ArtCAM

Post Processor Configuration Guide

By Delcam

Important Notice

Contents

Editing A Configuration File

Post Processor Statements

Example Configuration Files

Introduction

What is a post processor ?

A post processor is a program which converts tool moves in some standard neutral format into the format required by a specific machine tool controller.

Examples of output for two different machine tool controls are shown below. Both files were generated from the same toolpath within ArtCAM.

Heidenhain Output

```
0 BEGIN PGM 1248 MM
1 TOOL DEF 1 L+0,000 R+0,000
2 TOOL CALL 1 Z S3000,000
3 L Z+10,000 F9999 M03
4 L X+0,000 Y+0,000 Z+10,000 F9999 M03
5 L X+0,001 Y+9,999 Z+10,000 F M
6 L Z+0,000 F360 M
7 L X+0,101 F720 M
8 L X+3,000 F M
.
.
.
```

Fanuc Output

```
%
:1248
N10G91G28X0Y0Z0
N20G40G17G80G49
N30T1M6
N40G90G54
N50G43Z10.000H1
N60G0X0.000Y0.000S3000M3
N70X0.001Y9.999
N80G1X0.001Y9.999Z0.000F360.0
N90G1X0.101Y9.999Z0.000F720.0
N100X3.000
.
.
```

These post processed files are sent to the machine tool to control the path of the cutting tool.

Why a configurable post processor ?

Because of the wide range of controls on the market, it is not possible for Delcam to write a specific post processor for every control. However the vast majority of controls have fairly simple formatting requirements, and therefore Delcam supply a configurable post processor which can be used by customers or agents to create a post processor to support their control. This configurable post processor uses an ASCII configuration file to specify the output format required.

This approach has a number of advantages.

❑ New machines tools can be supported quickly by writing a new configuration file or modifying an existing one.

❑ Existing configuration files can be modified if they do not conform to a users existing working practice (e.g. using inches for output instead of mm).

❑ Users can create their own post processors without having to wait for Delcam or their agents to add support for a particular machine tool control.

Delcam supply a number of configuration files for the most popular machine controls. These configuration files provide a good basis for creating a new post processor and experimenting with the effect of modifying various fields (please make a safe copy of any configuration file before modifying it).

An example of a configuration file is shown on the following page.

```
; Heidenhain configuration file - Metric version
;
; History
;
; Who When      What
; === ======== =========================
; BEM 01/04/94 Written
; bem 21/03/00 Added DESCRIPTION and
;              FILE_EXTENSION fields
;
DESCRIPTION = "Heidenhain (*.tap)"
;
FILE_EXTENSION = "tap"
;
UNITS = MM
; Carriage return - line feed at end of each line
END_OF_LINE = "[13][10]"
; Block numbering
LINE_NUM_START     = 0
LINE_NUM_INCREMENT = 1
LINE_NUM_MAXIMUM   = 999999
; Set up default formatting for variables
; Spindle Speed
FORMAT = [S|@|S|1,3]
; Feed Rate
FORMAT = [F|#||1.0]
; Tool moves in x,y and z
FORMAT = [X|#| X|+1,3]
FORMAT = [Y|#| Y|+1,3]
FORMAT = [Z|#| Z|+1,3]
; Home tool positions
FORMAT = [XH|@|X|+1,3]
FORMAT = [YH|@|Y|+1,3]
FORMAT = [ZH|@|Z|+1,3]
; Set up program header
START = "[N] BEGIN PGM 1248 MM"
START = "[N] TOOL DEF 1 L+0,000 R+0,000"
START = "[N] TOOL CALL 1 Z [S]"
START = "[N] L [ZH] F9999 M03"
START = "[N] L [XH] [YH] [ZH] F9999 M03"
; Program moves
RAPID_FEED_RATE = 9999
```

```
FEED_RATE_MOVE  = "[N] L[X][Y][Z] F[F] M"
; End of file
END = "[N] STOP M02"
END = "[N] END PGM 1248 MM"
```

The preceding configuration file may appear daunting!
However this is quite a complicated example and we will
go through each section of the file independently later in
this manual.

Configuring A New Post Processor

Modifying an existing post processor

The easiest way to create a new post processor is by modifying an existing one which has a similar format.

There are two main types of control which are supported by the configurable post processor. These are...

1. Controls which accept positions as floating point numbers. e.g. X 12.345 Y 7.120 Z 0.563.

2. Controls which use integer positions in multiples of some predefined step size. e.g. X 12345 Y 7120 Z563 for steps of 0.001 mm.

For the first type of control (floating point values) the G Code configuration file (**gcode_inch.con** or **gcode_mm.con**) or the Heidenhain configuration file (**heiden.con**) may be the best basis for a new control. If the control uses a variant of standard ISO G codes and M codes use **gcode_inch.con** or **gcode_mm.con** depending on whether you require the output to be in mm or inches. For the second type of control which use a format more

like HPGL, the Roland configuration file (**roland.con**) will probably be a better starting point.

Creating a new post processor from scratch

It is perfectly possible to create a configuration file from scratch. The files are simple ASCII files which can be created using a text editor such as NOTEPAD supplied with Windows.

However it is strongly recommended that new configuration files are created using existing files as a base to work from. This recommendation is given because experience has shown this to be a faster and less error prone approach to creating a new configuration file (see the preceding section for details about modifying an existing file).

NOTE: If a word processor such as Write or Word is used to edit the configuration files, the files **must** be saved back to disk as text files i.e. they must **not** contain the formatting information normally kept in a word processing document.

Configuration File Structure

Overview

The entries in the configuration file fall into five main categories which are described briefly below.

Global Statements

These determine the descriptive name for the post and its file extensions, whether the output is in mm or inches, line numbering information, line terminators and default formatting for numeric fields.

Start Of File

This section deals with lines which must be output at the start of every program. This information usually has the program name, commands for switching on the spindle, moves from the home position etc.

Tool Moves

This section details how the actual tool moves are output for the machine tool. The tool moves are sub-divided into two further categories.

