the rough cut of a coreless cut. |
| optional_stop_die_on_rough_no_tab | Output Optional Stop code at the end of the rough cut on a Die when no tap cut is used. |
| optional_stop_on_rough_no_coreless | Output Optional Stop code at the end of the rough cut except for the rough cut of a coreless cut. |
| optional_stop_on_rough | Output Optional Stop code at the end of all rough cuts. |
| stop_last_skim | Output Stop code at the end of the last skim cut. |
| optional_stop_for_tab_forced | Output Stop code on tab cut, and force the output even if machine is stoped. |
| optional_stop_on_first_glue_stop | Output Stop code on first glue stop. |
| optional_stop_no_autothread | Output Stop code when not auto threading is used. |
| optional_stop_no_submerged | Output Stop code when not using submerged machining. |
| stop_for_tab_with_leadout_forced | Output Stop code on tab cut with leadout forced. |
| optional_stop_for_tab_with_leadout_forced | Output Optional Stop code on tab cut with leadout forced. |
| stop_on_first_glue_stop | Output Stop code on first glue stop. |

| | |
|------------------------|------------------------------------------------------------------------------------------|
| stop_no_output | Set the internal stop code variable to Stop, but do not output code. |
| high_pressure_pump_on | Output code to turn high pressure pump on. Code define in "M Codes/High pressure on:". |
| high_pressure_pump_off | Output code to turn high pressure pump off. Code define in "M Codes/High pressure off:". |
| cut_wire | Output code to cut wire. Code define in "M Codes/Wire thread:". |
| thread_wire | Output code to thread wire. Code define in "M Codes/Wire cut:". |

***Cutting Parameters**

| Command | Purpose |
|-------------------|----------------------------------------------------------------------------|
| submerged | Output code to fill tank if submerged machining is available and active. |
| wire_speed | Output code for wire speed if #522 is y. Prefix defined in question 611. |
| wire_tension | Output code for wire tension if #522 is y. Prefix defined in question 612. |
| generator_reg_num | Generator register number for Agie. |
| offset_reg_num | Offset register number. |
| flushing_reg_num | Flushing register number for agie. |
| angle_reg_num | Angle register number for agie. |
| wire_offset_force | Force the output of the "wire_offset" variable. |
| wire_offset_taper | Force the output of the "offset_taper" variable. |
| wire_offset_zero | Force the output of the "offset_zero" variable |
| ctc_register | Output the code for the current Cutting Conditions Register. |
| ctc_value | Output the Cutting Conditions value without prefix. |
| rapid_feed_rate | Output the feed rate using the rapid feed rate value. |

***Cutting Conditions**

| Command | Purpose |
|------------------------|----------------------------------------------------------------------------------------------------------------|
| cutting_cond_first_cut | Output code for cutting condition for entrance cut (power setting). Prefix define in "Prefixes/Cutting cond:". |
| cutting_cond | Output code for setting cutting conditions (power setting). Prefix define in "Prefixes/Cutting cond:". |

***Feed Rate**

| Command | Purpose |
|-----------|----------------------------|
| feed_rate | Output code for feed rate. |

Variables

| | |
|----------------------|--------------------------------------------------------|
| start_hole_feed_rate | Output code for rapid feed rate to edge of start hole. |
| feed_rate_first_cut | Output code for entrance cut feed rate. |
| feed_rate_force | Force the output of the "feed_rate" variable. |

*Offset

| Command | Purpose |
|----------------------|------------------------------------------------------------------------------------------------------------------|
| out_offset_variables | Output offsets register variables. (Used only if set to use variables for offset "Misc parameters/Taper-offset". |
| wire_offset | Output code to set wire offset. Codes defined in "G Codes/Offset left:", & "G Codes/Offset right:". |
| cancel_offset | Output cancel wire offset in 2Axis. Code defined in "G Codes/Cancel offset:". |
| cancel_offset_4axis | Output cancel wire offset in 4Axis. Code defined in "G Codes/Cancel 4axis offset:". |
| agie_wire_offset | Output code to set wire offset for agie machines. Prefix defined in "Prefixes/Wire offset:". |
| set_offset_register | Output code calling an offset register to set wire offset. |
| offset_register | Output code for the current offset register. |
| offset_value | Output offset amount. |

*Taper

| Command | Purpose |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------|
| out_taper_variables | Output taper angles as register variables. (Used only if set to use variables for taper "Misc parameters/Taper-offset". |
| cancel_taper | Output code for cancel taper. Code defined in "G Codes/Cancel taper:". |
| taper_angle | Output code to set taper angle. See taper format setting in "Misc parameters/Taper-offset". |
| taper_zero | Output taper zero degrees. Used to initialize taper to zero at the beginning of program. |
| taper_register | Output taper register number when taper is output using registers. |
| taper_value | Output taper angle. |
| taper_mode | Output taper mode (Left, Right, Off). |
| taper_angle_no_prefix | Output taper angle without prefix. |
| force_taper_zero_on_taper | Force the output of a taper of 0.0, When the set taper value is not 0.0. |

***Start Hole (Start hole drilling)**

| Command | Purpose |
|-------------------|-----------------------------------------------------------|
| drill_sub_calls | Output drill subroutine call for making start hole. |
| drill_end_of_file | Output drill end of file for making start holes. |
| drill_subprogram | Output drill sub program with hole positions. |
| drill_program | Output drill positions without sub programs. |
| z_drill | Output Z value for drilled hole. |
| z_clearance | Output Z clearance for positioning between drilled holes. |

