PartMaker 2016

User Manual

User Guide/PartMaker ConfigPost

Delcam
Advanced
Manufacturing
Solutions

Important User Notices

Delcam Ltd has no control over the use made of the software described in this manual and cannot accept responsibility for any loss or damage howsoever caused as a result of using the software. Users are advised that all the results from the software should be checked by a competent person, in accordance with good quality control procedures.

The functionality and user interface in this manual is subject to change without notice in future revisions of software.

Acknowledgements

This documentation references a number of registered trademarks and these are the property of their respective owners. For example, Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States.

Patents

PartMaker software is subject to the following patents:

Patent granted: US 6, 112, 133 Visual system and method for generating a CNC program for machining parts with planar and curvilinear surfaces

Patent granted: US 6, 741, 905 Visual system for programming of simultaneous and synchronous machining operations on lathes

PartMaker 2016. Published December 2015

Contents

Chapter 1: ConfigPost Overview

Welcome to ConfigPost! This chapter explains the basic concepts of ConfigPost and outlines the remaining chapters in this guide.

Introduction to ConfigPost

The process of generating an NC Program for a specific CNC machine is called **Post Processing.**

Many CAM systems generate what is known as a CL file (Cutter Location file). Such CAM systems convert this CL file into a G-code program for a selected CNC machine by running a collection of computer programs called "Post Processors".

PartMaker bypasses the intermediate step of generating CL-Data; it generates CNC programs when the user chooses the "***Generate NC Program"*** command from PartMaker's ***Job Optimizer*** menu. Since formats of CNC programs vary from machine to machine PartMaker needs to know the program format structure for every specific machine.

The CNC machine specific program structure is described using the ConfigPost application and its description is stored in a file that the ConfigPost application produces. Such a file is called "**Post Processor Configuration File**", usually abbreviated as "**Post Config File**", "**Post Processor**", or most commonly, to "**Post File**".

ConfigPost Levels and Types

Two Levels of ConfigPost: Lite and Pro

ConfigPost has two levels of usability which are described in the table below. The level of ConfigPost that you are using may prevent you from modifying certain aspects of the post files to get desired output. All users of PartMaker Version 7.5 and higher have automatically licensed ConfigPost/Lite. ConfigPost/Pro is a separately licensed, optional module to PartMaker.

	Pro	**Lite**
Creating New post processors	X	
Entering data into unused formats	X	
Using the Import WRI, Advanced Customization, and Channels features.	X	
Creating new variables	X	
Creating new procedures	X	
Creating new user data	X	
Modifying previously used formats	X	X
Using reserved words and previously created variables/procedures	X	X
Assigning Variables	X	X

Two Types of ConfigPost/Pro: Basic and Advanced

There are two types of ConfigPost/Pro: ConfigPost/Pro Basic and ConfigPost/Pro Advanced. Both types will support the 'Config-by-Click'® feature. The PartMaker CAM modules that they support are described below.

The ConfigPost/Pro Basic is required to create Post files for the following PartMaker applications:

- PartMaker/Mill without 4-axis and 5-axis options
- PartMaker/Turn
- PartMaker/WireEDM

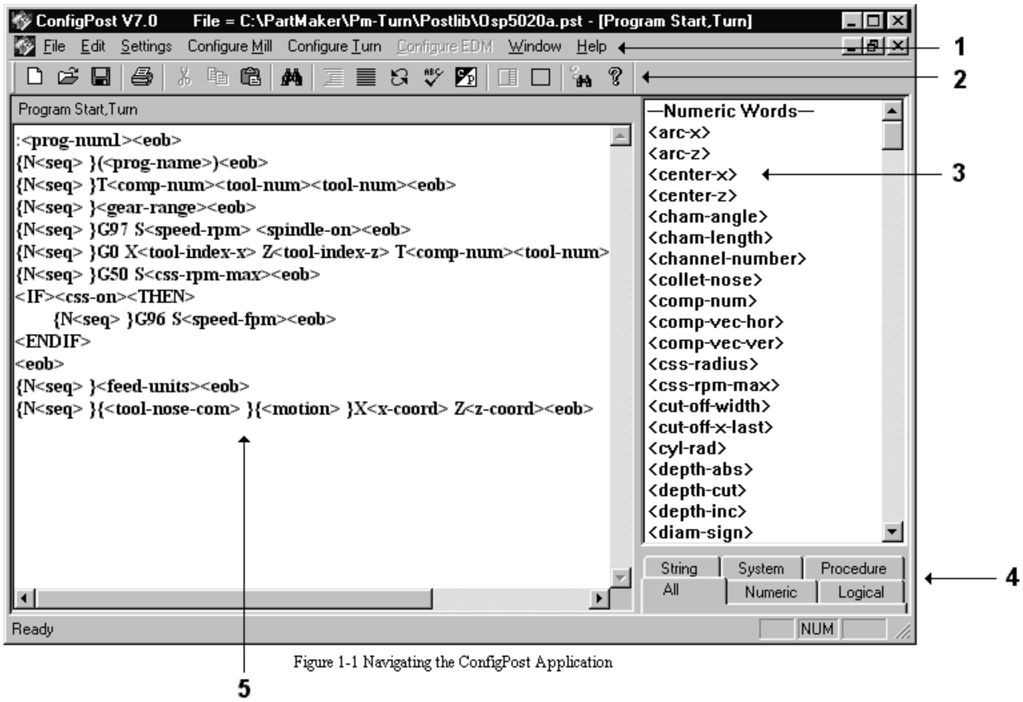
The ConfigPost/Pro Advanced is required to create Post files for the following PartMaker applications:

- PartMaker/TurnMill
- PartMaker/SwissCAM
- PartMaker/Mill with 4-axis and 5-axis options

Important! *Features not available with ConfigPost/Lite are noted where applicable.*

Navigating the ConfigPost Application

Pictured below in Figure 1-1 are the five main elements that you will be using while editing or creating a Post File.



```
:<prog-num1><eob>
{N<seq> }(<prog-name>)<eob>
{N<seq> }T<comp-num><tool-num><tool-num><eob>
{N<seq> }<gear-range><eob>
{N<seq> }G97 S<speed-rpm> <spindle-on><eob>
{N<seq> }G0 X<tool-index-x> Z<tool-index-z> T<comp-num><tool-num>
{N<seq> }G50 S<css-rpm-max><eob>
<IF><css-on><THEN>
      {N<seq> }G96 S<speed-fpm><eob>
<ENDIF>
<eob>
{N<seq> }<feed-units><eob>
{N<seq> }{<tool-nose-com> }{<motion> }X<x-coord> Z<z-coord><eob>
```

Figure 1-1 Navigating the ConfigPost Application

1 Config Post Menus

2 Config Post Toolbar

3 Reserved Word List

4 Reserved Word Selection Tabs

5 Program Format Window

Launching ConfigPost from the PartMaker Application

You can launch (start) ConfigPost from the **Job Optimizer Menu** within the PartMaker Application.
Open **PartMaker**; Go to the **Job Optimizer Menu**; Choose **Configure Post Processor** .

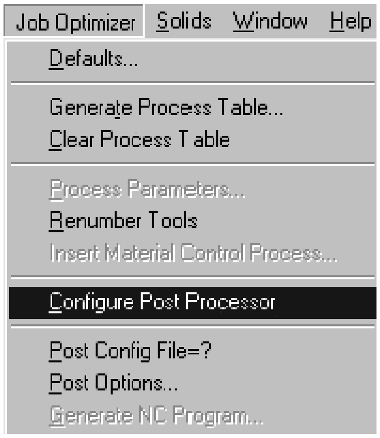


Figure 1-2 Configure Post Processor

- If you choose **Configure Post Processor** within PartMaker/Mill then ConfigPost with Target Application set to PartMaker/Mill will be opened.
- If you choose **Configure Post Processor** within PartMaker/Turn then ConfigPost with Target Application set to PartMaker/Turn will be opened.
- If you choose **Configure Post Processor** within PartMaker/WireEDM then ConfigPost with Target Application set to PartMaker/WireEDM will be opened.
- If you choose **Configure Post Processor** within PartMaker/Turn-Mill then ConfigPost with Target Application set to PartMaker/TurnMill will be opened.
- If you choose **Configure Post Processor** within PartMaker/SwissCAM then ConfigPost with Target Application set to PartMaker/SwissCAM will be opened.

If only basic ConfigPost is licensed then PartMaker/TurnMill and PartMaker/SwissCAM menus under Target Application will be disabled.

Using 'Config-by-Click'®

PartMaker's 'Config-by-Click'® feature is available for use in the PartMaker Mill, Turn, Turn-Mill, SwissCAM and Wire EDM modules. The 'Config-by-Click'® feature assists you in customizing the NC output of a PartMaker Post Config file by automatically placing the cursor in the correct program format in PartMaker's ConfigPost application.

See Appendix A 'Using PartMaker's Config-by-Click'® for more information on launching ConfigPost and initiating post processor customization with this powerful feature.

Starting ConfigPost as Stand-Alone Application

If you start ConfigPost as stand-alone application then the default Target Application will depend on the security device (dongle) configurations. If advanced ConfigPost is licensed then the default Target Application will be set to PartMaker/SwissCAM. If only the basic ConfigPost is licensed then the default Target Application will be set to PartMaker/Turn. PartMaker/TurnMill and PartMaker/SwissCAM menu commands under Target Application will be disabled. After you started ConfigPost as stand-alone application you can reset the Target Application.

If only the basic ConfigPost is licensed then PartMaker/TurnMill and PartMaker/SwissCAM menu commands under Target Application will be disabled.

What You Can Do With ConfigPost

The ConfigPost program lets you create and save custom post processor configuration files for the CNC machine you select to machine a part—this process is often referred to as configuring a post processor. Machine tool manufacturers use a variety of CNC program standards, but ConfigPost gives you the freedom to configure a post processor for virtually any CNC. These files commonly use an extension ".PST", e.g. ***FANUC.PST*** is a post processor configuration file for a machine equipped with a Fanuc control.

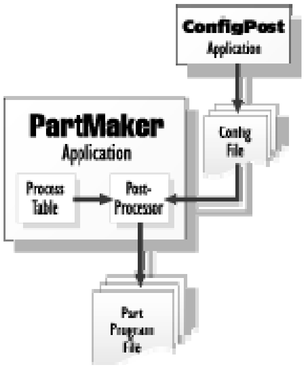


Figure 1-3 How ConfigPost Works

How Does ConfigPost Work?

After customizing a Post file for your CNC, you can generate a part program to machine any part using the Generate NC Program command in PartMaker's Job Optimizer menu. The data in the Process Table you create in PartMaker is converted into an NC program according to the format information in the Post file.

Figure 1-3 shows how a Post file you create in ConfigPost and a Process Table you create in PartMaker are used together to create a final part program for your CNC machine.

A Post file created in ConfigPost is used by PartMaker as a template to format the Process Table that is created in PartMaker. This template consists of program formats such as Program Start, Linear Move, and Tool Change that determine the appearance of a part program for a specific CNC.

Each program format uses letter address characters such as X and Y and reserved words such as <x-coord> and <y-coord> to indicate the sequence of data in each program format block. These letter address characters and reserved words determine how a Process Table is converted into a CNC part program.

Program formats are filled when PartMaker generates NC program. Reserved words in the Post file are substituted by their corresponding values and machine tool codes (called G and M codes) from the Process Table. The resulting part program is a text file that you can send to your CNC machine to machine a part.

Look at this sample data of a tool that will make two moves:

- Linear Move to X=2.5 in Y=1.5 in Z=-0.5 in Feedrate=10 in/min
- Linear Move to X=3.5 in Y=1.5 in Z=-0.5 in Feedrate=10 in/min

The final part program for these two moves would look like this:

- N25G1X2.5Y1.5Z-0.5F10.0
- N30G1X3.5Y1.5Z-0.5F10.0

Most CNC machines use a concept of modality to avoid redundant data in a part program. The same coordinates and/or commands do not need to be repeated in the part program. For such machines, redundant moves as in the previous example can be reduced to the following:

- N25G1X2.5Y1.5Z-0.5F10
- N30X3.5

In order to facilitate the output shown in the previous example, the Linear Move Format you specify in ConfigPost would look like:

```
{N <seq>}{<motion>}{X <x-coord>} {Y <y-coord>} {Z <z-coord>} {F <feed-upm>} <eob>
```

Figure 1-4 shows how PartMaker extracts all coordinate data from the Process Table and the format of the line from the Post file, and then the corresponding values are substituted to generate each line in the resulting part program.

{N <seq>}	{<motion>}	{X <x-coord>}	{Y<y-coord>}	{Z <z-coord>}	{F <feed-upm>}	<eob>
N 25	G1	X 2.5	Y 1.5	Z -0.5	F 10.0	<CR>

Figure 1-4 Example of a Linear Move format

As shown in Figure 1-4, you enter letter address characters N, X, Y, and Z as literals in each program format block of a Post file; they appear in the final part program file exactly as they appear in the program format block. Reserved words such as <seq**>** and <motion**>** identify the placement of corresponding values or strings in each program format block.

Important! *Each program format definition can include multiple lines, but each line must end with <eob>.*

Program formats determine the structure of a part program for a specific CNC machine as shown in the previous illustration, the example of a linear move format. For each process in a Process Table PartMaker converts program formats into lines of text by replacing reserved words with their corresponding values from the Process Table.

- In Figure 1-4, <x-coord>, <y-coord>, and <z-coord> are numeric reserved words that represent X, Y, and Z axis coordinates.
- <motion> is a string reserved word representing a type of motion (Rapid or Linear for example).
- <feed-upm> is a numeric reserved word representing the feed rate value in Units Per Revolution (UPM) for the line of code in the part program.
- <eob> is a string reserved word representing the end of a block (line) of code in a part program.
- The brackets { } are used as modality delimiters in program formats.

Using modality reduces the overall length of a part program—repeated coordinates and/or commands are automatically stripped from the final part program to prevent repetition of a reserved word whose value is the same as the previous occurrence of that word.

Note: *When modality delimiters are used with the {<seq>} reserved word, PartMaker will strip all sequence numbers from the part program during program generation. This is accomplished by specifying 0 for the "Block Increment" number in PartMaker's Post Options dialog for PartMaker/Mill, PartMaker/Turn and PartMaker/EDM.*

NC Program Output Flow Charts

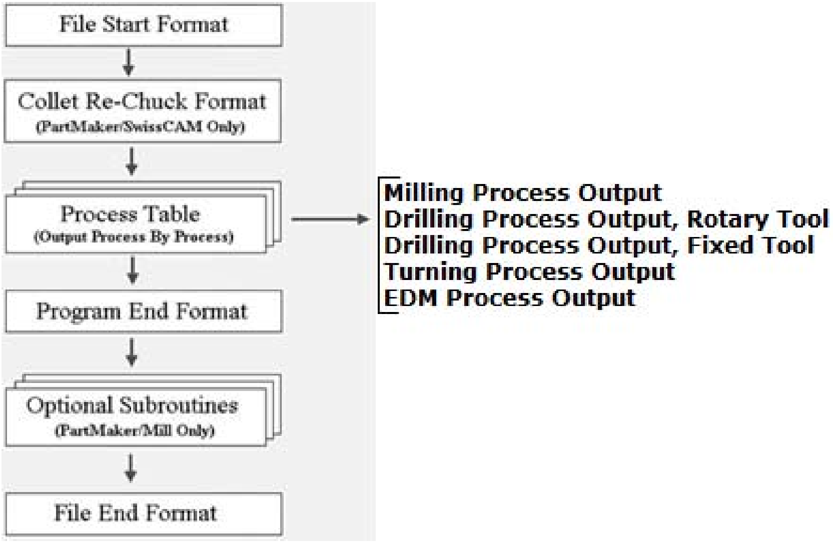


Figure 1-5 NC Program Output Flow Chart

Milling Process Output

The chart below shows how PartMaker outputs an NC program for a milling process. The first step is to output a Process Header Format as following: the Program Start Format is output if a given process is the first process in the Process Table, otherwise if a tool used in a given process needs to be activated (i.e. brought into position to begin cutting) the Tool Change Format is output, otherwise the Process Start Format is output.

Milling Process Output Steps

1> Process Header Format:

Program Start Format or
Tool Change Format or
Process Start Format

Process Header Formats contain tool and spindle related information such as Tool Number, Spindle Speed, Work offset.

2> Rapid Move Format (Tool moves to the Clearance plane)

3> Motion Format(s):

Linear Move Format(s) and/or
Circular Move Format(s) and/or
Rapid Move Format(s)

Motion Formats are used to output the tool path information into NC Program File.

Note: Motion Formats will be replaced by a Subroutine Call Format if subroutines are enabled in PartMaker/Mill

4> Process End Format

Drilling Process Output, Rotary Tool

The chart below shows how PartMaker outputs an NC program for a drilling process (as well as for any other hole making process such as tapping, reaming, etc.) using a **rotary ("live") tool.** The first step is to output a Process Header Format as following: the Program Start Format is output if a given process is the first process in the Process Table, otherwise if a tool used in a given process needs to be activated (i.e. brought into position to begin cutting) the Tool Change Format is output, otherwise the Process Start Format is output.

Drilling Process Output Steps

(rotary tool, milling face)

1> Process Header Format:

Program Start Format or
Tool Change Format or
Process Start Format

Process Header Formats contain tool and spindle related information such as Tool Number, Spindle Speed, Work offset.

2> Rapid Move Format (Tool moves to the Clearance plane)

3> Canned Cycle Format

4> Move to Hole Location Format(s)

Move to Hole Location Format is used to output the holes' locations into NC Program File.

Note: Move to Hole Location Format will be replaced by a Subroutine Call Format if subroutines are enabled in PartMaker/Mill.

5> Canned Cycle Cancel Format

6> Process End Format

Drilling Process Output, Fixed Tool

The chart below shows how PartMaker outputs an NC program for a drilling process (as well as for any other hole making process such as tapping, reaming, etc.) using a **fixed tool** making an axial hole on a turning face. The first step is to output a Process Header Format as following: the Program Start Format is output if a given process is the first process in the Process Table, otherwise if a tool used in a given process needs to be activated (i.e. brought into position to begin cutting) the Tool Change Format is output, otherwise the Process Start Format is output.

Drilling Process Output Steps (fixed tool, turning face)

1> Process Header Format

Program Start Format or
Tool Change Format or
Process Start Format

Process Header Formats contain tool and spindle related information such as Tool Number, Spindle Speed, Work offset.

2> Canned Cycle Format

Note: Canned Cycle Format can be replaced by Linear Move Format(s) and Rapid Move Format(s) if a "canned cycle" check box is not checked in the Process Parameters Dialog for a given process.

3> Process End Format

Turning Process Output

The chart below shows how PartMaker outputs an NC program for a turning process in the Process Table. The first step is to output a Process Header Format as following: the Program Start Format is output if a given process is the first process in the Process Table, otherwise if a tool used in a given process needs to be activated (i.e. brought into position to begin cutting) the Tool Change Format is output, otherwise the Process Start Format is output.

Turning Process Output Steps

1> Process Header Format

Program Start Format or
Tool Change Format or
Process Start Format

Process Header Formats contain tool and spindle related information such as Tool Number, Spindle Speed, Work offset.

2> Canned Cycle Format (optional)

3> Motion Format(s)

Linear Move Format(s) and/or
Circular Move Format(s) and/or
Rapid Move Format(s) and/or
Thread Move Format(s)

Motion Formats are used to output the tool path information into an NC Program File.

4> Process End

Note: *Some turning processes fall into category of Material Control Processes and they are handled as described in the section Handling Material Control Processes.*

Handling of EDM Processes in PartMaker

The chart below shows how PartMaker outputs an NC program for an EDM process in the Process Table. The first step is to output a Process Header Format as following: the Program Start Format is output if a given process is the first process in the Process Table, otherwise the Process Start Format is output.

EDM Process Output Steps

1> Process Header Format:

Program Start Format or
Process Start Format

Process Header Formats contain machine setup specific information

2> Motion Format(s):

Linear Move Format(s) and/or
Circular Move Format(s) and/or
Rapid Move Format(s) and/or
XY UV Move Format(s) (for 4-axis)

Motion Formats are used to output the tool path information into an NC Program File.

3> Process End

Chapter 2: Reserved Words and Program Formats

Using Reserved Words in ConfigPost

A reserved word is a symbol that represents a numeric value, a string (a sequence of alphanumeric characters such as MO4), a logical variable, or a conditional statement in a part program.

You define reserved words in program formats (Linear Move, for example) by enclosing each word with angular brackets (< >).

For example, **<tool-offset>** identifies a tool offset number used in the PartMaker Tools dialog. Additional ASCII characters used in program formats such as X, Y, Z, and F specify the letter for each word address.

There are four types of reserved words used in ConfigPost:

- Numeric
- String
- Logical
- System

Numeric Reserved Words

Numeric reserved words are replaced by their numeric values when PartMaker generates an NC Program. For example, the numeric reserved word <x-coord> is replaced by the current X-axis coordinate position.

Each numeric reserved word contains a corresponding Words Information record to specify its output format. See the Words Information Dialog section later in this chapter for more information.

Using the Prefix Characters $ and @

Numeric reserved words may be prefixed with a symbol such as $ or @ (<$ name> or <@ name> for example).

- The $ prefix signals ConfigPost to output the previous value of a reserved word.
- The @ prefix signals ConfigPost to output an incremental value (the difference between the current value of a reserved word and its previous value.)

String Reserved Words

String reserved words use string values—these strings are often assigned within ConfigPost itself. For example, <motion> can have values such as G0, G1, G2, and G3 that are defined in the Preparatory Functions dialog. Another example is <date>, whose value is the current date (July-31-03 for example).

Using Modality for String and Numeric Variables

Most CNC machines use the concept of modality to avoid redundant data, thereby reducing the overall length of a part program. The result is that repeated coordinates or commands are automatically stripped from the part program to prevent repeated occurrences of a reserved word whose value is the same as the previous occurrence of that word.

When you use modality delimiters with the <seq> reserved word, PartMaker will strip all sequence numbers from the part program during post processing. This is accomplished by specifying 0 for Block Increment in PartMaker’s Post Options dialog.

Important! *Never place modality delimeters around two reserved words, otherwise neither will be recognized.*

For example:

{<coolant>X<x-coord>} is wrong!

To make both those words modal, use the following syntax:

{<coolant>}{X<x-coord>}

Logical and System Reserved Words

Logical and system reserved words are used together to set up conditional statements in a Post file. This statements are evaluated by PartMaker. PartMaker will include or omit certain data from a program depending on whether a conditional statement is true or false.

Specifying Conditional Statements With Logical and System Reserved Words

Logical and system reserved words are used with program formats to specify conditional statements that are recognized and evaluated by PartMaker.

