Autodesk® PartMaker®

User Guide

ConfigPost



Contents

Welcome to ConfigPost help 1

Basic and multi-axis postprocessors 2

Understanding Distribution and Non-Distribution Post files 3

Getting Started with ConfigPost 4

Relevant PartMaker concepts 5
- PartMaker applications 6
- Process Table 6
- Generating NC programs 7
- Faces and machining functions 7
- Overview of synchronization modes 8
- Importance of the cut-off process 9
- Material Control Processes overview 10

Starting ConfigPost from within PartMaker 12
Starting ConfigPost as a standalone application 13
Understanding the ConfigPost screen layout 14

ConfigPost Overview 15

Introduction to postprocessing 16
What you can do with ConfigPost 17
How does ConfigPost work? 18

NC Program Output in PartMaker 21

Milling process output 22
Drilling process output, Rotary tool 23
Drilling process output, Fixed tool 24
Turning process output 25
Handling of EDM processes in PartMaker 26

Material Control Processes (MCPs) 27

NC Program output for MCPs 29
MCPs for PartMaker/Turn and Turn-Mill 30
MCP (Rechuck) for PartMaker/SwissCAM 31

Reserved Words 32

Using reserved words in ConfigPost 32
Numeric reserved words 32
String reserved words 33
Using the Prefix Characters $ and @ 33
Using modality for string and numeric variables 33
Logical and System Reserved Words 34

Program formats 36

Program Format Structure 37
Rules for Process Header Formats 39

Menus 42

File menu 43
New 43
Open 43
Save 43
Save As 43
Print Window 43
Import WRI File 43
Export WRI File 43
Compare Files 44
Recent Files List 44
Exit 45
Edit menu 45
Undo 45
Redo 45
Cut 45
Copy 45
Paste 46
Select All 46
Find 46
Replace 48
Insert 49
Display Control 50
Settings menu 53
Target Application 53
Machine Settings 53
User Data Names 53
Variables Names 53
Procedure Names 53
Advanced Customization 54
Sub-Spindle Info 55
Tools Info 55
Multi Coolant Info 55
Faces Info 55
Channels, Tool Moving in Z 56

Channels, Tool Stationary in Z 56
Internal Data 56
Configure Mill menu 57
NC File Header/Footer Formats 57
Process Header/Footer Formats 58
Motion Formats 59
Subprograms Related Formats 61
Canned Cycle Formats 63
Procedure Formats 65
Modes, Moving Stock 65
Modes, Stationary Stock 65
General Information 65
Preparatory Functions 65
Words Information 66
Configure Turn menu 66
NC File Header/Footer Formats 66
Process Header/Footer Formats 67
Motion Formats 69
Canned Cycles 70
Other Formats 74
Procedure Formats 74
Modes, Moving Stock 74
Modes, Stationary Stock 74
General Information 75
Preparatory Functions 75
Words Information 75
Configure EDM menu 75
NC File Header/Footer Formats 75
Process Header/Footer Formats 76
Motion Formats 77
Procedure Formats 79
General Information 79
Preparatory Functions 79
Words Information 79
Window menu 80
Close All 80
Cascade 80
Tile 80
Help menu 81
Help Topics 81
What's this? 81
About 85

Right-click context menus 86

Right-click context menu on the editor 86
Right-click context menu on the list of the reserved words 88

Dialogs 89

Advanced Customization dialog 89
Channels, Tool Moving in Z dialog 90
Channels, Tool Stationary in Z dialog 91
Comments dialog 92
Compare Files dialog 94
Multi Coolant Info dialog 95
Create Non-Distribution Post File 96
Define Target Application dialog 97
Faces Info dialog 98
General Information dialog 98
    General Information dialog - Mill 98
    General Information dialog - Turn 103
    General Information dialog - WireEDM 105
Insert Word dialog 107
Internal Data dialog 108
Machine Settings dialog 109
Open Distribution Post File 110
Preparatory Functions dialog 111
    Preparatory Functions dialog - Mill 111
    Preparatory Functions dialog - Turn 112
    Preparatory Functions dialog - WireEDM 114
Procedure Names dialog 116
    Procedure Names dialog - Mill 116
    Procedure Names dialog - Turn 116
    Procedure Names dialog - WireEDM 117
Post File Differences dialog 118
Tools Info dialog 120
User Data Names dialog 121
Variables Names dialog 123
Words Information dialog 124
    Words Information dialog - Mill 124
    Words Information dialog - Turn 126
    Words Information dialog - EDM 127

Advanced Features 129

Assigning Values to Reserved Words 130
Explanation Lines 130
Using ConfigPost's File Handling Features 131
Separating Lines Into Specific Files 133
    Example - How to create a Tool List using the Split/Merge feature 133
Using variables 136
    Assigning names to variables 136
    Assigning values to variables 136
    Inserting variables 137
Using the <include> reserved word 137
    Example – Using the <include> reserved word 137
Performing Math Operations 139
    Using Mathematical Functions 140
    Using Negative Values 140

Using Procedures 141
Assigning Names to Procedures 141
Example - Defining Procedures 142
Calling Procedures 143
Using the <prompt> reserved word 144
Example 1 - Using <prompt> to output text strings 144
Example 2 - Using <prompt> to perform calculations 145
Using User Data - User Conditions and User Parameters 145
Using User Data to insert tool stops at the end of a process 145
Import / Export WRI Files 148
How to Create a WRI File 148
How to Load WRI File 148
Structure of a WRI File 148

ConfigPost for Turn-Mill and SwissCAM 149

Channels Overview 150
Error checking in PartMaker 151
Error checking for modes 151
Error checking for Faces 152
Error checking for tool posts 153
Using ConfigPost for Turn-Mill and SwissCAM 153
How each Process is handled by ConfigPost for Turn-Mill 154
How each Process is handled by ConfigPost for SwissCAM 155
Differentiating Between Turning and Milling Processes 155
Using Different Milling Face Types 156
Programming Milling Processes with X-Oriented Tools 157
Using Motion Formats with X-Oriented Tools 157
Rules for Program Start, Tool Change, and Process Start with X-Oriented Tools 158
Main and Sub-Spindle File Handling 159
Synchronization Modes 160
Configuring a post processor to support imported PowerMill toolpaths 162

PartMaker Post Configuration files 167

Post Library - Milling, Standard Table 168
Post Library - Milling, 4-axis Rotary Table 180
Post Library - Milling, 5-axis Rotary Table 182
Post Library - Milling, Tombstone 184

Post Library - Turning 185

Post Library - WireEDM 187

Using Config-By-Click 188

Enabling Config-by-Click in PartMaker 189
Creating a Non-Distribution Post file for use with Config-by-Click 190
Using Config-by-Click to modify a Post file 191

Automatically reloading customized Post files in PartMaker 194

ASCII Codes 195

Autodesk Legal Notice 196

Welcome to ConfigPost help

Welcome to the ConfigPost help. Use this document to find out how to use ConfigPost.

If you are unfamiliar with using Windows, you can access help about Windows by clicking **Start > Help and Support** in Windows.

Basic and multi-axis postprocessors

Basic postprocessors are used by:

- PartMaker/Mill
- PartMaker/Turn
- PartMaker/WireEDM.

Multi-axis postprocessors are used by:

- PartMaker/Turn-Mill
- PartMaker/SwissCAM.

Understanding Distribution and Non-Distribution Post files

Autodesk provides two types of Post files:

- **Distribution Post files** — These files are distributed and maintained by Autodesk for all PartMaker customers. Only Autodesk can customize Distribution Post files.

  All the Post files supplied with an initial PartMaker/SwissCAM and PartMaker/Turn-Mill installation are Distribution Post files.

- **Non-Distribution Post files** — These files contain customer-specific information and cannot be redistributed to other customers.

  Autodesk may have created a Non-Distribution Post file specifically for your organization or you may be able to save a Distribution Post file as a Non-Distribution Post file, which you can then customize.

When you have created a Non-Distribution Post file, you can modify it as often as you wish using PartMaker's ConfigPost program.

Getting Started with ConfigPost

This section describes how to get started using ConfigPost. It contains details about:

- relevant PartMaker concepts (see page 4).
- launching ConfigPost from within PartMaker (see page 11).
- starting ConfigPost as a standalone application (see page 12).

Relevant PartMaker concepts

This section describes the PartMaker concepts that are important when using ConfigPost:

- PartMaker Applications (see page 5)
- Process Table (see page 6)
- NC Program Generation (see page 6)
- Faces and Machining Functions (see page 7)
- Synchronization Modes Overview (see page 8)
- Importance of the Cut-off Process (see page 8)
- Material Control Processes Overview (see page 9)

PartMaker applications

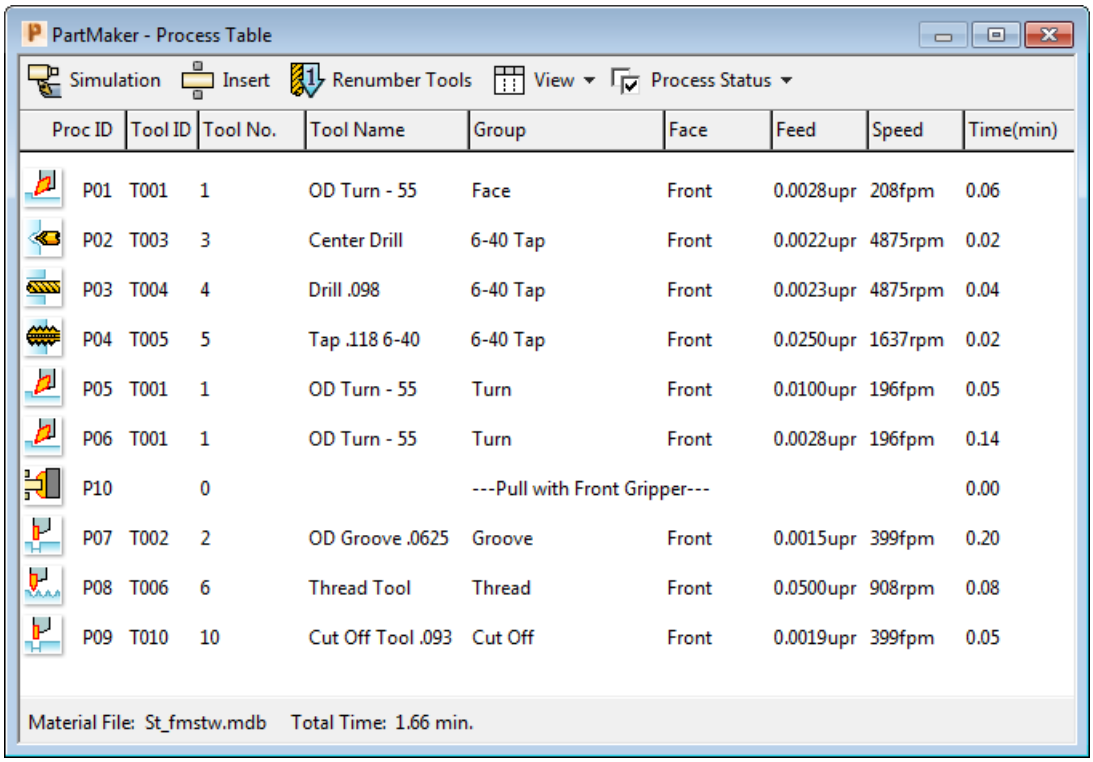
There are five PartMaker applications:

- PartMaker/Mill
- PartMaker/Turn
- PartMaker/WireEDM
- PartMaker/Turn-Mill
- PartMaker/SwissCAM

Process Table

To generate a Process Table in PartMaker, select **Job Optimizer > Generate Process Table** after you have created groups of part features. The Process Table enables you to view all the processes that are required to machine your part.

The following image shows a Process Table for PartMaker/Turn.



Generating NC programs

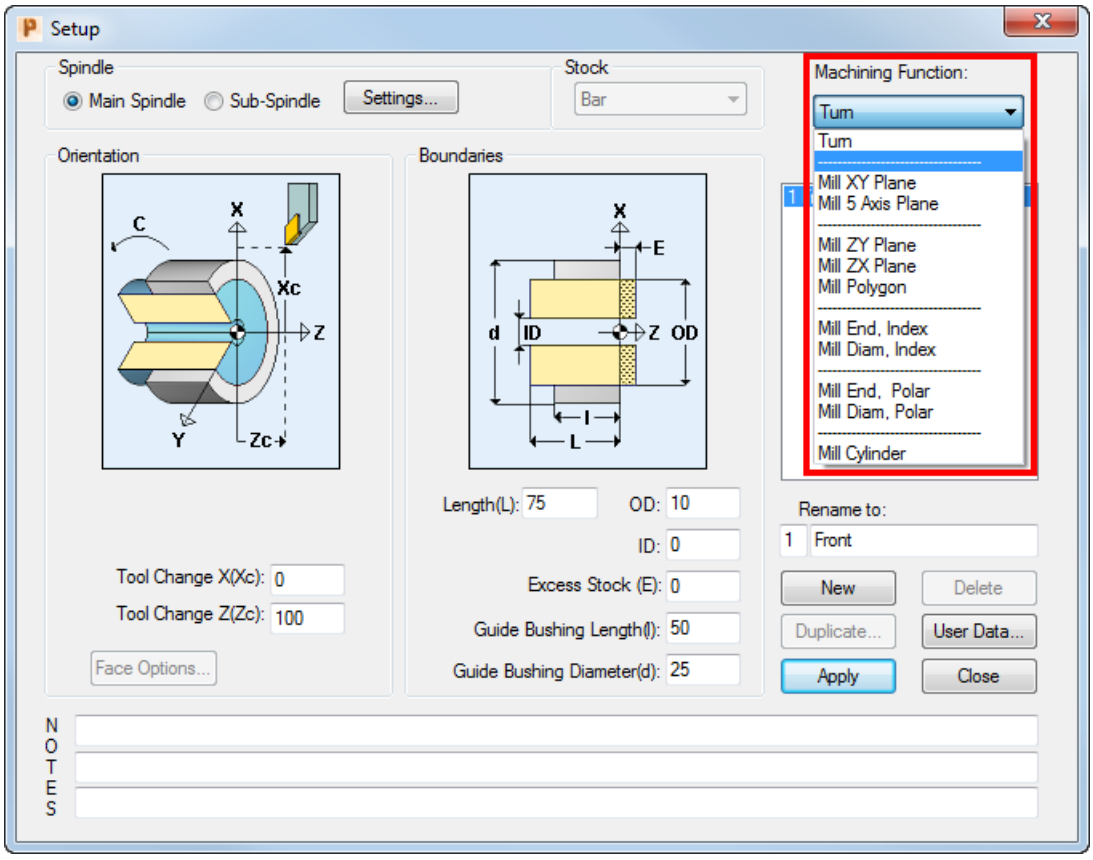
To generate an NC program in PartMaker, select **Job Optimizer > Generate NC Program**.

PartMaker converts the Process Table into CNC machine code, process by process. The process of generating CNC machine code is often called post processing.

Faces and machining functions

All PartMaker applications use the concept of Multiple Faces Programming (MFP).

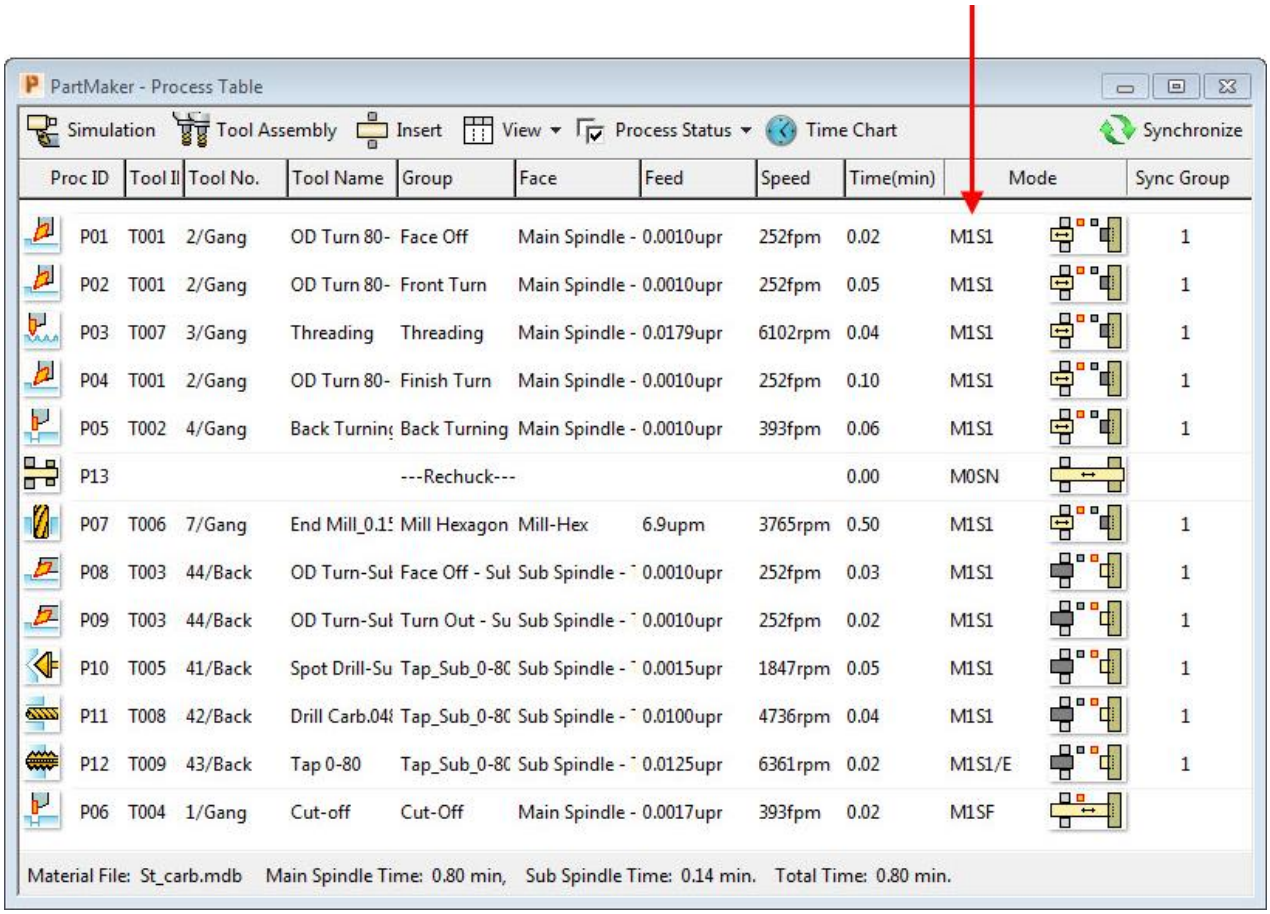
PartMaker/Mill (with the 4-axis and 5-axis option), PartMaker/Turn-Mill, and PartMaker/SwissCAM assign a **Machining Function** to every face as shown below.



The choice of **Machining Function** significantly affects a Post file's structure, as discussed in Using different milling face types (see page 156).

Overview of synchronization modes

The following image shows a Process Table for PartMaker/SwissCAM, which contains synchronization modes.



Synchronization modes are supported only by multi-axis postprocessors.

Using synchronization modes is the most critical aspect of developing postprocessors for PartMaker/Turn-Mill and PartMaker/SwissCAM. To learn how to use synchronization modes correctly, see Synchronization Modes (see page 160).

Importance of the cut-off process

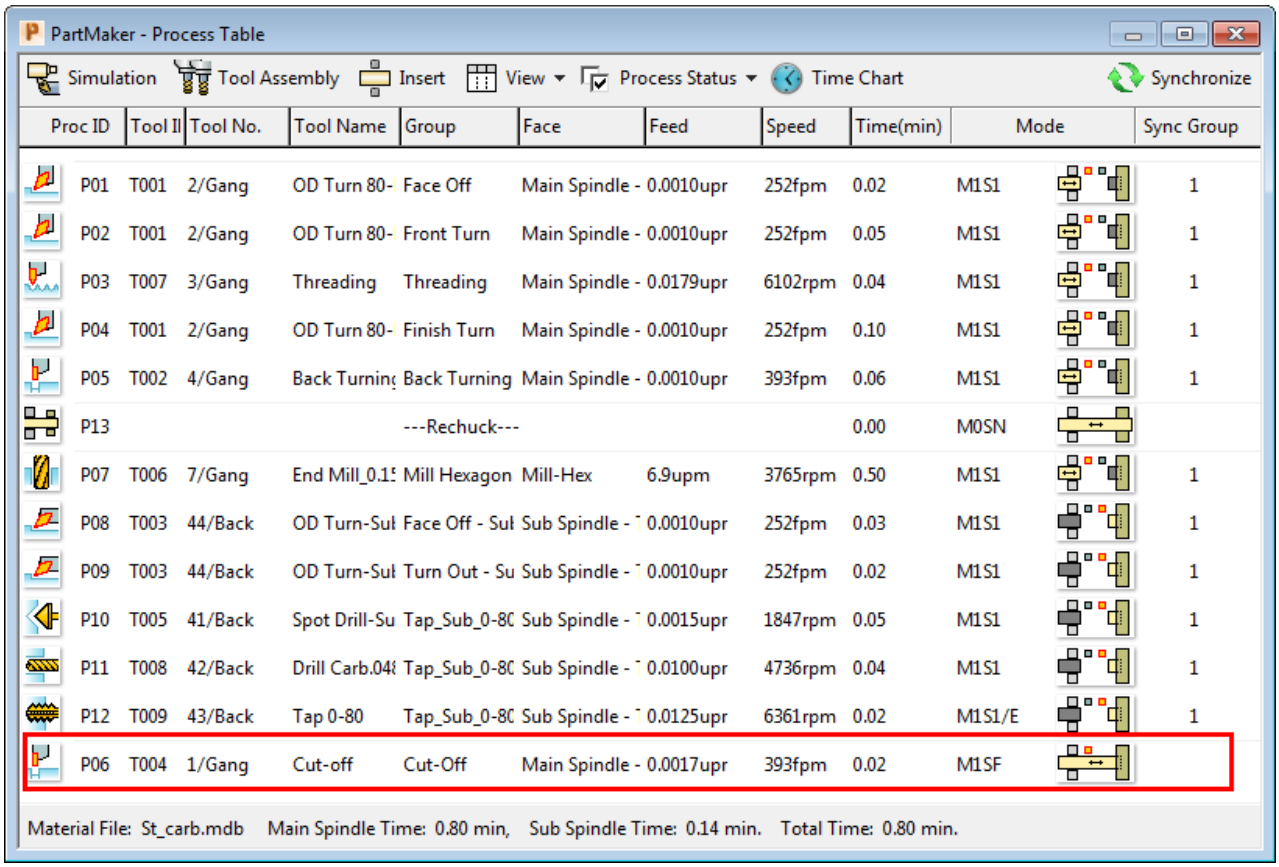
The following information applies only to multi-axis postprocessors for PartMaker/SwissCAM and PartMaker/Turn-Mill.

The cut-off process and the tool used by this process, called the cut-off tool, play a special role during postprocessing for machines equipped with a sub-spindle.

The G and M codes used during the cut-off process for the orientation and transfer of the part vary significantly from one machine to another.

Additionally, in PartMaker/SwissCAM the parameters of the cut-off tool are used in the Collet Re-Chuck format (see page 66) that is called prior to machining.

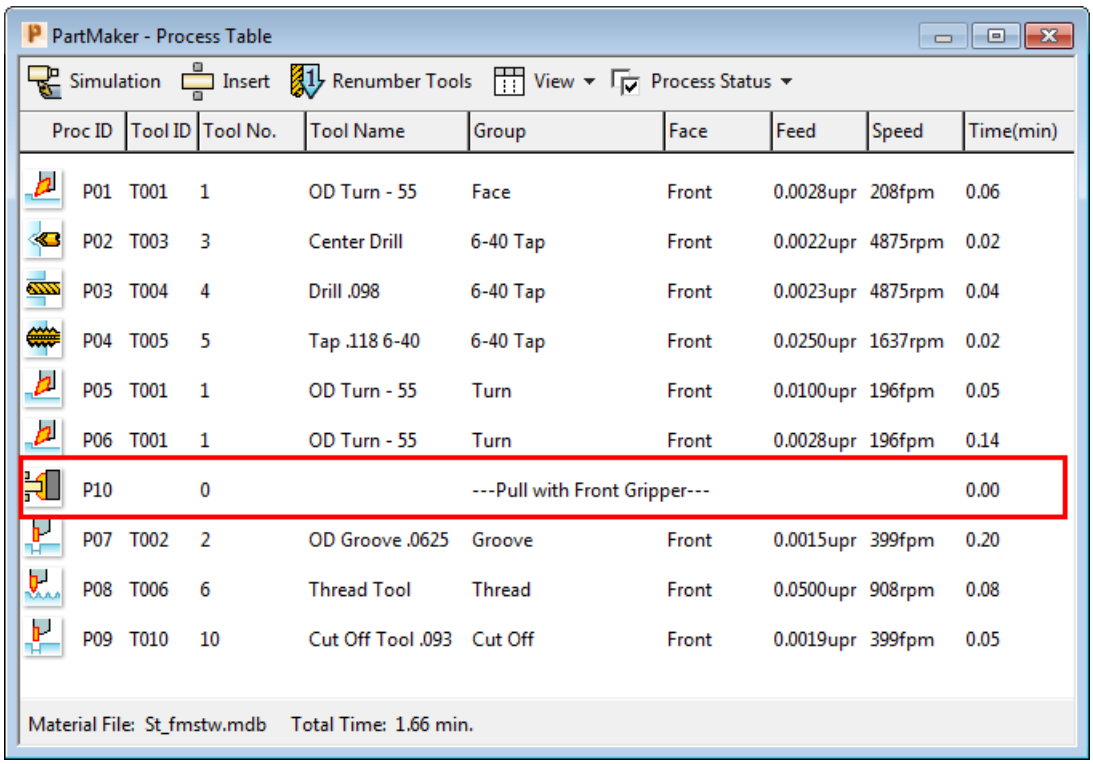
The following example shows how a cut-off process (Process 06) is displayed in the Process Table for PartMaker/SwissCAM.



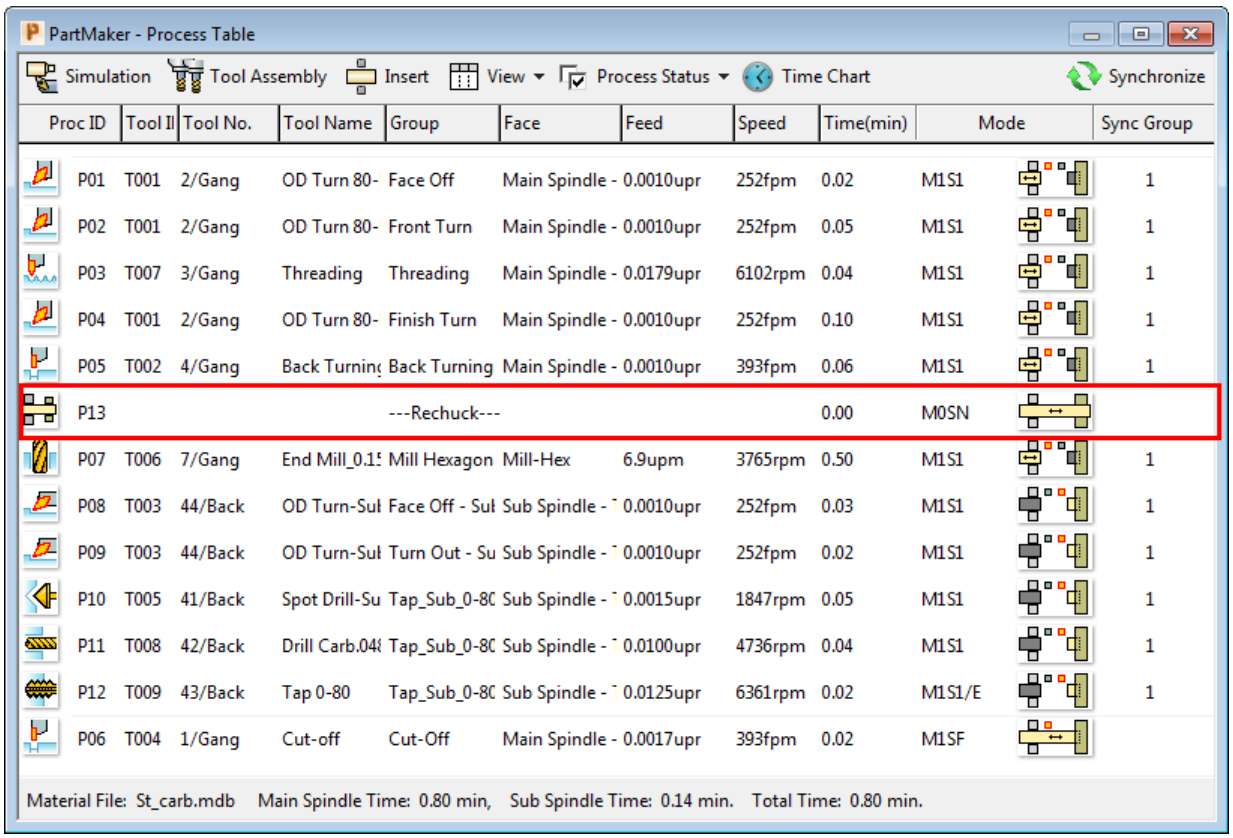
Material Control Processes overview

Material Control Processes (MCPs) are used by PartMaker/Turn, PartMaker/Turn-Mill, and PartMaker/SwissCAM.

This following image shows an example of a Material Control Process for PartMaker/Turn called **Pull with Front Gripper** (Process P10):



This following image shows an example of a Material Control Process for PartMaker/SwissCAM called **Rechuck** (Process P13):



Starting ConfigPost from within PartMaker

To start ConfigPost from within PartMaker, select **Job Optimizer > Configure Post Processor**.

If you choose **Configure Post Processor** from within:

- **PartMaker/Mill,** ConfigPost opens with the Target Application set to PartMaker/Mill.
- **PartMaker/Turn,** ConfigPost opens with the Target Application set to PartMaker/Turn.
- **PartMaker/WireEDM,** ConfigPost opens with Target Application set to PartMaker/WireEDM.
- **PartMaker/Turn-Mill,** ConfigPost opens with the Target Application set to PartMaker/Turn-Mill.
- **PartMaker/SwissCAM,** ConfigPost opens with the Target Application set to PartMaker/SwissCAM.

Starting ConfigPost as a standalone application

If you start ConfigPost as standalone application outside of PartMaker, then the default Target Application (see page 53) is set according to the name of the ConfigPost application as it appears in the \exec subfolder of your PartMaker installation folder.

For example:

conf-m.exe sets the target application to PartMaker/Mill

conf-t.exe sets the target application to PartMaker/Turn

conf-e.exe sets the target application to PartMaker/WireEDM

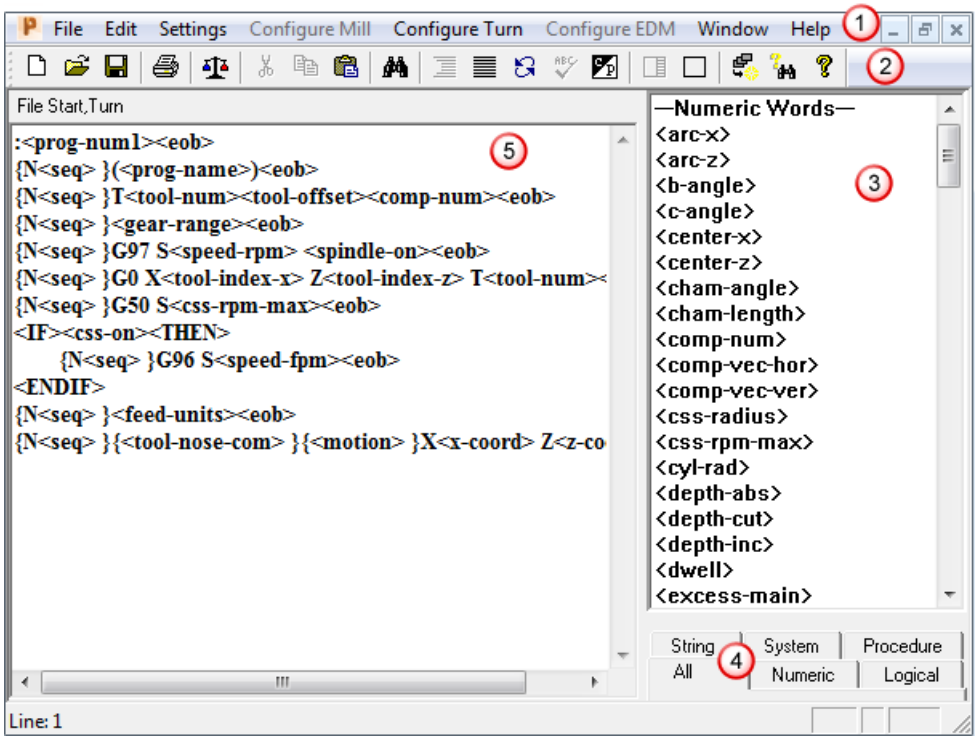
conf-tm.exe sets the target application to PartMaker/Turn-Mill

conf-sw.exe sets the target application to PartMaker/SwissCAM

If the ConfigPost filename is changed and does not match any of the above names, then the Target Application is set to PartMaker/SwissCAM.

Understanding the ConfigPost screen layout

The following image shows the different areas of the screen you use to edit or create a Post file:



1 ConfigPost menus (see page 42).

2 ConfigPost toolbar buttons, which gives quick access to commonly used menu options (see page 42).

3 List of reserved words (see page 32).

4 Tabs to display different types of reserved words (see page 32).

5 Program Format (see page 36) area.

ConfigPost Overview

This section provides an overview of ConfigPost:

- Introduction to post processing (see page 15)
- What you can do with ConfigPost (see page 16)
- How does ConfigPost work? (see page 18)

Introduction to postprocessing

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

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 *postprocessors*.

PartMaker bypasses the intermediate step of generating CL-Data; it generates CNC programs when the you select **Generate NC Program** 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.

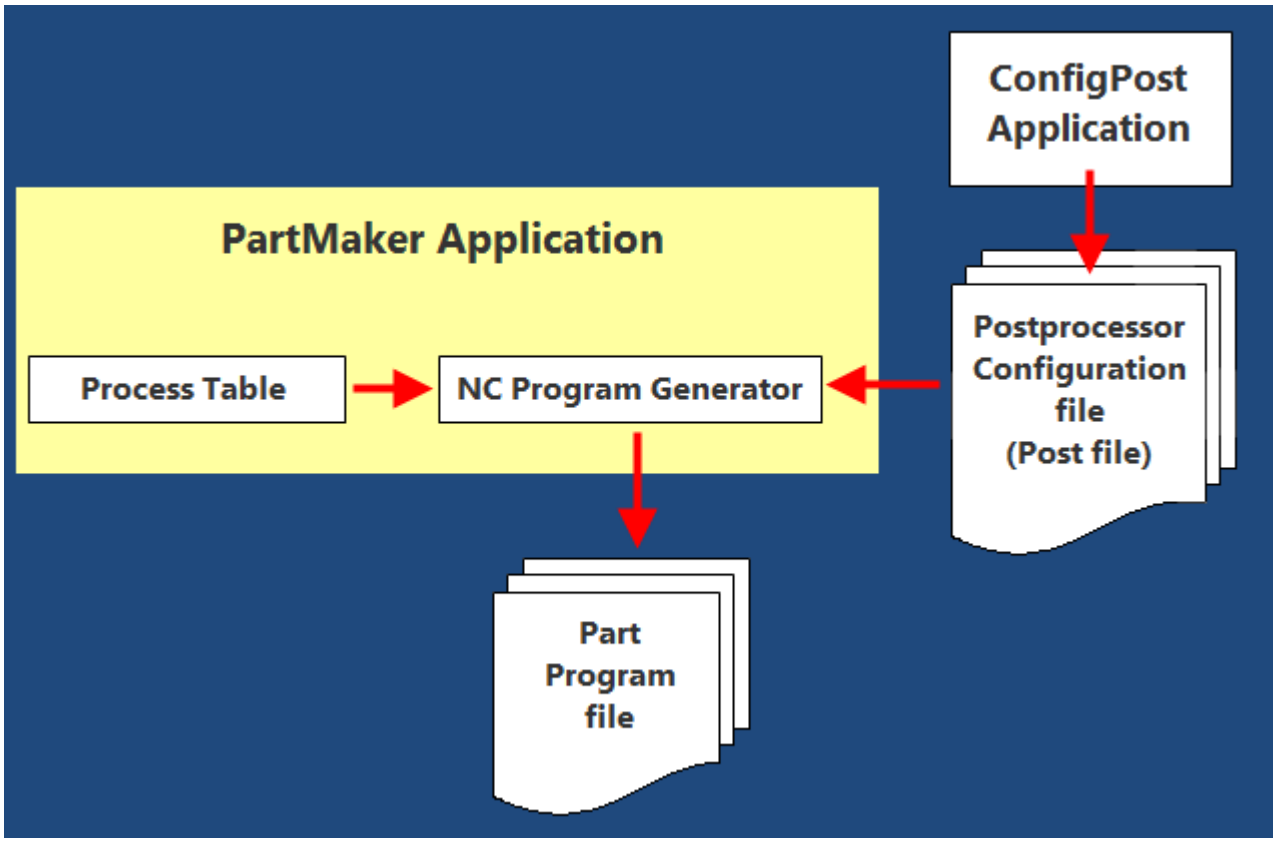
The CNC machine-specific program structure is described using the ConfigPost application and its description is stored in a file that the ConfigPost application produces. This file is called *Post Processor Configuration File*, usually abbreviated as *Post Config file*, *Post Processor*, or most commonly,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

What you can do with ConfigPost

ConfigPost enables you to 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 postprocessor.

Machine tool manufacturers use a variety of CNC program standards, but ConfigPost gives you the freedom to configure a postprocessor for virtually any CNC.

Postprocessor configuration files commonly use .PST as a file extension. For example, FANUC.PST is a postprocessor configuration file for a machine equipped with a Fanuc control.