***Script_File(AgieVision)**

| Command | Purpose |
|--------------------|-----------------------------------------------------------------|
| file_path | File path for all files when using script format. |
| open_script_file | Open the script file. |
| close_script_file | Close the script file. |
| open_iso_file | Open Iso file. |
| close_iso_file | Close Iso file. |
| iso_file_name | Iso file name. |
| set_to_nc_file | Set all writing functions to the nc file. |
| set_to_iso_file | Set all writing functions to the iso file. |
| set_to_script_file | Set all writing functions to the script file. |
| set_to_sbr_file | Set all writing functions to the sbr file. |
| iso_file | Iso file name. |
| script_file | Script file name. |
| material_name | Material name. |
| wire_name | Wire name. |
| punch_die_open | Output feature type (punch, die, open contour) for Agie Vision. |
| entry_mode | Output leadin/leadout mode for Agie Vision. |
| exit_mode | Output exit mode for Agie Vision. |
| stp_number | Start point number. |
| number_of_skims | Output number of skim passes. |
| tab_width | Output tab width. |
| reverse_tabs | Output setting for reverse tab cuts. |
| output_sub_calls | Output subprogram call line (" camw1"); |
| output_sub_def | Output subprogram define line ("sub camw1"); |

Variables

| | |
|----------------------------|-------------------------------------------------------------------------------|
| output_sub_number | Output the suprogram number ("1"). |
| zero_position_x | Work piece zero X value (ID_POSX). |
| zero_position_y | Work piece zero Y value (ID_POSY). |
| zero_position_z | Work piece zero Z value. |
| return_plane_distance | Work piece machining return Plane (ID_VALRETP). |
| security_plane_distance | Work piece machining security Plane (ID_VALSECP). |
| part_length | Work piece length (ID_DIML). |
| part_width | Work piece length (ID_DIMB). |
| priority_number | Machining order priority number (ID_PRIORITY). |
| quality_name1 | Output quality name for Normal cut and land cut for agie vision. |
| quality_name2 | Output quality name for Taper cut of land and taper for agie vision. |
| work_condition1 | Output work condition for Normal cut and land cut for agie vision. |
| work_condition1 | Output work condition for Taper cut of land and taper for agie vision. |
| ra_value1 | Output Ra value for Normal cut and land cut for agie vision. |
| ra_value2 | Output Ra value for Taper cut of land and taper for agie vision. |
| te_value1 | Output Te value for Normal cut and land cut for agie vision. |
| te_value2 | Output Te value for Taper cut of land and taper for agie vision. |
| tkm_value1 | Output Tkm value for Normal cut and land cut for agie vision. |
| tkm_value2 | Output Tkm value for Taper cut of land and taper for agie vision. |
| high_speed1 | Output high speed machining for Normal cut and land cut for agie vision. |
| high_speed2 | Output high speed machining for Taper cut of land and taper for agie vision. |
| smooth_finish1 | Output smooth finish setting for Normal cut and land cut for agie vision. |
| smooth_finish2 | Output smooth finish setting for Taper cut of land and taper for agie vision. |
| commutation_entry_distance | (ID_COMMPOINTENTRY) |

| | |
|-----------------------------------|---------------------------------------------------------------------------------------------|
| commutation_exit_distance | (ID_COMMPOINTEXIT) |
| four_axis_z_lower | Four axis cutting z lower section value (ID_POSZ). |
| start_hole_diameter | Start hole diameter value (ID_DIAMETER) |
| set_collar_cutting_land | Output the collar setting for land. |
| set_collar_cutting_taper | Output the collar setting for taper. |
| collar_postion | Z value for collar/land (ID_POSZ). |
| collar_taper_angle | Taper value for collar/land & taper (ID_TAPER). |
| collar_taper_height | Output the collar taper height. |
| collar_group_name | Output the collar group name. |
| end_point_x | End point X (ID_POSX). |
| end_point_y | End point Y (ID_POSY). |
| taper_collar_type | Collar/Land type (on top, on bottom). |
| die_clearance | Die clearance (ID_CLEARANCE). |
| prev_iso_file_name | Previous Iso file name. |
| collar_type | Output collar type for land and taper. |
| collar_land_height | Land heighth. |
| agie_work_name | Work piece name for taper in land and taper (LC1,LC2,LC3). |
| agie_group_name | Group name. |
| output_agie_taper_z | Output the taper Z value. |
| punch_die | Output the PUN / HOL value for punches and dies. |
| output_quality_one | Output that quality one is either quality is user defined or predefined. |
| output_quality_two | Output that quality two is either quality is user defined or predefined. |
| output_quality_one_coreless | Output that quality one for coreless is either quality is user defined or predefined. |
| output_quality_one_land_and_taper | Output that quality one for land and taper is either quality is user defined or predefined. |
| output_quality_two_land_and_taper | Output that quality two for land and taper is either quality is user defined or predefined. |
| output_die_clearance | Output die clearance in Agievision. |
| output_agie_taper | Output lines for taper in Agievision. |
| output_agie_leadout | Output lines for leadout in Agievision. |

***Script_File(Agie 123 with Jobs)**

| Command | Purpose |
|------------------------|-----------------------------------------------------------------------------------------------|
| agie_123_job_xr | Output X rapid block for agie 123 with jobs. |
| agie_123_job_yr | Output Y rapid block for agie 123 with jobs. |
| iso_file_path | Output the NC sub folder for the ISO file if user has selected NC files to create sub folder. |
| tech_file_name | Output the Technology file name. |
| tech_file_path | Output the Technology file path. |
| contour_direction | Output the contour chain direction (CCW or CW). |
| rough_pass_direction | Output the Rough pass contour direction (CCW or CW). |
| agie123_pass_direction | Output the contour direction (CCW or CW) using logic based on which pass is being cut. |