If a conditional statement is true, PartMaker includes certain sections of a program format in the part program. The format is:

```
<IF> <Logical reserved word> <THEN>
          ...
<ENDIF>
```

Example:

```
<IF><cw-spindle><THEN>
          M3
<ENDIF>
```

If a conditional statement is false, certain data from the program format is omitted. The format is:

```
<IFNOT><Logical reserved word><THEN>
          ...
<ENDIF>
```

Example:

```
<IFNOT><cw-spindle><THEN>
          M4
<ENDIF>
```

Examples:

Here are examples of conditional statements that use one of the mathematical comparison routines:

```
<IF><smaller>(3,4)<THEN>
          THREE IS LESS THAN FOUR<eob>
<ENDIF>
<IF><larger>(<tool-num>,26)<THEN>
          THIS TOOL NUMBER IS LARGER THAN 26<eob>
<ENDIF>
```

Here is a useful conditional statement that can be used to prompt the milling machine operator to turn the part over and press the cycle start button:

```
<IF><face-ang-chg><THEN>
          G0 Z<tool-change-z>M9<eob>
          Y<tool-change-z>M5<eob>
          M0<eob>
          (PLEASE TURN THE PART OVER)<eob>
          (AND PRESS THE CYCLE START BUTTON)<eob>
          G0 Y<y-coord><eob>
          Z<z-coord><eob>
<ENDIF>
```

Reserved Words List

Reserved Words for Mill

Word	Description	Type
<arc-cw>	This word is true if cutting a clock-wise arc	logical
<arc-x>	Used in a circular interpolation block to specify the signed X distance form the start point of an arc to the center of the arc along the X axis	numeric
<arc-y>	Used in a circular interpolation block to specify the signed Y distance form the start point of an arc to the center of the arc along the Y axis	numeric
<arc-z>	Used in a circular interpolation block to specify the signed Z distance form the start point of an arc to the center of the arc along the Z axis	numeric
<can-cyc>	Used to determine if a canned cycle is active **PartMaker dialog:** Window- Process Table - Process Parameters **Check box:** Canned Output	logical
<center-x>	Reproduces the absolute X coordinate to an arc's center in a circular block	numeric
<center-y>	Reproduces the absolute Y coordinate position of an arc's center in a circular block	numeric
<center-z>	Reproduces the absolute Z coordinate of an arc's center in a circular block	numeric
<clear>	Specifies the clearance value; reproduces Z clearance for Z-oriented tools or X clearance for X-oriented tools. **PartMaker dialog:** Window – Process Table -Process Parameters **Dialog fields:** Z_Clear (C) or X_Clear (C), depending on tool orientation	numeric
<clear-abs>	Specifies the absolute coordinate corresponding to clearance plane (X coordinate for X-oriented tools or Z coordinate for Z-oriented tools), attained when the tool is positioned at the clearance plan **PartMaker dialog:** Part Features - Hole Group Parameters **<clear-abs> = [z_surf(s) + z_clear(c)]** or **<clear-abs> = [x_surf(s) + x_clear(c)]**	numeric
<comp-end>	This word is true if the move represents the end section for Tool Diameter compensation (last element or move of path) <comp-end> is used in linear move formats.	logical
<comp-middle>	This word is true if the move represents the middle section for Tool Nose Radius compensation (between the first and last moves of path) <comp-middle> is used in linear move formats.	logical

Word	Description	Type
<comp-num>	Specifies the tool diameter compensation number **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Comp No:	numeric
<comp-on>	This word is true if Tool Nose Radius compensation is on <comp-on> is used in process header formats.	logical
<comp-start>	This word is true if the move represents the start section for Tool Nose Radius compensation (first element or move of path) <comp-start> is used in linear move formats.	logical
<comp-status>	Establishes a right/left tool relationship with the part when Tool Nose Radius compensation status is selected Typical values are G40, G41, and G42, assigned in the Preparatory Functions dialog and used in Linear Move Program Format **PartMaker dialog:** Window – Process Table - Process Parameters **Dialog field:** Assigns value based on selection of Nose Radius Compensation of None, Left or Right	string
<comp-vec-hor>	Specifies the horizontal cutter compensation vector component used with older Cincinnati Milacron controls; commonly called P.	numeric
<comp-vec-ver>	Specifies the vertical cutter compensation vector component used with older Cincinnati Milacron controls; commonly called Q	numeric
<coolant>	Generates the assigned M-code for coolant - Standard, High Pressure or Off **PartMaker dialog:** Job Optimizer - Defaults **Dialog field:** Coolant	string
<cw-spindle>	Used to determine if the spindle is rotating in the clockwise direction **PartMaker dialog:** ToolMinder - Tools **Dialog field:** True if the “Speed Factor” is set as a positive value	logical
<cycle>	Specifies a cycle type identifier for canned cycles Typical values for <cycle> are G81, G82, G83, etc. <cycle> is assigned in ConfigPost’s Preparatory Functions dialog.	string
<date>	Reproduces the date when the part program was post processed	string
<decel-status>	Specifies deceleration status This word holds the values of Inhibit/Resume, assigned in the Preparatory Functions dialog in ConfigPost.	string

Word	Description	Type
<depth>	Specifies the depth value; reproduces Z depth for Z-oriented tools or X depth for X-oriented tools **PartMaker dialog:** Part Features **-** Hole Group Parameters **Dialog field:** Nominal Depth (D)	numeric
<depth-abs>	Specifies the absolute X coordinate (for X oriented tools) or Z coordinate (for Z oriented tools) attained when the tool reaches the <depth> level **PartMaker dialog:** Part Features **-** Hole Group Parameters **<depth-abs> = [Nominal Depth(D) - Z_Surf(S)]** or **<depth-abs> = [Nominal Depth(D) - X_Surf(S)]**	numeric
<depth-inc>	Specifies the incremental Z-axis or X-axis depth. Calculated as: <depth-inc> = <depth> + <clear> **PartMaker dialog:** Part Features - Hole Group Parameters **<DEPTH-INC> = [Nominal Depth(D) + Z_Clear(C)]** or **<DEPTH-INC> = [Nominal Depth(D) + X_Clear(C)]**	numeric
<drilling>	This word is true if any one of the following tools are used: **Drill; Spot Drill; Center; Chamfer; Tap; Ream; Bore**	logical
<dwell>	Reproduces the dwell time for a tool. **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Dwell (sec):	numeric
<ENDIF>	Last element in a conditional statement (must be on a line by itself)	system
<eob>	Specifies the end of block code for each line of a part program. <eob> is assigned in ConfigPosts's General Information dialog	string
<eq-number>	This word is used for comparing the value of a numeric word to a number. For example, if the total time of the processes comes out to 1.2 min. the statement below is used: <IF><eq-number>(<time-total>,1.2)<THEN> time total =1.2<eob> <ENDIF> **-the line "**time total=1.2" **will be read into the nc program**	logical

Word	Description	Type
<eq-string>	This word is used for comparing a string word with a string. For example, if cutter compensation is set to the right, then the statement below is used: <IF><eq-string>(<comp-status>, G42)<THEN> comp is to the right <eob> <ENDIF> **-the line "**comp is to the right**" will be read into the nc program**	logical
<face-ang-chg>	This word is true if the angle of the face changes between two processes. For example, if the previous process was milling at a 0 degree angle (Mill ZY - machining function) and the current process is milling at a 90 degree angle (Mill ZY – machining function), then <face-ang-chg> is true for the current process.	logical
<face-angle>	Specifies the angle programmed in Mill XY, ZY, and ZX planes. <c-angle> can also be used in place of <face-angle> **PartMaker dialog:** View - Setup **Dialog field:** Angle(A) for **Face Windows** of Machining Function **Mill XY Plane** Angle(C) for **Face Windows** of Machining Function **Mill ZY Plane** and **Mill ZX Plane**	numeric
<face-change>	This word is true if there is a face change between two processes. For example, if the previous process was created on one Face Window and the current process is created on another Face Window, then <face-change> is true for the current process.	logical
<face-cyl>	Used to determine if the current process is a Cylinder function **PartMaker dialog:** View - Setup **Dialog field:** True if "Mill Cylinder" Machining Function is selected.	logical
<face-dia-index>	Used to determine if the current process is a Diameter Index function **PartMaker dialog:** View - Setup **Dialog field:** True if "Mill Diam, Index" Machining Function is selected.	logical
<face-name>	Specifies the name of the part face defined **PartMaker dialog:** View - Setup **Dialog field:** Rename To:	string
<face-note1>	Specifies the first "Note" line **PartMaker dialog:** View - Setup **Dialog field:** Notes	string

Word	Description	Type
<face-note2>	Specifies the second “Note” line **PartMaker dialog:** View - Setup **Dialog field:** Notes	string
<face-note3>	Specifies the third “Note” line **PartMaker dialog:** View - Setup **Dialog field:** Notes	string
<face-polygon>	Used to determine if the current process is a Polygon function **PartMaker dialog:** View - Setup **Dialog field:** True if “Mill Polygon” Machining Function is selected.	logical
<face-xy>	Used to determine if the current process is an XY function **PartMaker dialog:** View - Setup **Dialog field:** True if “Mill XY” Macining Function is selected.	logical
<face-z-coord>	Not used for PartMaker versions 4.7 and above	numeric
<feed-upm>	Specifies the programmed feed rate in units per minute. When the Process Table Display Option: Feeds in Units per Revolution and Surface Speed check box in the Defaults for Milling dialog is off, the feed rate will be output in units per minute. **PartMaker dialog:** Window – Process Table - Process Parameters **Dialog field:** Format of Cutting Feedrate and Plunge Feedrate	numeric
<feed-upr>	Specifies the programmed feed rate in units per revolution When the Process Table Display Option: Feeds in Units per Revolution and Surface Speed check box in the Defaults for Milling dialog is on, the feed rate will be displayed in units per revolution. **PartMaker dialog:** Window – Process Table - Process Parameters **Dialog field:** Format of Cutting Feedrate and Plunge Feedrate	numeric
<first-tool-use>	This word is true if it’s the first time a tool is called in the Process Header Format for Tool Change <first-tool-use> is typically used to build a tool list at the beginning of an NC program.	logical
<group-name>	Specifies the name of the group used in a process as shown in the process table **PartMaker dialog:** Part Features - Profile Group Parameters or ToolMinder - Cycles **Dialog field:** Cycle Name or Rename To:	string

Word	Description	Type
<header>	Used in Config Posts's Procedures to check what current Header Format is active: Program Start = PGS Tool Change = TCH Process Start = PRS Process End = PRE Example: **<IF><eq-string>(<header>,TCH)<THEN>**	string
<IF>	First element in a conditional statement, always followed by a logical reserved word to verify that a condition is true	system
<IFNOT>	First element in a conditional statement, always followed by a logical reserved word to verify that a condition is false	system
<include>	Specifies the user prompt value that queries the user for a value to be input at time of post processing Example: {What is your name:<include>Bob} Would prompt" What is your name" and make Bob the default; the <include> value is then entered into the program file exactly as entered by the operator The value of <include> is used, then forgotten. See Chapter 3, "ConfigPost Advanced Features" for more information about <include> statements.	string
<in-path>	This word is true if the tool has performed first non-rapid move in the tool path. For example, when configuring Thread Move (under Motion Formats in ConfigPost) to output a G92 canned thread cycle where the full path of the thread is only necessary for the first thread pass, the configuration should look as shown below: **<IFNOT><in-path><THEN>** **G92 X<x-coord> z<z-coord> f<pitch><eob>** **<ENDIF>** **<IF><in-path><THEN>** **X<x-coord><eob>** **<ENDIF>**	logical
<in-ramping>	This word is true if a Ramping operation is performed. **PartMaker dialog:** Part Features - Profile Group Parameters. **Dialog field:** Ramping.	logical
<in-subroutine>	This word is true within a subroutine section of a part program	logical

Word	Description	Type
<larger>	This word is true if the compared value is larger All: For example, if the total time of the processes is larger than 1.2 min. the statement below is used: <IF><larger>(<time-total>,1.2)<THEN> time total is greater then 1.2<eob> <ENDIF> **-the line "**time total is greater then 1.2" **will be read into the nc program**	logical
<metric-post>	Used to determine if the PartMaker Input setting is metric **PartMaker dialog:** View - Preferences **Radio button:** Metric	logical
<motion>	Produces the correct motion type (Rapid, Linear, Arc CW/Arc-C-CW, for example) for the various program formats <motion> is assigned in the ConfigPost Preparatory Functions dialog	string
<od-tool>	Used to determine if an X-oriented (OD) tool is being used. **PartMaker dialog:** ToolMinder - Tools True if the Orientation is set to "X-Tool"	logical
<part-length>	Specifies the defined length of the part **PartMaker dialog:** View - Setup **Dialog field:** Length (L)	numeric
<pitch>	Specifies the thread pitch, or 1/tpi **PartMaker dialog:** ToolMinder - Tools – Type: Tap **<pitch> = [1 / Threads per Inch]**	numeric
<process-id>	Specifies the process number as it appears in the **PartMaker** process table **PartMaker Window:** Window - Process Table **Dialog field:** Proc ID	string
<prog-name>	Reproduces the user-assigned output file name **PartMaker Window:** Job Optimizer – Generate NC program **Dialog field:** Save NC Program File As:	string
<prog-num1>	Specifies the program number that is to appear at the beginning of the nc program (for dual processor controls, <prog-num1> specifies the program number for main spindle or turret #1 programs **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Program #1 (under Program No.)	string

Word	Description	Type
<prompt>	Used similarly to <include>—except that the <prompt> value can be saved to a global variable. Example: {Enter a tool number <prompt> 1}<eob> ~<var-0><prompt><eob> <IF><eq-string>(<var-0>, 1)<THEN> tool-is-one<eob> <ENDIF> See Chapter 3, " ConfigPost Advanced Features" for more information about <prompt> statements.	string
<radius>	Reproduces an arc radius in a circular block The value of <radius> will be negative for arcs with angles greater than or equal to 180 degrees.	numeric
<rapid>	Specifies the rapid value; reproduces Z rapid for Z-oriented tools or X rapid for X-oriented tools **PartMaker dialog:** Part Features - Hole Group Parameters **Dialog field:** Z_Rapid (R) or X_Rapid (R), depending on tool orientation	numeric
<rapid-abs>	Specifies the absolute coordinate corresponding to rapid plane (X coordinate for X-oriented tools or Z coordinate for Z-oriented tools), attained when the tool is positioned at the Rapid plane **PartMaker dialog:** Part Features - Hole Group Parameters **<rapid-abs> = [Z_Surf(S) + Z_Rapid(R)]** or **<rapid-abs> = [X_Surf(S) + X_Rapid(R)]**	numeric
<retract>	Holds the values of Retract to Clear or Retract to Rapid when performing drilling operations <retract> is assigned in the ConfigPost Preparatory Functions dialog	string
<seq>	Specifies the sequence number of the current line Used only in Turn and Mill **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Block start and Block Increment	numeric
<smaller>	This word is true if the compared value is smaller For example, if the total time of the processes is smaller than 1.2 min. the statement below is used: <IF><smaller>(<time-total>,1.2)<Then> time total is smaller then 1.2<eob> <ENDIF> **-the line "**time total is smaller then 1.2" **will be read into the nc program**	logical

Word	Description	Type
<speed-fpm>	Specifies the spindle speed in constant surface speed mode, defined in Feet Per Minute (FPM) – rarely used for milling When the Process Table Display Option: Feeds in Units per Revolution and Surface Speed check box in the Defaults for Milling dialog is on, speed will be displayed in feet per minute.	numeric
<speed-rpm>	Specifies the spindle speed in fixed mode, defined in Revolutions Per Minute (RPM) **PartMaker dialog:** Window – Process Table - Process Parameters **Dialog field:** Revolution per Minute (RPM)	numeric
<spindle-on>	Used to specify spindle rotation direction <spindle-on> is assigned in the ConfigPost Preparatory Functions dialog in both Mill and Turn sections. **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Speed Factor (If it's a positive value, then <spindle-on> will output as **CW** direction. If it's a negative value, then <spindle-on> will output as **CCW** direction.	string
<step>	Specifies the tool step over in the X or Z axis (pecking amount when performing deep hole drilling) **PartMaker dialog:** ToolMinder - Tools – Type: Drill **Dialog field:** Axial Step	numeric
<step-abs>	Specifies the absolute coordinate corresponding to the first step or peck (X coordinate for X-oriented tools or Z coordinate for Z-oriented tools), attained after the first step or peck into the stock **PartMaker dialog:** Part Features - Hole Group Parameters **<step-abs> = [Z_Surf(S) - Axial Step]** or **<step-abs> = [X_Surf(S) - Axial Step]**	numeric
<step-inc>	Specifies the incremental first step **PartMaker dialog:** Window – Process Table – Process Parameters **<step-inc> = [Z-Clear(C) + Axial Step]** or **<step-inc> = [X_Clear(C) + Axial Step]**	numeric
<stock-id>	Reproduces the inside stock diameter for tubing or a pre-drilled workpiece **PartMaker dialog:** View - Setup **Dialog field:** ID	numeric

Word	Description	Type
<stock-od>	Reproduces the outside stock diameter of the workpiece **PartMaker dialog:** View - Setup **Dialog field:** OD	numeric
<subroutine-num>	Specifies the subroutine number identifier (system generated), was called "<macro-num>" prior to version 6.	numeric
<surf>	Specifies a surface on the workpiece where all clearances and depths are measured. <surf> for z-oriented tools is commonly set to 0. <surf> for x-oriented tools is commonly set to the machined part radius. **PartMaker dialog:** Part Features - Hole Group Parameters **Dialog field:** X_Surf(S) or Z_Surf (S)	numeric
<then>	Second element in a conditional statement, placed after a logical reserved word	system
<thread-mill>	Used to determine if a thread mill tool is being used **PartMaker dialog:** ToolMinder – Tools – Type: Thread Mill True if Type is set to "Thread Mill"	logical
<time-process>	Specifies the time of the tool use from the tool change until the next tool change **PartMaker dialog:** Window - Process Table **Dialog field:** Time (column on Process Table)	numeric
<time-total>	Specifies the total cutting time of all processes combined **PartMaker Window:** Window - Process Table **Dialog field:** Total Time	numeric
<tool-change-x>	Specifies the X coordinate for the tool change position Used in Mill only **PartMaker dialog:** Job Optimizer - Defaults **Dialog field**: Tool Ch X	numeric
<tool-change-y>	Specifies the Y coordinate for the tool change position Used in Mill only **PartMaker dialog:** Job Optimizer - Defaults **Dialog field**: Tool Ch Y	numeric
<tool-change-z>	Specifies the Z coordinate for the tool change position Used in Mill only **PartMaker dialog:** Job Optimizer - Defaults **Dialog field**: Tool Ch Z	numeric
<tool-diam>	Specifies the tool diameter **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Diameter(d)	numeric

Word	Description	Type
<tool-id>	Reproduces the tool identification number assigned to each tool **PartMaker dialog:** ToolMinder- Tools **Dialog field**: Tool ID	numeric
<tool-moves-in>	This word is true if the tool is moving into the stock Example - <z-coord> is less than <$z-coord> in ConfigPost’s Linear Move Formats	logical
<tool-moves-out>	This word is true if the tool is moving away from the stock Example - <z-coord> is greater than <$z-coord> in ConfigPost’s Linear Move Formats	logical
<tool-name>	Specifies the name of the tool **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Rename To:	string
<tool-note>	Specifies the note line from **PartMaker** Tools dialog **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Notes:	string
<tool-num>	Specifies the tool number **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Tool No:	numeric
<tool-num-next>	Represents the next tool number in the Process Table different from the current tool number. It may or may not be the tool of the next process. **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Tool No. (of next tool called from the process table)	numeric
<tool-offset>	Specifies the tool length/geometry offset number **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Offset No:	numeric
<tool-tap>	**Obsolete** True if a tapping tool (according to the Tool Type in PartMaker Tool Data dialog) is used in a process, otherwise <tool-tap> is false	logical
<tool-type>	Specifies the type of tool being used. <tool-type> can be equal to either of the following: **DR** (drill); **SD** (spot); **CH** (chamfer); **BO** (bore); **EM** (end mill); **CE** (center); **TA** (tap); **RE** (ream); **SM** (slot mill); **TM** (thread mill); **FM** (face mill); **DT** (dove tail); **CR** (corner round) **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Type:	string
<work-offset>	Specifies the work offset code programmed for each Face Window **PartMaker dialog:** View - Setup **Dialog field:** Work Offset	string

Word	Description	Type
<work-shift>	Reproduces the distance from the part face to the main spindle cap (absolute Z0) Used only in Turn and Turn-Mill **PartMaker dialog:** View - Setup **Dialog field:** Work Shift (I)	numeric
<x-coord>	Specifies the X-axis coordinate value with respect to the part origin	numeric
<y-coord>	Specifies the Y-axis coordinate value with respect to the part origin	numeric
<z-face-location>	Specifies the Z coordinate of the part surface. **PartMaker dialog:** View - Setup **Dialog field**: Z Face (F).	numeric
<z-coord>	Specifies the Z-axis coordinate value with respect to the part origin	numeric

Reserved Words for Turn

Word	Description	Type
<arc-cw>	This word is true if cutting a clock-wise arc	logical
<arc-x>	Used in a circular interpolation block to specify the signed X distance form the start point of an arc to the center of the arc along the X axis	numeric
<arc-z>	Used in a circular interpolation block to specify the signed Z distance form the start point of an arc to the center of the arc along the Z axis	numeric
<can-cyc>	Used to determine if a canned cycle is active **PartMaker dialog:** Window- Process Table - Process Parameters **Check box:** Canned Output	logical
<center-x>	Reproduces the absolute X coordinate to an arc’s center in a circular block	numeric
<center-z>	Reproduces the absolute Z coordinate of an arc’s center in a circular block	numeric
<cham-angle>	Specifies the chamfer angle at the end of a thread **PartMaker dialog:** Profile Group Parameters - Turning Cycle – Threading - “Chamfer” option **Dialog field:** Chamfer Angle (b)	numeric
<cham-length>	Specifies the chamfer length at the end of a thread **PartMaker dialog:** Profile Group Parameters - Turning Cycle – Threading - “Chamfer” option **Dialog field:** Chamfer Length (L2)	numeric
<comp-end>	This word is true if the move represents the end section for Tool Diameter compensation (last element or move of path) <comp-end> is used in linear move formats.	logical
<comp-middle>	This word is true if the move represents the middle section for Tool Nose Radius compensation (between the first and last moves of path) <comp-middle> is used in linear move formats.	logical
<comp-num>	Specifies the tool diameter compensation number **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Comp No:	numeric
<comp-on>	This word is true if Tool Nose Radius compensation is on <comp-on> is used in process header formats.	logical
<comp-start>	This word is true if the move represents the start section for Tool Nose Radius compensation (first element or move of path) <comp-start> is used in linear move formats.	logical

Word	Description	Type
<comp-status>	Establishes a right/left tool relationship with the part when Tool Nose Radius compensation status is selected Typical values are G40, G41, and G42, assigned in the Preparatory Functions dialog and used in Linear Move Program Format **PartMaker dialog:** Window – Process Table - Process Parameters **Dialog field:** Assigns value based on selection of Nose Radius Compensation of None, Left or Right	string
<comp-vec-hor>	Specifies the horizontal cutter compensation vector component used with older Cincinnati Milacron controls; commonly called P.	numeric
<comp-vec-ver>	Specifies the vertical cutter compensation vector component used with older Cincinnati Milacron controls; commonly called Q	numeric
<coolant>	Generates the assigned M-code for coolant - Standard, High Pressure or Off **PartMaker dialog:** Job Optimizer - Defaults **Dialog field:** Coolant	string
<css-on>	This word is true if Constant Surface Speed is active Used to determine if <speed-fpm> should be output. **PartMaker dialog:** Window- Process Table - Process Parameters **Check Box:** Constant Surface Speed	logical
<css-radius>	Specifies the X-coordinate position of the tool when CSS is turned on. Used with controls that require a current tool radius position when initiating Constant Surface Speed programming	numeric
<css-rpm-max>	Specifies the maximum RPM allowed when CSS is active **PartMaker dialog:** Process Parameters **Dialog field:** Max Speed	numeric
<cut-off-process>	Used to determine if a cut off process has been used **PartMaker dialog:** ToolMinder – Tools – Tool Properties **Check Box:** Use as Cutoff Tool **Note:** Also true if the current Cycle is set to Cutoff	logical
<cw-spindle>	Used to determine if the spindle is rotating in the clockwise direction **PartMaker dialog:** ToolMinder - Tools **Dialog field:** True if the “Speed Factor” is set as a positive value	logical
<cycle>	Specifies a cycle type identifier for canned cycles Typical values for <cycle> are G81, G82, G83, etc. <cycle> is assigned in ConfigPost’s Preparatory Functions dialog.	string

Word	Description	Type
<date>	Reproduces the date when the part program was post processed	string
<depth-abs>	Turn: Specifies the absolute Z-axis depth from the Z-axis origin for an Axial Hole **PartMaker dialog:** Part Features **-** Hole Group Parameters **<depth-abs> = [Nominal Depth(D) - Z_Surf(S)]**	numeric
<depth-cut>	Specifies the depth of cut that a tool will make in any single cut in roughing cycles (see <step2>) **PartMaker dialog:** Part Features - Profile Group Parameters for Cycles of Type = Turning, Contouring, and Threading **Dialog field:** Turning or Contouring =Depth of Cut (d) Threading = First Infeed (i)	numeric
<depth-inc>	Specifies the incremental Z-axis depth. **PartMaker dialog:** Part Features - Hole Group Parameters **Dialog field:** Nominal Depth(D)	numeric
<diam-clear>	**Obsolete** Specifies the clearance between the stock and the start point of a roughing tool path in X **PartMaker dialog:** Part Features – Profile Group Parameters **Dialog field:** Diam Clearance (Cd) **<diam-clear>** is obsolete, however, it is currently used with version 6.1 for Custom Macro B programming output instead of **<diam_clear_abs>** (see definition)	numeric
<drilling>	This word is true if any one of the following tools are used: **Drill; Spot Drill; Center; Chamfer; Tap; Ream; Bore**	logical
<dwell>	Reproduces the dwell time for a tool. **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Dwell (sec):	numeric
<eob>	Specifies the end of block code for each line of a part program. <eob> is assigned in ConfigPosts's General Information dialog	string

Word	Description	Type
<eq-number>	This word is used for comparing the value of a numeric word to a number. For example, if the total time of the processes comes out to 1.2 min. the statement below is used: <IF><eq-number>(<time-total>,1.2)<THEN> time total =1.2<eob> <ENDIF> **-the line "**time total=1.2" **will be read into the nc program**	logical
<eq-string>	This word is used for comparing a string word with a string. For example, if cutter compensation is set to the right, then the statement below is used: <IF><eq-string>(<comp-status>, G42)<THEN> comp is to the right <eob> <ENDIF> **-the line "**comp is to the right**" will be read into the nc program**	logical
<excess>	Specifies the amount of unfinished stock to be removed from the face of the material as part of the face machining. **PartMaker dialog:** View - Setup **Dialog field:** Excess Stock (E)	numeric
<face-change>	This word is true if there is a face change between two processes. For example, if the previous process was created on one Face Window and the current process is created on another Face Window, then <face-change> is true for the current process.	logical
<face-clear>	Specifies the clearance between the stock and the start point of a tool path in Z **PartMaker dialog:** Part Features - Group Parameters **Dialog field:** Face Clearance (Cf)	numeric
<face-name>	Specifies the name of the part face defined **PartMaker dialog:** View - Setup **Dialog field:** Rename To:	string
<face-note1>	Specifies the first "Note" line **PartMaker dialog:** View - Setup **Dialog field:** Notes	string
<face-note2>	Specifies the second "Note" line **PartMaker dialog:** View - Setup **Dialog field:** Notes	string