Positioning Moves

These moves are used for positioning the tool while it is clear of the workpiece and are usually made at the maximum speed of the machine tool.

Cutting Moves

These are moves made with the tool cutting material and are made at the speeds specified within ArtCAM when the toolpath was created.

For both types of moves a format can optionally be specified for the first move in the category and a different format for all subsequent moves. This is often used when the first move switches on a modal flag in the control (e.g. G0 to specify positioning moves) and then subsequent moves do not need this command to be re-output. See the **fanuc.con** configuration file for an example of this behaviour.

Toolchanges

If the machine tool is equiped with an automatic toolchanger, or the user wants to allow manual toolchanges part way through an output file, the TOOLCHANGE statement can be used to output the appropriate commands.

NOTE: Toolchangers are NOT supported by MillWizard.

End Of File

This section deals with lines which must be output at the end of every program. This section usually contains commands for switching off the spindle, moving the tool back to the home position etc.

Post processor variables

Values for the tool position, feed rate, line number etc. are represented within the configuration file as **variables**. These variables are used as place holders for the value to be output, and control the formatting of each value. e.g.

The variable [X] represents the current X position for the tool at the current point on the toolpath, and similarly [Y] and [Z] represent the Y and Z co-ordinates.

The following statement in the configuration file

```
FEED_RATE_MOVE  = "X[X] Y[Y] Z[Z]"
```

Would mean that by default for a tool position of

X = 12.3 Y = 4.56 Z = 7.89

The following would be output

X 12.3 Y 4.56 Z 7.89

If the statement was specified as

```
FEED_RATE_MOVE  = "[Z] [Y] [X]"
```

The following would be output

7.89 4.56 12.3

The above example show the output with the default variable formats. The configurable post processor allows a considerable amount of control over the format in which values are output.

Variables

The following variables are available for use within the configuration file.

Name	Value
[F]	Current feed rate in default units/min
[FR]	Rapid feed rate in default units/mm
[FP]	Plunge feed rate in default units/mm
[FC]	Standard feedrate in default units/mm
[N]	Current line number
[XH]	Tool home position X co-ord
[YH]	Tool home position Y co-ord
[ZH]	Tool home position Z co-ord
[RH]	Tool home position rotary angle
[N]	Current line number
[S]	Spindle speed in r.p.m
[SZ]	Safe Z position from toolpath
[T]	Current tool number
[TP]	Previous tool number
[X]	X co-ord of tool position in default units
[Y]	Y co-ord of tool position in default units
[Z]	Z co-ord of tool position in default units
[R]	Angle in degrees for rotary axis
[XMIN]	Minimum x value of relief
[YMIN]	Minimum y value of relief
[ZMIN]	Minimum z value of relief
[XMAX]	Maximum x value of relief
[YMAX]	Maximum y value of relief
[ZMAX]	Maximum z value of relief
[TOOLDESC]	Tool description string
[PATHNAME]	Pathname of postprocessed file
[FILENAME]	Filename of postprocessed file

Variable formatting

Variables can have a default specification controlling number of decimal places, prefix, scaling factor etc. The default can be set once in the configuration file and all uses of the variable will pick up the default formatting. It is also possible to override the default formatting at any time by more fully specifying the variables format when it is actually used.

Variable Specification

A variable specification has the following format

[V|O|S|F|m]

Where

V = Variable name e.g. X or ZH.

S = String to print before value when output.

O = @ if value always output.
# if only output if value has changed since last time value was output.

F = Format string. This controls the number of decimal places, leading zeros etc. and is documented later.

m = Scaling factor - a floating point number by which variable is multiplied before formatting is applied.

All fields are optional apart from the variable name. Defaults for unspecified fields are as follows

X,Y,Z,XH,YH,ZH   [V|@||1.3|1.0]
S,F,N   [V|@||1.0|1.0]

The table below shows the output for a number of values with different variable formatting options.

Variable	X=12.345	X=12.345	X=34.567
[X|@|X|1,3]	X12,345	X12,345	X34,567
[X|#|X|1,3]	X12,345		X34,567
[X|@|X|07.2]	X0012.35	X0012.35	X034.57
[X|@||1.0|10.0]	123	123	346

The second example in the table - [X|#|X|1,3] . Shows the effect of the # character for controlling output. The second X value is not output because it was the same as the previous value.

Format string

The format string governing the output format of variables has the following form:

[flags] [width] [.precision]

Each field of the format specification is a single character or a number signifying a particular format option.

flags

Optional character or characters that control justification of output and printing of signs, blanks, and decimal points. (See Flag Directives below.) More than one flag can appear in a format specification.

width

Number that specifies minimum number of characters output. (See Width Specification below.)

precision

Number that specifies maximum number of characters printed for all or part of the output field, (See Precision Specification below.)

Flag Directives

This field is responsible for the overall formatting of the numeric value for the variable. Justification, printing of leading zeros, prefixing of + and - signs for values etc. are all controlled by this field. The following table shows the possible values which can appear in this field.

NOTE: More than one flag directive can be specified so +0 would be a valid directive to specify that leading zeros and a leading + or - should be output.

Flag	Meaning	Default
-	Left justify the result within the given field width.	Right justify.
+	Prefix the output value with a sign (+ or -) .	Sign appears only for negative values (-).
0	If width is prefixed with 0, zeros are added until the minimum width is reached. If 0 and - appear, the 0 is ignored.	No padding.
blank (' ')	Prefix the output value with a blank if the output value is signed and positive; the blank is ignored if both the blank and + flags appear.	No blank appears.
#	The # flag forces the output value to contain a decimal point in all cases.	Decimal point appears only if digits follow it.

Table showing values for flag directives in variable format.

Width Specification

The second field of the format specification is the width specification. The width argument is a non negative decimal integer controlling the minimum number of characters printed. If the number of characters in the output value is less than the specified width, blanks are added to the left or the right of the values depending on whether the - flag (for left justification) is specified until the minimum width is reached. If width is prefixed with 0, zeros are added until the minimum width is reached (not useful for left-justified numbers).

