



FAQ #34: How do I edit my post processor?

Q: The post processor I'm using is very close to exactly what I want. I'd still like to make a few minor changes to it, though. How do I do this?

A: It is absolutely impossible to describe every change that can be made to post processors in BobCAD-CAM in any printed or online document, but it is definitely not impossible to describe how it's done. There are two methods available to edit them: using the post editor utilities shipped with the software and using a text editor to edit the raw files directly. The utility method used by this document is the simpler of the two methods, though the text editor method is more direct and more involved.

NOTE: Free post processor modifications and post creation is included with the BobCAD-CAM software purchase. Please contact Technical Support at (727) 489-0003 for information regarding a post processor for machine controllers. This document is intended for those who wish to modify or create post processors, and assumes that the end user is already familiar with Windows operating system software. Please also note that this document assumes the machine type to be a mill and that it covers only editing the post processor configuration file itself. Editing post processors for lathe or EDM machines follows all of the same procedures and differs only in the specifics of the machine type and available variables but is otherwise identical. Setting up the full machine configuration including assigning the post processor to a configured machine is beyond the scope of this document and covered elsewhere.

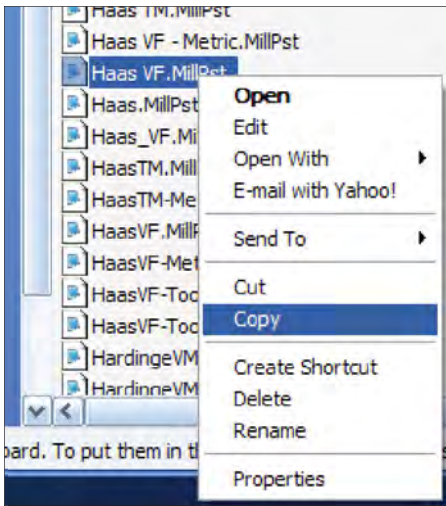
IMPORTANT: Be certain to create a copy of your post processor and rename the file to avoid overwriting the original, as saved changes cannot be easily undone!

Step 1: Open the desired post processor configuration file with the appropriate post editor utility.

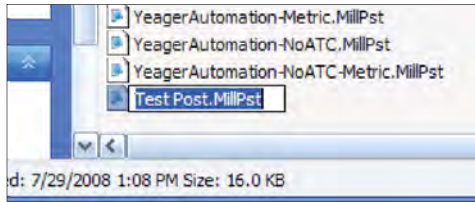
Navigate to the proper folder containing the post processors. For users of BobCAD-CAM V22, the correct folder will be C:\Program Files\BobCAD-CAM\BobCAD-CAM V22\Posts\ in the standard install. Note that if BobCAD-CAM is installed to a different location, this folder will be different, but will still contain the Posts subfolder. From there, the post processor files are separated by machine type in the Mill, Lathe, and WireEDM folders. For V23 users, the root folder for the files will normally be C:\Program Files\BobCAD-CAM\BobCAD-CAM V23\Posts\. Editing the post processors is the same between these versions, so this document covers both.

In each of the machine type folders, there is a utility for opening and editing the post processors. The files themselves are simple ASCII text files with different extensions matching the machine type they're intended for, i.e., .MillPst, .LathePst, and .EdmPst. They are named MillEditPost.exe, LatheEditPost.exe, and EdmEditPost.exe, respectively.

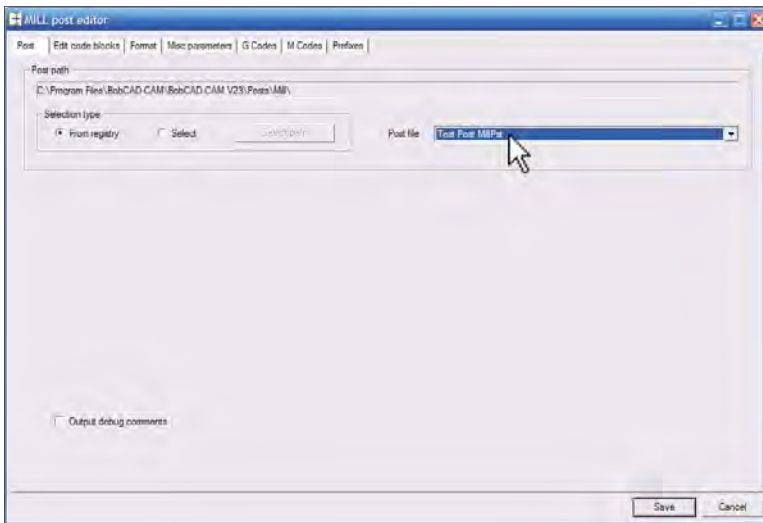
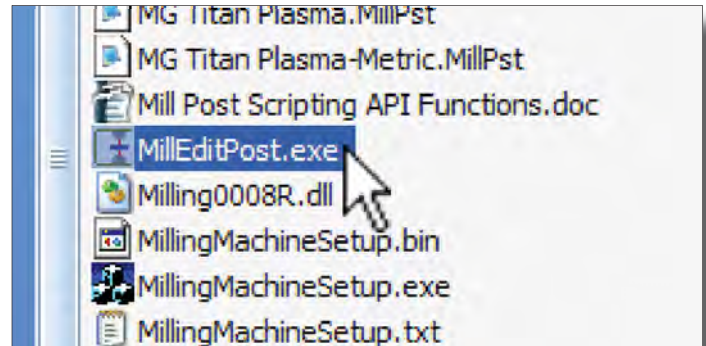
The utilities for opening the files are also named for the machine type: MillEditPost.exe, LatheEditPost.exe, and EdmEditPost.exe. Each of them resides directly in the appropriate folder under Posts, in the same folder as the post processor files it's intended to open.



