Post-Process (PPG) Reference Guide

RhinoCAM-PPG 2023

Monday, January 30, 2023

N049 G0
N050 G1Z-0. F29.334
N051 G02Z-0.125I-0.4375J-0.0002 F14.667
N052 G02Z-0.25I-0.4375J-0.0002
N053 G02X-0.4375Y-0.0002I-0.4375J-0.0002
N054 G02X0.4375Y0.0002I0.4375J0.0002
N055 G0Z0.25
N056 (Deep Drill)
N057 (Tool Diameter = 0.25 Leng
N058 G20 T3 M6
N059 G54
N060 S190 M3
N061 G90G0X0.75Y-0.75

MecSoft Corporation

RhinoCAM2023-PPG (Post-Processor Generator)

by MecSoft Corporation

User Notes:

Table of Contents

Index 93

Welcome

RhinoCAM2023

Prefer Printed Documentation? Check Here!

The RhinoCAM Post-Processor Generator is used to edit post processor files (SPM Files). These files are used by RhinoCAM during toolpath post-processing. RhinoCAM reads in a user specified SPM file, each file corresponding to a single machine tool controller, and generates the post-processed output using the rules resident in these files. Users have the ability to edit these files to modify these rules, thereby controlling the output that RhinoCAM generates.

For purposes of brevity, Rhino refers to both Rhinoceros 6 or Rhino 7.

Using the RhinoCAM Post-Processor Generator, these SPM files can be edited by following these steps:

First choose the required SPM file to edit from the Post Processor File Browser. After selecting the file, it can be edited using the Editor dialog. The format of various output blocks, such as motion, feed rates, spindle etc., can be edited by selecting the appropriate tabs in this dialog.

In addition to predefined block definitions, you can add startup codes as well as termination codes specific to the controller and shop practices. These blocks can be user-defined statements that may contain built in variables.

Related Topics

Good Thinks to Know

Post Processor File Browser

Main Editor

Variable List Dialog

Quick Start

RHINOCAM2023

Prefer Printed Documentation? Check Here!

Quick Start Guides for each RhinoCAM module are available in both PDF and Video format. Refer to the following information to access these resources:

What's New!

What's New in RhinoCAM 2023

Watch the What's New in 2023 Webinar!

The Complete Quick Start Video Play List

Here is a link to the complete 2023 Video Play List

How to Access the Quick Start Guide Documents

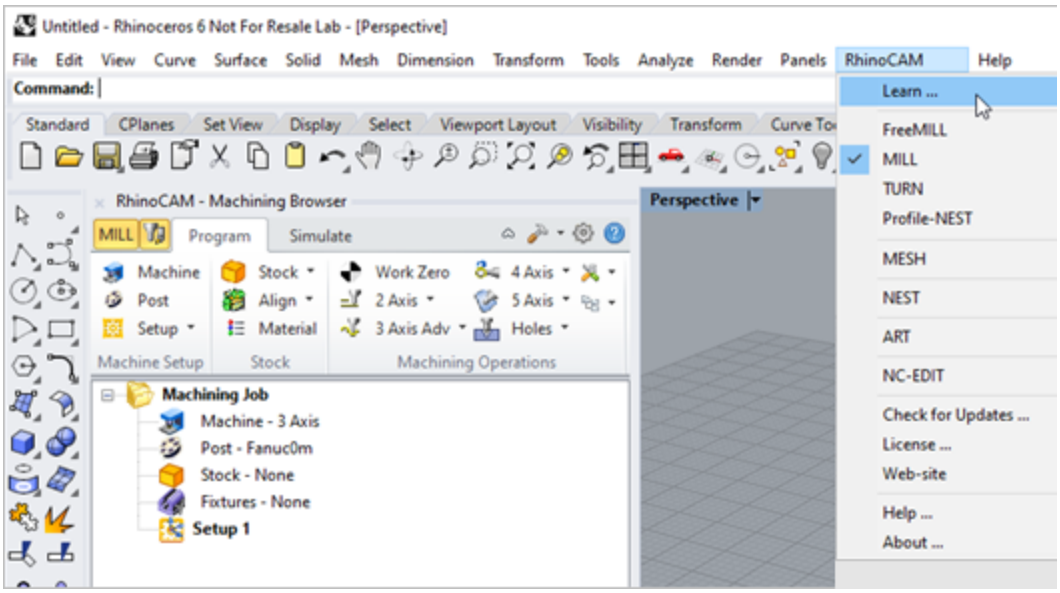
To help you quickly get started in working with each module, select one of the Help buttons located on the RhinoCAM Learning Resources dialog.

You will find:

- Quick Start Guides
- What's New documents
- Online Help links

The Quick Start Guides will help you step through an example tutorial which will illustrate how to use the module. To access the Learning Resources dialog:

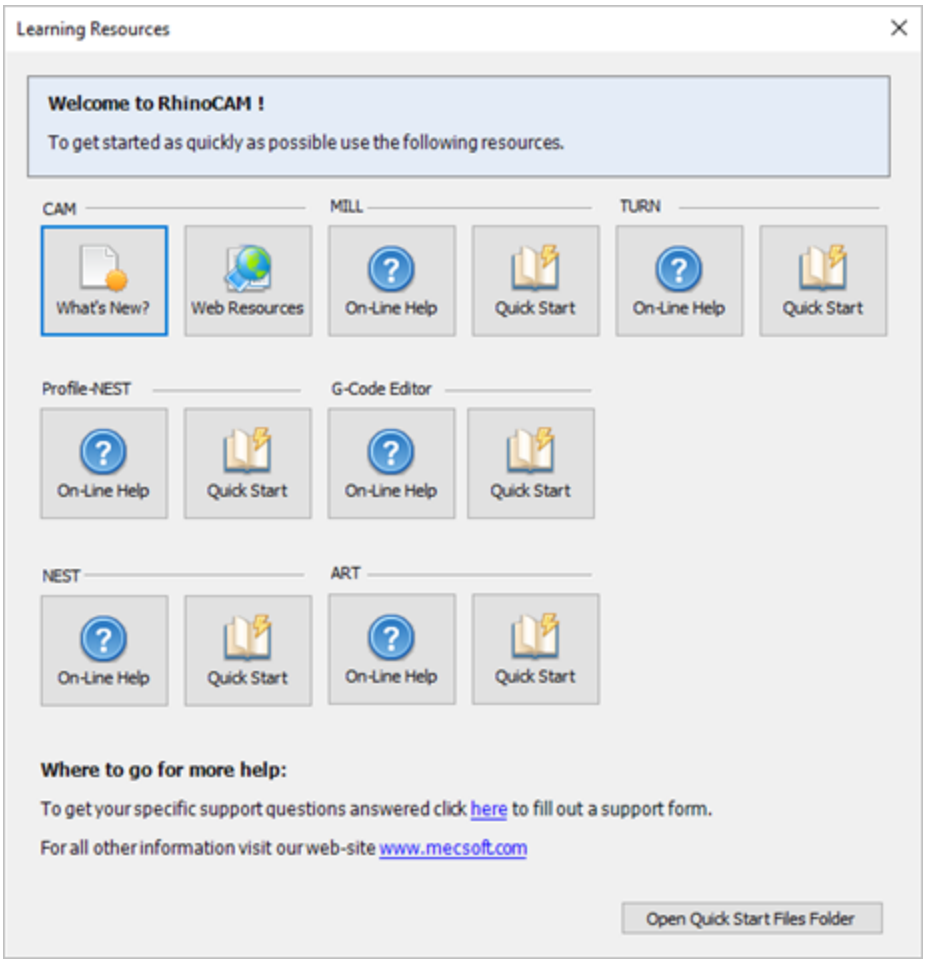
1. From the Rhino Main Menu, drop down the Main menu and select Learn ...



To access the Learning Resources dilog in RhinoCAM

2. Select a document from the Learning Resources dialog to get started using the module of your choice.

 You can also select the Open Quick Start Files Folder button located at the bottom of the dialog to open the Quick Start folder where the source files (start and completed versions) are located.



Learning Resources Dialog

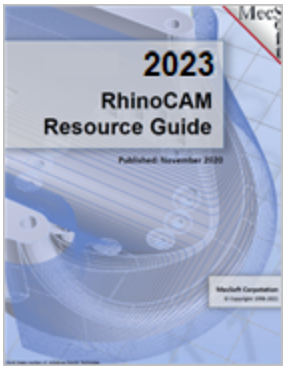
Related Topics

Find More Resources

Resource Guide

Download this PDF Guide for a list of the available RhinoCAM Resources.

2022 RhinoCAM Resource Guide



The 2023 RhinoCAM Resource Guide!

18 Pages

Lists PDF downloads and Online resources including Quick Start Guides, Reference Guides, Exercise Guides, Tutorials and More.

Prefer Printed Documentation? Check Here!

Good Things to Know

Here is a list of things you should know when posting G-Code using a customized post created from the Post-Processor Generator in RhinoCAM.

Algebraic Expressions when Posting

The Post Process Generator supports Algebraic Expressions in all input fields.

Here are some guidelines for using expressions:

1. Each expression should be placed in 'E{', 'E}' tags.
2. In expression can be used next operations: -,+,/,*,^
3. Negative values should be placed in parentheses '( )'
4. Expression parts can be placed in parentheses '( )'
5. Floating point numbers should be delimited by point symbol, use 0.xx in case of fractional numbers
6. Expressions can contain spaces in any place, spaces will be removed while parsing
7. Numbers in [-0.9; 0.9] can be written as [-.9; .9]

Examples:

- E{ ([SOME_VAR1]/2 + ([SOME_VAR]*(-3.2)))^3 E}
- E{[SOME_VAR1] + .3 E} SOME_TEXT E{ [SOME_VAR1] *(-1) E}

Posting Drill Cycles & Indexed Machining

Drill cycles will be converted to simulated cycles (i.e., using linear motions) if the setup the drill cycles appear in is not aligned with the machine Z axis. This is done only when the machine has a head configuration defined and Output all coordinates in local Setup Coordinate System is not checked. See Machine Tool Setup for more information.

Posting Cutter Compensation (G40, G41, G42)

All toolpaths except engraving are automatically compensated for the tool geometry. Cutter compensation is used typically to compensate for the difference in the dimensions of the actual cutter used in machining and the cutter used for programming in RhinoCAM. For example, if the cutter used in programming is 0.25 inches and due to tool wear the actual cutter is only 0.24 inches in size, you can compensate for this at the controller rather than having to re-program the operation in RhinoCAM.

Cutter compensation is used extensively in production (high volume) machining where the machine operator can compensate for tool wear before having to stop and replace the tool or insert.

In order to do this the user needs to do the following:

1. Turn cutter compensation on in the operation to Auto/ON or CONTROL/ON.
2. Specify the cutter compensation value and the compensation register in the controller (the controller needs to be capable of doing this).
3. Please make sure the post processor is configured to output cutter compensation. This is defined under the Cutter Compensation section in the post processor generator. Most controllers expect an X & Y motion on the same line as cutter compensation.

   Cutter Compensation Left

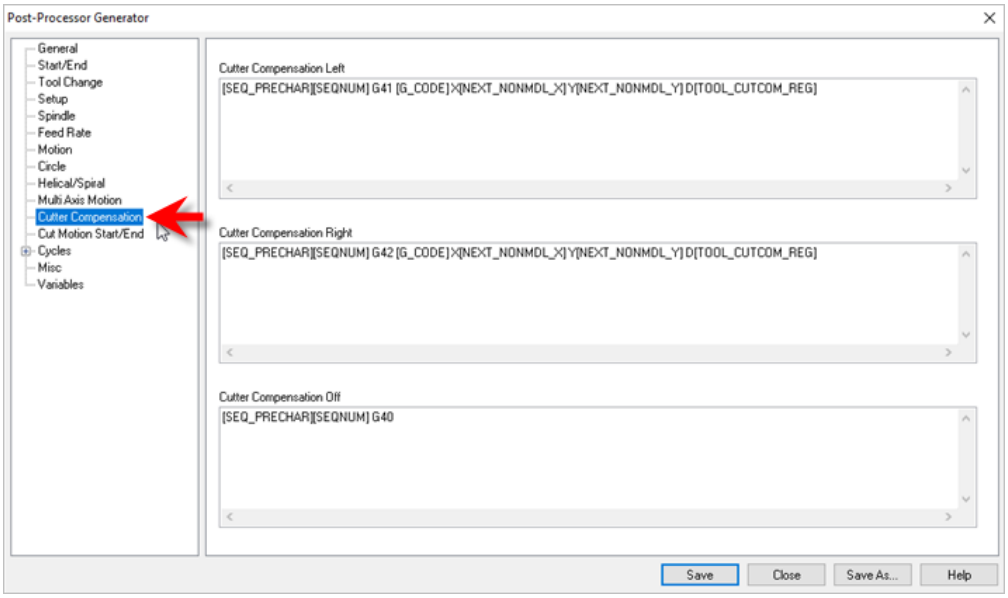
   [SEQ_PRECHAR][SEQNUM] G41 [G_CODE] X[NEXT_NONMDL_X] Y[NEXT_NONMDL_Y] D[TOOL_CUTCOM_REG]

   Cutter Compensation Right

   [SEQ_PRECHAR][SEQNUM] G42 [G_CODE] X[NEXT_NONMDL_X] Y[NEXT_NONMDL_Y] D[TOOL_CUTCOM_REG]

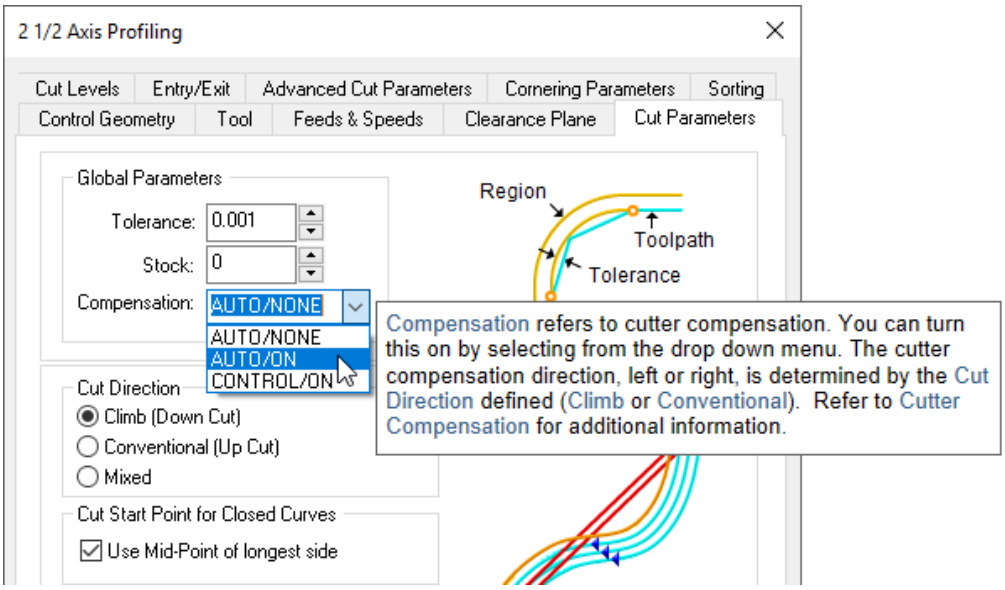
   Cutter Compensation Off

   [SEQ_PRECHAR][SEQNUM] G40



A few things to watch out for:

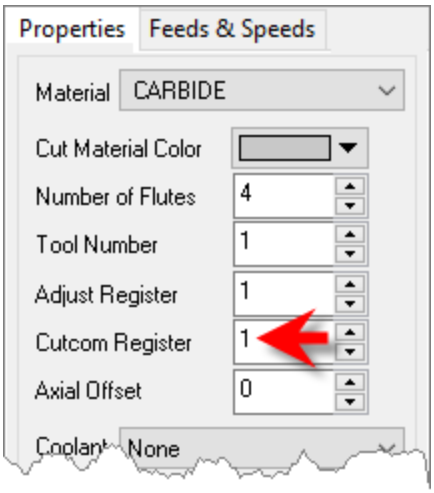
1. Cutter compensation makes sense only in 2-1/2 axis operations. If you are using roughing (pocketing & facing) the compensation will be turned on only in the final passes.
2. Make sure you are using Climb or Conventional cut traversal in any of the methods that you want to turn compensation on.



3. Make sure you have a linear motion for the controller to turn on the compensation for. If your first motion is an arc the controller will not be able to turn on the compensation. Thus, in 2-1/2 axis profiling, make sure there is a linear entry motion for the controller to be able to turn compensation on & linear exit to turn off compensation.

If you are looking to compensate for the full tool diameter, set Stock = -0.125 under the cut parameters tab. (0.125 being the radius of the tool). This would generate the toolpath ON the curve. This would invalidate the simulation as the tool tip stays on the drive geometry.

Note: The Cutcom Register is set under the Create/Select Tool definition dialog.



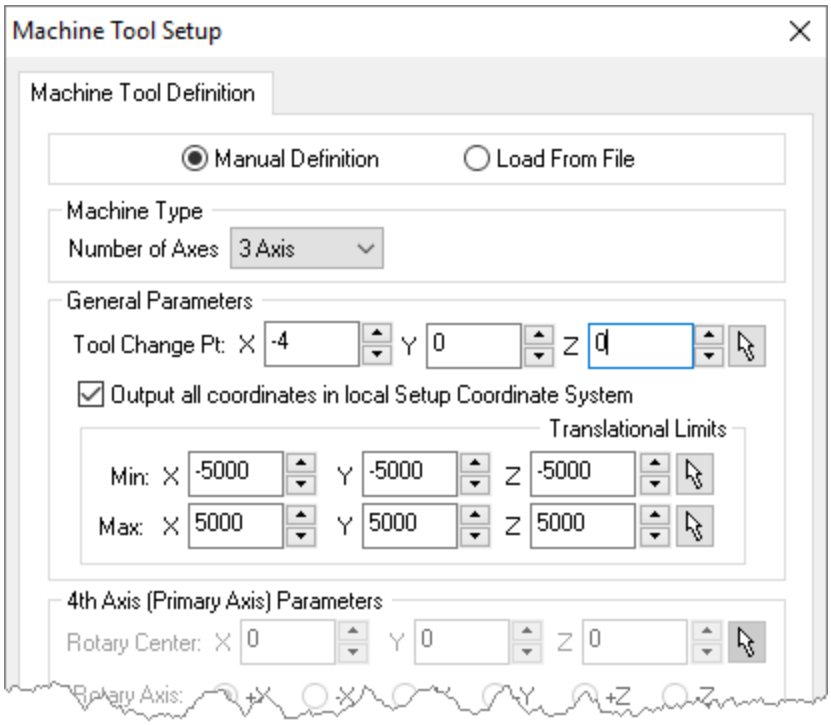
Posting a Tool Change Point

Implementing a Tool Change Point can be useful. For example in 2 and 3 Axis, you may want to change tools manually between operations (i.e., your CNC machine does not have an automatic tool changer). Also in 4 Axis you may want to ensure the tool is moved to a save location prior to a table rotation. To output a Tool Change Point to your posted g-code files, please do the following:

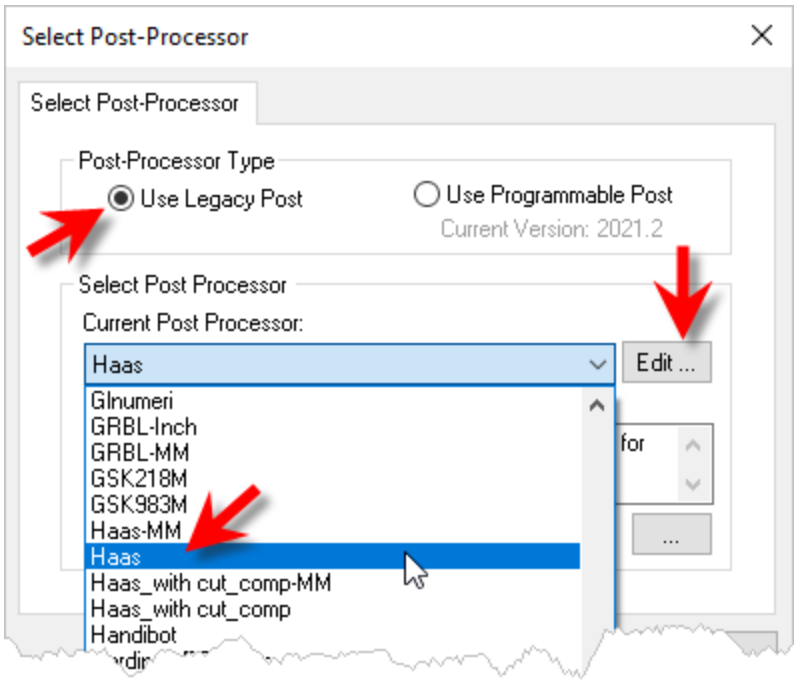
For 2 and 3 Axis Output

1. From the Machine Setup dialog (Program tab > Machine > General Parameters > Tool Change Pt), enter your required tool change point coordinates.
2. For the sample code (shown at the end of this section) we entered the following values in the Machine Setup dialog:

   X: -4, Y: 0 Z: 0

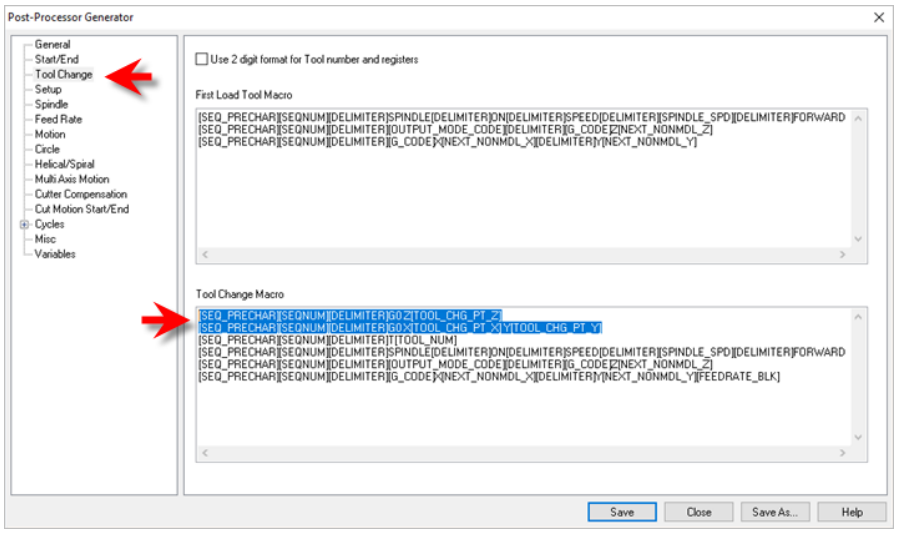


3. Edit your post processor by selecting Program tab > Post > Edit.



4. From the Post Process Generator dialog, select the Tool Change section from the left side of the dialog.
5. In the Tool Change Macro block section, replace the first line of text with the following two lines of text at the top of the macro. These two lines of text should precede the line that includes T[TOOL_NUM] as shown in the examples below.

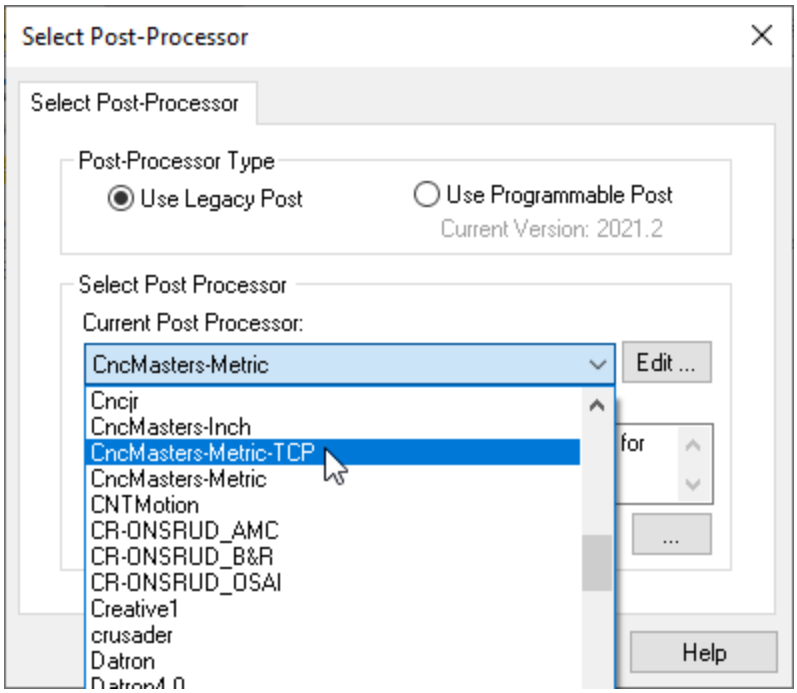
```
[SEQ_PRECHAR][SEQNUM][DELIMITER]G0 Z[TOOL_CHG_PT_Z]
[SEQ_PRECHAR][SEQNUM][DELIMITER]G0 X[TOOL_CHG_PT_X] Y[TOOL_CHG_PT_Y]
[SEQ_PRECHAR][SEQNUM][DELIMITER]T[TOOL_NUM]
...
...
```



6. If your controller expects to see an optional stop call BEFORE each tool change, you can add another line like below:

   [SEQ_PRECHAR][SEQNUM][DELIMITER]G0 Z[TOOL_CHG_PT_Z]
   [SEQ_PRECHAR][SEQNUM][DELIMITER]G0 X[TOOL_CHG_PT_X] Y[TOOL_CHG_PT_Y]
   [SEQ_PRECHAR][SEQNUM][DELIMITER]M01
   [SEQ_PRECHAR][SEQNUM][DELIMITER]T[TOOL_NUM]
   …
   …

7. From the Post Process Generator dialog, pick Save As.
8. Enter a unique name for your post file (**.spm*) for testing and pick Save.
9. From the Set Post-Processor Options dialog, select the revised post from the Current Post Processor list.



10. **Note**: If you do not see your revised post in the list, select the "..." button to the right of the "Folder where post-processor file are located" and select the folder where you saved your revised post file to (see Step 7 above) and pick OK.
11. You should now see your revised post in the list. Select it and pick OK.
12. Post a sample toolpath using the revised post.
13. Review the g-code test file and locate the first tool change lines of code.
14. Your sample test should look something like this depending on your post (based on the tool change point we used in Step 2 above). Note the tool change coordinates in blue:

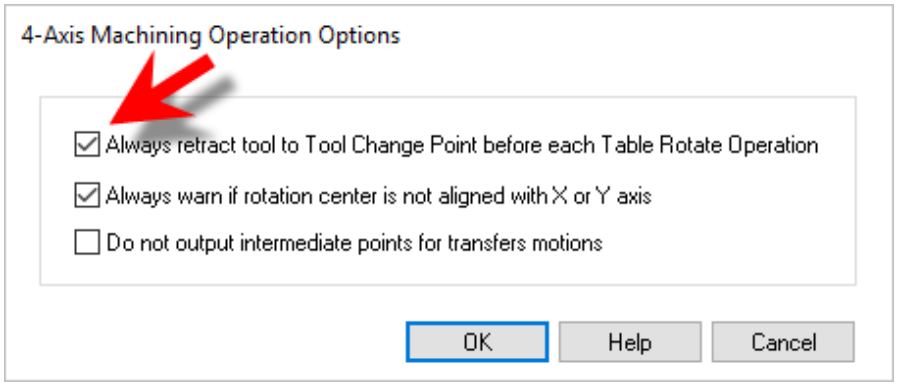
```
...
...
N66 ;2 1/2 Axis Profiling
N68 G0 Z0.
N70 G0 X-4. Y0.
N72 T1 M06
...
...
```

15. That's it!

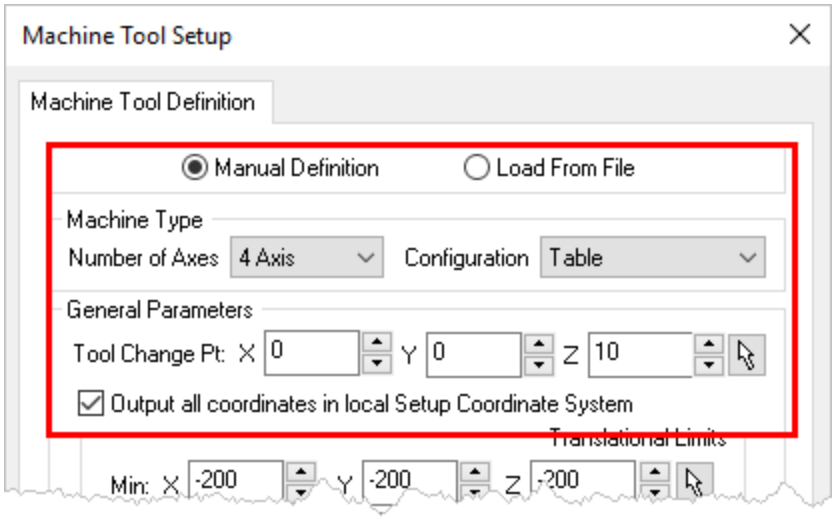
For 4 Axis Output

1. From the Program tab select 4 Axis.

2. From the 4 Axis menu select 4 Axis Options.
3. From the 4 Axis Operation Options dialog check the box to Always retract tool to Tool Change Point before each Table Rotate Operation.



4. Now from the Program tab select Machine and then Manual Definition.
5. For the Machine Type select 4 Axis.
6. Under General Parameters, enter the X, Y and Z coordinate values for the Tool Change Point.



7. Then check the box to Output all coordinates in local Setup Coordinate System and then pick OK to close the dialog.
8. Post the 4 Axis toolpath operation and verify that the Tool Change Point is being posted before the table rotation angle similar to this:

```
...
...
(Setup 2)
N6263 Z10.
N6264 X0.Y0.
```

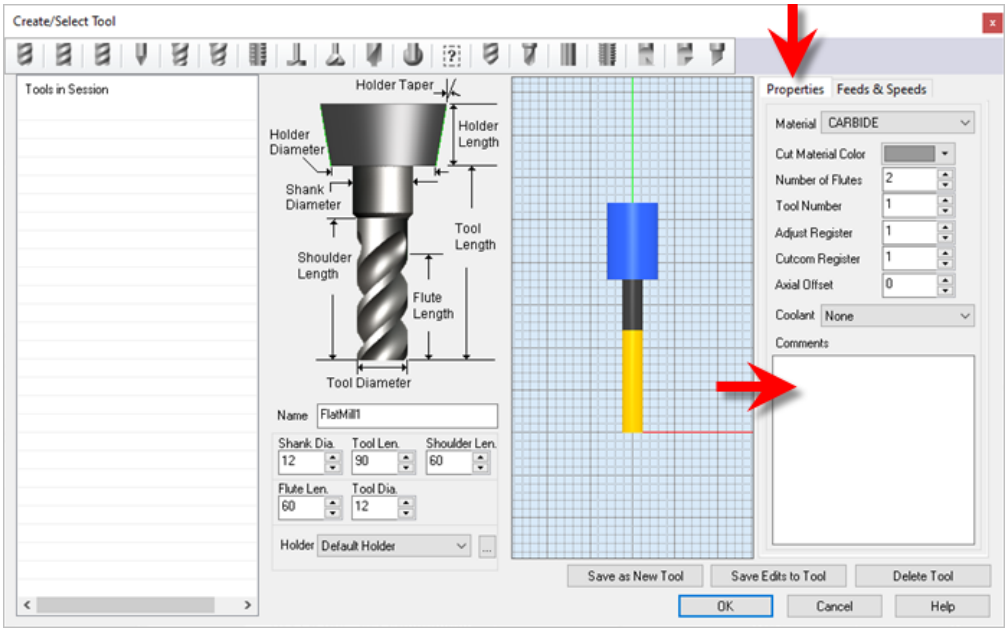
```
(Horizontal Roughing)
N6265 A180.F300.
...
...
```

Posting Tool Comments

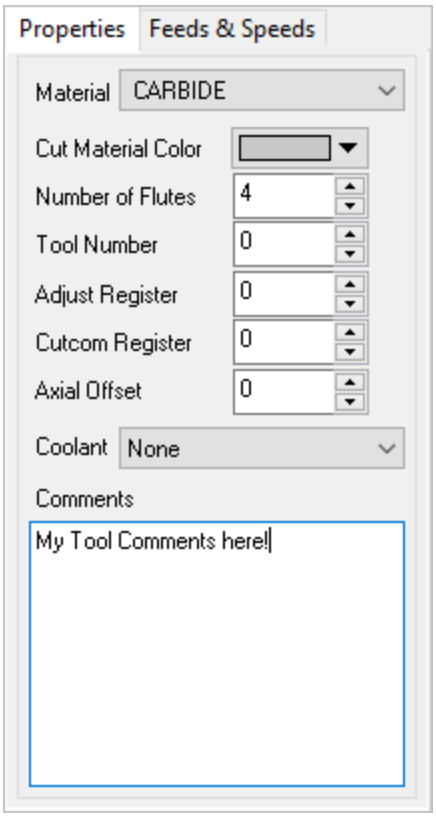
You can add comments associated with a Tool. These Comments are saved with the Tool in your Tool Library. They are also posted to your g-code when the tool is used.

Here are the steps to add Comments to a Tool:

1. Edit the Tool using the Create/Select Tool dialog.
2. Select the Properties tab on the right.

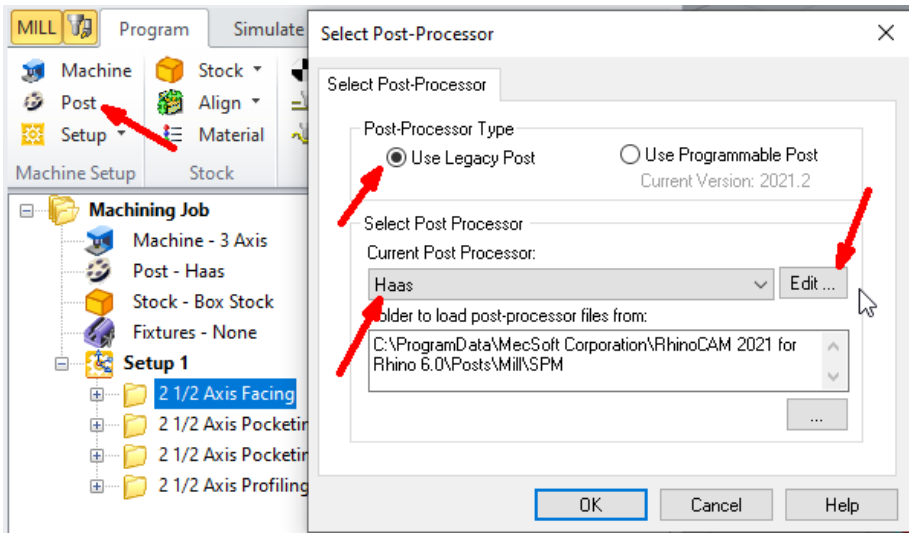


3. Add text to the Comments window.



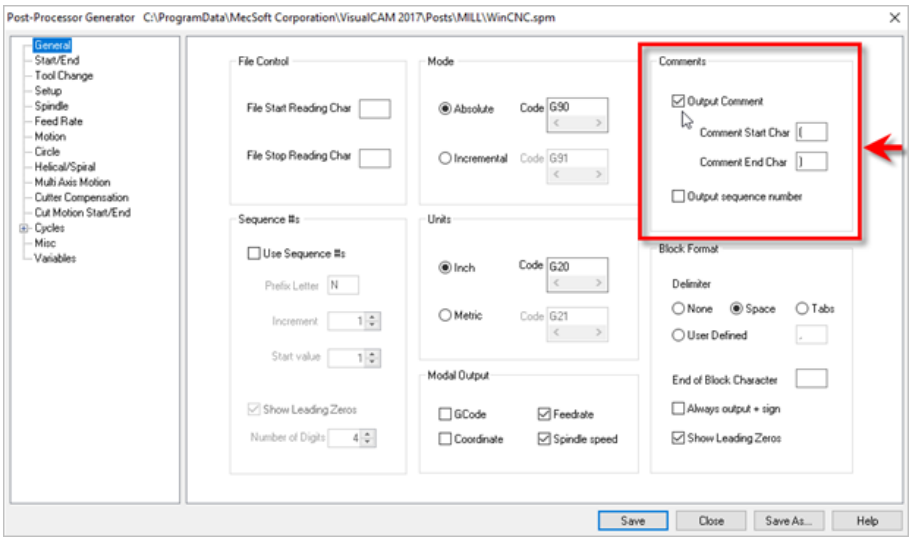
4. Make sure Comments are enabled in your post.

   A. Click on Post (Set Post-Processor Options), then click Edit.



   B. From the Post Processor Generator dialog, select the General tab from the left.

   C. Check the box to Output Comments. You can also change the start and end characters to use.

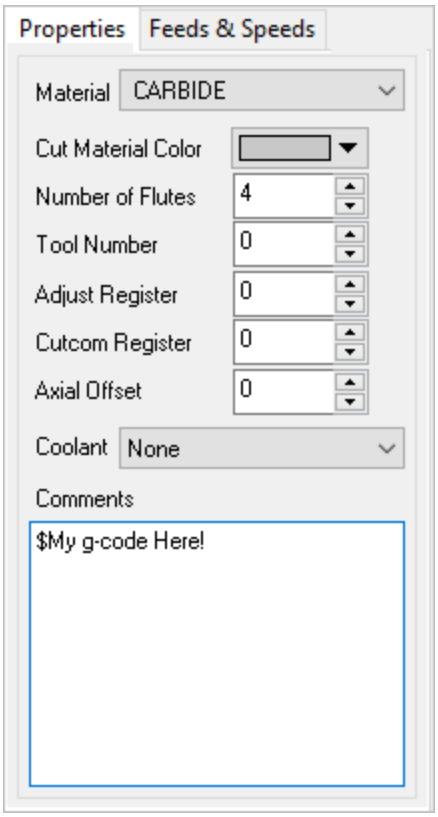


D. Then pick Save or Save As.

5. Now post your operations and see your comments:

```
...
...
G1 X0.5301 Y-0.7171 Z0.7480
G3 X0.7801 Y-0.4671 I0.0000 J0.2500 F2.6
G1 X0.7801 Y-0.2171 Z0.7480 F6.9
G0 Z0.9843
G0 X0.7801 Y-0.2171
(2 1/2 Axis Profiling)
(My Tool Comments Here!)
S18000
G0 Z0.9843
G0 X0.5301 Y-0.7097
G1 X0.5301 Y-0.7097 Z0.7480 F6.9
G1 X0.5873 Y-0.7097 Z0.7480 F3.4
G1 X0.5873 Y-0.6345 Z0.7480
G1 X0.4729 Y-0.6345 Z0.7480
...
...
```

6. If you want to post g-codes instead of comments, just place a $ character prior to the comment in the Create/Select Tools dialog. Adding $ as prefix will skip the comment start & end characters in the posted code.



...
...
G1 X0.4655 Y-0.7171 Z0.7480
G1 X0.5301 Y-0.7171 Z0.7480
G3 X0.7801 Y-0.4671 I0.0000 J0.2500 F2.6
G1 X0.7801 Y-0.2171 Z0.7480 F6.9
G0 Z0.9843
G0 X0.7801 Y-0.2171
(2 1/2 Axis Profiling)
My g-code Here!
S18000
G0 Z0.9843
G0 X0.5301 Y-0.7097
G1 X0.5301 Y-0.7097 Z0.7480 F6.9
G1 X0.5873 Y-0.7097 Z0.7480 F3.4
G1 X0.5873 Y-0.6345 Z0.7480
...
...