The width specification never causes a value to be truncated. If the number of characters in the output value is greater than the specified width, all characters of the value are printed (subject to the precision specification).

Precision Specification

The third field of the format specification is the precision specification. It specifies a non negative decimal integer, preceded by a period (.) or comma (,), which specifies the number of decimal places. If a comma is used values output using this format string will use commas instead of decimal points.

The precision value specifies the number of digits after the decimal point. If a decimal point appears, at least one digit appears before it. The value is rounded to the appropriate number of digits.

Default precision is 6; if precision is 0, or if the period (.) appears without a number following it, no decimal point is printed.

Unlike the width specification, the precision specification can cause rounding of a value.

Inserting control codes

In addition to the previously described variables, any ASCII character value can be inserted in a line by enclosing its decimal value in [] brackets e.g.

[13][10] would specify a carriage return, line feed sequence

Global Statements

These are statements which have an effect throughout the configuration file. The statements which fall into this category are...

Description

This specifies the descriptive name which is displayed in the drop down list used to select the post processor within the application. All the standard description strings incorporate the default extension in brackets at the end. In addition if there are both mm and inch versions of the same post processor the type should be indicated in the description.

DESCRIPTION = "G-Code (inch) (*.tap)"

File Extension

This specifies the file extension used for files created with this post processor.

FILE_EXTENSION = "tap"

Units

This statement specifies the units which the post processor is working with. All tool positions and feed rate variables will be in the specified units irrespective of the units used to create the relief and toolpath within ArtCAM. This field is optional and will default to mm.

UNITS =   **optional**

possible values:

MM   - all positions in mm, feed rates in mm/min

INCH - all positions in inches, feed rates in inches/min

default value: MM

Example

UNITS = INCH

Line Numbering

A number of statements are available to control the sequencing of line numbers within the output program. All these statements are optional as many controls do not require line numbers at all. Line numbers are actually output using the [N] variable and the actual formatting for the line number value output is specified by setting the format for the [N] variable.

LINE_NUM_START =   **optional**

Specifies the starting value for line numbers (accessed through N variable).

default

LINE_NUM_START = 10

LINE_NUM_INCREMENT =   optional

Specifies the increment used for the line numbers

default

      LINE_NUM_INCREMENT = 2

LINE_NUM_MAXIMUM =   optional

Specifies the maximum value allowed for line numbers. When the line number value exceeds this value, line numbering will start again at NUM_START.

default

      LINE_NUM_MAXIMUM = 999999

Rapid feed rate

This statement allows the feed rate for positioning moves to be specified for machines which do not use special codes (like G0) for positioning moves. The Heidenhain control identifies rapid moves as moves at a feed rate of 9999. For controls like FANUC which identify moves at rapid by codes like G0, this statement is not required.

RAPID_FEED_RATE =   **optional**

End of line

Most controls will expect a standard sequence of characters at the end of each line sent. rather than including these characters in every line specification in the configuration file, they can be specified using this statement and the post processor will append them to every line output. This statement is optional, but if it is not specified the user must add the required end of line characters to each line definition in the file which requires them.

END_OF_LINE =   **optional**

Example

        END_OF_LINE = "[13][10]"

This will add a carriage return, line feed combination to the end of each output line. This is the standard line termination for DOS text files.

Default variable format

As many lines in the configuration file may use variables like [N], [F], [X], [Y], [Z] etc. It is possible to specify a default format for these variables at the start of the configuration file. This means that full formatting information does not need to be entered every time the variable is used.

It is still possible to override the default formatting whenever a variable is used if the user wishes.

FORMAT = [V|O|S|F|m]   optional

Examples

FORMAT = [F|#||1.0]

Anywhere [F] is used in the file the feed rate value will only be output if it has changed (# field) and will not have any decimal point or decimal places displayed (1.0 field).e.g. for feed = 123.456

```
123
```

FORMAT = [X|@|X|+1,3]

Anywhere [X] is used in the file it will always be output (@ field), will be preceded by "X" and + or - to indicate its sign (+ field). The value itself will have 3 decimal places and instead of a decimal point a comma will be used. e.g. for X = 12.123

```
X12,123
```

FORMAT = [X|#| X |1.2]

Anywhere [X] is used in the file it will only appear if it has a different value to the last X value output, and will be output preceded by " X " with two decimal places. e.g. for X = 12.123

```
X 12.12
```

Start of file

These statements are used to specify the lines sent at the start of the file. As many lines as required can be specified using any of the defined variables **except** X, Y and Z (these will have an undefined value at the start of the file as the first tool move has not yet been made). The variables XH, YH and ZH can be used to initiate movement from the home tool position. This statement is optional.

START =   **optional**

Example for FANUC control

```
;  All programs start with % followed by program
;  number on next line
START = "%"
START = ":1248"
; set current tool position as home position
START = "N[N]G91G28X0Y0Z0"
; cutter comp cancel, specify XY plane, canned cycle cancel,
; tool length offset cancel
START = "N[N]G40G17G80G49"
; select first tool and do tool select
START = "N[N]T1M6"
; set absolute positioning
START = "N[N]G90G54"
; move in Z to first (home)  point - set tool length compensation
START = "N[N]G43[HZ|@|Z|1.3]"
; move in XY to first point at rapid (GO) and set spindle speed
START = "N[N]G0[XH|@|X|1.3][YH|@|Y|1.3][S|@|S|1.0]"
```

This may generate an output like the following, assuming LINE_NUM_START = 10 and

LINE_NUM_INCREMENT = 2 and the tool home position was set at 0,0,10 when the toolpath was generated within ArtCAM.

```
%
:1248
N10G40G17G80G49
N12T1M6
N14G90G54
N16G43Z10.000
N18G0X0.000Y0.000S12000
```

Tool moves

The statements described in this section are used to output the actual tool moves to the program file. Only one of the statements (FEED_RATE_MOVE) is compulsory, the others just allow more control over the output if required.