Create a copy of the post to edit and rename it before editing.



Double click on the post editor utility in the same folder as the post file.



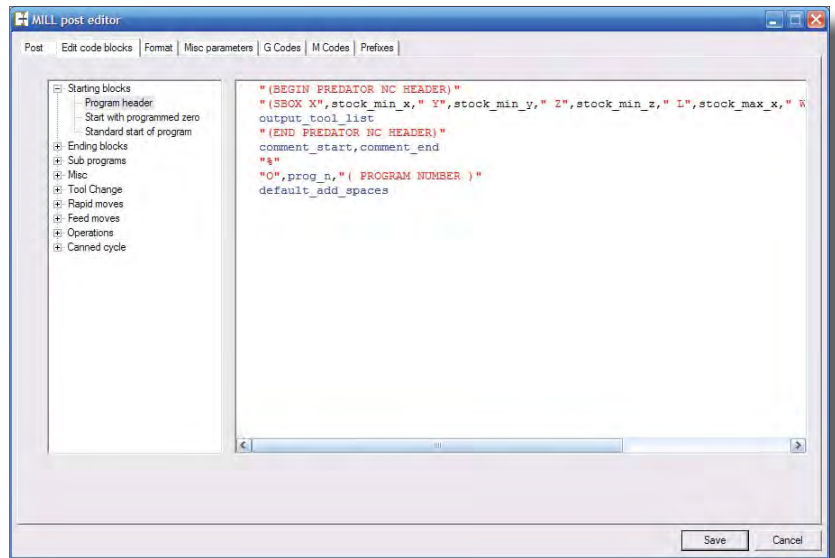
In the **Post** tab of the editor, under **Post file**, choose the newly copied and renamed file. Once it is chosen, the editor will already have it open and ready to edit.

Step 2: Begin editing the post processor file, one tab of the editor at a time.

NOTE: Most post editing is only very minor, requiring only to move existing variables around, change or add hard-coded output text.

Note in the lower left corner of the **Post** tab is the **Output debug comments** check box. This is a handy feature for determining from which code block the resultant NC code is generated. It will be discussed in detail further in this document. The following examples will be based upon the Haas VF.MillPst post processor. It has been selected for its complete use of standardized G and M codes.

After selecting the desired post to be modified, click on the **Edit code blocks** tab. In these fields is where the user may edit the various information to generate the desired machine tool NC output as generated by the BobCAD-CAM software.



The blocks are programmed using variables and commands, normally displayed in *blue*. Characters in quotes will be hard-coded and displayed in *red*. A variable may be encountered in *black*; this only means that the variable name is either misspelled or is not defined in the PostingVariables.txt file in the same folder as the post processor file. Even if a variable or command is displayed in black and not defined in the PostingVariables.txt file, *it still may function!* For this reason, do not automatically assume that because the variable is displayed in black that it is misspelled.

The above example is information contained in the **Program header** under **Starting blocks**, and in this example is used to set up for the Predator NC Header. It is used to define the basic stock geometry and tool definitions for the **Backplot** function within the Predator CNC Editor – Level2 software and since this information is located prior to the hard-coded “%”, it will be disregarded by the controller (though not *ALL* controllers; check the individual machine’s documentation to verify that the controller will ignore leading comments in an NC file). The code blocks are written as follows:

```

"(BEGIN PREDATOR NC HEADER)"
output_tool_list
"(SBOX X",stock_min_x," Y",stock_min_y," Z",stock_min_z," L",stock_max_x," W",stock_max_y,
" H",stock_max_z," )"
"(END PREDATOR NC HEADER)"
comment_start," ",comment_end
"%"
"O",prog_n,"( PROGRAM NUMBER )"
default_add_spaces

```

The corresponding output will be as follows:

```

(BEGIN PREDATOR NC HEADER)
(MTOOL T1 S1 D.5 H5.)
(MTOOL T2 S1 D.625 H5.)
(MTOOL T3 S1 D.125 H5.)
(MTOOL T4 S1 D.201 H5.)
(SBOX X-2.25 Y-2.25 Z-1. L4.5 W4.75 H1.)
(END PREDATOR NC HEADER)
( )
%
O12345 ( PROGRAM NUMBER )

```

The **Start with programmed zero** and **Standard start of program** under **Starting blocks** should contain the same code block layout. Currently the software will only reference the **Standard start of program**; the **Start with programmed zero** field is for future development. The below example is information contained in the **Standard start of program**. It is used to output information regarding the program creation, any user comments, initial safety line, first tool change, and location of first machining operation. The code blocks are written in this example as follows:

```

"( PROGRAM NAME: ",prog_name,")"
"( POST: ",machine_make," ",machine_model,")"
"( DATE: ",output_date," , TIME: ",output_time,")"
user_comment_1
user_comment_2
user_comment_3
user_comment_4
user_comment_5
user_comment_6
user_comment_7 Comments are only displayed if they contain characters.
user_comment_8 and have been selected to output in the BobCAD software.
user_comment_9
user_comment_10
user_comment_11
user_comment_12
user_comment_13
user_comment_14
user_comment_15
comment_start, " ", comment_end
n,inch_mode
n,rapid_move,absolute_coord,"G17","G40","G49",cancel_drill_cycle,"G99"
comment_start,comment_end
n,rapid_move,incremental_coord,"G28","Z0."
system_comment
feature_name_comment
n,t,"M06"
n,s,spindle_on
n,rapid_move,absolute_coord,work_coord,force_x,xr,force_y,yr
n,rapid_move,length_offset
n,coolant_on
comment_start," ",comment_end