***Agie**

| Command | Purpose |
|------------------------------|------------------------------------------------------------------------------------------------------------|
| cc_coreless_agie | Output the appropriate offset (G40,G41) of coreless cut depending on if it is the 1st, 2nd, of 3rd entity. |
| agie_start_of_file | Output the agie start of file block if it has not be output yet. |
| last_move_direction_coreless | Output the X,Y of the vector of the last movement of the coreless cut. |
| last_move_direction | Output the X,Y of the vector of the last movement of contour cut. |
| first_move_direction | Output the X,Y of the vector of the first movement of contour cut. |
| first_move_split_arc | Output the first move of the contour if it is an arc. Split the arc and output it as two arc moves. |
| x_f_distance_from_end | Output the X move of a line moving to a point a distance back from the actual end point. |
| y_f_distance_from_end | Output the Y move of a line moving to a point a distance back from the actual end point. |
| x_f_arc_distance_from_end | Output the X move of a arc moving to a point a distance back along the arc from the actual end point. |
| y_f_arc_distance_from_end | Output the Y move of a arc moving to a point a distance back along the arc from the actual end point. |
| arc_center_distance_from_end | Output the new arc center when arc cut distance from arc end is used. |
| prev_v_to_zero | Set the prev V movement value to 0.0. |

***Sodick**

| Command | Purpose |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| sodick_taper_height | Output taper height as stock height, or land height depending on current settings, Use in header for TP value. |
| sodick_taper_opposite_height | Output taper opposite height as stock height, or land height depending on current settings, Use in header for TN value. |
| memorize_rethread_on_stop | Output the code for memorizing the rethread point when the wire breaks. |

***Charmille**

| Command | Purpose |
|-----------------|-----------------------------------------------------------|
| g27_on_no_taper | Output a G27 code when no taper is used on 2Axis contour. |
| taper_zero_blum | Output taper 0.0. Custom variable for Blum. |

***Ona**

| Command | Purpose |
|----------------------------|------------------------------------------------------------------------------------------------------------|
| technology_file_name | Output the technology file name that was entered on the ONA posting page. |
| xcenterup_minus_xcenterlow | Output X center value for upper section arc of 4 axis part incremental from X Center of lower section arc. |
| ycenterup_minus_ycenterlow | Output Y center value for upper section arc of 4 axis part incremental from Y Center of lower section arc. |

***Wire Guides**

| Command | Purpose |
|------------------------|---------------------------------------------------------------------------|
| metric_upper_guide_pos | Output upper wire guide position in metric regardless of inch/metric mode |
| upper_guide_pos | Output upper wire guide position. |
| lower_guide_pos | Output lower wire guide position. |
| guide_span | Output the distance between upper guide, and lower guide. |

***Misc Characters**

| Command | Purpose |
|--------------|-----------------------------------------------------------------------------|
| cr_lf | Output carriage return line feed characters. |
| single_quote | Output single quote character. |
| quote | Output double quote character. |
| n_spaces | Output spaces for the number of characters in the previous sequence numbers |

Variables

| | |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <code>four_axis_delimeter</code> | Output delimiter character between upper and lower. Delimeter value define in "Misc parameters/4axis contour delimeter:". |
| <code>null</code> | Output a ascii charater 0 or NULL. |

*Misc Variables

| Command | Purpose |
|-------------------------------|---------------------------------------------------------------------|
| <code>on_error_jump</code> | Output code for on error jump. Code defined in "M Codes/On error:". |
| <code>incr_upper_plane</code> | Output incremental value from program plane to upper plane. |

Defining Custom Post Variables

Files can be created to define custom post variables that can be read by the post processor. Variables can be defined for 2 Axis Contour operations, 4 Axis Contour operations and for EDM Settings. The variables display as parameters and options on the Adv Posting tab in the operation dialog boxes and the EDM Settings dialog box.

The files, which can be created in a text editor, need to have the same name as the post with the following extensions:

| <i>Custom variables for</i> | <i>File Extension</i> | <i>Example for Fanuc.pst</i> |
|-----------------------------|-----------------------|------------------------------|
| 2 Axis Contour | .Custom2Axis | Fanuc.Custom2Axis |
| 4 Axis Contour | .Custom4Axis | Fanuc.Custom4Axis |
| EDM Settings | .CustomSettings | Fanuc.CustomSettings |

File Format

The format is the same for all the files, as follows:

- There are 9 possible check boxes that can be accessed.
- There are 20 edit boxes, and 20 combo boxes that are on top of each other (COMBO_BOX 1-20, and EDIT_BOX 1-20). You can use only one in a group. If you use Edit box 1, then you have to use Combo box 2. If you use Edit boxes 1-5, then you need to use a Combo box, you have to start with Combo box 6, etc.
- There are 5 additional combo boxes numbered 21-25. Use these combo boxes if you need longer strings.
- You can use blank lines between variables and a single quote (') at the start of a line for comments.

Syntax

CHECK_BOX,1,Output Value

DEFAULT_CHECK,1,0

Where

CHECK_BOX = Type

1 = Check box number, 1-20

Output Value = Check box description (Label)

DEFAULT_CHECK = Setting the Check box default

1 = Check box number, 1-20

0 = Setting the check box value to (0 = unchecked, 1 = checked)

EDIT_BOX,1,REAL
TEXT_LABEL,1,Part Height
DEFAULT_REAL,1,25.4

Where

EDIT_BOX = Type
1 = Edit box number, 1-20
REAL = Decimal input
TEXT_LABEL = Edit box label
1 = Edit box number, 1-20
Part Height = Edit box label description
DEFAULT_REAL = Edit box Decimal default
1 = Edit box number, 1-20
25.4 = Edit box decimal default value

EDIT_BOX,2, INTEGER
TEXT_LABEL,2,Offset Number
DEFAULT_REAL,2,5

Where

EDIT_BOX = Type
2 = Edit box number, 1-20
INTEGER = Integer input
TEXT_LABEL = Edit box label
2 = Edit box number, 1-20
Offset Number = Edit box label description
DEFAULT_INTEGER = Edit box Integer default
2 = Edit box number, 1-20
5 = Edit box integer default value