Word	Description	Type
<face-note3>	Specifies the third “Note” line **PartMaker dialog:** View - Setup **Dialog field:** Notes	string
<feed-units>	Inserts the G-code for feed rate units of measurement, either Units per Minute (UPM) or Units per Revolution (UPR) as set in ConfigPost Preparatory Functions, Turn dialog	string
<feed-upm>	Specifies the programmed feed rate in units per minute. (See<feed-units>) **PartMaker dialog:** Window – Process Table - Process Parameters **Dialog field:** Feed – Units Per Minute (UPM)	numeric
<feed-upr>	Specifies the programmed feed rate in units per revolution (See<feed-units>) **PartMaker dialog:** Window – Process Table - Process Parameters **Dialog field:** Feed – Units Per Revolution (UPR)	numeric
<first-tool-use>	This word is true if it’s the first time a tool is called in the Process Header Format for Tool Change <first-tool-use> is typically used to build a tool list at the beginning of an nc program.	logical
<gear-range>	Specifies the code necessary to place the lathe headstock in the correct gear for the current programmed spindle speed <gear-range> can be set in ConfigPost’s Preparatory Functions, Turn dialog. **PartMaker dialog:** Window -Process Table – Process Parameters **Dialog field:** Gear No:	string
<groove-width>	Specifies the width of a defined groove tool. **PartMaker dialog:** ToolMinder – Tools – Type: Groove **Dialog field:** Tip Size (g)	numeric
<group-name>	Specifies the name of the group used in a process as shown in the process table **PartMaker dialog:** Part Features - Profile Group Parameters or ToolMinder - Cycles **Dialog field:** Cycle Name or Rename To:	string
<header>	Used in Config Posts’s Procedures to check what current Header Format is active: Program Start = PGS Tool Change = TCH Process Start = PRS Process End = PRE Example: **<IF><eq-string>(<header>,TCH)<THEN>**	string

Word	Description	Type
<height>	Specifies the height of a thread as a number **PartMaker dialog:** Part Features – Group Parameters – Cycle Type for Threading Calculated as: **<height> = Thread Height (H) * Pitch (p)**	numeric
<id-tool>	**Obsolete** True if current tool is an id-tool **PartMaker Dialog:** Tool Data (accessed from ToolMinder menu) **Dialog Field:** TRUE if location is “In”	logical
<include>	Specifies the user prompt value that queries the user for a value to be input at time of post processing Example: {What is your name:<include>Bob} Would prompt” What is your name” and make Bob the default; the <include> value is then entered into the program file exactly as entered by the operator The value of <include> is used, then forgotten. See Chapter 3, “ConfigPost Advanced Features” for more information about <include> statements.	string
<infeed-angle>	Specifies the entry angle between passes during a threading cycle Typically this angle will be equal to the thread included angle **PartMaker dialog:** Part Features - Profile Group Parameters - Cycle type Threading **Dialog field:** Infeed Angle (a)	numeric
<in-path>	This word is true if the tool has performed first non-rapid move in the tool path. For example, when configuring Thread Move (under Motion Formats in ConfigPost) to output a G92 canned thread cycle where the full path of the thread is only necessary for the first thread pass, the configuration should look as shown below: **<IFNOT><in-path><THEN>** **G92 X<x-coord> z<z-coord> f<pitch><eob>** **<ENDIF>** **<IF><in-path><THEN>** **X<x-coord><eob>** **<ENDIF>**	logical
<int-depth-cut>	Specifies the depth of cut that a tool will make in any single cut in roughing cycles (identical to <depth-cut> but in a different integer format) **PartMaker dialog:** Part Features - Profile Group Parameters for Cycles of Type = Turning, Contouring, and Threading **Dialog field:** Turning or Contouring =Depth of Cut (d) Threading = First Infeed (i)	numeric

Word	Description	Type
<larger>	This word is true if the compared value is larger For example, if the total time of the processes is larger than 1.2 min. the statement below is used: <IF><larger>(<time-total>,1.2)<THEN> time total is greater then 1.2<eob> <ENDIF> **-the line “**time total is greater then 1.2” **will be read into the nc program**	logical
<lead-var>	Specifies the lead variation rate when programming variable lead threads **PartMaker dialog:** Part Features – Profile Group Parameters – Cycle Type for Threading - Thread Options **Dialog field:** Lead Variation Rate (used only for canned cycle output)	numeric
<metric-post>	Used to determine if the PartMaker Input setting is metric **PartMaker dialog:** View - Preferences **Radio button:** Metric	logical
<min-passes>	Specifies the minimum infeed for a threading cycle **PartMaker dialog:** Part Features - Profile Group Parameters - Cycle Type for Threading **Dialog field:** Minimal Infeed (j)	numeric
<mode-sub-sp-next>	Same as <mode-sub-sp>, but checks the mode status of the ***NEXT*** process	string
<motion>	Produces the correct motion type (Rapid, Linear, Arc CW/Arc-C-CW, for example) for the various program formats <motion> is assigned in the ConfigPost Preparatory Functions dialog	string
<num-mult>	Specifies the number of thread starts when programming multi-start threads **PartMaker dialog:** Part Features - Profile Group Parameters - Cycle Type for Threading – Options **Dialog field:** No. Threads for Multi-thread (used only for canned cycle output)	numeric
<num-passes>	Specifies the number of passes to take for finishing a thread; commonly referred to as “spring cuts” **PartMaker dialog:** Part Features - Profile Group Parameters - Cycle Type for Threading – Options **Dialog field:** Number of Spring Passes	numeric
<od-tool>	Used to determine if an external turning tool is being used **PartMaker dialog:** ToolMinder - Tools True if the Location is set to “Out” **Note:** For all other tool locations such as Face, In, and drilling type tools, <od-tool> would be false.	logical

Word	Description	Type
<part-length>	Specifies the defined length of the part **PartMaker dialog:** View - Setup **Dialog field:** Length (L)	numeric
<pitch>	Specifies the thread pitch, or 1/tpi **PartMaker dialog:** Part Features - Profile Group Parameters - Cycle Type for Threading **Dialog field:** Pitch (p)	numeric
<pre-shape-s>	Specifies the variable used to define the sequence number of one line of G-code before a contour shape is defined This value is typically used with OKUMA LAP cycles where the first line of the LAP cycle contour defines not the shape, but how the shape is to be cut.	numeric
<process-id>	Specifies the process number as it appears in the **PartMaker** process table **PartMaker Window:** Window - Process Table **Dialog field:** Proc ID	string
<prog-name>	Reproduces the user-assigned output file name **PartMaker Window:** Job Optimizer – Generate NC program **Dialog field:** Save NC Program File As:	string
<prog-num1>	Specifies the program number that is to appear at the beginning of the nc program (for dual processor controls, <prog-num1> specifies the program number for main spindle or turret #1 programs **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Program #1 (under Program No.)	string
<prompt>	Used similarly to <include>—except that the <prompt> value can be saved to a global variable. Example: {Enter a tool number <prompt> 1}<eob> ~<var-0><prompt><eob> <IF><eq-string>(<var-0>, 1)<THEN> tool-is-one<eob> <ENDIF> See Chapter 3, “ ConfigPost Advanced Features” for more information about <prompt> statements.	string
<radius>	Reproduces an arc radius in a circular block The value of <radius> will be negative for arcs with angles greater than or equal to 180 degrees.	numeric
<range-change>	Obsolete This word will be true if a gear range in the current process is different from the gear range in the previous process	logical
<ret-length>	Specifies the return length during a roughing cycle **PartMaker dialog:** Part Features - Profile Group Parameters **Dialog field:** Return Length (l)	numeric

Word	Description	Type
<seq>	Specifies the sequence number of the current line Used only in Turn and Mill **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Block start and Block Increment	numeric
<shape-end-q>	Specifies the sequence number associated with the last entity in a canned cycle defined shape	numeric
<shape-start-p>	Specifies the sequence number associated with the first entity in a canned cycle defined shape	numeric
<smaller>	This word is true if the compared value is smaller For example, if the total time of the processes is smaller than 1.2 min. the statement below is used: <IF><smaller>(<time-total>,1.2)<Then> time total is smaller then 1.2<eob> <ENDIF> **-the line “**time total is smaller then 1.2” **will be read into the nc program**	logical
<speed-fpm>	Specifies the spindle speed in constant surface speed mode, defined in Feet Per Minute (FPM) **PartMaker dialog:** Window – Process Table - Process Parameters **Dialog field:** Feet per Minute (FPM)	numeric
<speed-rpm>	Specifies the spindle speed in fixed mode, defined in Revolutions Per Minute (RPM) **PartMaker dialog:** Window – Process Table - Process Parameters **Dialog field:** Revolution per Minute (RPM)	numeric
<spindle-on>	Used to specify spindle rotation direction <spindle-on> is assigned in the ConfigPost Preparatory Functions dialog in both Mill and Turn sections. **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Speed Factor (If it’s a positive value, then <spindle-on> will output as **CW** direction. If it’s a negative value, then <spindle-on> will output as **CCW** direction.	string
<step1>	Specifies the tool step over in Z axis (pecking amount when performing deep hole drilling) **PartMaker dialog:** ToolMinder - Tools - for Drill and Groove tools **Dialog field:** Axial Step	numeric
<step2>	Specifies the depth of cut that a grooving tool will make in any single cut in roughing cycles (see <depth-cut>) **PartMaker dialog:** Part Features - Profile Group Parameters - Cycle Type for Grooving **Dialog field:** Depth of Cut (d)	numeric

Word	Description	Type
<stock-id>	Reproduces the inside stock diameter for tubing or a pre-drilled workpiece **PartMaker dialog:** View - Setup **Dialog field:** ID	numeric
<stock-od>	Reproduces the outside stock diameter of the workpiece **PartMaker dialog:** View - Setup **Dialog field:** OD	numeric
<surf>	Specifies a surface on the workpiece where all clearances and depths are measured. <surf> for z-oriented tools is commonly set to 0. <surf> for x-oriented tools is commonly set to the machined part radius. **PartMaker dialog:** Part Features - Hole Group Parameters **Dialog field:** X_Surf(S) or Z_Surf (S)	numeric
<taper-val>	Specifies the difference in the X-axis between the start and end of a thread <taper-val> is used in threading cycles and is for use with canned cycles only.	numeric
<thread-cycle>	Used to determine if a threading cycles is being used **PartMaker dialog:** Part Features – Profile Group Parameters – Cycle for Threading True if the Cycle is set to "Threading" **Note:** <thread-cycle> is NOT true for canned thread output	logical
<thread-final-x>	Specifies the X coordinate of the finished thread <thread-fin-x> is for use with canned cycles only.	numeric
<thread-final-z>	Specifies the Z coordinate of the finished thread <thread-fin-z> is for use with canned cycles only.	numeric
<time-process>	Specifies the time of the tool use from the tool change until the next tool change **PartMaker dialog:** Window - Process Table **Dialog field:** Time (column on Process Table)	numeric
<time-total>	Specifies the total cutting time of all processes combined **PartMaker Window:** Window - Process Table **Dialog field:** Total Time	numeric
<tool-broach>	**Obsolete** True if the current tool is defined as a broaching tool	logical
<tool-change-x>	Specifies the X coordinate for the tool change position **PartMaker dialog:** View - Setup **Dialog field**: Tool Change X (Xc)	numeric

Word	Description	Type
<tool-change-z>	Specifies the Z coordinate for the tool change position **PartMaker dialog:** View - Setup **Dialog field**: Tool Change Z (Zc)	numeric
<tool-id>	Reproduces the tool identification number assigned to each tool **PartMaker dialog:** ToolMinder- Tools **Dialog field**: Tool ID	numeric
<tool-index-x>	**Obsolete** Use **<tool-change-x>** instead.	numeric
<tool-nose-com>	Specifies the tool nose compensation value for a lathe, typically G40, G41 or G42. Recommended only to be used in Process Header program formats **PartMaker dialog:** Window – Process Table - Process Parameters **Dialog field:** Assigns value based on selection of Nose Radius Compensation of None, Left or Right	string
<tool-nose-rad>	Reproduces the nose radius of a turning tool. **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Nose Radius	numeric
<tool-note>	Specifies the note line from **PartMaker** Tools dialog **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Notes:	string
<tool-num>	Specifies the tool number **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Tool No:	numeric
<tool-num-next>	Represents the next tool number in the Process Table different from the current tool number. It may or may not be the tool of the next process. **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Tool No. (of next tool called from the process table)	numeric
<tool-offset>	Specifies the tool length/geometry offset number **PartMaker dialog:** ToolMinder - Tools	numeric
<tool-type>	Specifies the type of tool being used. <tool-type> can be equal to either of the following: **DR** (drill); **SD** (spot); **CH** (chamfer); **BO** (bore); **EM** (end mill); **CE** (center); **TA** (tap); **RE** (ream); **SM** (slot mill); **TM** (thread mill); **FM** (face mill); **DT** (dove tail); **CR** (corner round) **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Type:	string

Word	Description	Type
<upr-units-used>	Used to determine if the current process is using feedrate in units per rev **PartMaker dialog:** Window – Process Table – Process Parameters **Radio button:** True if the feed for “Units per Revolution (UPR)” is selected	logical
<work-offset>	Specifies the work offset code programmed for each Face Window **PartMaker dialog:** View - Setup **Dialog field:** Work Offset	string
<work-shift>	Reproduces the distance from the part face to the main spindle cap (absolute Z0) Used only in Turn and Turn-Mill **PartMaker dialog:** View - Setup **Dialog field:** Work Shift (I)	numeric
<x-coord>	Specifies the X-axis coordinate value with respect to the part origin	numeric
<x-finish-allow>	Specifies the X axis finishing amount for rough turning operations **PartMaker dialog:** Part Features – Profile Group Parameters - Cycle type for Turning and Contouring **Dialog field:** X Finish (Fx)	numeric
<z-face-location>	Specifies the Z coordinate of the face of the stock. **PartMaker dialog:** View –Preferences. **Dialog field**: Z Face (F).	numeric
<tool-index-z>	**Obsolete** Use **<tool-change-z>** instead.	numeric
<z-coord>	Specifies the Z-axis coordinate value with respect to the part origin	numeric
<z-finish-allow>	Specifies the Z axis finishing amount for rough turning operations **PartMaker dialog:** Part Features – Profile Group Parameters - Cycle type for Turning and Contouring **Dialog field:** Z Finish (Fz)	numeric

Reserved Words for Wire EDM

Word	Description	Type
<arc-x>	Used in a circular interpolation block to specify the signed X distance form the start point of an arc to the center of the arc along the X axis	numeric
<arc-y>	Used in a circular interpolation block to specify the signed Y distance form the start point of an arc to the center of the arc along the Y axis	numeric
<arm-stroke>	Specifies the maximum travel distance of the upper guide. **ConfigPost dialog:** General Information **Dialog field:** Upper Arm Stroke	numeric
<auto-wire-thread>	Used to determine if machine is equipped with an automatic wire threader **PartMaker dialog:** Job Optimizer – Post Options **Check box:** True if "Automatic Threader" is checked	logical
<center-x>	Reproduces the absolute X coordinate to an arc's center in a circular block	numeric
<center-y>	Reproduces the absolute Y coordinate position of an arc's center in a circular block	numeric
<comp-end>	This word is true if the move represents the end section for wire diameter compensation (last element or move of path) <comp-end> is used in linear move and XY,UV formats.	logical
<comp-middle>	This word is true if the move represents the middle section for Tool Nose Radius compensation (between the first and last moves of path) <comp-middle> is used in linear move formats.	logical
<comp-on>	This word is true if Tool Nose Radius compensation is on <comp-on> is used in process header formats.	logical
<comp-start>	This word is true if the move represents the start section for wire diameter compensation (first element or move of path) <comp-start> is used in linear move and XY,UV formats.	logical
<comp-status>	Establishes a right/left wire relationship with the part in order to compensate for the wire diameter Typical values are G40, G41, and G42, assigned in the Preparatory Functions dialog and used in Linear Move Program Format	string
<con-taper>	Used to determine if the type of taper is conical **PartMaker dialog:** Motion Record **Radio button:** Conical	logical
<curve-dist>	(Z) Distance between two profile curves in 4-axis machining.	numeric

Word	Description	Type
<cut-wire>	Used to determine if the wire should be cut at the end of the selected process. **PartMaker dialog:** Process Parameters **Check box:** Cut Wire	logical
<cyl-taper>	Used to determine if the type of taper is cylindrical **PartMaker dialog:** Motion Record **Radio button:** Cylindrical	logical
<date>	Reproduces the date when the part program was post processed	string
<eob>	Specifies the end of block code for each line of a part program. <eob> is assigned in ConfigPosts’s General Information dialog	string
<e-pack>	Reproduces the power settings macro # that is defined in the Cycles dialog. **PartMaker dialog:** ToolMinder - Cycles **Dialog field:** Power Setting	numeric
<eq-number>	This word is used for comparing the value of a numeric word to a number. For example, if the total time of the processes comes out to 1.2 min. the statement below is used: <IF><eq-number>(<time-total>,1.2)<THEN> time total =1.2<eob> <ENDIF> **-the line “**time total=1.2” **will be read into the nc program**	logical
<eq-string>	This word is used for comparing a string word with a string. For example, if cutter compensation is set to the right, then the statement below is used: <IF><eq-string>(<comp-status>, G42)<THEN> comp is to the right <eob> <ENDIF> **-the line “**comp is to the right**” will be read into the nc program**	logical
<face-change>	This word is true if there is a face change between two processes. For example, if the previous process was created on one Face Window and the current process is created on another Face Window, then <face-change> is true for the current process.	logical
<face-name>	Specifies the name of the part face defined **PartMaker dialog:** View - Setup **Dialog field:** Rename To:	string

Word	Description	Type
<face-note1>	Specifies the first "Note" line **PartMaker dialog:** View - Setup **Dialog field:** Notes	string
<face-note2>	Specifies the second "Note" line **PartMaker dialog:** View - Setup **Dialog field:** Notes	string
<face-note3>	Specifies the third "Note" line **PartMaker dialog:** View - Setup **Dialog field:** Notes	string
<feed-upm>	Specifies the programmed feed rate in units per minute. **PartMaker dialog:** Window – Process Table - Process Parameters **Dialog field:** Feed	numeric
<glue-stop>	This word is true if a Glue Stop has been set to a segment of a motion record. **PartMaker dialog:** Motion Record **Check box:** Glue Stop	logical
<group-name>	Specifies the name of the group used in a process as shown in the process table **PartMaker dialog:** Part Features - Profile Group Parameters or ToolMinder - Cycles **Dialog field:** Cycle Name or Rename To:	string
<guide-span>	(Z) Distance measured between the upper and lower guides **Config Post dialog:** General Information **Dialog field:** Guide Span	numeric
<include>	Specifies the user prompt value that queries the user for a value to be input at time of post processing Example: {What is your name:<include>Bob} Would prompt" What is your name" and make Bob the default; the <include> value is then entered into the program file exactly as entered by the operator The value of <include> is used, then forgotten. See Chapter 3, "ConfigPost Advanced Features" for more information about <include> statements.	string
<larger>	This word is true if the compared value is larger All: For example, if the total time of the processes is larger than 1.2 min. the statement below is used: <IF><larger>(<time-total>,1.2)<THEN> time total is greater then 1.2<eob> <ENDIF> **-the line "**time total is greater then 1.2" **will be read into the nc program**	logical

Word	Description	Type
<l-flow>	Controls Lower Flow for EDM **PartMaker dialog:** ToolMinder – Cycles **Dialog Field:** Lower Flow	numeric
<lin-taper>	Used to determine if the type of taper is linear **PartMaker dialog:** Motion Record **Radio button:** Linear	logical
<metric-post>	Used to determine if the PartMaker Input setting is metric **PartMaker dialog:** View - Preferences **Radio button:** Metric	logical
<offset-value>	Specifies the wire geometry offset number **PartMaker dialog:** ToolMinder - Cycles **Dialog field:** Offset	numeric
<oper-num>	Automatically assigns a sequential number to each operations in the Cycles dialog. Example: Roughing <oper-num> = 1 Skim 1 <oper-num> = 2 Skim 2 <oper-num> = 3	numeric
<part-height>	Reproduces the Part Height defined in the Setup dialog **PartMaker dialog:** View - Setup **Dialog field:** Part Height (H)	numeric
<process-id>	Specifies the process number as it appears in the **PartMaker** process table **PartMaker Window:** Window - Process Table **Dialog field:** Proc ID	string
<prog-name>	Reproduces the user-assigned output file name **PartMaker Window:** Job Optimizer – Generate NC program **Dialog field:** Save NC Program File As:	string
<prompt>	Used similarly to <include>—except that the <prompt> value can be saved to a global variable. Example: {Enter a tool number <prompt> 1}<eob> ~<var-0><prompt><eob> <IF><eq-string>(<var-0>, 1)<THEN> tool-is-one<eob> <ENDIF> See Chapter 3, “ConfigPost Advanced Features” for more information about <prompt> statements.	string
<radius>	Reproduces an arc radius in a circular block The value of <radius> will be negative for arcs with angles greater than or equal to 180 degrees.	numeric

Word	Description	Type
<remove-slug>	Used to determine if a slug should be removed at the end of the selected process (Usually represents an M00 program stop in code) **PartMaker dialog:** Process Parameters **Check box:** Remove Slug	logical
<seq>	Specifies the sequence number of the current line **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Block start and Block Increment	numeric
<smaller>	This word is true if the compared value is smaller For example, if the total time of the processes is smaller than 1.2 min. the statement below is used: <IF><smaller>(<time-total>,1.2)<Then> time total is smaller then 1.2<eob> <ENDIF> **-the line “**time total is smaller then 1.2” **will be read into the nc program**	logical
<taper-angle>	Reproduces the taper angle for 2-axis and taper cutting **PartMaker dialog:** Motion Record or ToolMinder - Cycles **Dialog field:** Angle (A) or Taper Angle	numeric
<taper-register>	Reproduces the register number for a motion record **PartMaker dialog:** Motion Record **Dialog field:** Register Number	numeric
<taper-used-in-proc>	This word is true if a taper has been programmed for the current process	logical
<thread-wire>	This word is true if: “Thread Wire on First Process” is checked in Post Options dialog It is NOT the first process and the wire was cut in the previous process	logical
<time-process>	Specifies the time of the tool use from the tool change until the next tool change **PartMaker dialog:** Window - Process Table **Dialog field:** Time (column on Process Table)	numeric
<time-total>	Specifies the total cutting time of all processes combined **PartMaker Window:** Window - Process Table **Dialog field:** Total Time	numeric
<tool-diam>	Specifies the wire diameter. **PartMaker dialog:** View - Setup **Dialog field:** Wire	numeric
<tool-name>	Specifies the name of the wire **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Rename To:	string

Word	Description	Type
<tool-note>	Specifies the note line from **PartMaker** Tools dialog **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Notes:	string
<u-coord>	Reproduces the X coordinate of a point on the top part surface during 4-axis machining	numeric
<u-flow>	Controls Upper Flow for EDM **PartMaker dialog:** ToolMinder – Cycles **Dialog Field:** Upper Flow	numeric
<u-relat>	Reproduces the X coordinate of a point on the top part surface relative to a X coordinate of a corresponding point on the bottom surface during 4-axis machining	numeric
<v-coord>	Reproduces the Y coordinate of a point on the bottom part surface during 4-axis machining	numeric
<v-relat>	Reproduces the Y coordinate of a point on the top part surface relative to a Y coordinate of a corresponding point on the bottom surface during 4-axis machining	numeric
<work-offset>	Specifies the work offset code programmed for each Face Window **PartMaker dialog:** View - Setup **Dialog field:** Work Offset	string
<x-coord>	Specifies the X-axis coordinate value with respect to the part origin	numeric
<y-coord>	Specifies the Y-axis coordinate value with respect to the part origin	numeric
<Z-lower-guide>	Reproduces the Z coordinate of the lower guide. **PartMaker dialog:** View - Setup **Dialog field:** Lower Guide (L)	numeric
<Z-upper-guide>	Reproduces the Z coordinate of the upper guide. **PartMaker dialog:** View - Setup **Dialog field:** Upper Guide (U)	numeric
<Z-wire-tilt>	Reproduces the Z coordinate where the wire is tilted when programming taper in the current process	numeric

Reserved Words for Advanced ConfigPost

The following table contains reserved words used for TurnMill and SwissCAM Applications.