```

The corresponding output will be as follows:

```

( PROGRAM NAME: POST TEST TEMPLATE.NC)
( POST: HAAS VF)
( DATE: THU. 01/31/2008 , TIME: 12:25PM)
(INFORMATION IN COMMENT 1 FIELD)
(THROUGH)
(INFORMATION IN COMMENT 15 FIELD)
( )
N10 G20
N20 G00 G90 G17 G40 G49 G80 G99
( )
N30 G91 G28 Z0.
(JOB 1 POCKET)
(TOOL #1 0.5000 ENDMILL ROUGH)
N40 T1 M06

```

```

N50 S10000 M03
N60 G90 G54 X0. Y0.
N70 G43 H1 Z.5
N80 M08
( )

```

The below example is information contained in the **End of program** under **Ending blocks**. It is used to output variables regarding the program end of file, turning off operations and returning to a safe location. The code blocks are often written as follows:

```

comment_start, " ",comment_end
n,coolant_off
n,rapid_move,incremental_coord,"G28","Z0."
n,spindle_off
n,first_tool_with_prefix,"M06"
n,rapid_move,absolute_coord,"X0.","Y0."
n,"M30"

```

The corresponding output will be as follows:

```

( )
N1620 M09
N1630 G00 G91 G28 Z0.
N1640 M05
N1650 T1 M06
N1660 G90 X0. Y0.
N1670 M30

```

The **Program trailer** under **Ending blocks** is similar in function to the **Program header**. In this example it contains only a hard-coded percent sign to define end of the NC file. Note that any end of program commands should be placed in this field for the subprograms to properly terminate, as the information in this field will be the last to appear at the end of the NC file.

The **Sub programs** portion of the tree to the left contains three separate fields: **Sub program call**, **Sub program definition**, and **Sub program return**. Note that these are NOT *subroutines* and should not be confused with them!

Under **Sub program call**:

```

n,sub_call,sub_num,"( SUBPROGRAM CALL )"

```

Under **Sub program definition**:

```

sub_num_with_prefix,sub_comment

```

Under **Sub program return**:

```

n,sub_return,"( SUBPROGRAM RETURN )"

```

The **Misc.** section of the tree contains the following various code block definitions for calling from within the post in other fields:

Under **Optional stop:**

```
n,optional_stop
```

Rotary Axis Index:

```
n,"B",rotary_angle
```

Cutter comp. left:

```
"G41",d_offset
```

Cutter comp. right:

```
"G42",d_offset
```

Cutter comp. off:

```
"G40"
```

Tool length comp.:

```
"G43",h,force_z,zr
```

Rigid tapping start:

```
n,"M29",s,"( RIGID TAP START )"
```

Rigid tapping end:

```
n,"M28","( RIGID TAP END )"
```

Offset register:

(left blank in this example)

Absolute coordinates:

```
absolute_coord
```

Incremental coordinates:

```
incremental_coord
```

Spindle speed low range:

```
"M40","( LOW RANGE SPINDLE )"
```

Spindle speed high range:

```
"M41", "( HIGH RANGE SPINDLE )"
```

Tool list format:

```
"(MTOOL T", list_tool_number, "S1", "D", tool_diameter, "H", tool_length, " )"
```

The **Tool Change** section contains the **Move to next cut change tool** and the **Move to next cut same tool** fields. These define how a tool change is to be executed in the NC file.

The example below is a code block definition contained in the **Move to next cut change tool** field. It is used to output variables regarding turning off operation, moving to the tool change position, executing the change and moving to the next machining operation. You will note that the code blocks are similar to, if not the same as, the first tool change defined in the **Standard start of program**. The code blocks are written as follows:

```
comment_start, " ", comment_end  
n, coolant_off  
n, spindle_off  
n, "G49"  
n, rapid_move, incremental_coord, "G28", "Z0."  
system_comment  
feature_name_comment  
n, t, "M06"  
n, s, spindle_on  
n, rapid_move, absolute_coord, work_coord, force_x, xr, force_y, yr  
n, rapid_move, length_offset  
n, coolant_on  
comment_start, " ", comment_end
```

The corresponding output will be as follows:

```
( )  
N790 M09  
N800 M05  
N810 G49  
N820 G91 G28 Z0.  
(JOB 1 POCKET)  
(TOOL #2 0.6250 ENDMILL FINISH)  
N830 T2 M06  
N840 S403 M03  
N850 G90 G54 X0. Y.4375  
N860 G43 H2 Z.5  
N870 M08  
( )
```

The **Move to next cut same tool** field is used mainly to set the spindle speed if different from previous operation, comment the operation, and move to the next cut. The code blocks are often written as follows:

```
comment_start," ",comment_end
system_comment
feature_name_comment
n,s
n,rapid_move,absolute_coord,work_coord,force_x,xr,force_y,yr
comment_start," ",comment_end
```