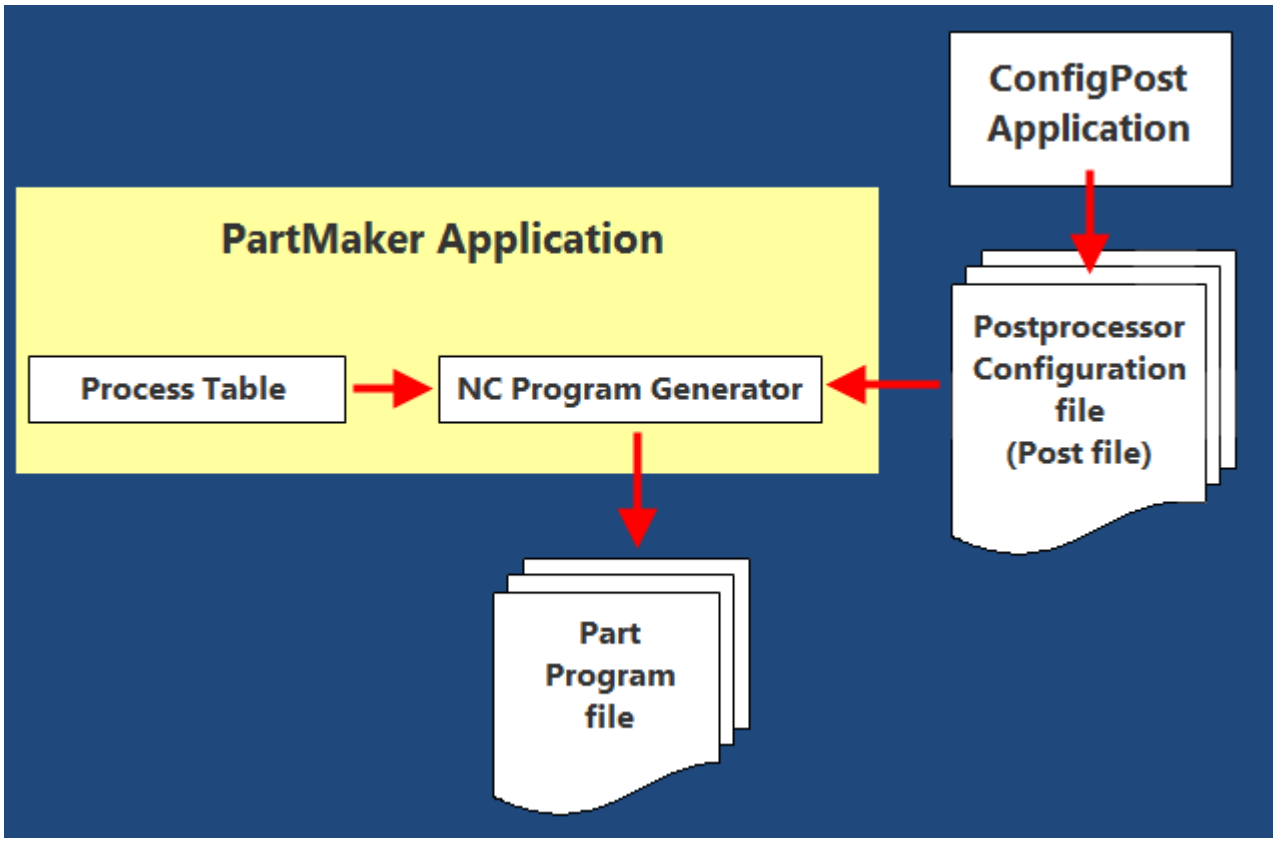


CNC Lathes equipped with a sub-spindle require two Post files: one with the extension .PST and one with the extension .SUB, for example, T42.PST and T42.SUB. When working with ConfigPost you open either a .PST file or a .SUB file. When you are working with PartMaker and select ***Post Config File*** *from PartMaker's* ***Job Optimizer*** *menu, both files are opened at once.*

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.

This image 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 you create in PartMaker. This template consists of program formats (see page 36), such as **Program Start**, **Linear Move**, and **Tool Change**, which 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 (see page 32) 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 (see page 36) are filled when PartMaker generates an 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 (see page 60) you specify in ConfigPost would look like:

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

The table below shows how PartMaker extracts all coordinate data from the Process Table and the format of the line from the Post file, 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>

As shown in the table, 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.



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 the table, <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.

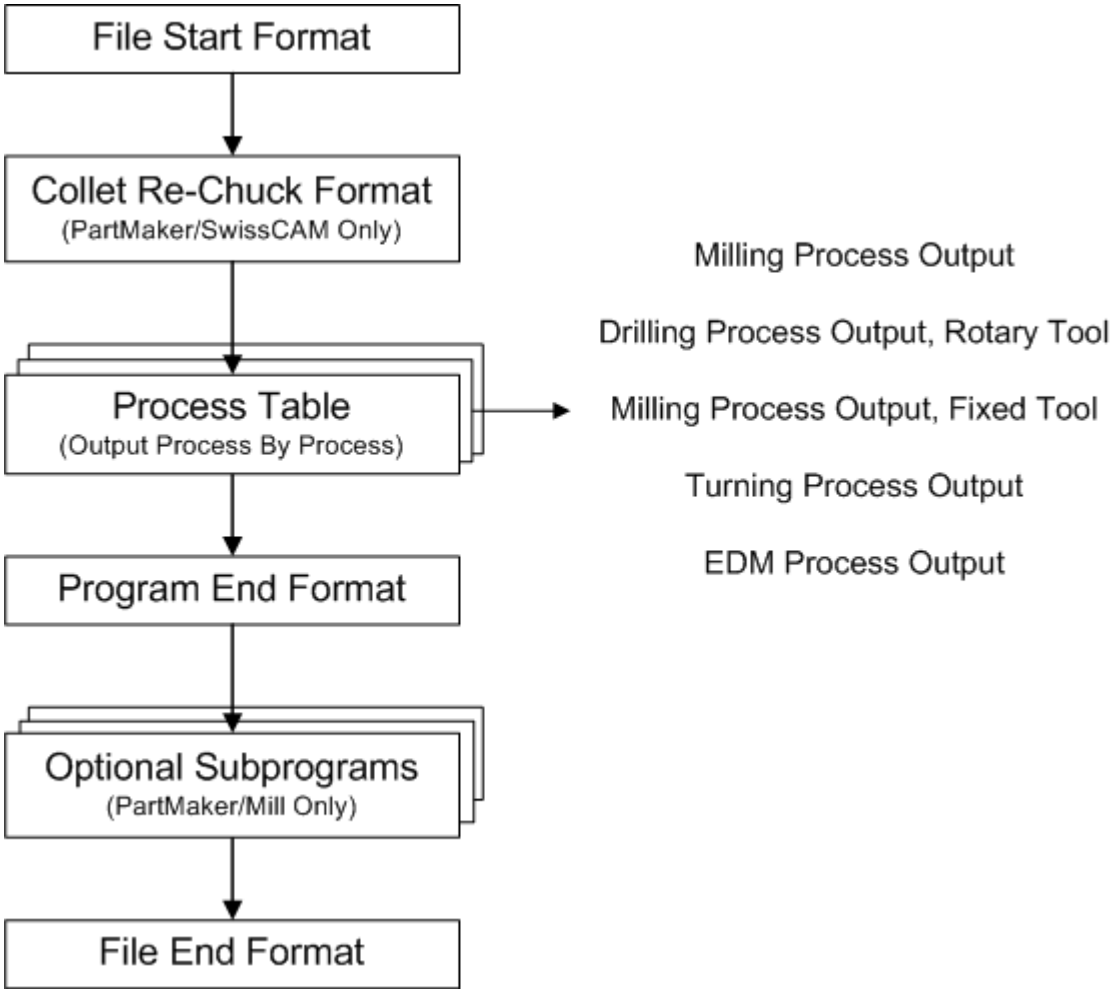
Using modality reduces the overall length of a part program - repeated coordinates and/or commands are automatically stripped from the final part program to prevent repetition of a reserved word whose value is the same as the previous occurrence of that word.



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/WireEDM.*

NC Program Output in PartMaker

The following diagram shows the steps taken by the NC Program:



Milling process output

These steps explain how PartMaker outputs an NC program for a milling process.

The first step is to output a **Process Header Format** (see page 39) as the 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 and 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)

   In PartMaker/Turn-Mill and PartMaker/SwissCAM this Rapid Move Format will be output for the following faces: Mill XY Plane, Mill ZY Plane, Mill End Index, Mill Diameter Index, Mill End Polar, Mill Cylinder. It will not be output for Mill ZX Plane, Mill Diam Polar, Mill Polygon.

3. Motion Formats:

   Linear Move Formats and/or

   Circular Move Formats and/or

   Rapid Move Formats

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

   *Motion Formats will be replaced by a **Subprogram Call Format** if subprograms are enabled in PartMaker/Mill.*

4. Rapid Move Format (Tool moves to the Clearance plane)

   In PartMaker/Turn-Mill and PartMaker/SwissCAM this Rapid Move Format will be output for the following faces: Mill XY Plane, Mill ZY Plane, Mill End Index, Mill Diameter Index, Mill End Polar, Mill Cylinder. It will not be output for Mill ZX Plane, Mill Diam Polar, Mill Polygon.

5 Process End Format.

Drilling process output, Rotary tool

These steps explain how PartMaker outputs an NC program for a drilling process (as well as for any other hole making process such as tapping, reaming, and so on) using a live rotary tool.

The first step is to output a **Process Header Format** as follows:

The **Program Start Format** (see page 39) 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 and brought into position to being 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 Formats

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



***Move to Hole Location Format** will be replaced by a **Subprogram Call Format** if subprograms are enabled in PartMaker/Mill.*

5 Canned Cycle Cancel Format.

6 Process End Format.

Drilling process output, Fixed tool

These steps explain 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 the following:

The **Program Start Format** (see page 39) 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 and 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. Rapid Move Format (Tool moves to the Clearance plane.
3. Canned Cycle Format.

   Canned Cycle Format *can be replaced by* ***Linear Move Formats*** *and* ***Rapid Move Formats*** *if the* ***canned cycle*** *option is not checked in the* ***Process Parameters*** *dialog for a given process.*
4. Process End Format.

Turning process output

These steps explain 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 the following:

The **Program Start Format** (see page 38) 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 and 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 Formats:

   Linear Move Formats and/or

   Circular Move Formats and/or

   Rapid Move Formats and/or

   Thread Move Formats

   Motion Formats are used to output the toolpath information into an NC Program File.
4. Process End



*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

PartMaker outputs an NC program for an EDM process in the Process Table as follows:

1. Process Header Format:

   Program Start Format or

   Process Start Format

   The **Program Start Format** (see page 41) is output if a given process is the first process in the **Process Table**, otherwise the **Process Start Format** is output.

   Process Header Formats contain machine-specific setup information.

2. Motion Formats:

   Linear Move Formats and/or

   Circular Move Formats and/or

   Rapid Move Formats and/or

   XY UV Move Formats (for 4-axis)

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

3. Process End.

Material Control Processes (MCPs)

A **Material Control Process** (MCP) can be used for **PartMaker/Turn**, **PartMaker/Turn-Mill**, and **PartMaker/SwissCAM** applications.

There are two categories of Material Control Processes:

- 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 (see page 33) `<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 (see page 34) `<mat-control-process>` is used to determine if the current process is a **Material Control Process**.

This example 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 MCPs:

Numeric Reserved Words (see page 32) — `<mcp-pull_distance>`, `<mcp-total-pull>`, `<catcher-x>`, `<catcher-z>`

String Reserved Word (see page 33) — `<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.



*To add cumulative pull distance to all Z-coordinates, check **Add Pull Distance To Z-Coordinates** in the **General Info, Turn** dialog (see page 102).*

NC Program output for MCPs

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

Output steps for MCP not using a Turret Tool

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

1 MCP-Action (see page 70).

Output steps for MCP using a Turret Tool

The following steps describe how PartMaker outputs an NC program for a **Material Control Process** that uses a turret tool.

1 Outputs a **Process Header Format** (see page 38) as follows:

Program Start

Tool Change

Process Start

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 and brought into position to begin cutting, the **Tool Change Format** is output, otherwise the **Process Start Format** is output.

2 Outputs an MCP-Action (see page 70) format.

3 Outputs a Process End (see page 69) format.



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.*



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, for example, 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.*

MCPs for PartMaker/Turn and Turn-Mill

Material Control Processes (see page 9) used by **PartMaker/Turn**:

- Bar Feed
- Pull with Side Gripper
- Pull with Front Gripper
- Tailstock Forward
- Tailstock Retract

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

- 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

Example:

An example of an MCP-Action format (see page 70) for HARDINGE_COBRA.pst (located in the \PM-Turn\Postlib folder with your PartMaker User Files) 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

PartMaker/SwissCAM uses only one Material Control Process (MCP), **Pull with Sub-Spindle**, which is called **Rechuck** in PartMaker/SwissCAM.



Do not confuse the Rechuck MCP with Collet Re-Chuck (see page 66).

Example:

An example of MCP-Action format (see page 70) 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>
```

Reserved Words

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.

Using reserved words in ConfigPost

You define reserved words in program formats (for example, a Linear Move) 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.

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 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.)

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.



Never place modality delimiters 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. PartMaker includes or omits certain data from a program depending on whether a conditional statement is true or false.

- 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 1:

```
<IFNOT><CW-SPINDLE><THEN>
M4
<ENDIF>
<IFNOT><Logical reserved word><THEN>
.......
<ENDIF>
```

  Example 2:

```
<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>
```

The following conditional statement 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>
```

Program formats

A Post file created in ConfigPost is used by PartMaker as a template to format the Process Table you create in PartMaker. This template consists of program formats (see page 36), such as **Program Start**, **Linear Move**, and **Tool Change**, which determine the appearance of a part program for a specific CNC.

Program Format Structure

Below is an example of a program format representing 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, for example `X` and `Y`.
- Reserved words such as `<x-coord>` and `<y-coord>`.
- Action Symbols `; @ $ ~ & # [ ] < > ( ) { }` that cause a designated action as listed in the table below.

Action Symbols	Designated Action
`;`	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.

< >	When used around text it represents a Reserved Word. **Example:** <y-coord> When used around a number it represents a character corresponding to this ASCII number. **Example:** <91> represents [.
( )	When used within [ ] it establishes a sequence of expressions evaluation. **Example:** [2*(<x-coord> +5)] *It is recommended to always use ( ) around negative numbers, for example, [(-1)*<x-coord>].* 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.

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 Process Header Formats

Different rules apply for Process Header Formats.

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 (see page 15):

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)`.



*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.



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.

Rules for Program Start, Tool Change, and Process Start 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)`



*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.



*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.*

Rules for Program Start, Tool Change, and Process Start (Turn)

This topic 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.



*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.*

Rules for Program Start and Process Start (EDM)

This topic 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.

Menus

The Menu bar is located at the top of the main window and contains several menus, such as **File** and **Edit**. Click on a menu to display a list of options available from that menu.

Some menu options contain additional submenus, indicated by a small arrow to the right of the text. Click the arrow to display a submenu. For example, clicking the **File** menu, followed by the **Export WRI File** option, displays additional options in a submenu.



Shortcuts to the most frequently used menu options are also available from toolbars.

File menu

Use the options on the **File** menu to open, save, print, and import or export files in ConfigPost.

New

Use this menu option to clear all Post File related information and reset ConfigPost to its startup configuration.

When you select **File > New**, ConfigPost prompts you to save the current Post file.

Open

Use this menu option to display the **Open** dialog, where you can open an existing Post file.

When you select **File > Open**, ConfigPost prompts you to save the Post file if you already have one open.

Save

Use this menu option to save the currently open Post file to the hard drive.

Save As

Use this menu option to display the **Save As** dialog, where you can specify a name and directory for the Post file.

Print Window

Use this menu option to print the contents of the current window.

Import WRI File

Use this menu option to display the **Import WRI File** dialog, where you can open an existing WRI file.

Export WRI File

Use the options on the **Export WRI File** menu to choose how to export program formats to a WRI file:

File > Export WRI File > Complete (see page 44)

File > Export WRI File > Non-Empty Formats Only (see page 44)

Complete

Use this menu option to display the **Export WRI File Complete** dialog, where you can specify a name and directory for the WRI file.

If the open Post file has an extension .pst, ConfigPost automatically appends _pst.wri to the WRI file name.

If the open Post file has an extension .sub, ConfigPost automatically appends _sub.wri to the WRI file name.

This option saves all Program Formats to WRI file, including empty formats.

Before selecting this option, ensure that the Post file is open.

Non-Empty Formats Only

Use this menu option to display the **Export WRI File Non-Empty Formats** dialog, where you can specify a file name and a directory name for a WRI file.

If the open Post file has an extension .pst, ConfigPost automatically appends _pst.wri to the WRI file name.

If the open Post file has an extension .sub, ConfigPost automatically appends _sub.wri to the WRI file name.

This option saves only non-empty Program Formats to a WRI file.

Before selecting this option, ensure that a Post Configuration file is loaded.

Compare Files

Use this menu option to display the **Compare Files** dialog (see page 93), which enables the user to compare two Post files.

This option can also be accessed through the icon:



Recent Files List

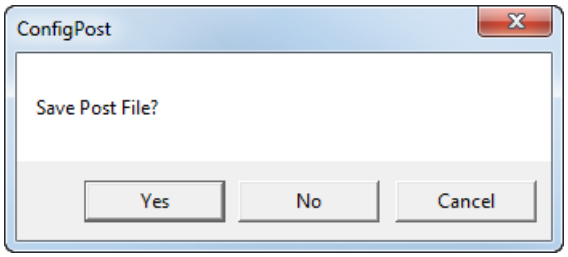
ConfigPost displays up to four recently used Post files.

When you select a file name from the **Recent Files List** you are prompted to save the Post file before this file is opened.

Exit

Use this menu option to exit the ConfigPost application.

When you select **Exit**, you are prompted to save the Post Configuration file.



Click **Yes** to save the current Post file to the hard drive. If the name of the Post file was not defined, the **Save As** dialog is displayed, where you can specify the file name and the directory name for the Post file.

Click **No** to discard changes and close all windows.

Click **Cancel** to return to the current window.

Edit menu

Use the options on the **Edit** menu to manipulate, search for, and control the display of text within ConfigPost.

Undo

Use this menu option to reverse text editing actions in the active window.

This option is available only when a window is active.

Redo

Use this menu option to redo actions you have undone.

This option is available only when a window is active.

Cut

Use this menu option to remove selected text from the active window and place it on the clipboard.

Copy

Use this menu option to copy selected text from the active window onto the clipboard.

Paste

Use this menu option to paste copied text from clipboard to the active window.

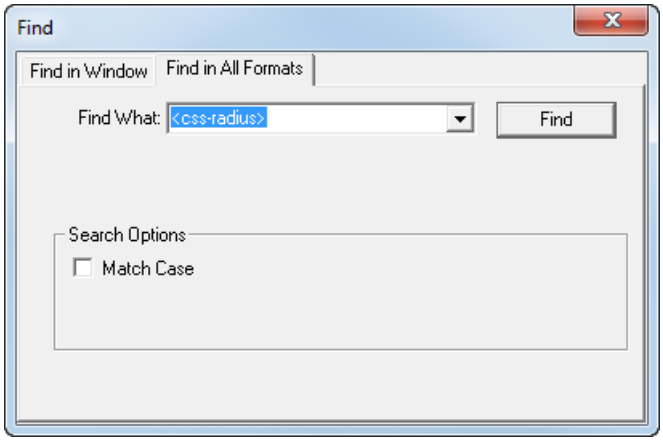
Select All

Use this menu option to select all text in the active window.

Find

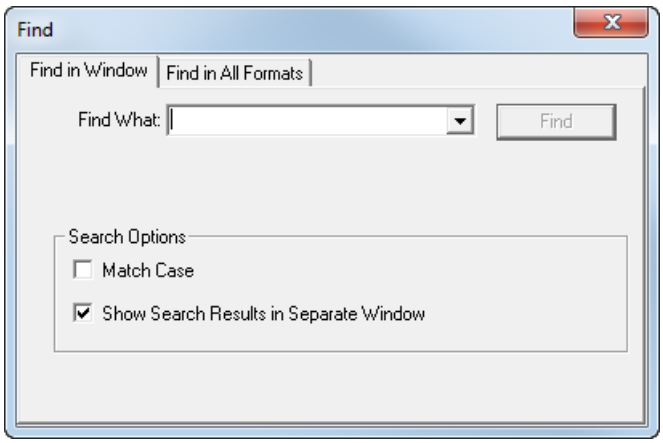
Use this menu option to display the **Find** dialog.

*You can also right-click in **Editor** or in the list of reserved words to access the **Find** dialog.*



Find in Window

Use this tab to search for text in the active window.



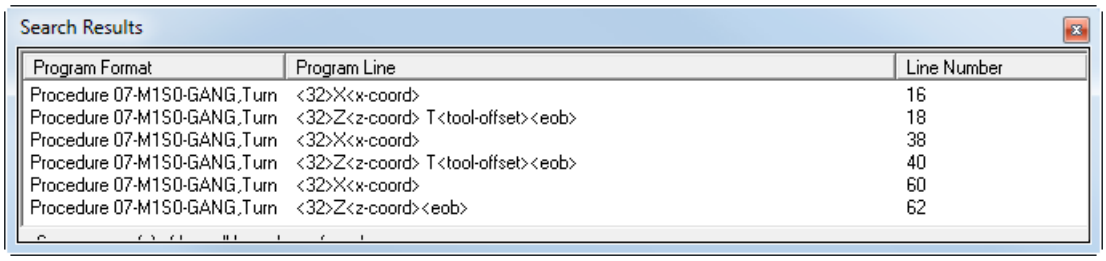
The following settings are available:

Find What — Enter the text you want to find in the Post file. If you select text in the editor window before opening the **Find** dialog, the selected text is displayed in this field.

Match case — Select this option to restrict the search to finding text that matches the combination of upper- and lower-case characters used in the **Find What** field.

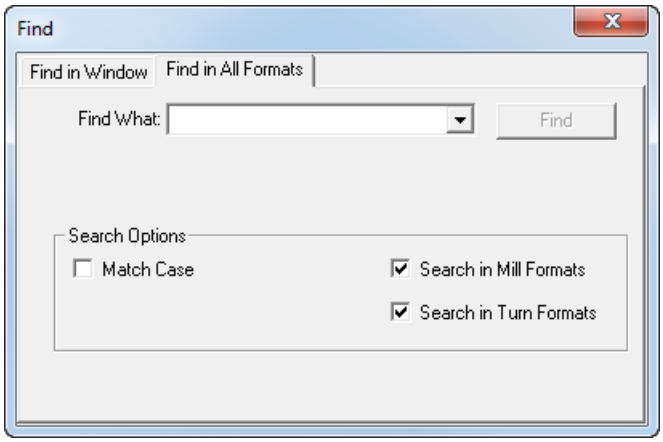
Show Search Results in Separate Window — This option enables you to output search results to the **Search Results** window. If this option is deselected, the matching text is highlighted in the editor window.

The matching result is displayed in **Search Results** dialog:



Find in All Formats

Use this option to perform a search throughout the loaded post processor file.



The following settings are available:

Find What — Enter the text you want to find in the Post file. If you select text in the editor window before opening the **Find** dialog, the selected text is displayed in this field.

Match case — Select this option to restrict the search to finding text that matches the combination of upper- and lower-case characters used in the **Find What** field.

Search in Mill Formats — Select this option to restrict the search to the **Mill Formats** only. **Mill Formats** can be found in the **Configure Mill** menu.

Search in Turn Formats — Select this option to restrict the search to the **Turn Formats** only. **Turn Formats** can be found in the **Configure Turn** menu.

If you double-click on the entry in the **Search Results** dialog the corresponding window will be open and the line where matching Find What string was found will be highlighted.

Right-click the **Search Results** window to display the context menu. The following options are available:

View In Notepad — Use this option to see the results of search in the **Notepad** file named Search_Results.dat.

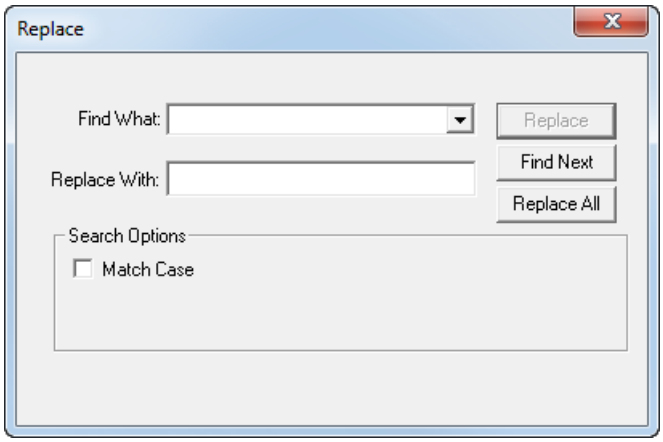
Close Search Results — Use this option to close the **Search Results** window.

Replace

Use this menu option to display the **Replace** dialog.

The **Replace** dialog performs **Search and Replace** operations only in the active window.

*You can also press the **Ctrl+R** keys to display the **Replace** dialog*



Find What — Enter the text you want to find in the active window.

Replace With — Enter the text with which you want to replace the text in the Find What field.

Match case — Select this option to restrict the search to finding text that matches the combination of upper- and lower-case characters used in the **Find What** field.

Find Next — Click this button to find the next occurrence of the text entered in **Find What** field.

Replace — Click this button to replace the currently selected text and find the next occurrence of the text.

Replace All — Click this button to replaces all occurrences of the text entered in the **Find What** field by the text entered in the **Replace With** field.

Insert

Use the options on the **Edit > Insert** menu to insert the following reserved words or standard blocks of text:

Selected Word

<eob>

<IF>...<THEN>

<IFNOT>...<THEN>

Selected Word

Use this option to insert the reserved word currently selected in the list of the reserved words to the active window at the current cursor position.



*You can also press the **Ctrl+W** keys.*

<eob>

Use this option to insert **<eob>** to the active window at the current cursor position.



*You can also press the **Ctrl+W** keys.*

<IF>...<THEN>

Use this option to insert the **IF-THEN-ENDIF** logical block to the active window at the current cursor position, and highlight the dots.

<IF>...<THEN>
<ENDIF>



*You can also press the **Ctrl+I** keys.*

<IFNOT>...<THEN>

Use this option to insert the **IFNOT-THEN-ENDIF** logical block to the active window at the current cursor position, and highlight the dots.

<IFNOT>...<THEN>
<ENDIF>



*You can also press the **Ctrl+T** keys.*

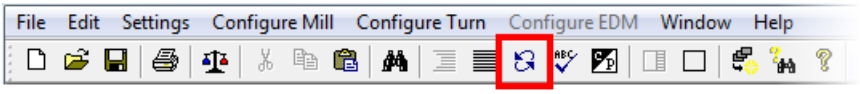
Display Control

Use the options on the **Edit > Display Control** menu to control the display of text within ConfigPost.

Refresh

Use this option to refresh the text in the active window. When ConfigPost refreshes the text in the active window, it first checks the syntax, then it indents the text based on the IF/IFNOT-THEN-ENDIF conditional statements (if this option is selected) performs **Logical Indent Text** (if **Logical Indent Text** is selected), and removes any empty lines.

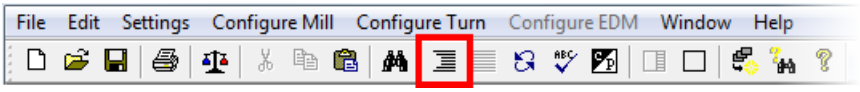
You can also click the **Refresh** button on the ConfigPost toolbar.



Logical Indent Text

Use this option to to indent the text in all windows based on the IF/IFNOT-THEN-ENDIF conditional statements. All lines under IF/IFNOT statements are shifted to the right. An ENDIF statement closes the logical block and is aligned with the previous IF/THEN statement.

You can also click the **Logical Indent Text** button on the ConfigPost toolbar.



Example of Logical Indent:

```
<IF><face-xy><THEN>
    {<motion> }{Z<z-coord> }<eob>
<ENDIF>
```

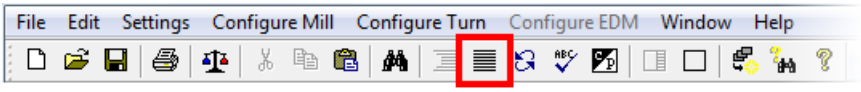
Example of Nested Logical Indent:

```
<IF><eq-string>(<tool-post>,GANG)<THEN>
    ~<spindle-on>M81<eob>
    <IFNOT><cw-spindle><THEN>
        ~<spindle-on>M80<eob>
    <ENDIF>
<ENDIF>
```

Left Justified Text

Use this option to enable left justified text indenting. The text is aligned at the left edge of the window.

You can also click the **Left Justified Text** button on the ConfigPost toolbar.



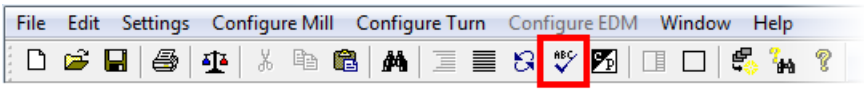
Example of Left Justified Text:

```
<IF><eq-string>(<tool-post>,GANG)
<THEN> ~<spindle-on>M81<eob>
<IFNOT><cw-spindle>
<THEN> ~<spindle-on>M80<eob>
<ENDIF>
<ENDIF>
```

Check Syntax

Use this option to check the validity of entered text.

You can also click the **Check Syntax** button on the ConfigPost toolbar.



The most common user errors are:

- Mismatched Parenthesis.
- Misspelled reserved word.
- Missing IF, IFNOT, THEN, ENDIF.

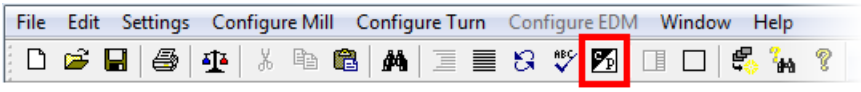
ConfigPost will issue an error, bring the window where error found in front and highlight the text (if possible).

Invert Colors

Use this option to invert the colors in the editor between:

- white background with black font; and,
- black background with white font.

You can also click the **Invert Colors** button on the ConfigPost toolbar.

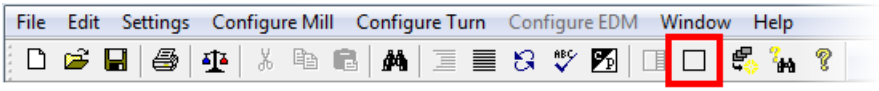


Hide/Show Reserved Words

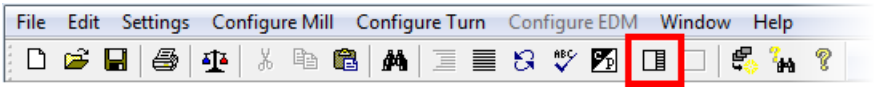
Use this option to show or hide the list of reserved words on the right side of every window.

You can also click the following buttons on the ConfigPost toolbar:

To Hide Reserved Words:



To Show Reserved Words:



Settings menu

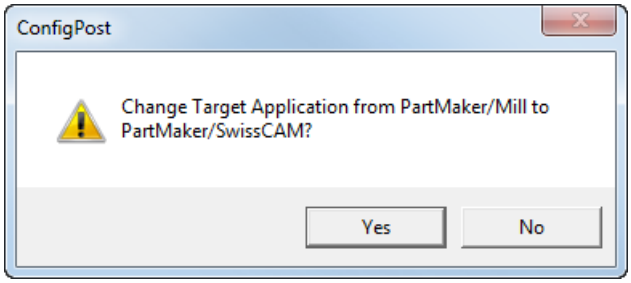
Use the options on the **Settings** menu to set the parameters that define the machine tool.

Target Application

Use the options on the **Settings > Target Application** menu to select the application that is using the post processor.

When you select an application, ConfigPost displays a confirmation dialog.

For example, if you change target application from PartMaker/Mill to PartMaker/SwissCAM the following confirmation dialog is displayed:



Click **Yes** to change the application. ConfigPost closes all open windows. Click **No** to cancel the change and remain with the original application.

Machine Settings

Use the **Settings > Machine Settings** menu option to display the **Machine Settings** dialog (see page 109).

User Data Names

Use the **Settings > User Data Names** menu option to display the **User Data Names** dialog (see page 120).

Variables Names

Use the **Settings > Variables Names** menu option to display the **Variables Names** dialog (see page 122).

Procedure Names

Use the options on the **Settings > Procedure Names** menu to name procedures for use in the milling, turning or EDM section of a Post file:

- Milling Procedures
- Turning Procedures
- EDM Procedures

Procedures are typically used in one of two ways; as sub-programs, or as storage.

When used as sub-programs, procedures are used for the same reason you would use a sub-program on a machine. You load a repetitive task into a procedure so that it can be recalled many times. By doing so, you can remove the repetition of NC code formatting throughout the post processor. This is also helpful if you have to change an NC code format that is used in many places. If it is in a procedure, you only need to change it in one location (in the procedure) and the change is made in all the necessary places.

When used as storage, you can load large formats of NC code into a procedure that would be cumbersome or difficult to manage in other ways. This is helpful because you can package similar functions together in one procedure.

Milling Procedures

Use this menu option to display the **Procedures Names, Mill** dialog (see page 116).

Turning Procedures

Use this menu option to display the **Procedures Names, Turn** dialog (see page 116).

EDM Procedures

Use this menu option to display the **Procedures Names, WireEDM** dialog (see page 117).

Advanced Customization

Use this menu option to display the **Advanced Customization** dialog (see page 89).



This feature is available in Multi-Axis Post Processors for PartMaker/Turn-Mill, and PartMaker/SwissCAM applications only.

Sub-Spindle Info



This feature is not used in Basic Post Processors.

Use this menu option to specify the location of the sub-spindle:

- Not Available
- Standalone
- Attached to End Working Slide
- Attached to Turret 1
- Attached to Turret 2
- Attached to Turret 3

The post processor uses this setting to examine your process table during NC code generation and check if the modes and tool posts are allowed according to the architecture of the machine, as defined in this section.

Tools Info

Use this menu option to display the **Tools Info** dialog (see page 119) and define what tool posts exist on the machine.



This feature is not used in Basic Post Processors.

Multi Coolant Info

Use this menu option to display the **Multi Coolant Info** dialog (see page 94).



This feature is not used in Basic Post Processors.

Faces Info

Use this menu option to display the **Faces Info** dialog (see page 97) and define preparatory functions for each tool post and each machining function.



This feature is not used in Basic Post Processors.

Channels, Tool Moving in Z

Use this menu option to display the **Channels, Tool Moving in Z** dialog (see page 89) and associate a tool post with a channel of NC code when the tool is moving.



This feature is not used in Basic Post Processors.

Channels, Tool Stationary in Z

Use this menu option to display the **Channels, Tool Stationary in Z** dialog (see page 90), where you can associate a tool post with a channel of NC code when the stock is moving.



This feature is not used in Basic Post Processors.

Internal Data

Use this menu option to display the **Internal Data** dialog (see page 107).

Configure Mill menu

Use the options on the **Configure Mill** menu to configure milling process output.

NC File Header/Footer Formats

Use the options on the **Configure Mill > NC File Header/Footer Formats** menu to configure the NC File Header/Footer formats for milling process output. File headers and footers begin and end the NC code.

The following options are available:

- **File Start**
- **Program End**
- **File End**
- **Developer Notes**

File Start

This program format is the first format called in ConfigPost. It is only called once and is used to output %, program numbers, and so on. It can also be used to initialize variables (see page 122) that are to be used in other formats.

File Start is where you create and manage multiple files so that you can correctly output NC code for multi-channel machines.

Program End

This program format is output at the end of a part program before sub-programs if the **End of File** option is used. You can use this program format to position the tool to an end point (using the `<tool-change-x>` and `<tool-change-y>` reserved words), turn coolant off, or restart the part program.

For incremental programming, you can use `<@tool-change-x>` and `<@tool-change-y>` to move to the first tool change position in the part program.

```
{N<seq> }M09<eob>
{N<seq> }G91G30Z<z-coord><eob>
{N<seq> }G30X<tool-change-x>Y<tool-change-y>M05<eob>
{N<seq> }M06M30<eob>
{N<seq> }G90M30<eob>
%<eob>
```

File End

This program format specifies the format of one or more lines to be placed at the end of the part program file and is normally used after the sub-programs section of a part program.

```
%<eob>
```

Developer Notes

This program format usually contains the post processor specific information. It is used by developer of the post processor for notes and comments. Machine specific M/G codes can be entered in this format. In addition, a history of changes can be inserted in this format.

Process Header/Footer Formats

Use the options on the **Configure Mill > Process Header/Footer Formats** to configure Process Header/Footer formats for milling process output. Process headers open and close individual processes or operations.

The following options are available:

- **Program Start**
- **Tool Change**
- **Process Start**
- **Process End**

Program Start

PartMaker outputs Program Start at the beginning of a part program. You must place general preparatory codes such as absolute, incremental, and so on, in this program format to define the dimension system. You must also define the first tool change in this program format. Reserved Words such as, `<speed>`, `<spindle-on>` and `<coolant>` as well as `<x-coord>` and `<y- coord>` specifying the first positioning move are commonly included in the Program Start format.

```
%<eob>
:<prog-num1> <eob>
(TOTAL MACHINING TIME: <time-total>) <eob>
{N<seq> }G80 G40 G17 <eob>
{N<seq> }G30 G91 Z0 <eob>
{N<seq> }T<tool-num> M6 <eob>
(<tool-name>) <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<z-coord> H<tool-offset> <coolant> <eob>
```

Tool Change

When you use the Tool Change format, a tool change block is output in the part program for all processes that require a tool change. Reserved Words such as `<speed>`, `<spindle-on>`, and `<coolant>` as well as `<x-coord>` and `<y-coord>` specifying the first positioning point are usually included in Tool Change format.

```
{N<seq> }M9<eob>
{N<seq> }G91 G30 Z0 M5<eob>
{N<seq> }T<tool-num> M6<eob>
(<tool-name>)<eob>
(TIME FOR THIS TOOL: <time-process>)<eob>
{N<seq> }M1<eob>
{N<seq> }G90 <work-offset> <motion> X<x-coord> Y<y-coord>
S<speed-rpm> <spindle-on><eob>
{N<seq> }G43 Z<z-coord> H<tool-offset> <coolant><eob>
```

Process Start

A process start block is output only between non-tool changing processes in the part program. The Process Start program format must contain all commands that might change between processes such as `<speed>`. You must also include the `<x-coord>`, `<y-coord>`, and `<z-coord>` reserved words in this program format.

```
{N<seq> }Z<z-coord><eob>
{N<seq> }{<work-offset> }{<motion> }X<x-coord> Y<y-coord>
{S<speed-rpm> }{<spindle-on>}<eob>
```

Process End

It is not recommended to use Process End format for Basic Post Processors. For multi-axis post processors, the Process End format is commonly used to cancel synchronization modes and retract the tool to a safe location before the tool change.

Motion Formats

Use the options on the **Configure Mill > Motion Formats** menu to configure Motion formats for milling process output.

Horizontal Rapid Move

Scenario A: If your CNC allows X, Y, and Z 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 X, Y, and Z axis motion programmed in the same line, you must use both the **Horizontal Rapid Move** format and the **Vertical Rapid Move** format.

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

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

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

Vertical Rapid Move

Scenario A: If your CNC allows X, Y, and Z 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 X, Y, and Z axis motion programmed in the same line, you must use both the **Horizontal Rapid Move** format and the **Vertical Rapid Move** format.

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

```
{N<seq> }{<motion> }{Z<z-coord> }<eob>
```

Linear Move

The **Linear Move** program format defines the part program output for linear moves. You must define the `<x-coord>`, `<y-coord>`, `<z-coord>`, `<motion>`, and `<feed-upm>` reserved words in this program format. You can use the `<comp-start>` reserved word to turn cutter compensation on or off.

```
{N<seq> }{<motion> }{<comp-status> }{X<x-coord> }{Y<y-coord> }{Z<z-coord> }
<IF><comp-start><THEN>
  D<comp-num>
<ENDIF>
```

```
{F<feed-upm>}
<eob>
```

Circular Move

In this program format, the reserved words <arc-x>, <arc-y>, <radius>, <center-x>, and <center-y> can be used.

```
{N<seq> }{<motion> }{X<x-coord> }{Y<y-coord> }I<arc-x>
J<arc-y> {F<feed-upm>}<eob>
```

Helical Move

In this program format, the reserved words <arc-x>, <arc-y>, <radius>, <center-x>, and <center-y> can be used. This program format should only be used if the machine is capable of helical interpolation (very few are):

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

Move to Hole Location

This program format defines the output for a specific hole location.

```
{N<seq> }{X<x-coord> }{Y<y-coord> }{Z<depth-abs>}<eob>
```

Subprograms Related Formats

Use the options in the **Configure Mill > Subprograms Related Formats** menu to configure Subprograms-Related formats for milling process output.