Word	Description	Type
<air-blast>	Used to determine if air blast for sub spindle is equipped and activated. **PartMaker dialog:** Job Optimizer – Post Options **Check Box:** Air Blast (under Machine Options)	logical
<b-angle>	**For PartMaker Version 6.3 and above** Used for 5-axis machining on an arbitrary plane. Reproduces the angle between the vector normal to this plane and the Z axis.	numeric
<bar-load>	Reproduces the bar change sub program number that is usually called at then end of a program **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Bar Load (under Program No.)	string
<b-axis-avail>	Used to determine if a tool post (turret) is equipped with a 180 degree articulating axis (B-axis) **PartMaker dialog:** Job Optimizer – Post Options **Check Box:** Turret B-axis (under Axis Support)	logical
<c-angle>	Specifies the C-axis angle for machining functions: Mill End Index; Mill Diam. Index; Mill Polygon; Mill End Polar; Mill Cylinder; Mill Diam. Polar; Mill XY; Mill ZY; Mill ZX	numeric
<catcher-num>	Reproduces the station number for ejecting or gripping the part **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Station No. (under Part Release Data)	string
<c-axis-main>	Used to determine if the main spindle is equipped with a continuous C-axis **PartMaker dialog:** Job Optimizer – Post Options **Check box:** Main Spindle C-axis	logical
<c-axis-sub>	Used to determine if the sub spindle is equipped with a continuous C-axis **PartMaker dialog:** Job Optimizer – Post Options **Check box:** Sub Spindle C-axis	logical
<channel-number>	Assigns a <tool-post> to a channel in order to generate multiple output files Defined in Config Post under Edit – Channels (Moving and Stationary Stock)	numeric
<collet-nose>	Reproduces the Collet Nose Extension on the Sub Spindle. Used in SwissCAM only. **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Sub Spindle Collet Nose Extension	numeric

Word	Description	Type
<turret-code>	**Obsolete** This word is not used for what intended originally. However it was used temporarily in version 6.1 for Macro-B output in Cit-M Post as an indicator of CUSTOM THREAD requiring WHILE loop.	string
<count>	**Obsolete** Internal counter that starts at the block start value and increments by the block increment value **PartMaker Dialog:** Post Options (accessed from Job Optimizer menu) **Dialog Field:** Block Start, Block Increment	numeric
<c-radius>	Specifies the C-axis radius (measured from the center of the cylinder to the programmed hole or profile location) Primarily used when drilling holes on the face of the part	numeric
<customer-pin>	Reproduces the 7-digit Customer PIN **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Customer PIN	string
<cut-off-detect>	Used to determine if machine is equipped with a cut off detection device. **PartMaker dialog:** Job Optimizer – Post Options **Check Box:** Cut-off Detection (under Machine Options)	logical
<cut-off-width>	Specifies the width of the insert in the designated cutoff tool **PartMaker dialog:** ToolMinder - Tools – Type: Groove – Tool Properties – Use as Cutoff-tool **Dialog field:** Tip Size (g)	numeric
<cut-off-x-last>	The last X coordinate of the cut-off process **PartMaker dialog:** Profile Group Parameters - Cutoff Cycle **Dialog field:** End X Point (Xe)	numeric
<cyl-rad>	Specifies the radius of the defined cylinder for turning stock. **PartMaker dialog:** View - Setup **Dialog field:** OD (<cyl-rad> is calculated as half the OD)	numeric
<diam-clear-abs>	**For PartMaker Version 6.3 and above** Value of the Z-coordinate corresponding to the tool clearance level for Thread Operation for Macro-B output.	numeric

Word	Description	Type
<diam-sign>	Used to define a specific tool that must be programmed with negative diameters when other tools must be programmed with positive diameters The value of <diam-sign> can be 1 or –1. **PartMaker dialog:** ToolMinder - Tools – Tool Properties **Check Box:** Negative Diameter (when the Negative Diameter check box is on, <diam sign> has a value of –1)	numeric
<eject-part>	Used to determine if part ejection is to occur for the selected process **PartMaker dialog:** Window- Process Table – Process Parameters or Set Modes Check box: **Eject Part (E)**	logical
<face-dia-polar>	Used to determine if the current process is a Diameter Polar function **PartMaker dialog:** View - Setup **Dialog field:** True if “Mill Diam, Polar” Machining Function is selected.	logical
<face-end-index>	Used to determine if the current process is a End Index function **PartMaker dialog:** View - Setup **Dialog field:** True if “Mill End, Index” Machining Function is selected.	logical
<face-end-polar>	Used to determine if the current process is a End Polar function **PartMaker dialog:** View - Setup **Dialog field:** True if “Mill End, Polar” Machining Function is selected.	logical
<face-type>	Specifies the type of face currently active <face-type> can have the following values: face-turn face-xy face-zy face-polygon face-end-index face-dia-index face-end-polar face-dia-polar face-cyl face-zx **PartMaker dialog:** View - Setup **Drop-down menu:** Machining Function	string

Word	Description	Type
<face-zx>	Used to determine if the current process is a ZX function **PartMaker dialog:** View - Setup **Dialog field:** True if “Mill ZX” Machining Function is selected.	logical
<face-zy>	Used to determine if the current process is a ZY function **PartMaker dialog:** View - Setup **Dialog field:** True if “Mill ZY” Machining Function is selected.	logical
<guide-bush-dist>	Specifies the distance from the left side of the cutoff tool insert to the guide bushing face Z=0 on a swiss screw machine. Used in SwissCAM only. If the Tool Orientation for the cutoff tool is set to Right, then the **<guide-bush-dist> = 0** If the Tool Orientation for the cutoff tool is set to Left, then the **<guide-bush-dist> = Width(w) – Tip Size (g)**	numeric
<I-angle>	Specifies the C-angle orientation of stock <i-angle> has the same value as <c-angle>, but may be formatted differently	numeric
<in-support-mode>	This word is true if the current process is either set to sub spindle Mode - SF, SN, or SP	logical
<is-5-axis>	**For PartMaker Version 6.3 and above** Used for 5-axis machining on an arbitrary plane. It is set to TRUE if the current process is a 5-axis process, otherwise set to FALSE.	logical
<last-main-proc>	This word is true if the current process is the last main spindle process of the Process Table	logical
<last-proc>	This word is true if the current process is the overall last process of the Process Table	logical
<last-sub-proc>	This word is true if the current process is the last sub spindle process of the Process Table	logical
<l-coord>	Specifies the converted angular value of the tool position into linear units for cylinder milling Used for some machine controls that require linear values for both axes when performing cylindrical machining.	numeric
<local-origin-x>	**For PartMaker Version 6.3 and above** Used for 5-axis machining on an arbitrary plane. Reproduces a “world” X coordinate of a point used as a local origin for this plane in a PartMaker Window.	numeric
<local-origin-y>	**For PartMaker Version 6.3 and above** Used for 5-axis machining on an arbitrary plane. Reproduces a “world” Y coordinate of a point used as a local origin for this plane in a PartMaker Window.	numeric

Word	Description	Type
<local-origin-z>	**For PartMaker Version 6.3 and above** Used for 5-axis machining on an arbitrary plane. Reproduces a "world" Z coordinate of a point used as a local origin for this plane in a PartMaker Window.	numeric
<M2-move-tool>	Used to determine if the secondary tool of an M2 sync group has a programmable Z-axis. Used only for Swiss applications. **PartMaker dialog:** Window – Process Table – Set Modes (for secondary tool in an M2 sync group) – Stock Motion **Check box:** Move Tool in Z	logical
<machine-ID>	For PartMaker internal use only	string
<mat-control-process>	Used to determine if the current process is a Material Control Process **PartMaker dialog:** Job Optimizer – Insert Material Control Process ⇒ True if any one of the MCP functions are selected as a process	logical
<mcp-function>	Specifies the type of Material Control Process currently active Used only for Swiss and Turn-Mill applications. <mcp-function> can have the following values: **BF** ---------- Bar Feed **PSG** -------- Pull with Side Gripper **PFG** -------- Pull with front Gripper **TF** ---------- Tailstock Forward **TR** ---------- Tailstock Retract **RMS** -------- Remove from Main Spindle **RSS** --------- Remove from Sub Spindle **PSS** --------- Pull with Sub Spindle (Rechuck for Swiss application) **TPM** -------- Transfer Part from Main to Sub Spindle **NON**--------- No MCP function – regular cutting process Used in the following context: **<IF><eq-string>(<mcp-function>,BF)<THEN>** - Example above checks to see if the current MCP function is a Bar Feed	string
<mcp-pull-dist>	Specifies the pull distance entered in a Material Control process Used in Turn and Turn-Mill only **PartMaker dialog:** Job Optimizer – Insert Material Control Process – Used in Functions "Pull with Side Gripper", "Pull with front Gripper", and "Pull with Sub Spindle" **Dialog field:** Pull Distance (Zp)	numeric

Word	Description	Type
<mcp-total-pull>	Updates the pull distance for every process. For example, when only one MCP function -“Pull with Sub Spindle” is used, then **<mcp-total-pull> = <mcp-pull-dist>** but, if more MCP functions -“Pull with Sub Spindle” are added, then **<mcp-total-pull> = total of all <mcp-pull-dist>**	numeric
<mode>	Specifies whether a process has been set to either a non-synchronous, synchronous, or support state. <MODE> can be equal to any of the following – **M0S1**; **M1S1**, **M1S0**; **M2S0**; **M2S1**; **M1S2**; **M0S2**; **M1SD** (swiss only); **M2SD** (swiss only); **M1SF** (swiss only); **M1SN** (swiss only); **M1SP** (turn-mill only); **M2SP** (turn-mill only); **M2SF** (swiss only); **M2SN** (swiss only) **PartMaker dialog:** Window – Process Table – Set Modes Set radio buttons to desired Mode for both main spindle and sub spindle operations	string
<mode-main-sp>	Specifies whether a main spindle process has been set to either a non-synchronous or synchronous state. <mode-main-sp> can be equal to any of the following – **M0**; **M1**; **M2** **PartMaker dialog:** Window – Process Table – Set Modes Set radio button to desired Mode for main spindle operations	string
<mode-main-sp-next>	Same as <mode-main-sp>, but checks the mode status of the ***NEXT*** process	string
<mode-next>	Same as <mode>, but checks the mode status of the ***NEXT*** process	string
<mode-sub-sp>	Specifies whether a sub spindle process has been set to either a non-synchronous, synchronous or support state. <mode-sub-sp> can be equal to any of the following – **S0**; **S1**; **S2**; **SF** (swiss only); **SN** (swiss only); **SP** (turn-mill only); **SD** **PartMaker dialog:** Window – Process Table – Set Modes Set radio button to desired Mode for sub spindle operations	string
<mode-sub-sp-next>	Same as <mode-sub-sp>, but checks the mode status of the ***NEXT*** process	string
<move-stock>	Used to determine if the stock on the main spindle is feeding in the Z-axis during machining. Used only for Swiss applications. **PartMaker dialog:** Window – Process Table – Set Modes - Stock Motion **Check Box:** Move Stock in Z	logical

Word	Description	Type
<new-face-type>	This word is similar to <face-change> where it checks to see if there is a face change between two processes. The only difference being that <new-face-type> checks if the NEXT process IS NOT the same machining function as the current one. If they are not the same, this word is then true.	logical
<new-mode>	This word is true if the Mode in the next process is different from the Mode in the current process	logical
<new-spindle>	This word determines if the next process is on a different spindle. <new-spindle> is true in Process End if: The current process is on the main spindle and the next process is on the sub spindle The current process is on the sub spindle and the next process is on the main spindle	logical
<new-spindle-type>	This word determines if the next machining fork is different from the current machining fork. For example, <new-spindle-type> is true in Process End if: The current process is turning on the main spindle and the next process is turning on the sub spindle or The current process is turning on the main spindle and the next process is milling on the main spindle.	logical
<new-sync-group>	This word is true if the <sync-group> value of the next process is different from the <sync-group> value of the current process	logical
<new-tool>	This word is true if the <tool-num> ***or*** <tool-offset> of the next process is different from the <tool-num> ***or*** <tool-offset> of the current process	logical
<new-tool-on-channel>	This word is true if the next tool number or tool offset on the same channel is different from the current tool number or tool offset. For example, if Turret #1 = Channel #1 and Turret #2 = Channel #2 and the Process Table has the following order: Proc 1 Tool #1 Offset #21 Turret #1 Proc 2 Tool #3 Offset #23 Turret #2 Proc 3 Tool #1 Offset #21 Turret #1 At the end of Process #1, <new-tool-on-channel> would be false since Process #2 is using a different channel and Process #3 ha s the same tool number and tool offset as the current process.	logical
<new-tool-post>	This word is true if the <tool-post> of the next process is different from the <tool-post> of the current process	logical

Word	Description	Type
<next-tl-num-on-chnl>	Specifies the next tool number on the same channel. For example, if Turret #1 = Channel #1 and Turret #2 = Channel #2 and the Process Table has the following order: Proc 1 Tool #1 Turret #1 Proc 2 Tool #3 Turret #2 Proc 3 Tool #7 Turret #1 At the end of Process #1, <next-tl-num-on-chnl> would be equal to "7" since Process #2 is using a different channel.	numeric
<non-zero-shifts>	Used to determine if at least one tool shift has been programmed **PartMaker dialog:** ToolMinder - Tools Data – Tool Properties True if Tool Shifts for either X, Y, or Z are all not equal to 0	logical
<norm-x>	Specifies the X component of the unit vector perpendicular to machined surface; used in multi-axis verification	numeric
<norm-y>	Specifies the Y component of the unit vector perpendicular to machined surface; used in multi-axis verification	numeric
<norm-z>	Specifies the Z component of the unit vector perpendicular to machined surface; used in multi-axis verification	numeric
<phase-angle>	Reproduces the angle used during main spindle and sub spindle phase synchronization. **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Phase Angle (under Job Settings)	string
<pinch-turning>	Used to determine if current process is set as a pinch turn operation **PartMaker dialog:** Part Features – Profile Group Parameters – Cycle for either Contouring or Turn **Check box:** Pinch Turning	logical
<prev-turning>	**Obsolete** True if the previous process was a turning process Used to determine if a turn-mill needs to be switched from milling mode to turning mode	logical
<primary-tool>	Used to determine the leading tool used in Mode - M2S* This word is true for the first process in a M2S* group	logical
<prog-num2>	Specifies the program number for sub spindle or turret #2 programs **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Program #2 (under Program No.)	string

Word	Description	Type
<release-x>	Reproduces the X-coordinate to position the device holding the workpiece (turret or spindle) before part ejection occurs. **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Release-X	numeric
<release-z>	Reproduces the Z-coordinate to position the device holding the workpiece (turret or spindle) before part ejection occurs. **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Release-Z	numeric
<speed-max-main>	Reproduces the maximum spindle speed for the main spindle **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Max RPM Main Spindle	numeric
<speed-max-sub>	Reproduces the maximum spindle speed for the sub spindle **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Max RPM Sub Spindle	numeric
<speed-min-main>	Reproduces the minimum spindle speed for the main spindle **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Min RPM Main Spindle	numeric
<speed-min-sub>	Reproduces the minimum spindle speed for the sub spindle **PartMaker dialog:** Job Optimizer - Post Options **Dialog field:** Min RPM Sub Spindle	numeric
<spindle-off>	Used in Process End Format to stop the spindle **ConfigPost dialog:** Faces Info (Settings Menu). **Dialog field:** Spindle Stop	string
<spindle-type>	Used in Config Post to check what current machining fork is active: Turning on the main spindle = MT Milling on the main spindle = MM Turning on the sub spindle = ST Milling on the sub spindle = SM Example: <IF><eq-string>(<spindle-type>,st)<THEN>	string
<stock-z-coord>	Specifies the Z coordinate value of the tool with respect to the face of the stock that has been pushed out of the guide bushing Used in SwissCAM only **<stock-z-coord> = <stock-z-pos> +<z-coord>**	numeric

Word	Description	Type
<stock-z-pos>	Reproduces a Z coordinate for when the stock is positioned away from the guide bushing Z=0 before each machining operation begins. (defaults to a value of 0) Used in SwissCAM only **PartMaker dialog:** Window – Process Table – Set Modes - Stock Motion **Dialog field:** Tool Position (Tz)	numeric
<sub-center-z>	**Obsolete** Reproduces the absolute Z coordinate of an arc's center in a circular block in the sub-spindle of a twin-spindle lathe. See <center-z>	numeric
<sub-depth-abs>	**Obsolete** Absolute Z-axis depth from the Z-axis origin for an Axial Hole created on the sub-spindle. Used in Turn-Mill applications where the Z-axis positive direction is the same on the sub-spindle as the main spindle **<sub-depth-abs> = <surf> + <depth>** See <depth-abs> for more information	numeric
<sub-feed>	Reproduces the sub spindle feed (upm) while moving on to the part to clamp and support or transfer. **PartMaker dialog:** Job Optimizer – Post options **Dialog field:** Sub Spindle Feed on to Part(UPM)	numeric
<sub-step-abs>	**Obsolete** Z-axis step of a hole created in a sub-spindle window. Used in Turn-Mill applications where the Z-axis positive direction is the same on the sub-spindle as the main spindle **<sub-step-abs> = <surf>+<step>** See <step-abs> for more information	numeric
<sub-support-z>	Reproduces the sub spindle part support Z coordinate while machining is performed **PartMaker dialog:** Window – Process Table – Set Modes – activate any mode using the word "support" **Dialog field:** Sub Spindle Support Z-Coordinate (SZ)	numeric
<sub-wait-z>	For machining in SF Mode, the sub spindle can be queued to start support with a Z coordinate that falls in the path of the tool that is cutting. Used only in Swiss **PartMaker dialog:** Window – Process Table – Set Modes – Wait Distances **Dialog field:** Sub Spindle Wait Z **Note:** For M2SF Mode, Wait Distances will ONLY be active for the first process of the sync group	numeric

Word	Description	Type
<sub-zclear-abs>	**Obsolete** Z clearance absolute value for a process created in a sub spindle window. Used in Turn-Mill applications where the Z-axis positive direction is the same on the sub-spindle as the main spindle **<sub-zclear-abs> = <surf> - <clear>** See <clear-abs> for more information	numeric
<sub-zcoord>	Specifies the Z coordinate of the tool position in a sub spindle window <sub-z-coord> is used in Turn-Mill and Swiss applications where the Z-axis positive direction is the same on the sub spindle as on the main spindle. **<sub-z-coorD> = [(-1.0)*<z-coord>]**	numeric
<sub-zrapid-abs>	**Obsolete** Z rapid absolute value of a tool in a sub spindle window. Used in Turn-Mill applications where the Z-axis positive direction is the same on the sub-spindle as the main spindle **<sub-zrapid-abs> = <surf> - <rapid>** See <rapid-abs> for more information	numeric
<sync-end>	**Obsolete** This word is true for the last process of any sync group in the Process Table.	logical
<sync-first-sub>	**Obsolete** This word is true if the first process of the current sync group in the Process Table is a sub spindle process	logical
<sync-start>	**Obsolete** This word is true for the first process of any sync group in the Process Table	logical
<sync-turret-on-main>	**Obsolete** This word is true if a **"Turret"** Tool Post is used in at least one main process in a given sync group	logical
<sync-turret-on-sub>	**Obsolete** This word is true if a **"Turret"** Tool Post is used in at least one sub spindle process in a given sync group	logical
<sync-group>	Specifies the Sync group numeric value derived from the Mode settings in the Process Table **PartMaker dialog:** Window - Process Table **Dialog field:** Sync Group (button)	string
<sync-used>	**Obsolete** This word is true if there is a current process is a member of a Sync Group.	logical

Word	Description	Type
<tool-angle>	Represents the inclination angle of the tool **PartMaker dialog:** Part Features - Hole Group Parameters (ONLY used in XY Machining Function) **Dialog field:** Inclination Angle	numeric
<tool-post>	Specifies the Tool Post being used for the selected tool <tool-post> can be equal to either of the following: **Turr**; **Turr1**; **Turr2**; **Turr3**; **Gang**; **Gang2**; **End**; **Back** **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Tool Post	string
<tool-post-next>	Represents the next Tool Post in the Process Table different from the current Tool Post. It may or may not be the Tool Post of the next process. <tool-post> can be equal to either of the following: **Turr**; **Turr1**; **Turr2**; **Turr3**; **Gang**; **Gang2**; **End**; **Back** **PartMaker dialog:** ToolMinder - Tools **Dialog field:** Tool Post	string
<tool-shift-x>	Specifies the tool shift value in the X-axis; commonly used to program “tool shifts” on Swiss-type lathes **PartMaker dialog:** ToolMinder - Tools - Tool Properties **Dialog field:** X	numeric
<tool-shift-y>	Specifies the tool shift value in the Y-axis; commonly used to program “tool shifts” on Swiss-type lathes **PartMaker dialog:** ToolMinder - Tools- Tool Properties **Dialog field:** Y	numeric
<tool-shift-z>	Specifies the tool shift value in the Z-axis; commonly used to program “tool shifts” on Swiss-type lathes **PartMaker dialog:** ToolMinder - Tools - Tool Properties **Dialog field:** Z	numeric
<tool-wait-z>	For machining in M2 Mode, the secondary tool can be queued to engage machining with a Z coordinate that falls in the path of the primary tool that is already cutting. Used only in Swiss **PartMaker dialog:** Window – Process Table – Set Modes – Wait Distances **Dialog field:** Tool Wait Z	numeric
<turret-1>	Used to determine if the current tool is assigned to Turret **PartMaker dialog:** ToolMinder - Tools True if Tool Post is set to “Turret”	logical

Word	Description	Type
<turret-2>	Used to determine if the current tool is assigned to Turret #2 **PartMaker dialog:** ToolMinder - Tools True if Tool Post is set to “Turret#2”	logical
<turret-3>	Used to determine if the current tool is assigned to Turret #3 **PartMaker dialog:** ToolMinder - Tools True if Tool Post is set to “Turret#3”	logical
<using-basket>	Used to determine if a parts catcher basket (mounted on a turret) is to be used during part ejection **PartMaker dialog:** Job Optimizer – Post Options **Check Box:** Parts Catcher Basket (under Part Release Data)	logical
<version>	For PartMaker internal use only	numeric
<wait-code>	Reproduces the wait code start for dual spindle lathes **PartMaker dialog:** Job Optimizer – Post Options **Dialog field:** Start (under Wait/Queue Commands)	string
<wait-code-inc>	Reproduces the wait code increment for dual spindle lathes **PartMaker dialog:** Job Optimizer – Post Options **Dialog field:** Increment (under Wait/Queue Commands)	string
<while-loop>	**For PartMaker Version 6.3 and above** True if a while loop has to be generated while programming Thread Operation for Macro-B output.	logical
<xi-coord>	Specifies the X-axis coordinate value with respect to the part origin <xi-coord> has the same value as <x-coord>, but may be formatted differently as needed.	numeric
<yi-coord>	Specifies the Y-axis coordinate value with respect to the part origin <yi-coord> has the same value as <y-coord>, but may be formatted differently as needed.	numeric
<zi-coord>	Specifies the Z-axis coordinate value with respect to the part origin <zi-coord> has the same value as <z-coord>, but may be formatted differently as needed.	numeric

Program Format Structure

Below is an example of a program format that represents a Linear Move:

```
{<motion>}{X <x-coord>} {Y <y-coord>} {Z <z-coord>} {F <feed-upm>} <eob>
```

Each program format consists of:

- General text such as man-readable comments and letter address characters e.g. X and Y
- Reserved words such as <x-coord> and <y-coord>
- Action Symbols ; @ $ ~ & # [ ] < > ( ) { } that cause a designated action as described below.

Usage of Action Symbols

Action Symbol	Usages
;	When used in the beginning of a line of text causes the text to be interpreted as an explanation line
@	When used in front of a Numeric Reserved Word it causes Incremental Output for this word **Example:** <@x-coord> is the difference between the current and the previous value of the X coordinate.
$	When used in front of a Reserved Word it causes the previous value of this word to be used. **Example:** <$x-coord> is the previous value of the X coordinate.
#	When used in front of a File Directive, namely **define**, **split**, **merge**, **remove**, it causes this File Directive to apply in the case of Multiple Output Files.
~	When used in the beginning of a line followed by a Reserved Word it causes this word to be assigned a value. **Example:** ~<z-coord><clear-abs> <eob> will result in the Z coordinate to be assigned a value equal to the clearance plane location.
[ ]	When used around Reserved Word representing numeric expressions causes such expressions to be evaluated. **Example:** [2*<x-coord>] is the double value of the current X coordinate.
< >	**Use 1**: When used around text it represents a Reserved Word. **Example:** <y-coord> **Use 2:** When used around a number it represents a character corresponding to this ASCII number. **Example:** <91> represents “[“.

Action Symbol	Usages
( )	**Use 1**: When used within [ ] it establishes a sequence of expressions evaluation. **Example:** [2*(<x-coord> +5)]. **Note**: It is recommended to always use ( ) around negative numbers, e.g. [(-1)*<x-coord>] **Use 2:** When used following a Logical Reserved Word such as <eq-number>, <eq-string>, <larger>, <smaller> it signals to compare expressions within. **Example:** <if><eq-number>(<x-coord>, <y-coord>)<then> checks that the current X coordinate is the same as the current Y coordinate.
{ }	When used around a Numeric Reserved Word or around a String Reserved Word it signals that it is a Modal Reserved Word **Example:** {X<x-coord>} will prevent the output of the letter X and the current X coordinate is the same as the previous X-coordinate.

Note: *If you want characters corresponding to the Action Symbols to appear in the NC File instead of causing a designated action you must use ASCII numbers to represent these characters.*

Example: *If you want to put into NC File an expression [a + b] you should enter it as <91>a+b<93>.*

Rules for Program Start, Tool Change, and Process Start

With Z-Oriented Tools (Mill)

The information in this section provides the rules or cases that apply to specific Process Header formats, namely, **Program Start**, **Tool Change** and **Process Start**.

The following rules apply to program formats you define in a Post file:

1. **Positioning**: The reserved words, <x-coord> and <y-coord> are assigned the values of the first path point and must appear in all Process Header formats to provide the first positioning move to the start of the path.
2. The reserved word **<z-coord>** must be used for the **Process Start** program format. **<z-coord>** is calculated as the largest RAPID value of the current and previous process, and allows the tool to be retracted from the part to a safe plane.
3. Using the **<z-coord>** reserved word is optional for Program Start and Tool Change program formats.

   <z-coord> is calculated in the following manner:

   <z-coord> = Tool Change (Z)

Note: *The Z coordinate Tool Change (Z) is defined in PartMaker Defaults dialog which can be accessed in PartMaker from the Job Optimizer menu.*

Whether **<z-coord>** is used or not, the post processor assumes that the tool is at this Z level after the code for any of the previously mentioned formats is generated.

Important! *The Tool Change format is output only if there is a change in tool number between processes. If there is not a change in tool number, the Process Start format is output.*

With X-Oriented Tools (Mill)

The information in this section provides the rules or cases that apply to specific Process Header formats, namely, Program Start, Tool Change and Process Start.

The following rules apply to program formats you define in a Post file.

1. Positioning should be as follows:

   YZ-Plane

   The reserved words <y-coord> and <z-coord> are assigned the values of the first path point and must appear in all Process Header formats to provide the first positioning move to the start of the path.