The corresponding output will be as follows:

```
( )
(JOB 2 CONTOUR)
(TOOL #1 0.5000 ENDMILL ROUGH)
N640 S374
N650 G90 G54 X0. Y2.25
( )
```

The **Rapid moves** section contains all code block definitions for rapid moves within the NC file. It contains the following fields:

First rapid move z:

```
n,rapid_move,zr
```

Position move z:

```
n,rapid_move,zr
```

Rapid move xy:

```
n,rapid_move,xr,yr
```

Rapid move z:

```
n,rapid_move,zr
```

Rapid move xyz:

```
n,rapid_move,xr,yr,zr
```

The **Feed moves** section contains all code block definitions for rapid moves within the NC file. It contains the following fields:

Feed move xy leadin:

```
n,cc,feed_move,x_f,y_f,feed_rate
```

Feed move xy:

```
n,feed_move,x_f,y_f,feed_rate
```

Feed move xy leadout:

`n,cc,feed_move,x_f,y_f,feed_rate`

Feed move z:

`n,feed_move,z_f,feed_rate`

Feed move xyz:

`n,feed_move,x_f,y_f,z_f,feed_rate`

Arc move:

`n,g_arc_move,x_f,y_f,arc_center,feed_rate`

The **Operations** section defines the code block for multiple operations within a single BobCAD (.bbcd) drawing file and the resultant generated g-code NC file. It contains the following fields:

Start of operation

`"(START OF OPERATION)"`

End of operation

`"(END OF OPERATION)"`

Canned Cycles

IMPORTANT: These code blocks are independent for each controller. As a result, they are the most difficult to properly program. However, they will greatly reduce the length of the g-code NC program file and increase the machine tool efficiency. Care must be taken when verifying the generated output to ensure correct functionality. The examples given here should serve as a good starting point for these, but remember to double-check *everything*.

The **Canned cycle** section contains all the fields for defining the code blocks for each cycle, the point format for each cycle (with subprogram and without) and the termination of the cycles. It contains the following fields:

Standard drill:

`n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,canned_feed_rate`

Standard drill with dwell:

`n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,dwell,canned_feed_rate`

Peck drill:

`n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,peck_drill_increment, dwell, canned_feed_rate`

High speed peck drill:

n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,peck_drill_increment, dwell,
canned_feed_rate

Tap:

rigid_tapping_start
n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,dwell,canned_feed_rate

Left handed tap:

rigid_tapping_start
n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,dwell,canned_feed_rate

Fine boring:

n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,dwell,canned_feed_rate

Boring 1:

n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,canned_feed_rate

Boring 2:

n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,canned_feed_rate

Boring 1 with dwell:

n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,dwell,canned_feed_rate

Boring 2 with dwell:

n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,dwell,canned_feed_rate

Back boring:

n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,dwell,canned_feed_rate

Canned cycle cancel:

rigid_tapping_end
n,cancel_drill_cycle

Canned cycle point format:

n,x_f,y_f,z_f,canned_feed_rate

Point format Standard drill:

n,x_f,y_f,z_f,canned_feed_rate

Point format Standard drill with dwell:

`n,x_f,y_f,z_f,canned_feed_rate`

Point format Peck drill:

`n,x_f,y_f,z_f,canned_feed_rate`

Point format High speed peck drill:

`n,x_f,y_f,z_f,canned_feed_rate`

Point format Tap:

`n,x_f,y_f,z_f,canned_feed_rate`

Point format Left handed tap:

`n,x_f,y_f,z_f,canned_feed_rate`

Point format Fine boring:

`n,x_f,y_f,z_f,canned_feed_rate`

Point format Boring 1:

`n,x_f,y_f,z_f,canned_feed_rate`

Point format Boring 2:

`n,x_f,y_f,z_f,canned_feed_rate`

Point format Boring 1 with dwell:

`n,x_f,y_f,z_f,canned_feed_rate`

Point format Boring 2 with dwell:

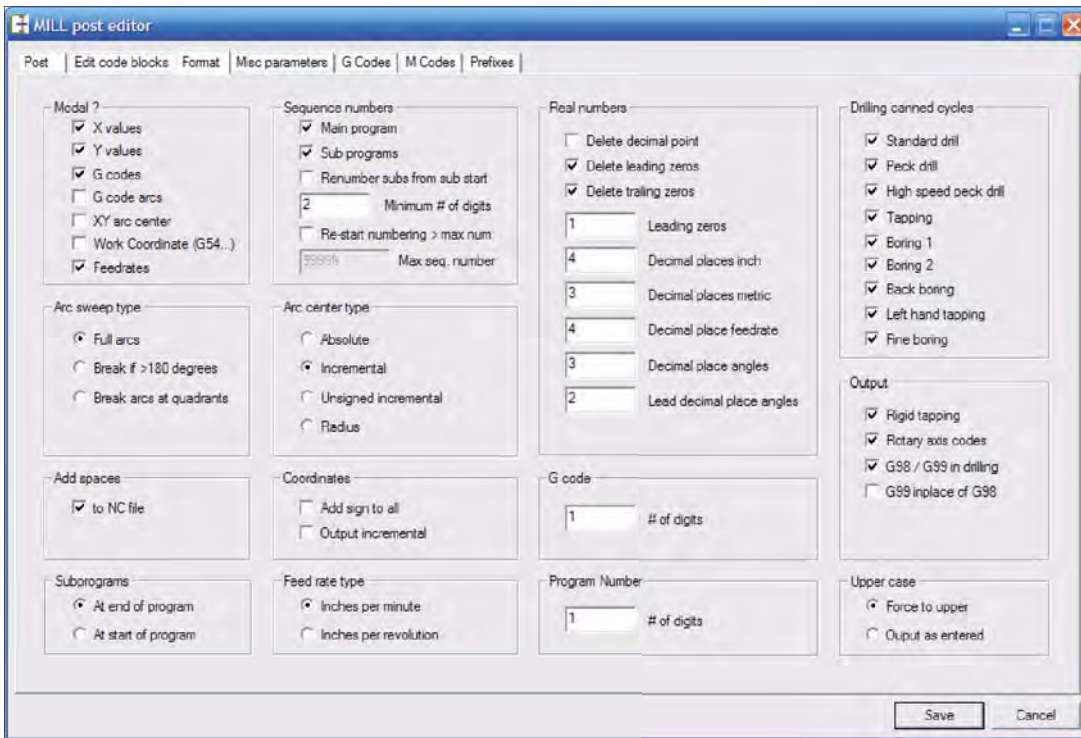
`n,x_f,y_f,z_f,canned_feed_rate`

Point format Back boring:

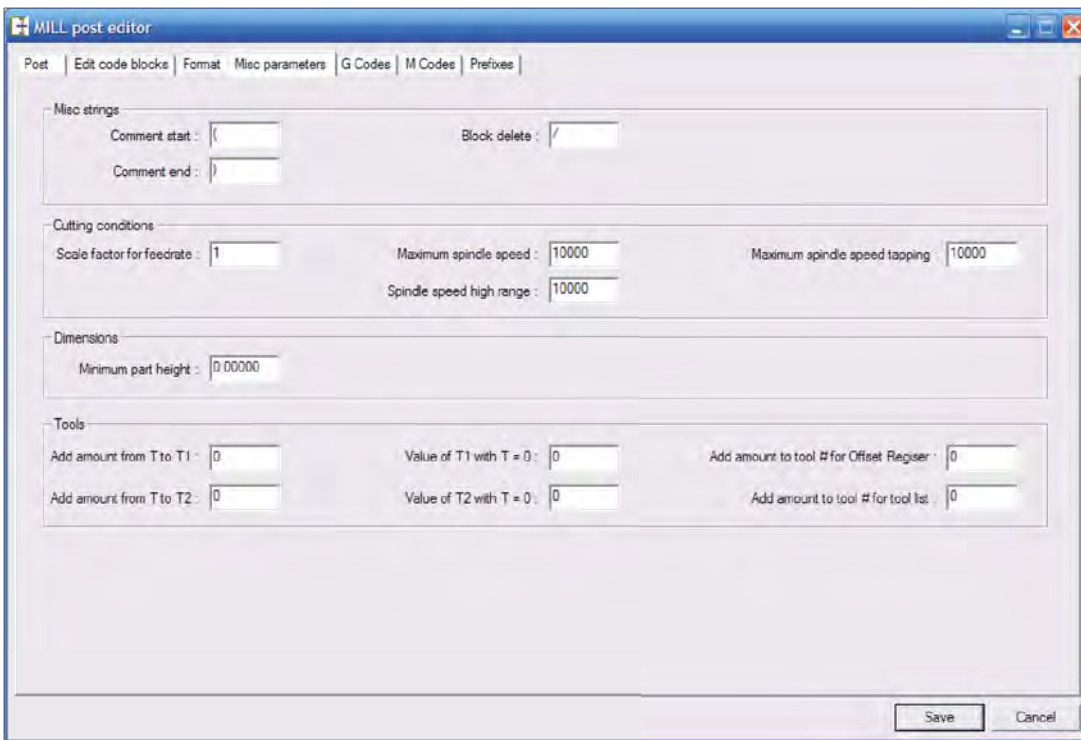
`n,x_f,y_f,z_f,canned_feed_rate`

The following tabs in the post editor contain small fields and check boxes for editing and tuning the NC program file to the required output. They are mostly self explanatory so there should not be a need to show the example output for each field change. The best method for understanding the functionality is to edit a single field, save the post configuration file, and post NC code to view results.

The **Format** tab contains fields to define or select which code is modal output (or not), how sequence numbers output, leading and trailing zeroes coordinates, drilling canned cycles for controller, arc center type, NC code spacing, and the like.