Linear Move in Subprogram

This program format is similar to the Linear Move (Motion) program format, but is used within a subprogram definition only without the <z-coord> reserved word.

```
{N<seq> }{<motion> }{<comp-status> }{X<@x-coord> }{Y<@y-
coord> }
<IF><comp-start><THEN>
  D<comp-num>
<ENDIF>
{F<feed-upm>}<eob>
```



The @ symbol in front of <x-coord> and <y-coord> signifies incremental output. If incremental output is generated, the post processor does not duplicate the path information for translations, it positions the translated location and uses the original incremental subprogram.

Circular Move in Subprogram

This program format is similar to the Circular Move (Motion) program format, but is used within a subprogram definition only.

```
{N<seq> }{<motion> }X<@x-coord> Y<@y-coord> I<arc-x>
J<arc-y>{ F<feed-upm>}<eob>
```

Helical Move in Subprogram

This Subprogram command is not used.

Move to Hole Location

This program format is similar to the Move to Hole Location (Motion) program format, but is used within a subprogram definition only.

```
{N<seq> }{X<x-coord> }{Y<y-coord>}<eob>
```

Open Subprogram

This program format specifies the format of the lines that precede a subprogram definition. You must use the `<subprogram-num>` reserved word and a preparatory function to define the incremental mode if you are using incremental subprograms.

```
:<subprogram-num><eob>
<IFNOT><drilling><THEN>
  {N<seq> }G91<eob>
<ENDIF>
```

Close Subprogram

This program format specifies the format of the lines that follow a subprogram definition. You might want to use a preparatory function to redefine the absolute mode if the main part program starts in the absolute mode.

```
<IFNOT><drilling><THEN>
  {N<seq>}G90<eob>
<ENDIF>
{N<seq>}M99<eob>
```

Call Subprogram

This program format specifies the format of a line that calls a subprogram. You must identify the subprogram using the <subprogram-num> reserved word.

```
<IF><drilling><THEN>
  {Z<depth-abs>}<eob>
<ENDIF>
M98 P<subprogram-num><eob>
```

Canned Cycle Formats

Use the options on the **Configure Mill > Canned Cycle Formats** menu to configure Canned Cycle formats for milling process output.

DRILL Cycle

This program format specifies a canned drilling cycle.

```
{N<seq> }{<retract> }<cycle> {X<x-coord> }{Y<y-coord>
}R<clear-abs> Z<depth-abs>
F<feed-upm><eob>
```

SPOT FACE Cycle

This program format specifies a canned spot face cycle.

```
{N<seq> }{<retract> }<cycle> {X<x-coord> }{Y<y-coord>
}R<clear-abs> Z<depth-abs>
P<dwell> F<feed-upm><eob>
```

DEEP HOLE Cycle

This program format specifies a canned deep hole cycle.

```
{N<seq> }{<retract> }<cycle> {X<x-coord> }{Y<y-coord>
}R<clear-abs> Z<depth-abs>
Q<step>F<feed-upm><eob>
```

CHIP BREAK Cycle

This program format specifies a canned chip break cycle:

```
{N<seq> }<cycle> {X<x-coord> }{Y<y-coord> }R<clear-abs>
Z<depth-abs>Q
<step>F<feed-upm><eob>
```

VARIABLE STEP Cycle

This program format specifies a canned variable step cycle:

```
<IFNOT><is-full-retract><THEN>
         G73 I<step> J<step-decrease> K< step-min>
Z<depth-abs> F<feed-upm><eob>
<ENDIF>
<IF><is-full-retract><THEN>
        G83 I<step> J< step-decrease> K< step-min>
Z<depth-abs> F<feed-upm><eob>
<ENDIF>
```

FLOAT TAP Cycle

This program format specifies a canned float tapping cycle.

```
{N<seq> }{<retract> }<cycle> {X<x-coord> }{Y<y-coord>
}R<clear-abs> Z<depth-abs>
F<feed-upm> <eob>
```

RIGID TAP Cycle

This program format specifies a canned rigid tap cycle.

```
{N<seq> }{<retract> }<cycle> {X<x-coord> }{Y<y-coord>
}R<clear-abs> Z<depth-abs>
F<feed-upm> <eob>
```

BORE, DWELL Cycle

This program format specifies a canned boring cycle with a dwell.

```
{N<seq> }{<retract> }<cycle> {X<x-coord> }{Y<y-coord>
}R<clear-abs> Z<depth-abs>
P<dwell> F<feed-upm><eob>
```

BORE Cycle

This program format specifies a canned boring cycle.

```
{N<seq> }{<retract> }<cycle> {X<x-coord> }{Y<y-coord>
}R<clear-abs> Z<depth-abs>
F<feed-upm><eob>
```

CIRCLE MILL Cycle

This program format specifies a canned circle milling cycle for machines that support such cycles (very few do).



If a canned Circle Mill cycle is not available with your CNC, leave this program format blank. This will signal PartMaker to generate the appropriate moves in "long hand" format.

THREAD MILL Cycle

This program format specifies a thread milling cycle for machines that support such cycles.



If a canned Thread Mill cycle is not available with your CNC, leave this program format blank. This will signal PartMaker to generate the appropriate moves "long hand" format.

HELICAL MILL Cycle

This program format specifies a canned helical mill cycle for machines that support such cycles. Helical mill cycles are typically not defined for lathes or mills, however this option is available for your use if provided by your control manufacturer.



If a canned Helical Mill cycle is not available with your CNC, leave this program format blank. This will signal PartMaker to generate the appropriate moves "long hand" format.

Canned Cycle Cancel

```
G80<eob>
```

Procedure Formats

For more information, see Procedures (see page 140).

Modes, Moving Stock

For more information, see Synchronization Modes (see page 160).



This feature is not used in Basic Post Processors.

Modes, Stationary Stock

For more information, see Synchronization Modes (see page 160).



This feature is not used in Basic Post Processors.

General Information

Use this menu option to display the **General Information** dialog (see page 98) where you can specify information for a part program such as machining data, subprograms support for milling and drilling operations (if your CNC is capable), and the type of circular interpolation your CNC uses.

Preparatory Functions

Use this menu option to display the **Preparatory Functions** dialog (see page 111), where you can specify machining codes to be used in the part program for cycle names, motion formats, cutter compensation, coolant type, and spindle direction.

Words Information

Use this menu option to display the **Words Information** dialog (see page 124) where you can specify parameters for reserved words used in a Post file.

Configure Turn menu

Use the options on the **Configure Turn** menu to configure turning process output using.

NC File Header/Footer Formats

Use the options on the **Configure Turn > NC File Header/Footer Formats** menu to configure the NC File Header/Footer formats for turning process output.

File Start

This program format is the first format called in ConfigPost. It is only called once and is used to output %, program numbers, and so on. It can also be used to initialize variables (see page 122) that are to be used in other formats.

Collet Re-Chuck

This program format is only used by PartMaker/SwissCAM. It is triggered by a checkbox in General Information (see page 102) called "Swiss Turn." This format is called once, right after the File Start format before any cutting process takes place. Collet Re-Chuck format is configured to produce G-codes required to setup the main collet at a distance from the face of the stock determined by the part length.



The values of reserved words such as `<tool-num>`, `<speed-rpm>`, etc. when used in the Collet Re-Chuck format are taken from the Cut-off Process.



The values of User Data (see page 120) are taken from the first process in the Process Table.

Program End

This program format is output at the end of a part program before subprograms if the End of File option is used. You can use this program format to position the tool to an end point (using the <tool-change-x> and <tool-change-z> reserved words, turn coolant off, or restart the part program. For incremental programming, you can use <@tool-change-x> and <@tool-change-z> to move to the first tool change position in the part program.

```
{N<seq> }M5<eob>
{N<seq> }G0 X<tool-index-x> Z<tool-index-z> T<$tool-num>00<eob>
{N<seq> }M9<eob>
{N<seq> }M30<eob>
```

File End

This program format specifies the format of one or more lines to be placed at the end of the part program file and is normally used after the subprograms section of a part program.

```
%<eob>
```

Process Header/Footer Formats

Use the options on the **Configure Turn > Process Header/Footer Formats** menu to configure Process Header/Footer formats for turning process output.

Program Start

Post outputs lines defined in Program Start at the beginning of a part program. You must place general preparatory codes such as absolute, incremental, etc. in this program format to define the dimension system. You must also define the first tool change in this program. Use of <tool-index-x> and <tool-index-z> reserved words is optional. The reserved words <x- coord> and <z-coord> must be included in the Program Start format.

```
:<prog-num1><eob>
{N<seq> }(<prog-name>)<eob>
{N<seq> }G97 S<speed-rpm> <spindle-on><eob>
{N<seq> }G0 X<tool-index-x> Z<tool-index-z><eob>
{N<seq> }T<tool-num><tool-offset> (<tool-name> <tool-note>)<eob>
<IF><css-on><THEN>
  {N<seq> }{G50 S<css-rpm-max>}<eob>
  {N<seq> }{G96 S<speed-fpm>}<eob>
<ENDIF>
{N<seq> }<coolant><eob>
```

```
{N<seq> }<feed-units><eob>
{N<seq> }{<tool-nose-com> }{<motion> }{X<x-coord> }{Z<z-
coord>}<eob>
```

Tool Change

When you use the Tool Change format, a tool change block is output in the part program for all processes that require a tool change. Reserved words such as `<speed-rpm>`, `<spindle- dir>`, and `<coolant>` are commonly included in the Tool Change format. Reserved words `<x-coord>` and `<z-coord>` must be included in the Tool Change format.

```
{N<seq> }G40 G0 X<tool-change-x> Z<tool-change-z>
T<$tool-num>00<eob>
<32><32><eob>
{N<seq> }G97 S<speed-rpm> <spindle-on><eob>
{N<seq> }G0 X<tool-change-x> Z<tool-change-z><eob>
{N<seq> }T<tool-num><tool-offset> (<tool-name> <tool-
note>)<eob>
<IF><css-on><THEN>
  {N<seq> }{G50 S<css-rpm-max>}<eob>
  {N<seq> }{G96 S<speed-fpm>}<eob>
<ENDIF>
{N<seq> }<feed-units><eob>
{N<seq> }{<tool-nose-com> }{<motion> }{X<x-coord> }{Z<z-
coord>}<eob>
```

Process Start

A process start block is output only between non-tool changing processes in the part program. The Process Start program format must contain all commands that might change between processes such as `<speed-rpm>`. You must also include the `<x-coord>` and `<z-coord>` reserved words in this program format.

```
<IFNOT><css-on><THEN>
  {N<seq> }{G97 S<speed-rpm>}<eob>
<ENDIF>
<IF><css-on><THEN>
  {N<seq> }{G50 S<css-rpm-max>}<eob>
  {N<seq> }{G96 S<speed-fpm>}<eob>
<ENDIF>
{N<seq> }{<feed-units>}<eob>
{N<seq> }{<tool-nose-com> }{<motion> }{X<x-coord> }{Z<z-
coord>}<eob>
```

Process End

For multi-axis postprocessors, the Process End format is often used to cancel synchronization modes and retract the tool to a safe location before the tool change.

Autodesk recommends that you do not use this format for basic postprocessors (see page 2).

Motion Formats

Use the options on the **Configure Turn > Motion Formats** menu to configure Motion formats for turning process output.

Rapid Move

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

```
{N<seq> }{<motion> }{X<x-coord> }{Z<z-coord>}<eob>
```

Linear Move

The Linear Move program format defines the part program output for linear moves. You must define the `<x-coord>`, `<z-coord>`, `<motion>`, and `<feed-upr>` or `<feed-upm>` reserved words in this program format. You can use the `<comp-start>` reserved word to turn cutter compensation on or off. Please note how the appropriate value for the feed rate is obtained.

```
{N<seq> }{<feed-units> }{<motion> }{X<x-coord> }{Z<z-
coord> }
<IF><upr-units-used><THEN>
  {F<feed-upr>}
<ENDIF>
<IFNOT><upr-units-used><THEN>
  {F<feed-upm>}
<ENDIF>
<eob>
```

Circular Move

In this program format, the reserved words `<arc-x>`, `<arc-z>`, `<radius>`, `<center-x>`, and `<center-z>` can be used.

```
{N<seq> }{<feed-units> }{<motion> }{X<x-coord> }{Z<z-
coord> }R<radius>
<IF><upr-units-used><THEN>
  {F<feed-upr>}<eob>
```

```
<ENDIF>
<IFNOT><upr-units-used><THEN>
  {F<feed-upm>}<eob>
  <ENDIF>
```

Thread Move

This motion format defines the linear move associated with threading, typically G32, G33, or G92 depending on your control. Note that `<pitch>` is substituted for `<feed-upr>` or `<feed- upm>`.

```
{N<seq> }<motion> X<x-coord> Z<z-coord> F<pitch><eob>
```

Thread Whirl Move

This motion format defines the linear move associated with thread whirling.

Two examples are shown below. Please consult your machine manual for details of the formatting it requires for thread whirling.

Example 1

The machine uses G32 and a Feed Rate in UPR. The stock position is controlled with the turning spindle:

```
{N<seq> }G32 X<x-coord> Z<z-coord> F<pitch><eob>rd> Z<z-coord> F
```

Example 2

The machine uses G1 and a Feed Rate in UPM. The stock positioning is controlled with the C-axis:

```
{N<seq> }G1 X<x-coord> C<@c-angle> Z<z-coord> F<feed-upm><eob>rd> Z<z-coord> F
```

MCP-Action

This program format contains the configuration for:

Material Control Processes for PartMaker/Turn and PartMaker/Turn-Mill (see page 29)

and

Material Control Processes for PartMaker/SwissCAM (see page 31).

Canned Cycles

Use the options on the **Configure Turn > Canned Cycles** menu to configure Canned Cycle formats for turning process output.

DRILL Cycle

This program format specifies a canned drilling cycle.

```
{N<seq> }<cycle> Z<depth-abs> K<step1> F<feed-upm><eob>
```

SPOT FACE Cycle

This program format specifies a canned spot face cycle.

```
{N<seq> }G1 Z<depth-abs>
<IF><upr-units-used><THEN>
  {F<feed-upr>}
<ENDIF>
<IFNOT><upr-units-used><THEN>
  {F<feed-upm>}
<ENDIF>
<eob>
{N<seq> }G4 U.5<eob>
{N<seq> }G0 X0 Z<face-clear><eob>
```

DEEP HOLE Cycle

This program format specifies a canned deep hole drilling cycle.

```
{N<seq> }<cycle> Z<depth-abs> K<step1>
<IF><upr-units-used><THEN> F<feed-upr>
<ENDIF>
<IFNOT><upr-units-used><THEN> F<feed-upm>
<ENDIF>
<eob>
```

CHIP BREAK Cycle

This program formats specifies a canned chip break cycle:

```
{N<seq> }<cycle> Q<step> Z<depth-abs>
<IF><upr-units-used><THEN> F<feed-upr>
<ENDIF>
<IFNOT><upr-units-used><THEN> F<feed-upm>
<ENDIF>
<eob>
```

VARIABLE STEP Cycle

This program format specifies a canned variable step cycle:

```
<IFNOT><is-full-retract><THEN>
         G73 I<step> J<step-decrease> K< step-min>
Z<depth-abs> F<feed-upm><eob>
<ENDIF>
<IF><is-full-retract><THEN>
        G83 I<step> J< step-decrease> K< step-min>
Z<depth-abs> F<feed-upm><eob>
<ENDIF>
```

FLOAT TAP Cycle

Float tapping cycles are typically not defined for lathes, however this option is available for your use if provided by your control manufacturer.

RIGID TAP Cycle

This program format specifies a canned rigid tapping cycle. Rigid tapping cycles are typically not defined for lathes; however this option is available for your use if provided by your control manufacturer.

```
{N<seq> }G32 Z<depth-abs> F<pitch><eob>
{N<seq> }G4 U.5<eob>
{N<seq> }G32 Z<face-clear> M14<eob>
```

BORE,DWELL Cycle

Bore with a dwell at the bottom of the motion is typically not defined for lathes, however this option is available for your use if provided by your control manufacturer.

FINISH Cycle

This cycle will finish a defined contour.

```
{N<seq> }<cycle> P<shape-start-p> Q<shape-end-q> <eob>
```

TURN Cycle

This cycle will rough down to a defined contour, using passes along the axis of the spindle.

```
{N<seq> }<cycle> U<x-finish-allow> W<z-finish-allow>
P<shape-start-p> Q<shape-end-q> D<int-depth-cut>
<IF><upr-units-used><THEN> F<feed-upr>
<ENDIF>
<IFNOT><upr-units-used><THEN> F<feed-upm>
<ENDIF>
<eob>
```

FACE-TURN Cycle

This cycle will rough to a contour, using cuts perpendicular to the spindle axis.

```
{N<seq> }<cycle> U<x-finish-allow> W<z-finish-allow>
P<shape-start-p> Q<shape-end-q> D<int-depth-cut>
<IF><upr-units-used><THEN> F<feed-upr>
<ENDIF>
<IFNOT><upr-units-used><THEN> F<feed-upm>
<ENDIF>
```

```
<eob>
```

CONTOUR Cycle

This cycle will rough to a defined contour, using offset duplicates of that contour.

```
{N<seq> } <cycle> P<shape-start-p> Q<shape-end-q> I<diam-
clear> K<face-clear> U<x-finish-allow> W<z-finish-allow>
D<num-passes>
<IF><upr-units-used><THEN> F<feed-upr>
<ENDIF>
<IFNOT><upr-units-used><THEN> F<feed-upm>
<ENDIF>
<eob>
```

GROOVE Cycle

This cycle will cut a groove on the circumference of cylindrical stock.

```
{N<seq> }<cycle> Z<thread-final-z> X<thread-final-x>
I<step1> K<step2>
<IF><upr-units-used><THEN>
  F<feed-upr>
<ENDIF>
<IFNOT><upr-units-used><THEN>
  F<feed-upm>
<ENDIF>
<eob>
```

FACE-GROOVE Cycle

This cycle will cut a groove in the face of the stock.

```
{N<seq> }<cycle> Z<thread-final-z> X<thread-final-x>
I<step1> K<step2>
<IF><upr-units-used><THEN> F<feed-upr>
<ENDIF>
<IFNOT><upr-units-used><THEN> F<feed-upm>
<ENDIF>
<eob>
```

THREAD Cycle

This cycle cuts threads on the circumference of the stock.

```
{N<seq> }<cycle> X<thread-final-x> Z<thread-final-z>
K<height> D<int-depth-cut>
A<infeed-angle> F<pitch> I<taper-val><eob>
```

FACE-THREAD Cycle

This cycle cuts threads on the face of the stock.

```
{N<seq> }<cycle> X<thread-final-x> Z<thread-final-z>
H<height> D<int-depth-cut>
B<infeed-angle> F<pitch> L<cham-length> E<lead-var>
K<taper-val> W<z-finish-allow> M32
M75<eob>
```

Other Formats

Use the options on the **Configure Turn > Other Formats** menu to configure Other formats for turning process output.

Configure Turn > Other Formats > End of Shape

This program format specifies the end of a profile shape definition. Not typically used.

Configure Turn > Other Formats > Tool Definition

This program format is used strictly for interfacing with verification software.

Configure Turn > Other Formats > Home Name Definition

This program format is used strictly for interfacing with verification software.

Procedure Formats

For more information, see Using Procedures (see page 140).

Modes, Moving Stock

For more information, see Synchronization Modes (see page 160).



This feature is not used in Basic Post Processors (see page 2).

Modes, Stationary Stock

For more information, see Synchronization Modes (see page 160).



This feature is not used in Basic Post Processors (see page 2).

General Information

This command displays the **General Information** dialog (see page 102) where you can specify information for a part program such as machining data, turret definition information and the type of circular interpolation your CNC uses.

Preparatory Functions

This command displays the **Preparatory Functions** dialog (see page 112) where you can specify machining codes to be used in the part program for cycle names, motion formats, cutter compensation, coolant type, and spindle direction.

Words Information

This command displays the **Words Information** dialog (see page 125) where you can specify parameters for reserved words used in a Post file (see page 15).

Configure EDM menu

Use the options on the **Configure EDM** menu to configure EDM process output.

NC File Header/Footer Formats

Use the options on the **Confiigure EDM > NC File Header/Footer Formats** menu to configure NC File Header/Footer formats for EDM process output.

File Start

This program format is the first format called in ConfigPost. It is only called once and is used to output %, program numbers, etc. It can also be used to initialize variables (see page 122) that are to be used in other formats.

Program End

This program format is output at the end of a part program. You can use this program format to empty the coolant tank and end the program.

```
{N<seq> }M88<eob>
{N<seq> }G23<eob>
{N<seq> }M02<eob>
%<eob>
```

File End

This program format specifies the end of the output file.

Process Header/Footer Formats

Use the options on the **Configure EDM > Process Header/Footer Formats** menu to configure Process Header/Footer formats for EDM process output.

Program Start

PartMaker outputs lines defined in Program Start at the beginning of a part program. You must place general preparatory codes such as absolute, incremental, etc. in this program format to define the dimension system. You must also define preparatory commands in this program format. Positioning moves (<x-coord> and <y-coord>) also must be entered in this program format.

```
;-------------------SET VAR<eob>
~<v05-FEED-UPM>25.<eob>
;-------------------<eob>
%<eob>
L<prog-num1>/<prog-name><eob>
{N<seq> }(CREATED BY MITSUBISHI FX20 POST VER A11)<eob>
{N<seq> }(TOTAL MACHINING TIME: <time-total>)<eob>
{N<seq> }(MACHINING <face-name> <group-name>)<eob>
{N<seq> }(MACHINING PROCESS <process-id>)<eob>
<p04-PROCESS>
{N<seq> }G62 X0 Y0<eob>
{N<seq> }G97 X0<eob>
{N<seq> }G90 G70 (INCH)<eob>
{N<seq> }M78 M78 (FILL TANK)<eob>
{N<seq> }G1 X<x-coord> Y<y-coord> F<v05-FEED-UPM><eob>
{N<seq> }G92 X<x-coord> Y<y-coord><eob>
  <IF><thread-wire><THEN>
<p01-THREAD-WIRE>
<ENDIF>
{N<seq> }M90<eob>
<IFNOT><eq-number>(<offset-value>,<$offset-value>)<THEN>
  {N<seq> }H01=<offset-value><eob>
<ENDIF>
```

Process Start

A process start block is output only between individual processes in the Process Table. The Process Start program format must contain all commands that might change between processes such as <offset-value>. You must also include the <x-coord> and <y-coord> reserved words in this program format.

```
;-------------------SET VAR<eob>
~<v05-FEED-UPM>25.<eob>
;-------------------<eob>
<32><32><eob>
{N<seq> }(MACHINING <face-name> <group-name>)<eob>
{N<seq> }(MACHINING PROCESS <process-id>)<eob>
<p04-PROCESS>
<IFNOT><thread-wire><THEN>
  <IF><eq-number>(<oper-num>,2)<THEN>
    {N<seq> }M78 M78 (FILL TANK)<eob>
    {N<seq> }M80 M82 M84<eob>
  <ENDIF>
<ENDIF>
{N<seq> }G1 X<x-coord> Y<y-coord> F<v05-FEED-UPM><eob>
<IF><thread-wire><THEN>
  {N<seq> }G92 X<x-coord> Y<y-coord><eob>
  <p01-THREAD-WIRE>
<ENDIF>
{N<seq> }M90<eob>
<IFNOT><eq-number>(<offset-value>,<$offset-value>)<THEN>
  {N<seq> }H01=<offset-value><eob>
<ENDIF>
```

Process End

It is not recommended to use Process End format for Basic Post Processors. For Multi-Axis Post Processors, Process End Format is commonly used to cancel synchronization modes and retract the tool to a safe location before the tool change.

Motion Formats

Use the options on the **Configure EDM > Motion Formats** menu to configure EDM Motion formats for EDM process output.

Rapid Move

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

```
~<v05-FEED-UPM>25.<eob>
{N<seq> }{<motion> }{X<x-coord> }{Y<y-coord> }{F<v05-
FEED-UPM>}<eob>
```

Linear Move

The Linear Move program format defines the part program output for linear moves. You must define the <comp-status>, <motion>, <x-coord>, <y-coord>, and <feed-upm> reserved words in this program format.

```
~<v05-FEED-UPM><feed-upm><eob>
;-------------IF COMP OFF, THEN INCREASE FEED<eob>
<IF><comp-end><THEN>
  ~<v05-FEED-UPM>5.<eob>
<ENDIF>
;-------------<eob>
{N<seq> }{<comp-status> }{<motion> }{X<x-coord> }{Y<y-
coord> }{F<v05-FEED-UPM> }{E<e-pack> }
<IF><taper-used-in-proc><THEN>
  <IF><comp-start><THEN>
    A<taper-angle><32>
  <ENDIF>
  <IFNOT><comp-start><THEN>
    {A<taper-angle>}<32>
  <ENDIF>
<ENDIF>
<IF><comp-start><THEN>
  H01
<ENDIF>
<IF><comp-end><THEN>
  M91
<ENDIF>
<eob>
<IF><glue-stop><THEN>
  /M1 (GLUE STOP)<eob>
<ENDIF>
```

Circular Move

In this program format, the reserved words <arc-x>, <arc-y>, <radius>, <center-x>, and <center-y> can be used, for example:

```
~<v05-FEED-UPM><feed-upm><eob>
{N<seq> }{<motion> }{X<x-coord> }{Y<y-coord> }R<radius>
{F<v05-FEED-UPM>}
<IF><taper-used-in-proc><THEN>
  {A<taper-angle>}
<ENDIF>
<eob>
<IF><glue-stop><THEN>
  /M1 (GLUE STOP)<eob>
```

```
<ENDIF>
```

XY UV Linear Move

```
{N<seq> }{<comp-status> }{<motion> }{X<x-coord> }{Y<y-
coord> }
{U<u-relat> }{V<v-relat> }{F<feed-upm> }{E<e-pack> }<eob>
```

XY UV Circular Move

```
{N<seq> }{<comp-status> }{<motion> }{X<x-coord> }{Y<y-
coord> }{I<center-x> }{J<center-y> }
{U<u-relat> }{V<v-relat> }{K<center-u> }{L<center-v> }
{F<feed-upm> }{E<e-pack> }<eob>
```



To use this format, you must first configure Multi Quadrant Arcs in the General Information dialog (see page 98).

Procedure Formats

For more information, see Using Procedures (see page 140)

General Information

This command displays the **General Information** dialog (see page 104) where you can specify information for a part program such as machining constants and the type of circular interpolation your CNC uses.

Preparatory Functions

This command displays the **Preparatory Functions** dialog (see page 114) where you can specify machining codes to be used in the part program for motion formats and cutter compensation.

Words Information

This command displays the **Words Information** dialog (see page 127) where you can specify parameters for reserved words used in a Post file (see page 15).

Window menu

Use the options on the **Window** menu allow you to arrange the active PartMaker windows.

Close All

Use this menu option to close all windows in the ConfigPost application.

Cascade

Use this menu option to cascade all of the windows in the ConfigPost application.

Tile

Use this menu option to tile all of the windows in the ConfigPost application.

Help menu

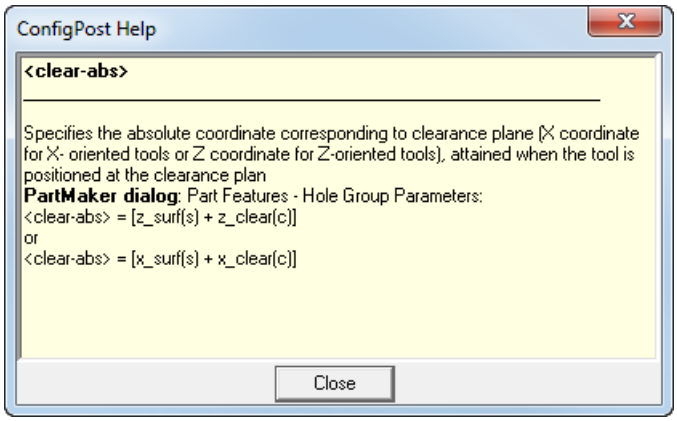
Use the options on the **Help** menu to access various sources of information about PartMaker.

Help Topics

Use this menu option to display the ConfigPost user guide PDF.

What's this?

Use this menu option to display information about a valid reserved word in a separate **ConfigPost Help** window, for example:

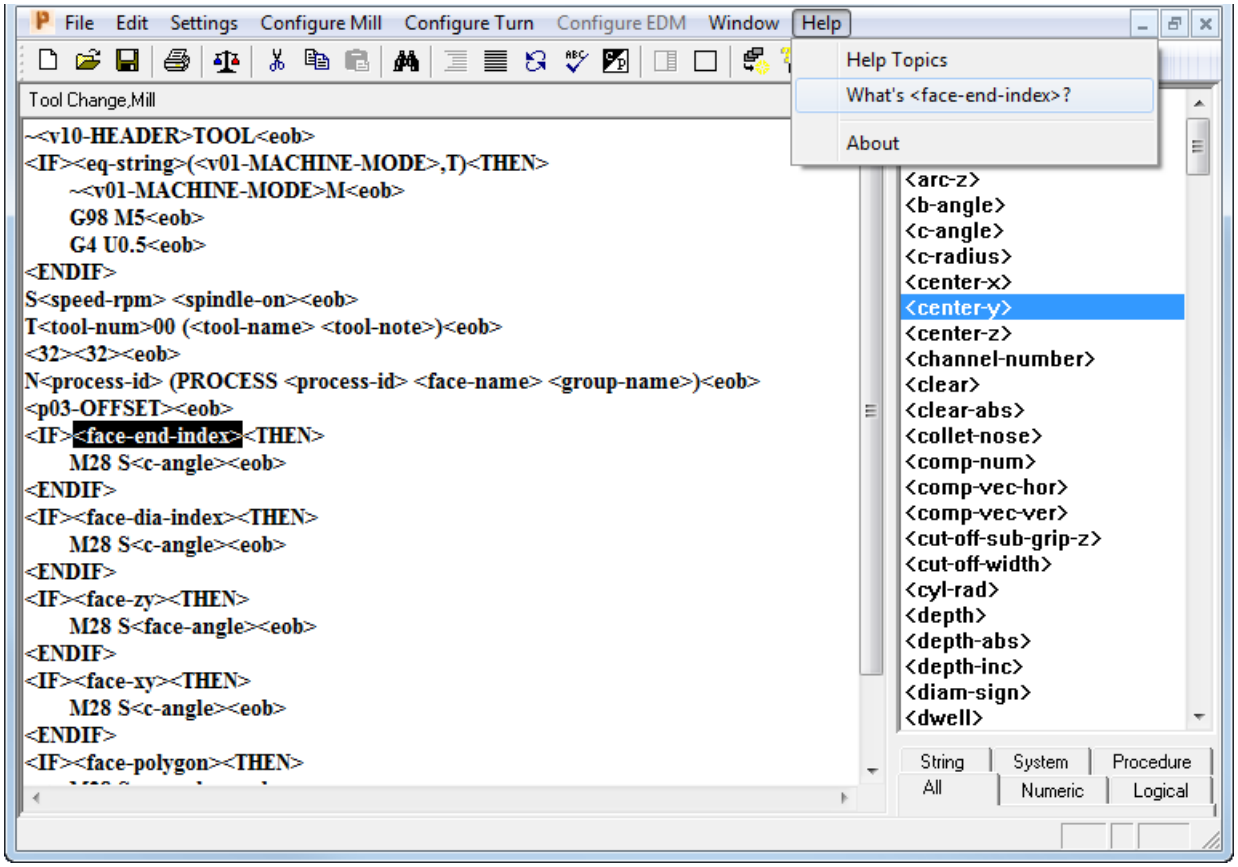


You can access this window by using the:

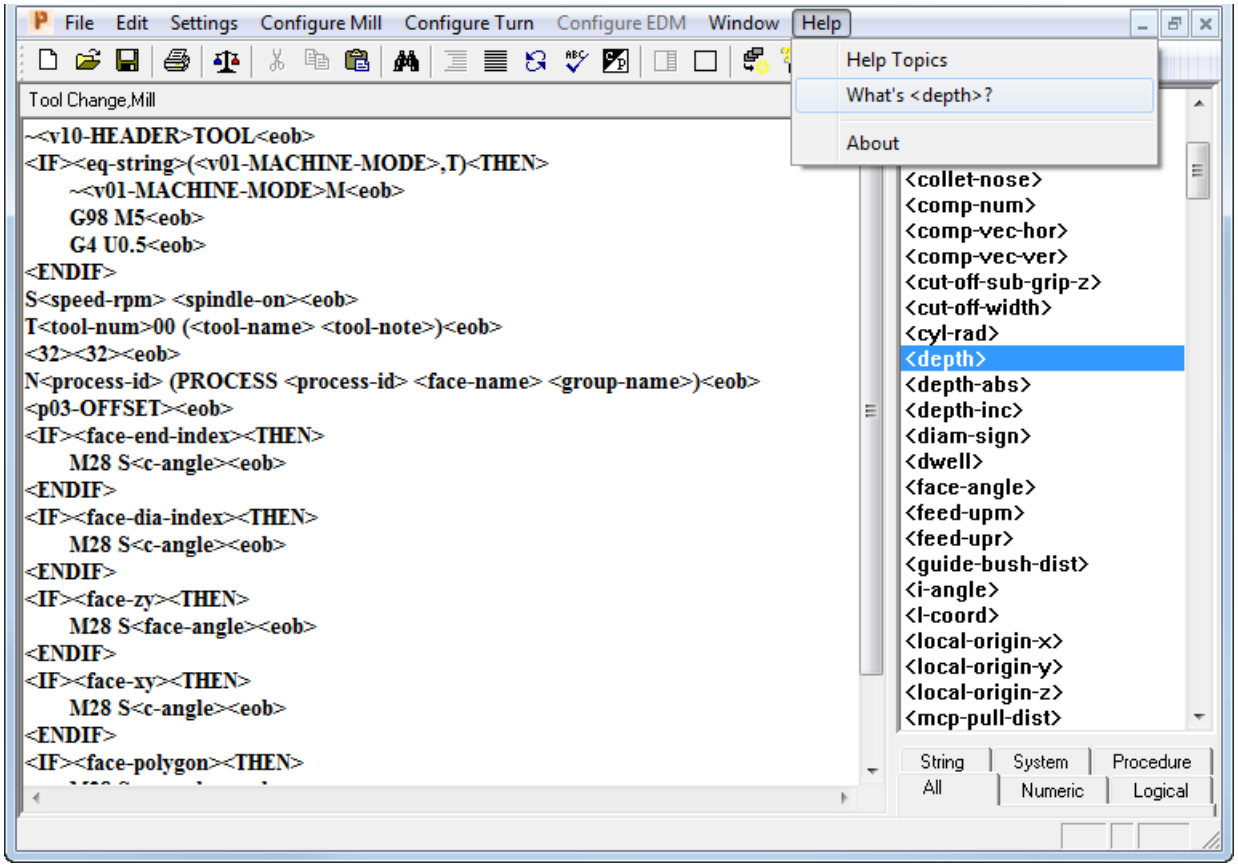
- **Help > What's This?** menu option.
- Pop-up menu over the editor.
- Pop-up menu over the list of reserved words.

Accessing What's this? command through the main menu Help > What's this?

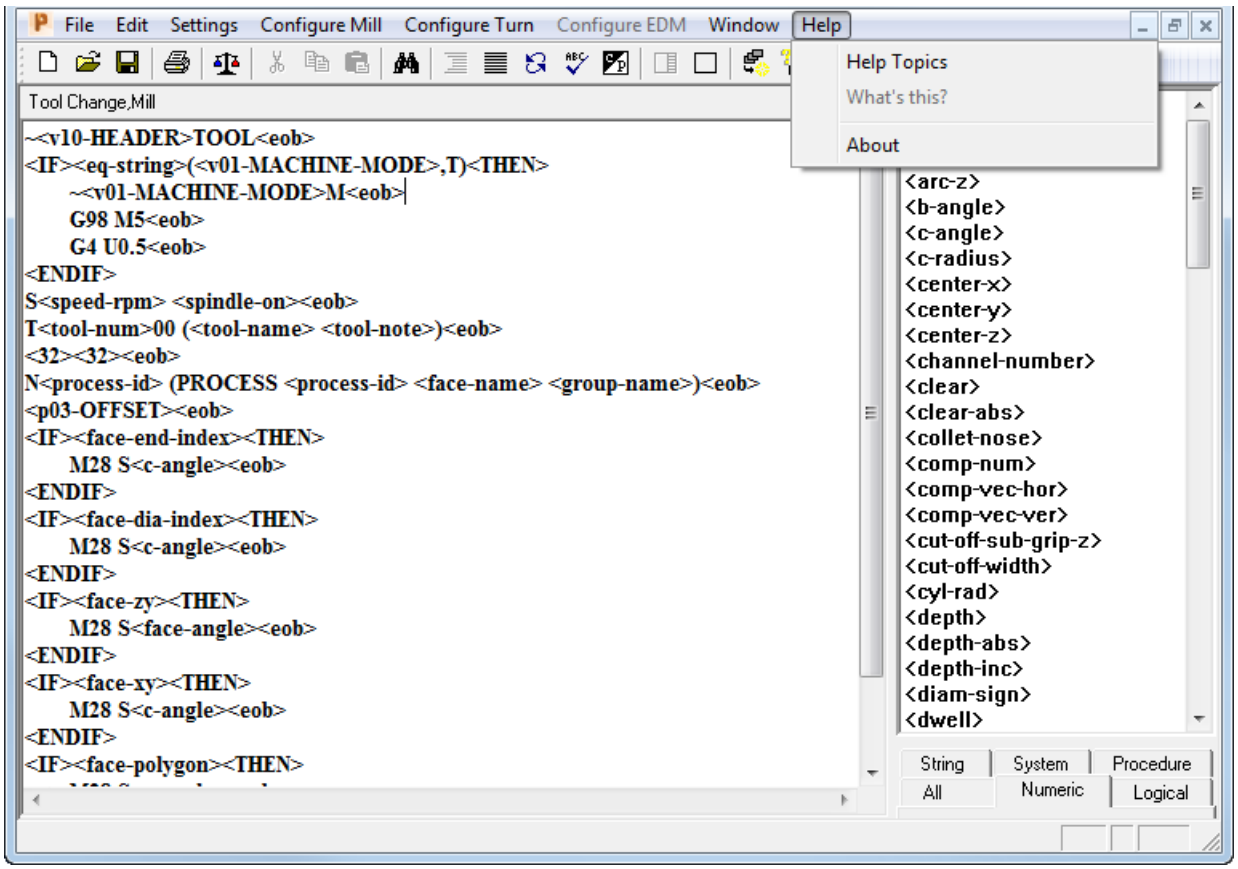
- If the valid reserved word is highlighted in the editor, for example `<face-end-index>`, the menu changes to **Help > What's <face-end-index>?**



- If no valid reserved word is selected in the editor, but reserved word, for example `<depth>`, is selected in the list of reserved words then menu changes to **Help > What's <depth>?**

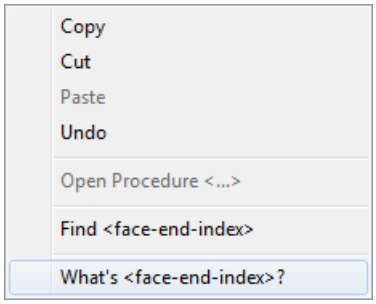


- If neither valid reserved word is selected in the editor, nor the reserved word is selected in the list of reserved words then the menu displays the generic name **Help > What's this?**



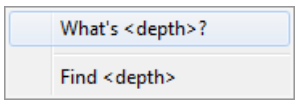
Accessing What's this? command through the context menu over the editor

When you right-click a valid reserved word, such as `<face-end-index>`, it is highlighted and the context menu changes to:



Accessing What's this? command through the context menu over the list of reserved words

When you right-click a valid reserved word in the list of reserved words, such as `<depth>`, it is highlighted and the context menu changes to:



About

Use this menu option to display the **About ConfigPost** dialog, where you can view version and licensing information about the ConfigPost application that is installed.