To understand why there are five statements available in this section a brief description of the ArtCAM toolpath will be given.

ArtCAM tool move types

ArtCAM outputs two major types of tool moves which are described below.

Rapid Moves - these are positioning moves made at the Safe Z height which is set when the toolpath is generated. These moves do not cut material and are usually made at the fastest speed at which the machine tool can move.

Feed Rate Moves - these are moves which actually cut the material. There are two types of moves within this category.

Plunge Moves - these are vertical movements down into the material at the start of each machining pass.

Normal Moves - these are the normal cutting moves which traverse the relief surface.

The only real difference between normal and plunge moves are the feed rates specified within ArtCAM.

Because some machine controls treat the first tool move of a particular type differently to subsequent moves of the same type, the post processor allows the two different moves to be output with separate formatting. This leads to four possible tool move types

1. First move at feed rate
2. Subsequent moves at feed rate
3. First move at rapid
4. Subsequent moves at rapid

A statement (called FEED_RATE_MOVE) must be provided for formatting the second type of move. Statements for the other three are optional and will default to the best 'match' for a move which is defined. This is illustrated by the diagram shown below.

In the diagram a statement within a dashed box is optional, the FEED_RATE_MOVE is compulsory and hence is always present. The diagram indicates that for a first rapid move, the post processor first tries to format the output using the FIRST_RAPID_MOVE statement. If this statement does not exist the RAPID_RATE_MOVE statement is used. If this statement does not exist the FEED_RATE_MOVE statement is used. The other types of moves follow a similar route as shown on the diagram.

Feed rate tool moves

The output of feed rate moves are controlled by the FEED_RATE_MOVE statement and the FIRST_FEED_RATE_MOVE statement.

The FEED_RATE_MOVE statement is the only statement which is compulsory. If it is not specified ArtCAM will report an error when an attempt is made to use the post processor.

For controls (such as Fanuc) which require a modal command to be output at the start of linear cutting moves the FIRST_FEED_RATE_MOVE should be defined.

Example output is demonstrated for two moves at feedrate 789.12 after a rapid move

      X123.45   Y234.56   Z5.67

      X123.45   Y345.89   Z4.21

Example for Fanuc control

```
FORMAT = [X|#|X|1.3]
FORMAT = [Y|#|Y|1.3]
FORMAT = [Z|#|Z|1.3]
FORMAT = [F|@|F|1.1]
FORMAT = [N|@|N|1.0]
.
.

FIRST_FEED_RATE_MOVE = "[N]G1[X|@][Y|@][Z|@][F]"
FEED_RATE_MOVE = "[N][X][Y][Z]"
```

Output

```
N1226G1X123.450Y234.560Z5.670F789.1
N1228Y345.890Z4.210
```

Rapid rate tool moves

The output of rapid feed rate moves are controlled by the RAPID_RATE_MOVE statement and the FIRST_RAPID_RATE_MOVE statement.

For controls (such as Fanuc) which require a modal command to be output at the start of positioning moves the FIRST_RAPID_RATE_MOVE should be defined.

The example below shows part of the configuration file for a Fanuc control. Note that the RAPID_RATE_MOVE statement is actually the same as the FEED_RATE_STATEMENT shown previously and hence could have been omitted. The rest of the statements and values are the same as shown previously.

Example for Fanuc control

```
FIRST_RAPID_RATE_MOVE = "[N]G0[X|@][Y|@][Z|@]"
RAPID_RATE_MOVE = "[N][X][Y][Z]"
```

Output

N1226G0X123.450Y234.560Z5.670

N1228Y345.890Z4.210

If a feed rate variable ([F]) appears in a rapid statement the value used is that defined in the RAPID_FEED_RATE statement which was described previously in the global statements section. If no RAPID_FEED_RATE is defined the last cutting feed rate is output.

Change in feed rate

Some controls require changes in feed rate to be output as a separate line in the program file. To support these controls the post processor has an optional **FEED_RATE_CHANGE** statement. If this statement is defined and there is a change in the feed rate (between rapid, plunge or normal feed rates) a line will be output using this statement.

NOTE: the 'normal' XYZ move will also be output using the appropriate statement.

The example shown below assumes that the first move being output is the first move at rapid and hence the FEED_RATE_CHANGE statement is used and also the RAPID_RATE_MOVE.

Example for Roland control

```
FORMAT = [X|@||1.0|100.0]
FORMAT = [Y|@||1.0|100.0]
FORMAT = [Z|@||1.0|100.0]
FORMAT = [F|@||1.0|0.0166]
FEED_RATE_CHANGE = "V [F]"
RAPID_FEED_RATE = 3840

.
RAPID_RATE_MOVE="Z [X],[Y],[Z]"
```

Output

V 64

Z 12345,23456,567

Z 12345,34589,421

The rapid feed rate in this example was set to 3840 because the control has a maximum speed of 64 mm/s - this is equivalent to 3840 mm/min. The value is transformed into the required units using the [F|@||1.0|0.0166] variable specification which states that the feed rate value should be multiplied by 0.0166 (which is the same as dividing by 60) to change mm/min into mm/s.

The Roland control also uses positions specified in increments of 0.01mm hence all the X,Y and Z values are multiplied by 100.0 and output without any decimal places.

Toolchanges

The TOOLCHANGE statement is used to output one or more lines into the file when a toolchange occurs. The variable [T] holds the new toolnumber. By default most

of the standard configuration files do not have any TOOLCHANGE statements and hence ArtCAM will not allow toolpaths using multiple tools to be created for these machines. The lines specified by the TOOLCHANGE statements are only output for the 2nd and subsequent tool moves. Any formating required for the first toolchange should be included in the START statements which are output at the start of the file.

TOOLCHANGE =   **optional**

Example for FANUC control

```
;
; Toolchange moves (only output for
;                    2nd and later tools)
;
TOOLCHANGE = "[N]G91G28Z0"
TOOLCHANGE = "[N]G49H0"
TOOLCHANGE = "[N]T[T]M6"
TOOLCHANGE = "[N]G90G54"
```

End of file

These statements are used to specify the lines sent at the send of the file. As many lines as required can be specified using any of the defined variables. The variables XH, YH and ZH can be used to initiate movement back to the home tool position. This statement is optional.