Related Topics

Post Processor File Browser

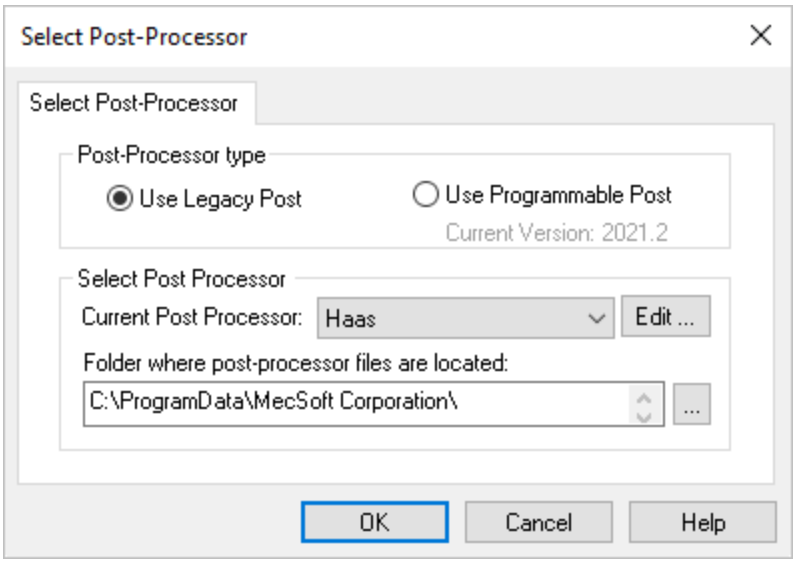
Main Editor

Variable List Dialog

Use Legacy Post

Post Select this option from the Post-Processor Options dialog to use your *"Legacy Post"* when posting toolpath operations. *"Legacy Post"* refers to your post definition (*.spm) file you used prior to version 2021. This option also lists the over 300 existing post-processors. See Current Post Processor below.

Dialog Box: Set Post-Processor Options



Dialog Box: Set Post-Processor Options

Post Processor Type

Use these options to define the post type to use when posting toolpath operations. Each option is documented further in the Post-processor Generator On-Line Help.

Use Legacy Post

Select this option to use your *"Legacy Post"* when posting toolpath operations. *"Legacy Post"* refers to your post definition (*.spm) file you used prior to version 2021. This option also lists the over 300 existing post-processors. See Current Post Processor below.

Use Programmable Post

Select this option to use your *"Programmable Post"* when posting toolpath operations. *"Programmable Post"* refers to your programmable post created with our version 2021 and newer CAM plugins. This option will list only one post processor selection called PostModifier. This post definition is defined using the Programmable Post API.

Current Post Processor

You can change the default post processor by selecting a post from the list of available post processors under Current Post Processor. If Use Legacy Post is selected, the Current Post Processor list will contain MecSoft's standard list of over 300 post processors. If Use Programmable Post is selected, this will list only one post processor selection called PostModifier. This post definition is defined using the Programmable Post API. Refer to the Programmable Post API documentation for more about the API.

Folder where post-processor files are located

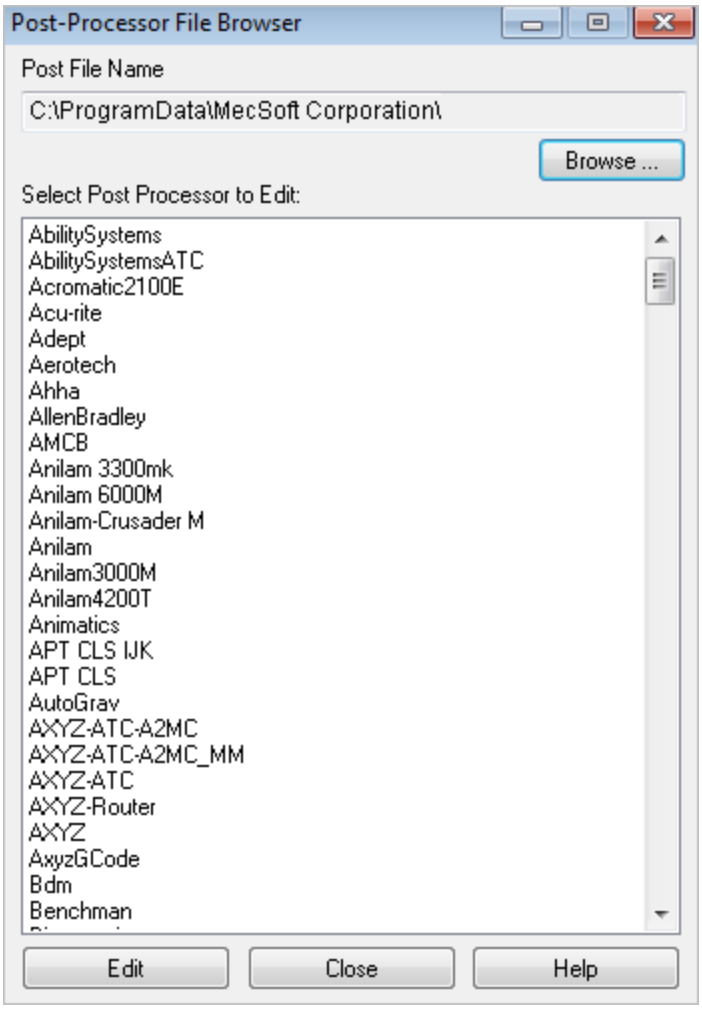
MILL module uses macro files with a *.spm* extension to handle post-processing to different controllers. These files are typically located in the "Posts" directory under the installation folder(*C:\ProgramData\MecSoft Corporation\\Posts\Mill*).

5.1 Dialogs

5.1.1 Post Processor File Browser

The dialog shown below is used to select the post processor file to be edited. (SPM File). The name and location of the post processor file can be either entered in the edit box provided, or can be selected using the browse button. You can double click on the required SPM File to invoke the Editor.

Dialog Box: Post Processor File Browser



Dialog Box: Post Processor File Browser

Related Topics

Introduction

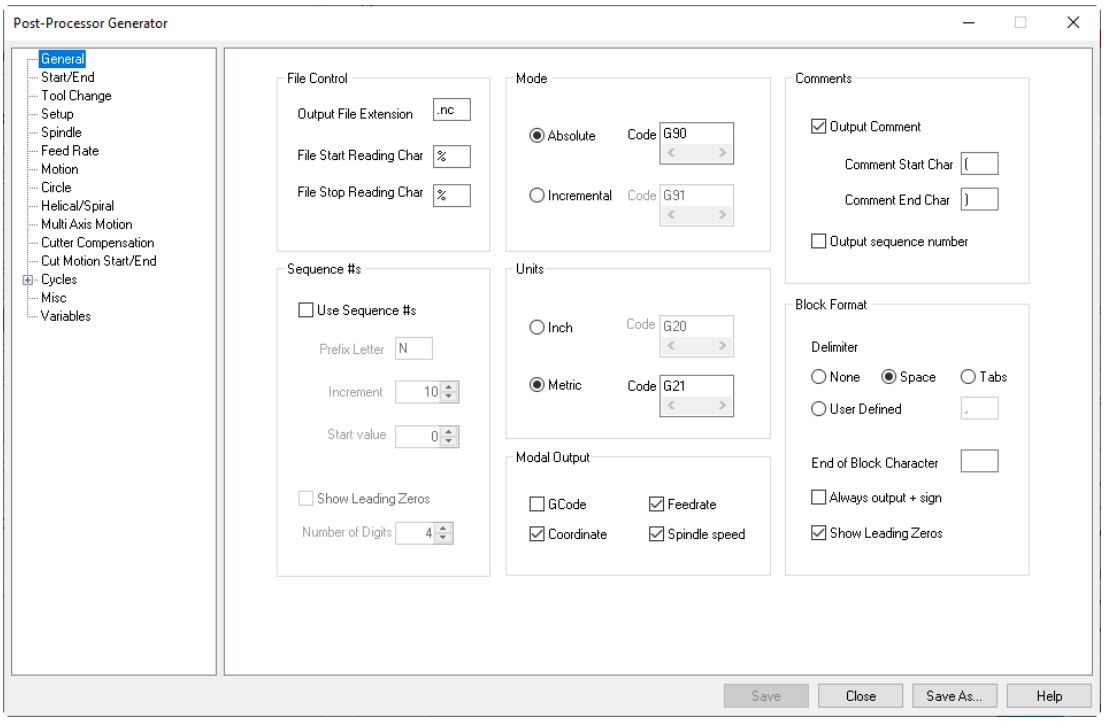
Post Processor Editor

Variable List Dialog

5.1.2 PPG Editor

The RhinoCAM Post-Processor Generator (PPG) Editor is shown below. This Editor is divided into sections (listed on the left) allowing you to define each block type. Refer to the list of sections below.

Dialog Box: Post Processor Generator: Editor



Dialog Box: Post Processor Generator: Editor

PPG Editor Sections

- General
  This folder helps you set up file information, G-Code format, mode and the units of operation.
- Start End
  Start and End code specifier.
- Tool Change
  Load and Tool Change Macro specifier.
- Setup
  Setup change and Rotate Table Setup specifier.
- Spindle
  Spindle code specifier.
- Feed Rate
  Feed Rate specifier.
- Motion
  Motion block specifier
- Circle
  Circle block specifier.
- Helical/Spiral
  Helical and Spiral motion block specifier.

- Multi Axis Motion
  Multi Axis Motion specifier.
- Cutter Compensation
  Motion blocks for Cutter Compensation Right, Left and Off.
- Cut Motion Start/End
  Cut Motion Start and End macro blocks.
- Cycles
  Cycle G-Code and format specifier.
- Miscellaneous
  Coolant and Compensation code specifier
- Variables
  Lists variables and their values used in post-processing

Related Topics

Introduction

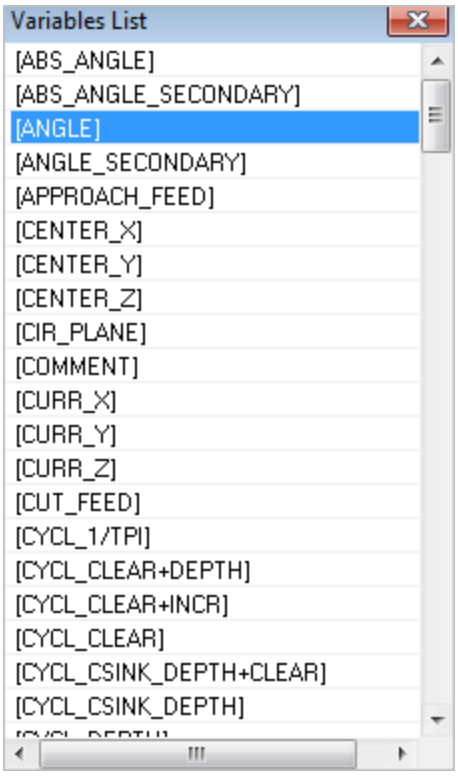
Post Processor File Browser

Variable List Dialog

5.1.3 Variable List Dialog

The Variable List dialog shown below can be invoked by pressing the right mouse button from within edit boxes that are used in setting up startup and termination code for a post-processor. This dialog can be used to add variables to the active edit box for macros.

Dialog Box: Variable List



Dialog Box: Variable List

Related Topics

Post Processor File Browser

Post Processor Editor

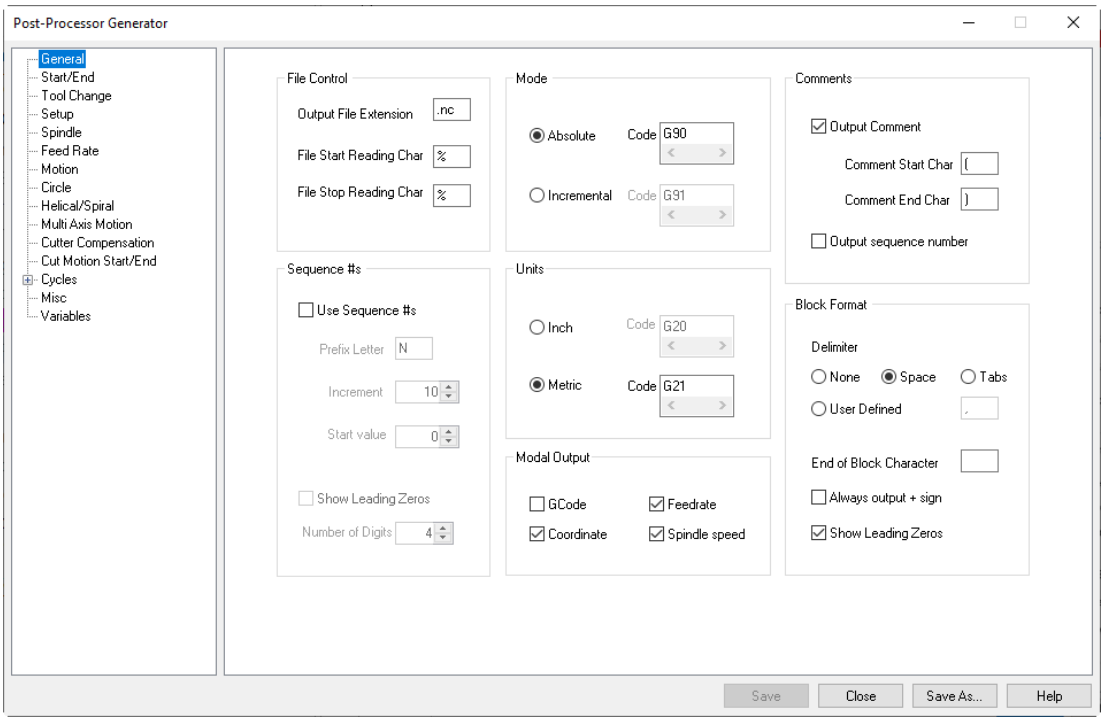
Variables List

5.2 PPG Editor

5.2.1 General

The General section allows you to define the general behavior of the post-processor output. Each editable parameter is described below.

PPG Editor: General



PPG Editor: General

File Control

This section allows you to set the default extension of the output file.

- Output File Extension
  This allows you to set the posted file extension here in the PPG. Once set, go to the Post Preferences dialog (CAM Preferences > Post) and check the box "File Extension from Post Processor". The file extension will also be shown in that dialog.
- File Start Reading Char
  The user can also set up an optional File Start character. This file start character will be the first character written to the output file.
- File Stop Reading Char
  Similarly the file end character will be the last character written to the output file. Most standard controllers look for a percent sign (%) as this last character.

Mode

Coordinate values can be set to be either Absolute or Incremental. In the Absolute mode, coordinate values (X, Y, Z, I, J, K) are output as absolute values. In the Incremental mode, coordinate values are output as incremental values.

Comments

This section defines the general format of commentary blocks.

- Output Comment: Set parameter to output comments
- Comment Start Char: Comment start character
- Comment End Char: Comment end character
- Output Sequence Number: Toggle sequence number for comments

If you want the comments to be output just as it is without the Start Character and the End Character, then in RhinoCAM, when you are inserting a comment, type in a $ sign in front of it, like:

$ Comment

Sequence #s

This section allows you to control the format of sequence numbers to the output file. This option is available for all G-code lines except Tool change Macro and Start and End Code.

- Use Sequence #s
  Sequence number output can be turned on or off.
- Prefix Letter
  When on, a prefix letter can optionally be added at the beginning of every sequence number.
- Increment
  Sequence numbers can also be output in increments rather than sequentially. This increment value can be specified here.
- Start Value
  This specifies the starting value for the sequence numbers.
- Show Leading Zeros
  In addition the number of digits output as well as presence/absence of leading zeros in the sequence numbers can be controlled.

Units

This tells the post-processor the units of the output file by outputting a units code that can be defined here. Output units can either be in the English system (inches) or in the Metric system (mm).

- Inch
  Sets the units to Inches and post the code G20 by default. This code can be changed if desired.
- Metric
  Sets the units to Millimeters and post the code G21 by default. This code can be changed if desired.

Block Format

This section defines the general format of all output blocks. Each of the options is described below.

- Delimiter:
  This is the delimiter used between G-codes.
  Example where delimiter is set to [D]: G01[D]X1.0[D]Y2.0[D]Z3.0[D]S3000M03[D]F20

  None: No delimiter is posted in the output.
  Space: A space character is output as the delimiter
  Tab: A tab character is output as the delimiter
  User Defined: Select this option and then enter the delimiter character to use.
- End of Block Character: End of block character to output
- Always output +sign: Outputs a '+' (no quote marks included) for positive values.
- Show Leading Zeros: Check this box to include leading zeros in the posted code. Example (G01 X+1.0 Y+1.0 Z+1.0)

Modal Output

The Post processor generator allows the following parameters to be set as modal or non-modal. The modal output setting will output the value of a variable only if it is different from the value that was last output.

- Gcode: G-Code modal option sample
- Coordinate: Coordinates modal option sample
- Feedrate: Feed Rate modal option sample
- Spindle speed: Spindle Speed modal option sample

An example of non-modal data is shown below. The repeated values are shown in colored text.

S1000M03
G00 X1.0 Y2.0 Z0.0 F10
S2000M03
G01 X1.0 Y2.0 Z3.0 F20
G01 X1.0 Y3.0 Z3.0 F20
S2000M03

Related Topics

Other PPG Editor Sections:

Start/End

Tool Change

Setup

Spindle

Feed Rate

Motion

Circle

Helical/Spiral Cycles

Multi Axis Motion

Cutter Compensation

Cut Motion Start/End

Cycles

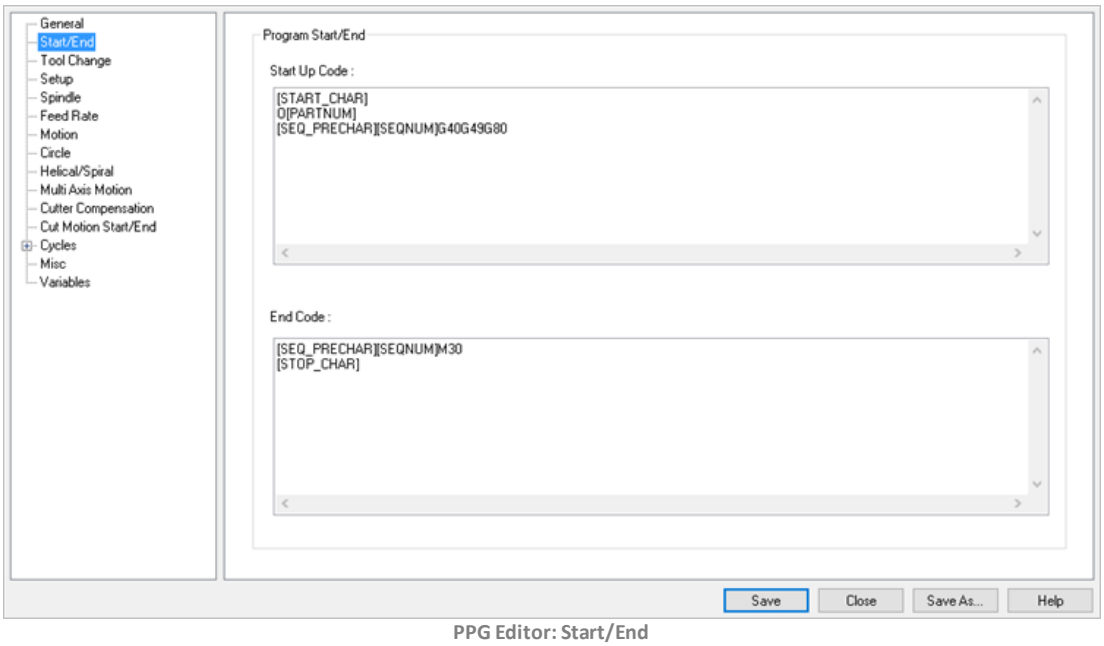
Miscellaneous

Variables

5.2.2 StartEnd

This tab is used to set the Start and End G-Codes.