EDIT_BOX,3, STRING
TEXT_LABEL,3,String value
DEFAULT_STRING,3,TESTING

Where

EDIT_BOX = Type
3 = Edit box number, 1-20
STRING = String input
TEXT_LABEL = Edit box label
3 = Edit box number, 1-20
String value = Edit box label description
DEFAULT_STRING = Edit box string default
3 = Edit box number, 1-20
TESTING = Edit box string default value

COMBO_BOX,5,Choice 1,Choice 2,Choice 3,Choice 4,Choice 5
TEXT_LABEL,5,Pick One
DEFAULT_COMBO_INDEX,5,3

Where

COMBO_BOX = Type of question

5 = Combo box number, (1-20, short) (21-25, long)

Choice 1,Choice 2,Choice 3,Choice 4,Choice 5 = How many selections and the description of each. Choices must be on one line and the line can have a maximum of 256 characters.

TEXT_LABEL = Combo box label

5 = Combo box number, (1-20, short) (21-25, long)

Pick One = Combo box label description

DEFAULT_COMBO_INDEX = Combo box selection default

5 = Combo box number, (1-20, short) (21-25, long)

3 = Default 3rd selection

Explanation of Options

The fields explained below would display these options on the Adv Posting tab:

Output value checkbox

Part Height edit box

Integer Value edit box

Pick one drop down

String Value edit box

CHECK_BOX,1,Output Value

Activate check box , #1, Use "Output Value" as the label.

DEFAULT_CHECK,1,1

Set default value for check box, #1, set to checked. (0 = unchecked, 1 = checked)

EDIT_BOX,1,REAL

Activate Edit box, #1, Set edit type to a real number.

TEXT_LABEL,1,Part Height

Set the text label for edit box, #1, to "Part Height".

DEFAULT_REAL,1,25.4

Set the default value edit box, #1, to 25.4.

COMBO_BOX,2,Choice 1,Choice 2,Choice 3,Choice 4,Choice 5

Activate combo box, #2, and set 5 selection choices (Choice 1, Choice 2, Choice 3, Choice 4, Choice 5)

TEXT_LABEL,2,Pick One

Set Text label for combo box, #2, to "Pick One".

DEFAULT_COMBO_INDEX,2,3

Set the default selection for combo box, #2, to index #3 (0 based index which would be 4th item).

Defining Custom Post Variables

EDIT_BOX,3,INTEGER

Activate edit box, #3, and set the value type to Integer.

TEXT_LABEL,3,Integer Value

Set the text label for edit box, #3, to “Integer Value”.

DEFAULT_INTEGER,3,222

Set the default integer value for edit box, #3, to 222

EDIT_BOX,4,STRING

Activate edit box, #4, and set the data type to String.

TEXT_LABEL,4,String Value

Set the text label for edit box, #4, to “String Value”

DEFAULT_STRING,4,TESTING

Set the default string value for edit box, #4, to “TESTING”

Sample File

The text file would look like this:

CHECK_BOX,1,Output Value

DEFAULT_CHECK,1,1

EDIT_BOX,1,REAL

TEXT_LABEL,1,Part Height

DEFAULT_REAL,1,25.4

COMBO_BOX,2,Choice 1,Choice 2,Choice 3,Choice 4,Choice 5

TEXT_LABEL,2,Pick One

DEFAULT_COMBO_INDEX,2,3

EDIT_BOX,3,INTEGER

TEXT_LABEL,3,Integer Value

DEFAULT_INTEGER,3,222

EDIT_BOX,4,STRING

TEXT_LABEL,4,String Value

DEFAULT_STRING,4,TESTING

Chapter 2 Post Scripting API Functions

This chapter contains the Wire EDM Post API functions for Visual Basic scripting (VBScript).

For information on how to use VBScript and details about the language elements, copy and paste the Microsoft Developer Network link below into your Internet browser:

[http://msdn.microsoft.com/en-us/library/d1wf56tt\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/d1wf56tt(VS.85).aspx)

API Functions

1. **short EDM_GetUnits()**

Return:

0 = Inch.

1 = Metric.

2. **short EDM_GetCuttingPassNumber()**

Return:

pass number

0 = Rough

1 = Skim #1

.

.

.

7 = Skim #7.

3. **short EDM_GetPassIsGlueStop()**

Return:

0 = Pass is not a glue stop.

1 = Pass is a glue stop.

4. **short EDM_GetProgramHasTapers()**

Return:

0 = Program contains no tapers.

1 = Program contains at least one tapered cut.

5. **short EDM_GetProgramFourAxis()**

Return:

0 = Program contains no 4Axis.

1 = Program contains at least one 4Axis cut.

6. **short EDM_GetProgramTwoAxisNoTaper()**

Return:

0 = Program contains atleast one taper cut.

1 = Program contains no tapered cuts.

7. **void EDM_SetReturnString(LPCTSTR ReturnString)**

For the case that the block will output code within an existing posting line (i.e., n, rapid_move, program_block_1,xr,yr).

This would be used in Place of “EDM_OutputText”, and “EDM_ProcessPostLine”.

Set:

Set the return string that will used to output code with and existing post line.

8. void EDM_OutputText(LPCTSTR OutString)

Output:

Line of text with CR/LF to the NC text file.

9. void EDM_ProcessPostLine(LPCTSTR OutString)

Output:

Output Line of variables and constants with CR/LF to the NC text file. (i.e.

“n,rapid_move,xr,yr,’M08”). The variables and constants will be treated just as it would in normal posting blocks.

10. void EDM_SetXFeed(double X)

Set:

X Feed move value used for 2Axis contour and 4Axis XYUV.

11. double EDM_GetXFeed()

Return:

X Feed move value used for 2Axis contour and 4Axis XYUV.