   ZC-Plane

   The reserved words <z-coord> and <c-angle> are assigned the values of the first path point and must appear in all process header formats to provide the first positioning move to the start of the path.
2. The reserved word <x-coord> must be used for the Process Start program format.

   <x-coord> is calculated as the largest RAPID value of the current and previous process, and allows the tool to be retracted from the part to a safe plane.

3 Using the <x-coord**>** reserved word is optional for the Program Start and Tool Change program formats.

<x-coord> is calculated in the following manner:

<x-coord> = Tool Change (X)

Note: *The X coordinate Tool Change (X) is defined in PartMaker Defaults dialog which is accessed in PartMaker's Job Optimizer menu.*

Whether <x-coord> is used or not, the post processor assumes that the tool is at this X level after the code for any of the previously mentioned formats is generated.

Important! *The Tool Change format is output only if there is a tool change between processes. If there is not a change in tool number, the Process Start format is output.*

Turn

The information in this section provides the rules or cases that apply to Program Header formats for Turning.
The following rules apply to program formats you define in a Post file.

The reserved words, <x-coord> and <z-coord> are assigned the values of the first path point and must appear in all program formats to provide the first positioning move to the start of the path.

Important! *The Tool Change format is output only if there is a change in tool number between processes. If there is not a change in tool number, the Process Start format is output.*

WireEDM

The information in this section provides the rules or cases that apply to Program Header formats for EDM.

The following rules apply to program formats you define in a Post file.

The reserved words, <x-coord> and <y-coord> are assigned the values of the first path point and must appear in all program formats to provide the first positioning move to the start of the path.

Chapter 3: ConfigPost Advanced Features

Assigning Values to Reserved Words

Reserved words' values are assigned when PartMaker generates an NC program, i.e., reserved words in the Post file are substituted by their corresponding values and machine tool codes (called G and M codes) from the Process Table. ConfigPost also allows you to assign values to reserved words within the Post file itself. Reserved words can be assigned values in any of ConfigPost's program formats as shown below:

```
~<reserved word being assigned>VALUE BEING ASSIGNED<eob>
```

The example below shows how to set the reserved word <tool-num> to 10:

```
~<tool-num>10<eob>
```

Another example shows how to assign a value of one reserved word to another:

```
~<tool-num><tool-offset><eob>
```

Note: *To help clarify the assignment, you can insert a comment:*

```
~Comment <reserved word being assigned>VALUE BEING ASSIGNED<eob>
```

When inserting a comment with a variable assignment, it is often helpful to put an "=" sign before the comment to indicate to yourself that you are assigning a variable. For example, you might insert a comment like this:

```
~=Set tool number to 10 <tool-num>10<eob>
```

Explanation Lines

ConfigPost allows to put explanations into any program format. An explanation line has to start with **;** (semicolon) and it has to end with <eob>.

During Post Processing PartMaker ignores explanations if the check box "Strip Explanation Lines" is checked in the General Information,Mill or General Information,Turn or General Information,WireEDM Dialog.

If "Strip Explanation Lines" is not checked then during Post Processing PartMaker puts all explanations lines into a G-Code file.

Using ConfigPost's File Handling Features

PartMaker has the ability to create more than one output file and/or combine output files. This function is very useful when working with dual path controls and/or creating tool lists, for example.

When you want to split a file, the first four lines of the post processor must be as follows:

```
#define MACHINE NAME<eob>
#split<prog-name>.TXT FILE1.EXT FILE2.EXT OUT_NAME.TXT<eob>
#merge MERGEFILE.EXT FILE1.EXT FILE2.EXT<eob>
#remove FILE1.EXT FILE2.EXT<eob>
```

The four lines shown above must be placed in the File Start section of your Post file.

The MACHINE NAME can be any name that you choose.

Note: *At this time, MACHINE NAME is not used.*

The **#split** command instructs PartMaker to take the initial output file <PROG-NAME>.TXT and split it into as many files as are defined by the arguments that follow.

Unlike basic PartMaker applications, this <PROG-NAME>.txt will not be the final G-code program that you send to the CNC.<PROG-NAME>.txt will be split into the number of additional files you have designated, as shown in the following illustration.

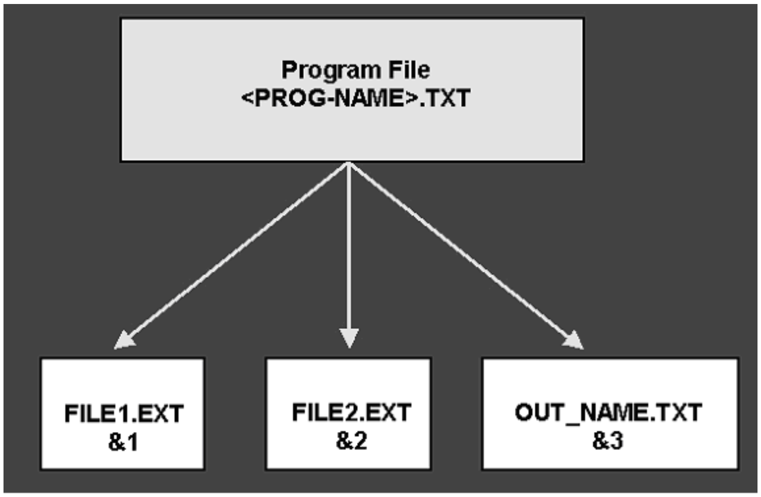


Figure 3-1 Program File

Each argument, separated by a blank space, represents a valid file name that will be created during post processing. The names may be user-defined, without path statements; the only required name is OUT_NAME.TXT that is used for special purposes as described later in this section.

<PROG-NAME >.TXT is the name of the file initially created by ConfigPost as entered by the user.

<PROG-NAME >.TXT is a text file.

FILE1.EXT is the first file to be created from the split. This file may have any legal file name (such as FILENAME.TXT or NAME.XXX (it does not have to be a .TXT extension).

FILE2.EXT is the second file to be created from the split.

Note: *As with FILE1.EXT, the file name does not require a .TXT extension.*

OUT_NAME.TXT is a special file name that must be created any time the split/merge function is used.

Note: *Information written to the OUT_NAME.TXT file is used to communicate to the programmer. The contents of this file are shown after post processing in PartMaker is completed. The first line of this file must be the word* ***FILES*** *followed by the names of the output files created that you want the user to view/edit when they press the Edit Program button in PartMaker. Any additional text in the file is for informational purposes only and will be displayed on the screen when post processing is completed. See the example below for more information on using* ***OUT_NAME.TXT****.*

```
#merge MERGEFILE.EXT FILE1.EXT FILE2.EXT FILE3.EXT<eob>
```

Merge is used to paste files back together in a specific order.

MERGEFILE.EXT may be any valid file name of the user's choice.

FILE1.EXT is the first file to be pasted in; FILE2.EXT, FILE3.EXT, etc. are any successive files you want to paste into MERGEFILE.EXT.

#remove FILE1.EXT<eob>: Any files following "#**remove"** will be deleted at the end of post processing.

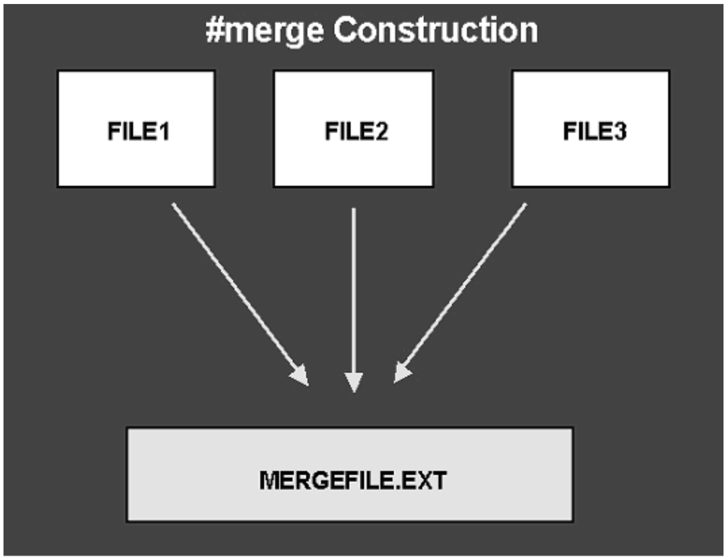


Figure 3-2 Merge Construction

Important! *You can create up to eight files using ConfigPost's file splitting capability.*

Separating Lines Into Specific Files

To put specific lines of code into the desired file, you use the command string & followed by an integer that defines the file location as follows:

&1 Put this line in the first file defined Inserting &1 is optional

&2 Put this line in the second fine defined

&3 Put this line in the third file defined

&1&2 Put this line in the first and second file defined

The only rule is that the &# characters must be the first characters on the line in which they appear. If no &# character appears, the line will be output to the first file (represented by &1) by default.

Creating a Tool List using the Split/Merge Feature

Let's look at an example of how to use the file handling features of ConfigPost by creating a tool list at the beginning of a G-code output file **as well as** a separate file, which only contains the tool list.

You can create a tool list in any Mill or Turn post processor that came with your PartMaker software. You have to take the following steps:

In the beginning of File Start format insert:

```
#define TOOL LIST POST FILE<eob>
#split <prog-name>.TXT<prog-name>.TMP<prog-name>.TLS
OUT_NAME.TXT<eob>
#merge <prog-name>.TXT <prog-name>.TLS <prog-name>.TMP<eob>
#remove <prog-name>.TMP <eob>
&2%<eob>
2O<prog-num1><eob>
&2(T<tool-num> <tool-name><tool-note>)<eob>
```

Looking at the previous text:

&1 corresponds to **<prog-name>.TMP**

&2 corresponds to **<prog-name>.TLS**

&3 corresponds to **OUT_NAME.TXT**

Putting % and <prog-num> at the beginning of the tool list ensures that the data needed at the beginning of the program is truly at the beginning and not after the tool list.

In the beginning of Tool Change format and Program Start format insert:

```
<if><first-tool-use><then>
          &2(T<tool-num> <tool-name><tool-note>)<eob>
<endif>
```

This construction tells PartMaker to write data to the **<prog-name>.TLS** file if a tool is called for the first time.

In the beginning of File End format insert:

```
&3FILES <prog-name>.TXT <prog-name>.TLS <eob>
&3<32><32><eob>
&3Files created are:<eob>
&3                              <prog-name>.TXT<eob>
&3                              <prog-name>.TLS <eob>
```

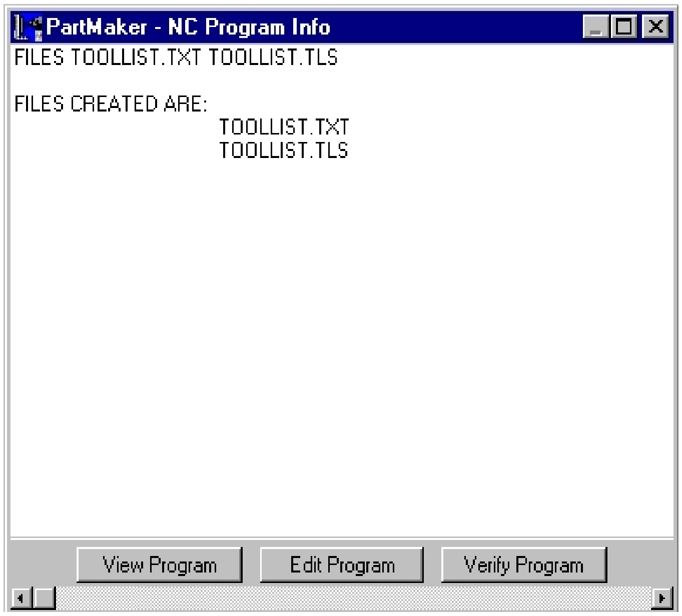
In this example, OUT_NAME.TXT is the last of **three** new files being created during the **#split** operation, the appearance &3 tells PartMaker to place data into the file OUT_NAME.TXT.

To view a post processed JOB file that outputs the two files constructed above:

1. In ConfigPost, save the Post file you just modified by choosing File menu, Save As, then name the file TOOLLIST.PST or a similar name when prompted.
2. In PartMaker, load the Post file you just modified by choosing Job Optimizer, Post Config File=?.
3. In PartMaker, open a completed .JOB file and choose Job Optimizer menu, Generate NC Program.
4. When prompted, name your program TOOLLIST.TXT, and then click<OK>.
5. You should see the contents of the OUT_NAME.TXT file as in Fig. 3-1.
6. Click the View Program button and you should see the Tool List as in Fig. 3-3.

TOOLLIST.TLS is a text file containing just the list of tools for the job you opened in step 3

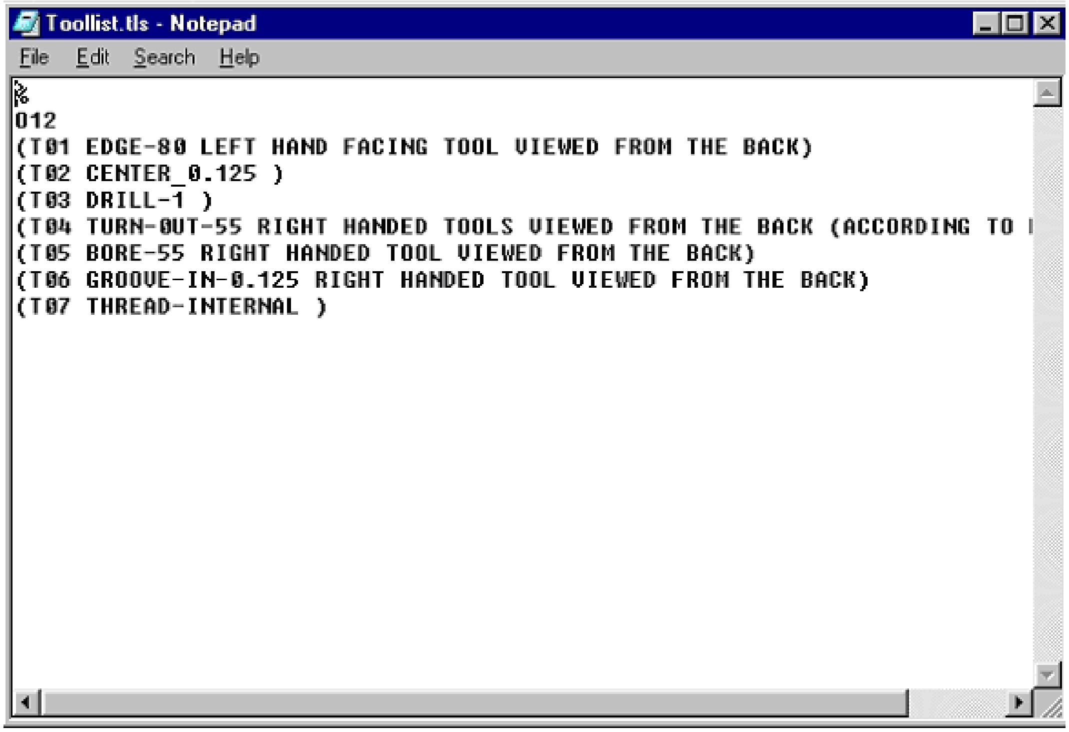
TOOLLIST.TXT is the completed .TXT file.



```
FILES TOOLLIST.TXT TOOLLIST.TLS

FILES CREATED ARE:
                         TOOLLIST.TXT
                         TOOLLIST.TLS
```

Figure 3-3 Toollist.txt & Toollist.tls



```
%
O12
(T01 EDGE-80 LEFT HAND FACING TOOL VIEWED FROM THE BACK)
(T02 CENTER_0.125 )
(T03 DRILL-1 )
(T04 TURN-0UT-55 RIGHT HANDED TOOLS VIEWED FROM THE BACK (ACCORDING TO
(T05 BORE-55 RIGHT HANDED TOOL VIEWED FROM THE BACK)
(T06 GROOVE-IN-0.125 RIGHT HANDED TOOL VIEWED FROM THE BACK)
(T07 THREAD-INTERNAL )
```

Figure 3-4 Toollist.tls-Notepad

Using <include> Statements

The ConfigPost **<include>** reserved word allows you to customize your post processor to include additional information in the part program that a user enters “on the fly” when prompted. The **<include>** reserved word should be used as following:

```
{Please enter your name: <include> Programmer}<eob>
```

Each time **<include>** is found in a part program format in a Post file, PartMaker will display the current process number, corresponding tool name, and a prompt to enter additional process information.

Example of using <include> reserved word:

An example of an application using an **<include>** statement can be found in the DEMO.PST file in your PartMaker POSTLIB (PartMaker Mill users) directory. If you only have PartMaker Turn, you can copy the formats below into any of the Post files provided with your PartMaker software. DEMO.PST is a generic Fanuc post processor and can be used “as is” or you can copy its functions into another Post file you want to use.

1. Start ConfigPost and choose File, Open to display the Open dialog.
2. Locate and double-click DEMO.PST in the POSTLIB directory.
3. Go to the Configure Mill Menu and Choose Process Header Formats.
4. The Program Start format shown below should have the line:

Program Start format:

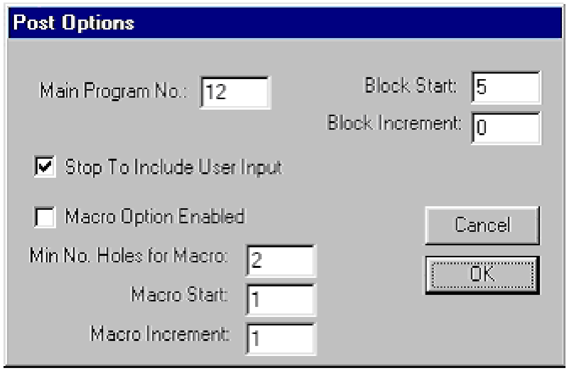
```
%<eob>
:<prog-num1><eob>

{Please enter your name: <include> Programmer}<eob>

(TOTAL MACHINING TIME: <time-total>)<eob>
(MACHINING <face-name> FACE)<eob>
(MACHINING PROCESS <process-id>)<eob>
{N<seq> }G80 G40 G17<eob>
{N<seq> }G30 G91 Z0<eob>
{N<seq> }T<tool-num> M6<eob>
(<tool-name> <tool-note>)<eob>
(TIME FOR THIS TOOL: <time-process>)<eob>
{N<seq> }G90 <work-offset> <motion> X<x-coord> Y<y-coord> S<speed-rpm> <spindle-on><eob>
{N<seq> }G43 Z<rapid-abs> H<tool-offset> <coolant><eob>
```

5. In PartMaker, with existing .JOB, .TDB and .CDB files open (you can use DEMO.JOB or another job file), choose **Job Optimizer, Post Config File=** to display a dialog.
6. Locate and open DEMO.PST or another Post file you want to use.

7 Choose **Job Optimizer, Post Options** to display the Post Options dialog.



Post Options

Main Program No.: 12

Block Start: 5

Block Increment: 0

☑ Stop To Include User Input

□ Macro Option Enabled

Min No. Holes for Macro: 2

Macro Start: 1

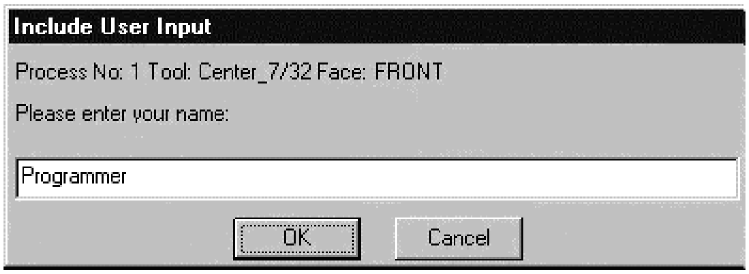
Macro Increment: 1

Figure 3-5 Post Options

8 Make sure the Stop to Include User Input check box is turned on as shown in the previous illustration, then click <OK>.

9 Choose Generate NC Program from the Job Optimizer Menu.

As post processing is being carried out, the Include User Input dialog will be displayed:



Include User Input

Process No: 1 Tool: Center_7/32 Face: FRONT

Please enter your name:

Programmer

Figure 3-6 Include User Input

10 Enter your name in the text box to replace the default name (Programmer), then click <OK>.

*Notice how the name will be inserted into the output file instead of the entire line containing the **<include>** statement. The value assigned to the **<include>** reserved word will be promptly "forgotten" by the Post file after this line has been passed over during the post processing session.*

Performing Math Operations

ConfigPost has the ability to perform mathematical operations using square brackets as operational indicators, for example:

```
[<z-coord>/2.0]
```

The result of mathematical statements may be used in conditional statements, for example:

```
<if><smaller>(<z-coord>,[4.0 / 2.0])<then>
THE TOOL IS LESS THAN 2 INCHES ABOVE THE PART<eob>
<endif>
```

Please notice that in the case of the comparison operation, it is not necessary to enclose the mathematical operation in square brackets; instead, the operation uses parentheses and commas as delimiters as shown below:

```
<if><smaller>(<z-coord>,4.0 / 2.0)<then>
THE TOOL IS LESS THAN 2 INCHES ABOVE THE PART<eob>
<endif>
```

Note: *The output format of mathematical statements is determined by the format of the input. If BOTH values are integers, then the output is always an integer (no decimal point). If either value has a decimal, then the output is always a real number (decimal point).*

Using Mathematical Functions

The following functions are supported :

SIN, COS, TAN, SQRT

Examples of usage:

```
~<var-3>[SIN(30)]<eob>
~<var-4>[SQRT((<x-coord>)*(<x-coord>)+(<y-coord>)*(<y-coord>))]<eob>
```

Note1*: Angle measured in degrees must be used with SIN, COS, TAN.*

Note2*: Upper case letters only must be used in function names: SIN, COS, TAN, SQRT.*

Using Negative Values

When performing operations where negative numbers may be encountered, you must enclose the number in parentheses, for example:

```
[(<x-coord>) *(-1.0)]<eob>
```

Using <prompt> Statements with Variables

The ConfigPost **<prompt>** keyword allows you to customize your post processor to include additional information in the part program that a user enters “on the fly” when prompted. Each time **<prompt>** is found in a part program, PartMaker will display the current process number, corresponding tool name, and a prompt to enter additional process information.

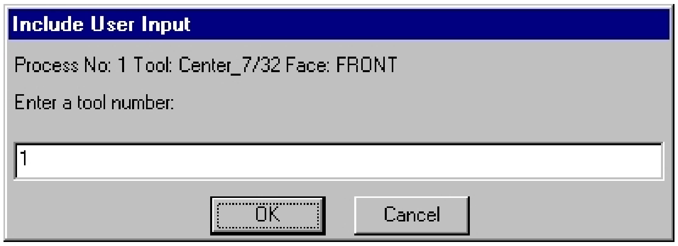
The difference between the <prompt> and <include> statements is that with the **<prompt>** construction the value entered by the user is assigned to **<prompt>**; it will not output automatically as **<include>** did. **<prompt>** is a very useful construction when you want the value entered by the user to be used on a line that appears later in the Post file.

Example 1: Using **<prompt>:**

Here is a generic construction that you can create to see how prompt works:

```
{Enter a tool number:<prompt>1}<eob>
<if><eq-number>( <prompt>,1)<THEN>
        TOOL-IS-ONE<eob>
<endif>
<ifnot><eq-number>( <prompt>,1)<then>
        TOOL-IS-LARGER-THAN-ONE<eob>
<endif>
```

Using this construction, you will be prompted with the dialog below when generating an NC Program with a Post file including this construction:


Include User Input

Process No: 1 Tool: Center_7/32 Face: FRONT

Enter a tool number:

1

Figure 3-7 Enter a Tool Number

If you enter 1 at the prompt, you will see the following in your G-code output:

TOOL-IS-ONE

If you input something other than 1 at the prompt, you will see the following in your G-code output:

TOOL-IS-LARGER-THAN-ONE

Example 2: Using **<prompt>**:

This example is a construction used to configure a bar puller. Notice how the value given to **<prompt>** is passed to a variable that is then used later in a mathematical calculation.

```
{Please enter the puller grip length: <prompt>0.5}<eob>
~<var-3><prompt><eob>
G1 G98 Z-[<part-length>+<cut-off-width>+<var-3>] F100. <EOB>
```

Important! *Whenever using the* ***<include>*** *or* ***<prompt>*** *statements, be sure to turn on the* ***Stop to Include User Input*** *check box in the Post Options dialog.*

Variables

Variables Names Dialog

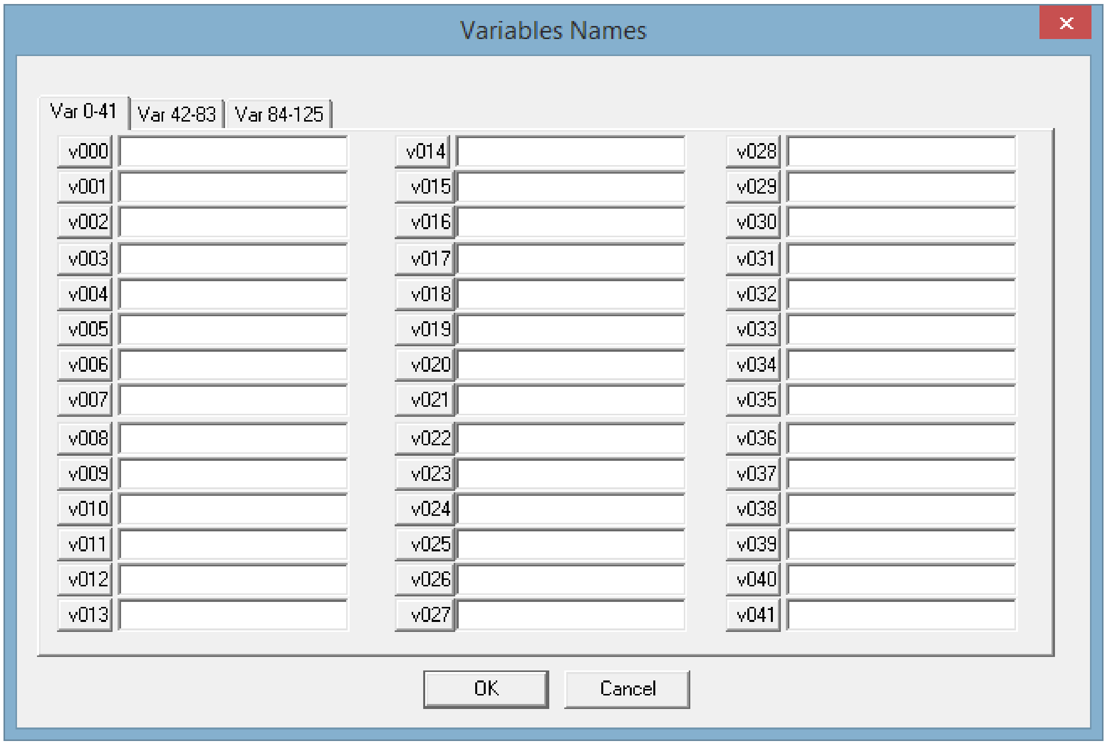


Figure 3-8 Variables Names

Important! *You can only create, delete and modify variables in the* ***Variable Names Dialog*** *if you have licensed ConfigPost/Pro.*

This dialog appears when you choose **Variables Names…** command from the **Settings** Menu.