PPG Editor: Start/End



PPG Editor: Start/End

Startup Code

First macro output in the generated NC file.
Example:
[START_CHAR]
O[PARTNUM]
[SEQ_PRECHAR][SEQNUM]G40G49G80

End Code

Last macro output in the generated NC file.
Example:
[SEQ_PRECHAR][SEQNUM]M30
[STOP_CHAR]

Related Topics

How to edit macros

Other PPG Editor Sections:

General

Tool Change

Setup

Spindle

Feed Rate

Motion

Circle

Helical/Spiral Cycles

Multi Axis Motion

Cutter Compensation

Cut Motion Start/End

Cycles

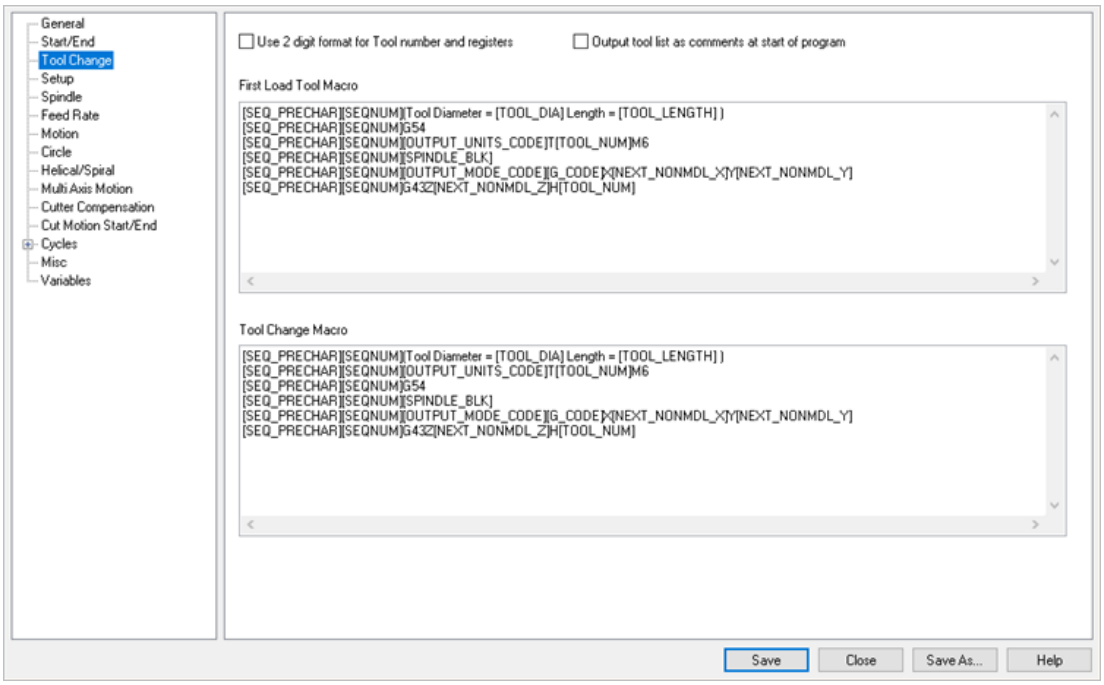
Miscellaneous

Variables

5.2.3 Tool Change

This tab is used to set the Load Tool and Tool Change macros.

PPG Editor: Tool Change



PPG Editor: Tool Change

Use 2 digit format for Tool number and registers

Check this box to use 2-digit format for Tool Numbers and Tool Registers (i.e., T01)

Output tool list as comments at start of program

Check this box to include the tool list as a comment at the start of the program.

Example:

```
%
Ob
N1G40G49G80
( BEGIN TOOL LIST )
( TOOL 1 - FLAT MILL- 1/2 INCH - DESC: 0.5000 DIA, 2 FLUTE,  CARBIDE MAT )
( TOOL 2 - BALL MILL- 1/4 INCH - DESC: 0.2500 DIA, 2 FLUTE,  CARBIDE MAT )
( END OF TOOL LIST )
(Setup 1)
(Horizontal Roughing)
N2(Tool Diameter = 0.5 Length = 4.0 )
N3G54
...
...
...
```

First Load Tool Marco

Macro for the first load tool command.

Example:

```
[SEQ_PRECHAR][SEQNUM](Tool Diameter = [TOOL_DIA] Length = [TOOL_LENGTH] )
[SEQ_PRECHAR][SEQNUM]G54
[SEQ_PRECHAR][SEQNUM][OUTPUT_UNITS_CODE]T[TOOL_NUM]M6
[SEQ_PRECHAR][SEQNUM][SPINDLE_BLK]
[SEQ_PRECHAR][SEQNUM][OUTPUT_MODE_CODE][G_CODE]X[NEXT_NONMDL_X]
Y[NEXT_NONMDL_Y]
[SEQ_PRECHAR][SEQNUM]G43Z[NEXT_NONMDL_Z]H[TOOL_NUM]
```

Tool Change Macro

Macro for tool change command. (Not including the first load tool.)

Example:

```
[SEQ_PRECHAR][SEQNUM](Tool Diameter = [TOOL_DIA] Length = [TOOL_LENGTH] )
[SEQ_PRECHAR][SEQNUM][OUTPUT_UNITS_CODE]T[TOOL_NUM]M6
[SEQ_PRECHAR][SEQNUM]G54
[SEQ_PRECHAR][SEQNUM][SPINDLE_BLK]
[SEQ_PRECHAR][SEQNUM][OUTPUT_MODE_CODE][G_CODE]X[NEXT_NONMDL_X]
Y[NEXT_NONMDL_Y]
[SEQ_PRECHAR][SEQNUM]G43Z[NEXT_NONMDL_Z]H[TOOL_NUM]
```

Related Topics

How to edit macros

Other PPG Editor Sections:

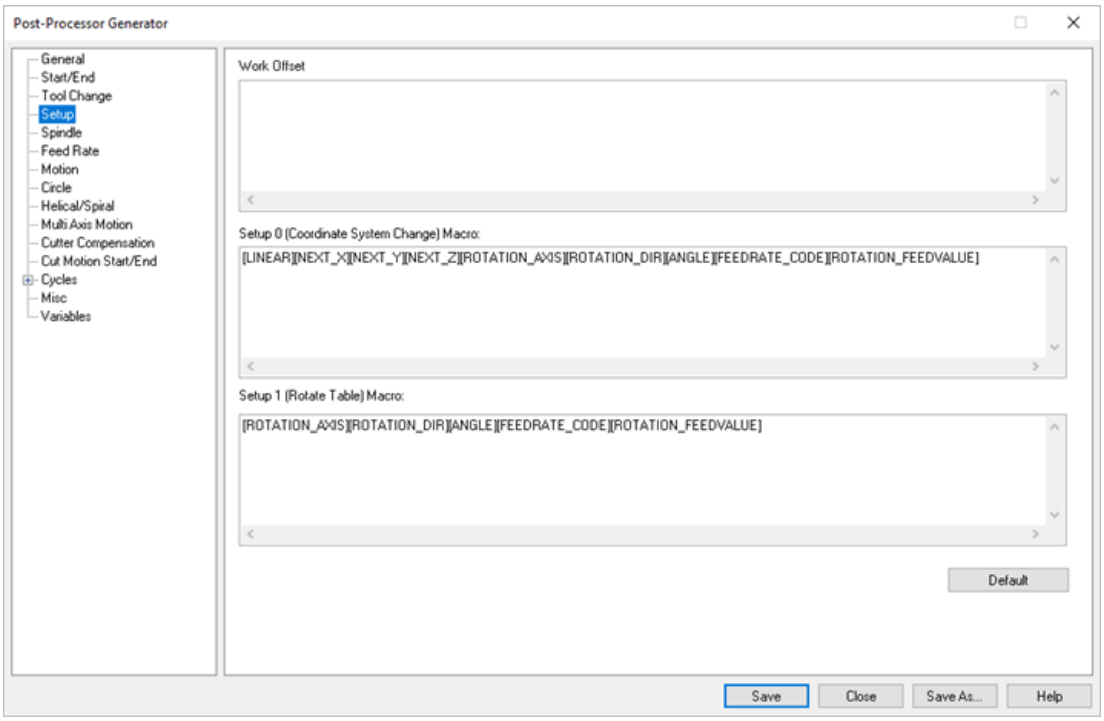
- General
- Start/End
- Setup
- Spindle
- Feed Rate
- Motion
- Circle
- Helical/Spiral Cycles
- Multi Axis Motion
- Cutter Compensation
- Cut Motion Start/End
- Cycles

Miscellaneous

Variables

5.2.4 Setup

PPG Editor: Setup



PPG Editor: Setup

Work Offset

You can define macros here that will control the format of each Work Offset in the posted g-code file.

Setup 0 (Coordinate System Change) Macro

Macro to execute before every Setup change.

Example:

[LINEAR][NEXT_X][NEXT_Y][NEXT_Z][ROTATION_AXIS][ROTATION_DIR][ANGLE][FEEDRATE_CODE][ROTATION_FEEDVALUE]

Setup 1 (Rotate Table) Macro

Macro to execute before every Rotate Table Setup change.

Example:

[ROTATION_AXIS][ROTATION_DIR][ANGLE][FEEDRATE_CODE][ROTATION_FEEDVALUE]

Related Topics

Other PPG Editor Sections:

General

Start/End

Tool Change

Spindle

Feed Rate

Motion

Circle

Helical/Spiral Cycles

Multi Axis Motion

Cutter Compensation

Cut Motion Start/End

Cycles

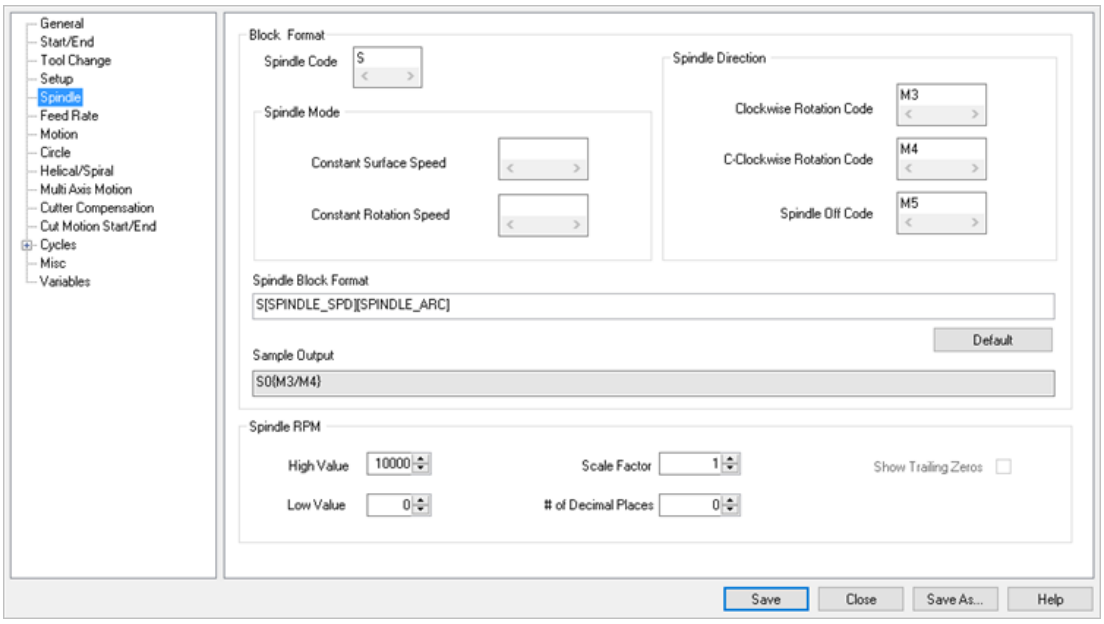
Miscellaneous

Variables

5.2.5 Spindle

This tab is used to set parameters for controlling the spindle.

PPG Editor: Spindle



PPG Editor: Spindle

Block Format

Sets the spindle block format.

- Spindle Code: Register used for the feed rate value.
- Spindle Direction: Specify the direction code for the spindle

  Clockwise Rotation Code: Clockwise spindle code

  C-Clockwise Rotation Code: Counter clockwise spindle code

  Spindle Off Code: Spindle off code
- Spindle Mode:

  Constant Surface Speed:

  Constant Rotation Speed:
- Spindle Block Format: Defines the block format for the spindle. Example: S[SPINDLE_SPD][SPINDLE_ARC]
- Default: Reset all values in this section to their system defaults.
- Sample output: This field displays sample output of the spindle block. It is a non-editable field.

Spindle RPM

- High Value: Maximum spindle value. The spindle RPM is capped to this high value.

- Low Value: Minimum spindle value. The spindle RPM is capped to this low value.
- Scale Factor: Scale factor of Spindle value.
- # of Decimal Places: Number of digits output after the decimal point
- Show Trailing Zeros: Show trailing zeros when # of Decimal Places is set to a value greater than zero.

Related Topics

Other PPG Editor Sections:

General

Start/End

Tool Change

Setup

Feed Rate

Motion

Circle

Helical/Spiral Cycles

Multi Axis Motion

Cutter Compensation

Cut Motion Start/End

Cycles

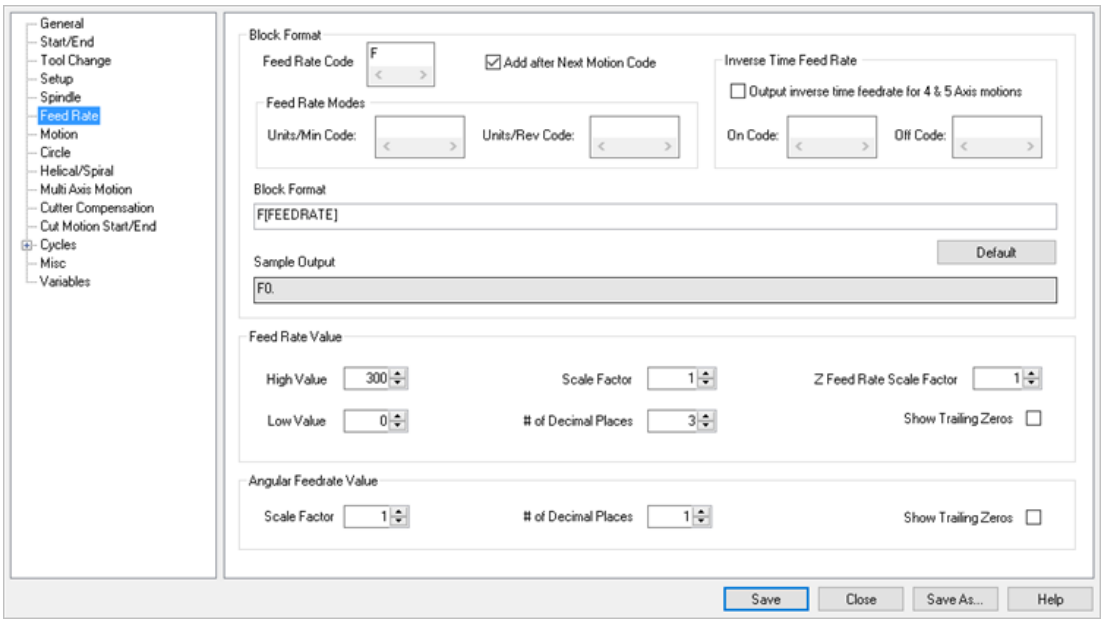
Miscellaneous

Variables

5.2.6 Feed Rate

This tab sets the feeds and speeds parameters

PPG Editor: Feed Rate



PPG Editor: Feed Rate

Block Format

Characters for the feed rate block:

- Feed Rate Code: Register used for the feed rate value
- Add after Next Motion Code:
- Feed Rate Modes

  Units/Min Code: Typically a G94

  Unites/Rev Code: Typically a G95

- Inverse Time Feed Rate

  When Output inverse time feedrate for 4 & 5 Axis motions is selected the On Code is output at the start of the operation and Off Code at the end.

  You turn on the Inverse Time Feedrate using a checkbox in the Post/Feedrate tab. You can set up the code for On and Off in the same tab. Once this is set, then all 4 and 5 axis motions will be processed with this feedrate being output.

  The way the feedrate is computed is as follows:

  First the distance traveled by the tool is computed for each move = dist

  Then the time taken for traversing the move is computed thus: timeForTravel = dist / cutFeedRate;

  Then the inverse time feedrate is = 1.0/timeForTravel;

And it is output for each move if different from previous move.

On Code: G93

Off Code: G94

- Block Format: Format for the feed rate block
- Default: Reset all values in this section to their system defaults.
- Sample Output: Sample output displays sample code of the feed rate. It is a non-editable field.

Feed Rate values

Parameters for the adjustment of feed rate value:

- High Value: Maximum Feed rate value.
- Low Value: Minimum Feed rate value.
- Scale Factor: Scale factor of Feed rate value.
- # of Decimal Places: Number of digits output after the decimal point.
- Z Feed Rate Scale Factor: Feed rate scale factor for Z Feed Rate (use [ZFEEDRATE] variable to get the Z feed rate)
- Show Trailing Zeros: Show trailing zeros when # of Decimal Places is set to a value greater than zero.

Angular Rate values

- Scale Factor: Enter the scale factor for angular feed rate values.
- # of Decimal Places: Number of digits output after the decimal point
- Show Trailing Zeros: Show trailing zeros when # of Decimal Places is set to a value greater than zero.

Related Topics

Other PPG Editor Sections:

General

Start/End

Tool Change

Setup

Spindle

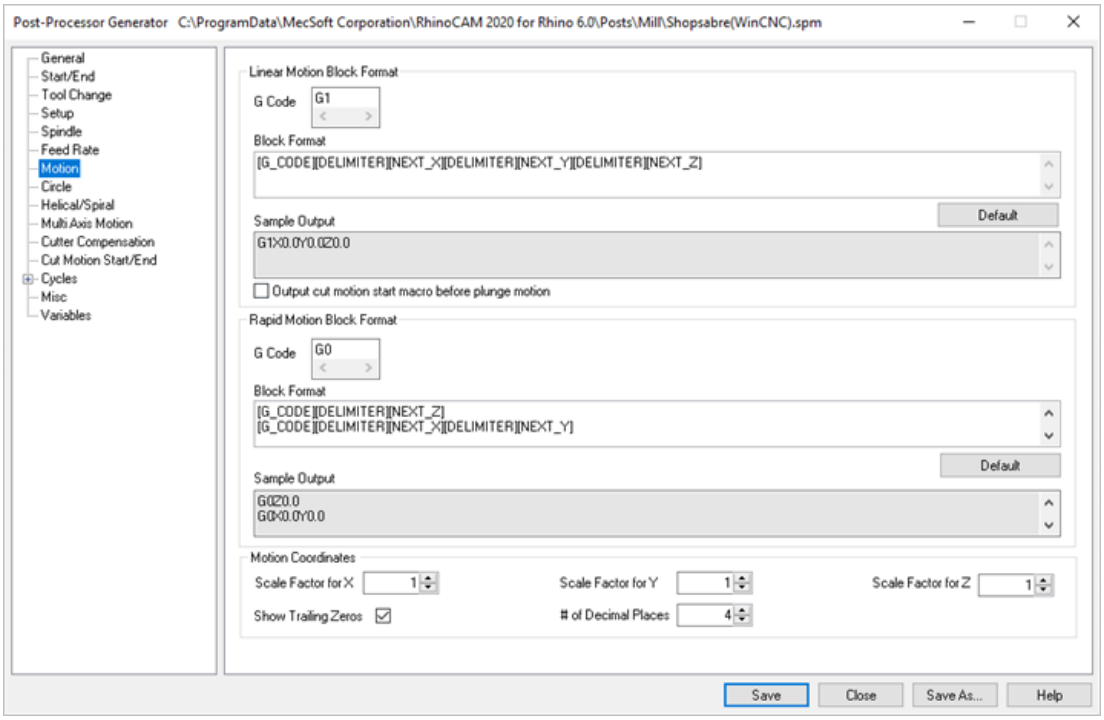
Motion

Circle

5.2.7 Motion

This tab is used to define the linear motion outputs of a post-processor. You can use multiple lines to define the Block Format in both the Linear Motion Block and the Rapid Motion Block.