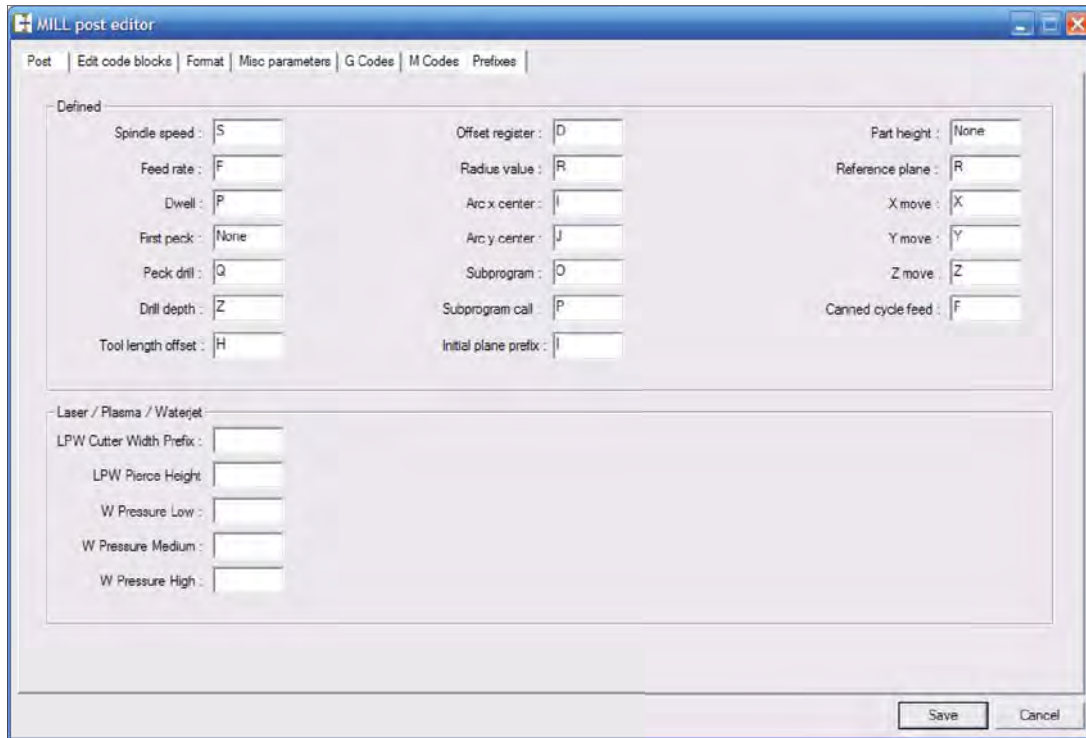
The **Misc. parameters** tab contains fields to define or select the strings for comments and block deletes, cutting conditions to set maximum spindle speeds and gear ranges, etc.



The **G Codes** tab contains fields to enter the format of the defined variables to be output in the NC file and the work offsets number, g code definition, and number of offsets.

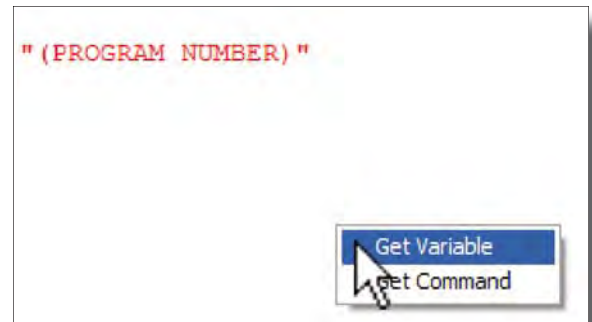
The **M Codes** tab contains fields to enter the format of the defined variables to be output in the NC file.

The **Prefixes** tab contains fields to enter the format of the defined variables to be output in the NC file.

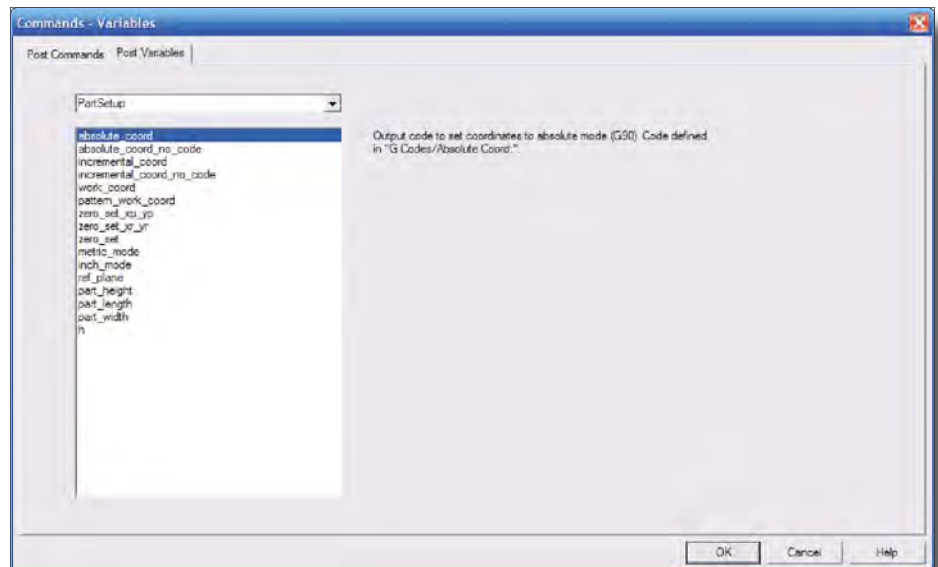


Note on Editing the Code blocks field – Getting Variables/Commands

All commands and variables can be pulled up in a list and selected for insertion without typing when editing a field in the Edit code blocks tab. Simply place the text cursor at the location to enter the desired item, then right click the mouse and select the either **Get Variable** or **Get Command**.



A new window called, **Commands – Variables** will display. Selections may be changed by clicking on the corresponding **Post Commands** or **Post Variables** tab. Select the desired folder, by click on the on the pull-down menu and double click on the variable name. The variable may also be chosen by selecting it and clicking **OK** at the bottom of the dialog. With either method, the highlighted variable will be placed in the location we determine in the **Edit code blocks** tab.



Note on Editing the Code blocks field – Outputting debug comments

As mentioned earlier, on the **Post** tab in the lower left hand corner is the Output debug comments check box. If this box is checked and the post is saved, the code will post with comments. These comments will indicate where in the **Edit code blocks** field the resulting output sections were derived.

