

Version 9

Mastercam®

MP Post Processor Reference Guide

Volume 1
Basic MP Post Processing

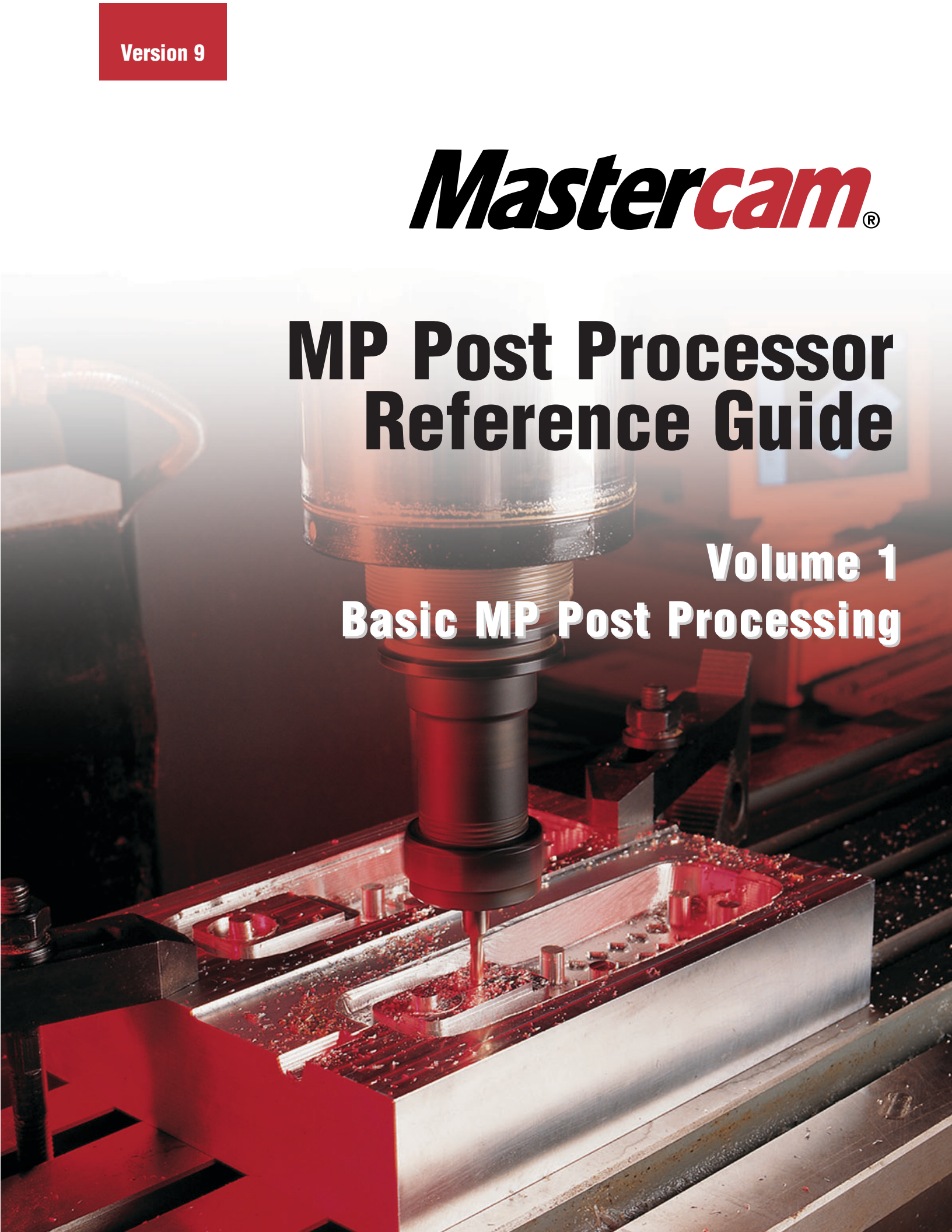


Table of Contents

1 – How to Use the MP Post Processor Reference Guide.....	1-1
How to contact CNC Software	1-1
Additional resources	1-2
2 – Introduction to MP Post Processors	2-1
What is an MP post?	2-1
How does an MP post work?	2-2
Post processing in greater detail.....	2-3
Reading the NCI file.....	2-3
Postblocks.....	2-4
What files do I need for the posting process?	2-5
The post customization file (*.PST).....	2-5
The post text file (*.TXT)	2-5
The post bin file (*.PSB).....	2-6
The post executable file (*.DLL).....	2-6
What are the components of the MP language?	2-6
Variables	2-7
String literals	2-7
Command variables	2-8
Postblocks.....	2-8
Postlines	2-8
Conditional branching	2-10
Mathematical operators	2-10
Formula functions	2-10
User prompts	2-10
Post comments	2-11
String select function.....	2-11
Lookup tables.....	2-11
Buffer files	2-12
Support for subprograms and canned cycles	2-12
System numbered questions	2-12
How is an MP post organized?	2-12
Definition area.....	2-13
Postblock area	2-14
System numbered question area	2-14

Sample post: MP_EZ.PST	2-14
Post modification guidelines	2-18
MP language syntax rules	2-19
General post formatting rules	2-19
Variable initialization rules	2-20
Postblock and postline rules	2-20
Limits	2-21
Common post tasks or How do I ?	2-22
3 – Changes to the MP Post.....	3-1
Tool change after technique	3-1
Enabling the tool change after technique	3-1
Postblock recall status	3-2