PPG Editor: Motion



PPG Editor: Motion

Linear Motion Block

Used to define the output format for the cut motions (e.g.: G1). The sample output can be seen in the non-editable Sample Output field. Multiple lines of code are supported.

- G Code: Typically a G1

- Block Format: Example: [G_CODE][NEXT_X][NEXT_Y][NEXT_Z]
- Default: Reset all values in this section to their system defaults.
- Sample Output: This displays how the posted output will look.
- Output cut motion start macro before plunge motion: Checking this box will output the cut motion start macro defined under Cut Motion Start/End section before the plunge motion in the posted code.

Rapid Motion Block Format

Used to define the output format for the rapid motions (e.g.: G0). The sample output can be seen in the non-editable Sample Output field. Multiple lines of code are supported.

- G Code: Typically G0
- Block Format: Example:
  [G_CODE][NEXT_Z]
  [NEXT_X][NEXT_Y]
- Default: Reset all values in this section to their system defaults.
- Sample Output: This displays how the posted output will look.

Motion Coordinates

- Scale Factor for X, Y, Z: Scale factor of the coordinate values. (Includes circles and cycles)
- Show Trailing Zeros: Show the trailing zeros (e.g. 5.4 is output as 5.4000)
- # of Decimal Places: Number of digits output after the decimal point

Related Topics

Other PPG Editor Sections:

General

Start/End

Tool Change

Setup

Spindle

Feed Rate

Circle

Helical/Spiral Cycles

Multi Axis Motion

Cutter Compensation

Cut Motion Start/End

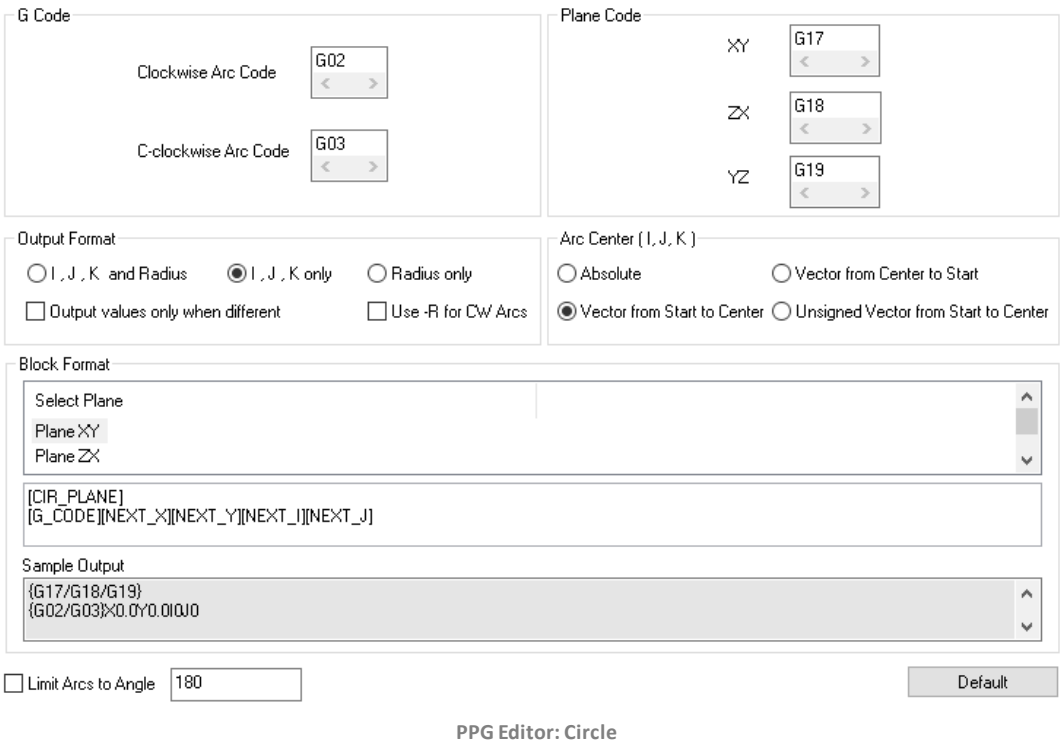
Cycles

Miscellaneous

Variables

5.2.8 Circle

This tab is used to define the circle block output.



PPG Editor: Circle

G Code

Arc Direction Code:

- Clockwise Arc Code: Typically G02
- C-clockwise Arc Code: Typically G03

Plane Code

The G Code for the principal planes in which the Arc motion is output.

- XY: Typically G17
- ZX: Typically G18
- YZ: Typically G19

Output Format

These options help define the output format for the circle command.

- I,J,K and Radius: Output I,J,K and Radius
- I,J,K only: Output only I,J,K.
- Radius only: Output only Radius.
- Output values only when different: Output I,J,K,R values only when different from the previous values.
- Use -R for CW Arcs: Prefixes -R before clockwise arc motions.

Arc Center (I,J,K)

Defines the calculation of the arc center coordinates.

- Absolute: is the absolute center
- Vector from Center to Start: is (Center - Start)
- Vector from Start to Center: is (Start - Center)
- Unsigned vector: is the unsigned distance between center and start.

Block Format

Used to specify the block format for three different planes. Use the default button to see the default values and edit them if needed.

Sample Output

This field displays sample output of the arc motion block. This field is non-editable. It indicates the parameters selected from the available options.

Limit Arcs to Angle

This is used to limit arcs to a certain angle. This is helpful for certain types of controllers which cannot output arcs greater than a certain angle

Default

Reset all values in this section to their system defaults.

Related Topics

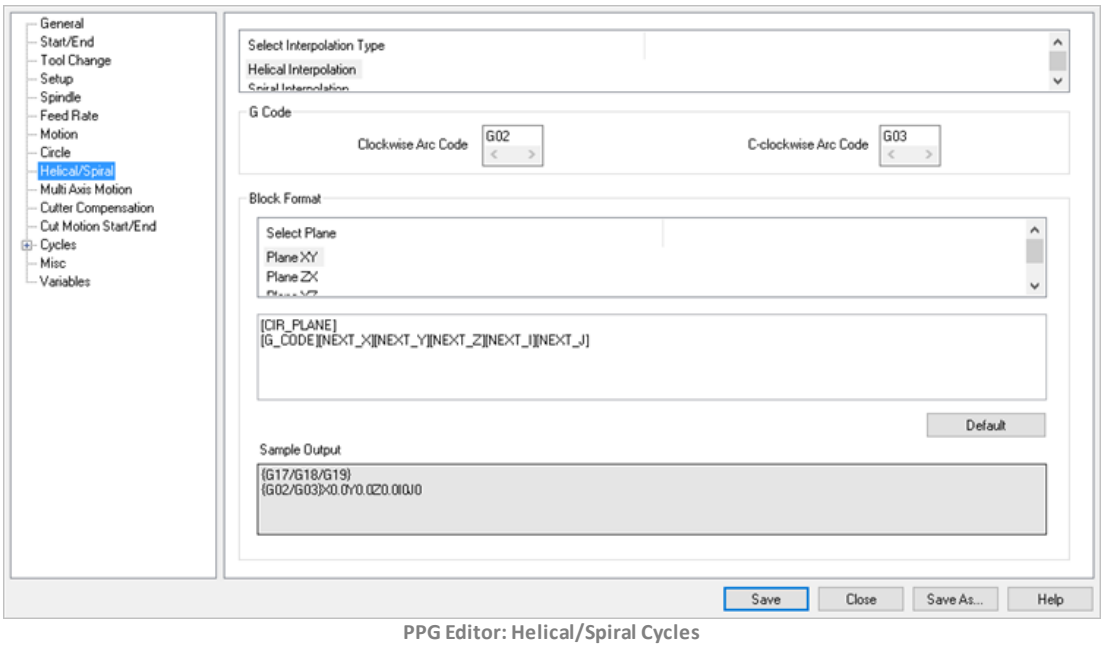
Other PPG Editor Sections:

- General
- Start/End
- Tool Change
- Setup
- Spindle
- Feed Rate
- Motion
- Helical/Spiral Cycles
- Multi Axis Motion
- Cutter Compensation
- Cut Motion Start/End
- Cycles
- Miscellaneous
- Variables

5.2.9 Helical/Spiral

This tab is used to define the Helix and Spiral block output. For both these cycles, the Arc Center and the Plane Code are the same as those defined in the Circle Section. Please define those first before defining these cycles.

PPG Editor: Helical/Spiral Cycles



PPG Editor: Helical/Spiral Cycles

Select Interpolation Type

Used to select the Interpolation type to define the parameters for the Helical Cycles or the Spiral Cycles.

G Code

Used to define the Clockwise and the Counter Clockwise Codes for the Helical or Spiral Cycles

Block Format

Used to specify the block format for three different planes. Use the default button to see the default values and edit them if needed.

Default

Reset all values in this section to their system defaults.

Sample Output

This field displays sample output of the Helical/Spiral motion block. This field is non-editable. It indicates the parameters selected from the available options.

Related Topics

Other PPG Editor Sections:

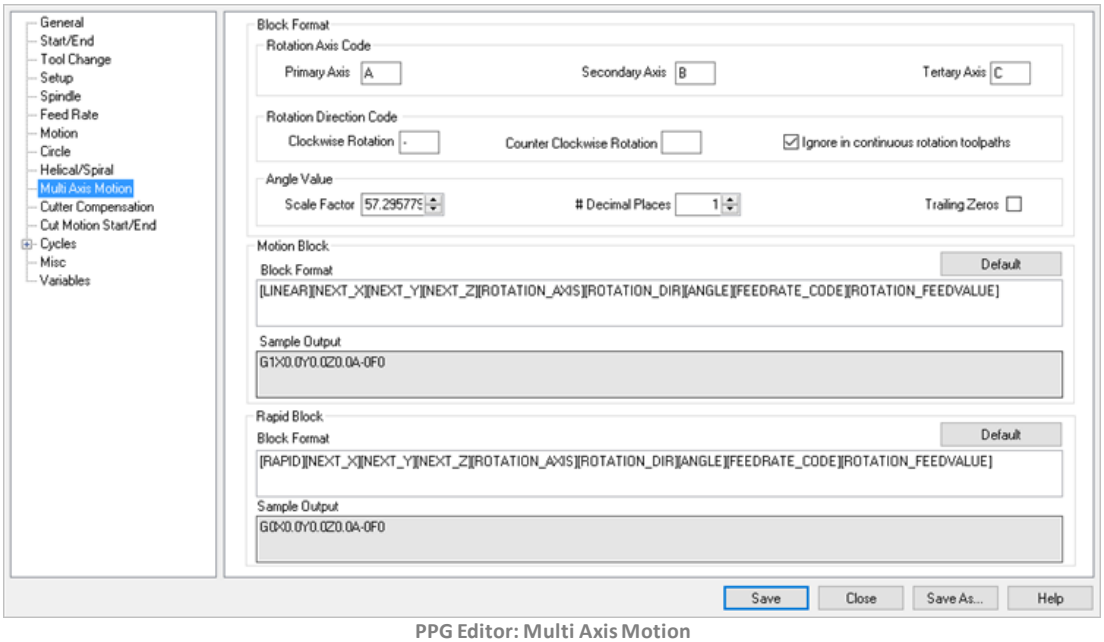
General

Start/End

5.2.10 Multi Axis Motion

This tab is used to set 4th Axis Motion Section parameters.

PPG Editor: Multi Axis Motion



PPG Editor: Multi Axis Motion

Block Format

Rotation Axis Code

Characters for Rotation Axis Code.

- Primary Axis: Sets the primary axis code. Typically A
- Secondary Axis: Sets the secondary axis code. Typically B
- Tertary Axis: Sets the Tertary axis code. Typically C

Rotation Direction Code (Only for Rotate Table)

Characters for Rotation Direction Code

- Clockwise Rotation: Clockwise rotation code
- Counter Clockwise Rotation: Counter Clockwise rotation code
- Ignore in continuous rotation toolpaths: Check this box to ignore rotation direction in 4 Axis continuous rotation toolpaths.

Angle Values

Angle Value for 4th Axis Motion

- Scale Factor: Scale Factor for Angle Value (the angle is in radians, to convert to degrees use a scale factor of 57.295779513082
- # Decimal Places: No of Decimal Places
- Trailing Zeros: Number of Trailing Zeros after Decimal places

Motion Block

Motion Code for 4th Axis Motion

- Block Format: Helps to define the output format for the 4th Axis motion code.
- Sample Output: Sample output displays sample code of the 4th Axis motion. It is a non-editable field.
- Default: Reset all values in this section to their system defaults.

Rapid Block

Rapid Code for 4th Axis Motion

- Block Format: Helps to define the output format for the 4th Axis rapid code
- Sample Output: Sample output displays sample code of the 4th Axis rapids. It is a non-editable field.
- Default: Reset all values in this section to their system defaults.

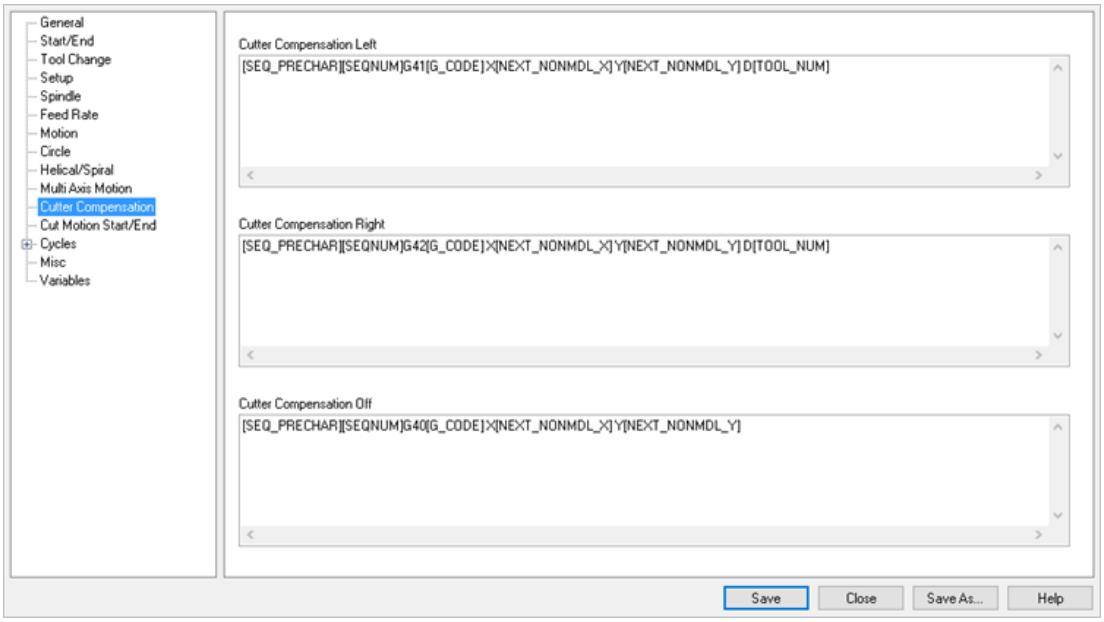
Related Topics

Other PPG Editor Sections:

- General
- Start/End
- Tool Change
- Setup
- Spindle
- Feed Rate
- Motion
- Circle
- Helical/Spiral Cycles
- Cutter Compensation
- Cut Motion Start/End
- Cycles
- Miscellaneous
- Variables

5.2.11 Cutter Compensation

PPG Editor: Cutter Compensation



PPG Editor: Cutter Compensation

Cutter Compensation Left

Macro used to define the output when the cutter compensation left is detected in the output

Cutter Compensation Right

Macro used to define the output when the cutter compensation right is detected in the output

Cutter Compensation Off

Macro used to define the output when the cutter compensation cancel (off) is detected in the output

Related Topics

Other PPG Editor Sections:

General

Start/End

Tool Change

Setup

Spindle

Feed Rate

Motion

Circle

Helical/Spiral Cycles

Multi Axis Motion

Cut Motion Start/End

Cycles

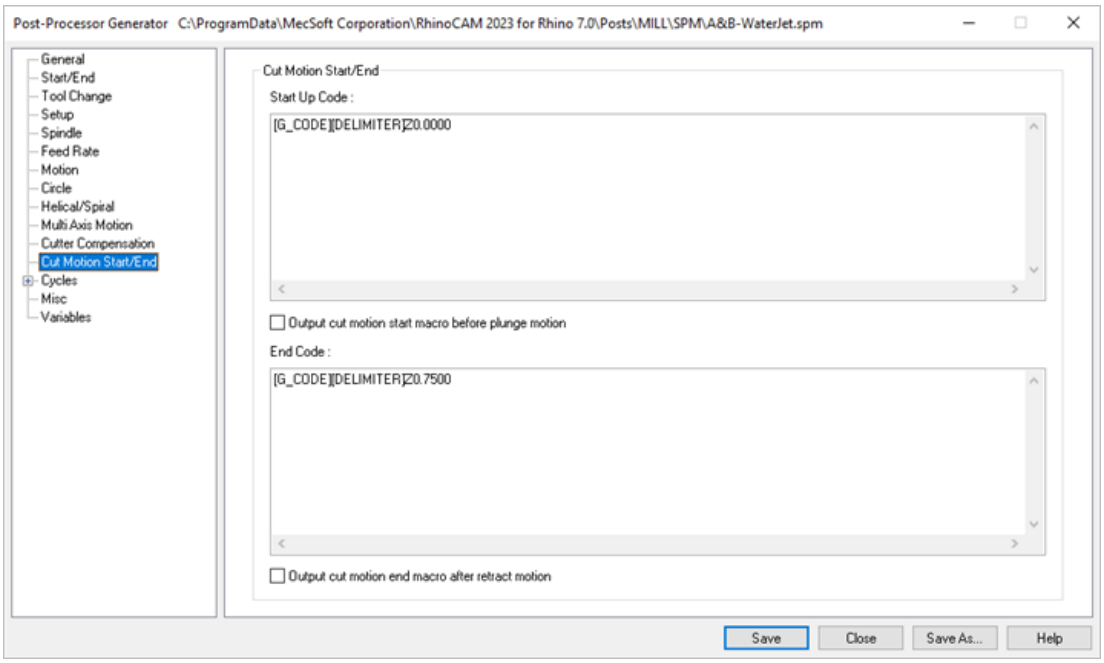
Miscellaneous

Variables

5.2.12 Cut Motion Start/End

The Cut Motion Start/End dialog allows you to define Start and End cut motion codes.

PPG Editor: Cut Motion Start/End



PPG Editor: Cut Motion Start/End

Output cut motion start macro before plunge motion

You can have the Cut Motion Start Up Code posted before plunge motions by checking the box.

Output cut motion end macro after retract motion

You can have the Cut Motion End Macro posted after the retract motions by checking this box.