12. void EDM_SetYFeed(double Y)

Set:

Y Feed move value used for 2Axis contour and 4Axis XYUV.

13. double EDM_GetYFeed()

Return:

Y Feed move value used for 2Axis contour and 4Axis XYUV.

14. void EDM_SetUFeed(double U)

Set:

U Feed move value used for 4Axis XYUV.

15. double EDM_GetUFeed()

Return:

U Feed move value used for 4Axis XYUV.

16. void EDM_SetVFeed(double V)

Set:

V Feed move value used for 4Axis XYUV.

17. double EDM_GetVFeed()

Set:

V Feed move value used for 4Axis XYUV.

18. void EDM_SetXLowerFeed(double X)

Set:

X Lower Feed move value used for 4Axis entity to entity.

19. double EDM_GetXLowerFeed()

Return:

X Lower Feed move value used for 4Axis entity to entity.

20. void EDM_SetYLowerFeed(double Y)

Set:

Y Lower Feed move value used for 4Axis entity to entity.

21. double EDM_GetYLowerFeed()

Return:

Y Lower Feed move value used for 4Axis entity to entity.

22. void EDM_SetXUpperFeed(double X)

Set:

X Upper Feed move value used for 4Axis entity to entity.

23. double EDM_GetXUpperFeed()

Return:

X Upper Feed move value used for 4Axis entity to entity.

24. void EDM_SetYUpperFeed(double Y)

Set:

Y Upper Feed move value used for 4Axis entity to entity.

25. double EDM_GetYUpperFeed()

Return:

Y Upper Feed move value used for 4Axis entity to entity.

26. void EDM_SetPreviousXFeed(double X)

Set:

Previous X feed move value used for 2Axis and 4Axis XYUV.

27. double EDM_GetPreviousXFeed()

Return:

Previous X feed move value used for 2Axis and 4Axis XYUV.

28. void EDM_SetPreviousYFeed(double Y)

Set:

Previous Y feed move value used for 2Axis and 4Axis XYUV.

29. double EDM_GetPreviousYFeed()

Return:

Previous Y feed move value used for 2Axis and 4Axis XYUV.

30. void EDM_SetPreviousUFeed(double U)

Set:

Previous U feed move value used for 4Axis XYUV.

31. double EDM_GetPreviousUFeed()

Return:

Previous U feed move value used for 4Axis XYUV.

32. void EDM_SetPreviousVFeed(double V)

Set:

Previous V feed move value used for 4Axis XYUV.

33. double EDM_GetPreviousVFeed()

Return:

Previous V feed move value used for 4Axis XYUV.

34. void EDM_SetPreviousUpperXFeed(double UpperX)

Set:

Previous Upper X feed move value used for 4Axis Contour.

35. double EDM_GetPreviousUpperXFeed()

Return:

Previous Upper X feed move value used for 4Axis Contour.

36. void EDM_SetPreviousUpperYFeed(double UpperY)

Set:

Previous Upper Y feed move value used for 4Axis Contour.

37. double EDM_GetPreviousUpperYFeed()

Return:

Previous Upper Y feed move value used for 4Axis Contour.

38. void EDM_SetPreviousLowerXFeed(double LowerX)

Set:

Previous Lower X feed move value used for 4Axis Contour.

39. double EDM_GetPreviousLowerXFeed()

Return:

Previous Lower X feed move value used for 4Axis Contour.

40. void EDM_SetPreviousLowerYFeed(double LowerY)

Set:

Previous Lower Y feed move value used for 4Axis Contour.

41. double EDM_GetPreviousLowerYFeed()

Return:

Previous Lower Y feed move value used for 4Axis Contour.

42. BSTR MakeRealString(double RealValue)

Return:

String formatted accoring to Question:

414. for metric

415. for Inch.

43. BSTR MakeXString(double XValue)

Return:

String formatted accoring to Question:

414. for metric

415. for Inch.

With prefix from Qustion 676.

44. BSTR MakeYString(double YValue)

Return:

String formatted accoring to Question:

414. for metric

415. for Inch.

With prefix from Qustion 677.

45. BSTR MakeUString(double UValue)

Return:

String formatted accoring to Question:

414. for metric

415. for Inch.

With prefix from Qustion 606.

46. BSTR MakeVString(double VValue)

Return:

String formatted accoring to Question:

414. for metric

415. for Inch.

With prefix from Qustion 607.

47. short EDM_GetNumberOfIntMemoryLoc()

Return:

The number of integer memory locations allocated for global integer storage.

48. short EDM_GetNumberOfDoubleMemoryLoc()

Return:

The number of real number memory locations allocated for global real number storage.

49. short EDM_GetNumberOfStringMemoryLoc()

Return:

The number of string memory locations allocated for global string storage.

50. void EDM_SetIntMemoryLoc(short Index,short SetInteger)

Set:

Set the integer storage at Index with the value in SetInteger.

51. short EDM_GetIntMemoryLoc(short Index)

Return:

Get the integer storage value at Index.

52. Double EDM_GetDoubleMemoryLoc(short Index)

Return:

Get the real number storage value at Index.

53. void EDM_SetDoubleMemoryLoc(short Index, double SetDouble)

Set:

Set the real number storage at Index with the value in SetDouble.

54. BSTR EDM_GetStringMemoryLoc(short Index)

Return:

Get the string storage value at Index.

55. void EDM_SetStringMemoryLoc(short Index, LPCTSTR SetString)

Set:

Set the string storage at Index with the value in SetString.

56. short EDM_GetProcessAsPunchDie()

Return:

The part process order:

0 = PROCESS_AS_DIE.

1 = PROCESS_AS_PUNCH.

57. short EDM_GetProcessOrder()

Return:

The part process order set in edm settings.