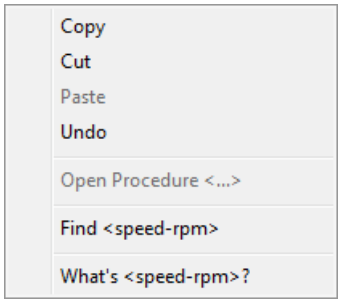
Right-click context menus

Right-click over the editor or over the list of reserved words to access the context menu.

Right-click context menu on the editor

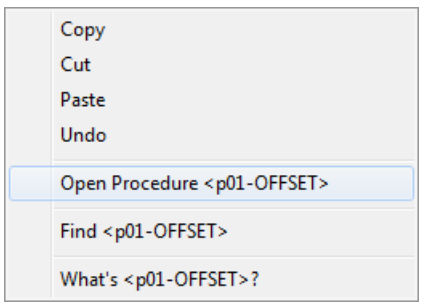
When you right-click a reserved word in the editor, ConfigPost automatically highlights it.

The following context menu is displayed (if you selected `<speed-rpm>`):



- **Copy**, **Cut**, **Paste**, **Undo** — These options enable you to perform standard window operations.
- **Open Procedure <...>** — Select this option to open a procedure window (see page 140) corresponding to the selected procedure name.

  For example, if `<p01-OFFSET>` is a valid procedure name and it is selected in the editor, the context menu changes to:

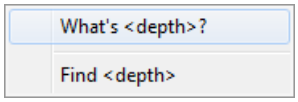


- **Find** — Select this option to open the **Find** dialog.
- **What's this?** — Select this option to display the help for a selected valid reserved word.

Right-click context menu on the list of the reserved words

When you right-click on the list of the reserved words, ConfigPost automatically highlights the valid reserved word and displays the context menu.

When you right-click a valid reserved word in the list of reserved words, such as `<depth>`, it is highlighted and the context menu changes to:



- **What's this?** — Select this option to display the help for a selected valid reserved word.
- **Find** — Select this option to open the **Find** dialog.

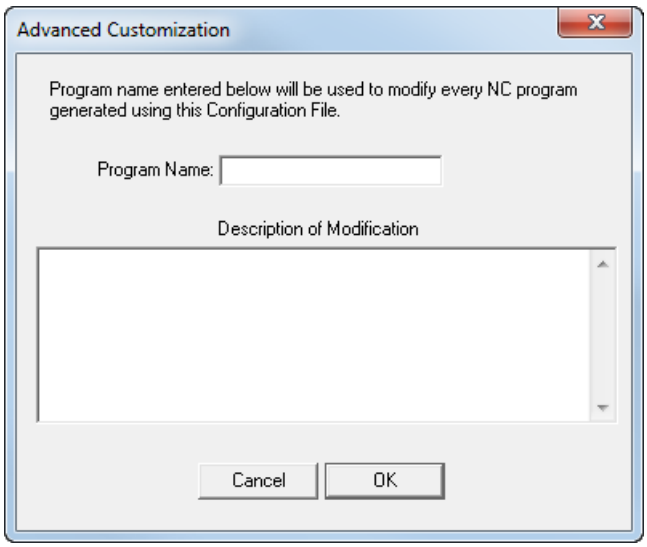
Dialogs

This section contains details of all the dialogs available in ConfigPost.

Advanced Customization dialog

Use the **Advanced Customization** dialog to create a compiled .exe file that is run after PartMaker has finished post processing.

To display the dialog, select **Settings > Advanced Customization** from the Menu bar.

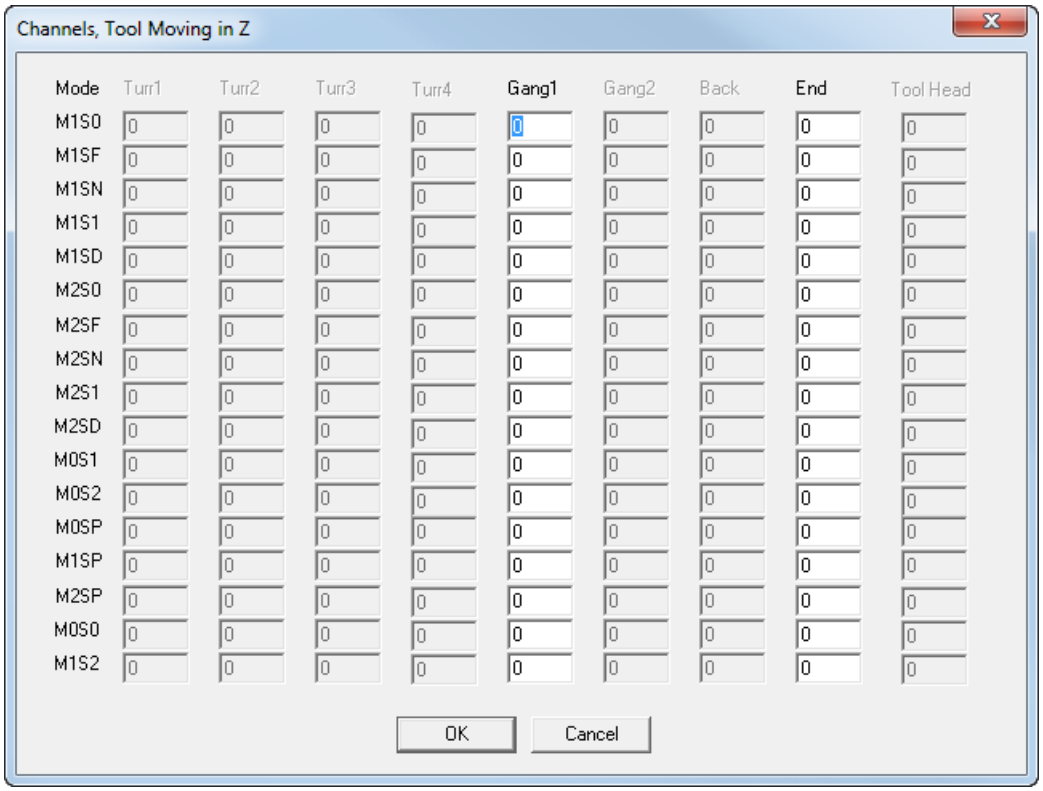


Channels, Tool Moving in Z dialog

This feature is not used in Basic Post Processors.

Use the **Channels, Tool Moving in Z** dialog to enter a specific channel number for available tool posts. The fields become enabled when the **Tools Info** dialog (see page 119) contains valid **Tools Ranges**.

To display this dialog, select **Settings > Channels, Tool Moving in Z** from the menu bar.

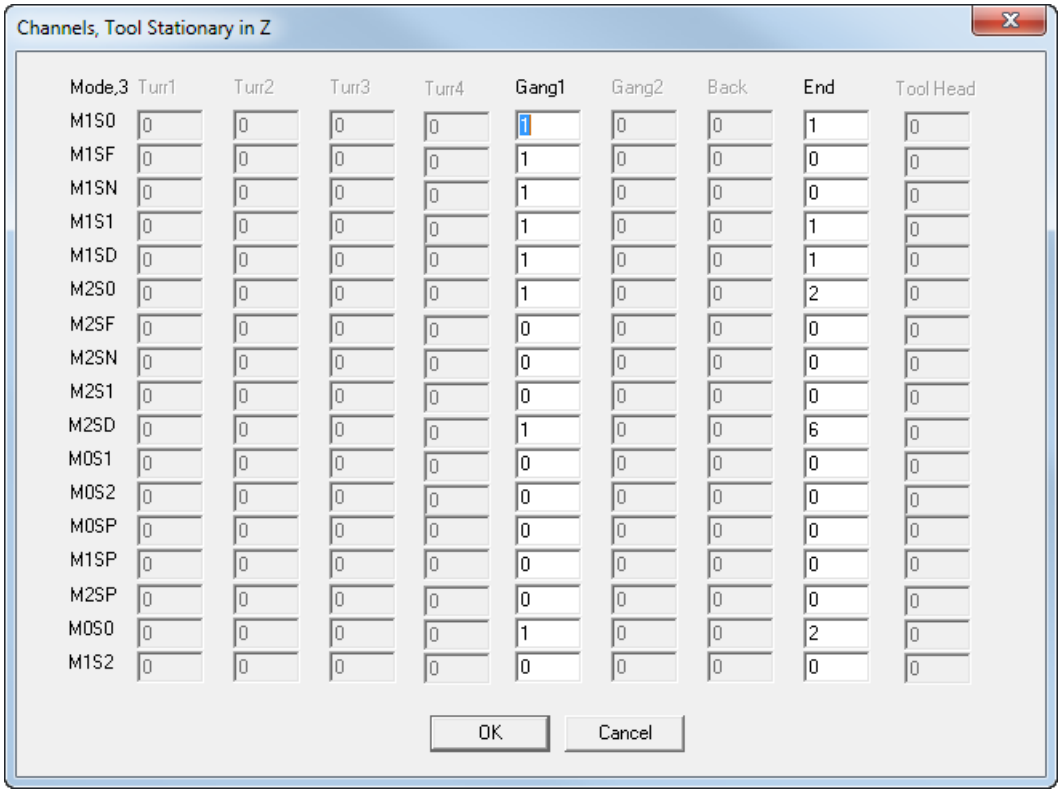


Channels, Tool Stationary in Z dialog

This feature is not used in Basic Post Processors.

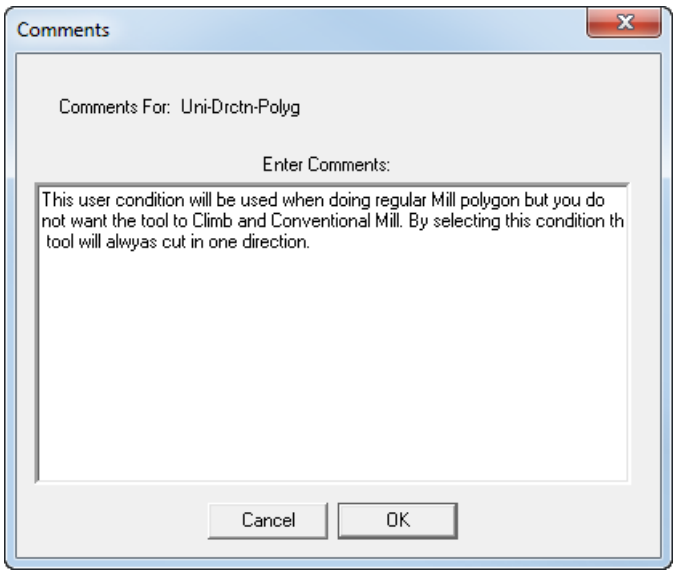
Use the **Channels, Tool Stationary in Z** dialog to enter specific channel numbers for available tool posts. The fields become available when the **Tools Info** dialog (see page 119) contains valid **Tools Ranges** values.

To display the dialog, select **Settings > Channels, Tool Stationary in Z** from the menu bar.



Comments dialog

Use the **Comments** dialog to enter comments for Procedures, User Data, and Variables.

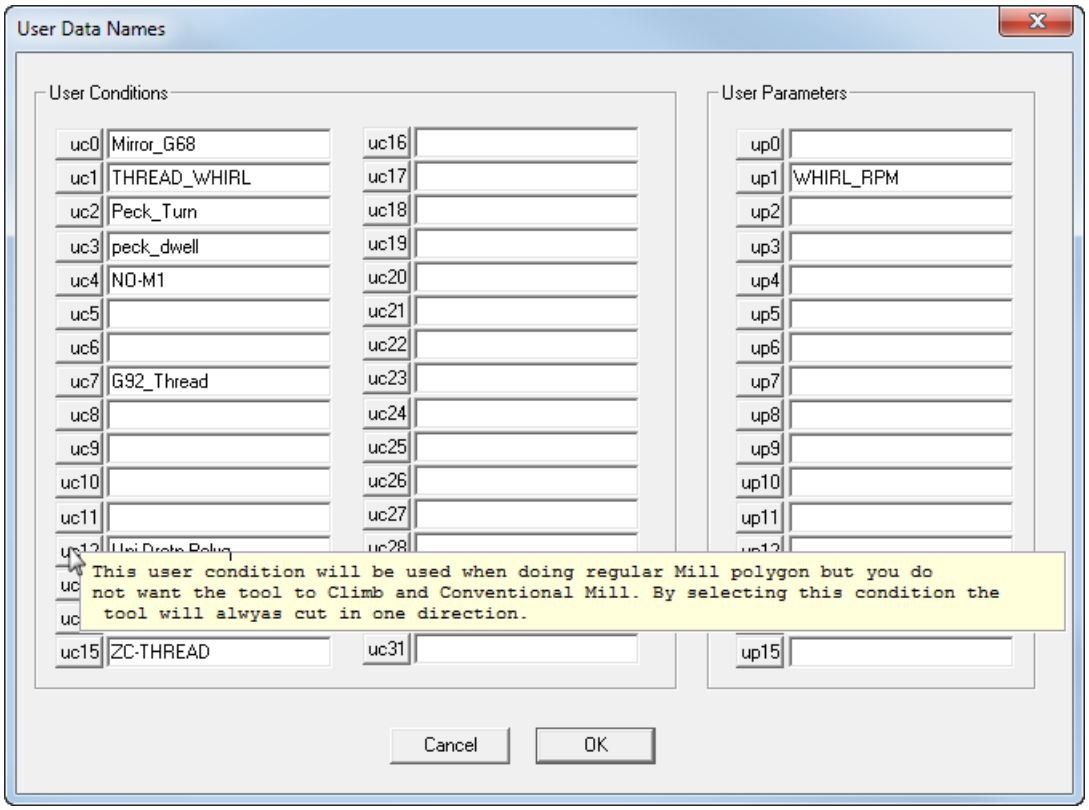


To display the dialog, click on a raised button in one of the following dialogs:

- **Procedure Names** dialog (see page 115).
- **User Data Names** dialog (see page 120).
- **Variables Names** dialog (see page 122).

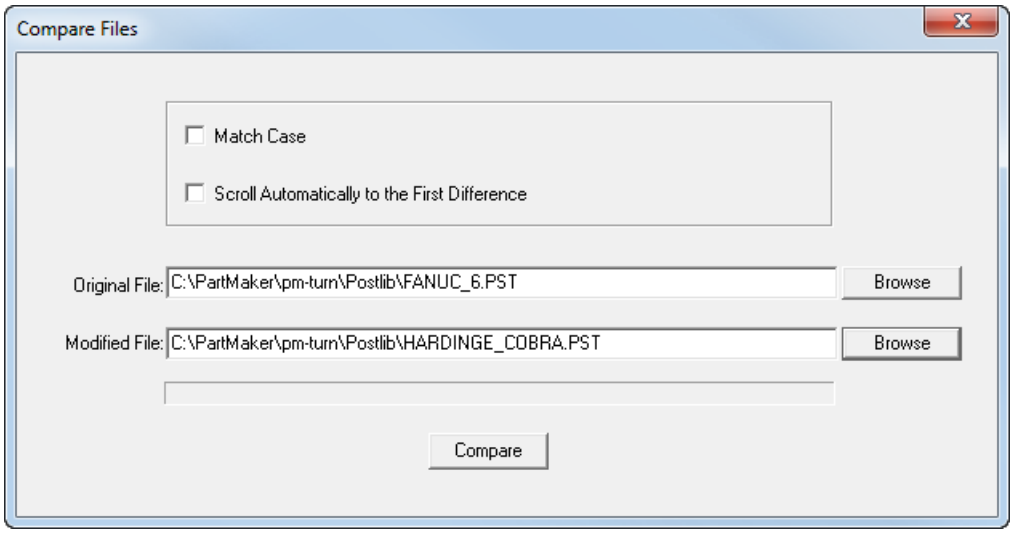
The comments entered become available as tool-tips in the dialogs where the comments dialog was called from.

In order to see the tooltip with entered comments, position the cursor over the raised button. For, example:



Compare Files dialog

Use the **Compare Files** dialog to compare two Post files.



To display this dialog, select **Compare Files** from the **File** menu.

The following settings are available:

Match Case — Select this option to enable you to match case during file comparison.

Scroll Automatically to the First Difference — Select this option to enable the compare function to jump to the first found difference when the new format is displayed.

Original File — This field specifies the full path and the file name of the file that you are going to compare.

Modified File — This field specifies the full path and the file name of the file that you are going to compare.

Browse — Click to browse for the Post file you want to compare.

Compare — Performs the file comparison and displays the result in the **Post File Differences** dialog (see page 118).

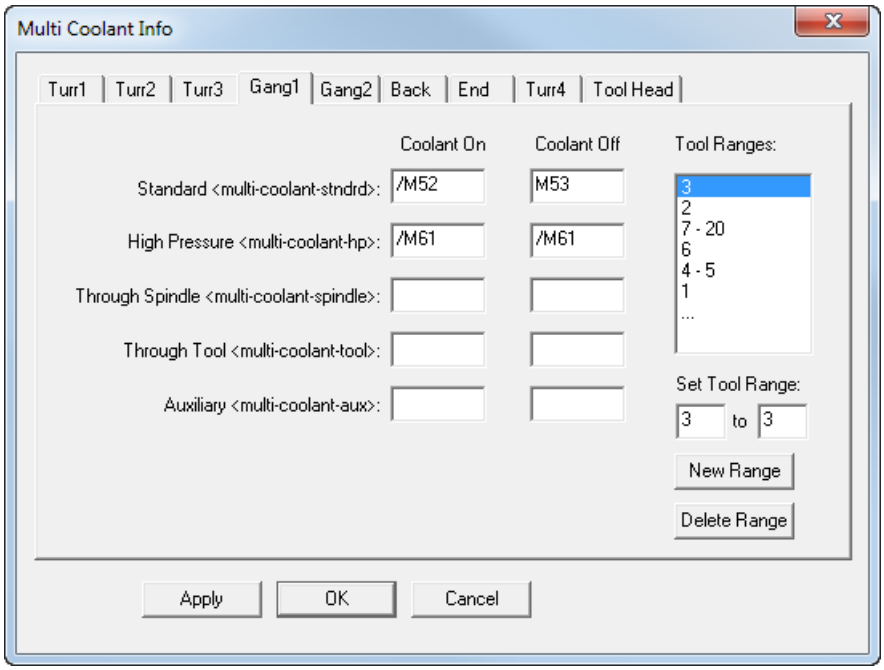
Multi Coolant Info dialog

Use this dialog to enter **Tool Range** specific coolant M codes for your chosen tool post.

This feature is not used in Basic Post Processors.

To display the **Multi Coolant Info** dialog, select one of the options from the **Settings > Multi Coolant Info** submenu.

The fields become available when the **Tools Info** dialog (see page 119) contains valid **Tool Ranges** for your selected tool post.



New Range — Click to create a new range in the **Tool Ranges** list box.

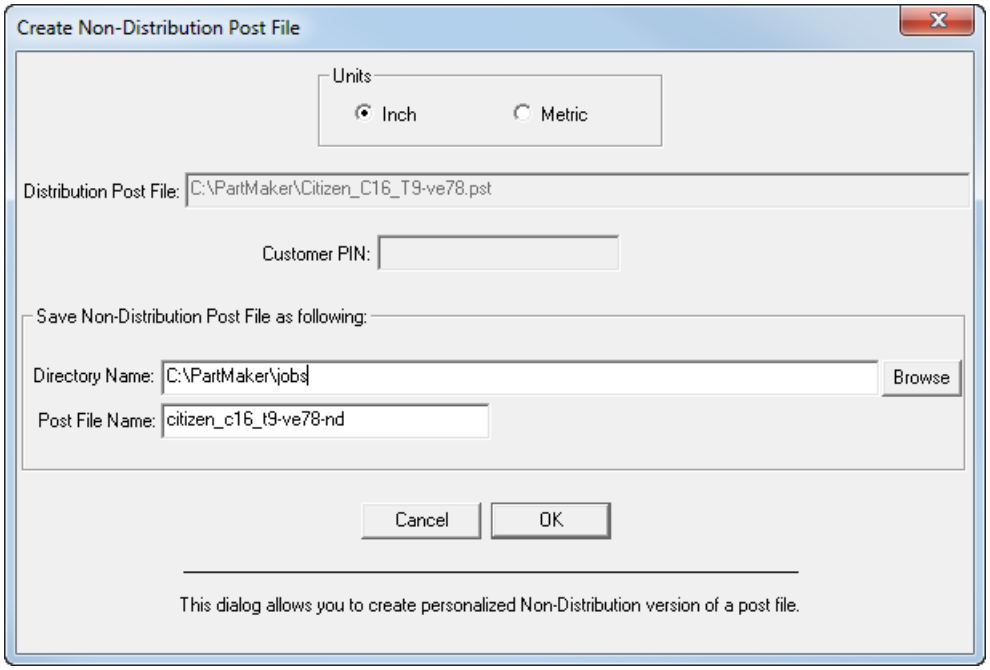
Delete Range — Click to remove an existing range in the **Tool Ranges** list box.

Set Tool Range — These fields specify the start and end of the active tool range selected in the **Tool Ranges** list box

Coolant Codes — These fields contain the on and off coolant codes for specific coolant types. The entire code must be entered in these fields.

Create Non-Distribution Post File

ConfigPost displays the **Create Non-Distribution Post File** dialog when you click the **Create Non-Distribution Post File** button in the **Open Distribution Post File** dialog (see page 109).



To create a Non-Distribution Post file:

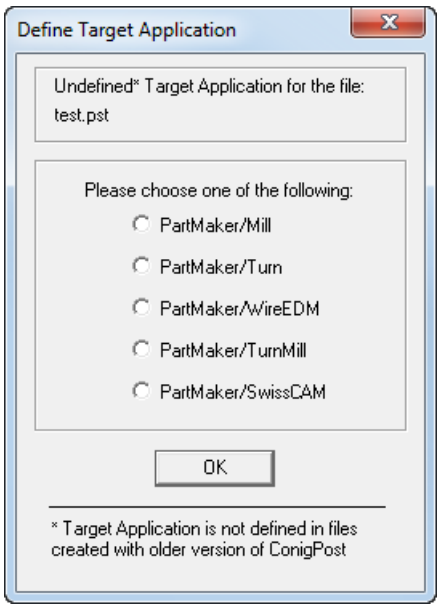
1. Select the **Units** that you want to use in the Post file.
2. Check you have opened the correct Distribution Post file, by checking the following details:
   - **Distribution Post File** displays the directory and filename of the Distribution Post File you opened.
   - **Customer PIN** displays the Customer PIN you entered on the **Open Distribution Post File** dialog (see page 109).

   If the details are incorrect, click **Cancel** to exit from the dialog. If the details are correct, continue to the next step.
3. Check the **Directory Name** where PartMaker will save the Non-Distribution Post file and the **Post File Name** for the file are correct. You can enter a different directory and/or filename, if required.
4. Click **OK**.

   ConfigPost creates a new Non-Distribution Post (.pst) file (and .sub file, if required). It then opens the .pst file in ConfigPost.

Define Target Application dialog

The **Define Target Application** dialog is displayed when you open a Post file that was created with an older version of ConfigPost.



After the target application is defined by choosing a radio button, the target application cannot be changed.

Target application will be saved with the Post file. When this Post file is accessed again, this dialog no longer appears.

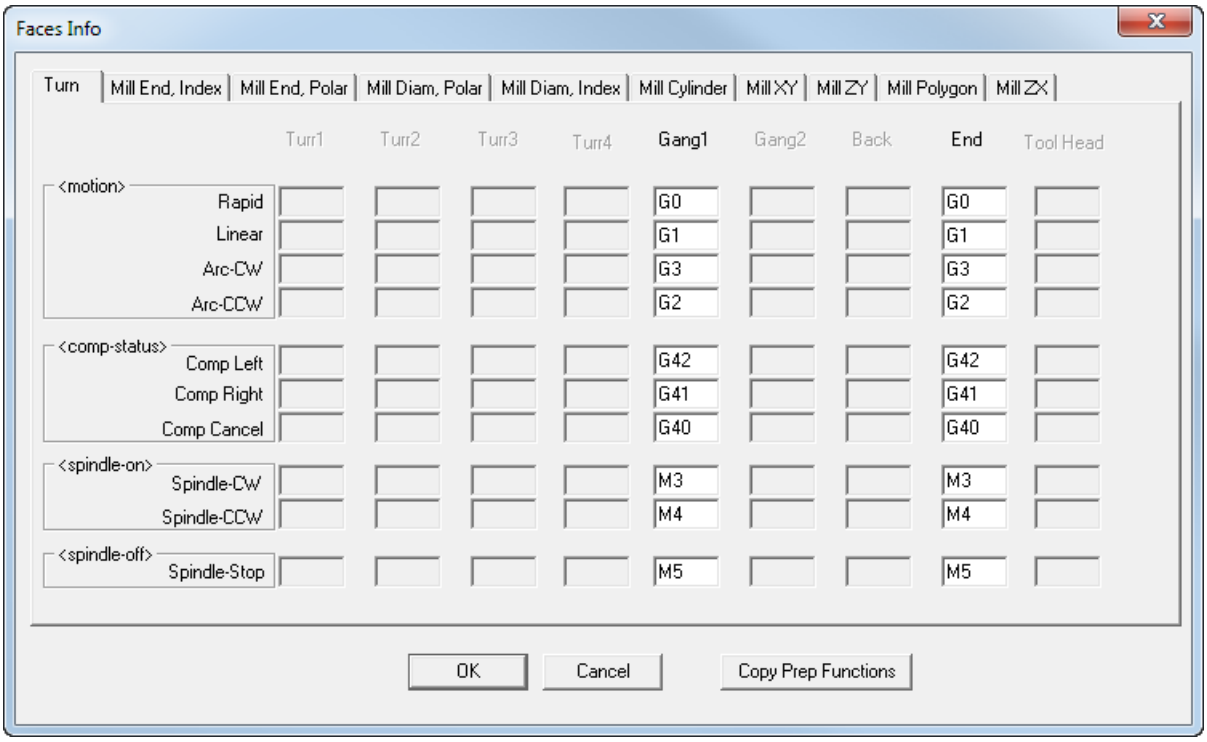
Faces Info dialog

This feature is not used in Basic Post Processors.

Use the **Faces Info** dialog to enter tool post specific G/M codes for your chosen face type.

To display this dialog, select one of the options from the **Settings > Faces Info** submenu.

The fields are available when the **Tools Info** dialog (see page 119) contains valid **Tools Ranges**.



Copy Prep Functions — This option enables you to copy G/M codes from the **Preparatory Functions** dialog (see page 110).

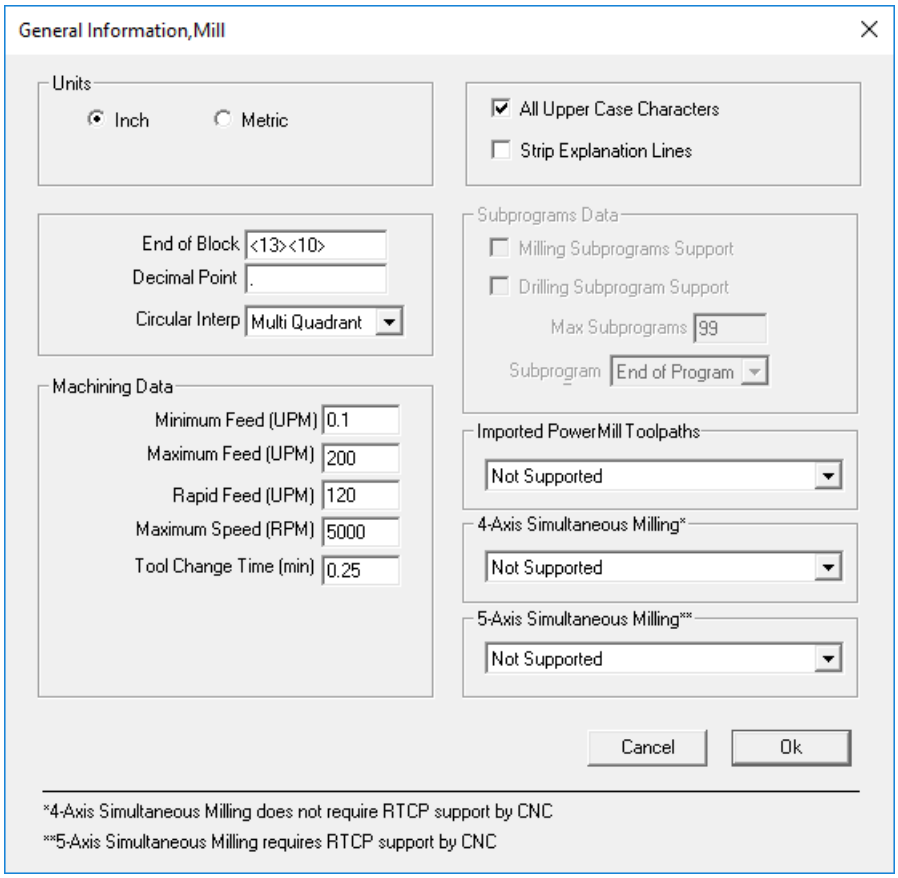
General Information dialog

Use the **General Information** dialog to specify information for a part program and the type of circular interpolation your CNC uses.

General Information dialog - Mill

The **General Information** dialog enables you to specify information for a part program such as machining data, subprograms support for milling and drilling operations (if your CNC is capable) and the type of circular interpolation your CNC uses.

To display this dialog, select **Configure Mill > General Information** from the menu bar.



Units — Select the units.

End of Block — Specify the end of block character sequence.

Decimal Point — Specify the character to be used for decimal points. (For U.S. machines, a period is normally used; for European machines, a comma is often used.)

Circular Interpolation — Select the type of circular interpolation used by your CNC. You can choose *Multi Quadrant* or *Single Quadrant*.

Machining Data

Minimum Feed (UPM) — Specify the minimum feed rate value to be used by your CNC.

Maximum Feed (UPM) — Specify the maximum feed rate value to be used by your CNC.

Rapid Feed (UPM) — Specify the rapid feed rate value to be used by your CNC.

Maximum Speed (RPM) — Specify the maximum spindle speed to be used by your CNC.

Tool Change Time (min) — Specify the tool change time to be used by your CNC.



*Machining data can be viewed in the **Defaults** dialog in PartMaker.*

All Upper Case Characters — Select this option to convert all notes to upper-case characters.

Strip Explanation Lines — Select this option to prevent explanation lines from being written to the NC program file. Explanation lines help improve configuration file documentation. Each explanation line must begin with a semicolon as shown in the following example:

```
;TOOL CHANGE ON THE NEXT LINE<eob> This is an explanation
line. It starts with a semicolon.
T<tool-num><tool-offset>M06<eob>
```

Subprograms Data

These options are applicable only to PartMaker/Mill:

Milling Subprograms Support — Select this option if you are using subprograms in a part program and your CNC supports milling subprograms.

Drilling Subprograms Support — Select this option if you are using subprograms in a part program and your CNC supports drilling subprograms.

Max Subprograms — Specify the maximum number of subprograms that can be used in a part program.

Subprogram Type — Specify the type of subprogram to be used in a part program:

- **Start of Program** — Select to place subprogram definitions at the beginning of the part program file.
- **End of Program** — Select to place subprogram definitions at the end of the part program file. We recommend you use the File End program format at the end of these subprogram definitions.
- **Not Available** — Select to specify that subprograms will not be output for this part program.
- **Individual Files** — Select to specify that each subprogram definition will be placed in an individual file. The name of each file consists of the name you specified in ConfigPost and the subprogram number assigned by the system. For example, FANUC03 for Subprogram 03 FANUC is the main part program name and the number 03 is the (internal) subprogram name. The subprogram name is added to the main part program name as the disk file name.

- **Separate File** — Select to specify that all subprograms will be placed in one file. The name of the subprogram file consists of the output name you specify in ConfigPost (MA represents the subprogram specification) and a file extension of .TXT. For example, the FANUCMA.TXT subprogram file and FANUC.TXT main part program file would be output.

Entering Control Characters

ConfigPost sometimes allows or requires the use of one or more ASCII control characters (see page 195) (non-printable characters). You can embed these control characters in a program format by entering the decimal equivalent of the ASCII code delimited by angular brackets (<>). For example, <13> denotes the ASCII equivalent of a carriage return.

Imported PowerMill Toolpaths

Use this option to specify whether the post processor supports toolpaths that have been imported into PartMaker from PowerMill. This option is available only when using PartMaker/SwissCAM, PartMaker/Turn-Mill, or PartMaker/Mill.

You can select:

- **Supported** — Select this option if the post processor supports the post processing of toolpaths imported from PowerMill.
- **Unsupported** — Select this option if the post processor does not support the post-processing of toolpaths imported from PowerMill.

If a PartMaker job file attempts to post process a toolpath imported from PowerMill when **Unsupported** is selected, PartMaker is unable to post process the toolpath and displays an error message.



*As well as selecting **Supported**, the correct formatting must also be configured to post process toolpaths imported from PowerMill, see Configuring a post processor to support imported PowerMill toolpaths (see page 161) for more details.*

4-Axis Simultaneous Milling

Use this option to specify whether the post processor supports 4-Axis Simultaneous toolpaths. This option is available when using PartMaker/SwissCAM, PartMaker/Turn-Mill and PartMaker/Mill.

You can select:

- **Supported** — Select this option if the post processor supports 4-Axis Simultaneous toolpaths.
- **Not Supported** — Select this option if the post processor does not support 4-Axis Simultaneous toolpath.

5-Axis Simultaneous Milling

Use this option to specify whether the post processor supports 5-Axis Simultaneous toolpaths. This option is available when using PartMaker/SwissCAM, PartMaker/Turn-Mill and PartMaker/Mill.



To use 5-axis simultaneous toolpaths, your CNC machine must support Rotation Tool Center Point (RTCP).

You can select:

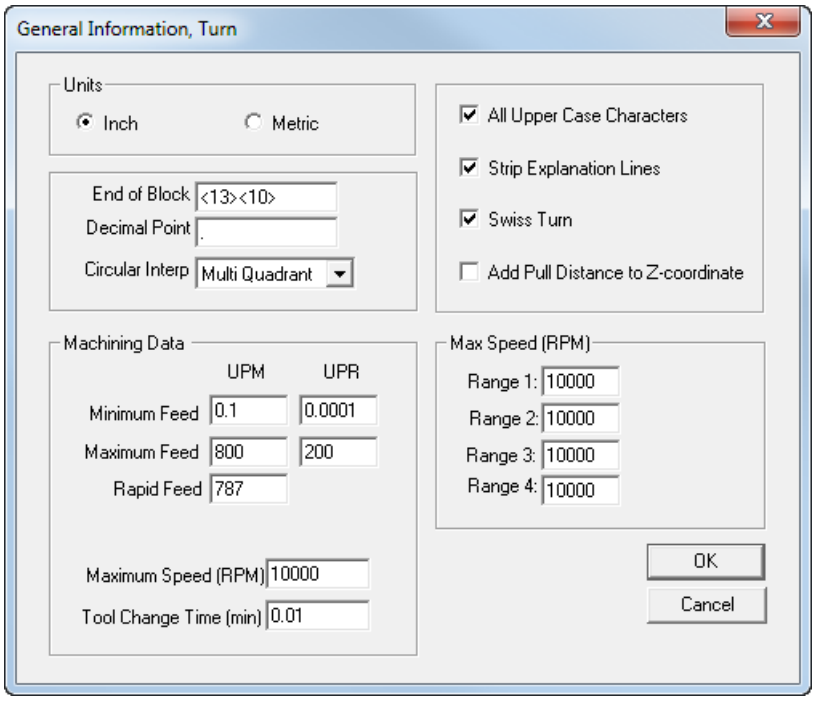
- **Not Supported** — Select this option if the post processor does not support 5-Axis Simultaneous toolpaths.
- **(X,Y, Z) in Rotated Coordinate System** — Select this option if the post processor supports 5-Axis Simultaneous toolpaths and outputs the X,Y coordinates in a coordinate system that rotates about the Z-axis with the stock.
- **(X,Y,Z) in Machine Coordinate System** — Select this option if the post processor supports 5-Axis Simultaneous toolpaths and outputs the X,Y coordinates in the Machine Coordinate system, which does not change with stock rotation.

If a PartMaker job file attempts to post process a 5-axis toolpath when **Unsupported** is selected, PartMaker is unable to post process the toolpath and displays an error message.

General Information dialog - Turn

The **General Information** dialog enables you to specify information for a part program such as machining data, subprograms support for milling and drilling operations (if your CNC is capable) and the type of circular interpolation your CNC uses.

To display this dialog, select **Configure Turn > General Information** from the menu bar.



Units — Select the units.

End of Block — Specify the end of block character sequence.

Decimal Point — Specify the character to be used for decimal points. (For U.S. machines, a period is normally used; for European machines, a comma is often used.)

Circular Interpolation — Select the type of circular interpolation used by your CNC. You can choose *Multi Quadrant* or *Single Quadrant*.

Machining Data

Minimum Feed — Enter the minimum programmable feed rate in UPM and UPR.

Maximum Feed — Enter the maximum programmable feed rate in UPM and UPR.

Maximum Speed — Enter the maximum programmable spindle speed.

Rapid Feed — Enter the maximum rapid traverse rate.

Tool Change Time (min) — Enter the tool change time (in minutes) to be used by your CNC.



*Machining data can also be viewed in the **Defaults** dialog in PartMaker.*

All Upper Case Characters — Select this option to convert all notes to upper-case characters.

Strip Explanation Lines — Select this option to prevent explanation lines from being written to the NC program file. Explanation lines help improve configuration file documentation. Each explanation line must begin with a semicolon as shown in the following example:

```
;TOOL CHANGE ON THE NEXT LINE<eob>
T<tool-num><tool-offset>M06<eob>
```

Swiss Turn — Select this option if you are specifying information for Swiss-type machines. Checking this box indicates to PartMaker to begin reading a post configuration file in the Collet Re-Chuck,Turn section located under Other Formats.

Add Pull Distance to Z-coordinate — When using Material Control Process "Pull with Sub-Spindle", selecting this option updates all Z-related words (`<z-coord>`, `<depth-abs>`, etc.) for all consecutive processes on the main spindle by adding the "Pull Distance(Zp)".