Related Topics

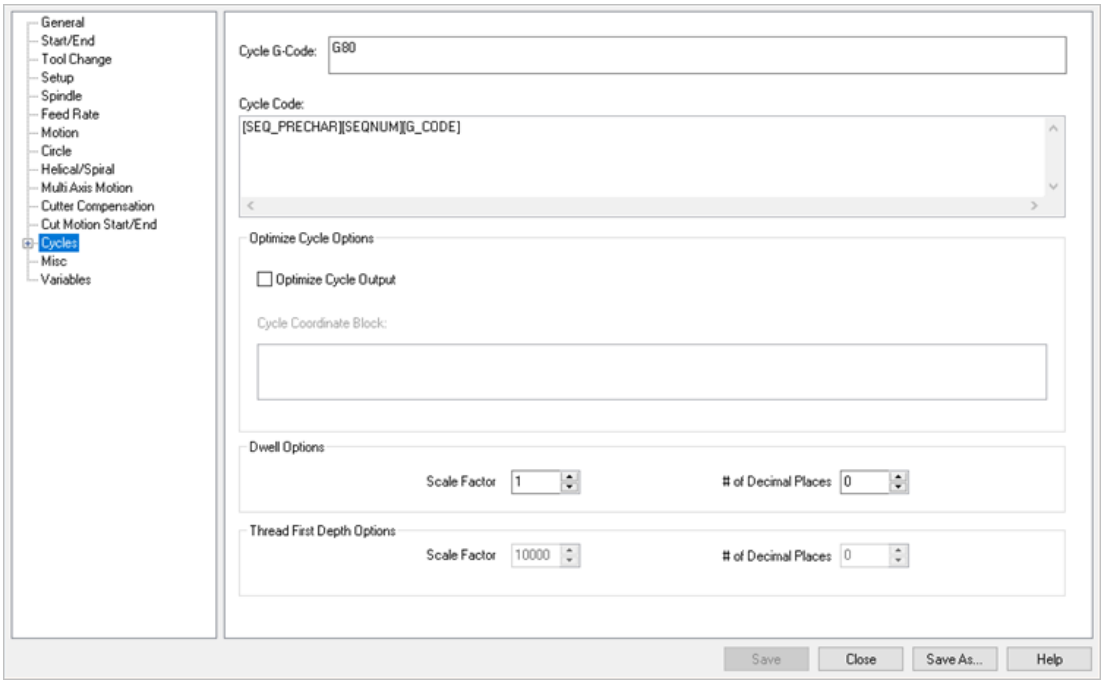
Other PPG Editor Sections:

- General
- Start/End
- Tool Change
- Setup
- Spindle
- Feed Rate
- Motion
- Circle
- Helical/Spiral Cycles
- Multi Axis Motion

5.2.13 Cycles

This tab is to set the cycle parameters in the RhinoCAM Post-Processor.

PPG Editor: Cycles



PPG Editor: Cycles

Cycle G-Code

Macros for the cycle commands. It represents the selected G-code value that defines each specific cycle. This value is displayed in the edit box below all the available options. It can be changed if required.

Cycle Code

The following Cycles are supported:

- Cycle Off (G80)
- Standard Drill (G81)
- Standard Drill with Dwell On (G82)
- Deep (G83)

- Break Chip (G87)
- Counter Sink (G82)
- Tap (Clockwise) (G84)
- Tap (C-Clockwise) (G84)
- Peck Tapping (Clockwise) (G84)
- Peck Tapping (C-Clockwise) (G84)
- Rigid Tap (Clockwise) (T00)
- Rigid Tap (C-Clockwise) (T01)
- Bore (Drag) Dwell Off (G85)
- Bore (Drag) Dwell On (G89)
- Bore (No Drag) Dwell Off Orient On (G76)
- Bore (No Drag) Dwell On Orient On (G76)
- Bore (No Drag) Dwell Off Orient Off (G86)
- Bore (No Drag) Dwell On Orient Off (G86)
- Bore (Reverse) Dwell Off (G87)
- Bore (Manual) Dwell On (G88)
- Bore (Reverse) Dwell Off (G77)
- Bore (Reverse) Dwell On (G77)
- User Defined Drill Cycle 1
- User Defined Drill Cycle 2
- User Defined Drill Cycle 3
- User Defined Drill Cycle 4
- User Defined Tap Cycle 1
- User Defined Tap Cycle 2
- User Defined Tap Cycle 3
- User Defined Tap Cycle 4
- User Defined Bore Cycle 1
- User Defined Bore Cycle 2
- User Defined Reverse Bore Cycle 1
- User Defined Reverse Bore Cycle 2
- Turn Thread Cycle Automatic
- Turn Thread Cycle Box Cycle
- Turn Thread Cycle Single Block
- Machine Control Cycle 1
- Machine Control Cycle 2
- Machine Control Cycle 3
- Machine Control Cycle 4

Optimize Cycle Options

Optimize Cycle output will define the cycle format only once and will output the X,Y values for all the other holes. This will result in significant reduction in the file size for output.

- Optimize Cycle Output: Enables the optimized cycle options.

- Cycle Coordinate Block: Specific cycle block variables to use.

Dwell Options

- Scale Factor for Dwell: Sets the scale factor for dwell output.
- # of Decimal Places: Sets the number of decimal places for output.

Thread First Depth Options

- Scale Factor: Sets the scale factor for the first thread depth.
- # of Decimal Places: Sets the number of decimal places for output.

Related Topics

How to edit macros

Other PPG Editor Sections:

General

Start/End

Tool Change

Setup

Spindle

Feed Rate

Motion

Circle

Helical/Spiral Cycles

Multi Axis Motion

Cutter Compensation

Cut Motion Start/End

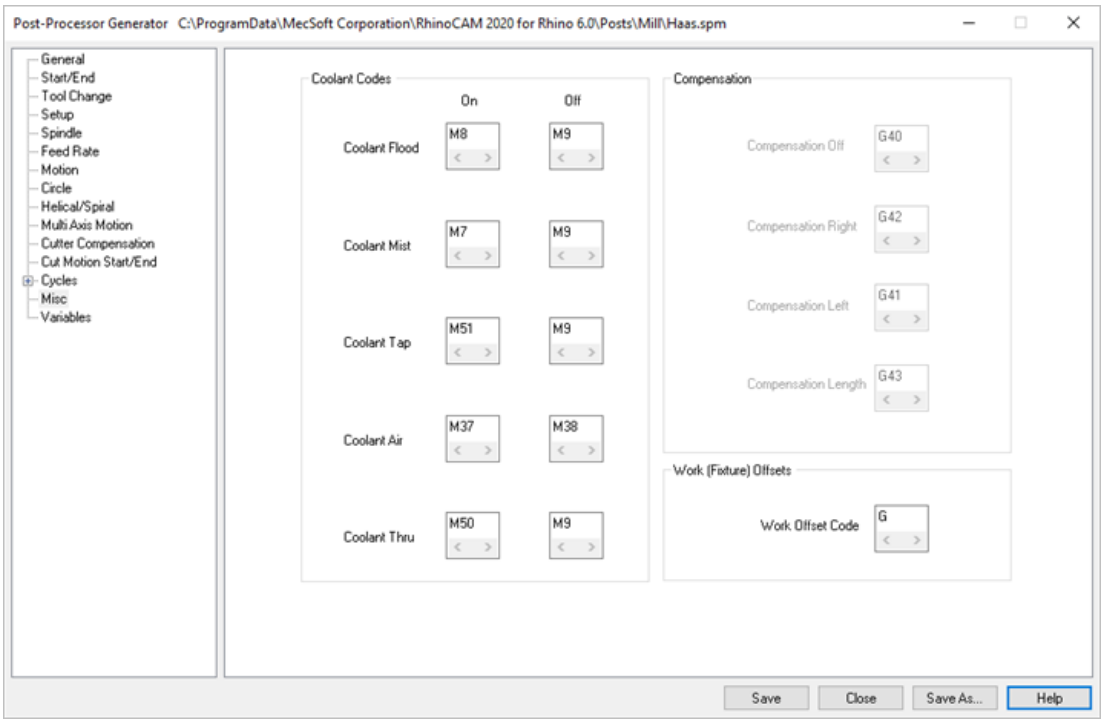
Miscellaneous

Variables

5.2.14 Miscellaneous

This tabbed dialog is used to set miscellaneous parameters. The various miscellaneous parameters are shown in the dialog.

PPG Editor: Miscellaneous



PPG Editor: Miscellaneous

Coolant Codes

- Coolant Flood: Coolant Flood On/Off code
- Coolant Mist: Coolant Mist On/Off code
- Coolant Tap: Coolant Tap On/Off code
- Coolant Air: Coolant Air On/Off code
- Coolant Thru: Coolant Thru On/Off code

Compensation

- Compensation Off: (This is not editable - reserved for future releases)
- Compensation Left: (This is not editable - reserved for future releases)
- Compensation Right: (This is not editable - reserved for future releases)
- Compensation Length: (This is not editable - reserved for future releases)

Work (Fixture) Offsets

- Work (Fixture) Offsets: Sets the work offset prefix code. Typically G. You can enable the output of the work offset code by selecting Output Work Offset from the Work Zero dialog and entering the offset code. For example, enabling and entering 54 in the Work Zero dialog activates this portion of the post and prefixes the code with this value. Example: G54.

Related Topics

Other PPG Editor Sections:

General

Start/End

Tool Change

Setup

Spindle

Feed Rate

Motion

Circle

Helical/Spiral Cycles

Multi Axis Motion

Cutter Compensation

Cut Motion Start/End

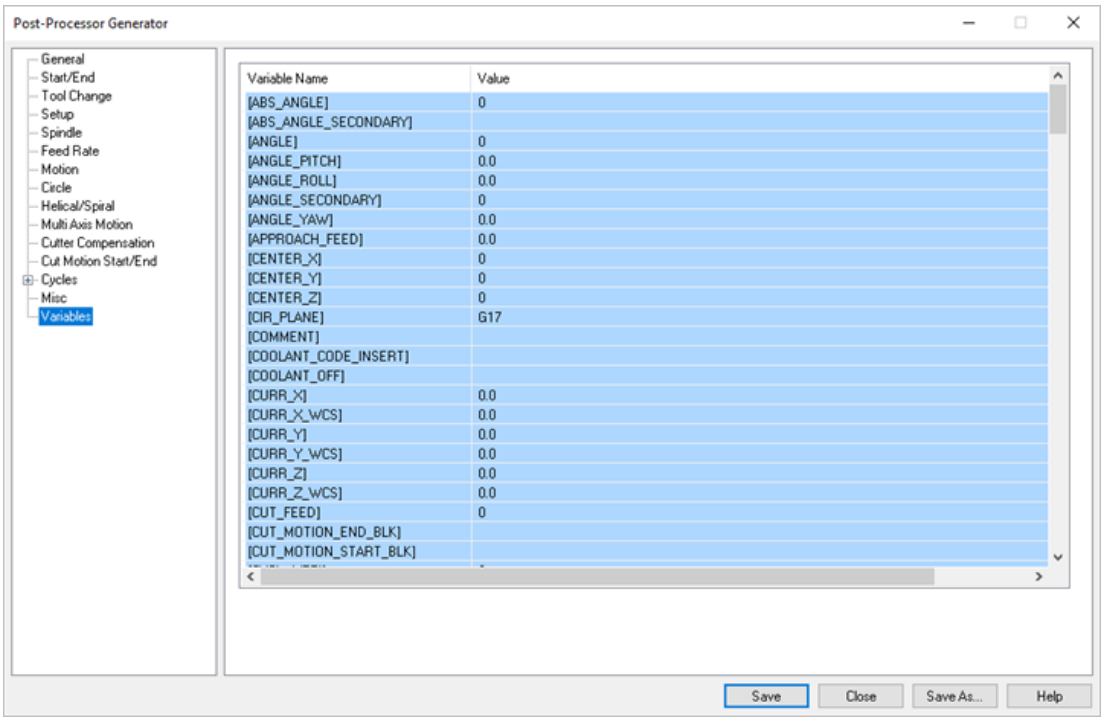
Cycles

Variables

5.2.15 Variables

This tab lists all the variables used in the macros in the RhinoCAM Post Processor.

PPG Editor: Variables



PPG Editor: Variables

Variables

The variables that can be used are listed in the table below.

The variables that start with “CYCL_” are used only for the cycles commands.

Variable	Comments
[ABS_ANGLE]	
[ABS_ANGLE_SECONDARY]	
[ANGLE]	
[ANGLE_PITCH]	
[ANGLE_ROLL]	
[ANGLE_SECONDARY]	
[ANGLE_YAW]	
[APPROACH_FEED]	
[CENTER_X]	
[CENTER_Y]	

[CENTER_Z]	
[CIR_PLANE]	
[COMMENT]	Output comments.
[COOLANT_CODE_INSERT]	
[COOLANT_OFF]	
[CURR_X]	The X coordinate of current point.
[CURR_X_WCS]	The X coordinate of current point in World Coordinates
[CURR_Y]	The Y coordinate of current point.
[CURR_Y_WCS]	The Y coordinate of current point in World Coordinates
[CURR_Z]	The Z coordinate of current point.
[CURR_Z_WCS]	The Z coordinate of current point in World Coordinates
[CUT_FEED]	
[CUT_MOTION_END_BLK]	
[CUT_MOTION_START_BLK]	
[CYCL_1/TPI]	1/TPI (only for TAP cycle)
[CYCL_CLEAR+DEPTH]	Drill Depth + Clear. (only for cycles except C-SINK)
[CYCL_CLEAR+INCR]	
[CYCL_CLEAR]	Clearance. (only for cycles)
[CYCL_CSINK_DEPTH+CLEAR]	Drill Depth + Clear. (only for C-SINK cycle)
[CYCL_CSINK_DEPTH]	Drill Depth. (only for C-Sink cycle)
[CYCL_DEPTH]	Drill Depth. (only for cycles except C-SINK)
[CYCL_DWELL]	Dwell. (only for cycles)
[CYCL_INCR]	Step Increment. (only for cycles)
[CYCL_IPM]	IPM. (only for cycles except TAP)
[CYCL_IPR]	IPR. (only for TAP cycle)
[CYCL_NEG_CLEAR+DEPTH]	-(Drill Depth + Clear). (only for cycles except C-SINK)
[CYCLE_NUM_STEPS]	

[CYCL_ORIENT]	Orient. (only for cycles)
[CYCL_SCALED_DWELL]	Dwell * Scale Factor (only for cycles)
[CYCL_TPI]	
[CYCLE_Z]	This variable stores the Z value of drill points.
[CYCL_Z+CLEAR]	Next Z + Clearance. (only for cycles)
[CYCL_Z+DEPTH+CLEAR]	Next Z + Depth + Clearance. (only for cycles)
[CYCL_Z+DEPTH]	Next Z + Depth. (only for cycles)
[CYCL_Z-DEPTH]	Next Z - Depth. (only for cycles)
[DELIMITER]	Delimiter definition.
[DEPART_FEED]	Outputs the Departure feedrate value for the currently operation.
[ENGAGE_FEED]	Outputs the Engage feedrate value for the currently operation.
[EOB]	The end of block character.
[EXTRUSION]	
[EXTRUSION_BLK]	
[EXTRUSION_CODE]	
[FEEDRATE]	FeedRate Value.
[FEEDRATE_BLK]	FeedRate Block.
[FEEDRATE_CODE]	Outputs the Feedrate code
[FEEDRATE_UNITS_CODE]	Outputs the Feedrate Units code
[FIRST_TOOL_NUM]	
[G_CODE]	The next G-Code. This is translated to linear, rapid, arc or cycle G-Code.
[HELIX_ANGLE]	
[HELIX_CCW_ARC]	
[HELIX_CW_ARC]	
[HELIX_LEAD]	
[HELIX_NUM_REV]	

[HELIX_RADIUS]	
[HELIX_TOTAL_DEPTH]	
[INPUTFILE_NAME]	
[INPUTFILE_NAME_LONG]	
[INV_TIME_FEEDRATE_OFF]	
[INV_TIME_FEEDRATE_ON]	
[LINEAR]	The linear motion code.
[MOP_NAME]	
[NEXT_ABS_X_WCS]	The next absolute X coordinate point in World Coordinates.
[NEXT_ABS_Y_WCS]	The next absolute Y coordinate point in World Coordinates.
[NEXT_ABS_Z_WCS]	The next absolute Z coordinate point in World Coordinates.
[NEXT_I]	
[NEXT_J]	
[NEXT_K]	
[NEXT_NONMDL_E]	
[NEXT_NONMDL_I]	
[NEXT_NONMDL_J]	
[NEXT_NONMDL_K]	
[NEXT_NONMDL_L]	
[NEXT_NONMDL_R]	
[NEXT_NONMDL_X]	The next non-modal X coordinate point in local Machine Coordinates.
[NEXT_NONMDL_X_WCS]	The next non-modal X coordinate point in World Coordinates.
[NEXT_NONMDL_Y]	The next non-modal Y coordinate point in local Machine Coordinates.
[NEXT_NONMDL_Y_WCS]	The next non-modal Y coordinate point in World Coordinates.

[NEXT_NONMDL_Z]	The next non-modal Z coordinate point in local Machine Coordinates.
[NEXT_NONMDL_Z_WCS]	The next non-modal Z coordinate point in World Coordinates.
[NEXT_R]	
[NEXT_TOOL_NAME]	
[NEXT_TOOL_NUM]	
[NEXT_X]	The next X coordinate point in Machine Coordinates.
[NEXT_X_WCS]	The next X coordinate point in World Coordinates.
[NEXT_Y]	The next Y coordinate point in Machine Coordinates.
[NEXT_Y_WCS]	The next Y coordinate point in World Coordinates.
[NEXT_Z]	The next Z coordinate point in Machine Coordinates.
[NEXT_Z_WCS]	The next Z coordinate point in World Coordinates.
[OUTPUTE_MODE_CODE]	
[OUTPUT_UNITS_CODE]	English or Metric outputs code.
[OUTPUTFILE_NAME]	
[OUTPUT_FILENAME_LONG]	
[PARTNAME]	
[PARTNUM]	
[PLUNGE_FEED]	
[POST_NAME]	
[POST_NAME_LONG]	
[PREV_TOOL_ADJST_REG]	
[PREV_TOOL_CUTCOM_REG]	
[PREV_TOOL_NUM]	
[PREV_TOOL_NUM_FLUTES]	

[PREV_TOOL_ZOFFSET]	
[PROGRAM_END_BLK]	
[PROGRAM_START_BLK]	
[RAPID]	The rapid motion code.
[RAPID_FEED]	
[RETRACT_FEED]	
[ROTATION_AXIS]	
[ROTATION_AXIS_SECONDARY]	
[ROTATION_DIR]	
[ROTATION_DIR_SECONDARY]	
[ROTATION_FEEDVALUE]	
[ROTATION_MODE]	
[SEQ_PRECHAR]	Letter that is prefixed before the sequence number
[SEQNUM]	The actual sequence number.
[SETUP_NAME]	
[SETUP_X]	
[SETUP_XAXIS_X]	
[SETUP_XAXIS_Y]	
[SETUP_XAXIS_Z]	
[SETUP_Y]	
[SETUP_YAXIS_X]	
[SETUP_YAXIS_Y]	
[SETUP_YAXIS_Z]	
[SETUP_Z]	
[SETUP_ZAXIS_X]	
[SETUP_ZAXIS_Y]	

[SETUP_ZAXIS_Z]	
[SPINDLE_ARC]	Spindle direction.
[SPINDLE_BLK]	Spindle Block.
[SPINDLE_CODE]	
[SPINDLE_SPD]	Spindle speed.
[SPINDLE_SPD_MAX]	
[SPINDLE_SPD_TYPE]	
[SPIRAL_ANGLE]	
[SPIRAL_CCW_ARC]	
[SPIRAL_CW_ARC]	
[SPIRAL_END_RADIUS]	
[SPIRAL_LEAD]	
[SPIRAL_NUM_REV]	
[SPIRAL_START_RADIUS]	
[SPIRAL_TOTAL_LENGTH]	
[START_CHAR]	The program start character.
[START_POSITION_X]	
[START_POSITION_Y]	
[START_POSITION_Z]	
[START_X]	The X coordinate of start point.
[START_X_WCS]	The X coordinate of start point in World Coordinates.
[START_Y]	The Y coordinate of start point.
[START_Y_WCS]	The Y coordinate of start point in World Coordinates.
[START_Z]	The Z coordinate of start point.
[START_Z_WCS]	The Z coordinate of start point in World Coordinates.
[STEP_NEXT_X]	

[STEP_NEXT_Y]	
[STEP_NEXT_Z]	
[STEP_START_DEPTH]	
[STOCK_LENGTH_X]	UNDEFINED is output if there is no stock defined when posting occurs.
[STOCK_LENGTH_Y]	
[STOCK_LENGTH_Z]	
[STOCK_MAX_X]	
[STOCK_MAX_Y]	
[STOCK_MAX_Z]	
[STOCK_MIN_X]	
[STOCK_MIN_Y]	
[STOCK_MIN_Z]	
[STOP_CHAR]	The program end character.
[TEMPERATURE]	
[TEMPERATURE_BED_SET_CODE]	
[TEMPERATURE_BED_WAIT_CODE]	
[TEMPERATURE_EXTRUDER_SET_CODE]	
[TEMPERATURE_EXTRUDER_WAIT_CODE]	
[TEMPERATURE_SET_BLK]	
[TEMPERATURE_SET_CODE]	
[TEMPERATURE_WAIT_BLK]	
[TEMPERATURE_BED_WAIT_CODE]	
[THREAD_ANGLE]	
[THREAD_DEPTH]	
[THREAD_DIR]	
[THREAD_FINISH_NUMCUTS]	
[THREAD_FINISH_STOCK]	
[THREAD_FINISH_Z]	
[THREAD_FIRST_DEPTH]	
[THREAD_INFEED_TYPE]	

[THREAD_LENGTH]	
[THREAD_MAJOR_DIR]	
[THREAD_MIN_DEPTH]	
[THREAD_MINOR_DIA]	
[THREAD_PITCH]	
[THREAD_PULL_OUT_DIST]	The pull out value is specified under G76 parameters in threading mop in turning.
[THREAD_TAPER]	
[TIME_STAMP]	
[TOOL_ADJ_REG]	Tool Adjust Register number.
[TOOL_CHG_PT_X]	
[TOOL_CHG_PT_Y]	
[TOOL_CHG_PT_Z]	
[TOOL_CUTCOM_REG]	
[TOOL_DIA]	Tool Diameter.
[TOOL_LENGTH]	Tool length.
[TOOL_NAME]	
[TOOL_NUM]	Tool number.
[TOOL_NUM_FLUTES]	
[TOOL_RAD]	Tool Radius.
[TOOL_ZOFFSET]	
[TOOLPATH_MAX_X]	
[TOOLPATH_MAX_Y]	
[TOOLPATH_MAX_Z]	
[TOOLPATH_MIN_X]	
[TOOLPATH_MIN_Y]	
[TOOLPATH_MIN_Z]	

[VMPFILE_NAME]	
[WMPFILE_NAME_LONG]	
[WORK_OFFSET_CODE]	
[WORK_OFFSET_NUM]	
[WORK_OFFSET_PREFIX]	
[ZFEEDRATE]	
INV_TIME_FEEDRATE_FLAG	

Variables

The variables that can be used are listed in the table below.