END =   **optional**

Example for FANUC control

; return to reference point in Z

```
END = "[N]G28G91Z0"
```

; cancel tool length offset

```
END = "[N]G49H0
```

; return to reference point in XY

```
END = "[N]G28X0Y0"
```

; and end program

```
END = "[N]M30"
```

This may give an output like the following assuming LINE_NUM_INCREMENT = 2

```
N4798G28G91Z0
N4800G49H0
N4802G28X0Y0
N4804M30
```

Editing A Configuration File

Location of files

The post processor configuration files will be stored in a sub-directory called **postp**, below the directory in which ArtCAM was originally installed.

Change to the specified directory and copy the configuration file you wish to use as a starting point for your new post processor to a new file. e.g. copy the file **fanuc.con** to **new_mc.con**.

NOTE: Remember to change the "DESCRIPTION" statement within the file to ensure it is displayed with a different name to the original.

If creating a new file from scratch just create the file as a plain text file.

To display the new post processor, ArtCAM will need to be restarted as the list of post processors is built when the program starts up. However, any edits you make to the file will be effective immediately as the post processor configuration file is reloaded each time a file is postprocessed.

Modifying the post processor output

By editing the configuration file with a simple text editor such as NOTEPAD while ArtCAM is still running you can quickly and easily see the results of your changes on the output. Simply perform the following steps repeatedly until the desired output is achieved. The following assumes that the **DESCRIPTION** field in the new post processor was set to "Test Post" and **FILE_EXTENSION** was set to "tst"

❑ Modify the configuration file (e.g. **new_mc.con**) using NOTEPAD

❑ Save the configuration file without exiting NOTEPAD

❑ Switch back to ArtCAM

❑ Choose Test Post as the output and save the post processed file.

❑ Select and output file e.g. **test.tst** - overwrite any existing version

❑ View the output using another copy of NOTEPAD

❑ If the output is not correct repeat this procedure

Post Processor Statements

Alphabetical list of statements

This section provides an alphabetical listing of the commands which may appear in a configuration file. For more details of how the statements are used refer back to the earlier sections in this guide.

DESCRIPTION

This statement specifies the string which is displayed to the user in the drop down list used to select a post processor.

default

> The name of the configuration file

END

This statement is used to specify the lines sent at the end of the file. As many lines as required can be specified using any of the defined variables except X, Y and Z. The variables XH, YH and ZH can be used to initiate movement back to the home tool position.

default

> None

END_OF_LINE

This statement is used to specify characters output at the end of every line.

Example

> END_OF_LINE = "[13][10]"

will add a carriage return, line feed combination to the end of each output line. This is the standard line termination for DOS text files.

default

> None

FEED_RATE_CHANGE

This statement allows the user to specify a separate command which is output if the feedrate changes. This field is optional and the default is to not output anything different (although by using [F|#] feedrates can be output on the normal move line only when there is a change of feed-rate).

default

None

FEED_RATE_MOVE

This statement controls the output of normal cutting moves at feedrate. As these are the most common moves in the file a number of examples will be shown. The first move at feed rate will use the FIRST_FEED_RATE_MOVE statement if it is defined. These statements are also used for outputting plunge moves as these are also linear moves (though only moving in Z), however the feedrate will be the plunge feedrate specified within ArtCAM. All examples include correct formatting for variables set up using the FORMAT statement.

Example output is demonstrated for two moves at feedrate 789.12 after a rapid move

X123.45	Y234.56	Z5.67
X123.45	Y345.89	Z4.21

Fanuc

```
FORMAT = [X|#|X|1.3]
FORMAT = [Y|#|Y|1.3]
FORMAT = [Z|#|Z|1.3]
FORMAT = [F|@|F|1.1]
FORMAT = [N|@|N|1.0]
.
.
FIRST_FEED_RATE_MOVE = "[N]G1[X|@][Y|@][Z|@][F]"
FEED_RATE_MOVE = "[N][X][Y][Z]"

N1226G1X123.450Y234.560Z5.670F789.1
N1228Y345.890Z4.210
```

Roland

```
FORMAT = [X|@||1.0|100.0]
FORMAT = [Y|@||1.0|100.0]
FORMAT = [Z|@||1.0|100.0]
FORMAT = [F|@||1.0|0.0166]
FEED_RATE_CHANGE = "V [F]"
.
.
FEED_RATE_MOVE = "Z [X],[Y],[Z]"


V 13
Z 12345,23456,567
Z 12345,34589,421
```

Heidenhain

```
FORMAT = [X|#|X|+1,3]
FORMAT = [Y|#|Y|+1,3]
FORMAT = [Z|#|Z|+1,3]
FORMAT = [F|#||1,0]
.
.
FEED_RATE_MOVE = "[N] L [X][Y][Z] F[F] M"


1226 L X+123,450 Y+234,560 Z+5,670 F789 M
1228 L Y+345,890 Z+4,210 F M
```

default

None - but user must supply definition

FILE_EXTENSION

default

None - but user should supply definition

FIRST_FEED_RATE_MOVE

This statement is used for formating the first move made at feed rate. If it is not defined the move is formated using the FEED_RATE_MOVE statement.

default

> None

FIRST_RAPID_RATE_MOVE

This statement is used for formating the first move made at rapid rate. If it is not defined the move is formated using the RAPID_RATE_MOVE statement.

default

> None

FORMAT

This statement can appear a number of times specifying the default formatting for variables. Any field set here can still be overridden explicitly when the variable is later used. All fields except the variable name are optional and the default values are

FORMAT = [V|O|S|F|m]