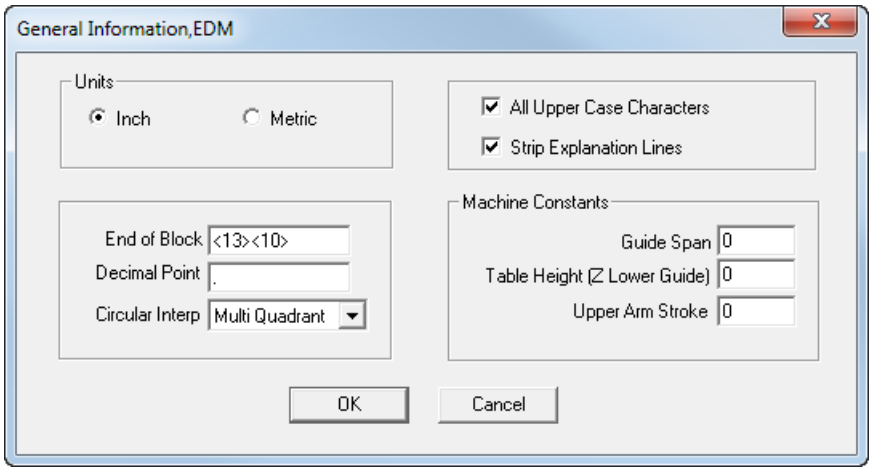
Max. Speed (RPM) — Range 1-4. These options represent the maximum attainable spindle speeds in each of the lathe's gear ranges. If a range is not available, set the value to zero.

Entering Control Characters — ConfigPost sometimes allows - or requires - the use of one or more ASCII control characters (non-printable characters). You can embed these control characters in a program format by entering the decimal equivalent of the ASCII code delimited by angular brackets (`<>`). For example, `<13>` denotes the ASCII equivalent of a carriage return.

General Information dialog - WireEDM

The **General Information** dialog enables you to specify information for a part program, such as machining data and the type of circular interpolation your CNC uses.

To display this dialog, select **Configure EDM > General Information** from the menu bar.



Units — Select the units.

End of Block — Enter the end of block character sequence.

Decimal Point — Enter the character to be used for decimal points. (For U.S. machines, a period is normally used; for European machines, a comma is often used).

Circular Interpolation — Select the type of circular interpolation used by your CNC. You can choose *Multi Quadrant* or *Single Quadrant*.

Machining Constants

Guide Span — Enter the (Z) distance measured between the upper and lower guides.

Table Height (Z Lower Guide) — Enter the distance from the work table down to the lower guide.

Upper Arm Stoke — Enter the maximum travel distance of upper guide.

All Upper Case Characters — Select this option to convert all notes to upper-case characters.

Strip Explanation Lines — Select this option to prevent explanation lines from being written to the NC program file.

Explanation lines help improve configuration file documentation. Each explanation line must begin with a semicolon as shown in the following example:

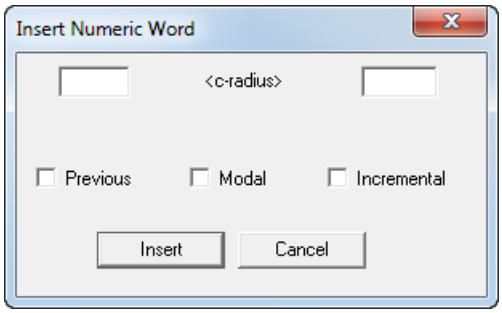
```
;TOOL CHANGE ON THE NEXT LINE<eob> This is an explanation
line. It starts with a semicolon.
T<tool-num><tool-offset>M06<eob>
```

Entering Control Characters — ConfigPost sometimes allows or requires the use of one or more ASCII control characters (non-printable characters). You can embed these control characters in a program format by entering the decimal equivalent of the ASCII code delimited by angular brackets (<>). For example, <13> denotes the ASCII equivalent of a carriage return.

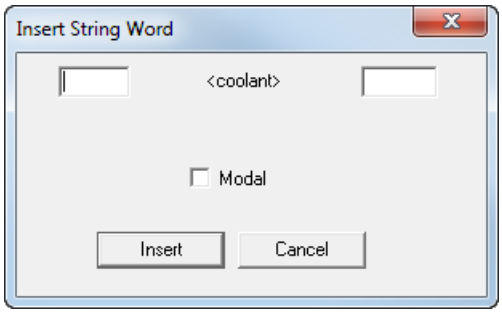
Insert Word dialog

To insert a reserved word in the ConfigPost window, double-click the word in the word list on the right side of the ConfigPost Window, or select a word and choose **Selected Word** from the **Edit > Insert** menu.

1. If you select a numeric reserved word, the **Insert Numeric Word** dialog appears. You can insert the word as a modal or incremental word, or use the value of the previous reserved word.



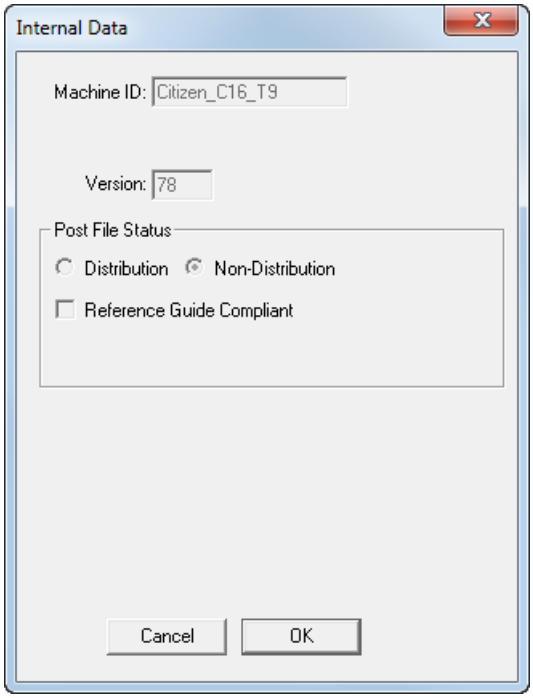
2. If you select a logical reserved word, the selected word is inserted at the current cursor location.
3. If you select a string reserved word, the **Insert String Word** dialog appears. You can insert the word "as is" or as a modal word.



Internal Data dialog

The **Internal Data** dialog enables you to view the Post file specific information.

To display this dialog, select **Settings > Internal Data** from the menu bar. The fields in this dialog are always disabled.



Machine ID — Shows the internal name of the CNC Machine.

Version — Shows the internal version of the Post file.

Post File Status

Distribution — Shows that the Post file was distributed by Autodesk.

Non-Distribution — Shows that the Post file contains customer-specific information and cannot be redistributed.

Reference Guide Compliant — Shows that the content in the Reference Guide generated by the post processor matches the capabilities of the post processor and the machine.

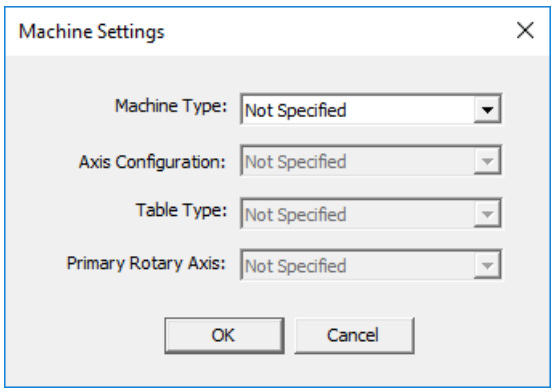
Naming convention for Post files

When you create, modify, or save Post files in ConfigPost the character 'e' is appended after 'v' in the Post file name. For example, Citizen_M3_32-ve122.pst.

If the post is Reference Guide Compliant, the character ‘c’ is added after the version number of the post. For example, Citizen_M3_32-ve122c.pst.

Machine Settings dialog

Use the **Machine Settings** dialog to select your machine type, axis configuration, table type, and primary rotary axis.

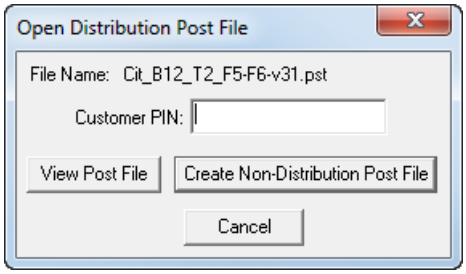


To display this dialog, select **Settings > Machine Settings** from the **Main** menu when the **Target Application** is set to PartMaker/Mill.

Select **Vertical Mill-Turn** from the **Machine Type** list to ensure the **Configure Turn** menu is available on the menu bar.

Open Distribution Post File

This dialog is displayed when you open a Distribution Post file in ConfigPost or enable Config-by-Click (see page 188) in PartMaker when a Distribution Post file is loaded.



Complete the dialog as follows:

Customer PIN — Enter the customer PIN that was provided with your PartMaker installation. ConfigPost uses the PIN to ensure that only information relevant to your organization is displayed in the Post file.

View Post File — Click this button to view the Distribution Post file within ConfigPost. You cannot modify a Distribution Post file.

Create Non-Distribution Post File — Click this button to create a Non-Distribution Post file from this Post file. You can modify a Non-Distribution file to suit the needs or your organization. ConfigPost displays the **Create Non-Distribution Post File** dialog (see page 95).

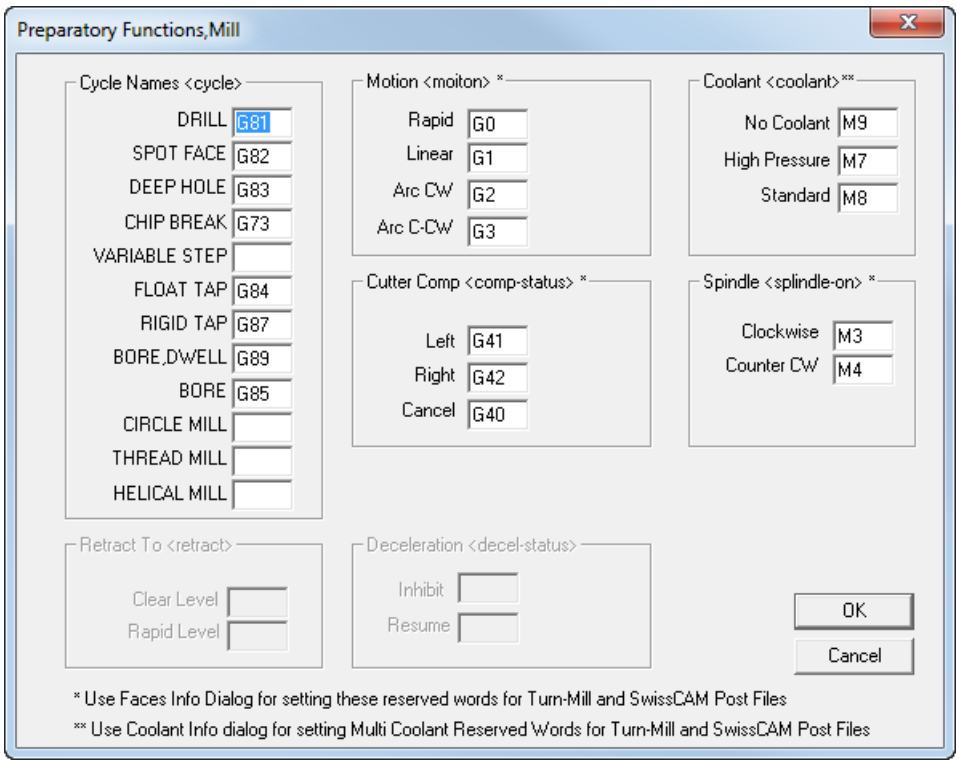
Preparatory Functions dialog

The **Preparatory Functions** dialog allows you to specify machining codes for preparatory functions, such as motion formats and cutter compensation.

Preparatory Functions dialog - Mill

Use the **Preparatory Functions,Mill** dialog to specify machining codes to be used in the part program for cycle names, motion formats, cutter compensation, coolant type, and spindle direction. Each string reserved word shown in the illustration is replaced by the specified G or M machining code when the part program is created.

To display the **Preparatory Functions,Mill** dialog, select **Configure Mill > Preparatory Functions** from the menu bar.



Cycle Names — Specifies the codes required by your CNC for machining cycles in the part program. These codes replace the string reserved word `<cycle>`.

Retract to — Specifies the codes required by your CNC for retract/return in the part program.

- Clear Level
- Rapid Level

These codes replace the string reserved word `<retract>`.

Motion — Specifies the codes required by your CNC for motion in the part program:

- Rapid
- Linear
- Arc CW
- Arc C-CW

These codes replace the string reserved word `<motion>`.

Cutter Comp — Specifies the codes required by your CNC for cutter diameter compensation in the part program:

- Left
- Right
- Cancel

These codes replace the string reserved word `<comp-status>`.

Deceleration — Specifies the codes to be used to inhibit deceleration, then resume later. The Inhibit code is used for tangent locations where deceleration is not necessary.

Coolant — Specifies the codes required by your CNC for coolant type in the part program:

- No Coolant
- High Pressure
- Standard

These codes replace the string reserved word `<coolant>`.

Spindle — Specifies the codes required by your CNC for spindle rotation in the part program:

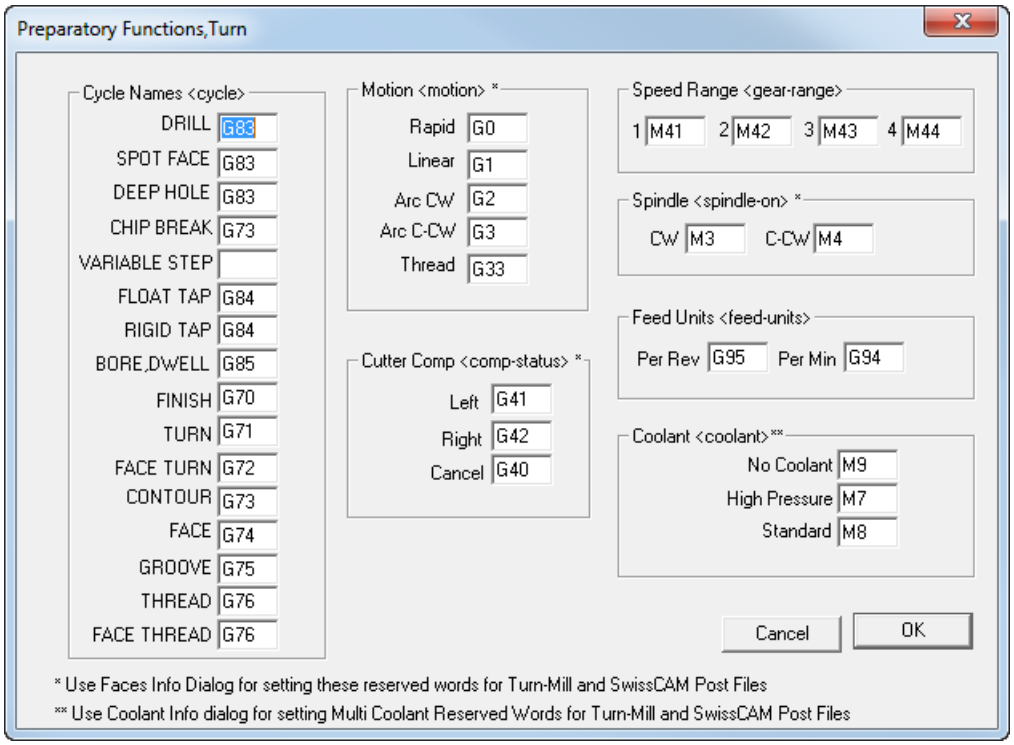
- Clockwise
- Counter CW

These codes replace the string reserved word `<spindle>`.

Preparatory Functions dialog - Turn

Use the **Preparatory Functions,Turn** dialog to specify machining codes to be used in the part program for cycle names, motion formats, cutter compensation, coolant type, and spindle direction. Each string reserved word shown in the illustration is replaced by the specified G or M machining code when the part program is created.

To display the **Preparatory Functions,Turn** dialog, select **Configure Turn > Preparatory Functions** from the menu bar.



Cycle Names — Specifies the codes required by your CNC for machining cycles in the part program.

These codes replace the string reserved word `<cycle>`.

Motion — Specifies the codes required by your CNC for motion in the part program:

- Rapid
- Linear
- Arc CW
- Arc C-CW
- Thread

These codes replace the string reserved word `<motion>`.

Cutter Comp — Specifies the codes required by your CNC for cutter diameter compensation in the part program:

- Left
- Right
- Cancel

These codes replace the string reserved word `<comp-status>`.

Speed Range — These codes program the spindle gearbox to the appropriate gear to match the required spindle speed.

These codes replace:

- Range 1 string reserved words
- Range 2 string reserved words
- Range 3 string reserved words
- Range 4 string reserved words

Spindle — Specifies the codes required by your CNC for spindle rotation in the part program:

- Clockwise
- Counter CW

These codes replace the string reserved word `<spindle>`.

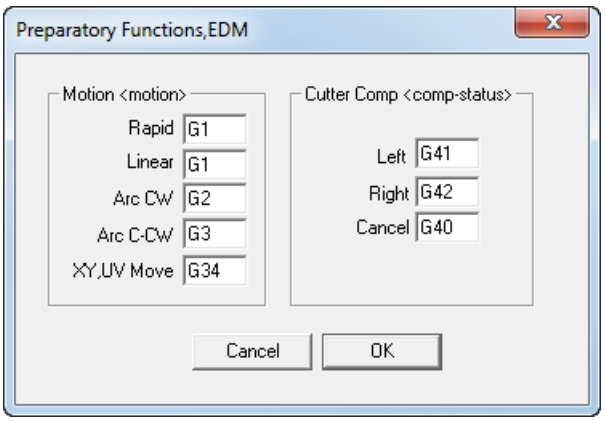
Feed Units — Specifies the codes required by your CNC for feed units per revolution and per minute.

Coolant — Codes used to activate the coolant.

Preparatory Functions dialog - WireEDM

Use the **Preparatory Functions,EDM** dialog to specify machining codes to be used in the part program for motion formats and cutter compensation. Each string reserved word shown in the illustration is replaced by the specified G or M machining code when the part program is created.

To display the **Preparatory Functions,EDM** dialog, select **Configure EDM > Preparatory Functions** from the menu bar.



Motion — Specifies the codes required by your CNC for motion in the part program:

- Rapid
- Linear
- Arc CW
- Arc C-CW

- XY,UV Move

These codes replace the string reserved word <motion>.

Cutter Comp — Specifies the codes required by your CNC for wire diameter compensation in the part program:

- Left
- Right
- Cancel

These codes replace the string reserved word <comp-status>.

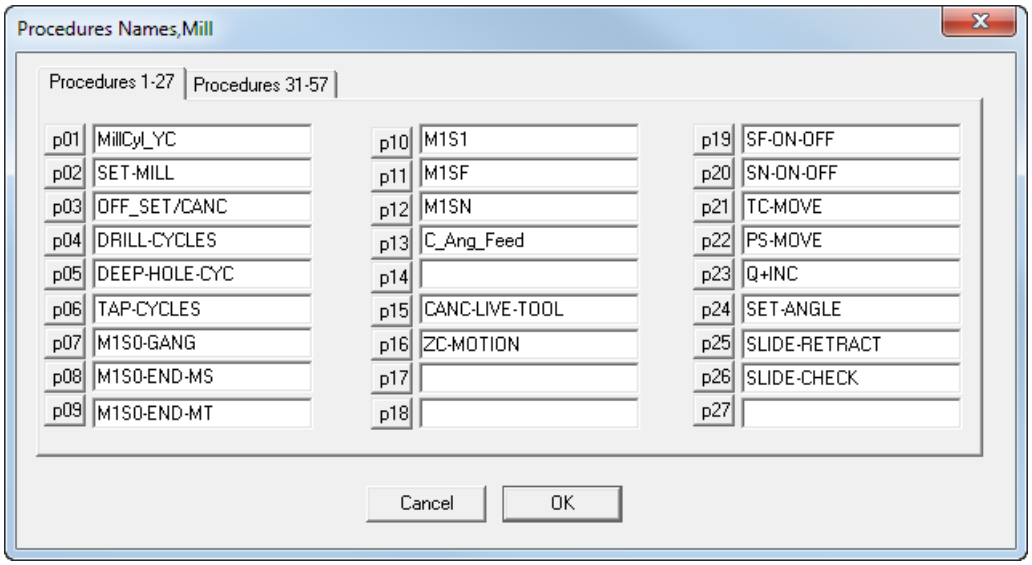
Procedure Names dialog

The **Procedure Names** dialog enables you to name procedures.

Procedure Names dialog - Mill

Use the **Procedure Names,Mill** dialog to name procedures for use in the Milling section of ConfigPost.

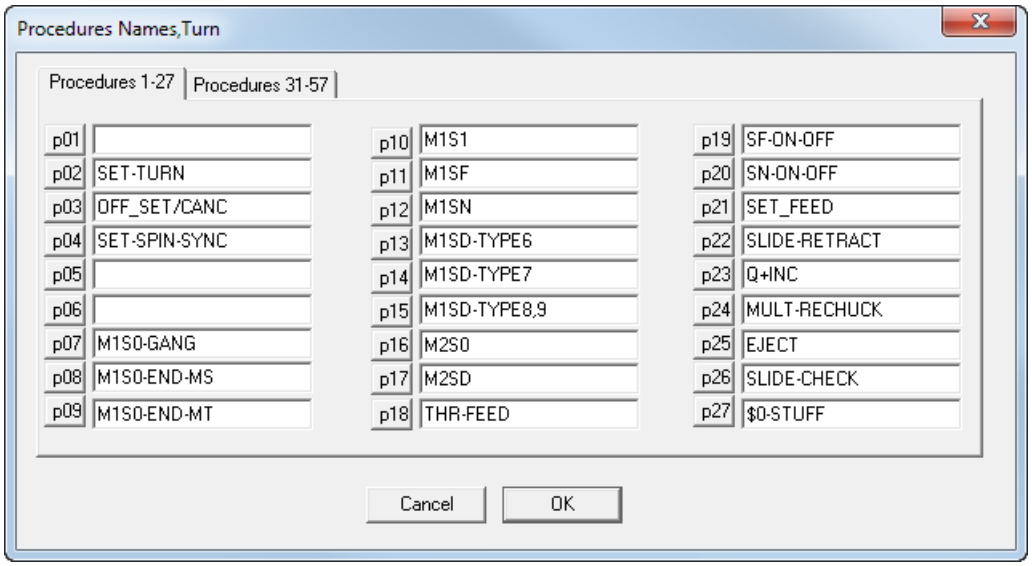
To display this dialog, select **Settings > Procedure Names > Milling Procedures** from the menu bar.



Procedure Names dialog - Turn

Use the **Procedures Names,Turn** dialog to name procedures for use in the PartMaker/Turn section of ConfigPost.

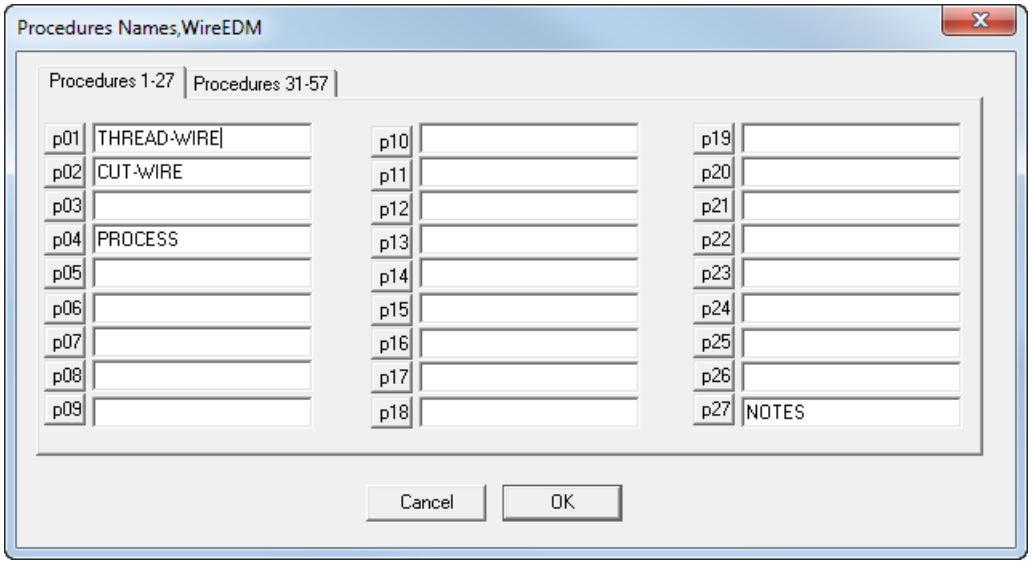
To display this dialog, select **Settings > Procedure Names > Turning Procedures** from the menu bar.



Procedure Names dialog - WireEDM

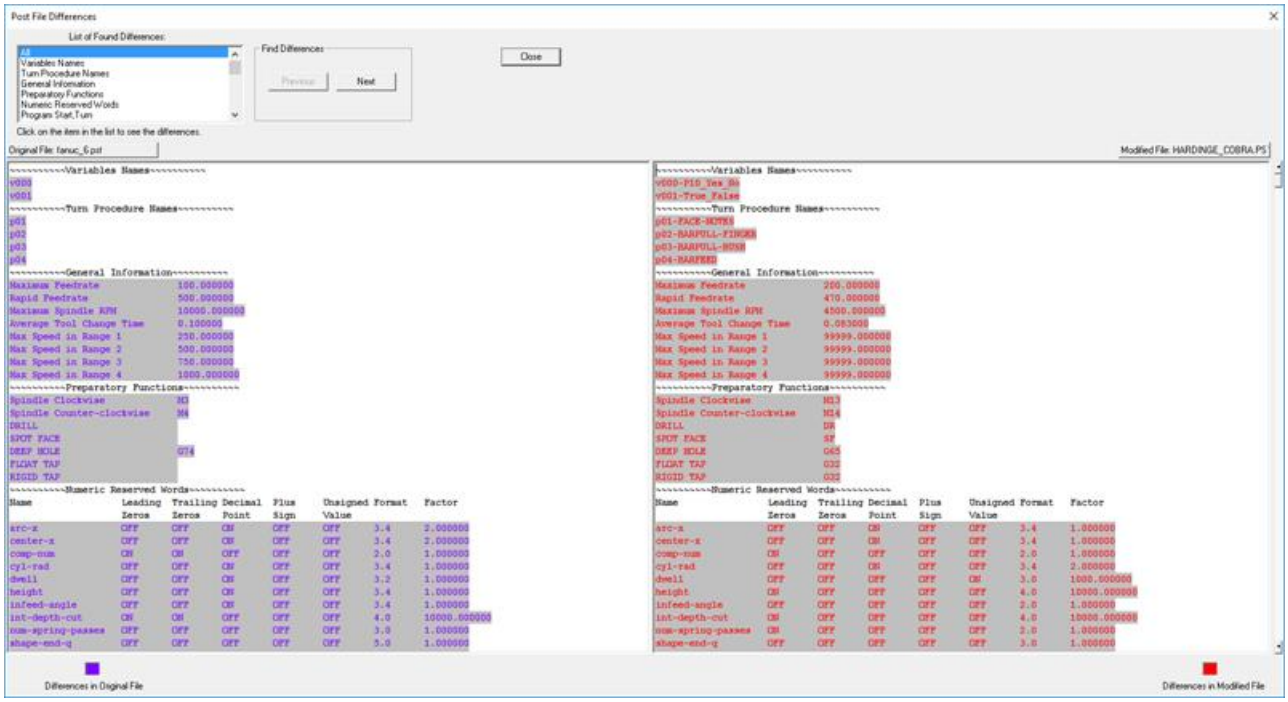
Use the **Procedures Names,WireEDM** dialog to name procedures for use in the EDM section of ConfigPost.

To display this dialog, select **Settings > Procedure Names > EDM Procedures** from the menu bar.



Post File Differences dialog

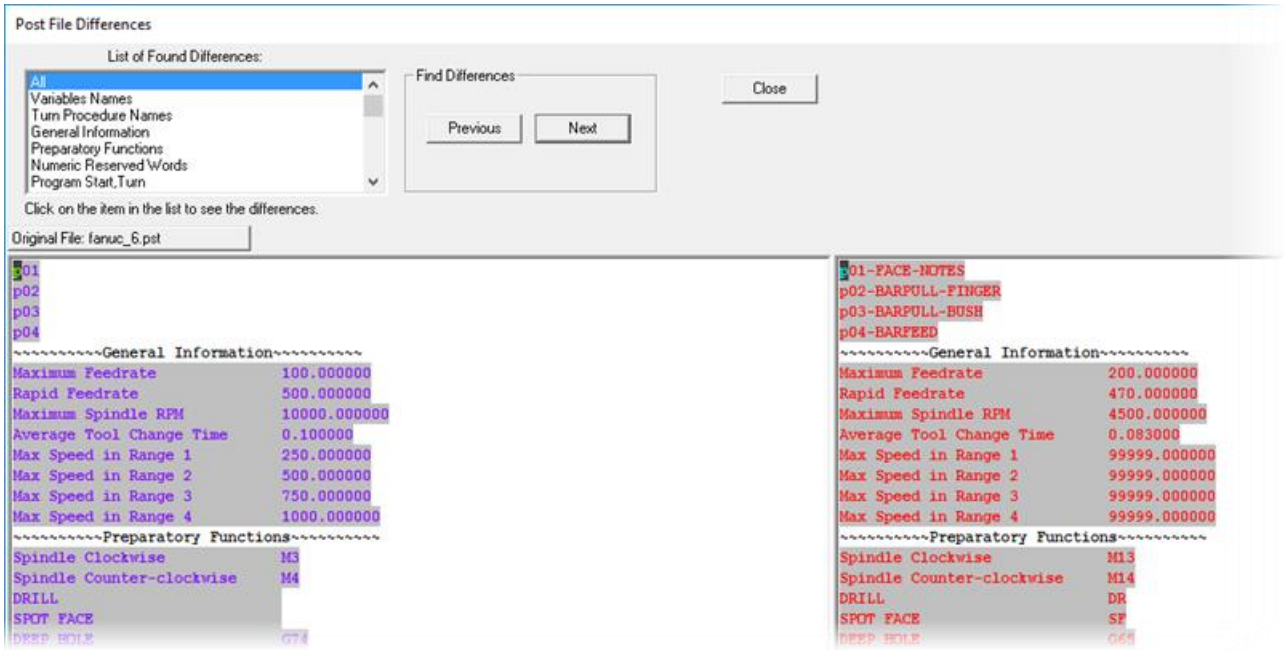
To display the **Post File Differences** dialog, click the **Compare** button in the **Compare Files** dialog (see page 93).



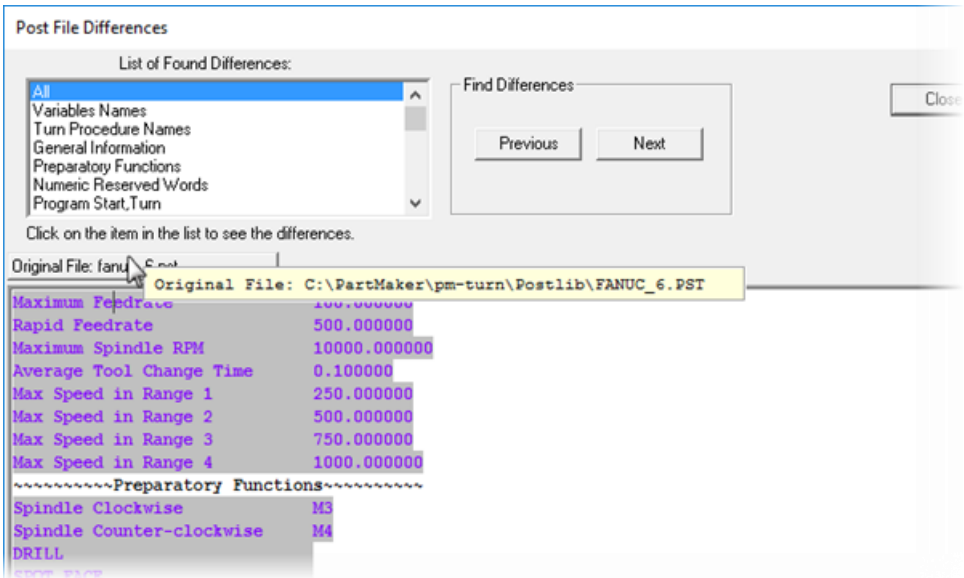
List of Found Differences — Displays the list of all program formats where differences between 2 files were found.

You can click on each individual program format, or you can click on "All" to see differences in all program formats.

Next/Previous — These buttons enable you to jump to the next or previous difference. The first character is highlighted in the window where differences are found.



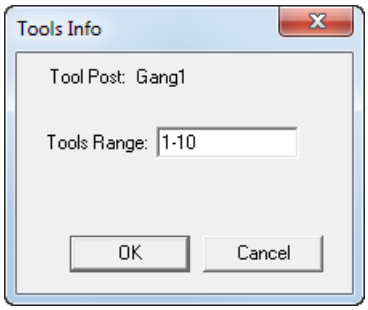
If you position the cursor over the Original File or Modified File window headers, the full file path is displayed.



Tools Info dialog

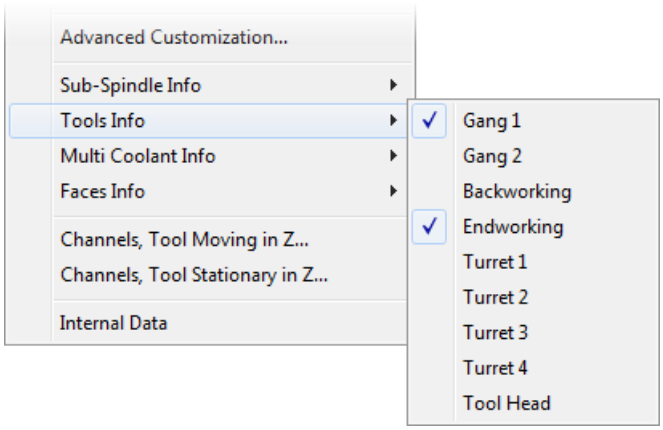
This feature is not used in Basic Post Processors.

Use the **Tools Info** dialog to enter the **Tools Range** for a specific tool post.



To display this dialog, select one of the following options from the **Settings > Tools Info** menu.

When there is a value in the **Tools Range** field, the tool post is marked with a tick in the **Tools Info** menu.



User Data Names dialog

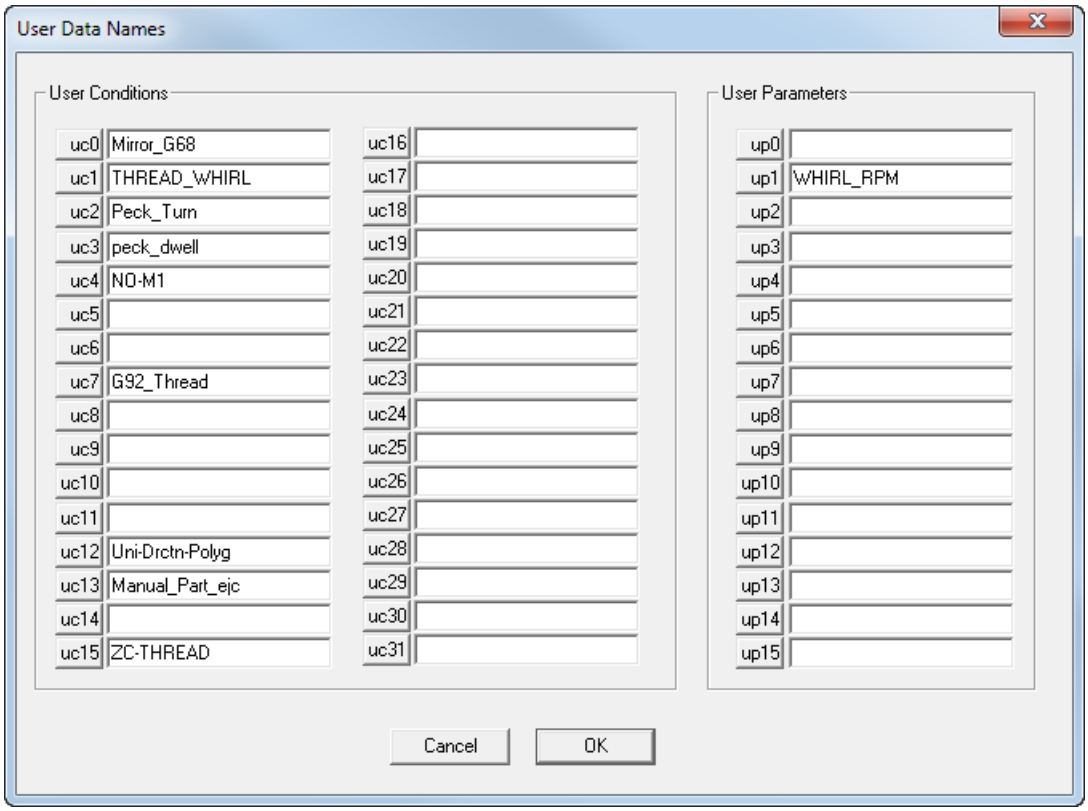
Use the **User Data Names** dialog to add your own logical reserved words (called **User Conditions**) and string reserved words (called **User Parameters**) to a Post file.

When the Post file is selected in PartMaker, users can use the **User Data** dialog to activate User Conditions and assign values to User Parameters.

Some of the Post files provided with your PartMaker installation may already include User Data.

To create User Data using the **User Data Names** dialog:

1. In ConfigPost, select **Settings > User Data Names** to display the **User Data Names** dialog:

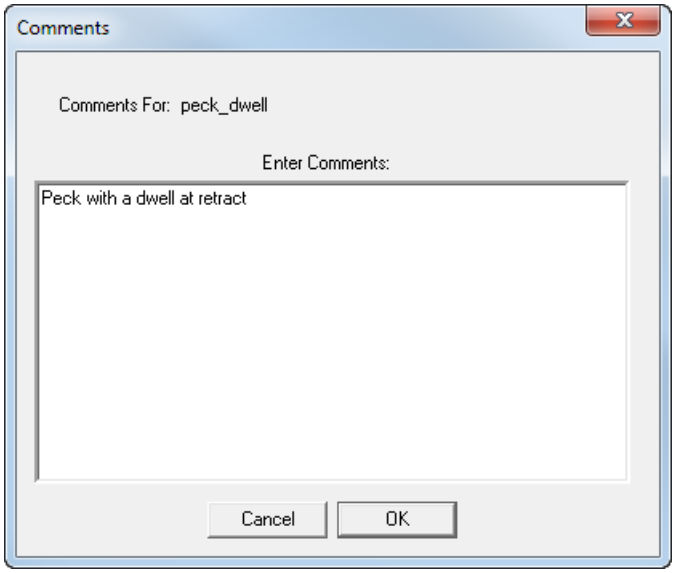


2. Enter the names of the User Conditions and User Parameters you want to create. You can enter up to 31 User Conditions and 15 User Parameters.

   The names of User Conditions and User Parameters must not contain blank spaces.

3. To enter comments for a User Condition or User Parameter, click the button alongside its name to display the **Comments** dialog. PartMaker users can view these comments when using the **User Data** dialog in PartMaker.