The variables that start with “CYCL_” are used only for the cycles commands.

Variable	Comments
[ABS_ANGLE]	
[ABS_ANGLE_SECONDARY]	
[ANGLE]	
[ANGLE_PITCH]	
[ANGLE_ROLL]	
[ANGLE_SECONDARY]	
[ANGLE_YAW]	
[APPROACH_FEED]	
[CENTER_X]	
[CENTER_Y]	
[CENTER_Z]	
[CIR_PLANE]	
[COMMENT]	Output comments.
[COOLANT_CODE_INSERT]	
[COOLANT_OFF]	
[CURR_X]	The X coordinate of current point.

[CURR_X_WCS]	The X coordinate of current point in World Coordinates
[CURR_Y]	The Y coordinate of current point.
[CURR_Y_WCS]	The Y coordinate of current point in World Coordinates
[CURR_Z]	The Z coordinate of current point.
[CURR_Z_WCS]	The Z coordinate of current point in World Coordinates
[CUT_FEED]	
[CUT_MOTION_END_BLK]	
[CUT_MOTION_START_BLK]	
[CYCL_1/TPI]	1/TPI (only for TAP cycle)
[CYCL_CLEAR+DEPTH]	Drill Depth + Clear. (only for cycles except C-SINK)
[CYCL_CLEAR+INCR]	
[CYCL_CLEAR]	Clearance. (only for cycles)
[CYCL_CSINK_DEPTH+CLEAR]	Drill Depth + Clear. (only for C-SINK cycle)
[CYCL_CSINK_DEPTH]	Drill Depth. (only for C-Sink cycle)
[CYCL_DEPTH]	Drill Depth. (only for cycles except C-SINK)
[CYCL_DWELL]	Dwell. (only for cycles)
[CYCL_INCR]	Step Increment. (only for cycles)
[CYCL_IPM]	IPM. (only for cycles except TAP)
[CYCL_IPR]	IPR. (only for TAP cycle)
[CYCL_NEG_CLEAR+DEPTH]	-(Drill Depth + Clear). (only for cycles except C-SINK)
[CYCLE_NUM_STEPS]	
[CYCL_ORIENT]	Orient. (only for cycles)
[CYCL_SCALED_DWELL]	Dwell * Scale Factor (only for cycles)
[CYCL_TPI]	
[CYCLE_Z]	This variable stores the Z value of drill points.

[CYCL_Z+CLEAR]	Next Z + Clearance. (only for cycles)
[CYCL_Z+DEPTH+CLEAR]	Next Z + Depth + Clearance. (only for cycles)
[CYCL_Z+DEPTH]	Next Z + Depth. (only for cycles)
[CYCL_Z-DEPTH]	Next Z - Depth. (only for cycles)
[DELIMITER]	Delimiter definition.
[DEPART_FEED]	Outputs the Departure feedrate value for the currently operation.
[ENGAGE_FEED]	Outputs the Engage feedrate value for the currently operation.
[EOB]	The end of block character.
[EXTRUSION]	
[EXTRUSION_BLK]	
[EXTRUSION_CODE]	
[FEEDRATE]	FeedRate Value.
[FEEDRATE_BLK]	FeedRate Block.
[FEEDRATE_CODE]	Outputs the Feedrate code
[FEEDRATE_UNITS_CODE]	Outputs the Feedrate Units code
[FIRST_TOOL_NUM]	
[G_CODE]	The next G-Code. This is translated to linear, rapid, arc or cycle G-Code.
[HELIX_ANGLE]	
[HELIX_CCW_ARC]	
[HELIX_CW_ARC]	
[HELIX_LEAD]	
[HELIX_NUM_REV]	
[HELIX_RADIUS]	
[HELIX_TOTAL_DEPTH]	
[INPUTFILE_NAME]	
[INPUTFILE_NAME_LONG]	
[INV_TIME_FEEDRATE_OFF]	

[INV_TIME_FEEDRATE_ON]	
[LINEAR]	The linear motion code.
[MOP_NAME]	
[NEXT_ABS_X_WCS]	The next absolute X coordinate point in World Coordinates.
[NEXT_ABS_Y_WCS]	The next absolute Y coordinate point in World Coordinates.
[NEXT_ABS_Z_WCS]	The next absolute Z coordinate point in World Coordinates.
[NEXT_I]	
[NEXT_J]	
[NEXT_K]	
[NEXT_NONMDL_E]	
[NEXT_NONMDL_I]	
[NEXT_NONMDL_J]	
[NEXT_NONMDL_K]	
[NEXT_NONMDL_L]	
[NEXT_NONMDL_R]	
[NEXT_NONMDL_X]	The next non-modal X coordinate point in local Machine Coordinates.
[NEXT_NONMDL_X_WCS]	The next non-modal X coordinate point in World Coordinates.
[NEXT_NONMDL_Y]	The next non-modal Y coordinate point in local Machine Coordinates.
[NEXT_NONMDL_Y_WCS]	The next non-modal Y coordinate point in World Coordinates.
[NEXT_NONMDL_Z]	The next non-modal Z coordinate point in local Machine Coordinates.
[NEXT_NONMDL_Z_WCS]	The next non-modal Z coordinate point in World Coordinates.
[NEXT_R]	
[NEXT_TOOL_NAME]	
[NEXT_TOOL_NUM]	

[NEXT_X]	The next X coordinate point in Machine Coordinates.
[NEXT_X_WCS]	The next X coordinate point in World Coordinates.
[NEXT_Y]	The next Y coordinate point in Machine Coordinates.
[NEXT_Y_WCS]	The next Y coordinate point in World Coordinates.
[NEXT_Z]	The next Z coordinate point in Machine Coordinates.
[NEXT_Z_WCS]	The next Z coordinate point in World Coordinates.
[OUTPUTE_MODE_CODE]	
[OUTPUT_UNITS_CODE]	English or Metric outputs code.
[OUTPUTFILE_NAME]	
[OUTPUT_FILENAME_LONG]	
[PARTNAME]	
[PARTNUM]	
[PLUNGE_FEED]	
[POST_NAME]	
[POST_NAME_LONG]	
[PREV_TOOL_ADJST_REG]	
[PREV_TOOL_CUTCOM_REG]	
[PREV_TOOL_NUM]	
[PREV_TOOL_NUM_FLUTES]	
[PREV_TOOL_ZOFFSET]	
[RAPID]	The rapid motion code.
[RAPID_FEED]	
[RETRACT_FEED]	
[ROTATION_AXIS]	
[ROTATION_AXIS_SECONDARY]	

[ROTATION_DIR]	
[ROTATION_DIR_SECONDARY]	
[ROTATION_FEEDVALUE]	
[ROTATION_MODE]	
[SEQ_PRECHAR]	Letter that is prefixed before the sequence number
[RT_NXT_X]	The next X coordinate. (Modal)
[RT_NXT_Y]	The next Y coordinate. (Modal)
[RT_NXT_Z]	The next Z coordinate. (Modal)
[RT_NXT_NONMDL_X]	The next X coordinate. (NonModal)
[RT_NXT_NONMDL_Y]	The next Y coordinate. (NonModal)
[RT_NXT_NONMDL_Z]	The next Z coordinate. (NonModal)
[SEQNUM]	The actual sequence number.
[SPINDLE_ARC]	Spindle direction.
[SPINDLE_BLK]	Spindle Block.
[SPINDLE_CODE]	
[SPINDLE_SPD]	Spindle speed.
[SPINDLE_SPD_MAX]	
[SPINDLE_SPD_TYPE]	
[SPIRAL_ANGLE]	
[SPIRAL_CCW_ARC]	
[SPIRAL_CW_ARC]	
[SPIRAL_END_RADIUS]	
[SPIRAL_LEAD]	
[SPIRAL_NUM_REV]	
[SPIRAL_START_RADIUS]	
[SPIRAL_TOTAL_LENGTH]	
[START_CHAR]	The program start character.
[START_POSITION_X]	

[START_POSITION_Y]	
[START_POSITION_Z]	
[START_X]	The X coordinate of start point.
[START_X_WCS]	The X coordinate of start point in World Coordinates.
[START_Y]	The Y coordinate of start point.
[START_Y_WCS]	The Y coordinate of start point in World Coordinates.
[START_Z]	The Z coordinate of start point.
[START_Z_WCS]	The Z coordinate of start point in World Coordinates.
[STEP_NEXT_X]	
[STEP_NEXT_Y]	
[STEP_NEXT_Z]	
[STEP_START_DEPTH]	
[STOCK_LENGTH_X]	UNDEFINED is output if there is no stock defined when posting occurs.
[STOCK_LENGTH_Y]	
[STOCK_LENGTH_Z]	
[STOCK_MAX_X]	
[STOCK_MAX_Y]	
[STOCK_MAX_Z]	
[STOCK_MIN_X]	
[STOCK_MIN_Y]	
[STOCK_MIN_Z]	
[STOP_CHAR]	The program end character.
[TEMPERATURE]	
[TEMPERATURE_BED_SET_CODE]	
[TEMPERATURE_BED_WAIT_CODE]	
[TEMPERATURE_EXTRUDER_SET_CODE]	
[TEMPERATURE_EXTRUDER_WAIT_CODE]	
[TEMPERATURE_SET_BLK]	
[TEMPERATURE_SET_CODE]	
[TEMPERATURE_WAIT_BLK]	

[TEMPERATURE_BED_WAIT_CODE]	
[THREAD_ANGLE]	
[THREAD_DEPTH]	
[THREAD_DIR]	
[THREAD_FINISH_NUMCUTS]	
[THREAD_FINISH_STOCK]	
[THREAD_FINISH_Z]	
[THREAD_FIRST_DEPTH]	
[THREAD_INFEED_TYPE]	
[THREAD_LENGTH]	
[THREAD_MAJOR_DIR]	
[THREAD_MIN_DEPTH]	
[THREAD_MINOR_DIA]	
[THREAD_PITCH]	
[THREAD_PULL_OUT_DIST]	The pull out value is specified under G76 parameters in threading mop in turning.
[THREAD_TAPER]	
[TIME_STAMP]	
[TOOL_ADJ_REG]	Tool Adjust Register number.
[TOOL_CHG_PT_X]	
[TOOL_CHG_PT_Y]	
[TOOL_CHG_PT_Z]	
[TOOL_CUTCOM_REG]	
[TOOL_DIA]	Tool Diameter.
[TOOL_LENGTH]	Tool length.
[TOOL_NAME]	
[TOOL_NUM]	Tool number.

[TOOL_NUM_FLUTES]	
[TOOL_RAD]	Tool Radius.
[TOOL_ZOFFSET]	
[VMPFILE_NAME]	
[WMPFILE_NAME_LONG]	
[WORK_OFFSET_NUM]	
[WORK_OFFSET_PREFIX]	
[ZFEEDRATE]	
INV_TIME_FEEDRATE_FLAG	

Related Topics

Other PPG Editor Sections:

General

Start/End

Tool Change

Setup

Spindle

Feed Rate

Motion

Circle

Helical/Spiral Cycles

Multi Axis Motion

Cutter Compensation

Cut Motion Start/End

Cycles

Miscellaneous

5.3 Macros

5.3.1 Macros

The following is a list of the available macros in the RhinoCAM Post-Processor generator.

StartEnd Tab

These macros are output before and after a tool path and therefore Runtime variables cannot be used.

Program Start up Code	Program End Code

Tool Change Tab

These macros are output only when a tool is loaded or changed.

First Load Tool Macro	Tool Change Macro

Cycles Tab

These macros are output only when the motion command is a cycle command.

Drill Code	Deep Code	Break Chip Code
Counter Sink Code	Tap Code	Bore [Drag] Code
Bore [No Drag[Code	Bore [Manual] Code	Bore [Reverse] Code

5.3.2 How to edit Macros

This section describes the procedure for editing macros.

You can directly type the macro. With the exception of '[' and ']' characters as variables.

Usage of Variable List Dialog
You can also add variables directly in the following manner.

	Operation	Dialog to operate
1	Click the right mouse button on the required edit box	Main Editor
2	Variable List Dialog is displayed.	Variable List Dialog
3	Select the position.	Main Editor
4	Select the variable to add. (This procedure is for adding) Double click the left mouse button on the variable list and the highlighted variable is added.	Variable List Dialog
5	Use the [Add] or [Undo] or [Undo All] buttons to perform the standard add and undo operations.	Variable List Dialog

Note : Multiple operations would require you to repeat steps 3 to 5.

Use Programmable Post

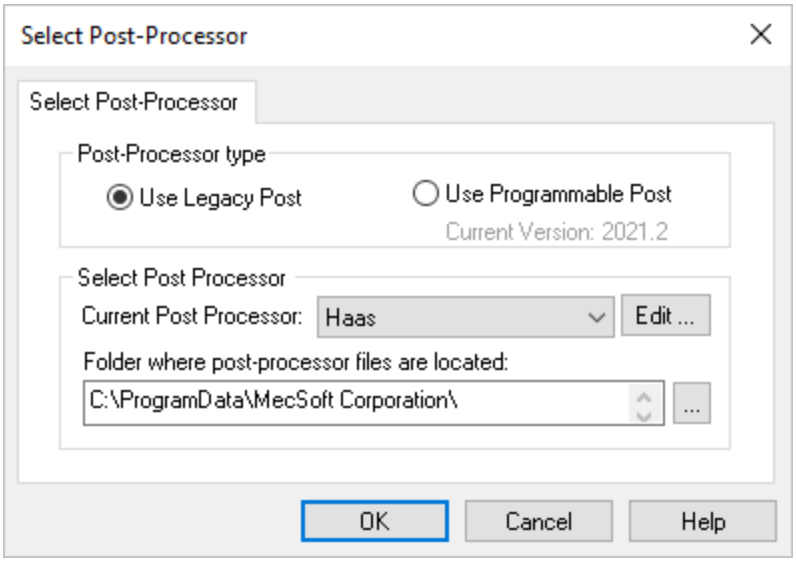
This documentation describes Programmable Post Processor and details on how create post scripts for it.

The Programmable Post Processor allows you to use Python script files for G-Code generation during toolpath processing by each MecSoft CAM plugin. You can create new post scripts or modify existing ones. Using the Python language for post scripts allows you to more efficiently generate the needed G-Code output allowing the use of programming elements, such as condition processing, loops, logical operations, and more.

6.1 Selecting a Programmable Post

Perform the following steps to use a programmable post-processor.

1. Select Use Programmable Post from Set Post-Processor Options dialog. When Use Programmable Post is selected the Python post scripts will be used for G-Code generation.



Dialog Box: Set Post-Processor Options

2. Select an available post script file or create a new one.
3. Press OK to save the options.

6.2 Writing Scrips

Script used in Programmable Post Processor uses Python programming language. That's why to execute it correctly, the script should have an appropriate code structure and format. As scripts written using Python programming language users can use its language features like loops, conditions, functions, etc. Also, all language restrictions applied to post scripts.

All post scripts should be placed in Post/[module_name]/PY folder, along with default.spm files. (The default.spm is only used for general post initialization).

The beginning of the file must contain the following line. It imports data types for event handling:

```
from post_ext import *
```

If you want to use the option Extension from Post-Processor then the variable GENERAL_OutputFileExt should be specified as follows:

```
GENERAL_OutputFileExt = ".nc"
```

6.3 Post Events

Programmable Post Processor has event driven architecture. Before outputting a specific block of G-code event is called. List of available events can be found at "Post event list" section.

User can specify output g-code block format on each event and Set/Get Global Post variables using objects from event parameters:

```
def OnEndProcessing(blockData: PostBlockData, globalData: PostGlobalData):
  return
```

PostBlockData object used to get or set G-Code block format for called event. Some block formats can contain multiple lines. For these cases each line should be separated by '\n' symbol (user can use .splitlines() and .join functions from python to process them).