The example below this a program using **Output debug comments** (the comments themselves are highlighted in yellow).

```
***** 28 - Rapid moves//Position move Z *****
(BEGIN PREDATOR NC HEADER)
(MTOOL T1 S1 D.5 H5.)
(SBOX X-2.25 Y-2.25 Z-1. L4.5 W4.75 H1.)
(END PREDATOR NC HEADER)
( )
% O100 ( PROGRAM NUMBER )
***** 2 - Starting blocks//Standard start of program *****
( PROGRAM NAME: POST TEST TEMPLATE.NC)
( POST: HAAS VF)
( DATE: FRI. 02/01/2008 , TIME: 11:12AM)
( )
N10 G20
N20 G00 G90 G17 G40 G49 G80 G99
( )N30 G91 G28 Z0.
(JOB 2 CONTOUR)
(TOOL #1 0.5000 ENDMILL ROUGH)
N40 T1 M06
N50 S385 M03
N60 G90 G54 X0. Y2.25
N70 G43 H1 Z.25
N80 M08
( )
***** 40 - Operations//Start of operation *****
( START OF OPERATION )
***** 50 - Rapid moves//Rapid move Z *****
N90 Z.1
***** 51 - Feed moves//Feed move Z *****
N100 G01 Z-1. F7.
***** 56 - Feed moves//Feed move xy leadin *****
N110 G41 D1 X-.25 F15.
***** 64 - Feed moves//Arc move *****
N120 G03 X0. Y2. I.25 J0.
***** 53 - Feed moves//Feed move XY *****
N130 G01 X2.
***** 53 - Feed moves//Feed move XY *****
N140 Y-2.
***** 53 - Feed moves//Feed move XY *****
N150 X-2.
***** 53 - Feed moves//Feed move XY *****
N160 Y2.
***** 53 - Feed moves//Feed move XY *****
N170 X0.
***** 53 - Feed moves//Feed move XY *****
N180 X.1
***** 64 - Feed moves//Arc move *****
N190 G03 X.35 Y2.25 I0. J.25
***** 57 - Feed moves//Feed move xy leadout *****
N200 G40 G01 X.1
```

```

***** 50 = Rapid moves//Rapid move Z *****
N210 G00 Z.25
***** 5 = Ending blockes//End of program *****
( )
N220 M09
N230 G91 G28 Z0.
N240 M05
N250 T1 M06
N260 G90 X0. Y0.
N270 M30
***** 24 = Ending blocks//Program trailer *****
%
```

Note that each debug comment has a number prefix. This number corresponds to the function number in the post processor file. Remember to turn off **Output debug comments**, save the changes and re-post code prior to saving or sending the NC file to the machine. The example below this the same program with **Output debug comments** turned off for comparison:

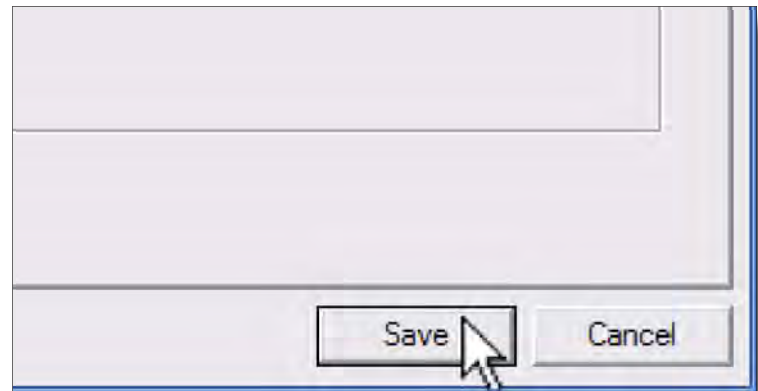
```

(BEGIN PREDATOR NC HEADER)
(MTOOL T1 S1 D.5 H5.)
(SBOX X-2.25 Y-2.25 Z-1. L4.5 W4.75 H1.)
(END PREDATOR NC HEADER)
( )
%
O100 ( PROGRAM NUMBER )
( PROGRAM NAME: POST TEST TEMPLATE.NC)
( POST: HAAS VF)
( DATE: FRI. 02/01/2008 , TIME: 11:21AM)
( )
N10 G20
N20 G00 G90 G17 G40 G49 G80 G99
( )
N30 G91 G28 Z0.
(JOB 2 CONTOUR)
(TOOL #1 0.5000 ENDMILL ROUGH)
N40 T1 M06
N50 S385 M03
N60 G90 G54 X0. Y2.25
N70 G43 H1 Z.25
N80 M08
( )
( START OF OPERATION )
N90 Z.1
N100 G01 Z-1. F7.
N110 G41 D1 X-.25 F15.
N120 G03 X0. Y2. I.25 J0.
N130 G01 X2.
N140 Y-2.
N150 X-2.
N160 Y2.
N170 X0.
N180 X.1
N190 G03 X.35 Y2.25 I0. J.25
N200 G40 G01 X.1
N210 G00 Z.25
( )
N220 M09
N230 G91 G28 Z0.
N240 M05
```

```
N250 T1 M06
N260 G90 X0. Y0.
N270 M30
%
```

Step 3: Save the post processor file.

The simplest step in the process - click the **Save** button at the bottom of the editor utility and it will save the changes to the selected file.



Ending Note

The post processor files are actually ASCII text files, and as such they may be opened with a text editor. Notepad, NotePad++, or the Predator CNC Editor may be used to manually edit the post processor file without going through the post editor utility.