For example, to enter a comment for User Condition 3, click uc3 and then enter the comment in the **Comments** dialog:

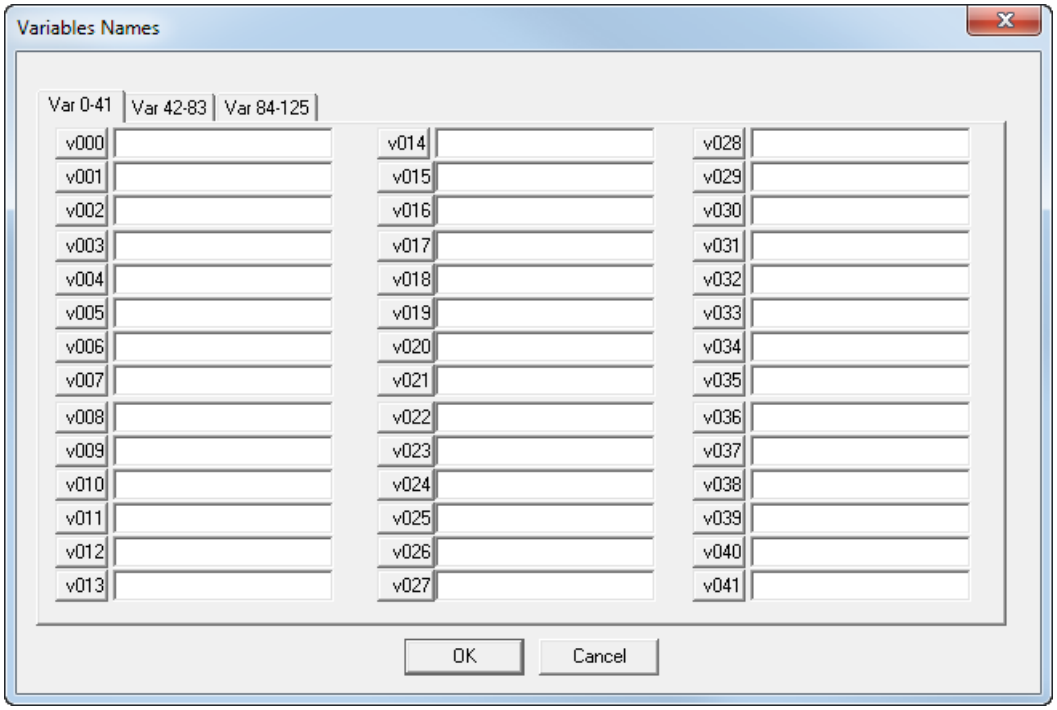


4. When you have entered the comment, click **OK** to return to the **User Data Names** dialog.
5. Repeat steps 2–4 for each User Condition and User Parameter you want to create.
6. When you have created all the User Conditions and User Parameters you require, click **OK**.

Variables Names dialog

Use the **Variables Names** dialog to enter the names of user-defined variables. For Turn-Mill and SwissCAM Post files, the same variables can be used either in the Milling or Turning section of ConfigPost.

To display the **Variables Names** dialog, select **Settings > Variables Names** from the menu bar.



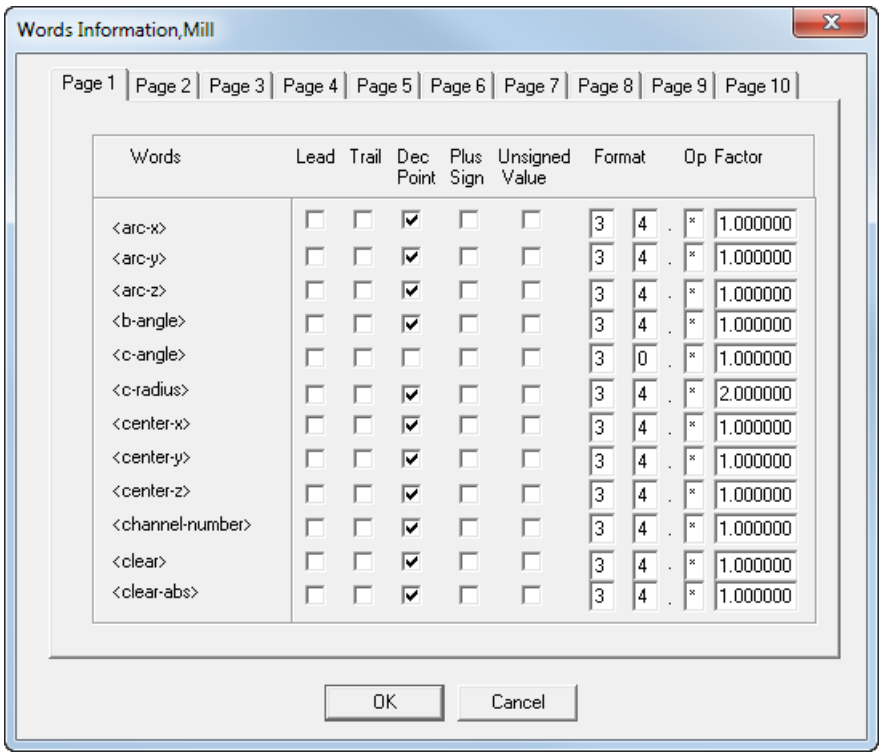
Words Information dialog

The **Words Information** dialog allows you to specify the information for a reserved word that affects its appearance in a part program.

Words Information dialog - Mill

The **Words Information** dialog has multiple pages; you can view any page by clicking its corresponding check box on the right side of the dialog. Use the check boxes and text entry areas in the table to specify the format of any reserved word.

To display this dialog, select **Configure Mill > Words Information** from the menu bar.



Page — Specifies the page of reserved words you're currently viewing. The words are displayed in the same order they appear in the **ConfigPost** window. There are multiple pages of reserved words information.

Leading Zeros — When on, specifies the output format of leading zeros. For example, 2.5 with three leading zeros and with a decimal point would display as 002.5 (or 0025 without a decimal point).

Trailing Zeros — When on, specifies output of trailing zeros. For example, 2.5 with four trailing zeros and with a decimal point would display as 2.5000 (or 25000 without a decimal point).

Decimal Point — When on, uses a decimal point for a value. (See the two previous examples.)

Plus Sign — When on, outputs positive values with plus signs (+), +2.5 for example.

Unsigned Value — When on, always generates positive values.

Format — Specifies the number of digits for the numeric value represented by the reserved word. The first value specifies the maximum number of digits to the left of the decimal point; the second value specifies the maximum number of digits to the right of the decimal point.

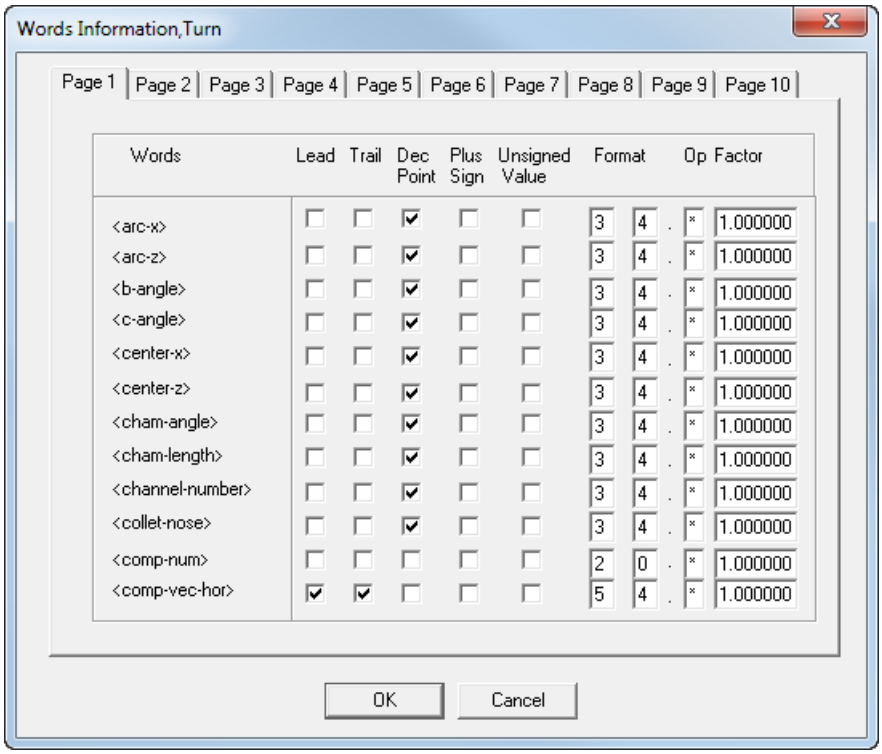
Op — Specifies a mathematical operation. **Op** can be any of the following: +, –, or *.

Factor — Applies **Op** to the value by the specified factor during post processing.

Words Information dialog - Turn

This dialog appears when you choose **Words Information** from the **Configure Turn** menu.

The **Words Information** dialog has multiple pages; you can view any page by clicking its corresponding check box on the right side of the dialog. Use the check boxes and text entry areas in the table to specify the format of any reserved word.



Page — Specifies the page of reserved words you're currently viewing. The words are displayed in the same order they appear in the ConfigPost window. There are six pages of reserved words information.

Leading zeros — When on, specifies the output format of leading zeros. For example, 2.5 with three leading zeros and with a decimal point would display as 002.5 (or 0025 without a decimal point).

Trailing zeros — When on, specifies output of trailing zeros. For example, 2.5 with four trailing zeros and with a decimal point would display as 2.5000 (or 25000 without a decimal point).

Decimal Point — When on, uses a decimal point for a value. (See the two previous examples.)

Plus Sign — When on, outputs positive values with plus signs (+), +2.5 for example.

Unsigned Value — When on, always generates positive values.

Format — Specifies the number of digits for the numeric value represented by the reserved word. The first value specifies the maximum number of digits to the left of the decimal point; the second value specifies the maximum number of digits to the right of the decimal point.

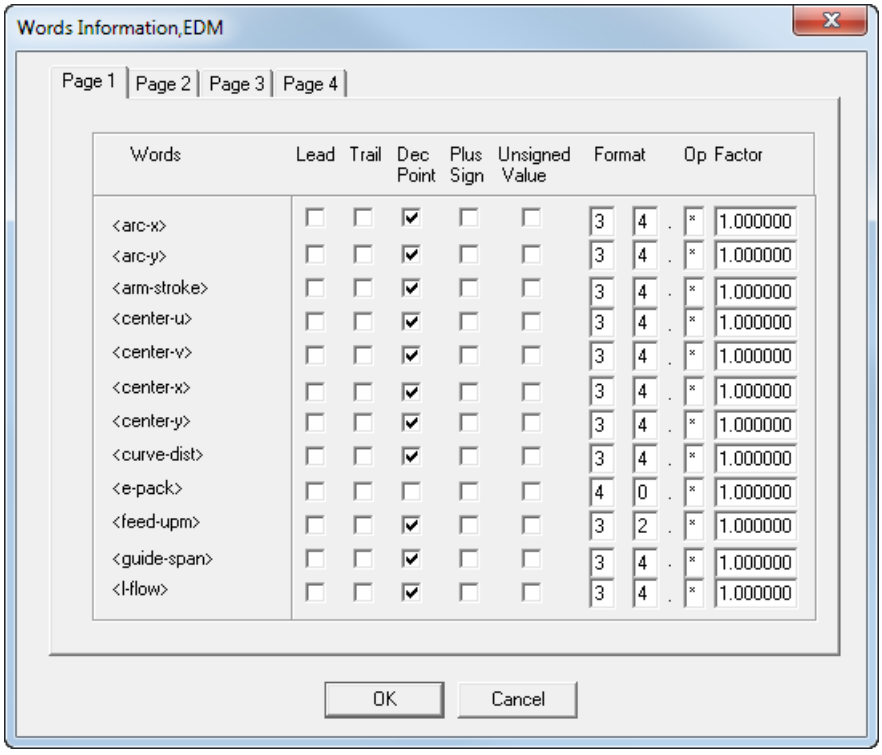
Op — Specifies a mathematical operation. **Op** can be any of the following: +, –, or *.

Factor — Applies **Op** to the value by the specified factor during post processing.

Words Information dialog - EDM

This dialog appears when you choose **Words Information...** from the **Configure EDM** menu.

The **Words Information** dialog has multiple pages; you can view any page by clicking its corresponding check box on the right side of the dialog. Use the check boxes and text entry areas in the table to specify the format of any reserved word.



Page — Specifies the page of reserved words you're currently viewing. The words are displayed in the same order they appear in the ConfigPost window. There are multiple pages of reserved words information.

Leading Zeros — When on, specifies the output format of leading zeros. For example, 2.5 with three leading zeros and with a decimal point would display as 002.5 (or 0025 without a decimal point).

Trailing Zeros — When on, specifies output of trailing zeros. For example, 2.5 with four trailing zeros and with a decimal point would display as 2.5000 (or 25000 without a decimal point).

Decimal Point — When on, uses a decimal point for a value. (See the two previous examples.)

Plus Sign — When on, outputs positive values with plus signs (+), +2.5 for example.

Unsigned Value — When on, always generates positive values.

Format — Specifies the number of digits for the numeric value represented by the reserved word. The first value specifies the maximum number of digits to the left of the decimal point; the second value specifies the maximum number of digits to the right of the decimal point.

Op — Specifies a mathematical operation. **Op** can be any of the following: +, –, or *.

Factor — Applies **Op** to the value by the specified factor during post processing.

Advanced Features

This section describes some of ConfigPost's advanced features:

- Assigning Values to Reserved Words (see page 129)
- Explanation Lines (see page 130)
- Using ConfigPost’s File Handling Features (see page 131)
- Separating Lines Into Specific Files (see page 133)
- Using Variables (see page 135)
- Using <include> reserved word (see page 137)
- Performing Math Operations (see page 139)
- Using Procedures (see page 140)
- User Data Names (see page 120)
- Import/Export WRI files (see page 147)

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>
```



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

Example of an explanation line is shown below:

```
;This is an explanation line, it starts with a semicolon
<eob>
```

During Post Processing PartMaker ignores explanations if the check box "Strip Explanation Lines" is checked in the **General Information,Mill** (see page 98) or **General Information,Turn** (see page 102) or **General Information,WireEDM** dialog (see page 104).

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 can 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>
```

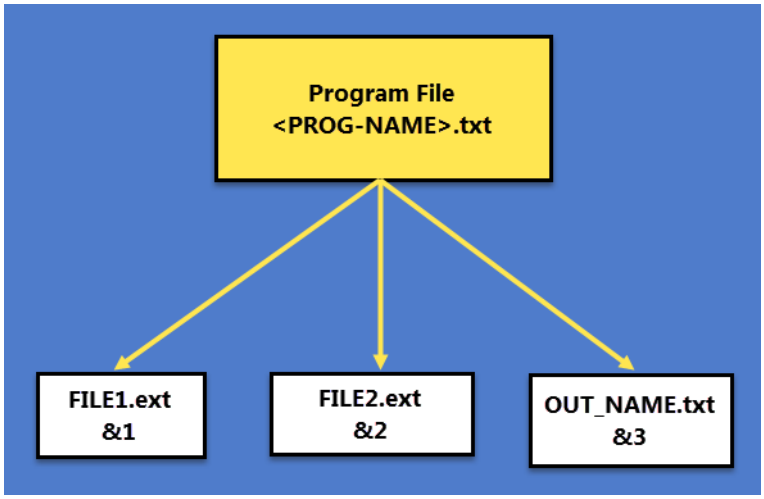
The four lines shown above must be placed in the File Start section of your Post file.

`MACHINE NAME` can be any name that you choose.

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.



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

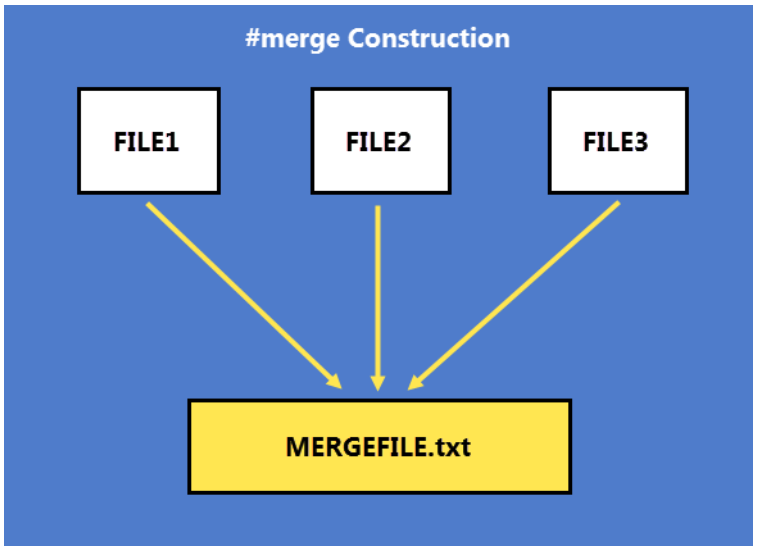
OUT_NAME.TXT is a special file name that must be created any time the split/merge function is used.



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 information only, and will be displayed on the screen when post processing is completed.

#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.



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.

Example - How to create 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:

1 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.

2 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.

3 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`.

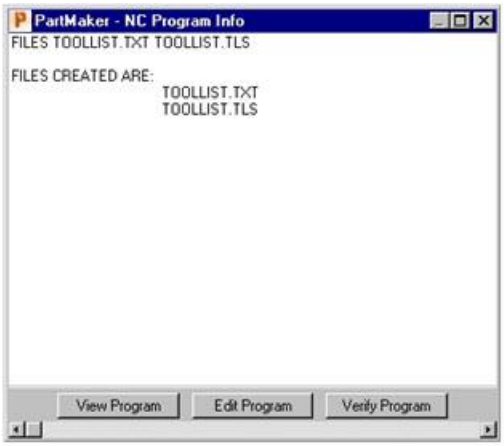
4 To view a post processed JOB file that outputs the two files constructed above:

a In ConfigPost, save the Post file you just modified by choosing **File > Save As**, then name the file `TOOLLIST.PST` or a similar name when prompted.

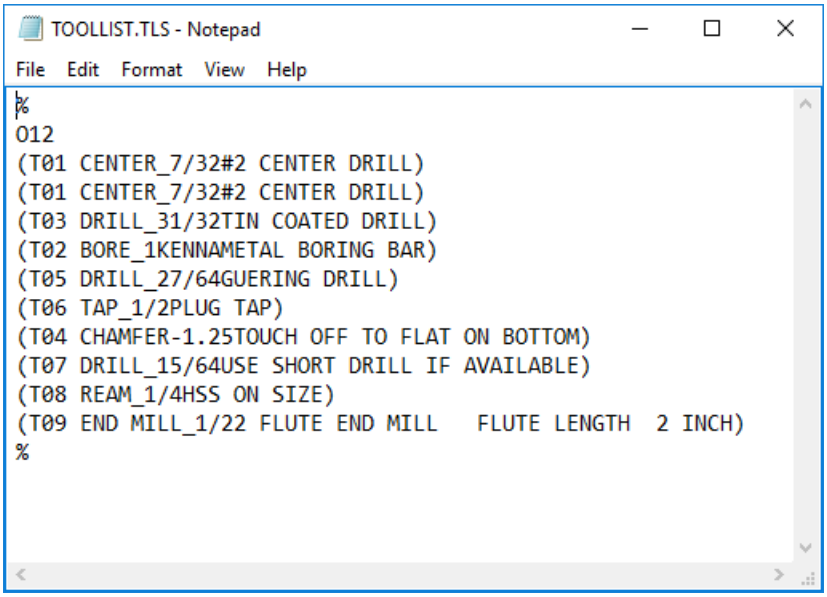
b In PartMaker, load the Post file you just modified by choosing **Job Optimizer > Post Config File=?**.

c In PartMaker, open a completed .JOB file and choose **Job Optimizer > Generate NC Program**.

d When prompted, name your program TOOLLIST.TXT, and then click **Save**. You should see the contents of the OUT_NAME.TXT file as:



e Click the **View Program** button and you should see the Tool List as:



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

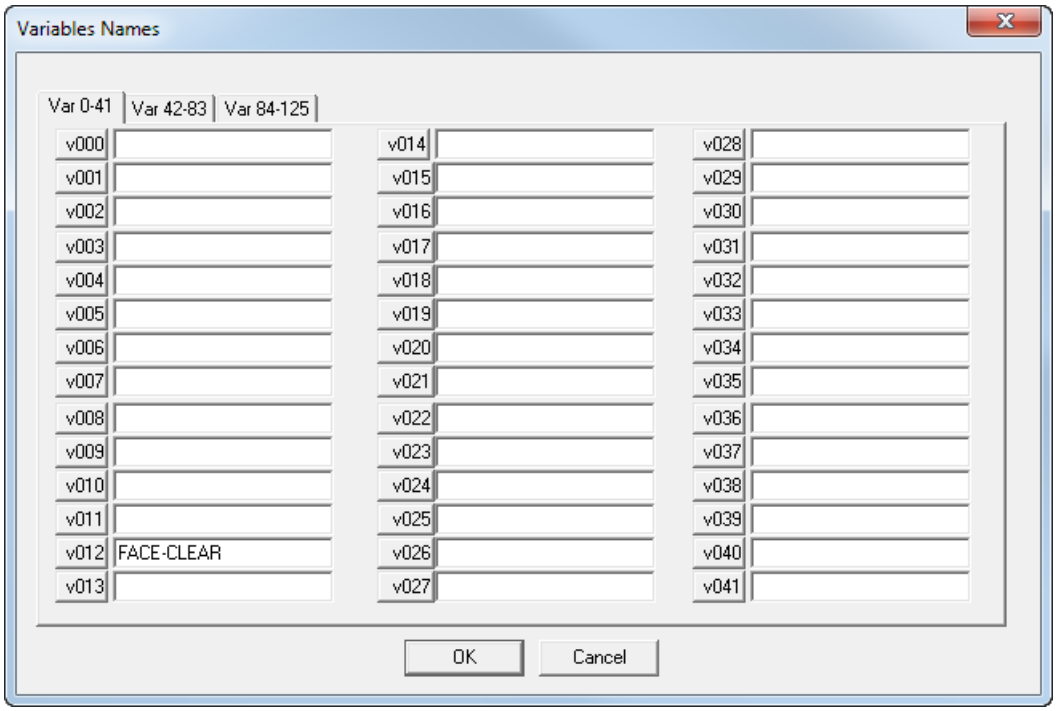
TOOLLIST.TXT is the completed .TXT file.

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 labelled <var-0> through <var-124>. 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 (see page 122).



Assigning values to variables

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

```
~<VARIABLE> VALUE BEING ASSIGNED <EOB>
```

For example, if you assume **v012** in the **Variables Names** dialog (see page 122) is given the name FACE-CLEAR, this variable can be assigned as follows:

```
<IF><metric-post><THEN>
  ~<v012-FACE-CLEAR>2.<eob>
<ENDIF>
<IFNOT><metric-post><THEN>
```

```
  ~<v012-FACE-CLEAR>0.05<eob>
<ENDIF>
```

Inserting variables

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

Using the <include> reserved word

Use the `<include>` reserved word to prompt the PartMaker user to enter additional information during postprocessing. PartMaker includes the information that the user enters in the NC Program file.

See Example – Using the <include> reserved word (see page 137) for more information.

If you want to use the information that the PartMaker user enters within the NC Program file (for example, to output different text strings or to perform calculations), use the `<prompt>` reserved word (see page 144).

Example – Using the <include> reserved word

The following example describes how to add an `<include>` statement to the DEMO.PST file in your PartMaker/Mill Postlib folder. The DEMO.PST file is a generic Fanuc postprocessor.



If you are using PartMaker Turn only, you can copy the formats below into any of the Post files provided with your PartMaker software.

1. Start ConfigPost and select **File > Open** to display the **Open** dialog.
2. Double-click DEMO.PST in the Postlib folder top open it.
3. Select **Configure Mill > Process Header Formats**.

4 Enter the following Program Start format to the Post file:

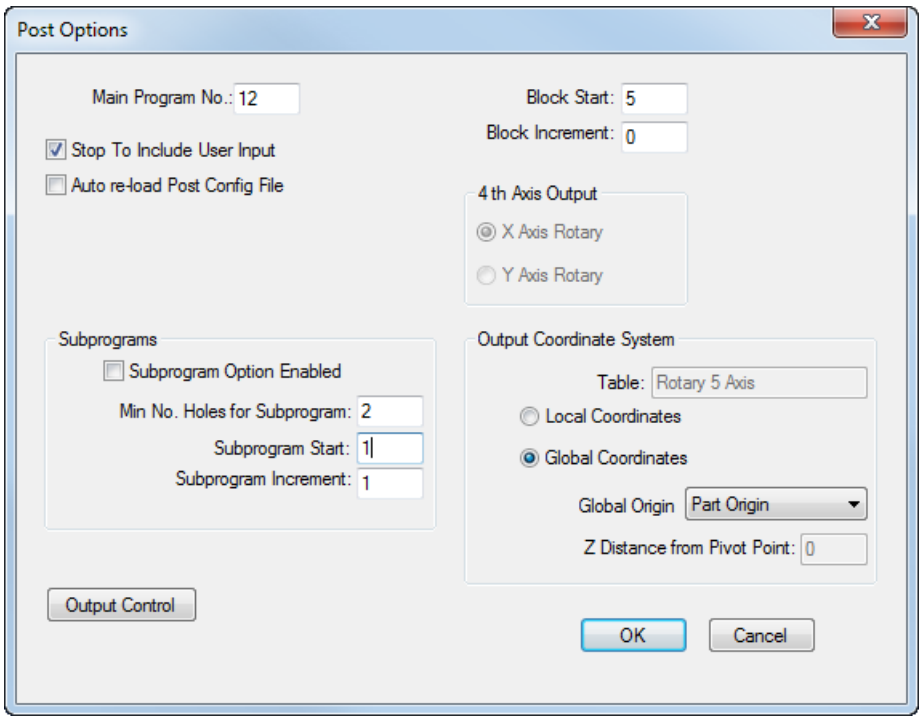
```
%<eob>
:<prog-num1><eob>
(This job was programmed by {Please enter your name:
<include> Programmer})<eob>
(TOTAL MACHINING TIME: <time-total>)<eob>
(MACHINING <face-name> FACE)<eob>
(MACHINING PROCESS <process-id>)<eob>
{N<seq> }G80 G40 G17<eob>
{N<seq> }G30 G91 Z0<eob>
{N<seq> }T<tool-num> M6<eob>
(<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 Save and close the Post file.

6 In PartMaker, with existing .job, .tdb, and .cdb files open, select **Job Optimizer > Post Config File=** to display the **Open Post Configuration File** dialog.

7 Select and open the **Post** file you saved in Step 5.

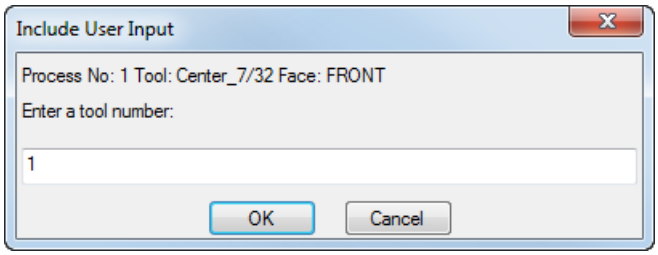
8 Select **Job Optimizer > Post Options** to display the **Post Options** dialog.



9 Select the **Stop to Include User Input** option, then click **OK**.

10 Select **Job Optimizer > Generate NC Program** to generate an NC Program using the postprocessor.

As PartMaker performs the postprocessing it displays the **Include User Input** dialog:



11 Enter your name in the field to replace the default name (Programmer), then click **OK**.

PartMaker includes the name you enter in this dialog in the NC Program file to replace `<include> Programmer`.

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>
```



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, ASIN, ACOS, ATAN**.

Examples of usage:

```
~<var-3>[SIN(30.0)]<eob>
~<var-4>[SQRT((<x-coord>)*(<x-coord>)+(<y-coord>)*(<y-coord>))]<eob>
```



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



Upper case letters only must be used in function names: SIN, COS, TAN, SQRT, ASIN, ACOS, ATAN.



Decimal point must be used with SIN, COS, TAN, ASIN, ACOS, ATAN.



Make sure to encompass each reserved word in parenthesis.

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 Procedures

Procedures in ConfigPost allow the user to simplify a post processor by making it modular.

Using Procedures enables you to decrease the size of a post configuration file by allowing you 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 in the following topics:

Assigning Names to Procedures (see page 141)

Example - Defining Procedures (see page 142)

Calling Procedures (see page 143)

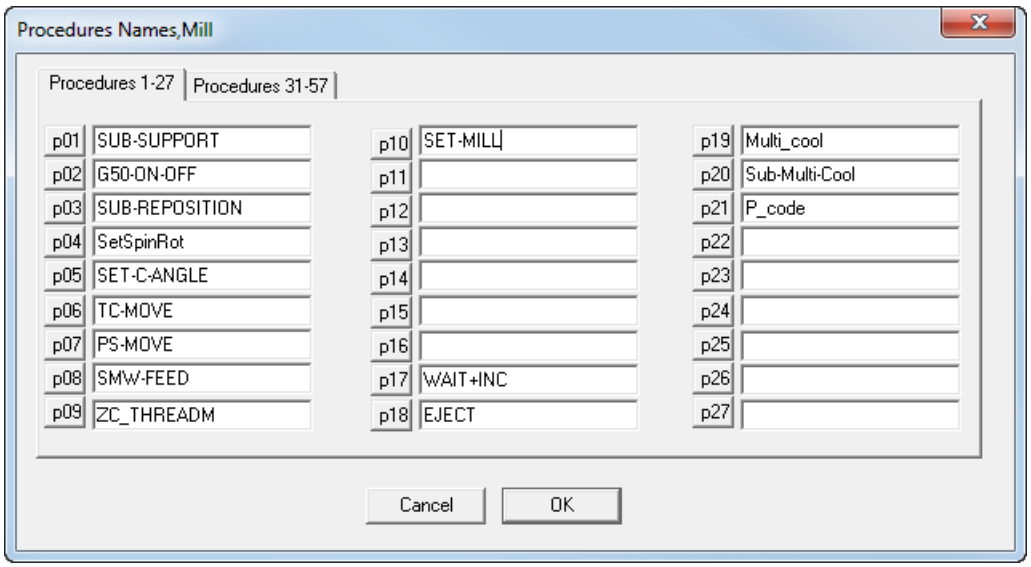
You can follow these same steps to implement new procedures or modify existing procedures post processors for your machines.

Assigning Names to Procedures

Use the **Procedures Names** dialog to assign meaningful names to procedures. To display this dialog, select **Settings > Procedure Names** from the menu bar. Just as with Variables, it is vital to assign procedures meaningful names to make your Post file more readable.

1. To assign a name to a procedure, select **Settings > Procedure Names**.

2 Select either Milling Procedures (see page 116), Turning Procedures (see page 116) or EDM Procedures (see page 117). The dialog for your selected option is displayed. For example:



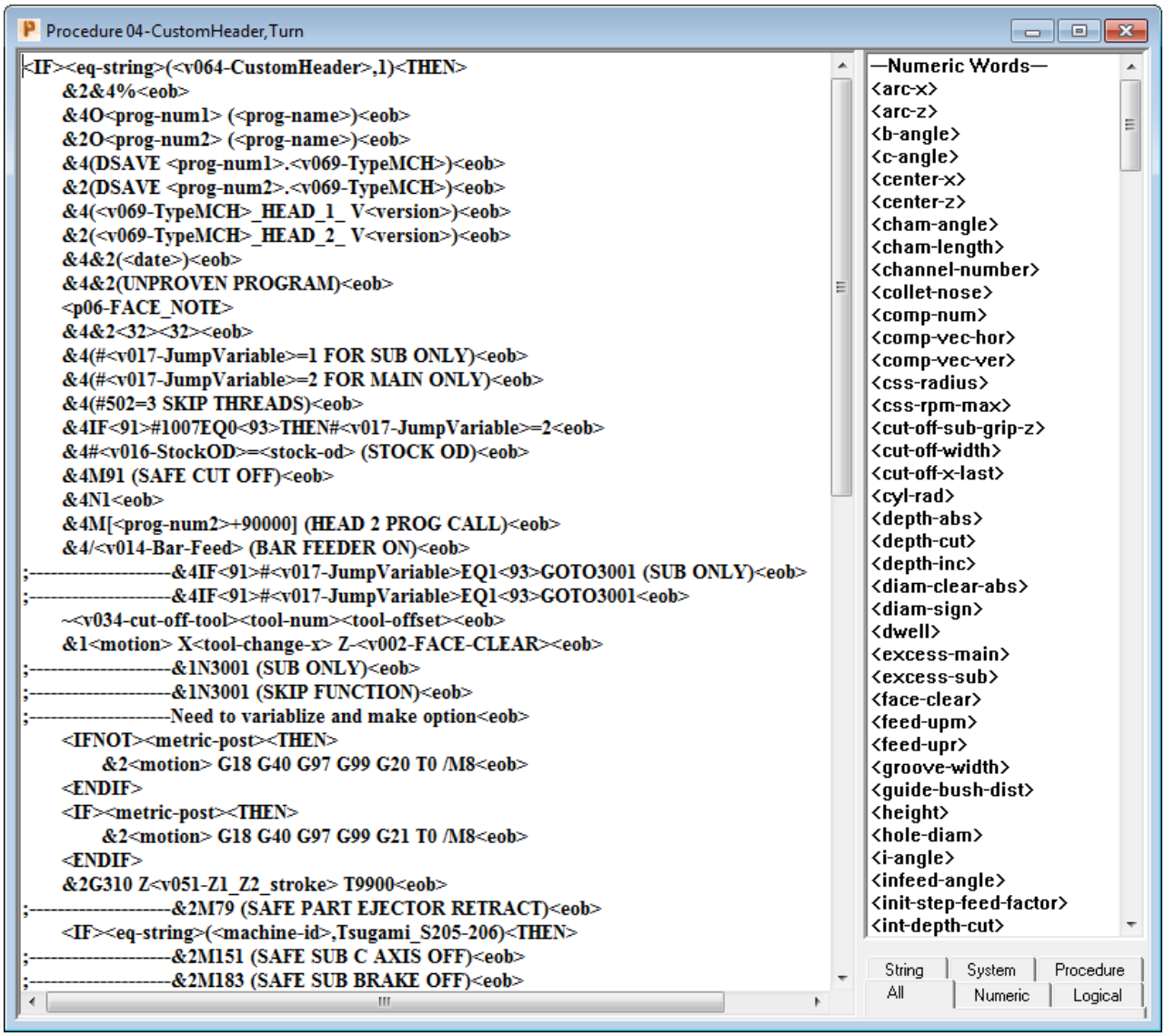
Important for Turn-Mill and SwissCAM:

*When creating a procedure to be used in both Milling and Turning processes, you must define 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.*

Example - Defining Procedures

1 Select **Configure Turn > Procedure Formats** from the menu bar.

2 Select the procedure that is to be defined. The format for that specific procedure will appear in a separate window.



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>
```

Using the <prompt> reserved word

Use the `<prompt>` reserved word to prompt the PartMaker user to enter additional information during postprocessing. You can use the information entered in several ways. For example:

Example 1 – Using <prompt> to output text strings (see page 144)

Example 2 – Using <prompt> to perform calculations (see page 145)

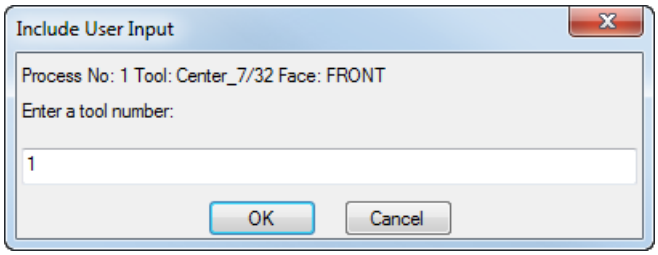
If you want the information entered by the PartMaker user to be included in the NC Program file exactly as the user entered it, use the `<include>` reserved word (see page 137).

Example 1 - Using <prompt> to output text strings

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:



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`.

*PartMaker prompts the user to enter additional information during postprocessing only if the **Stop to Include User Input** option is selected in the **Post Options** dialog.*

Example 2 - Using <prompt> to perform calculations

This example is a construction used to configure a bar puller. Notice how the value given to <prompt> (see page 144) 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>
```



*PartMaker prompts the user to enter additional information during postprocessing only if the **Stop to Include User Input** option is selected on the **Post Options** dialog.*

Using User Data - User Conditions and User Parameters

User Data *(see page 120)* enables you to add your own logical reserved words (called **User Conditions**) and string reserved words (called **User Parameters**) to a Post Configuration file.



Some of the Post Configuration files provided with your PartMaker installation may already include User Data.

When the Post Configuration file is loaded in PartMaker, users can use the **User Data** dialog to activate User Conditions and assign values to User Parameters.

Using User Data to insert tool stops at the end of a process

This example customizes the generic Fanuc postprocessor (Demo.pst) that is supplied with PartMaker/Mill to enable the postprocessor to insert stop commands into an NC program.



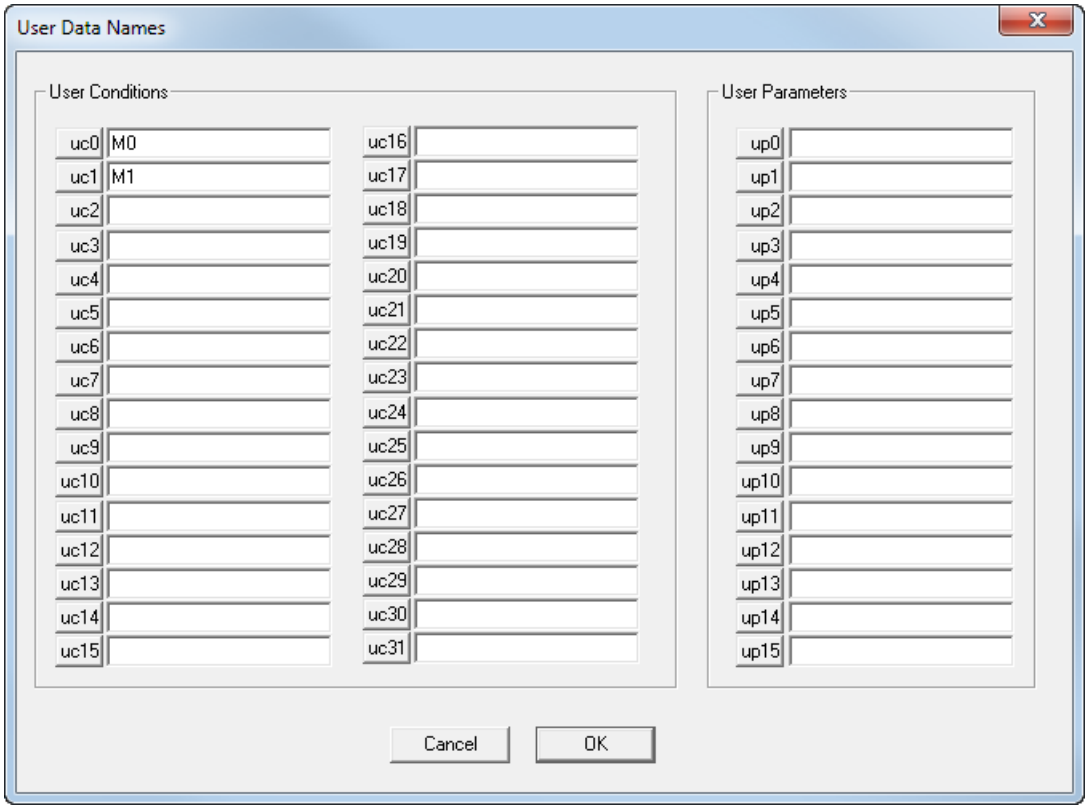
For PartMaker/Turn, follow these steps to modify any Post Configuration file (for example, Fanuc6a.pst) in the Postlib folder.