PostBlockData	
Methods	**Description**
Set(string)	Set block format string
Get()	Get block format string

Example of modification block output format:

```
def OnRapidMotion(blockData: PostBlockData, globalData: PostGlobalData):
  data = blockData.Get()
  data = data + " - Modified by OnRapidMotion"
  blockData.Set(data)
  return
```

To access global(system) post variables user can use PostGlobalData object, list of variables can be found at "Post variable list" section

PostGlobalData Object Methods	
Methods	**Description**
GetIntVar(string)	Get variable value as Int
GetFloatVar(string)	Get variable value as float
GetStrVar(string)	Get variable value as string
SetIntVar(string, value)	Set int value for variable
SetFloatVar(string, value)	Set float value for variable
SetStrVar(string, value)	Set string value for variable

Example of modification of variable from variable list:

```
def OnRapidMotion(blockData: PostBlockData, globalData: PostGlobalData):
  nextX = float(globalData.GetStrVar("[NEXT_NONMDL_X]")) # e.g. converting to float by using Python
  nextX = nextX + 1.3
  globalData.SetFloatVar("[NEXT_NONMDL_X]", nextX)
  return
```

6.4 Post Event List

Post Event List	
Event	**Description**
OnStartProcessing(blockData: PostBlockData, globalData: PostGlobalData)	Starting post processing event
OnEndProcessing(blockData: PostBlockData, globalData: PostGlobalData)	End post processing event
OnMOpStart(globalData: PostGlobalData)	Starting processing Mop
OnMOpEnd(globalData: PostGlobalData)	Finished generating G-Code for Mop
OnComment(blockData: PostBlockData, globalData: PostGlobalData)	Processing comment output
OnSetup(blockData: PostBlockData, globalData: PostGlobalData)	Processing Mop Setup output

OnWorkZero(blockData: PostBlockData, globalData: PostGlobalData)	Processing Work Zero Mop output
OnToolChange(blockData: PostBlockData, globalData: PostGlobalData)	Changing tool output
OnSpindleSpeed(blockData: PostBlockData, globalData: PostGlobalData)	Processing Spindle Speed output
OnFeedRate(blockData: PostBlockData, globalData: PostGlobalData)	Processing Feed Rate output
OnToolCompensation(blockData: PostBlockData, globalData: PostGlobalData)	Processing Tool Compensation output
OnCoolant(blockData: PostBlockData, globalData: PostGlobalData)	Processing coolant output
OnRapidMotion(blockData: PostBlockData, globalData: PostGlobalData)	Processing Rapid Motion output
OnLinearMotion(blockData: PostBlockData, globalData: PostGlobalData)	Processing Linear Motion output
OnCirclularMotion(blockData: PostBlockData, globalData: PostGlobalData)	Processing Circular Motion output
OnSpiralMotion(blockData: PostBlockData, globalData: PostGlobalData)	Processing Spiral Motion output
OnHelicalMotion(blockData: PostBlockData, globalData: PostGlobalData)	Processing Helical Motion output
On4AxisRapidMotion(blockData: PostBlockData, globalData: PostGlobalData)	Processing Rapid Motion output for 4Axis machine
On4AxLinearMotion(blockData: PostBlockData, globalData: PostGlobalData)	Processing Linear Motion output for 4Axis machine
On5AxRapidMotion(blockData: PostBlockData, globalData: PostGlobalData)	Processing Rapid Motion output for 5Axis machine
On5AxLinearMotion(blockData: PostBlockData, globalData: PostGlobalData)	Processing Linear Motion output for 5Axis machine
OnCycleStart(blockData: PostBlockData, globalData: PostGlobalData)	Starting processing cycles
OnCyclePoint(blockData: PostBlockData, globalData: PostGlobalData)	Processing cycle point output
OnCycleEnd(globalData: PostGlobalData)	Ending processing cycles
OnDwell(blockData: PostBlockData, globalData: PostGlobalData)	Processing Dwell output

OnRotateHead(blockData: PostBlockData, globalData: PostGlobalData)	Processing Rotate head output
OnRotateTable(blockData: PostBlockData, globalData: PostGlobalData)	Processing Rotate table output
OnMachineRewind(blockData: PostBlockData, globalData: PostGlobalData)	

6.5 Post Variables List

Post Variables List	
Variable	**Description**
[ABS_ANGLE]	
[ABS_ANGLE_SECONDARY]	
[ANGLE]	
[ANGLE_SECONDARY]	
[APPROACH_FEED]	
[CENTER_X]	
[CENTER_Y]	
[CENTER_Z]	
[CIR_PLANE]	
[COMMENT]	Output comments.
[CURR_X]	The X coordinate of current point.
[CURR_X_WCS]	The X coordinate of current point in World Coordinates
[CURR_Y]	The Y coordinate of current point.
[CURR_Y_WCS]	The Y coordinate of current point in World Coordinates
[CURR_Z]	The Z coordinate of current point.
[CURR_Z_WCS]	The Z coordinate of current point in World Coordinates
[CUT_FEED]	
[CYCL_1/TPI]	1/TPI (only for TAP cycle)

[CYCL_CLEAR+DEPTH]	Drill Depth + Clear. (only for cycles except C-SINK)
[CYCL_CLEAR+INCR]	
[CYCL_CLEAR]	Clearance. (only for cycles)
[CYCL_CSINK_DEPTH+CLEAR]	Drill Depth + Clear. (only for C-SINK cycle)
[CYCL_CSINK_DEPTH]	Drill Depth. (only for C-Sink cycle)
[CYCL_DEPTH]	Drill Depth. (only for cycles except C-SINK)
[CYCL_DWELL]	Dwell. (only for cycles)
[CYCL_INCR]	Step Increment. (only for cycles)
[CYCL_IPM]	IPM. (only for cycles except TAP)
[CYCL_IPR]	IPR. (only for TAP cycle)
[CYCL_NEG_CLEAR+DEPTH]	-(Drill Depth + Clear). (only for cycles except C-SINK)
[CYCLE_NUM_STEPS]	
[CYCL_ORIENT]	Orient. (only for cycles)
[CYCL_SCALED_DWELL]	Dwell * Scale Factor (only for cycles)
[CYCL_TPI]	
[CYCL_Z+CLEAR]	Next Z + Clearance. (only for cycles)
[CYCL_Z+DEPTH+CLEAR]	Next Z + Depth + Clearance. (only for cycles)
[CYCL_Z+DEPTH]	Next Z + Depth. (only for cycles)
[CYCL_Z-DEPTH]	Next Z - Depth. (only for cycles)
[DELIMITER]	Delimiter definition.
[DEPART_FEED]	Outputs the Departure feedrate value for the currently operation.
[ENGAGE_FEED]	Outputs the Engage feedrate value for the currently operation.
[EOB]	The end of block character.
[EXTRUSION]	
[EXTRUSION_BLK]	
[EXTRUSION_CODE]	
[FEEDRATE]	FeedRate Value.
[FEEDRATE_BLK]	FeedRate Block.

[FEEDRATE_CODE]	Outputs the Feedrate code
[FEEDRATE_UNITS_CODE]	Outputs the Feedrate Units code
[FIRST_TOOL_NUM]	
[G_CODE]	The next G-Code. This is translated to linear, rapid, arc or cycle G-Code.
[HELIX_ANGLE]	
[HELIX_CCW_ARC]	
[HELIX_CW_ARC]	
[HELIX_LEAD]	
[HELIX_NUM_REV]	
[HELIX_RADIUS]	
[HELIX_TOTAL_DEPTH]	
[INPUTFILE_NAME]	
[INPUTFILE_NAME_LONG]	
[INV_TIME_FEEDRATE_OFF]	
[INV_TIME_FEEDRATE_ON]	
[LINEAR]	The linear motion code.
[MAXZ]	
[MINZ]	
[MOP_NAME]	
[NEXT_ABS_X_WCS]	The next absolute X coordinate point in World Coordinates.
[NEXT_ABS_Y_WCS]	The next absolute Y coordinate point in World Coordinates.
[NEXT_ABS_Z_WCS]	The next absolute Z coordinate point in World Coordinates.
[NEXT_I]	
[NEXT_J]	
[NEXT_K]	
[NEXT_NONMDL_E]	
[NEXT_NONMDL_I]	
[NEXT_NONMDL_J]	

[NEXT_NONMDL_K]	
[NEXT_NONMDL_L]	
[NEXT_NONMDL_R]	
[NEXT_NONMDL_X]	The next non-modal X coordinate point in local Machine Coordinates.
[NEXT_NONMDL_X_WCS]	The next non-modal X coordinate point in World Coordinates.
[NEXT_NONMDL_Y]	The next non-modal Y coordinate point in local Machine Coordinates.
[NEXT_NONMDL_Y_WCS]	The next non-modal Y coordinate point in World Coordinates.
[NEXT_NONMDL_Z]	The next non-modal Z coordinate point in local Machine Coordinates.
[NEXT_NONMDL_Z_WCS]	The next non-modal Z coordinate point in World Coordinates.
[NEXT_R]	
[NEXT_TOOL_NAME]	
[NEXT_TOOL_NUM]	
[NEXT_X]	The next X coordinate point in Machine Coordinates.
[NEXT_X_WCS]	The next X coordinate point in World Coordinates.
[NEXT_Y]	The next Y coordinate point in Machine Coordinates.
[NEXT_Y_WCS]	The next Y coordinate point in World Coordinates.
[NEXT_Z]	The next Z coordinate point in Machine Coordinates.
[NEXT_Z_WCS]	The next Z coordinate point in World Coordinates.
[OUTPUTE_MODE_CODE]	
[OUTPUT_UNITS_CODE]	English or Metric outputs code.
[OUTPUTFILE_NAME]	
[OUTPUT_FILENAME_LONG]	
[PARTNAME]	

[PARTNUM]	
[PLUNGE_FEED]	
[POST_NAME]	
[POST_NAME_LONG]	
[PREV_TOOL_ADJST_REG]	
[PREV_TOOL_CUTCOM_REG]	
[PREV_TOOL_NUM]	
[PREV_TOOL_NUM_FLUTES]	
[PREV_TOOL_ZOFFSET]	
[RAPID]	The rapid motion code.
[RAPID_FEED]	
[RETRACT_FEED]	
[ROTATION_AXIS]	
[ROTATION_AXIS_SECONDARY]	
[ROTATION_DIR]	
[ROTATION_DIR_SECONDARY]	
[ROTATION_FEEDVALUE]	
[ROTATION_MODE]	
[RT_NXT_X]	The next X coordinate. (Modal)
[RT_NXT_Y]	The next Y coordinate. (Modal)
[RT_NXT_Z]	The next Z coordinate. (Modal)
[RT_NXT_NONMDL_X]	The next X coordinate. (NonModal)
[RT_NXT_NONMDL_Y]	The next Y coordinate. (NonModal)
[RT_NXT_NONMDL_Z]	The next Z coordinate. (NonModal)
[SEQ_PRECHAR]	Letter that is prefixed before the sequence number
[SEQNUM]	The actual sequence number.
[SPINDLE_ARC]	Spindle direction.
[SPINDLE_BLK]	Spindle Block.
[SPINDLE_CODE]	
[SPINDLE_SPD]	Spindle speed.

[SPINDLE_SPD_MAX]	
[SPINDLE_SPD_TYPE]	
[SPIRAL_ANGLE]	
[SPIRAL_CCW_ARC]	
[SPIRAL_CW_ARC]	
[SPIRAL_END_RADIUS]	
[SPIRAL_LEAD]	
[SPIRAL_NUM_REV]	
[SPIRAL_START_RADIUS]	
[SPIRAL_TOTAL_LENGTH]	
[START_CHAR]	The program start character.
[START_POSITION_X]	
[START_POSITION_Y]	
[START_POSITION_Z]	
[START_X]	The X coordinate of start point.
[START_X_WCS]	The X coordinate of start point in World Coordinates.
[START_Y]	The Y coordinate of start point.
[START_Y_WCS]	The Y coordinate of start point in World Coordinates.
[START_Z]	The Z coordinate of start point.
[START_Z_WCS]	The Z coordinate of start point in World Coordinates.
[STEP_NEXT_X]	
[STEP_NEXT_Y]	
[STEP_NEXT_Z]	
[STEP_START_DEPTH]	
[STOCK_LENGTH_X]	UNDEFINED is output if there is no stock defined when posting occurs.
[STOCK_LENGTH_Y]	
[STOCK_LENGTH_Z]	
[STOCK_MAX_X]	

[STOCK_MAX_Y]	
[STOCK_MAX_Z]	
[STOCK_MIN_X]	
[STOCK_MIN_Y]	
[STOCK_MIN_Z]	
[STOP_CHAR]	The program end character.
[TEMPERATURE]	
[TEMPERATURE_BED_SET_CODE]	
[TEMPERATURE_BED_WAIT_CODE]	
[TEMPERATURE_EXTRUDER_SET_CODE]	
[TEMPERATURE_EXTRUDER_WAIT_CODE]	
[TEMPERATURE_SET_BLK]	
[TEMPERATURE_SET_CODE]	
[TEMPERATURE_WAIT_BLK]	
[TEMPERATURE_BED_WAIT_CODE]	
[THREAD_ANGLE]	
[THREAD_DEPTH]	
[THREAD_DIR]	
[THREAD_FINISH_NUMCUTS]	
[THREAD_FINISH_STOCK]	
[THREAD_FINISH_Z]	
[THREAD_FIRST_DEPTH]	
[THREAD_INFEED_TYPE]	
[THREAD_LENGTH]	
[THREAD_MAJOR_DIR]	
[THREAD_MIN_DEPTH]	
[THREAD_MINOR_DIA]	
[THREAD_PITCH]	
[THREAD_PULL_OUT_DIST]	The pull out value is specified under G76 parameters in threading mop in turning.
[THREAD_TAPER]	

[TIME_STAMP]	
[TOOL_ADJ_REG]	Tool Adjust Register number.
[TOOL_CHG_PT_X]	
[TOOL_CHG_PT_Y]	
[TOOL_CHG_PT_Z]	
[TOOL_CUTCOM_REG]	
[TOOL_DIA]	Tool Diameter.
[TOOL_LENGTH]	Tool length.
[TOOL_NAME]	
[TOOL_NUM]	Tool number.
[TOOL_NUM_FLUTES]	
[TOOL_RAD]	Tool Radius.
[TOOL_ZOFFSET]	
[VMPFILE_NAME]	
[WMPFILE_NAME_LONG]	
[WORK_OFFSET_NUM]	
[WORK_OFFSET_PREFIX]	
[ZFEEDRATE]	
INV_TIME_FEEDRATE_FLAG	

6.6 Example Script

Here is an example Python post script, (PostScriptExample.py), for a programmable post-processor which sets following data:

- Linear/Rapid motion format
- Start/End block format
- Setting "[SEQ_PRECHAR]" variable
- Tool Change block format (different for fist tool change)
- File extension for "File Extension from Post Processor" option

```
from post_ext import *
```

```
# Set output file extension
GENERAL_OutputFileExt = ".nc"


# Define block format for outputten code
LinearMotionCodeBlock = "[G_CODE][DELIMITER][NEXT_X][DELIMITER][NEXT_Y][DELIMITER]
[NEXT_Z]"
RapidMotionCodeBlock = "[G_CODE][DELIMITER][NEXT_Z]\n[NEXT_X][DELIMITER][NEXT_Y]"
StartProcessingBlock = "[START_CHAR]\n[SEQ_PRECHAR][SEQNUM][DELIMITER]G40[DELIMITER]
G49[DELIMITER]G80[DELIMITER]G98 - Start processing"
EndProcessingBlock = "[SEQ_PRECHAR][SEQNUM][DELIMITER]M30\n[STOP_CHAR] - End
processing"


FirstToolChangeCodeBlock = ("; First Tool Change\n"
"[SEQ_PRECHAR][SEQNUM][DELIMITER][OUTPUT_UNITS_CODE][DELIMITER]T[TOOL_NUM]
[DELIMITER]M06",
"[SEQ_PRECHAR][SEQNUM][SPINDLE_BLK]",
"[SEQ_PRECHAR][SEQNUM][DELIMITER][OUTPUT_MODE_CODE][DELIMITER][G_CODE][DELIMITER]
X[NEXT_NONMDL_X][DELIMITER]Y[NEXT_NONMDL_Y]",
"[SEQ_PRECHAR][SEQNUM][DELIMITER]G43[DELIMITER]Z[NEXT_NONMDL_Z][DELIMITER]
H[TOOL_ADJST_REG]")


ToolChangeCodeBlock = ("[SEQ_PRECHAR][SEQNUM][DELIMITER][OUTPUT_UNITS_CODE]
[DELIMITER]T[TOOL_NUM][DELIMITER]M06",
"[SEQ_PRECHAR][SEQNUM][SPINDLE_BLK]",
"[SEQ_PRECHAR][SEQNUM][DELIMITER][OUTPUT_MODE_CODE][DELIMITER][G_CODE][DELIMITER]
X[NEXT_NONMDL_X][DELIMITER]Y[NEXT_NONMDL_Y]",
"[SEQ_PRECHAR][SEQNUM][DELIMITER]G43[DELIMITER]Z[NEXT_NONMDL_Z][DELIMITER]
H[TOOL_ADJST_REG]")



# Global vars for processing
ToolChangeNum = 0


# Helper functions
def SetBlockData(blockData: PostBlockData, value):
```

```
  block = '\n'
  if type(value) == tuple:
    block = block.join(value)
  else:
    value = value.splitlines()
    block = block.join(value)
  blockData.Set(block)

# Set post vars
def InitializeVars(globalData: PostGlobalData):
  globalData.SetStrVar("[SEQ_PRECHAR]", "#")

def OnStartProcessing(blockData: PostBlockData, globalData: PostGlobalData):
  InitializeVars(globalData)
  SetBlockData(blockData, StartProcessingBlock)
  return

def OnEndProcessing(blockData: PostBlockData, globalData: PostGlobalData):
  SetBlockData(blockData, EndProcessingBlock)
  return

def OnMOpStart(globalData: PostGlobalData):
  return

def OnMOpEnd(globalData: PostGlobalData):
  return

def OnComment(blockData: PostBlockData, globalData: PostGlobalData):
  return

def OnSetup(blockData: PostBlockData, globalData: PostGlobalData):
  return
```

```
def OnWorkZero(blockData: PostBlockData, globalData: PostGlobalData):
  return

def OnToolChange(blockData: PostBlockData, globalData: PostGlobalData):
  global ToolChangeNum

  if ToolChangeNum == 0:
   SetBlockData(blockData, FirstToolChangeCodeBlock) # 1st tool change
  else:
   SetBlockData(blockData, ToolChangeCodeBlock)
  ToolChangeNum = ToolChangeNum + 1
  return

def OnSpindleSpeed(blockData: PostBlockData, globalData: PostGlobalData):
  return

def OnFeedRate(blockData: PostBlockData, globalData: PostGlobalData):
  return

def OnToolCompensation(blockData: PostBlockData, globalData: PostGlobalData):
  return

def OnCoolant(blockData: PostBlockData, globalData: PostGlobalData):
  return

def OnRapidMotion(blockData: PostBlockData, globalData: PostGlobalData):
  SetBlockData(blockData, RapidMotionCodeBlock)
  return

def OnLinearMotion(blockData: PostBlockData, globalData: PostGlobalData):
  SetBlockData(blockData, LinearMotionCodeBlock)
```

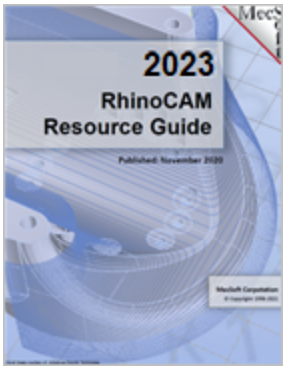
```
  return

def OnCirclularMotion(blockData: PostBlockData, globalData: PostGlobalData):
  return

def OnSpiralMotion(blockData: PostBlockData, globalData: PostGlobalData):
  return

def OnHelicalMotion(blockData: PostBlockData, globalData: PostGlobalData):
  return

def On4AxisRapidMotion(blockData: PostBlockData, globalData: PostGlobalData):
  SetBlockData(blockData, RapidMotionCodeBlock)
  return

def On4AxLinearMotion(blockData: PostBlockData, globalData: PostGlobalData):
  SetBlockData(blockData, LinearMotionCodeBlock)
  return

def On5AxRapidMotion(blockData: PostBlockData, globalData: PostGlobalData):
  return

def On5AxLinearMotion(blockData: PostBlockData, globalData: PostGlobalData):
  return

def OnCycleStart(blockData: PostBlockData, globalData: PostGlobalData):
  return

def OnCyclePoint(blockData: PostBlockData, globalData: PostGlobalData):
  return

def OnCycleEnd(globalData: PostGlobalData):
```

```
  return

def OnDwell(blockData: PostBlockData, globalData: PostGlobalData):
  return

def OnRotateHead(blockData: PostBlockData, globalData: PostGlobalData):
  return

def OnRotateTable(blockData: PostBlockData, globalData: PostGlobalData):
  return

def OnMachineRewind(blockData: PostBlockData, globalData: PostGlobalData):
  return
```

Find More Resources

Download this PDF Guide for a list of the available RhinoCAM Resources.

2022 RhinoCAM Resource Guide



The 2023 RhinoCAM Resource Guide!

18 Pages

Lists PDF downloads and Online resources including Quick Start Guides, Reference Guides, Exercise Guides, Tutorials and More.

Prefer Printed Documentation? Check Here!

Index

- A -

- B -

- C -

- D -

- F -

- H -

- I -

- L -

- M -

- P -

- S -

- T -

- U -

- V -