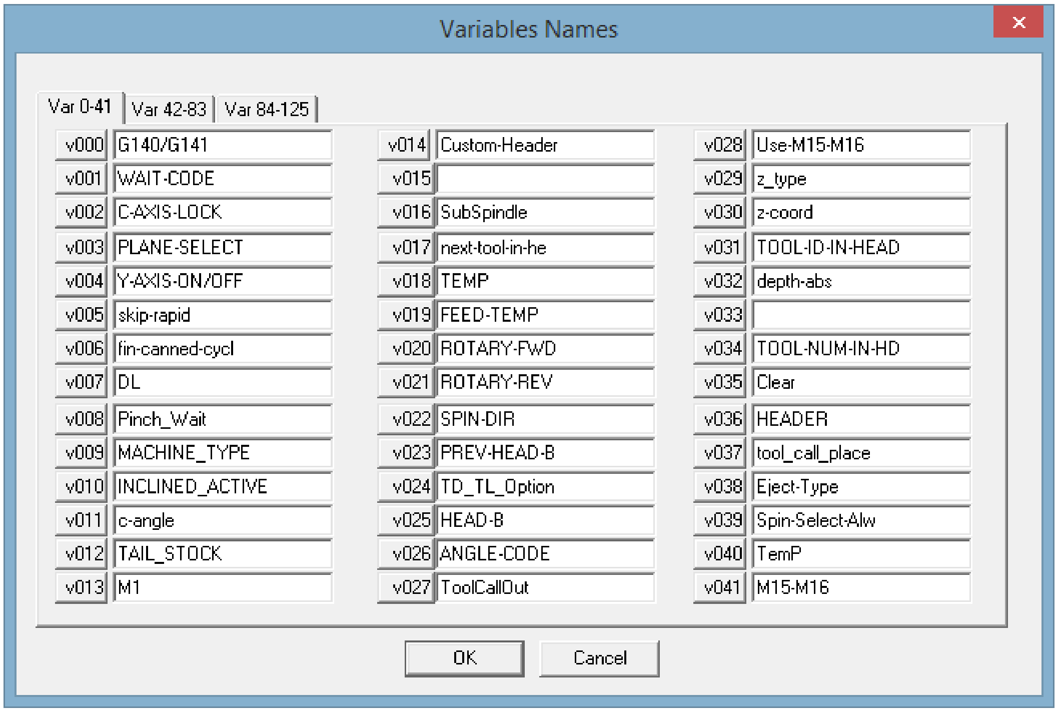
This command displays the Variables Names dialog where you can enter the names of user defined variables. For Turn-Mill and SwissCAM Post files the same variables can be used in either the Milling or the Turning section of ConfigPost.

Using Variables

ConfigPost provides you with 125 variables that can be assigned string or numeric values. These variables can be found with the string reserved words and are labeled **<var-0>** through **<var-125>**. Once variables are assigned values they will retain those values until they are reassigned other values.

Assigning Names to Variables

The first step in using Variables is to assign them names. Assigning meaningful names to variables helps you understand their usage in the Post file. Variable names are assigned in the Variables Names dialog accessed from the Settings menu.



v000	G140/G141	v014	Custom-Header	v028	Use-M15-M16
v001	WAIT-CODE	v015		v029	z_type
v002	C-AXIS-LOCK	v016	SubSpindle	v030	z-coord
v003	PLANE-SELECT	v017	next-tool-in-he	v031	TOOL-ID-IN-HEAD
v004	Y-AXIS-ON/OFF	v018	TEMP	v032	depth-abs
v005	skip-rapid	v019	FEED-TEMP	v033	
v006	fin-canned-cycl	v020	ROTARY-FWD	v034	TOOL-NUM-IN-HD
v007	DL	v021	ROTARY-REV	v035	Clear
v008	Pinch_Wait	v022	SPIN-DIR	v036	HEADER
v009	MACHINE_TYPE	v023	PREV-HEAD-B	v037	tool_call_place
v010	INCLINED_ACTIVE	v024	TD_TL_Option	v038	Eject-Type
v011	c-angle	v025	HEAD-B	v039	Spin-Select-Alw
v012	TAIL_STOCK	v026	ANGLE-CODE	v040	TemP
v013	M1	v027	ToolCallOut	v041	M15-M16

Figure 3-9 Global Variables

Assigning Values to Variables

Variables can be assigned string or numeric values using the following construction:

~<VARIABLE> VALUE BEING ASSIGNED <EOB>

Example: Let's assume **var-12** in Variables Names Dialog was given a name "FACE-CLEAR". This variable can be assigned in the following manner:

```
<IF><metric-post><THEN>
        ~<v12-FACE-CLEAR>2.<eob>
<ENDIF>
<IFNOT><metric-post><THEN>
        ~<v12-FACE-CLEAR>0.05<eob>
<ENDIF>
```

Inserting Variables

Variables can be inserted into a Post file in the same manner as any other reserved word can be.

Procedures

Important! *You can only create and delete and Procedures if you have licensed ConfigPost/Pro.*

Using Procedures

Procedures in ConfigPost allow the user to simplify a post processor by making it modular. Using Procedures enables the user to decrease the size of a post configuration file by allowing him to call sections of machine code with a single reserved word instead of repeating code in every program format that might require it. This is done by placing all the required ConfigPost reserved words and literals in a procedure format and calling the procedure when needed. Procedures can be very helpful for such programming exercises as activating an auxiliary device like a bar puller, which can be called during any number of different program formats.

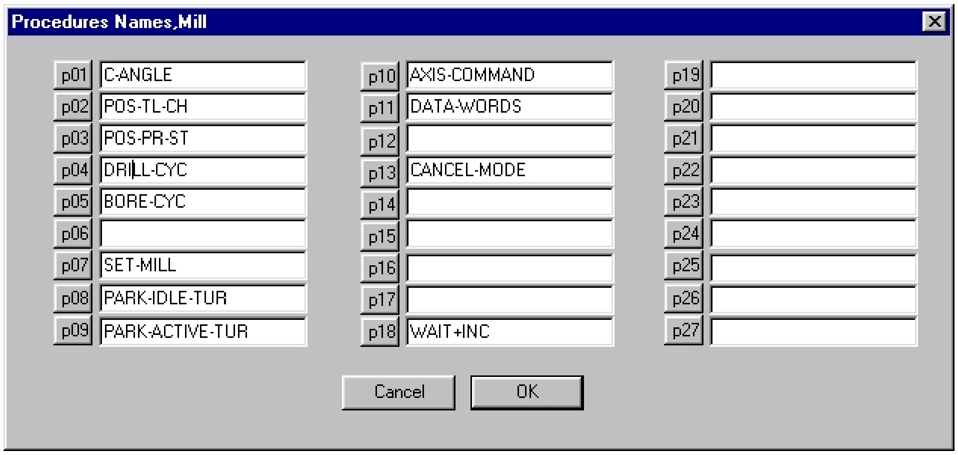
The steps involved in creating and using procedures are described below. The example shown below is a bar feeding sequence taken from a Hardinge TT-65 Post Processor. You can follow these same steps to implement new procedures or modify existing procedures post processors for your machines.

Assigning Names to Procedures

When using procedures, the first step is to assign them meaningful names. This can be done in the Procedure Names dialog accessed from the Settings menu.
Like Variables, it is vital to assign procedures meaningful names in order to make your Post file more readable.

1 To assign a name to a procedure first go to the Settings menu and choose Procedure Names.

2 Select either Procedure Names,Mill or Procedure Names,Turn or Procedure Names,WireEDM Dialog. The screen shown on the following page will appear (this is an example for Procedure Names, Mill).



p01	C-ANGLE	p10	AXIS-COMMAND	p19	
p02	POS-TL-CH	p11	DATA-WORDS	p20	
p03	POS-PR-ST	p12		p21	
p04	DRILL-CYC	p13	CANCEL-MODE	p22	
p05	BORE-CYC	p14		p23	
p06		p15		p24	
p07	SET-MILL	p16		p25	
p08	PARK-IDLE-TUR	p17		p26	
p09	PARK-ACTIVE-TUR	p18	WAIT+INC	p27	

Figure 3-10 Procedure Names, Mill

Note: *This dialog is taken from the Hardinge TT-65 post processor and may appear different from the user's screen.*

Important for Turn-Mill and SwissCAM! *When creating a procedure to be used in both Milling and Turning processes, you must define a* ***separate*** *identical procedures in both the Milling and Turning sections of your *.PST file. In addition, procedures must be separately defined in *.SUB files in order to be accessible for sub spindle operations.*

Defining Procedures

Example:

3 Go to the Configure Turn menu and choose the Procedure Formats menu.

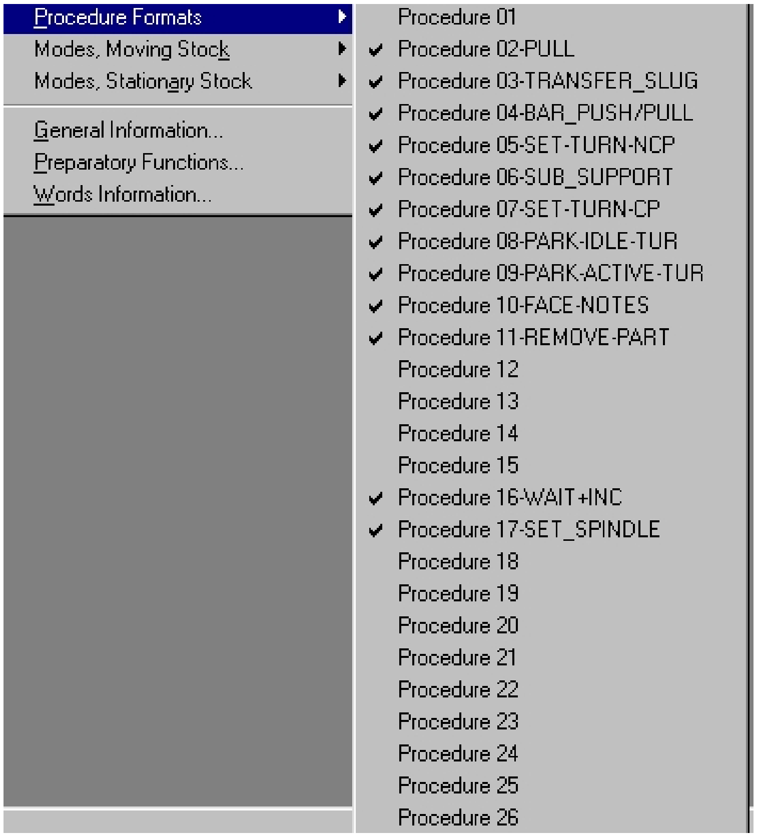
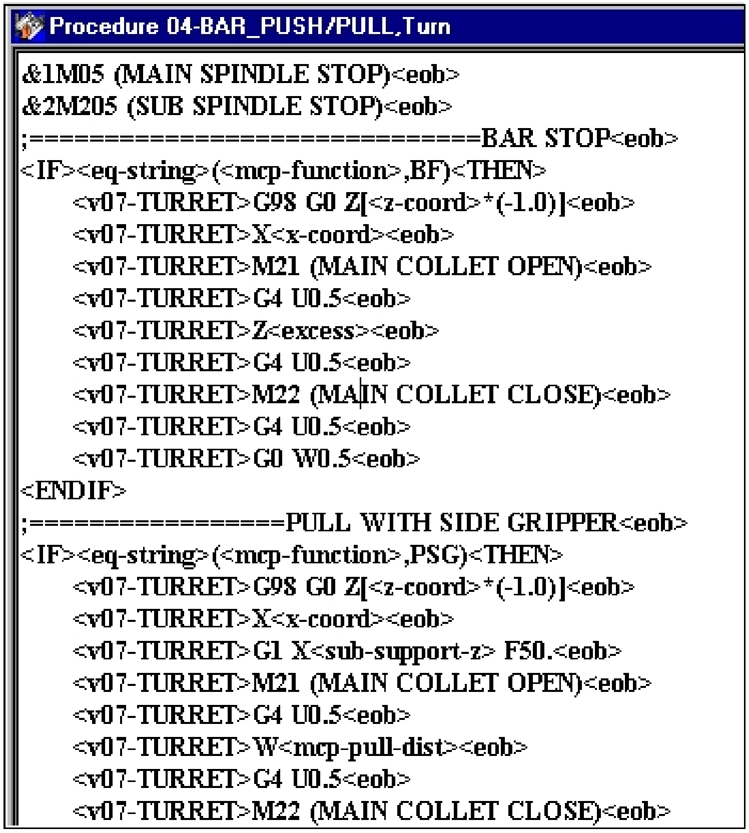


Figure 3-11 Procedure Formats

4 Select the procedure that is to be defined. The format for that specific procedure will appear in a separate window. (See Figure 3-12 on the following page).



```
&1M05 (MAIN SPINDLE STOP)<eob>
&2M205 (SUB SPINDLE STOP)<eob>
;=============================BAR STOP<eob>
<IF><eq-string>(<mcp-function>,BF)<THEN>
      <v07-TURRET>G98 G0 Z[<z-coord>*(-1.0)]<eob>
      <v07-TURRET>X<x-coord><eob>
      <v07-TURRET>M21 (MAIN COLLET OPEN)<eob>
      <v07-TURRET>G4 U0.5<eob>
      <v07-TURRET>Z<excess><eob>
      <v07-TURRET>G4 U0.5<eob>
      <v07-TURRET>M22 (MAIN COLLET CLOSE)<eob>
      <v07-TURRET>G4 U0.5<eob>
      <v07-TURRET>G0 W0.5<eob>
<ENDIF>
;=================PULL WITH SIDE GRIPPER<eob>
<IF><eq-string>(<mcp-function>,PSG)<THEN>
      <v07-TURRET>G98 G0 Z[<z-coord>*(-1.0)]<eob>
      <v07-TURRET>X<x-coord><eob>
      <v07-TURRET>G1 X<sub-support-z> F50.<eob>
      <v07-TURRET>M21 (MAIN COLLET OPEN)<eob>
      <v07-TURRET>G4 U0.5<eob>
      <v07-TURRET>W<mcp-pull-dist><eob>
      <v07-TURRET>G4 U0.5<eob>
      <v07-TURRET>M22 (MAIN COLLET CLOSE)<eob>
```

Figure 3-12 Procedure 04 Bar Push/Pull

Note: *This dialog is taken from the Hardinge TT-65 post processor and may appear differently on the user’s screen.*

Calling Procedures

Inserting a procedure can be done in the same manner as inserting any other reserved word.

1 Using the mouse, scroll down to the Procedure Calls list located after the Procedures Words.

2 Double click on the desired procedure. It will automatically be inserted in the program.

3 Below is an example of calling a procedure depending on a user condition:

```
;--BARFEED<EOB>
<IF><UC2-BARFEED><THEN>
<p04-BAR_PUSH/PULL><EOB>
<ENDIF
```

Material Control Processes (MCP)

Usage of MCP

A Material Control Process ("MCP") can be used for PartMaker/Turn, PartMaker/TurnMill and PartMaker/SwissCAM applications.

Material Control Processes can be broken into two categories:

- Processes that **use a turret tool**:
    - BF Bar Feed
    - PSG Pull with Side Gripper
    - PFG Pull with Front Gripper
    - RMS Remove from Main Spindle
    - RSS Remove from Sub Spindle
- Processes that do not use a turret tool:
    - TF Tailstock Forward
    - TR Tailstock Retract
    - PSS Pull with Sub Spindle (called Rechuck in PartMaker/SwissCAM)
    - TPM Transfer Part from Main to Sub Spindle

The String Reserved Word **<mcp-function>** specifies a name of an MCP.

<mcp-function> can have the following values:

BF, PSG, PFG, RMS, RSS, TF,TR,PSS,TPM.

The Logical Reserved Word **<mat-control-process>** is used to determine if the current process is a Material Control Process.

The example below checks to see if the current process is the Bar Feed Process:

```
<IF><mat-control-process> <THEN>
        <IF><eq-string>(<mcp-
        function>,BF)<THEN>
        ...
        <ENDIF>
<ENDIF>
```

Other reserved words used with MCP:

Numeric Reserved Words: <mcp-pull_distance>, <mcp-total-pull>, <catcher-x>, <catcher-z>

String Reserved Word **<catcher-num>**.

<mcp-pull_distance>: specifies the pull distance entered in PartMaker's Material Control Process Parameters dialog

<mcp-total-pull>: specifies cumulative pull distance

<catcher-x>, <catcher-z>, <catcher-num> are used with Remove MCP (RSS, RMS) for indexing the turret from the gripper to the part release position.

Note: *in order to add cumulative pull distance to all z-coordinates check "Add Pull Distance To Z-Coordinates" in General Info, Turn dialog.*

NC Program Output for MCP

A Program Format called MCP-Action format is used to configure the output for Material Control Processes..

The chart below shows how PartMaker outputs an NC program for a Material Control Process **that does not use a turret tool**.

Output Steps for MCP Not Using a Turret Tool
1> MCP-Action

The chart below shows how PartMaker outputs an NC program for a Material Control Process **that uses a turret tool**. The first step is to output a Process Header Format as following: the Program Start Format is output if a given process is the first process in the Process Table, otherwise if a tool used in a given process needs to be activated (i.e. brought into position to begin cutting) the Tool Change Format is output, otherwise the Process Start Format is output.

Output Steps for MCP Using a Turret Tool
1> Process Header: Program Start Tool Change Process Start
2> MCP-Action
3> Process End

Note 1: *Tool positioning rules in Process Header Formats for MCP are similar to the rules for tool positioning in turning cutting processes, that is* ***<x-coord>, <z-coord>*** *of the first position point have to be used. However, because MCP has no toolpath, this first point (called "Position Point") is either calculated by PartMaker or entered by the user in PartMaker's Material Control Process Parameters dialog.*

Note 2: *If you configure a Post File in a manner that Process Header Formats will handle MCPs you must use logic to avoid certain actions not required by MCPs, e.g. turning on the spindle. Another alternative is to start every Process Header Formats with a line* ***<ifnot><mat-control-process> <then>*** *and configure the MCP-Action to handle the entire MCP.*

MCP for PartMaker/Turn and TurnMill

Material Control Processes used by **PartMaker/Turn:**

Bar Feed, Pull with Side Gripper, Pull with front Gripper, Tailstock Forward, Tailstock Retract.

Material Control Processes used by **PartMaker/TurnMill:**

Bar Feed, Pull with Side Gripper, Pull with front Gripper, Tailstock Forward, Tailstock Retract,Remove from Main Spindle, Remove from Sub Spindle,Pull with Sub Spindle ,Transfer Part from Main to Sub Spindle.

An example of an MCP-Action format for **Cobra.pst** (located in **PartMaker\PM-Turn\Postlib** folder) is shown below:

```
<IF><eq-string>(<mcp-function>,PSG)<THEN>
        <p02-BARPULL-FINGER><eob>
<ENDIF>
<IF><eq-string>(<mcp-function>,PFG)<THEN>
        <p03-BARPULL-BUSH><eob>
<ENDIF>
<IF><eq-string>(<mcp-function>,BF)<THEN>
        <p04-BARFEED><eob>
<ENDIF>
<IF><eq-string>(<mcp-function>,TF)<THEN>
        <32><32><eob>
        (TAILSTOCK FORWARD)<eob>
        M84<eob>
<ENDIF>
<IF><eq-string>(<mcp-function>,TR)<THEN>
        <32><32><eob>
        (TAILSTOCK RETRACT)<eob>
        M85<eob>
<ENDIF>
```

MCP Rechuck for PartMaker/SwissCAM

There is only one Material Control Process used by PartMaker/SwissCAM, namely, ***Pull with Sub Spindle called "Rechuck" in PartMaker/SwissCAM.***

Note: *MCP Rechuck should* ***not*** *be confused with Collet Re-Chuck.*

An example of MCP-Action format for the Citizen-M Post File is shown below:

```
<IF><eq-string>(<mcp-function>,PSS)<THEN>
        <p13-SET-TURN>
        <IFNOT><eq-string>(<mode-sub-sp-next>,SF)<THEN>
                <p21-M0SN-RECHUCK>
        <ENDIF>
        <IF><eq-string>(<mode-sub-sp-next>,SF)<THEN>
                <p22-M0SF-RECHUCK>
        <ENDIF>
        ~<v18-RECHUCK-ZR><z-coord><eob>
        ~<v37-PREV-MOVE-TOOL>0<eob>
        ~<v22-PRE-STCK-Z-POS>0<eob>
<ENDIF>
```

Import/Export WRI Files

What is a WRI file?

A WRI File is a text file produced by ConfigPost application. The WRI File shows all Post Processor File Formats as text. This file can be opened by any editor such as Notepad or Wordpad. The WRI file can be modified and loaded back by ConfigPost application.

How to Create a WRI File?

In order to create a WRI File the Post Processor File(with extension *.pst or *.sub) must be loaded. This can be done by using the File Open command. From the File Menu choose Export WRI File, then Complete or Export WRI File and Non-Empty Formats Only.

- When Export WRI File Complete command is chosen then ConfigPost Exports all Program Formats including empty ones.
- When Export WRI File Non-Empty Formats Only command is chosen then ConfigPost Exports only Non-Empty Program Formats.

Important! *If you have not licensed ConfigPost/Pro, you will NOT be able to import WRI files.*

How to Open the WRI File?

In order for the WRI File to be opened by ConfigPost application the Post Processor File(with extension *.pst or *.sub) must be loaded. Then choose Import WRI File... from the File Menu. If you have made modifications in the WRI File they will take an effect right after WRI File is opened.

Structure of WRI File

The WRI File contains all information from the Post Processor File(with extension *.pst or *.sub) in text form.

When the WRI File is created ConfigPost places data from all dialogs and Program Formats into WRI File. Each Program Format has a unique tag which allows ConfigPost to load WRI File back. Therefore the **tags should not be modified**.

Example of Tag:

```
![1][1]---Tool Change,Turn----------------------------
```

Chapter 4: ConfigPost for Turn-Mill and SwissCAM

Using ConfigPost for PartMaker/Turn-Mill and PartMaker/SwissCAM

Turn-Mill Centers and Swiss type lathes are capable of producing parts with greater precision at lower cost than other CNC machines because they eliminate the need for human intervention between turning and milling operations. A complex part can be completely manufactured on one machine, with only one setup.

These machines perform a full set of lathe operations: facing, turning, grooving and threading while also being able to mill and drill holes on different surfaces. Multiple turrets and dual spindles further increase the machines' productivity and complexity.

ConfigPost for PartMaker/Turn-Mill and ConfigPost for PartMaker/SwissCAM tool station has been structured to handle this increased complexity by incorporating programming formats for both Milling and Turning. Users of ConfigPost for PartMaker/Turn-Mill and PartMaker/SwissCAM will notice both the Configure Mill and Configure Turn menus are enabled when in this application. ConfigPost for Turn-Mill and SwissCAM can also handle configuration of all Milling and Turning post configuration files supported by PartMaker.

How Each Process is Handled by ConfigPost for Turn-Mill

The following illustrations show how PartMaker/Turn-Mill handles each process in a Process Table involving turning as well as milling and drilling operations.