PUNCH:

0 = By Complete Feature

1 = Rough & Skim / Glue Stop

2 = Rough / Skim / Glue Stop

DIE:

3 = By Complete Feature

4 = Rough / Glue Stop / All Skims

5 = Rough / Glue Stop / Skim Passes

6 = With Glue Stops / Without Glue Stops

58. short EDM_GetNumberOfFeatures()

Return:

Total number of **Feature**.

59. short EDM_GetCurrentFeatureNumber()

Return:

Current feature number.

60. short EDM_GetNumberOfContours()

Return:

Total number of contours (cut passes).

61. short EDM_GetCurrentContoursNumber()

Return:

Current contours (cut pass) number.

62. short EDM_GetNumberOfContourEntities()

Return:

Total number of the current contours geometric entities.

63. short EDM_GetCurrentContourEntityNumber()

Return:

The number of the current contours, current geometric entity.

64. short EDM_GetNumberOfStartHoles()

Return:

The number start holes in the program, this usually should be the same as the current number of features.

65. short EDM_GetOffsetSide()

Return:

0 = No Offset.

1 = Offset Left.

2 = Offset Right.

66. short EDM_GetIsPatternContour()

Return:

0 = Contour not part of a pattern.

1 = Contour is part of a pattern.

67. double EDM_GetFeedRate()

Return:

The current feed rate.

68. double EDM_GetTaperAngle()

Return:

The current taper angle.

69. double EDM_GetLowerGuidePosition()

Return:

The Z positions of the lower wire guide.

70. double EDM_GetUpperGuidePosition()

Return:

The Z positions of the upper wire guide.

71. double EDM_GetSlideHeight()

Return:

The slide height, This is only for Japax.

72. double EDM_GetLandHeight()

Return:

The land height for land and taper.

73. double EDM_GetGlobalStockHeight()

Return:

The actual height of the defined stock.

74. double EDM_GetGreatestZValue()

Return:

The greatest Z value in the program.

75. double EDM_GetFeatureStockHeight()

Return:

The operation stock height.

76. double EDM_GetMaximumTaperAngle()

Return:

The defined maximum taper angle.

77. double EDM_GetMaximumXYUVDifference()

Return:

The maximum XY length between the XY values and the UV values.

78. double EDM_GetZClearanceRapidMoves()

Return:

Z clearance value for rapid moves, used only with programmable Z machines.

79. double EDM_GetZClearanceFeedMoves()

Return:

Z clearance value for feed moves, used only with programmable Z machines.

80. double EDM_GetMinimumWorkPieceHeight()

Return:

The minimum height of work piece allowed.

81. Short EDM_GetLeadinType()

Return:

The Leadin Type.

0 = Arc.

1 = Perpendicular blend.

2 = Perpendicular

3 = Parallel

4 = Select Point.

5 = Select Point Blend.

82. Short EDM_GetLeadoutType()

Return:

The Leadout Type.

0 = Arc.

1 = Perpendicular blend.

2 = Perpendicular

3 = Parallel

4 = Select Point.

5 = Select Point Blend.

83. Short EDM_GetFirstLineNumber()

Return:

NC line number start value.

84. Short EDM_GetCurrentLineNumber()

Return:

Current NC Line Number.

85. Short EDM_GetLineNumberIncrement()

Return:

NC Line number increment.

86. Short EDM_GetIn4AxisCutting()

Return:

If the current feature is using 4Axis cutting.

0 = No.

1 = Yes.

87. Short EDM_GetIsIncremental()

Return:

If the coordinate output is incremental.

0 = No.

1 = Yes.

88. Short EDM_GetInsideOutsideOpen()

Return:

The cutting shape type.

-1 = Not Set.

0 = Inside cut.

1 = Outside cut.

2 = Open contour.

89. Short EDM_GetFeatureType()

Return:

Feature Type.

300 = Contour 2Axis.

301 = Contour 4Axis.

302 = Coreless cut.

90. Short EDM_GetReverseSkimPasses()

Return:

0 = Skims in same direction.

1 = Reverse each skim pass from previous.

91. Short EDM_GetPowerSetting()

Return:

Current power setting value (epack for mits).

92. Short EDM_GetUseAutoThreader()

Return:

0 = Manual wire threading.

1 = Automatic wire threading.

93. Short EDM_GetUseSubmergedMachining()

Return:

0 = Do not use submerged cutting.

1 = Use suberged cutting.

94. Short EDM_GetIsWireThreaded()

Return:

0 = Wire is not threaded.

1 = Wire is threaded.

95. short EDM_GetOutputSubPrograms()

Return:

0 = Output program using long form.

1 = Output program using subprograms.

96. short EDM_GetIsCorelessCut()

Return:

Used only with Contour 2Axis.

0 = Not using coreless cutting.

1 = This feature is using coreless cutting.

97. short EDM_GetUseLandAndTaper()

Return:

Used only with Contour 2Axis.

0 = No Land and taper is used.

1 = Land and taper is used.

98. short EDM_GetLandTopOrBottom()

Return:

If land and taper is used.

0 = Land is on the bottom.

1 = Land is on the top.

99. short EDM_GetAutoTaperRegNumber()

Return:

Get the automatically created taper register number.

100. short EDM_GetAutoOffsetRegNumber()

Return:

Get the automatically created offset register number.

101. short EDM_GetAutoCtcRegNumber()

Return:

Get the automatically created Ctc register number.

102. short EDM_GetManualReg1Number()

Return:

Get the manually entered register number for those machines that allow this option.

103. short EDM_GetManualReg2Number()

Return:

Get the manually entered register number for those machines that allow this option.

104. short EDM_GetManualReg3Number()

Return:

Get the manually entered register number for those machines that allow this option.

105. short EDM_GetManualReg4Number()

Return:

Get the manually entered register number for those machines that allow this option.