Do not use the rapid position variables (<i>xr, yr, zr</i>)	3-4
Absolute / incremental support	3-4
X, Y and Z output	3-5
The use of Whatno	3-6
Posts transition from Version 7 to Version 8 and 9	3-6
NCI Header Gcode 1050	3-6
Updating posts for Version 9 (UPDATEPST9)	3-7
Version 8 canned text	3-8
V8 Lathe canned turning cycles	3-8
V9 changes	3-8
Troubleshooting upgrading	3-8
Common problems	3-8
Changes to the MP post executable	3-9

4 – Post Debugging Techniques	4-1
Pre-defined post debugging switches	4-1
The bug1 variable	4-1
The bug2 variable	4-2
The bug3 variable	4-4
The bug4 variable	4-4
Error messages in MP	4-4
Common error messages	4-5
Using the tilde (~) modifier to force variable output	4-8
Subprograms and Lathe canned cycle output.....	4-8
Debugging posts with buffer files	4-9
Notes on no_updprv and nc_lout	4-9
5 – Using Numbered Questions.....	5-1
Overview	5-1
Numbered question format.....	5-1
Using the post to access the response to a numbered question	5-2
Predefined numbered questions	5-2
6 – Using the Post Text File	6-1
Overview	6-1
MILL9.TXT	6-1
Example post text file modifications	6-1
Customizing the Miscellaneous Values dialog box	6-1
Customizing the Drill Cycle parameters dialog box	6-4
Customizing the Canned Text dialog box	6-6
Additional customization	6-6
7 – Postblocks.....	7-1
The postblock role.....	7-1
Postblock structure	7-2
Traditional postblocks	7-3
Bracketed postblocks.....	7-3
Postlines	7-3
Postline format rules	7-3
Postblock declaration.....	7-4
Traditional postblocks	7-4
Bracketed postblocks.....	7-5
Postblock types.....	7-5
User-defined postblocks	7-5
Predefined postblocks.....	7-5