> Where

```
V  = Letter representing variable.
O  = @ if value always output.
     # if only output if changed
       since last value output.
S  = String to print before value.
F  = Format string
```

```
                         m  = Scaling factor - a floating
                              point number by which variable
                              is multiplied before formatting
                              is applied.
```

default

```
    X,Y,Z,XH,YH,ZH,R  [V|@||1.3|1.0]
    S,F,FR,FC,FP,N     [V|@||1.0|1.0]
```

LINE_NUM_INCREMENT

Specifies the increment used for the line numbers
(accessed through [N] variable).

default

LINE_NUM_INCREMENT = 2

LINE_NUM_MAXIMUM

Specifies the maximum value allowed for line numbers.
When the line number value exceeds this value, line
numbering will start again at LINE_NUM_START.

default

LINE_NUM_MAXIMUM = 999999

LINE_NUM_START

Specifies the starting value for line numbers (accessed through [N] variable).

default

LINE_ NUM_START = 10

ROTARY_AXIS

This is only used in post processors which are used to drive a machine using a rotary axis. Either the X or Y direction of the relief is 'wrapped' onto the rotary axis.

If this statement is present ArtCAM will display the following dialog. The default value for the diameter will assume that the complete length of the relief along the axis being wrapped is the circumference of the cylinder.
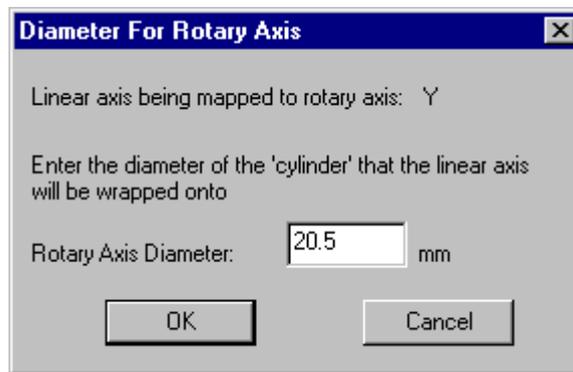
If X is specified then where [X] would have appeared in the file [R] should be used to output the rotary move.

possible values:

X   - all moves in X are wrapped onto rotary axis

Y   - all moves in Y are wrapped onto rotary axis

default

None

RAPID_RATE_MOVE

This statement controls the output of tool positioning moves at rapid feedrate. The first move at feed rate will use the FIRST_RAPID_RATE_MOVE statement if it is defined. These statements are also used for outputting retract moves as these are also rapid moves (though only moving in Z). All examples include correct formatting for variables set up using the FORMAT statement. If no FIRST_RAPID_RATE_MOVE is specified RAPID_RATE_MOVE is used. If no RAPID_RATE_MOVE is specified the FEED_RATE_MOVE is used (with feed rate set to RAPID_FEED_RATE).

Example output is demonstrated for two moves at rapid feed rate

X123.45	Y234.56	Z5.67
X123.45	Y345.89	Z4.21

Fanuc

```
FORMAT = [X|#|X|1.3]
FORMAT = [Y|#|Y|1.3]
FORMAT = [Z|#|Z|1.3]
FORMAT = [N|@|N|1.0]
.
.
FIRST_RAPID_RATE_MOVE = "[N]G0[X|@][Y|@][Z|@]"
RAPID_RATE_MOVE = "[N][X][Y][Z]"


N1226G0X123.450Y234.560Z5.670
N1228Y345.890Z4.210
```

Roland

```
FORMAT = [X|@||1.0|100.0]
FORMAT = [Y|@||1.0|100.0]
FORMAT = [Z|@||1.0|100.0]
FORMAT = [F|@||1.0|0.0166]
FEED_RATE_CHANGE = "V [F]"
RAPID_FEED_RATE = 3840
.
.
RAPID_RATE_MOVE="Z [X],[Y],[Z]"


V 64
Z 12345,23456,567
Z 12345,34589,421
```

Heidenhain

```
FORMAT = [X|#| X|+1,3]
FORMAT = [Y|#| Y|+1,3]
FORMAT = [Z|#| Z|+1,3]
FORMAT = [F|#||1,0]
RAPID_FEED_RATE = 9999
.
.
RAPID_RATE_MOVE = "[N] L[X][Y][Z] F[F] M"


1226 L X+123,450 Y+234,560 Z+5,670 F999 M
1228 L Y+345,890 Z+4,210 F M
```

RETRACT_MOVE

This statement is used for formating the first retract move made. If it is not defined the move is formated using the FIRST_RAPID_RATE_MOVE statement.

default

> None

START

This statement is used to specify the lines sent at the start of the file. As many lines as required can be specified using any of the defined variables except X, Y and Z. The variables XH, YH and ZH can be used to initiate movement from the home tool position.

default

> None

See the more detailed description and examples earlier in this document for more details.

TOOLCHANGE

This statement is used to specify one or more lines which are output for the 2nd and subsequent tool changes.

NOTE: The output for the first tool should be included in the START section of the configuration file.

As many lines as required can be specified using any of the defined variables except X, Y and Z. The variables XH, YH and ZH can be used to initiate movement to and from the home tool position. The variable T contains the new tool number.

default

None

See the more detailed description and example for the FANUC at the end of this document for more details.

UNITS

possible values:

MM   - all positions in mm, feed rates in mm/min

INCH - all positions in inches, feed rates in inches/min

default

UNITS = MM

Specifies the units which the variables will be output using. This setting is valid whatever units are used for

the actual model within ArtCAM. e.g. if the ArtCAM model use is mm and the command

UNITS = INCH

appears in the configuration file all moves and feed-rates will be converted to inches.

Example Configuration Files

Introduction

This section includes a listing of three of the configuration files supplied with ArtCAM. Examples of the output produced by each file are also provided. These examples should help in finding the right style of variable formatting to use when a particular output format is desired.