1. Start ConfigPost and choose **File > Open** to display the **Select File** dialog.
2. Select Demo.pst (or another Post Configuration file) in the Postlib folder.
3. Select **Settings > User Data Names** to display the **User Data Names** dialog.

4 If they do not already exist, enter:

a **M0** in the **UC0** field.

b **M1** in the **UC1** field.



5 Click **OK** to close the dialog.

6 Select **Configure Mill > Process Header/Footer Formats > Process End,** and then enter:

```
Process End Format
;below is an example how to insert  User Conditions
<eob>
<IF><uc0-M0><THEN>
   {N<seq> }M0<eob>
<ENDIF>
<IF><uc1-M1><THEN>
   {N<seq> }M1<eob>
<ENDIF>
```

*If you are using PartMaker/Turn, use the **Configure Turn** menu.*

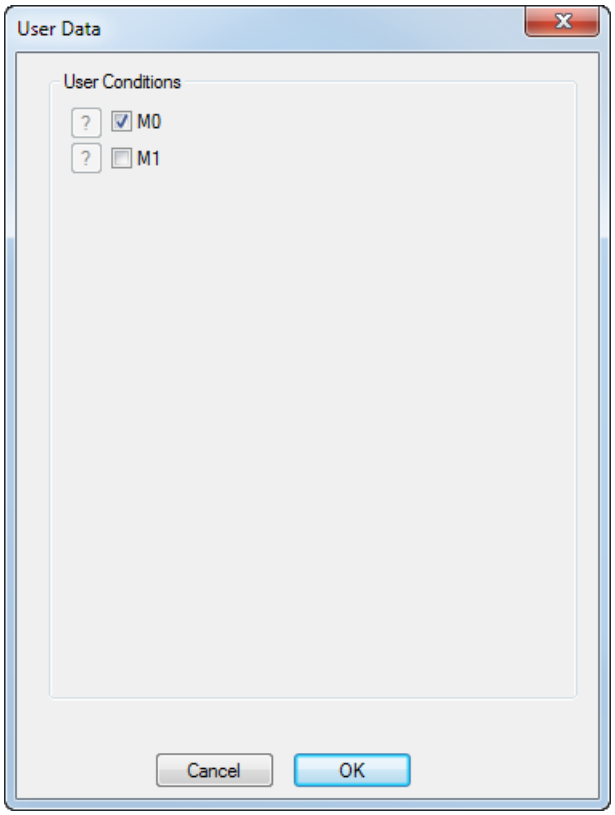
7 In PartMaker, select **Job Optimizer > Post Config File** to display the **Open Post Configuration File** dialog.

8 Select **Demo.pst** (or the post configuration file you have edited) and click **Open** to load this file into PartMaker.

9 In PartMaker's **Process Table**, double-click a process after which you want to place an M1 or M0 command. PartMaker displays the **Process Parameters** dialog.

10 Click the **User Data** button to display the **User Data** dialog.

Select the User Condition that you want to use. The following example activates M0, so the postprocessor outputs an M0 command, or optional stop, during postprocessing.



You must select only one User Condition for this example.

Import / Export WRI Files

A WRI file is a text file produced by ConfigPost application.

A WRI file allows you to see all Post Processor File Formats as text. You can modify a WRI file using an editor such as Notepad or Wordpad. Once modified, you can load the WRI file back into ConfigPost.

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 (see page 42). From the **File** menu choose **Export WRI File > Complete** or **Export WRI File > Non-Empty Formats Only**.

- When **Export WRI File > Complete** (see page 42) is chosen, then ConfigPost exports all Program Formats including empty ones.
- When **Export WRI File > Non-Empty Formats Only** (see page 42) is chosen, then ConfigPost exports only Non-Empty Program Formats.

How to Load WRI File

In order for WRI file to be loaded 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 then they will take effect right after the WRI file is loaded.

Structure of a WRI File

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

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

Example of a Tag:

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

ConfigPost for Turn-Mill and SwissCAM

This section describes how to use ConfigPost for Turn-Mill and SwissCAM.

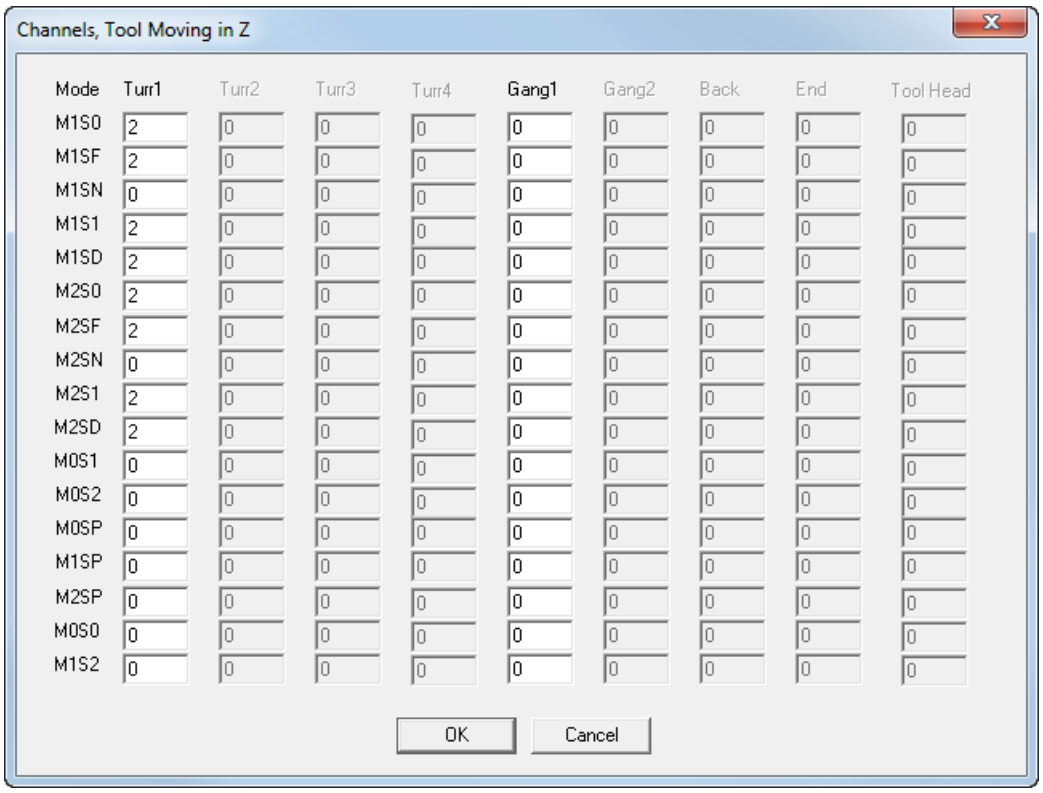
Channels Overview

Multi-channels controls (sometimes called multi-path controls) are commonly used by Turn-Mill Centers and Swiss-Type Lathes. A channel represents either a separate file when a multi-file output is required or a specific partition of a single file when a single output file is required. Star Micronics CNC Swiss Type Lathes with Fanuc Controls are typical examples of machines with two-channel controls that require a two file output. Citizen-M Swiss Type Lathes with Mitsubishi controls are typical examples of machines with three channels controls that require a single file with three separate partitions called $1, $2, $3.

To generate correct programs for multi-channel controls, PartMaker has to know the rules about where to place the code for each individual tool motion. These rules vary from machine to machine and depend on the following information:

- Tool Post
- Synchronization Mode
- Spindle Selection (Main spindle or Sub-spindle)
- Tool Status (either stationary in Z or movable in Z).

PartMaker uses the Channel Number entered in the **Channels, Tool Moving in Z** and **Channels, Tool Stationary in Z** dialogs to assign `<channel-number>` and `<next-on-channel>` reserved words.



Error checking in PartMaker

Post files contain the following error-checking features when generating NC code in PartMaker:

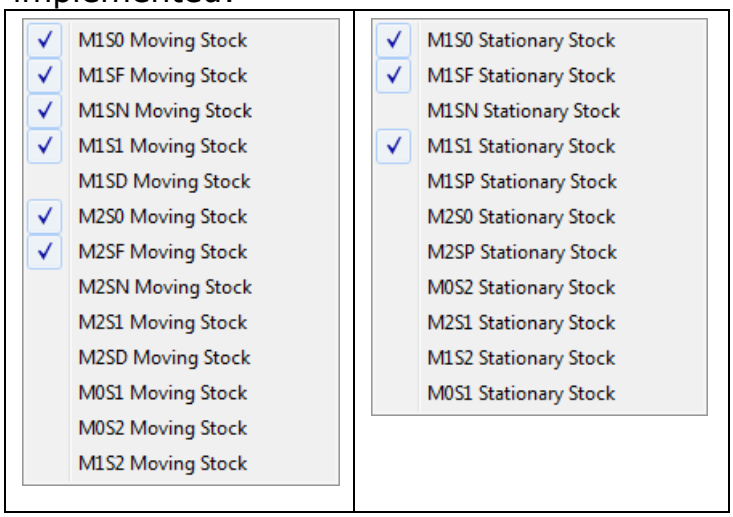
- Error checking for modes.
- Error checking for implemented faces.
- Error checking for tool posts.

Error checking for modes

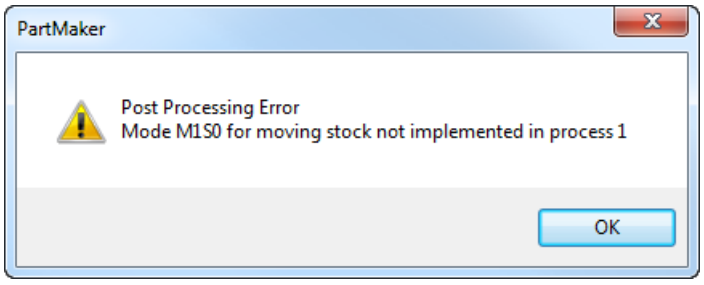
To view a list of the modes for moving stock that have been implemented for the Post file, select **Configure Mill > Modes > Moving Stock** or **Configure Turn > Modes > Moving Stock**.

To view a list of the modes for stationary stock that have been implemented for the Post file, select **Configure Mill > Modes > Stationary Stock** or **Configure Turn > Modes > Stationary Stock**.

A check mark is displayed next to the modes that have been implemented:



When generating an NC program, PartMaker checks that each mode used in PartMaker Process Table has been implemented in the Post File. If a mode used in PartMaker's Process Table has not been implemented, PartMaker displays a message during post processing. For example:



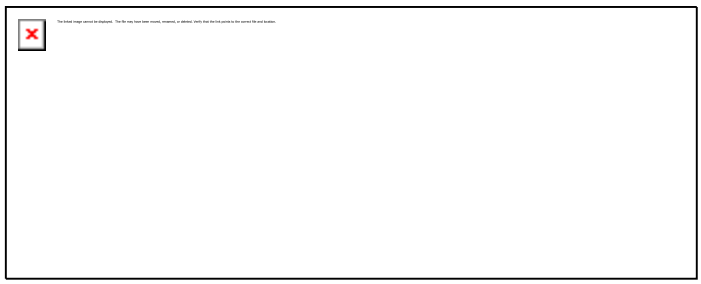
Error checking for Faces

To view a list of the Faces that have been implemented for the Post file, select **Settings > Faces Info**.

A check mark is displayed next to the Faces that have been implemented:



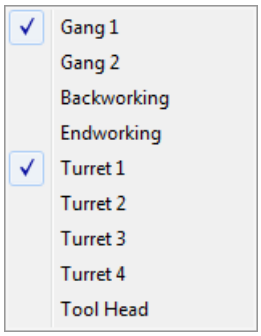
When generating an NC program, PartMaker checks that each Face used in PartMaker Process Table has been implemented in the Post File. If a Face used in PartMaker's Process Table has not been implemented, PartMaker displays a message during post processing. For example:



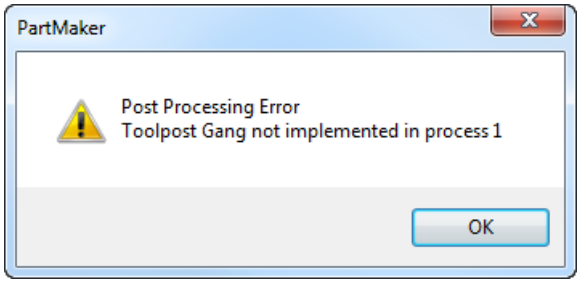
Error checking for tool posts

To view a list of the tool posts that have been implemented for the Post file, select **Settings > Tools Info**.

A check mark is displayed next to the tool posts that have been implemented:



When generating an NC program, PartMaker checks that each tool post used in the Process Table has been implemented in the Post file. If a tool post is used in a Process Table, but has not been implemented in the Post file PartMaker displays a message during post processing. For example:



Using ConfigPost for Turn-Mill and 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 that 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

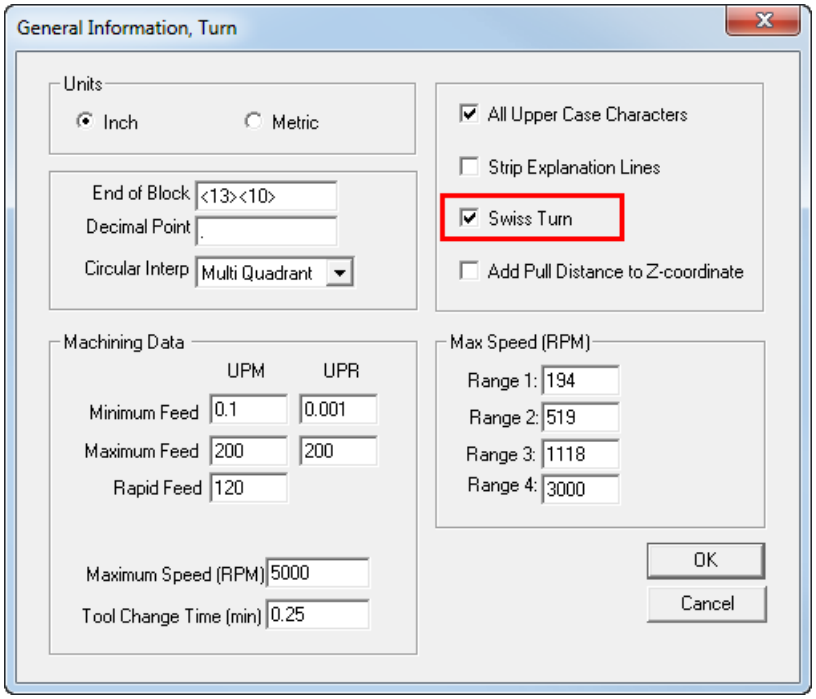
The following illustrations show how PartMaker/Turn-Mill handles each process in a Process Table involving turning as well as milling and drilling operations.

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.



Differentiating Between Turning and Milling Processes

PartMaker can 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 several different types of Milling process output, depending on the type of "live tooling" operation being performed. ConfigPost can 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 can 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.

See also:

Rules for Program Start, Tool Change, and Process Start With Z-Oriented Tools (Mill) (see page 39)

Rules for Program Start, Tool Change, and Process Start with X-Oriented Tools (Mill) (see page 39)

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 you define in a Post file (see page 15).

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)
```



*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.*



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

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) comprises 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** is loaded automatically and is used to generate output for the sub-spindle when sub- spindle operations are being programmed.



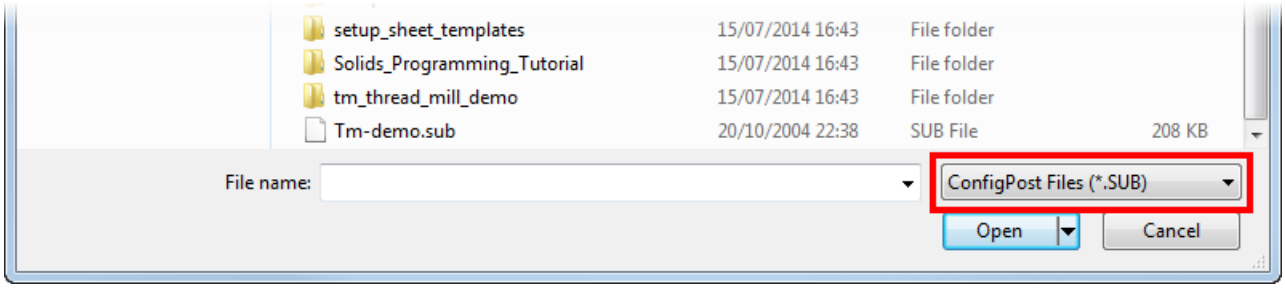
*You specify sub-spindle processes in PartMaker by selecting the **Sub-Spindle** option in the **Setup** dialog for a given **Face Window**.*

You can use ConfigPost for Turn-Mill to configure milling and turning process output for both .pst and .sub file outputs. Configuring post processors for files with a .pst extension is performed in the standard manner.

To open a post configuration file with a .sub extension for customization:

1 Start ConfigPost for Turn-Mill and select **File > Open** to display the **Open** dialog.

2 In the drop-down list beside the **File Name** field, select **ConfigPost Files(*.SUB).**



3 To load the post configuration for sub-spindle, select the **Tm-demo.sub** file and then click **Open**.

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>

*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** (for example, **M1S0**, **M1S1**, **M2S0**) 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>
```

Configuring a post processor to support imported PowerMill toolpaths

This topic summarizes how to modify a post processor to support imported PowerMill toolpaths into PartMaker/SwissCAM or PartMaker/Turn-Mill.

In addition to modifying the formats described in this topic, you must also ensure that the **Imported PowerMill Toolpaths** option is set appropriately on the **General Information, Mill** dialog (see page 98).

PartMaker can post process five different toolpath types imported from PowerMill into a Face window with a *Mill 5-Axis Plane* machining function. You can usually reuse existing formatting because of its similarity to an existing PartMaker Face window:

PartMaker name for toolpath type	Output for <5-axis-path-type>	Similarity to formatting used in PartMaker Face window
5-axis Simultaneous	5_AXIS_SIMUL	The post processing is identical to that of a 5-axis simultaneous toolpath created in PartMaker.
4-axis Simultaneous	4_AXIS_SIMUL	The post processing will closely resemble the configuration already in place for **Mill Diameter, Polar**.
3 Plus 2 Axis	3_PLUS_2_AXIS	The post processing will closely resemble the configuration already in place for **Mill 5-Axis Plane.**
3 Axis XYZ	3_AXIS_XYZ	The post processing will closely resemble the configuration already in place for **Mill XY Plane**.
3 Axis XCZ	3_AXIS_XCZ	The post processing will closely resemble the configuration already in place for **Mill End Polar**.

Typically, you will need to modify the following formats to add support for imported PowerMill toolpaths.

- Linear Move

A standard Linear Move Format:

```
<IF><face-5-axis><THEN>
  IF><eq-string>(<5-axis-path-type>,5_AXIS_SIMUL)<THEN>
    &1{<motion> }{X<x-coord> }{Y<y-coord> }{Z<z-coord>
    }{B<b-angle> }{C<c-angle> }{F<feed-upm>}<eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,4_AXIS_SIMUL)<THEN>
    &1{<motion> }{X<x-coord> }{Y<y-coord> }{Z<z-coord>
    }{C<c-angle>}{F<feed-upm>}<eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_PLUS_2_AXIS)<THEN>
    &1{<comp-status> }{<motion> }{X<x-coord> }{Y<y-coord>
    }{Z<z-coord>}{F<feed-upm>}<eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_AXIS_XYZ)<THEN>
    &1{<motion> }{X<x-coord> }{Y<y-coord> }{Z<z-coord>
    }{F<feed-upm>}<eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_AXIS_XCZ)<THEN>
    &1{motion> }{X<x-coord> }{C<y-coord> }{Z<z-coord>
    }{F<feed-upm>}<eob>
  <ENDIF>
<ENDIF>
```

- Horizontal Rapid Move

A standard Horizontal Rapid Move Format:

```
<IF><face-5-axis><THEN>
  <IF><eq-string>(<5-axis-path-type>,5_AXIS_SIMUL)<THEN>
    &1{<motion> }{X<x-coord> }{Y<y-coord> }{Z<z-coord>
    }{B<b-angle> }{C<c-angle> }<eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,4_AXIS_SIMUL)<THEN>
    &1{<motion> }{X<x-coord> }{Y<y-coord> }{Z<z-coord>
    }{C<c-angle>}<eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_PLUS_2_AXIS)<THEN>
    &1{<motion> }{X<x-coord> }{Y<y-coord> }{Z<z-coord>
    }<eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_AXIS_XYZ)<THEN>
    &1{<motion> }{X<x-coord> }{Y<y-coord> }{Z<z-coord>
    }<eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_AXIS_XCZ)<THEN>
    &1{motion> }{X<x-coord> }{C<y-coord> }{Z<z-coord>
    }<eob>
  <ENDIF>
```

```
<ENDIF>
```

- Vertical Rapid Move

A standard Vertical Rapid Move Format:

```
<IF><face-5-axis><THEN>
  <IF><eq-string>(<5-axis-path-type>,5_AXIS_SIMUL)<THEN>
;-------Not required<eod>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,4_AXIS_SIMUL)<THEN>
;-------Not required<eod>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_PLUS_2_AXIS)<THEN>
    &1{<motion> }{X<x-coord> }{Y<y-coord> }{Z<z-coord>
    }<eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_AXIS_XYZ)<THEN>
;-------Not required<eod>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_AXIS_XCZ)<THEN>
;-------Not required<eod>
  <ENDIF>
<ENDIF>
```



PowerMill and 5-axis Simultaneous toolpaths do not use the Vertical Rapid Move format.

- Circular Move

A standard Circular Move Format:

```
<IF><face-5-axis><THEN>
  <IF><eq-string>(<5-axis-path-type>,5_AXIS_SIMUL)<THEN>
;-------Not required<eod>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,4_AXIS_SIMUL)<THEN>
;-------Not required<eod>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_PLUS_2_AXIS)<THEN>
    &1{<motion> }{X<x-coord> }{Y<y-coord> }R<radius><eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_AXIS_XYZ)<THEN>
    &1{<motion> }{X<x-coord> }{Y<y-coord> }R<radius><eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_AXIS_XCZ)<THEN>
    &1{motion> }{X<x-coord> }{C<y-coord> }R<radius><eob>
  <ENDIF>
<ENDIF>
```

- Helical Move

A standard Helical Move Format:

```
<IF><face-5-axis><THEN>
  <IF><eq-string>(<5-axis-path-type>,5_AXIS_SIMUL)<THEN>
;-------Not required<eod>
```

```
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,4_AXIS_SIMUL)<THEN>
;-------Not required<eod>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_PLUS_2_AXIS)<THEN>
    &1{<motion> }{X<x-coord> }{Y<y-coord> }{Z<z-coord>
    }R<radius><eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_AXIS_XYZ)<THEN>
;-------Not required<eod>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_AXIS_XCZ)<THEN>
;-------Not required<eod>
  <ENDIF>
<ENDIF>
```

- Tool Change and Process Start

For Tool Change and Process Start:

```
<IF><face-5-axis><THEN>
  <v07-TURRET>(5 AXIS FACE <5-axis-path-type>)<eob>
  <IF><eq-string>(<5-axis-path-type>,5_AXIS_SIMUL)<THEN>
    &1G0 B<b-angle><eob>
    &1 G43.4 D<tool-offset> (TCP ON)<eob>
    &1<motion> X<x-coord> Y<y-coord> Z<z-coord> C<c-
    angle><eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,4_AXIS_SIMUL)<THEN>
    &1B<b-angle><eob>
    &1<motion> X<x-coord> Y<y-coord> Z<z-coord> c<c-angle
    ><eob>
  ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_PLUS_2_AXIS)<THEN>
    &1G68.1 X<local-origin-x> Y<local-origin-y> Z<local-
    origin-z> I0 J1 K0 R[(<b-angle>)*(-1.0)] <eob>
    &1motion> C<c-angle><eob>
    &1X<x-coord> Y<y-coord> Z<z-coord><eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_AXIS_XYZ)<THEN>
    &1G98 G17 <motion> Z<z-coord> Y<y-coord> C<c-
    angle><eob>
    &1X<x-coord><eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_AXIS_XCZ)<THEN>
    &1G98 G0 X[<stock-od>+<tool-diam>] Y0<eob>
    &1Z<z-coord><eob>
    &1C0<eob>
    &1G12.1 (POLAR ON)<eob>
    &1<motion> X<x-coord> C<y-coord> F<feed-upm><eob>
  <ENDIF>
<ENDIF>
```

- Process End

For Process End:

```
<IF><face-5-axis><THEN>
  v07-TURRET>(5 AXIS FACE <5-axis-path-type>)<eob>
  <IF><eq-string>(<5-axis-path-type>,5_AXIS_SIMUL)<THEN>
    &1G0 G49(Cancel RTCP)<eob>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,4_AXIS_SIMUL)<THEN>
;----------No Cancelation is Typically Required<eod>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_PLUS_2_AXIS)<THEN>
    &1G69 (Cancel 5-axis Plane)<eod>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_AXIS_XYZ)<THEN>
;----------No Cancelation is Typically Required<eod>
  <ENDIF>
  <IF><eq-string>(<5-axis-path-type>,3_AXIS_XCZ)<THEN>
    &1G13.1 (POLAR CANCEL)<eod>
  <ENDIF>
<ENDIF>
```

PartMaker Post Configuration files

PartMaker/Mill, PartMaker/Turn and PartMaker/EDM installations include a **\postlib** folder beneath your PartMaker User Files folder, which contains sample Post Configuration files (or Post files) for many popular CNC machines.



The location of your PartMaker User Files folder is specified during the installation process. The default location is ***C:\PartMaker****.*

PartMaker uses a Post file as a template to convert the Process Table into a part program file according to CNC machine guidelines. This process is known as postprocessing.

You can customize these files, or create additional Post files, using the ConfigPost application provided with PartMaker.

Post Library - Milling, Standard Table

A-B_8200.PST	Sample Config file for **Allen Bradley 8200** control. ▪ Cutter diameter compensation is supported. ▪ Multiple fixture numbers H1,H2... are supported. ▪ Milling Subprograms are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms can be enabled in ConfigPost's **General Info** dialog. ▪ Enter the fixture number in the Work Offset field in PartMaker's **Faces** dialog.
ACRA_850.PST	Sample Config file for a **Vickers Acramatic 850** control as found on the Arrow mill (and others). ▪ Arc centers are absolute. ▪ Multiple fixture offsets are supported. ▪ Cutter diameter compensation is supported.
ACRA_1000.PST	Sample Config file for **Acramatic controls 10H, 10HC-1000/1500/2500** ("Big Blue"). ▪ Single quadrant circular interpolation. ▪ Vector type (P,Q) cutter diameter compensation is supported.
ACRA_2100.PST	Sample Config file for **Vickers Acramatic 2100** control. ▪ Arc centers are absolute. ▪ Multiple fixture offsets are supported. ▪ Cutter diameter compensation is supported.
ANILAM.PST	Sample generic Config file for **Anilam** control on a CNC Mill.
ANILAM_1400.PST	Sample Config file for **Anilam 1400** control on a mill with an automatic tool changer. ▪ Arc centers are always absolute. ▪ Multiple work offsets are supported. ▪ Uses M6 for tool change command.
ANILAM_1400_NO_ATC.PST	Sample Config file for **Anilam 1400** control on a mill with a manual tool changer. ▪ Arc centers are always incremental. ▪ Multiple work offsets are supported. ▪ Uses M6 for tool change command.
BOSS_4A.PST	Sample Config file for **Bridgeport BOSS4,5** controls. ▪ Uses single quadrant circular interpolation with

	unsigned I,J values. ▪ Cutter diameter compensation is not supported.
BOSS_6.PST	Sample Config file for **Bridgeport BOSS5,6,7** controls. ▪ Cutter diameter compensation is supported. ▪ 16 Milling Subprograms are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms can be enabled in ConfigPost's **General Info** dialog.
BOSS_10.PST	Sample Config file for **Bridgeport BOSS8,9,10** and new **BOSS10** compatible controls. ▪ Cutter diameter compensation is supported. ▪ 99 Milling Subprograms are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms can be enabled in ConfigPost's **General Info** dialog. ▪ M3 / M4 and M5 are not output for spindle control. If these codes are required, then use the DX32.PST.
CENTROID.PST	Sample **Centroid** Config file. ▪ Drilling Subprograms are supported. ▪ Milling Subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter diameter compensation is supported.
CONTOUR.PST	Sample **Contour** Config file. ▪ Drilling Subprograms are supported. ▪ Milling Subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter diameter compensation is supported.
CRUSADER-G.PST	Sample Config file for **Anilam Crusader-G** mill. ▪ Cutter diameter compensation is supported.
CRUDSADER-M.PST	Sample Config file for **Anilam Crusader-M** mill. ▪ Cutter diameter compensation is supported.
DEMO.PST	Functionally the same as **Fanuc.pst**. Additional comments included to further illustrate how ConfigPost can be used to improve NC program format.
DX32.PST	Sample Config file for **Bridgeport DX32** and new **BOSS10** compatible controls that require **M3 / M4** and **M5** to activate the spindle.

	▪ Cutter diameter compensation is supported. ▪ 99 Milling Subprograms are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms can be enabled in ConfigPost's **General Info** dialog.
DYNACONV.PST	Sample **Dynapath 10/20/30** Config file in CONVERSATIONAL mode. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms are not supported. ▪ Multiple Work Offsets are supported. ▪ Cutter diameter compensation is supported. ▪ Each line must begin with an event (1) and end with a $.
DYNAPATH_CONV.PST	Sample **Dynapath 20/30** Config file for CONVERSATIONAL (G-code) programming with 2-digit tool numbers: ▪ Cutter diameter compensation is supported. ▪ Multiple Work Offsets are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms are not supported.
DYNAPATH_EIA_2.PST	Sample **Dynapath 20/30** Config file for **EIA (G-code)** programming with 2 digit tool numbers. ▪ Cutter diameter compensation is supported. ▪ Multiple Work Offsets are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms are not supported.
DYNAPATH_EIA_4.PST	Sample **Dynapath 20/30** Config file for **EIA (G-code)** programming with 4 digit tool numbers. ▪ Cutter diameter compensation is supported. ▪ Multiple Work Offsets are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms are not supported.
EZPATH.PST	Sample **EZ Path** Config file. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms are not supported. ▪ Multiple Work Offsets are supported. ▪ Cutter diameter compensation is supported.
FADAL.PST	Sample Config file for **FADAL** controls. ▪ Cutter diameter compensation is supported.

	▪ Multiple work offset coordinate systems are supported.
FADAL_ROTARY_A.PST	Sample Config file for **FADAL** controls with programmable A rotary axis. ▪ Cutter diameter compensation is supported. ▪ Fixture offsets E1 - E99 supported.
FADAL-E.PST	Sample Config file for **FADAL** controls. ▪ Cutter diameter compensation is supported. ▪ Multiple fixture numbers E1,E2... are supported. ▪ Fanuc style Milling and Drilling Subprograms are supported. ▪ Milling and Drilling Subprograms can be enabled in ConfigPost's **General Info** dialog. ▪ You should enter the fixture number in PartMaker **Faces** dialog in the **Work Offset** field.
FADAL_400.PST	Sample Config file for **FADAL** controls. ▪ Cutter diameter compensation is supported. ▪ Fanuc style Milling and Drilling Subprograms are supported. ▪ Milling and Drilling Subprograms can be enabled in ConfigPost's **General Info** dialog. ▪ You should enter the fixture number in PartMaker **Faces** dialog in the **Work Offset** field.
FAGOR.PST	Sample Config file for **Fagor 80XX** series control. ▪ Multiple Work Offsets are supported. ▪ Drilling subprograms are supported. ▪ Milling subprograms are supported. ▪ Cutter diameter compensation is supported. ▪ Tool changes are assumed to be with an automatic changer, using the **Taa.bb** format, where **aa** is the tool number and **bb** is the offset number.
FAGOR_NO_ATC.PST	Sample Config file for **Fagor 80XX** series control. ▪ Multiple Work Offsets are supported. ▪ Drilling subprograms are supported. ▪ Milling subprograms are supported. ▪ Cutter diameter compensation is supported. ▪ Tool changes are assumed to be with an manual changer, using the **T.aa** format where **aa** is the tool offset to be applied.

FANUC_6.PST	Sample Config file for **Fanuc 6** type controls. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Milling and Drilling Subprograms are supported. ▪ Milling and Drilling Subprograms can be enabled in ConfigPost's **General Info** dialog.
FANUC_10.PST	Sample Config file for **Fanuc 10** type controls. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Milling and Drilling Subprograms are supported. ▪ Milling and Drilling Subprograms can be enabled in ConfigPost's **General Info** dialog.
FANUC_11.PST	Sample Config file for **Fanuc 11** type controls. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Milling and Drilling Subprograms are supported. ▪ Milling and Drilling Subprograms can be enabled in ConfigPost's **General Info** dialog.
FANUC_15.PST	Sample Config file for **Fanuc 15** type controls. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Milling and Drilling Subprograms are supported. ▪ Milling and Drilling Subprograms can be enabled in ConfigPost's **General Info** dialog.
FANUC_WITHOUT_M6.PST	Sample Config file for **Fanuc** type controls without M6. Tool changes are executed solely on a T call. Typically used with machines that have no Automatic Tool Changer. ▪ Cutter diameter compensation is supported. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Milling and Drilling Subprograms are supported. ▪ Milling and Drilling Subprograms can be enabled in ConfigPost's **General Info** dialog.
FANUC_ROTARY_A.PST	Sample Config file for **Fanuc** control with rotary A axis.

	▪ Drilling subprograms are supported. ▪ Milling subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter Diameter compensation is supported.
FANUC_O.PST	Sample Config file for the **Fanuc O** series control. ▪ Drilling subprograms are supported. ▪ Milling subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter Diameter compensation is supported.
FANUC_16I.PST	Sample Config file for the **Fanuc 16 / 16i** series control. ▪ Drilling subprograms are supported. ▪ Milling subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter Diameter compensation is supported.
FANUC_18I.PST	Sample Config file for the **Fanuc 18 / 18i** series control. ▪ Drilling subprograms are supported. ▪ Milling subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter Diameter compensation is supported.
FANUC_160.PST	Sample Config file for the **Fanuc 160** series control. ▪ Drilling subprograms are supported. ▪ Milling subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter Diameter compensation is supported.
FANUC_180.PST	Sample Config file for the **Fanuc 180** series control. ▪ Drilling subprograms are supported. ▪ Milling subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter Diameter compensation is supported.
FANUC_20i.PST	Sample Config file for the **Fanuc 20 / 20i** series control. ▪ Drilling subprograms are supported. ▪ Milling subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter Diameter compensation is supported.
FANUC_21i.PST	Sample Config file for the **Fanuc 21 / 21i** series

	control. ▪ Drilling subprograms are supported. ▪ Milling subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter Diameter compensation is supported.
FANUC_30i.PST	Sample Config file for the **Fanuc 30 / 30i** series control. ▪ Drilling subprograms are supported. ▪ Milling subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter Diameter compensation is supported.
FANUC_31i.PST	Sample Config file for the **Fanuc 31 / 31i** series control. ▪ Drilling subprograms are supported. ▪ Milling subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter Diameter compensation is supported.
FANUC_32i.PST	Sample Config file for the **Fanuc 32 / 32i** series control. ▪ Drilling subprograms are supported. ▪ Milling subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter Diameter compensation is supported.
GE1050_IJ.PST	Sample Config file for **GE 1050** controls. ▪ Uses unsigned I,J values for circular interpolation.
G&L_NUM_800.PST	Sample Config file for **Giddings and Lewis Numeric 800**.
G&L_NUM_800M.PST	Sample **Giddings and Lewis** Config file. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms are not supported. ▪ Multiple Work Offsets are not supported. ▪ Cutter diameter compensation is supported.
HAAS_VMC.PST	Sample Config file for **HAAS** controls as implemented on VMCs. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Multiple work offsets (G54, G55,..) are supported. ▪ Fanuc style Milling and Drilling Subprograms are

	supported. ▪ Milling and Drilling Subprograms can be enabled in ConfigPost's **General Info** dialog.
HH_CONV.PST	Sample Config file for **Heidenhain** conversational control. ▪ Cutter diameter compensation is supported. ▪ Milling Subprograms are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms can be enabled in ConfigPost's **General Info** dialog.
HITACHI_SEIKI.PST	Sample **Hitachi Seiki** Config file. ▪ Drilling Subprograms are supported. ▪ Milling Subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter diameter compensation is supported.
HURCO_ULTIMAX.PST	Sample **Hurco Ultimax II** or **Ultimax III** Config file. ▪ Cutter Diameter compensation is supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms are not supported.
HARDINGE_VMC_FANUC.PST	Sample Config File for **Hardinge VMC** with **Fanuc** control.
HARDINGE_VMC_SIEMENS.PST	Sample Config File for **Hardinge VMC** with **Siemens** control.
LEBLOND.PST	Sample Config File for **Leblond Makino**.
MAZAK.PST	Sample Config file for **MAZAK** controls. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Tools preloading is not supported. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Milling Subprograms are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms can be enabled in ConfigPost's **General Info** dialog.
MAZAK_FUSION.PST	Sample Config file for **MAZAK** FUSION controls. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Tools preloading is supported. ▪ Multiple work offsets (G54,G55,..) are supported.

	▪ Milling Subprograms are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms can be enabled in ConfigPost's **General Info** dialog.
MAZAK_MATRIX.PST	Sample Config file for **MAZAK** MATRIX controls. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Tools preloading is supported. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Milling Subprograms are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms can be enabled in ConfigPost's **General Info** dialog.
MELDAS.PST	Sample Config file for **Mitsubishi MELDAS** control. ▪ Cutter compensation is supported. ▪ Multiple work offsets are supported. ▪ Tools are changed with a M6 command. ▪ Drilling Subprograms are supported. ▪ Milling Subprograms are supported.
MELDAS_NO_ATC.PST	Sample Config file for **Mitsubishi MELDAS** control for use with machines without automatic tool changer. Tool changes are assumed to be by hand. ▪ T-Codes are not used. Appropriate tool is determined by the H-code used in the G43 line associated with the tool changing process. ▪ Cutter compensation is supported. ▪ Multiple work offsets are supported ▪ Drilling Subprograms are supported.
MITSUI_SEIKI.PST	Sample **Mitsui Seiki** Config file. ▪ Drilling Subprograms are supported. ▪ Milling Subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter diameter compensation is supported.
MIYANO_VMC.PST	Sample **Miyano VMC** Config file. ▪ Drilling Subprograms are supported. ▪ Milling Subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter diameter compensation is supported.
OKADA.PST	Sample **Okada** Config file.

	▪ Drilling Subprograms are not supported. ▪ Milling Subprograms are not supported. ▪ Multiple Work Offsets are supported. ▪ Cutter diameter compensation is supported.
OKK.PST	Sample Config file for **OKK Machining Centers**.
OSP_5000_VMC.PST	Sample Config file for **OKUMA 5000 series** controls. ▪ Cutter diameter compensation is supported. ▪ Single fixture number G15H1 is supported. ▪ Milling Subprograms are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms can be enabled in ConfigPost's **General Info** dialog.
OSP_7000_VMC.PST	Sample Config file for **OKUMA 7000 series** controls. ▪ Cutter diameter compensation is supported. ▪ Single fixture number G15H1 is supported. ▪ Milling Subprograms are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms can be enabled in ConfigPost's **General Info** dialog.
OSP_5000_HMC.PST	Sample Config file for **OKUMA 5000 series** controls on horizontal machining centers. ▪ Cutter diameter compensation is supported. ▪ Multiple fixture numbers G15 H1,G15 H2... are supported. ▪ Milling Subprograms are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms can be enabled in ConfigPost's **General Info** dialog.
OSP_7000_HMC.PST	Sample Config file for **OKUMA 5000 series** controls on Horizontal Mills ▪ Cutter diameter compensation is supported. ▪ Multiple fixture numbers G15 H1,G15 H2... are supported. ▪ You should enter the fixture number in PartMaker **Faces Dialog** in the **Work Offset** field. ▪ Milling Subprograms are supported. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms can be enabled in ConfigPost's **General Info** dialog.