106. short EDM_GetPassCuttingDirection()

Return:

Cutting direction for skim cuts.

-1 = Counter Clockwise.

1 = Clockwise.

107. short EDM_GetRoughPassDirection()

Return:

Cutting direction for rough pass.

-1 = Counter Clockwise.

1 = Clockwise.

108. double EDM_GetStartHoleX()

Return:

X value for the current start hole.

109. double EDM_GetStartHoleY()

Return:

Y value for the current start hole.

110. double EDM_GetFirstCutX()

Return:

X value for the current first cut.

111. double EDM_GetFirstCutY()

Return:

Y value for the current first cut.

112. double EDM_GetStartHoleRapidFeedRate()

Return:

Rapid feed value used to move the wire to the edge of the start hole.

113. double EDM_GetFirstCutFeedRate()

Return:

Feedrate of the first cut.

114. double EDM_GetArcCenterX()

Return:

Arc center X value, used for 2Axis and 4Axis XYUV.

- 115. double EDM_GetArcCenterY()**
Return:
Arc center Y value, used for 2Axis and 4Axis XYUV.
- 116. double EDM_GetLowerArcCenterX()**
Return:
Lower Arc center X value, used for 4Axis contouring.
- 117. double EDM_GetLowerArcCenterY()**
Return:
Lower Arc center Y value, used for 4Axis contouring.
- 118. double EDM_GetUpperArcCenterX()**
Return:
Upper Arc center X value, used for 4Axis contouring.
- 119. double EDM_GetUpperArcCenterY()**
Return:
Upper Arc center Y value, used for 4Axis contouring.
- 122. double EDM_GetArcRelCenterX()**
Return:
Get Arc center I value.
- 123. double EDM_GetArcRelCenterY()**
Return:
Get Arc center J value.
- 124. double EDM_GetArcRadius()**
Return:
Arc radius.
- 125. double EDM_GetWireDiameter()**
Return:
Wire Diameter.
- 126. double EDM_GetStopDistance()**
Return:
Glue stop distance, or length.
- 127. double EDM_GetFirstRapidX()**
Return:
X value for the first rapid to start the program.

- 128. double EDM_GetFirstRapidY()**
Return:
Y value for the first rapid to start the program.
- 129. double EDM_GetFirstRapidZ()**
Return:
Z value for the first rapid to start the program.
- 130. double EDM_GetPartLength()**
Return:
Overall part length (X dimension).
- 131. double EDM_GetPartWidth()**
Return:
Overall part width (Y dimension).
- 132. double EDM_GetLandAndTaperAngle()**
Return:
Taper value used in land and taper.
- 133. double EDM_GetOffsetAmount()**
Return:
Current offset amount.
- 134. double EDM_GetFirstOffset()**
Return:
First offset amount.
- 135. double EDM_GetUpperPlaneZ()**
Return:
Z value of the upper cutting plane.
- 136. double EDM_GetLowerPlaneZ()**
Return:
Z value of the lower cutting plane.
- 137. double EDM_GetStartHoleDiameter()**
Return:
Start hole diameter.
- 138. short EDM_GetUserCheckBoxVariable(short Index)**
Return:
User defined operation check box variable 1..9

139. short EDM_GetUserEditIntegerVariable (short Index)

Return:

User defined operation integer variable 1..25

140. double EDM_GetUserEditRealVariable (short Index)

Return:

User defined operation real number variable 1..25

141. BSTR EDM_GetUserEditStringVariable (short Index)

Return:

User defined operation string variable 1..25

142. short EDM_GetUserSelectComboVariable (short Index)

Return:

User defined operation combo box index variable 1..25

143. short EDM_GetUserCheckBoxSettingsVariable(short Index)

Return:

User defined settings check box variable 1..9

144. short EDM_GetUserEditIntegerVariable (short Index)

Return:

User defined settings integer variable 1..25

145. double EDM_GetUserEditRealVariable (short Index)

Return:

User defined settings real number variable 1..25

146. BSTR EDM_GetUserEditStringVariable (short Index)

Return:

User defined settings string variable 1..25

147. short EDM_GetUserSelectComboVariable (short Index)

Return:

User defined settings combo box index variable 1..25

148. int EDM_GetMemorizedLineNumber()

Return:

Memorized Line Number.

149. int EDM_GetPriorityNumber()

Return:

AgieVision Priority number.

- 150. double EDM_GetPartHeight()**
Return:
Part Height.
- 151. double EDM_GetZeroPositionX()**
Return:
Zero Position X.
- 152. double EDM_GetZeroPositionY()**
Return:
Zero Position Y.
- 153. double EDM_GetZeroPositionZ()**
Return:
Zero Position Z.
- 154. double EDM_GetReturnPlaneDistance()**
Return:
Return Plane Distance.
- 155. double EDM_GetSecurityPlaneDistance()**
Return:
Security Plane Distance.
- 156. double EDM_GetTabWidth()**
Return:
Stop Distance.
- 157. double EDM_GetThreadPointX()**
Return:
Get Thread Point X.
- 158. double EDM_GetThreadPointY()**
Return:
Get Thread Point Y.
- 159. double EDM_GetCommutationEntryDistance()**
Return:
AgieVision Commutation Entry Distance.
- 160. double CScriptControlMacroDispatch::EDM_GetEndPointX()**
Return:
End Point X.

161. double EDM_GetEndPointY()

Return:
End Point Y.

162. double EDM_GetFourAxisZLower()

Return:
Get Four Axis Z Lower.

163. double EDM_GetCollarLandHeight()

Return:
AgieVision Collar Land Height.

164. double CScriptControlMacroDispatch::EDM_GetCollarPosition()

Return:
AgieVision Collar Position.

165. double EDM_GetDieClearance()

Return:
Punch Die Clearance.

166. BSTR EDM_GetWorkpieceName()

Return:
Workpiece Name.