Postlines	7-8
Postline types.....	7-8
Postline statements.....	7-9
Restrictions on using postline types and statements	7-15
Postblock/postline examples.....	7-16
Postblock rules summary.....	7-18
Postline rules summary.....	7-18
Postblock call dependencies.....	7-19
Postblock calls for Gcode 0 and 1 in Mill/Router/Lathe	7-19
Postblock calls for Gcode 2 and 3 in Mill/Router/Lathe	7-21
Postblock calls for Gcode 11 in Mill Router/	7-21
Postblock calls for Gcode 81 and 100 in Mill/ Router/Lathe	7-23
Postblock calls for tool end in Mill/ Router/Lathe	7-26
Postblock calls for Gcode 0, 1, 20, 21 in Wire	7-26
Postblock calls for Gcode 2, 3, 22, 23 in Wire	7-27
8 – How Preparatory Postblocks Work.....	8-1
The preparatory postblocks.....	8-1
Enabling the preparatory postblocks.....	8-2
Setting the tooltable variable.....	8-2
Effect of the tooltable variable on postblock calls	8-3
What happens during the preparatory processing procedure	8-3
Preparatory processing flowchart	8-4
The pwrtp preparatory processing procedure	8-5
The ptooltbl preparatory processing procedure	8-6
Ending the preparatory processing procedure	8-8
Using tooltable as a command variable	8-8
Using the taper and radius list in Wire	8-9
Enabling the taper and radius lists.....	8-9
Taper and radius list specifics.....	8-10
9 – Numeric Variables.....	9-1
Numeric variable label/initialization	9-1
Initialization (:):.....	9-2
Global formulas (=)	9-2
Examples	9-2
Format assignment	9-3
Implied numeric variable definitions.....	9-3
Value in formula statement or conditional branching statement	9-3

ASCII literal	9-3
Formatting numeric variables	9-4
Format statements	9-4
Format assignments	9-9
When a format statement/assignment is not made	9-10
Default format assignments	9-10
Numeric formatting rules summary	9-11
Modality with numeric variables	9-12
Forcing output of modal numeric variables	9-12
Numeric variable output statements	9-12
Numeric variables	9-13
Forced numeric variables (*)	9-13
Update numeric variables (!)	9-13
Format numeric variables (@)	9-13
Dependent numeric variables (`)	9-13
Debug numeric variables (~)	9-14
Command variables	9-14
Previous numeric variable	9-14
Numeric variable types	9-15
Numeric variables in formulas and conditional branching	9-16
10 – String Variables	10-1
String labeling and initialization	10-1
Implied string definitions	10-2
String literal	10-2
Dependent string literal	10-3
ASCII literal	10-3
Null character definition	10-3
String variable output statements	10-4
String variable	10-4
String select function call	10-5
String literal	10-6
ASCII literal	10-7
String functions	10-7
Capturing system numbered questions	10-9
11 – Post Functions	11-1
Difference between formulas and functions	11-1
MP post functions	11-1
fnewsub function	11-2

String select function (fstrel)	11-3
String select tables.....	11-3
String Select Function (fstrsel) description	11-3
Using upstr.....	11-4
Lookup table function flktbl	11-5
Table definition.....	11-5
Table formulas flook, frange, and finc.....	11-5
flook	11-6
frange.....	11-7
finc	11-7
Fleader command/function.....	11-8
Force function	11-9
12 – Formulas.....	12-1
Difference between formulas and functions.....	12-1
Formula usage	12-1
Assignment formula types.....	12-1
Assignment formula usage	12-2
Boolean formulas	12-3
Boolean formula usage	12-4
Definitions	12-4
Basic definitions	12-4
Vector and matrix definitions.....	12-6
Arrays in MP	12-6
Entity and variable type definitions	12-6
Formula structure	12-8
Nesting types	12-9
Understanding vector and matrix functions.....	12-10
Precedence of operation	12-11
Boolean formula usage	12-11
Rules for using “if”, “while” and “else” conditional branching statements.....	12-13
MP post formula types	12-14
Post formulas by type.....	12-16
13 – The NCI File	13-1
NCI file structure	13-1
NCI code definitions.....	13-2
Additional calculated variables.....	13-3
NCI Gcode Summaries	13-4