PROT_MX3.PST	Sample Config file for **ProtoTrack-MX3** controls. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Fully Fanuc compatible.
PROT_PLUS.PST	Sample Config file for **ProtoTrack-PLUS** controls. ▪ Cutter diameter compensation is supported. ▪ Not fully Fanuc compatible because it uses ";" at the end of each line. ▪ Milling Subprograms are supported.
QUICK.PST	Sample Config file for **QUICKMILL (Canadian)**. ▪ Cutter diameter compensation is supported. ▪ Multiple work offsets are supported. ▪ Drilling Subprograms are supported. ▪ Milling Subprograms are supported.
RAMPING.PST	Sample **Ramping** Config file. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms are not supported. ▪ Multiple Work Offsets are supported. ▪ Cutter diameter compensation is supported.
ROBO_DRILL.PST	Sample **Robo_Drill** Config file. ▪ Drilling Subprograms are supported. ▪ Milling Subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter diameter compensation is supported.
SIEMENS_810.PST	Sample **Siemens** Config file. ▪ Drilling Subprograms are supported. ▪ Milling Subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter diameter compensation is supported.
SIEMENS_820.PST	Sample **Siemens** Config file. ▪ Drilling Subprograms are supported. ▪ Milling Subprograms are supported. ▪ Multiple Work Offsets are supported. ▪ Cutter diameter compensation is supported.
SIEMENS_840.PST	Sample **Siemens** Config file. ▪ Drilling Subprograms are supported. ▪ Milling Subprograms are supported. ▪ Multiple Work Offsets are supported.

	▪ Cutter diameter compensation is supported.
SUPERMAX.PST	Sample Config file for **Supermax CNC Mill**.
TAKSAWA.PST	Sample **Taksawa** Config file. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms are not supported. ▪ Multiple Work Offsets are supported. ▪ Cutter diameter compensation is supported.
TECHNO.PST	Sample Config file for **TECHNO** mill, which is basically a subset of the standard **FANUC** control language. ▪ Cutter compensation is not supported. ▪ Milling subprograms are not supported. ▪ Drilling subprograms are not supported.
TOSNUC.PST	Sample Config file for **Toshiba CNC Mill**.
TREE.PST	Sample Config file for **Tree CNC Mill**.
YANG.PST	Sample **Yang** Config file. ▪ Drilling Subprograms are not supported. ▪ Milling Subprograms are not supported. ▪ Multiple Work Offsets are supported.
YASNAC_MX.PST	Sample Config file for **YASNAC MX1 / MX2** controls. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Fanuc style **Milling** and **Drilling Subprograms** are supported. ▪ Milling and Drilling Subprograms can be enabled in ConfigPost's **General Info** dialog.

Post Library - Milling, 4-axis Rotary Table

ACRA 2100-V*.PST	Sample Config file for **Acra** controls. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face Cylinder is supported. ▪ Face 4-axis is supported.
DAEWOO-V*.PST	Sample Config file for a **Daewoo** machine. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face Cylinder is supported. ▪ Face 4-axis is supported.
DYNA_DELTA10-V*.PST	Sample Config file for **Dynapath** machines. ▪ Cutter diameter compensation is supported. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face 4-axis is supported.
FADAL_4525-V*.PST	Sample Config file for **Fadal** machines. ▪ Cutter diameter compensation is supported. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face 4-axis is supported.

HAAS_4_AXIS-V*.PST	Sample Config file for a **Haas** machine. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face Cylinder is supported. ▪ Face 4-axis is supported.
HARDINGE_GX-480-V*.PST	Sample Config file for a **Hardinge** machine. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face Cylinder is supported. ▪ Face 4-axis is supported.
OKUMA_OSP_E100L-v*.PST	Sample Config file for a **Okuma** machine. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face Cylinder is supported. ▪ Face 4-axis is supported.
TOYODA_FA450-V*.PST	Sample Config file for a **Toyoda** machine. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face Cylinder is supported. ▪ Face 4-axis is supported.

Post Library - Milling, 5-axis Rotary Table

BANDIT_ROUTER-V*.PST	Sample Config file for a **Bandit Router**. ▪ Cutter diameter compensation is supported. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face Cylinder is supported. ▪ Face 4-axis is supported. ▪ Face 5-axis is supported.
HAAS-5-AXIS-V*.PST	Sample Config file for a **Haas** machine. ▪ Cutter diameter compensation is supported. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face Cylinder is supported. ▪ Face 4-axis is supported. ▪ Face 5-axis is supported.
HARDINGE_VMC760-V*.PST	Sample Config file for a **Hardinge** machine. ▪ Cutter diameter compensation is supported. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face Cylinder is supported. ▪ Face 4-axis is supported. ▪ Face 5-axis is supported.

MORI_SUPERMILLER-V*.PST	Sample Config file for a **Mori** machine. ▪ Cutter diameter compensation is supported. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face Cylinder is supported. ▪ Face 4-axis is supported. ▪ Face 5-axis is supported.
ROBODRILL_5-AXIS-V*.PST	Sample Config file for **Robodrill** controls. ▪ Cutter diameter compensation is supported. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face Cylinder is supported. ▪ Face 5-axis is supported.
SUPERMAX_5-AXIS-V*.PST	Sample Config file for **Supermax** controls. ▪ Cutter diameter compensation is supported. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face Cylinder is supported. ▪ Face 5-axis is supported.

Post Library - Milling, Tombstone

MAKINO_A71_HORIZ-V*.PST	Sample Config file for a **Makino** machine. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face Cylinder is supported. ▪ Face 4-axis is supported.
MORI_MH50_HORIZONTAL-V*.PST	Sample Config file for a **Mori** machine. ▪ Cutter diameter compensation is supported. ▪ Uses M6 for tool change. ▪ Multiple work offsets (G54,G55,..) are supported. ▪ Face XY is supported. ▪ Face Polygon is supported. ▪ Face diameter index is supported. ▪ Face Cylinder is supported.

Post Library - Turning

ALLEN_BRADLEY .PST	**Allen Bradley** lathe control
DAEWOO.PST	**Daewoo** lathe with Fanuc control
DIAMETER.PST	Generic EIA control with X in diameter format
EZ-PATH_1.PST	**Bridgeport** lathe with an **EZ-Path** control
EZ-PATH_2.PST	**Bridgeport** lathe with **EZ-Path 2** control
FAGOR.PST	Lathe with **Fagor 8xxx** control
FALCON.PST	**Falcon** lathe
FANUC_6.PST	**Fanuc 6** controlled lathe
FANUC_10.PST	**Fanuc 10** controlled lathe
FANUC_11.PST	**Fanuc 11** controlled lathe
FANUC_16i.PST	**Fanuc 16 / 16i** controlled lathe
FANUC_18i.PST	**Fanuc 18 / 18i** controlled lathe
FANUC_20i.PST	**Fanuc 20 / 20i** controlled lathe
FANUC_21i.PST	**Fanuc 21 / 21i** controlled lathe
FANUC_30i.PST	**Fanuc 30 / 30i** controlled lathe
FANUC_31i.PST	**Fanuc 31 / 31i** controlled lathe
FANUC_32i.PST	**Fanuc 32 / 32i** controlled lathe
FANUC_O.PST	**Fanuc O** controlled lathe
FEMCO.PST	**Femco** CNC lathe
HAAS.PST	**Haas** CNC lathe
HARDINGE_CHNC .PST	**Hardinge CHNC** with Fanuc control
HARDINGE_COBR A.PST	**Hardinge Cobra** with Fanuc control

HARDINGE_GT_27.PST	**Hardinge GT27** with Fanuc control
HARDINGE_GT_27SP.PST	**Hardinge GT27 Super Precision** with Fanuc control
HITACHI.PST	**HITACHI Seiki** lathes
HYUNDAI.PST	**Hyundai Lathes** with a **Siemens 840** control
IKEGAI_TU.PST	**IKEGAI_TU** lathe
LEBLOND.PST	**LEBLOND** lathes with Fanuc control.
MAZAK_T_PLUS.PST	**Mazak** lathe with T32 Plus control in EIA.
MAZAK_FUSION.PST	**Mazak** lathe with Fusion control
MAZAK_MATRIX.PST	**Mazak** lathe with Matrix control
MACHINE_MATE.PST	Lathe with a **Machine Mate** control
MDSI.PST	**MDSI** controls applied to lathes
MITSUBISHI.PST	**Mitsubishi Meldas 6XX** series controlled lathe
MIYANO.PST	**MIYANO** lathes with Fanuc controls
OKUMA_CAPTAIN.PST	**Okuma Captain** lathe
OMNITURN.PST	**OMNI.PST** supports an **Omniturn** lathe
OKUMA_OSP5020.PST	**Okuma** OSP5000 control on lathe
OKUMA_OSP7000.PST	**Okuma** OSP7000 control on lathe
OKUMA_OSP8000.PST	**Okuma** OSP8000 control on lathe
PA8000.PST	**PA8000** controlled lathe
RADIUS.PST	Generic EIA lathe with X formatted for Radius
ROMI-G30HT.PST	**Romi G30-HT** lathe
SIEMENS_810.PST	Lathe with Siemens 810 control
SIEMENS_820.PST	Lathe with Siemens 820 control
SIEMENS_840.PST	Lathe with Siemens 840 control
STORM-C.PST	**Clausing STORM** lathe with a **FANUC O TC** control
TAKAMATSU.PST	**TAKAMATSU** lathe

WASINO.PST	**WASINO** lathe with Fanuc control
YANG.PST	**Yang** lathe with Fanuc control
YASNAC_MX.PST	**Lathe with YASNAC** MX1 / MX2 control

Post Library - WireEDM

AGIE_CLASSIC-V*.PST	Sample Config file for the 'classic' **AGIE G-code format,** which was used **prior** to the D Series machines.
AGIE_D_SERIES-V*.PST	Sample Config file for **D Series machine models: 100D, 200D, 300D**.
CHARMILLES_290-V*.PST	Sample Config file for a **Robofil Model 290** machine.
CHARMILLES_310-V*.PST	Sample Config file for a **Robofil Model 310** machine.
MITSUBISHI_FA-V*.PST	Sample Config file for **Mitsubishi FA Series** machines.
FANUC_ROBOCUT-V*.PST	Sample Config file for **Fanuc Robocut Alpha-iA Series** machines.
MITSUBISHI_FX20-V*.PST	Sample Config file for **Mitsubishi FX20** machines.
MAKINO_DUO_SERIES-V*.PST	Sample Config file for **Makino 'Wire Wizard' machines, DUO 43 or similar machines from the DUO Series,** software Version 7.04.
MITSUBISHI_C_H1_HA-V*.PST	Sample Config file for **Mitsubishi C / H1 / HA Series** machines.
PRIMA_E_SERIES–V*.PST	Sample Config file for **Ona Prima E Series** machines.
SODICK_LN1W-V*.PST	Sample Config file for **Sodick machines with a LN1W** control.

Using Config-By-Click

PartMaker's unique Config-by-Click feature provides a quick and easy way to customize Post files.

Using Config-by-Click, double-clicking on a line of NC code displayed in PartMaker takes you directly to the program format that is responsible for generating that line of NC code in ConfigPost.

To start using Config-by-Click, you must first enable it within PartMaker (see page 188).

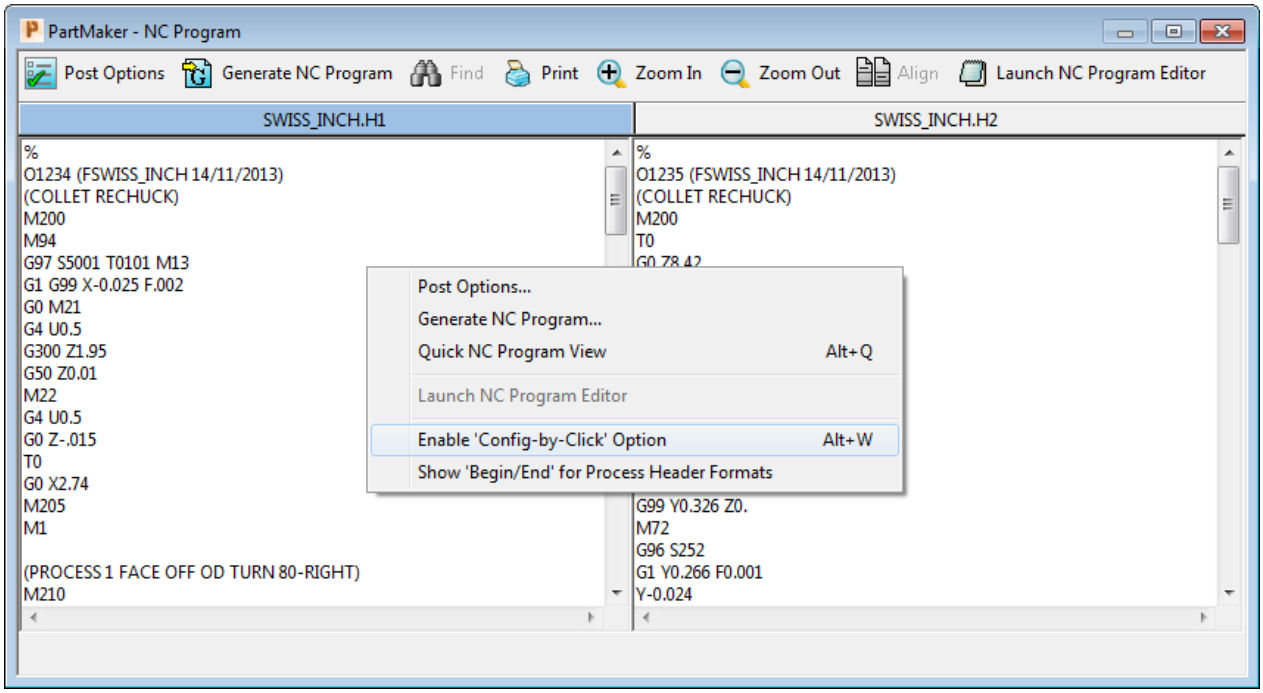


When using Config-by-Click to customize Post files, you may find it useful to select the option to automatically reload customized Post Configuration files (see page 194) in PartMaker.

Enabling Config-by-Click in PartMaker

To use Config-by-Click, you must first enable it within PartMaker by:

- Selecting **Job Optimizer > Configure Post Processor > Enable 'Config-by-Click' Option**
- Right-clicking in an NC Program window and selecting **Enable 'Config-by-Click' Option**.



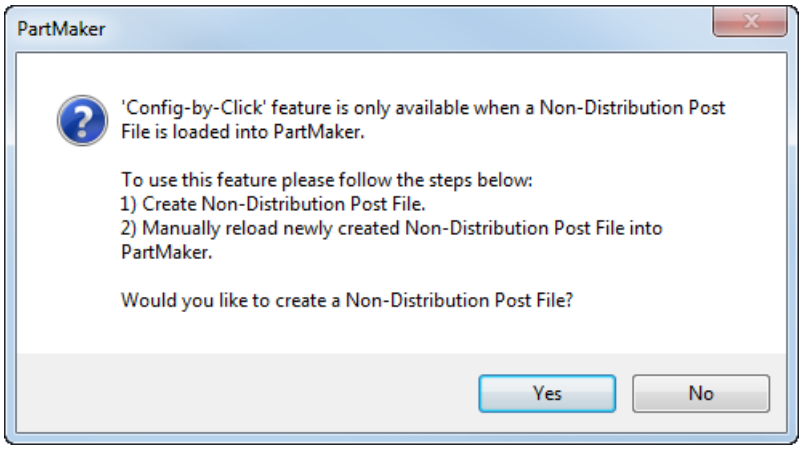
When you enable Config-by-Click, PartMaker regenerates the NC Program file to display the additional Config-by-Click information (see page 191) in the NC Program window.

The **Enable Config-by-Click** menu option acts as a toggle, so selecting the menu option again disables Config-by-Click. When you disable this option, PartMaker regenerates the NC Program file to hide the additional Config-by-Click information.

You can use Config-by-Click only with Non-Distribution Post files. If a Distribution Post file is loaded when you enable Config-by-Click, PartMaker prompts you to create a Non-Distribution Post file (see page 190).

Creating a Non-Distribution Post file for use with Config-by-Click

If you enable Config-by-Click when a Non-Distribution Post file (see page 3) is loaded in PartMaker/SwissCAM or PartMaker/Turn-Mill, PartMaker displays the following message:



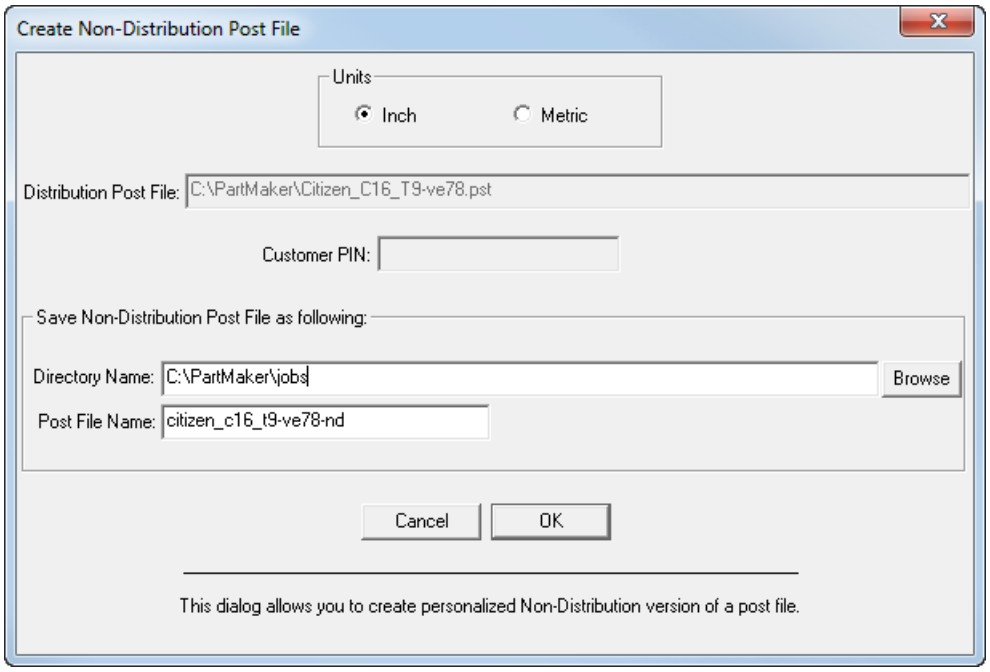
To create a Non-Distribution Post file, which you can edit using Config-by-Click:

1 Click **Yes**.

This displays the **Open Distribution Post File** dialog within ConfigPost.

2 In the **Open Distribution Post file** dialog, enter your **Customer PIN** and click **Create Non-Distribution Post File**.

This displays the **Create Non-Distribution Post File** dialog.





Your Customer PIN is supplied with your PartMaker installation. Contact PartMaker Support if you do not know your Customer PIN.

3 Check you have opened the correct Distribution Post file, by checking the following details:

- **Distribution Post File** displays the directory and filename of the Distribution Post File you opened.
- **Customer PIN** displays the Customer PIN you entered on the **Open Distribution Post File** dialog (see page 109).

If the details are incorrect, click **Cancel** to exit from the dialog. If the details are correct, continue to the next step.

4 Check the **Directory Name** where PartMaker will save the Non-Distribution Post file and the **Post File Name** for the file are correct. You can enter a different directory and/or filename, if required.

5 Click **OK**.

ConfigPost creates a new Non-Distribution Post (.pst) file (and .sub file, if required). It then opens the .pst file in ConfigPost.

6 To modify the Non-Distribution Post file you have just created using Config-by-Click, you must:

a Load the Non-Distribution Post file you have just created into PartMaker using the **Job Optimizer > Post Config File =** menu option.

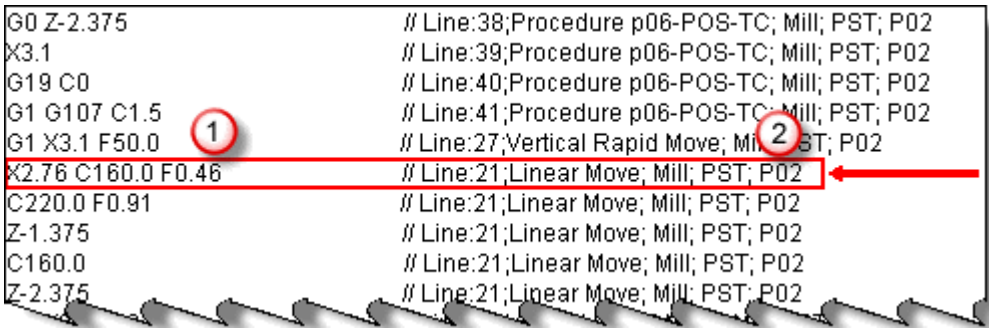
b Enable Config-by-Click (see page 188) in PartMaker.

c Use Config-by-Click (see page 191) to modify the file.

Using Config-by-Click to modify a Post file

When you have loaded a Non-Distribution Post File in PartMaker and enabled Config-by-Click, you can then use Config-by-Click to modify a Post file:

1 View the NC Program details in PartMaker's NC Program window:



shows the NC code output produced using the Post file appears in the left-hand column.

shows details about the Program Format in the Post file, which is responsible for generating that line of NC code output.

For the highlighted line in the example above, the G-code line in the left column corresponds to the following Post file information:

Line Number: Line 21

Format Name: Linear Move

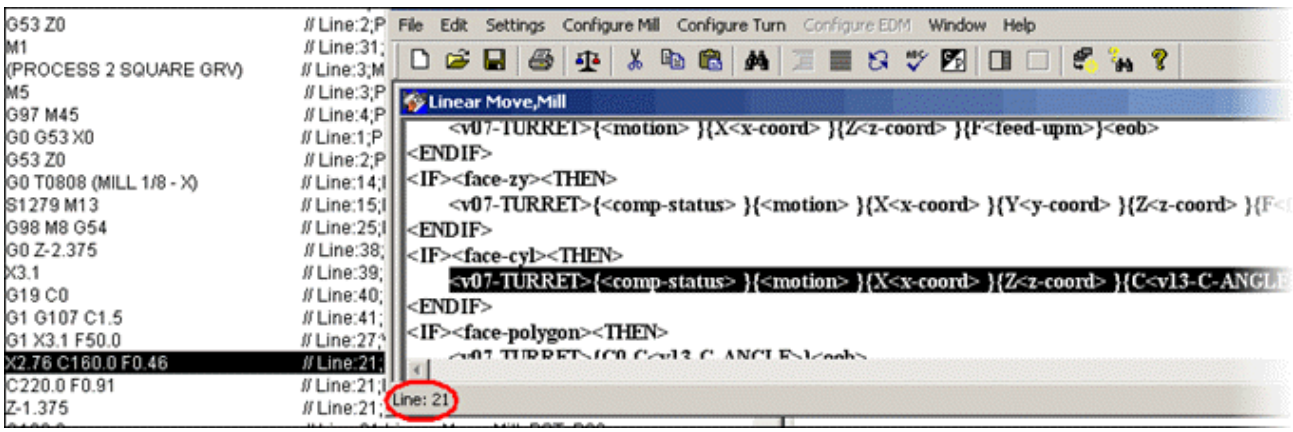
Mill or Turn: Mill

PST or SUB: PST

Proc ID: P02

2 To jump to the exact location in the Post file where the program format is used, double-click on that line in the NC Program window in PartMaker.

PartMaker then highlights the correct program format line in the Post file in ConfigPost. For example:



3 Within ConfigPost, modify the .pst or .sub file to achieve the desired output.



*Autodesk recommends that after **each** change you make to the Post file, you verify the changes made to the output by saving the Post file and regenerating the NC Program in PartMaker. Doing this prevents any mismatches between the line of code you want to modify and the location highlighted in ConfigPost.*

4 Before generating the final NC Program to run on your machine, disable the Config-by-Click feature (see page 188) in PartMaker.

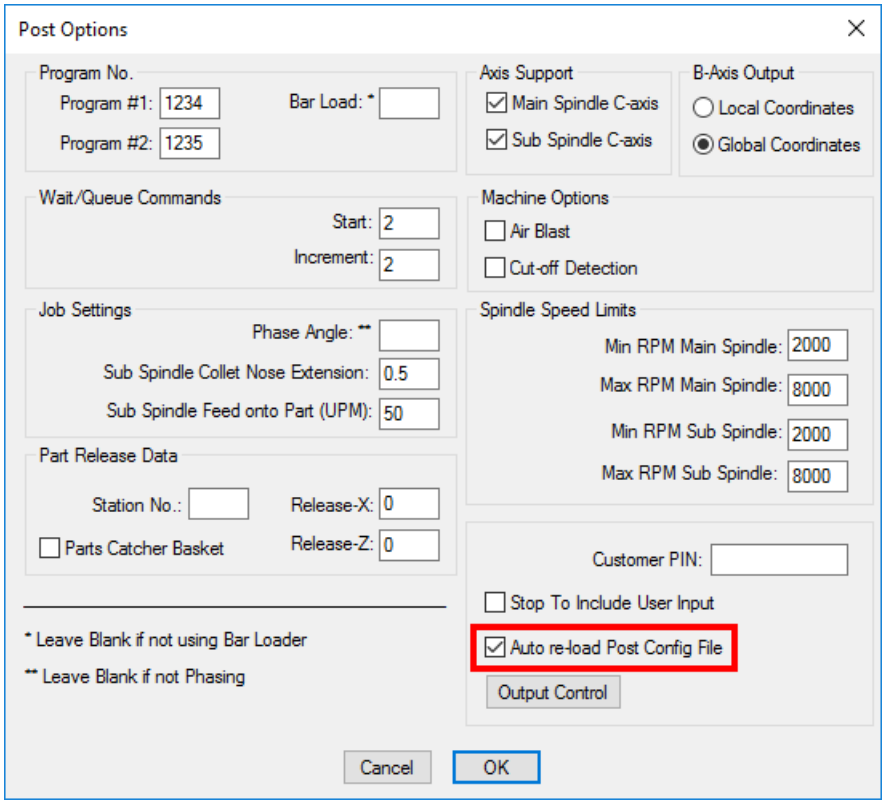
If you do not disable Config-by-Click before generating the final NC Program, the NC Program file will contain program format locations.



You can access Config-by-Click only from PartMaker's NC Program window. You cannot access Config-by-Click when viewing NC Program files in other applications, such as Notepad, MS Word or third-party CNC editors (such as, WinCNC and Predator).

Automatically reloading customized Post files in PartMaker

When you are customizing a Post file, it may find it useful to select the **Auto-Reload Post Config File** option on the **Post Options** dialog in PartMaker, so PartMaker automatically reloads the file you have customized in ConfigPost when you generate NC program output.



When the **Auto-Reload Post Config File** option is:

- **selected**, PartMaker automatically reloads the currently loaded Post file (specified using **Job Optimizer > Post Config File =** option) when you generate an NC program using the **Job Optimizer > Generate NC Program** option.
- **deselected**, you must use the **Job Optimizer > Post Config File =** option to reload the Post file into PartMaker each time you customize it, so you can generate an NC program using it.



*The **Auto-Reload Post Config File** option reloads only the Post file that is currently loaded, using the **Job Optimizer > Post Config File =** option. If you save the customized Post file with a new filename in ConfigPost, you must reload the new Post file manually in PartMaker.*

ASCII Codes

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		

Autodesk Legal Notice

© 2021 Autodesk, Inc. All Rights Reserved. Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License that can be viewed online at http://creativecommons.org/licenses/by-nc-sa/3.0/. This license content, applicable as of 16 December 2014 to this software product, is reproduced here for offline users:

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. **"Adaptation"** means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. **"Collection"** means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(g) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. **"Distribute"** means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. **"License Elements"** means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, Noncommercial, ShareAlike.

e. **"Licensor"** means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

f. **"Original Author"** means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

g. **"Work"** means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

h. **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

i. **"Publicly Perform"** means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

j. **"Reproduce"** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights described in Section 4(e).

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(d), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(d), as requested.

b. You may Distribute or Publicly Perform an Adaptation only under: (i) the terms of this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-NonCommercial-ShareAlike 3.0 US) ("Applicable License"). You must include a copy of, or the URI, for Applicable License with every copy of each Adaptation You Distribute or Publicly Perform. You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License. You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.

c. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.

d. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and, (iv) consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(d) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License if Your exercise of such rights is for a purpose or use which is otherwise than noncommercial as permitted under Section 4(c) and otherwise waives the right to collect royalties through any statutory or compulsory licensing scheme; and,

iii. Voluntary License Schemes. The Licensor reserves the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License that is for a purpose or use which is otherwise than noncommercial as permitted under Section 4(c).

f. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING AND TO THE FULLEST EXTENT PERMITTED BY APPLICABLE LAW, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THIS EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at http://creativecommons.org/.

Certain materials included in this publication are reprinted with the permission of the copyright holder.

Creative Commons FAQ

Autodesk's Creative Commons FAQ can be viewed online at https://knowledge.autodesk.com/customer-service/share-the-knowledge, and is reproduced here for offline users.

Creative Commons is a simple, open licensing model which allows individuals to freely modify, remix, and share digital content created for learning and support.

Borrow from the Autodesk Learning, Support and Video libraries to build a new learning experience for anyone with any particular need or interest. It’s out there. You can use it. It’s yours.

In collaboration with Creative Commons, Autodesk invites you to share your knowledge with the rest of the world, inspiring others to learn, achieve goals, and ignite creativity.

What is Creative Commons?

Creative Commons (CC) is a nonprofit organization that offers a simple licensing model that frees digital content to enable anyone to modify, remix, and share creative works.

How do I know if Autodesk learning content and Autodesk University content is available under Creative Commons?

All Autodesk learning content and Autodesk University content released under Creative Commons is explicitly marked with a Creative Commons icon specifying what you can and cannot do. Always follow the terms of the stated license.

What Autodesk learning content is currently available under Creative Commons?

Over time, Autodesk will release more and more learning content under the Creative Commons licenses.

Currently available learning content:

- Autodesk online help-Online help for many Autodesk products, including its embedded media such as images and help movies.
- Autodesk Learning Videos-A range of video-based learning content, including the video tutorials on the Autodesk YouTube™ Learning Channels and their associated iTunes® podcasts.
- Autodesk downloadable materials-Downloadable 3D assets, digital footage, and other files you can use to follow along on your own time.

Is Autodesk learning and support content copyrighted?

Yes. Creative Commons licensing does not replace copyright. Copyright remains with Autodesk or its suppliers, as applicable. But it makes the terms of use much more flexible.

What do the Autodesk Creative Commons licenses allow?

Autodesk makes some of its learning and support content available under two distinct Creative Commons licenses. The learning content is clearly marked with the applicable Creative Commons license. You must comply with the following conditions:

- **Attribution-NonCommercial-ShareAlike (CC BY-NC-SA)** This license lets you copy, distribute, display, remix, tweak, and build upon our work noncommercially, as long as you credit Autodesk and license your new creations under the identical terms. Terms of this license can be viewed online at https://creativecommons.org/licenses/by-nc-sa/3.0/us/
- **Attribution-NonCommercial-No Derivative Works (CC BY-NC-ND)** This license lets you copy, distribute, and display only verbatim copies of our work as long as you credit us, but you cannot alter the learning content in any way or use it commercially. Terms of this license can be viewed online at https://creativecommons.org/licenses/by-nc-nd/3.0/us/deed.en_US
- **Special permissions on content marked as No Derivative Works** For video-based learning content marked as No Derivative Works (ND), Autodesk grants you special permission to make modifications but only for the purpose of translating the video content into another language.

These conditions can be modified only by explicit permission of Autodesk, Inc. Send requests for modifications outside of these license terms to *creativecommons@autodesk.com*.

Can I get special permission to do something different with the learning content?

Unless otherwise stated, our Creative Commons conditions can be modified only by explicit permission of Autodesk, Inc. If you have any questions or requests for modifications outside of these license terms, email us at *creativecommons@autodesk.com*.

How do I attribute Autodesk learning content?

You must explicitly credit Autodesk, Inc., as the original source of the materials. This is a standard requirement of the Attribution (BY) term in all Creative Commons licenses. In some cases, such as for the Autodesk video learning content, we specify exactly how we would like to be attributed.

This is usually described on the video's end-plate. For the most part providing the title of the work, the URL where the work is hosted, and a credit to Autodesk, Inc., is quite acceptable. Also, remember to keep intact any copyright notice associated with the work. This may sound like a lot of information, but there is flexibility in the way you present it.

Here are some examples:

"This document contains content adapted from the Autodesk® Maya® Help, available under a Creative Commons Attribution-NonCommercial-Share Alike license. Copyright © Autodesk, Inc."

"This is a Finnish translation of a video created by the Autodesk Maya Learning Channel @ www.youtube.com/mayahowtos. Copyright © Autodesk, Inc."

"Special thanks to the Autodesk® 3ds Max® Learning Channel @ www.youtube.com/3dsmaxhowtos. Copyright © Autodesk, Inc."

Do I follow YouTube's standard license or Autodesk's Creative Commons license?

The videos of the Autodesk Learning Channels on YouTube are uploaded under YouTube's standard license policy. Nonetheless, these videos are released by Autodesk as Creative Commons Attribution-NonCommercial-No Derivative Works (CC BY-NC-ND) and are marked as such.

You are free to use our video learning content according to the Creative Commons license under which they are released.

Where can I easily download Autodesk learning videos?

Most of the Autodesk Learning Channels have an associated iTunes podcast from where you can download the same videos and watch them offline. When translating Autodesk learning videos, we recommend downloading the videos from the iTunes podcasts.

Can I translate Autodesk learning videos?

Yes. Even though our learning videos are licensed as No Derivative Works (ND), we grant everyone permission to translate the audio and subtitles into other languages. In fact, if you want to recapture the video tutorial as-is but show the user interface in another language, you are free to do so. Be sure to give proper attribution as indicated on the video's Creative Commons end-plate. This special permission only applies to translation projects. Requests for modifications outside of these license terms can be directed to creativecommons@autodesk.com.

How do I let others know that I have translated Autodesk learning content into another language?

Autodesk is happy to see its learning content translated into as many different languages as possible. If you translate our videos or any of our learning content into other languages, let us know. We can help promote your contributions to our growing multilingual community. In fact, we encourage you to find creative ways to share our learning content with your friends, family, students, colleagues, and communities around the world. Contact us at creativecommons@autodesk.com.

I have translated Autodesk learning videos into other languages. Can I upload them to my own YouTube channel?

Yes, please do and let us know where to find them so that we can help promote your contributions to our growing multilingual Autodesk community. Contact us at creativecommons@autodesk.com.

Can I repost or republish Autodesk learning content on my site or blog?

Yes, you can make Autodesk learning material available on your site or blog as long as you follow the terms of the Creative Commons license under which the learning content is released. If you are simply referencing the learning content as-is, then we recommend that you link to it or embed it from where it is hosted by Autodesk. That way the content will always be fresh. If you have translated or remixed our learning content, then by all means you can host it yourself. Let us know about it, and we can help promote your contributions to our global learning community. Contact us at creativecommons@autodesk.com.

Can I show Autodesk learning content during my conference?

Yes, as long as it's within the scope of a noncommercial event, and as long as you comply with the terms of the Creative Commons license outlined above. In particular, the videos must be shown unedited with the exception of modifications for the purpose of translation. If you wish to use Autodesk learning content in a commercial context, contact us with a request for permission at creativecommons@autodesk.com.

Can I use Autodesk learning content in my classroom?

Yes, as long as you comply with the terms of the Creative Commons license under which the learning material is released. Many teachers use Autodesk learning content to stimulate discussions with students or to complement course materials, and we encourage you to do so as well.

Can I re-edit and remix Autodesk video learning content?

No, but for one exception. Our Creative Commons BY-NC-ND license clearly states that "derivative works" of any kind (edits, cuts, remixes, mashups, and so on) are not allowed without explicit permission from Autodesk. This is essential for preserving the integrity of our instructors' ideas. However, we do give you permission to modify our videos for the purpose of translating them into other languages.

Can I re-edit and remix Autodesk downloadable 3D assets and footage?

Yes. The Autodesk Learning Channels on YouTube provide downloadable 3D assets, footage, and other files for you to follow along with the video tutorials on your own time. This downloadable material is made available under a Creative Commons Attribution-NonCommercial-ShareAlike (CC BY-NC-SA) license. You can download these materials and experiment with them, but your remixes must give us credit as the original source of the content and be shared under the identical license terms.

Can I use content from Autodesk online help to create new materials for a specific audience?

Yes, if you want to help a specific audience learn how to optimize the use of their Autodesk software, there is no need to start from scratch. You can use, remix, or enrich the relevant help content and include it in your book, instructions, examples, or workflows you create, then Share-Alike with the community. Always be sure to comply with the terms of the Creative Commons license under which the learning content is released.

What are the best practices for marking content with Creative Commons Licenses?

When reusing a CC-licensed work (by sharing the original or a derivative based on the original), it is important to keep intact any copyright notice associated with the work, including the Creative Commons license being used. Make sure you abide by the license conditions provided by the licensor, in this case Autodesk, Inc.

Trademarks

The following are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and other countries: 123D, 3ds Max, ADSK, Alias, ArtCAM, ATC, AutoCAD LT, AutoCAD, Autodesk, the Autodesk logo, Autodesk 123D, Autodesk Alias, Autodesk Forge, Autodesk Fusion, Autodesk Inventor, AutoSnap, BIM 360, Buzzsaw, CADmep, CAMduct, Civil 3D, Configurator 360, Dancing Baby (image), DWF, DWG, DWG (design/logo), DWG Extreme, DWG TrueConvert, DWG TrueView, DWGX, DXF, Eagle, ESTmep, FBX, FeatureCAM, Flame, FormIt 360, Fusion 360, The Future of Making Things, Glue, Green Building Studio, InfraWorks, Instructables, Instructables (stylized robot design/logo), Inventor, Inventor HSM, Inventor LT, Make Anything, Maya, Maya LT, Moldflow, MotionBuilder, Mudbox, Navisworks, Netfabb, Opticore, PartMaker, Pier 9, PowerInspect, PowerMill, PowerShape, Publisher 360, RasterDWG, RealDWG, ReCap, ReCap 360, Remake, Revit LT, Revit, Scaleform, Shotgun, Showcase, Showcase 360, SketchBook, Softimage, Tinkercad, TrustedDWG, VRED.

All other brand names, product names or trademarks belong to their respective holders.

Disclaimer

THIS PUBLICATION AND THE INFORMATION CONTAINED HEREIN IS MADE AVAILABLE BY AUTODESK, INC. "AS IS." AUTODESK, INC. DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE REGARDING THESE MATERIALS.



Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. Please see the Autodesk Creative Commons FAQ for more information