167. BSTR EDM_GetMaterialName()

Return:
Material Name.

168. BSTR EDM_GetWireName()

Return:
Wire Name.

169. BSTR EDM_GetIsoFileName()

Return:
Iso File Name.

170. BSTR EDM_GetIsoFilePath()

Return:
Iso File Path.

171. BSTR EDM_GetFeatureName()

Return:
Feature Name.

172. BSTR EDM_GetFilePath()

Return:
File Path.

173. BSTR EDM_GetAgieWorkName()

Return:
Agie Work Name.

174. BSTR EDM_GetPrevIsoFileName()

Return:
Prev Iso File Name.

175. BSTR EDM_GetReverseTabs()

Return:
Reverse Tabs.

176. BSTR EDM_GetEntryMode()

Return:
Entry Mode.

177. BSTR EDM_GetPunchDieOpen()

Return:
Is it a Punch, Die, or Open.

178. BSTR EDM_GetQualityName1()

Return:
Quality Name 1.

179. BSTR_EDM_GetQualityName2()

Return:
Quality Name 2.

180. BSTR EDM_GetFileName()

Return:
NC output file name.

181. **double EDM_GetMoveDirectionLength()**

Return:
Agie parameter in post setting the move direction
Vector length.

182. **double EDM_GetStopBeforeEndOfGlueStopDistance()**

Return:
Agie parameter for distance to stop before the
End of the glue stop.

183. short EDM_GetGlueStopType()

Return:

This returns the glue stop type.

0 = Stop on contour.

1 = Stop with leadin/leadout.

184. BSTR EDM_GetComments(BSTR StrValue)

Input:

StrValue = The post command name for each comment (ie user_comment_1).

StrValue Inputs:

```
comment_start
comment_end
system_comment
feature_name_comment
pass_name_comment
feature_pass_names_comment
sub_comment
output_date
output_time
prog_n
prog_name
machine_make
machine_model
pass_name
feature_name EDM_GetFeatureName() already have call
workpiece_name EDM_GetWorkpieceName() already have call
feature_pass_names_comment
user_comment_1
user_comment_2
user_comment_3
user_comment_4
user_comment_5
user_comment_6
user_comment_7
user_comment_8
user_comment_9
user_comment_10
user_comment_11
user_comment_12
user_comment_13
user_comment_14
user_comment_15
user_comment_16
```

Return:

The Comment of each passed String. The return BSTR value will be in Cap's

185. BSTR EDM_GetGcode(BSTR StrValue)

Input:

StrValue = The post command name for each GCode (ie feed_move).

StrValue Inputs:

```
feed_move
rapid_move
```

```

cc
measurement
g_taper
g_arc_move
g_lower
g_upper
four_axis_xyuv_start
four_axis_xyuv_end

```

Return:

The GCode string value of each passed command string String. The return BSTR value will be in Cap's.

186. BSTR EDM_GetMcode(BSTR StrValue)

Input:

StrValue = The post command name for each MCode (ie stop).

```

StrValue Inputs:
end_of_file
stop
optional_stop
stop_on_rough
stop_tab_no_submerged
stop_no_submerged
stop_no_autothread
stop_for_tab_forced
optional_stop_last_skim
stop_for_tab
optional_stop_for_tab
stop_die_on_rough_no_tab
stop_on_rough_no_coreless
optional_stop_die_on_rough_no_tab
optional_stop_on_rough_no_coreless
optional_stop_on_rough
stop_last_skim
optional_stop_for_tab_forced
optional_stop_on_first_glue_stop
optional_stop_no_autothread
optional_stop_no_submerged
stop_for_tab_with_leadout_forced
optional_stop_for_tab_with_leadout_forced
stop_on_first_glue_stop
stop_no_output
high_pressure_pump_on
high_pressure_pump_off
cut_wire
thread_wire

```

Return:

The MCode string value of each passed command string String. The return BSTR value will be in Cap's.

187. BSTR EDM_GetCuttingConditionsFirstCut()

Return:

Cutting Conditions code for First Cut Move.

188. BSTR EDM_GetCuttingConditions()

Return:

Cutting Conditions code for current cut pass.

189. BSTR EDM_GetWireSpeed()

Return:

Wire speed for machines that use this parameter.

190. BSTR EDM_GetWireTension()

Return:

Wire tension for machines that use this parameter.

191. BSTR EDM_GetSodickTaperHeight()

Return:

Sodick taper height for land and taper.

192. BSTR EDM_GetSodickTaperOppositeHeight()

Return:

Sodick opposite taper height for land and taper.

193. BSTR EDM_GetTechnologyFileName()

Return:

Get technology file name for machines that use it.

194. BSTR EDM_GetXCenterUpMinusXCenterLow()

Return:

Get code to output the X differences between the upper arc and lower arc (Used for Ona Machines).

195. BSTR EDM_GetYCenterUpMinusYCenterLow()

Return:

Get code to output the Y differences between the upper arc and lower arc (Used for Ona Machines).

196. double CScriptControlMacroDispatch::EDM_GetOnaBreak()

Return:

OnaBreak.

197. short CScriptControlMacroDispatch::EDM_GetOnaVoltage()

Return:

OnaVoltage.

198. short CScriptControlMacroDispatch::EDM_GetOnaGap()

Return:

OnaGap.

199. **short** CScriptControlMacroDispatch::EDM_GetOnaWireStress()
Return:
OnaWireStress.
200. **short** CScriptControlMacroDispatch::EDM_GetOnaWaterHydrate()
Return:
OnaWaterHydrate.
201. **short** CScriptControlMacroDispatch::EDM_GetOnaWater()
Return:
OnaWater.
202. **short** CScriptControlMacroDispatch::EDM_GetOnaAutoCorners()
Return:
OnaAutoCorners.