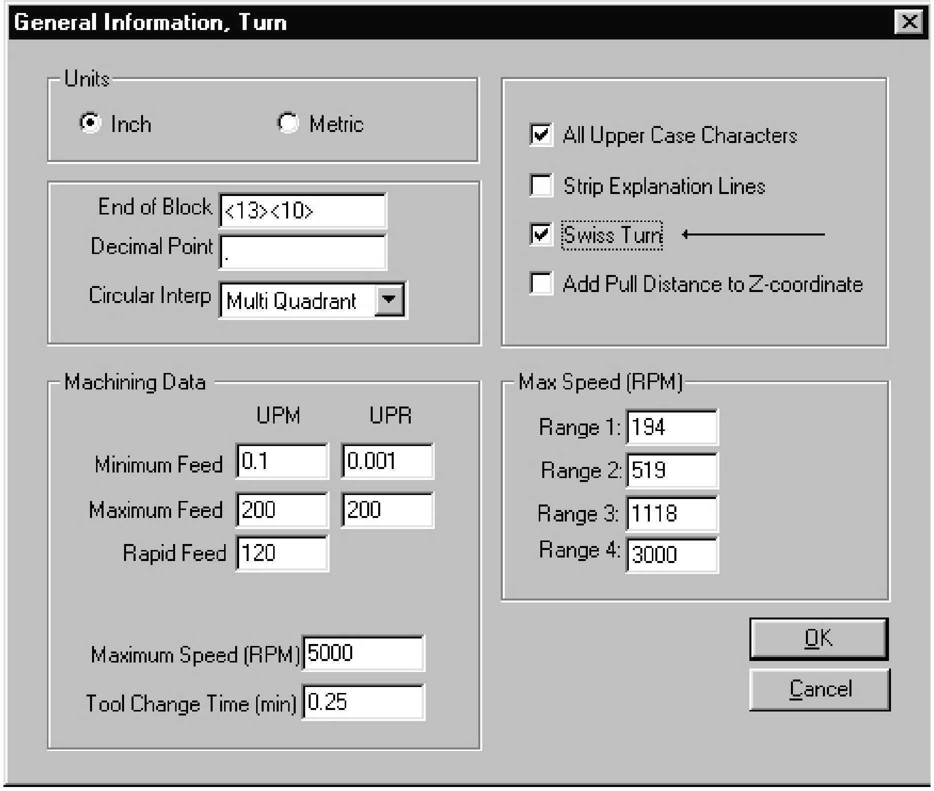
If a process is a milling or drilling process, it will be converted to G-code through the Milling section in ConfigPost, i.e. PartMaker will recognize a process on the Process Table as a milling process and apply milling or hole making formats to it. To configure milling process output for PartMaker/Turn-Mill programs, edit the program formats found in the Configure Mill menu in ConfigPost for PartMaker/Turn-Mill.

If a process is a turning process, it will be converted to G-code through the turning section in ConfigPost, i.e., PartMaker will recognize a process on the Process Table as a turning process and apply turning formats to it. To configure turning process output for PartMaker/Turn-Mill programs, edit the program formats found in the Configure Turn menu in ConfigPost for Turn-Mill.

How Each Process is Handled by ConfigPost for SwissCAM

ConfigPost for SwissCAM handles processes in the same manner as it does for Turn-Mill centers. The only difference is that posts for Swiss type lathes look to the collet Re-Chuck program format before entering the Program Start program format. The collet Re-Chuck format contains NC code necessary to be included at the beginning of a program and contains information not available to the Program Start program format. (i.e. information relating to the cut-off tool.)

ConfigPost recognizes posts as Swiss if the Swiss Turn check box in the General Information dialog is checked.



General Information, Turn

Units: ● Inch ○ Metric

End of Block: <13><10>

Decimal Point: .

Circular Interp: Multi Quadrant

☑ All Upper Case Characters

□ Strip Explanation Lines

☑ Swiss Turn

□ Add Pull Distance to Z-coordinate

Machining Data

	UPM	UPR
Minimum Feed	0.1	0.001
Maximum Feed	200	200
Rapid Feed	120	

Maximum Speed (RPM): 5000

Tool Change Time (min): 0.25

Max Speed (RPM)

Range 1:	194
Range 2:	519
Range 3:	1118
Range 4:	3000

Differentiating Between Turning and Milling Processes

PartMaker is able to differentiate between turning and milling processes by the type of face window in which a process has been generated. If a process has been created in a turning Face Window, PartMaker will use the program formats in the Configure Turn menu. If the face window in which a process was created is anything other than a Face Window of type Turn, PartMaker will format the output of that process based on the program formats in the Configure Mill menu.

Using Different Milling Face Types

In addition to accommodating turning and milling operations, Turn-Mill programs must also accommodate a number of different types of Milling process output, depending on the type of “live tooling” operation being performed. ConfigPost is able to handle the variety of milling process output necessary for Turn-Mill programming by differentiating between the face types designated in PartMaker.

To understand how PartMaker differentiates between different types of milling process output, look at the Linear Move example below. PartMaker uses each Face Window type as a logical reserved word and as such is able to generate the proper milling output whether the current milling Face Window is planar or rotational.

Linear Move, Mill

```
<IF><face-end-index><THEN>
        {<motion> }{X<c-radius> }{Z<z-coord> }{F<feed-upm>}<eob>
<ENDIF>
<IF><face-end-polar><THEN>
        {<comp-status> }{<motion> }{X<x-coord> }{C<y-coord> }{Z<z-coord> }{F<feed-upm>}<eob>
<ENDIF>
<IF><face-dia-polar><THEN>
        {<motion> }{X<x-coord> }{C<y-coord> }{F<feed-upm>}<eob>
<ENDIF>
<IF><face-dia-index><THEN>
        {<motion> }{Z<z-coord> }{X<x-coord> }{F<feed-upm>}<eob>
<ENDIF>
<IF><face-cyl><THEN>
        {<motion> }{<comp-status> }{X<x-coord> }{Z<z-coord> }{C<c-angle> }{F<feed-upm>}<eob>
<ENDIF>
<IF><face-zy><THEN>
        {<comp-status> }{<motion> }{X<x-coord> }{Y<y-coord> }{Z<z-coord> }{F<feed-upm>}<eob>
<ENDIF>
<IF><face-polygon><THEN>
        {G0 C<c-angle>}<eob>
        {G1 Y<y-coord> }{F<feed-upm>}<eob>
        <ENDIF>
<IF><face-xy><THEN>
        {<comp-status> }{<motion> }{X<x-coord> }{Y<y-coord> }{Z<z-coord> }{F<feed-upm>}<eob>
<ENDIF>
<IF><face-zx><THEN>
        {<comp-status> }{<motion> }{X<x-coord> }{Z<z-coord> }{F<feed-upm>}<eob>
<ENDIF>
```

Programming Milling Processes with X-Oriented Tools

When using X-oriented tools for Face Windows of Machining Function type Mill YZ Plane, the handling of coordinates changes according to the following scheme:

Z-Oriented Tool	X-Oriented Tool
<x-coord>	<y-coord>
<y-coord>	<z-coord>
<z-coord>	<x-coord>

According to the table above, 2D positioning and cutting moves are in the YZ plane and the plunge-in axis is the X-axis.

The situation is very similar for X-oriented tools for Faces Windows of machining functions Mill Diam, Index and Mill Cylinder, however, 2D coordinates are defined in terms of Z and C coordinates instead of Z and Y coordinates. To differentiate between these two cases, each plane will be referred to as ZC Plane and ZY Plane, respectively.

X-oriented tools are also used for Face Windows of Machining Function Mill Diam, Polar and Mill Polygon. The type of motion required is solely 2D and plunge-in is not required.

Using Motion Formats with X-Oriented Tools

Scenario A: If your CNC allows Y, Z, and X axis motion programmed in the same line, you must place all three reserved words in the Horizontal Rapid Move format, leaving the Vertical Rapid Move format empty.

Scenario B: If your CNC does not allow Y, Z, and X axis motion programmed in the same line, you must use both the Horizontal Rapid Move format and the Vertical Rapid Move format.

Horizontal Rapid Move, Mill

The Horizontal Rapid Move program format determines the output of rapid positioning moves in the part program. When you use modality delimiters { } around the <y-coord> and <z-coord> reserved words in this program format, PartMaker will strip redundant Y or Z coordinates from each line as they are encountered.

If Scenario A above is true, you must leave the Vertical Rapid Move format empty.

```
For ZY Plane:
        {n<seq> }{<motion> }Y<y-coord>Z<z-coord><eob>
For ZC Plane:
        {n<seq> }{<motion> }C<c-angle>Z<z-coord><eob>
```

Vertical Rapid Move, Mill

If Scenario A is true, you must leave the Vertical Rapid Move format empty. If Scenario B is true, you must include the <x-coord> reserved word in this program format.

```
{n<seq> }{<motion> }X<x-coord><eob>
```

Rules for Program Start, Tool Change, and Process Start

With X-Oriented Tools

The information in this section provides the rules or cases that apply to specific Process Header formats, namely, Program Start, Tool Change and Process Start.

The following rules apply to program formats that you define in a Post file.

1. **Positioning**

   - **YZ-Plane**

     The reserved words <y-coord> and <z-coord> are assigned the values of the first path point and must appear in all process header formats to provide the first positioning move to the start of the path.

   - **ZC-Plane**

     The reserved words <z-coord> and <c-angle> are assigned the values of the first path point and must appear in all Process Header formats to provide the first positioning move to the start of the path.

2. The reserved word **<x-coord>** must be used for the Process Start program format.

   <x-coord> is calculated as the largest RAPID value of the current and previous process, and allows the tool to be retracted from the part to a safe plane.

3. Using the **<x-coord>** reserved word is optional for the Program Start and Tool Change program formats.

   <x-coord> is calculated in the following manner:

   <x-coord> = Tool Change (X)

Note: *The X coordinate Tool Change (X) is defined in PartMaker Defaults dialog which is accessed in PartMaker's Job Optimizer menu. Whether <x-coord> is used or not, the PartMaker assumes that the tool is at this X level after the code for any of the previously mentioned formats is generated.*

Important! *The Tool Change format is output only if there is a change in tool number between processes. If there is not a change in tool number, the Process Start format is output.*

Main and Sub Spindle File Handling

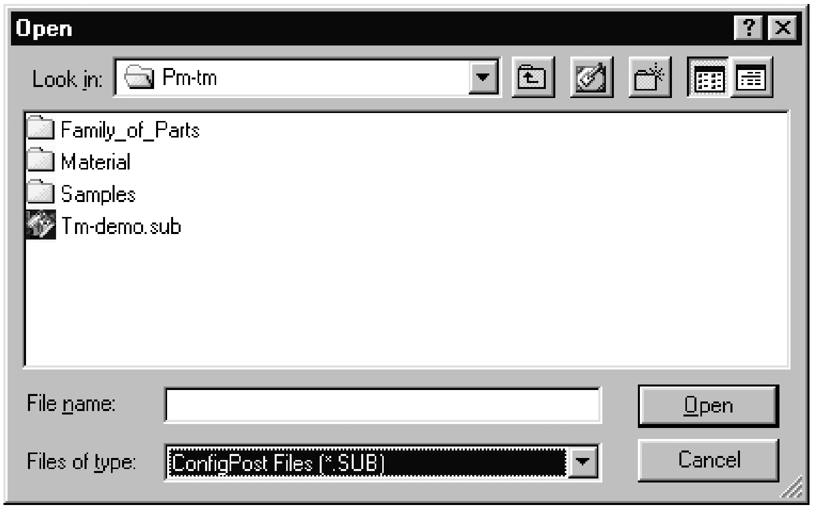
In order to accommodate turn-mill operations that involve processes being performed on either the main or sub spindle, PartMaker uses a separate post configuration file for the sub-spindle. The post processor for the FANUC18T control (provided with your PartMaker Turn-Mill software) is actually comprised of two files:

- **Tm-demo.pst**: Post configuration file for main spindle operations.
- **Tm-demo.sub**: Post configuration file for sub spindle operations.

When you load **Tm-demo.pst** in PartMaker by choosing Post Config File=?, **Tm-demo.sub** will automatically be loaded and used to generate output for the sub-spindle when sub-spindle operations are being programmed. Remember, you specify sub-spindle processes in PartMaker by turning off the Main Spindle check box in the Setup dialog for a given Face Window.

You can use ConfigPost for Turn-Mill to configure milling and turning process output for files types of both .PST and .SUB. Configuring post processors for files with a .PST extension is performed in the standard manner. Opening a post configuration file with a .SUB extension for customization is performed as follows:

1. Start ConfigPost for Turn-Mill and choose File, Open to display the Select File dialog.
2. In "Files of Type:" combo box change type to ConfigPost Files(*.SUB) as shown below:



Look in: Pm-tm

Files of type: ConfigPost Files (*.SUB)

3. To load the post configuration for sub-spindle click on the **Tm-demo.sub** file and then click on Open button.
4. Notice that the only major difference between **Tm-demo.pst** and **Tm-demo.sub** is in the Process Start formats found under both the Configure Mill and Configure Turn menus. When you view Program Start for *.SUB files you might see the following warning:

Program Start

DO NOT START PROGRAMS ON THE SUB SPINDLE WITH THIS POST PROCESSOR<EOB>

Note: *When working with post processors which consist of both *.PST and *.SUB components in ConfigPost, you can launch two instances of ConfigPost simultaneously in order two work with both main and sub spindle configuration files at once.*

Synchronization Modes

These formats are only called when the software Procedure **<P28-SET-ALL-MODES>** is called in one of the Process Headers (Program Start, Tool Change, Process Start) and Process End. Once called, PartMaker jumps to the appropriate Mode Format depending on the **Mode (M1S0, M1S1, M2S0, etc.)** set for the process.

Modes, Moving Stock: Only used in Swiss applications. Modes in this format are called when the process is set to “**Move Stock in Z**” in the Stock Motion dialog.

Modes, Stationary Stock: Used in Swiss and Turn-Mill applications. Modes in this format are called when the process is set to **“Move Tool in Z**” in the Stock Motion dialog.

Below, illustrates an example of how the layout of a Mode header should look. Notice that the string word **<header>** is used to check which Process Header the **<P28-SET-ALL-MODES>** Procedure is being called from.

```
;=====================<eob>
<IF><eq-string>(<header>,PGS)<THEN>
PROGRAM START DATA HERE<eob>
<ENDIF>
;=====================<eob>
<IF><eq-string>(<header>,TCH)<THEN>
TOOL CHANGE DATA HERE<eob>
<ENDIF>
;=====================<eob>
<IF><eq-string>(<header>,PRS)<THEN>
PROCESS START DATA HERE<eob>
<ENDIF>
;=====================<eob>
<IF><eq-string>(<header>,PRE)<THEN>
PROCESS END DATA HERE<eob>
<ENDIF>
```

Appendix A: Using ‘Config-by-Click’® in PartMaker

Introduction

This appendix explains how to use PartMaker’s ‘Config-by-Click’® feature to quickly customize the NC programs produced by various post processors being used with your PartMaker Software. Post processor customization by you, the user, assures that your own shop specific programming preferences will be reflected in your post processor output. The ‘Config-by-Click’® feature allows you to make virtually all post processor modifications quickly and easily, making post customization very productive, even for users with minimal or no post processor customization experience.

In PartMaker, the CNC machine specific program structure is described using the ConfigPost application and its description is stored in a file that the ConfigPost application produces. Such a file is called "Post Processor Configuration File",usually abbreviated as “Post Config file", "Post Processor", or most commonly, to "Post File".

This appendix applies to PartMaker Versions 7.5 and higher. PartMaker’s ‘Config-by-Click’® feature is available for use in the PartMaker Mill, Turn, Turn-Mill, SwissCAM and Wire EDM modules for those users who have licensed ConfigPost Lite or ConfigPost Pro.

The ‘Config-by-Click’® feature assists you in customizing the NC output of a PartMaker Post Config file by automatically placing the cursor in the correct program format in PartMaker’s ConfigPost application. Customizing your Post Config file’s output with the ‘Config-by-Click’® feature can be achieved by clicking on the line of NC code that you wish to modify in PartMaker’s NC Viewer after choosing Generate NC Program from the Job Optimizer Menu.

This appendix describes the sequence of steps required to successfully use the ‘Config-by-Click’® feature.

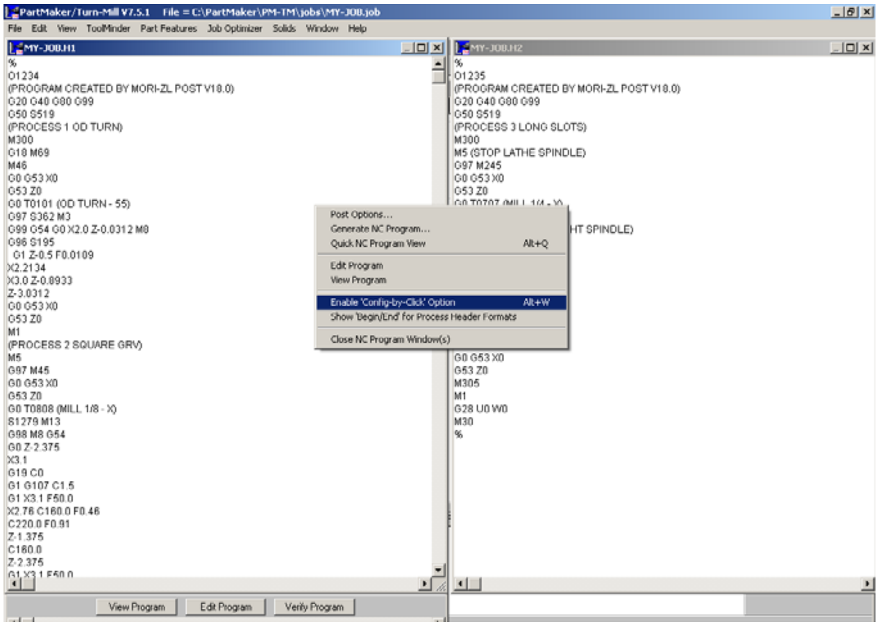
Important! *Readers of this appendix should already have a basic knowledge of the PartMaker ConfigPost application.*

Using ‘Config-by-Click’®

1 Enabling ‘Config-by-Click’®:

Before you can use PartMaker’s ‘Config-by-Click’® feature, it first must be enabled. You can enable this feature using two different methods:

a While in PartMaker’s NC Viewer, right click in the window and select the **Enable ‘Config-by-Click’**® Option from the drop down menu as shown in Figure 1.


PartMaker/Turn-Mill V7.5.1 File = C:\PartMaker\PM-TM\jobs\MY-JOB.job
File Edit View ToolMinder Part Features Job Optimizer Solids Window Help
MY-JOB.H1
MY-JOB.H2
Post Options…
Generate NC Program…
Quick NC Program View Alt+Q
Edit Program
View Program
Enable 'Config-by-Click' Option Alt+W
Show 'Begin/End' for Process Header Formats
Close NC Program Window(s)
View Program
Edit Program
Verify Program

```
%
O1234
(PROGRAM CREATED BY MORI-ZL POST V18.0)
G20 G40 G80 G99
G50 S519
(PROCESS 1 OD TURN)
M300
G18 M69
M46
G0 G53 X0
G53 Z0
G0 T0101 (OD TURN - 55)
G97 S362 M3
G99 G54 G0 X2.0 Z-0.0312 M8
G96 S195
 G1 Z-0.5 F0.0109
X2.2134
X3.0 Z-0.8933
Z-3.0312
G0 G53 X0
G53 Z0
M1
(PROCESS 2 SQUARE GRV)
M5
G97 M45
G0 G53 X0
G53 Z0
G0 T0808 (MILL 1/8 - X)
S1279 M13
G98 M8 G54
G0 Z-2.375
X3.1
G19 C0
G1 G107 C1.5
G1 X3.1 F50.0
X2.76 C160.0 F0.46
C220.0 F0.91
Z-1.375
C160.0
Z-2.375
G1 X3.1 F50.0
```

```
%
O1235
(PROGRAM CREATED BY MORI-ZL POST V18.0)
G20 G40 G80 G99
G50 S519
(PROCESS 3 LONG SLOTS)
M300
M5 (STOP LATHE SPINDLE)
G97 M245
G0 G53 X0
G53 Z0
G0 T0707 (MILL 1/4 - X)
HT SPINDLE)
G0 G53 X0
G53 Z0
M305
M1
G28 U0 W0
M30
%
```

Figure 1: PartMaker’s NC Viewer

b From Job Optimizer, select Configure Post Processor and choose the Enable ‘Config-by-Click’® Option as shown in Figure 2.

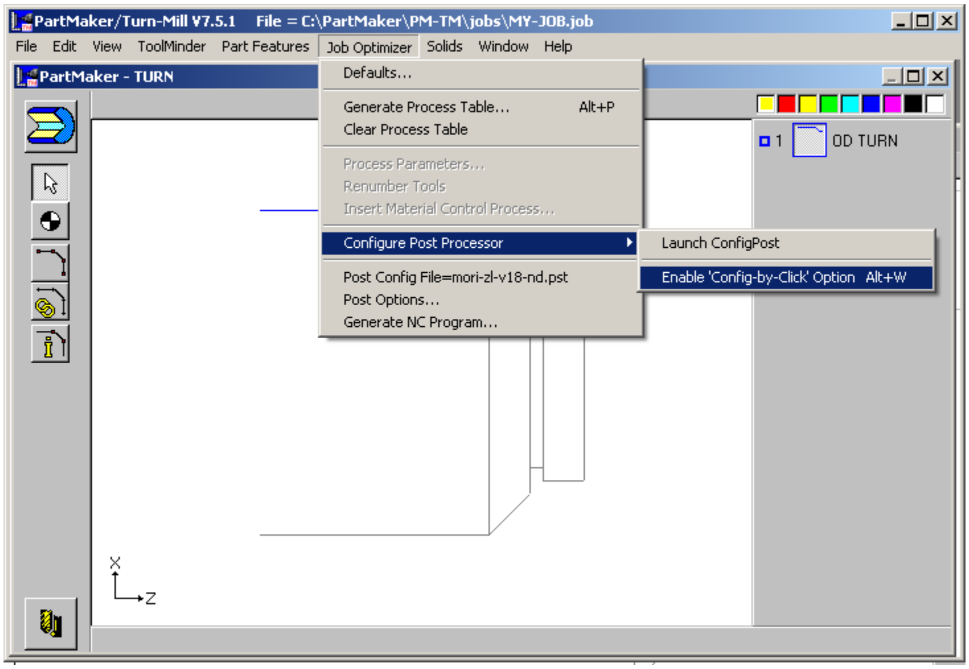

PartMaker/Turn-Mill V7.5.1 File = C:\PartMaker\PM-TM\jobs\MY-JOB.job
File Edit View ToolMinder Part Features Job Optimizer Solids Window Help
PartMaker - TURN
Defaults…
Generate Process Table… Alt+P
Clear Process Table
Process Parameters…
Renumber Tools
Insert Material Control Process…
Configure Post Processor
Post Config File=mori-zl-v18-nd.pst
Post Options…
Generate NC Program…
Launch ConfigPost
Enable 'Config-by-Click' Option Alt+W
1 OD TURN
X
Z

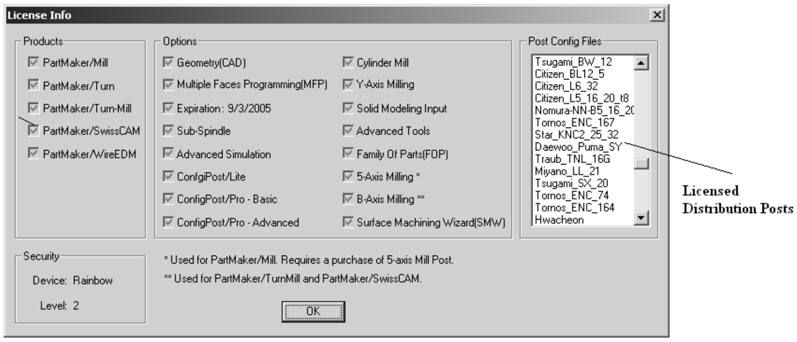
Figure 2: Enabling ‘Config-by-Click’® from the Menu bar

2 Creating a non-distribution Post File:

Note: *This step may only be necessary for PartMaker Turn-Mill and Swiss CAM and some PartMaker Mill users. If you are customizing a library post processor for the PartMaker Mill, Turn or Wire EDM applications, please proceed to* ***Using the "Auto-Reload Post Config File" Feature****.*

Once you have enabled the 'Config-by-Click'® feature, it may be necessary to create a Non-Distribution Post File before proceeding with the customization process. All Post Processors initially supplied with the PartMaker Turn-Mill and SwissCAM modules are referred to as "Distribution Post Files." Distribution Post Files are post processors that are distributed to the PartMaker user community at large and maintained directly by PartMaker Inc.'s Applications Department. Distribution Post Files can only be modified by PartMaker Inc. Once you have created a Non-Distribution Post File, you can modify that post file as often as you wish going forward.

The Distribution Post Files that you have licensed can be viewed by choosing **License Info** from the **Help** menu as shown in Figure 3 below.



License Info

Products
- ☑ PartMaker/Mill
- ☑ PartMaker/Turn
- ☑ PartMaker/Turn-Mill
- ☑ PartMaker/SwissCAM
- ☑ PartMaker/WireEDM

Options
- ☑ Geometry(CAD)
- ☑ Multiple Faces Programming(MFP)
- ☑ Expiration : 9/3/2005
- ☑ Sub-Spindle
- ☑ Advanced Simulation
- ☑ ConfgiPost/Lite
- ☑ ConfigPost/Pro - Basic
- ☑ ConfigPost/Pro - Advanced
- ☑ Cylinder Mill
- ☑ Y-Axis Milling
- ☑ Solid Modeling Input
- ☑ Advanced Tools
- ☑ Family Of Parts(FOP)
- ☑ 5-Axis Milling *
- ☑ B-Axis Milling **
- ☑ Surface Machining Wizard(SMW)

Post Config Files
- Tsugami_BW_12
- Citizen_BL12_5
- Citizen_L6_32
- Citizen_L5_16_20_t8
- Nomura-NN-B5_16_20
- Tornos_ENC_167
- Star_KNC2_25_32
- Daewoo_Puma_SY
- Traub_TNL_16G
- Miyano_LL_21
- Tsugami_SX_20
- Tornos_ENC_74
- Tornos_ENC_164
- Hwacheon

Security

Device: Rainbow

Level: 2

* Used for PartMaker/Mill. Requires a purchase of 5-axis Mill Post.