Heidenhain

```
;
; Heidenhain configuration file Metric version
;
; History
;
; Who When      What
; === ======== =======================================
; BEM 01/04/94 Written
; bem 21/03/00 Added DESCRIPTION and FILE_EXTENSION fields
;
DESCRIPTION = "Heidenhain (*.tap)"
;
FILE_EXTENSION = "tap"
;
UNITS = MM
;
; Carriage return - line feed at end of each line
;
END_OF_LINE = "[13][10]"
;
; Block numbering
;
LINE_NUM_START     = 0
LINE_NUM_INCREMENT = 1
LINE_NUM_MAXIMUM   = 999999
;
; Set up default formatting for variables
;
; Spindle Speed
FORMAT = [S|@|S|1,3]
; Feed Rate
FORMAT = [F|#||1.0]
; Tool moves in x,y and z
FORMAT = [X|#| X|+1,3]
FORMAT = [Y|#| Y|+1,3]
FORMAT = [Z|#| Z|+1,3]
; Home tool positions
FORMAT = [XH|@|X|+1,3]
FORMAT = [YH|@|Y|+1,3]
FORMAT = [ZH|@|Z|+1,3]
;
; Set up program header
;
START = "[N] BEGIN PGM 1248 MM"
START = "[N] TOOL DEF 1 L+0,000 R+0,000"
START = "[N] TOOL CALL 1 Z [S]"
START = "[N] L [ZH] F9999 M03"
START = "[N] L [XH] [YH] [ZH] F9999 M03"
;
; Program moves
;
RAPID_FEED_RATE = 9999
FEED_RATE_MOVE  = "[N] L[X][Y][Z] F[F] M"
;
; End of file
;
END = "[N] STOP M02"
END = "[N] END PGM 1248 MM"
```

Example Output For Heidenhain Configuration File

```
0 BEGIN PGM 1248 MM
1 TOOL DEF 1 L+0,000 R+0,000
2 TOOL CALL 1 Z S3000,000
3 L Z+10,000 F9999 M03
4 L X+0,000 Y+0,000 Z+10,000 F9999 M03
5 L X+0,001 Y+9,999 Z+10,000 F M
6 L Z+0,000 F360 M
7 L X+0,101 F720 M
8 L X+3,000 F M
9 L X+3,100 Z+0,012 F M
10 L X+3,200 Z+0,182 F M
.
.
.
225 L X+3,800 Z+0,092 F M
226 L X+3,900 Z+0,142 F M
227 L X+4,600 F M
228 L X+4,700 Z+0,092 F M
229 L X+4,800 Z-0,000 F M
230 L X+9,999 Y+0,002 Z+0,000 F M
231 L Z+10,000 F9999 M
232 L X+0,000 Y+0,000 F M
233 STOP M02
234 END PGM 1248 MM
```

Fanuc

```
;
; Fanuc configuration file
;
; History
;
; Who When     What
; === ======== =======================================
; BEM 01/04/94 Written
; BEM 18/07/97 Added toolchange output
; bem 21/03/00 Added DESCRIPTION and FILE_EXTENSION fields
;
DESCRIPTION = "Fanuc (*.fan)"
;
FILE_EXTENSION = "fan"
;
;
UNITS = MM
;
; Cariage return - line feed at end of each line
;
END_OF_LINE = "[13][10]"
;
; Block numbering
;
LINE_NUM_START     = 0
LINE_NUM_INCREMENT = 10
LINE_NUM_MAXIMUM   = 999999
;
; Set up default formating for variables
;
; Line numbering
FORMAT = [N|@|N|1.0]
; Spindle Speed
FORMAT = [S|@|S|1.0]
; Feed Rate
FORMAT = [F|#|F|1.1]
; Tool moves in x,y and z
FORMAT = [X|#|X|1.3]
FORMAT = [Y|#|Y|1.3]
FORMAT = [Z|#|Z|1.3]
; Home tool positions
FORMAT = [XH|@|X|1.3]
FORMAT = [YH|@|Y|1.3]
FORMAT = [ZH|@|Z|1.3]
;
; Set up program header
;
START = "%"
START = ":1248"
START = "[N]G91G28X0Y0Z0"
START = "[N]G40G17G80G49"
START = "[N]T1M6"
START = "[N]G90G54"
START = "[N]G43[ZH]H1"
START = "[N]G0[XH][YH][S]M3"
;
; Program moves
;
FIRST_RAPID_RATE_MOVE  = "[N]G0[X][Y][Z]"
FIRST_FEED_RATE_MOVE   = "[N]G1[X][Y][Z][F]"
FEED_RATE_MOVE         = "[N][X][Y][Z]"
;
```

```
; Toolchange moves (only output for 2nd and later tools)
;
TOOLCHANGE = "[N]G91G28Z0"
TOOLCHANGE = "[N]G49H0"
TOOLCHANGE = "[N]T[T]M6"
TOOLCHANGE = "[N]G90G54"
;
; End of file
;
END = "[N]G28G91Z0"
END = "[N]G49H0"
END = "[N]G28X0Y0"
END = "[N]M30"
```

Example Output For Fanuc Configuration File

```
%
:1248
N10G91G28X0Y0Z0
N20G40G17G80G49
N30T1M6
N40G90G54
N50G43Z10.000H1
N60G0X0.000Y0.000S3000M3
N70X0.001Y9.999
N80G1X0.001Y9.999Z0.000F360.0
N90G1X0.101Y9.999Z0.000F720.0
N100X3.000
N110X3.100Z0.012
N120X3.200Z0.182
N130X3.300Z0.249
.
.
.
N2240Y0.003
N2250X0.103
N2260X3.700
N2270X3.800Z0.092
N2280X3.900Z0.142
N2290X4.600
N2300X4.700Z0.092
N2310X4.800Z-0.000
N2320X9.999Y0.002Z0.000
N2330G0X9.999Y0.002Z10.000
N2340G0X0.000Y0.000Z10.000
N2350G28G91Z0
N2360G49H0
N2370G28X0Y0
N2380M30
```