L Lathe NCI Gcodes	13-4
M Mill NCI Gcodes	13-6
R Router NCI Gcodes	13-7
W Wire NCI Gcodes	13-8
14 – Getting the Next Move	14-1
Enabling the get next file routine	14-1
Variables captured	14-3
Reading the “get next file”	14-4
Special cases	14-4
15 – Working with Drill Cycles	15-1
Version 8 NCI drill cycle format changes	15-1
The new NCI drill data	15-1
Processing with rpd_typ_v7	15-12
Processing with rpd_typ_v7 enabled (old form)	15-12
Processing with rpd_typ_v7 disabled (new form)	15-12
Processing long code	15-13
Simple drill (drillcyc = 0)	15-14
Peck drill (drillcyc = 1)	15-15
Chip break drill (drillcyc = 2)	15-16
Tap (drillcyc = 3)	15-17
Bore, feed out (drillcyc = 4)	15-18
Bore, stop, rapid out (drillcyc = 5)	15-19
Bore, fine (drillcyc = 6)	15-20
Bore, standard (drillcyc = 7)	15-21
Cancel drill cycles	15-21
Custom drill cycles	15-22
Version 9.1 drill cycle sample	15-23
Drill cycle variables	15-27
16 – Working with Ccontext	16-1
New canned text for Version 8	16-1
The 1025 canned text line	16-1
Format of the 1025 canned text line in the NCI	16-2
Cant_pos1 ... cant_pos10	16-2
cant_val1 ... cant_val10	16-2
cant_no	16-3
Customizing the post .TXT file to display text in Mastercam	16-3
Associating cant_val with the correct output string	16-4

Adding canned text options to a Version 8 (or 9) MP post.....	16-7
17 – File Manipulation.....	17-1
NC program output files	17-1
Controlling alternate file defaults.....	17-1
Changing default output file extensions	17-1
Changing default output file names	17-2
Setting output file paths	17-2
Managing output files	17-2
Selecting an output file.....	17-2
Opening output files	17-3
Merging output files.....	17-4
Clearing output files	17-5
Changing the output file	17-5
Closing output files.....	17-6
General file manipulation formulas.....	17-6
General file manipulation techniques	17-6
Closing and reopening an alternate file with subprograms	17-6
Renaming the NC file after posting	17-7
Opening an alternate file as a buffer file	17-8
18 – How Prompt Questions Work	18-1
Overview	18-1
Format of a prompt question in the post	18-1
Question definition for a numeric response	18-2
Question definition for string response	18-2
Inserting a question into a postline	18-2
Example	18-2
Merging files through prompt questions	18-3
19 – Posting Comments into the NC File.....	19-1
Using pcomment to format comments	19-1
Manual entry postblock example	19-1
Using scomm to capture comments.....	19-2
Using the commlng variable	19-2
Using the commvar variable	19-2
Comments and the tool change block.....	19-3
Using the comment command with tool changes	19-3
Example using the comment command.....	19-4

How the post processes comments 19-4

Using the pscomm0 postblock..... 19-5

A – Glossary..... A-1

1

How to Use the MP Post Processor Reference Guide

Welcome to Volume 1 of the MP Post Processor Reference Guide. The Guide contains information that will assist you in modifying post processors written in the MP language, CNC Software's proprietary scripting language for Mill, Lathe, Wire, and Router applications. It is not intended to teach you how to write a post processor. Your Mastercam dealer is your best source of assistance in developing and customizing postprocessors.

The MP Post Processor Reference Guide consists of three volumes:

Volume 1, Basic MP Post Processor Programming. Presents basic concepts, principles, and rules you must understand to modify an MP post processor: postblocks, variables, formulas, functions, NCI Gcode, drill cycles, canned text, etc. If you are planning to modify an MP post processor or just want to understand how it is organized, you should read this volume.

Volume 2, Advanced MP Language Concepts and Applications. Describes how the MP language implements special applications including subprograms, mill/turn applications, lathe canned and threading cycles, and wire processing. Also discusses how MP language uses buffer files, handles intersections, and breaks up motion.

Volume 3, MP Post Language Reference. Contains detailed descriptions of syntax and dependencies of postblocks, variables, formulas, NCI Gcode, and parameters.

Volume 1 contains a [Glossary](#).

Note: The MP Post Processor Reference Guide is designed to be used online, although you can also print it.

How to contact CNC Software

Important Information	
Address	CNC Software, Inc. 671 Old Post Road Tolland, Connecticut, 06084-2830 USA
Phone	(860) 875-5006
Fax	(860) 872-1565
Internet Address	http://www.mastercam.com
E-mail	support@mastercam.com

Additional resources

- ◆ Please email any comments, suggests and/or corrections to the MP Post documentation to posts@mastercam.com
- ◆ For an ongoing discussion of Mastercam-related topics, visit the eMastercam online forum at www.emastercam.com