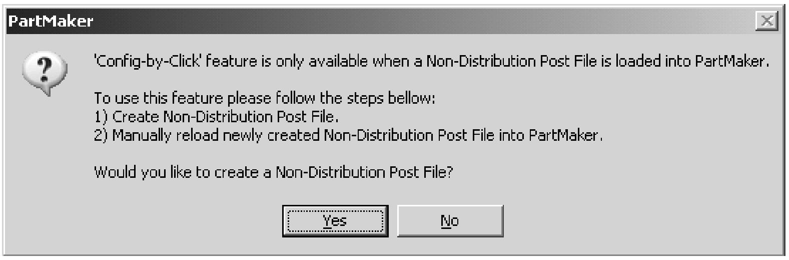
** Used for PartMaker/TurnMill and PartMaker/SwissCAM.

Licensed Distribution Posts

Figure 3: License Info Dialog

When attempting to modify a Distribution Post File, PartMaker will prompt you with the dialog box shown in Figure 4 below. If the dialog box below does not appear, the post file that you are modifying is already a Non-Distribution Post File. If this is the case, you can proceed to **Using the 'Auto-Reload Post Config File Feature.**

Click <**Yes>** in the dialog to open the ConfigPost Application in which you can create a non-distribution post file.



PartMaker

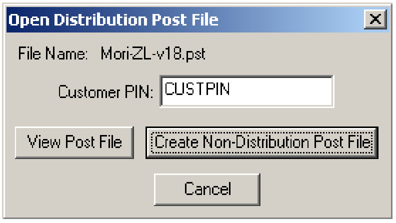
'Config-by-Click' feature is only available when a Non-Distribution Post File is loaded into PartMaker.

To use this feature please follow the steps bellow:
1) Create Non-Distribution Post File.
2) Manually reload newly created Non-Distribution Post File into PartMaker.

Would you like to create a Non-Distribution Post File?

Figure 4: Warning in PartMaker that a Distribution Post File is currently loaded

PartMaker's ConfigPost Application will open and prompt you with the dialog shown in Figure 5 below.



Open Distribution Post File

File Name: Mori-ZL-v18.pst

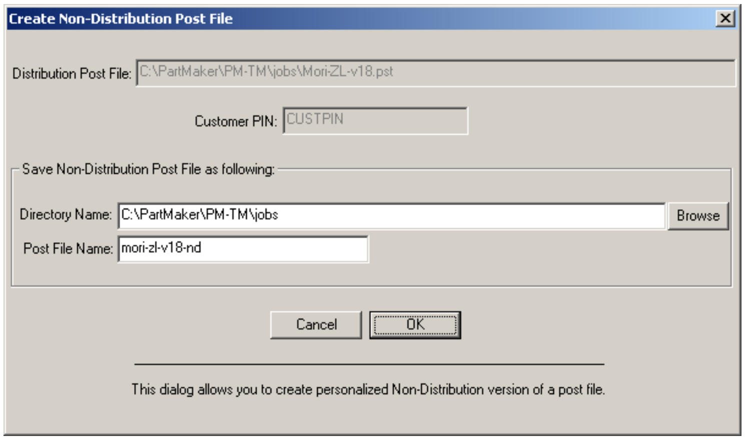
Customer PIN: CUSTPIN

Figure 5: Dialog to create a non-distribution post file

Enter your customer PIN provided by PartMaker Inc., to allow the software to keep any changes previously made using that PIN by PartMaker Inc. Click <**Create Non-Distribution Post File>** to create new post files that you can customize. This will create a new *.PST file, as well as a *.SUB file if applicable.

Note: *If you are not sure of your customer PIN, please contact PartMaker's customer support (support@partmaker.com)*

The dialog box shown in Figure 6 lets you specify a directory in which to save the new post files. The post files may also be renamed in this dialog. Click <**OK>** to finalize creating a non-distribution post file.


Create Non-Distribution Post File

Distribution Post File: C:\PartMaker\PM-TM\jobs\Mori-ZL-v18.pst

Customer PIN: CUSTPIN

Save Non-Distribution Post File as following:

Directory Name: C:\PartMaker\PM-TM\jobs

Post File Name: mori-zl-v18-nd

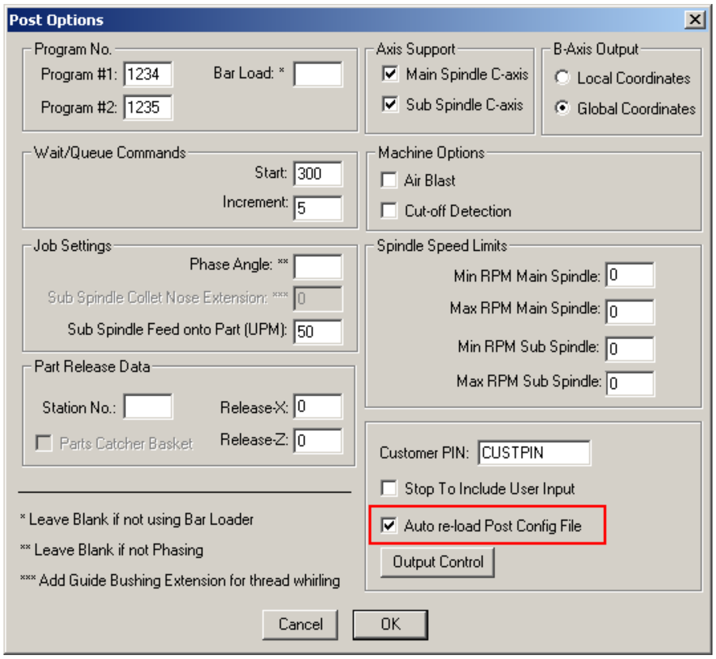
This dialog allows you to create personalized Non-Distribution version of a post file.

Figure 6: Create Non-Distribution Post File Dialog

3 Using the "Auto-Reload Post Config File" Feature

The "Auto-Reload Post Config File" Feature allows PartMaker to automatically reload the post file each time you select **Generate NC Program**. This feature saves you time by not making you choose **Post Config File**= from the **Job Optimizer Menu,** and reloading the post file that you have customized.

To automatically reload the post file when regenerating the NC program, choose **Post Options** from the **Job Optimizer** menu. In the **Post Options** dialog check the **Auto-Reload Post Config File** box as shown in Figure 7 below.


Post Options

Program No.
Program #1: 1234
Program #2: 1235
Bar Load: *

Axis Support
☑ Main Spindle C-axis
☑ Sub Spindle C-axis

B-Axis Output
○ Local Coordinates
● Global Coordinates

Wait/Queue Commands
Start: 300
Increment: 5

Machine Options
☐ Air Blast
☐ Cut-off Detection

Job Settings
Phase Angle: **
Sub Spindle Collet Nose Extension: *** 0
Sub Spindle Feed onto Part (UPM): 50

Spindle Speed Limits
Min RPM Main Spindle: 0
Max RPM Main Spindle: 0
Min RPM Sub Spindle: 0
Max RPM Sub Spindle: 0

Part Release Data
Station No.:
Release-X: 0
Release-Z: 0
☐ Parts Catcher Basket

Customer PIN: CUSTPIN
☐ Stop To Include User Input
☑ Auto re-load Post Config File

* Leave Blank if not using Bar Loader
** Leave Blank if not Phasing
*** Add Guide Bushing Extension for thread whirling

Figure 7: Checking the Auto re-load Post Config File box

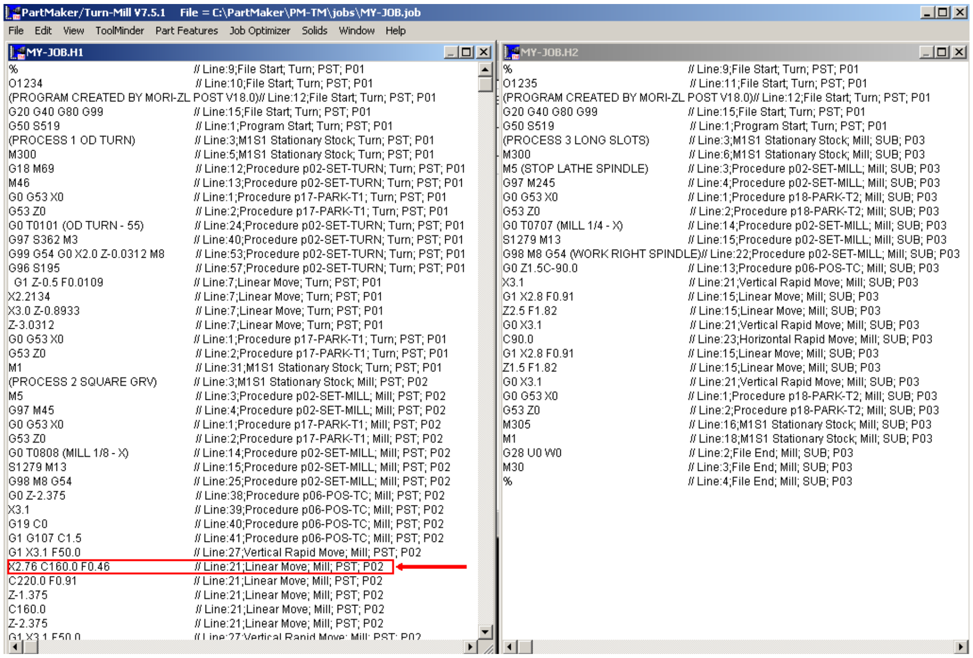
Using the **Auto-Reload Post Config File** feature is not required to successfully use the 'Click-to-Config'® feature, but can speed up the process of customizing a post processor. If you choose not to use this feature, you will have to manually reload the post file to view the customized post's output after changes are made.

Note: *The "Auto-Reload Post Config File" will only reload the post that is currently loaded in* ***Job Optimizer*** *under the* ***Post Config File****=. If you save the post file by a different file name in the ConfigPost Application, then you will have to reload the post file manually.*

4 Using 'Config-by-Click'® to modify a post file:

If you had to create a Non-Distribution Post file from (**2) Creating a non-distribution Post File**, please load the Non-Distribution Post file at this time and repeat the steps described in (**1) Enabling 'Config-by-Click'** before continuing**.**

Upon loading a post file and enabling the 'Config-by-Click'® feature, PartMaker's NC Viewer will display both the NC code output from the post processor file (left column), as well as the location of the program format in the Post File (right column) as shown in Figure 8.



```
%                                    // Line:9;File Start; Turn; PST; P01
O1234                                // Line:10;File Start; Turn; PST; P01
(PROGRAM CREATED BY MORI-ZL POST V18.0)// Line:12;File Start; Turn; PST; P01
G20 G40 G80 G99                      // Line:15;File Start; Turn; PST; P01
G50 S519                             // Line:1;Program Start; Turn; PST; P01
(PROCESS 1 OD TURN)                  // Line:3;M1S1 Stationary Stock; Turn; PST; P01
M300                                 // Line:5;M1S1 Stationary Stock; Turn; PST; P01
G18 M69                              // Line:12;Procedure p02-SET-TURN; Turn; PST; P01
M46                                  // Line:13;Procedure p02-SET-TURN; Turn; PST; P01
G0 G53 X0                            // Line:1;Procedure p17-PARK-T1; Turn; PST; P01
G53 Z0                               // Line:2;Procedure p17-PARK-T1; Turn; PST; P01
G0 T0101 (OD TURN - 55)              // Line:24;Procedure p02-SET-TURN; Turn; PST; P01
G97 S362 M3                          // Line:40;Procedure p02-SET-TURN; Turn; PST; P01
G99 G54 G0 X2.0 Z-0.0312 M8          // Line:53;Procedure p02-SET-TURN; Turn; PST; P01
G96 S195                             // Line:57;Procedure p02-SET-TURN; Turn; PST; P01
 G1 Z-0.5 F0.0109                    // Line:7;Linear Move; Turn; PST; P01
X2.2134                              // Line:7;Linear Move; Turn; PST; P01
X3.0 Z-0.8933                        // Line:7;Linear Move; Turn; PST; P01
Z-3.0312                             // Line:7;Linear Move; Turn; PST; P01
G0 G53 X0                            // Line:1;Procedure p17-PARK-T1; Turn; PST; P01
G53 Z0                               // Line:2;Procedure p17-PARK-T1; Turn; PST; P01
M1                                   // Line:31;M1S1 Stationary Stock; Turn; PST; P01
(PROCESS 2 SQUARE GRV)               // Line:3;M1S1 Stationary Stock; Mill; PST; P02
M5                                   // Line:3;Procedure p02-SET-MILL; Mill; PST; P02
G97 M45                              // Line:4;Procedure p02-SET-MILL; Mill; PST; P02
G0 G53 X0                            // Line:1;Procedure p17-PARK-T1; Mill; PST; P02
G53 Z0                               // Line:2;Procedure p17-PARK-T1; Mill; PST; P02
G0 T0808 (MILL 1/8 - X)              // Line:14;Procedure p02-SET-MILL; Mill; PST; P02
S1279 M13                            // Line:15;Procedure p02-SET-MILL; Mill; PST; P02
G98 M8 G54                           // Line:25;Procedure p02-SET-MILL; Mill; PST; P02
G0 Z-2.375                           // Line:38;Procedure p06-POS-TC; Mill; PST; P02
X3.1                                 // Line:39;Procedure p06-POS-TC; Mill; PST; P02
G19 C0                               // Line:40;Procedure p06-POS-TC; Mill; PST; P02
G1 G107 C1.5                         // Line:41;Procedure p06-POS-TC; Mill; PST; P02
G1 X3.1 F50.0                        // Line:27;Vertical Rapid Move; Mill; PST; P02
X2.76 C160.0 F0.46                   // Line:21;Linear Move; Mill; PST; P02
C220.0 F0.91                         // Line:21;Linear Move; Mill; PST; P02
Z-1.375                              // Line:21;Linear Move; Mill; PST; P02
C160.0                               // Line:21;Linear Move; Mill; PST; P02
Z-2.375                              // Line:21;Linear Move; Mill; PST; P02
G1 X3.1 F50.0                        // Line:27;Vertical Rapid Move; Mill; PST; P02
```

```
%                                    // Line:9;File Start; Turn; PST; P01
O1235                                // Line:11;File Start; Turn; PST; P01
(PROGRAM CREATED BY MORI-ZL POST V18.0)// Line:12;File Start; Turn; PST; P01
G20 G40 G80 G99                      // Line:15;File Start; Turn; PST; P01
G50 S519                             // Line:1;Program Start; Turn; PST; P01
(PROCESS 3 LONG SLOTS)               // Line:3;M1S1 Stationary Stock; Mill; SUB; P03
M300                                 // Line:6;M1S1 Stationary Stock; Mill; SUB; P03
M5 (STOP LATHE SPINDLE)              // Line:3;Procedure p02-SET-MILL; Mill; SUB; P03
G97 M245                             // Line:4;Procedure p02-SET-MILL; Mill; SUB; P03
G0 G53 X0                            // Line:1;Procedure p18-PARK-T2; Mill; SUB; P03
G53 Z0                               // Line:2;Procedure p18-PARK-T2; Mill; SUB; P03
G0 T0707 (MILL 1/4 - X)              // Line:14;Procedure p02-SET-MILL; Mill; SUB; P03
S1279 M13                            // Line:15;Procedure p02-SET-MILL; Mill; SUB; P03
G98 M8 G54 (WORK RIGHT SPINDLE)// Line:22;Procedure p02-SET-MILL; Mill; SUB; P03
G0 Z1.5C-90.0                        // Line:13;Procedure p06-POS-TC; Mill; SUB; P03
X3.1                                 // Line:21;Vertical Rapid Move; Mill; SUB; P03
G1 X2.8 F0.91                        // Line:15;Linear Move; Mill; SUB; P03
Z2.5 F1.82                           // Line:15;Linear Move; Mill; SUB; P03
G0 X3.1                              // Line:21;Vertical Rapid Move; Mill; SUB; P03
C90.0                                // Line:23;Horizontal Rapid Move; Mill; SUB; P03
G1 X2.8 F0.91                        // Line:15;Linear Move; Mill; SUB; P03
Z1.5 F1.82                           // Line:15;Linear Move; Mill; SUB; P03
G0 X3.1                              // Line:21;Vertical Rapid Move; Mill; SUB; P03
G0 G53 X0                            // Line:1;Procedure p18-PARK-T2; Mill; SUB; P03
G53 Z0                               // Line:2;Procedure p18-PARK-T2; Mill; SUB; P03
M305                                 // Line:16;M1S1 Stationary Stock; Mill; SUB; P03
M1                                   // Line:18;M1S1 Stationary Stock; Mill; SUB; P03
G28 U0 W0                            // Line:2;File End; Mill; SUB; P03
M30                                  // Line:3;File End; Mill; SUB; P03
%                                    // Line:4;File End; Mill; SUB; P03
```

Figure 8: PartMaker's NC Viewer with the 'Config-by-Click'® feature active

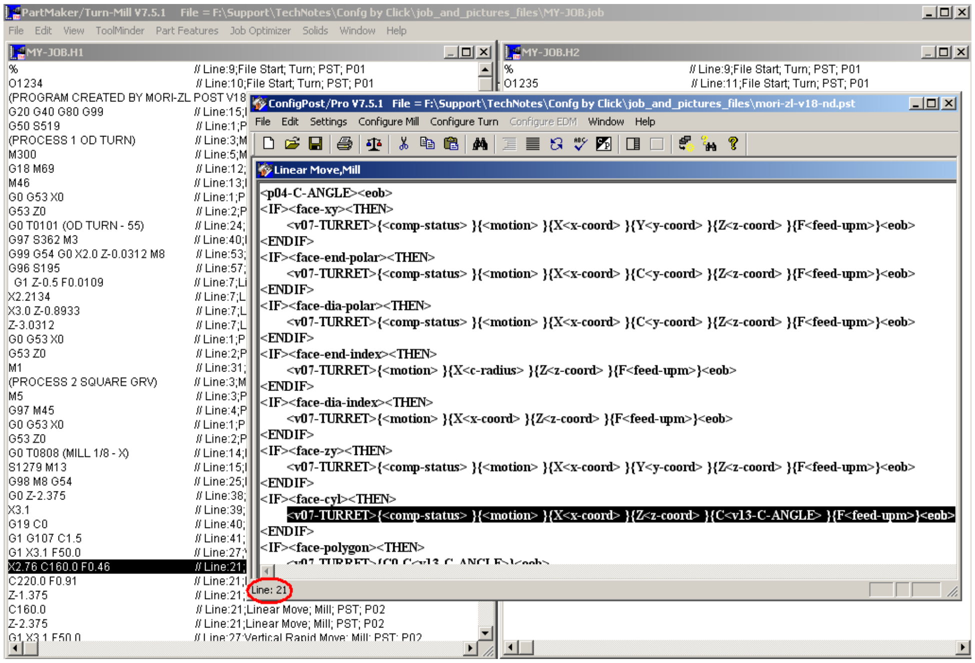
At the end of the file MY-JOB.H1 in Figure 8 there is a line with a rectangle around it:

left column	**right column**
X2.76 C160.0 F0.46	// Line:21;Linear Move; Mill; PST; P02

Information in the right column means that the G-code line in the left column corresponds to the following Post File information:

Line number	**Format Name**	**Mill or Turn**	**PST or SUB**	**Proc ID**
Line: 21	Linear Move	Mill	PST	P02

Double click on this line of code in PartMaker's NC Viewer to have PartMaker automatically highlight the correct program format line in the post file as shown in Figure 9. Here, you may modify the .PST file or .SUB file to achieve desired output. After modifying the post file in ConfigPost, make sure you save the file before attempting to view the changes in PartMaker.


PartMaker/Turn-Mill V7.5.1 File = F:\Support\TechNotes\Confg by Click\job_and_pictures_files\MY-JOB.job
File Edit View ToolMinder Part Features Job Optimizer Solids Window Help
MY-JOB.H1
MY-JOB.H2
ConfigPost/Pro V7.5.1 File = F:\Support\TechNotes\Confg by Click\job_and_pictures_files\mori-zl-v18-nd.pst
File Edit Settings Configure Mill Configure Turn Configure EDM Window Help
Linear Move,Mill
Line: 21

```
%                                  // Line:9;File Start; Turn; PST; P01
O1234                              // Line:10;File Start; Turn; PST; P01
(PROGRAM CREATED BY MORI-ZL POST V18
G20 G40 G80 G99                    // Line:15;I
G50 S519                           // Line:1;P
(PROCESS 1 OD TURN)                // Line:3;M
M300                               // Line:5;M
G18 M69                            // Line:12;
M46                                // Line:13;I
G0 G53 X0                          // Line:1;P
G53 Z0                             // Line:2;P
G0 T0101 (OD TURN - 55)            // Line:24;
G97 S362 M3                        // Line:40;I
G99 G54 G0 X2.0 Z-0.0312 M8        // Line:53;
G96 S195                           // Line:57;
 G1 Z-0.5 F0.0109                  // Line:7;Li
X2.2134                            // Line:7;L
X3.0 Z-0.8933                      // Line:7;L
Z-3.0312                           // Line:7;L
G0 G53 X0                          // Line:1;P
G53 Z0                             // Line:2;P
M1                                 // Line:31;
(PROCESS 2 SQUARE GRV)             // Line:3;M
M5                                 // Line:3;P
G97 M45                            // Line:4;P
G0 G53 X0                          // Line:1;P
G53 Z0                             // Line:2;P
G0 T0808 (MILL 1/8 - X)            // Line:14;I
S1279 M13                          // Line:15;I
G98 M8 G54                         // Line:25;I
G0 Z-2.375                         // Line:38;
X3.1                               // Line:39;
G19 C0                             // Line:40;
G1 G107 C1.5                       // Line:41;
G1 X3.1 F50.0                      // Line:27;
X2.76 C160.0 F0.46                 // Line:21;
C220.0 F0.91                       // Line:21;I
Z-1.375                            // Line:21;
C160.0                             // Line:21;Linear Move; Mill; PST; P02
Z-2.375                            // Line:21;Linear Move; Mill; PST; P02
G1 X3.1 F50.0                      // Line:27;Vertical Rapid Move; Mill; PST; P02
```

```
%                                  // Line:9;File Start; Turn; PST; P01
O1235                              // Line:11;File Start; Turn; PST; P01
```

```
<p04-C-ANGLE><eob>
<IF><face-xy><THEN>
    <v07-TURRET>{<comp-status> }{<motion> }{X<x-coord> }{Y<y-coord> }{Z<z-coord> }{F<feed-upm>}<eob>
<ENDIF>
<IF><face-end-polar><THEN>
    <v07-TURRET>{<comp-status> }{<motion> }{X<x-coord> }{C<y-coord> }{Z<z-coord> }{F<feed-upm>}<eob>
<ENDIF>
<IF><face-dia-polar><THEN>
    <v07-TURRET>{<comp-status> }{<motion> }{X<x-coord> }{C<y-coord> }{Z<z-coord> }{F<feed-upm>}<eob>
<ENDIF>
<IF><face-end-index><THEN>
    <v07-TURRET>{<motion> }{X<c-radius> }{Z<z-coord> }{F<feed-upm>}<eob>
<ENDIF>
<IF><face-dia-index><THEN>
    <v07-TURRET>{<motion> }{X<x-coord> }{Z<z-coord> }{F<feed-upm>}<eob>
<ENDIF>
<IF><face-zy><THEN>
    <v07-TURRET>{<comp-status> }{<motion> }{X<x-coord> }{Y<y-coord> }{Z<z-coord> }{F<feed-upm>}<eob>
<ENDIF>
<IF><face-cyl><THEN>
    <v07-TURRET>{<comp-status> }{<motion> }{X<x-coord> }{Z<z-coord> }{C<v13-C-ANGLE> }{F<feed-upm>}<eob>
<ENDIF>
<IF><face-polygon><THEN>
    <v07-TURRET>{C0 C<v13-C-ANGLE>}<eob>
```

Figure 9: ConfigPost window highlighting a formatting line #21 as indicated in the lower left corner of the window. A corresponding G-code line is highlighted in the background

The 'Config-by-Click'® feature has been designed to locate the program format in the post processor where the output is formatted. It is advised that after making a change to the post file, you should reload the post file and verify the changes to the output by regenerating the NC program before clicking to modify the post file in a new location. This will prevent any mismatches between the G-code line and the location highlighted in ConfigPost.

Important! *Before generating the final NC program to run on your machine, disable the 'Config-by-Click'® feature following the steps described in **(1) Enable 'Config-by-Click'**. <u>If you do not disable this feature, then the NC program file will have format locations still included in it</u>.*

Note: *'Config-by-Click'® can only be accessed from PartMaker's NC Viewer. It can not be accessed from outside applications such as Notepad, MS Word or third party CNC editors (i.e. WinCNC, Predator etc).*

ASCII Code Table

Below is a table showing the American Standard Code for Information Interchange (ASCII).

These characters can be output in ConfigPost by including their corresponding ASCII number inside < > brackets.

	0	1	2	3	4	5	6	7	8	9
0	\000	\001	\002	\003	\004	\005	\006	\a	\b	\t
10	\n	\v	\f	\r	\016	\017	\020	\021	\022	\023
20	\024	\025	\026	\027	\030	\031	\032	\033	\034	\035
30	\036	\037	*space*	!	“	#	$	%	&	‘
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~	\177		