Roland

```
;
; Roland configuration file
;
;
; History
;
; Who When     What
; === ======== ======================================
; BEM 01/04/94 Written
; BEM 13/05/94 Modified header to remove @ command
; TM  07/02/96 Modified - added pause command
; TM  07/02/96 Modified - Additional XY feedrates
; BEM 21/03/00 Added DESCRIPTION and FILE_EXTENSION fields
;
DESCRIPTION = "Roland (*.rol)"
;
FILE_EXTENSION = "rol"
;
; Set units used in file - MM
;
UNITS = MM
;
; Cariage return - line feed at end of each line
;
END_OF_LINE = "[13][10]"
;
; Set up default formating for variables
;
; Feed Rate
;
; Roland use mm/s for output feed rate so divide by 60
; this is ame as * (1/60)
;
FORMAT = [F|@||1.0|0.0166]
;
; Tool moves in x,y and z
;
; Roland uses positions in 0.01 of a mm
;
FORMAT = [X|@||1.0|100]
FORMAT = [Y|@||1.0|100]
FORMAT = [Z|@||1.0|100]
; Home tool positions
FORMAT = [XH|@||1.0|100]
FORMAT = [YH|@||1.0|100]
FORMAT = [ZH|@||1.0|100]
;
; Set up program header
;
; Reset all values to default except P1 and P2
;
START = "^DF"
;
; Pause before moving to start position
;
START = "!NR"
;
; Set up Z2 to safe z (home z) - raise spindle
; move to home position and start spindle
;
;START = "@0,[ZH]M[XH],[YH]! 1"
START = "! 1;Z [XH],[YH],[ZH]"
```

```
;
; Start moving at rapid rate
;
START = "V 64;F 64"
;
; Set feed rate used for rapid moves 64mm/s = 3840mm/min
;
RAPID_FEED_RATE = 3840
;
; Set up data output on change of feed rate
;
FEED_RATE_CHANGE = "V [F];F [F]"
;
; Program moves
;
FEED_RATE_MOVE  = "Z [X],[Y],[Z]"
;
; End of file - go home and stop spindle
;
END = "H"
```

Example Output For Roland Configuration File

```
^DF
@0,1000M0,0! 1
V 64
Z 0,1000,1000
V 6
Z 0,1000,0
V 12
Z 10,1000,0
Z 300,1000,0
Z 310,1000,1
Z 320,1000,18
Z 330,1000,25
Z 360,1000,52
Z 380,1000,64
Z 390,1000,74
Z 400,1000,79
.
.
.
Z 0,100,-0
Z 0,90,-0
Z 0,0,-0
Z 10,0,-0
Z 370,0,-0
Z 380,0,9
Z 390,0,14
Z 460,0,14
Z 470,0,9
Z 480,0,-0
Z 1000,0,0
V 64
Z 1000,0,1000
V 64
Z 0,0,1000
H
```

Rotary – Generic G- Code

```
;
; Linear Y Axis is wrapped onto A rotary axis
;
; History
;
; Who When     What
; === ======= ======================================
; BEM 26/07/99 Rewrote as more generic G-Code

; bem 21/03/00 Added DESCRIPTION and FILE_EXTENSION fields

;

DESCRIPTION = "Rotary (Y-axis) (*.tap)"

;

FILE_EXTENSION = "tap"
;
UNITS = MM
;
; Rotary axis control – Wrap Y axis moves onto rotary axis
;
ROTARY_AXIS = Y
;
; Cariage return - line feed at end of each line
;
END_OF_LINE = "[13][10]"
;
; Set up default formating for variables
;
; Line numbering
FORMAT = [N|@|N|1.0]
; Spindle Speed
FORMAT = [S|@|S|1.0]
; Feed Rate
FORMAT = [F|#|F|1.0]
; Tool moves in x,y and z
FORMAT = [X|#|X|1.4]
FORMAT = [Y|#|Y|1.4]
FORMAT = [Z|#|Z|1.4]
; Using A rotary axis with Y axis wrapped onto it
FORMAT = [R|#|A|1.4]
; Home tool positions
FORMAT = [XH|@|X|1.4]
FORMAT = [YH|@|Y|1.4]
FORMAT = [ZH|@|Z|1.4]
FORMAT = [RH|@|A|1.3]
;
; Set up program header
;
START = "T1M6"
START = "G0[ZH]"
START = "G0[XH][YH][S]M3"
;
; Program moves
;
FIRST_RAPID_RATE_MOVE  = "G0[X][R][Z][F]"
FIRST_FEED_RATE_MOVE   = "G1[X][R][Z][F]"
FEED_RATE_MOVE         = "[X][R][Z]"
;
```

```
; End of file
;
END = "G0[ZH]"
END = "G0[XH][YH]"
END = "M30"
```

Example Output For Rotary Configuration File

```
T1M6
G0Z10.0000
G0X0.0000Y0.0000S10000M3
X0.0010A0.0360Z5.0000
G1Z0.0000F600
G1A0.5544F1800
A161.2211
A167.4779Z0.0317
A175.8263Z0.0388
A201.3863Z0.0416
A211.2971Z0.0294
A217.0355Z-0.0000
A359.9639
X0.0154
X1.0006
A359.4455
A233.2067
A224.8619Z0.1790
A217.0391Z0.3012
A208.6907Z0.3927
A199.3019Z0.4549
A189.3911Z0.4777
A180.0023Z0.4607
A170.6135Z0.4062
A161.7431Z0.3140
A152.8763Z0.1791
A144.5315Z0.0000
A0.0360
X1.0151
X2.0004
A0.5544
A134.0961
A144.5253Z0.2239
A154.9581Z0.3883
.
.
.
A161.2218Z0.3715

A151.8330Z0.2394

A142.9662Z0.0612

A140.3562Z0.0001
A0.0360
X9.0127
X9.9979
A0.5544
A153.9182
A160.1750Z0.0323
A170.0858Z0.0398
A207.1226Z0.0425
A218.0774Z0.0301
A224.8598Z0.0001
X9.9980A359.9639Z0.0000
G0Z5.0000F9999
```

```
G0X0.0000A0.0000Z10.0000
G0Z10.0000
G0X0.0000Y0.0000
M30
```