IMPORTANT: Extreme care should be taken when editing with a text editor application, since it is not possible to verify the variable spelling or post format, and the text color (syntax highlighting) will not be indicated. The file below is the original Haas VF.MillPst file as it is displayed in Notepad, the default Windows text editor. For reference, there is a post variables text file in each machine type's subfolder under Posts\.

```
****Version number MONTH DAY YEAR****
1000. Version Information = Version Month Day Year? "2.0 01 14 2008"
0. File header
"(BEGIN PREDATOR NC HEADER)"
output_tool_list
"(SBOX X",stock_min_x," Y",stock_min_y," Z",stock_min_z," L",stock_max_x," W",stock_max_y,
" H",stock_max_z,")"
"(END PREDATOR NC HEADER)"
comment_start," ",comment_end
"%"
"0",prog_n,"( PROGRAM NUMBER )"
default_add_spaces
1. Start of file programmed zero
"( PROGRAM NAME: ",prog_name,")"
"( POST: ",machine_make," ",machine_model,")"
"( DATE: ",output_date," , TIME: ",output_time,")"
user_comment_1
user_comment_2
user_comment_3
user_comment_4
user_comment_5
user_comment_6
user_comment_7
user_comment_8
user_comment_9
user_comment_10
user_comment_11
```

```
user_comment_12
user_comment_13
user_comment_14
user_comment_15
comment_start," ",comment_end
n,inch_mode
n,rapid_move,absolute_coord,"G17",cancel_offset,"G49",cancel_drill_cycle,"G99"
comment_start,comment_end
n,rapid_move,incremental_coord,"G28","Z0."
system_comment
feature_name_comment
n,t,"M06"
n,s,spindle_on
n,rapid_move,absolute_coord,work_coord,force_x,xr,force_y,yr
n,rapid_move,length_offset
n,coolant_on
comment_start," ",comment_end
2. Start of file Standard
"( PROGRAM NAME: ",prog_name,")"
"( POST: ",machine_make," ",machine_model,")"

"( DATE: ",output_date,", TIME: ",output_time,")"
user_comment_1
user_comment_2
user_comment_3
user_comment_4
user_comment_5
user_comment_6
user_comment_7
user_comment_8
user_comment_9
user_comment_10
user_comment_11
user_comment_12
user_comment_13
user_comment_14
user_comment_15
comment_start," ",comment_end
n,inch_mode
n,rapid_move,absolute_coord,"G17",cancel_offset,"G49",cancel_drill_cycle,"G99"
comment_start," ",comment_end
n,rapid_move,incremental_coord,"G28","Z0."
system_comment
feature_name_comment
n,t,"M06"
n,s,spindle_on
n,rapid_move,absolute_coord,work_coord,force_x,xr,force_y,yr
n,rapid_move,length_offset
n,coolant_on
comment_start," ",comment_end
3. Tool change
comment_start," ",comment_end
n,coolant_off
n,spindle_off
n,"G49"
n,rapid_move,incremental_coord,"G28","Z0."
system_comment
feature_name_comment
n,t,"M06"
n,s,spindle_on
```

```

n,rapid_move,absolute_coord,work_coord,force_x,xr,force_y,yr
n,rapid_move,length_offset
n,coolant_on
comment_start," ",comment_end
4. Null tool change
comment_start," ",comment_end
system_comment
feature_name_comment
n,s
n,rapid_move,absolute_coord,work_coord,force_x,xr,force_y,yr
comment_start," ",comment_end
5. End of file for non-zero tool
comment_start," ",comment_end
n,coolant_off
n,rapid_move,incremental_coord,"G28","Z0."
n,spindle_off
n,first_tool_with_prefix,"M06"
n,rapid_move,absolute_coord,"X0.,""Y0."
n,"M30"
6. Optional Stop
n,optional_stop
7. Sub program call
n,sub_call,sub_num,"( SUBPROGRAM CALL )"
8. Sub definition
" "
sub_num_with_prefix,sub_comment
9. Sub program return

n,sub_return,"( SUBPROGRAM RETURN )"
10. Rotary axis index string
n,"B",rotary_angle
11. Cancel cutter compensation
"G40"
12. Cutter compensation left
"G41",d_offset
13. Cutter compensation right
"G42",d_offset
14. Tool length compensation
" G43",h,force_z,zr
20. Spindle speed low range
"M40","( LOW RANGE SPINDLE )"
21. Spindle speed high range
"M41","( HIGH RANGE SPINDLE )"
22. Rigid tapping start.
n,"M29",s,"( RIGID TAP START )"
23. Rigid tapping end.
n,"M28","( RIGID TAP END )"
24. File trailer.
"%"
25. Format for offset when using offset registers.
26. Set debug.
debug_off
27. First Rapid Move.
n,rapid_move,zr
28. Rapid Move.
n,rapid_move,zr
40. Start of 2axis contour.
"( START OF OPERATION )"
50. Line first rapid move Z.

```

n,rapid_move,zr
51. Line feed move Z.
n,feed_move,z_f,feed_rate
52. Line rapid move XY.
n,rapid_move,xr,yr
53. Line feed move XY.
n,feed_move,x_f,y_f,feed_rate
54. Line rapid move XYZ.
n,rapid_move,xr,yr,zr
55. Line feed move XYZ.
n,feed_move,x_f,y_f,z_f,feed_rate
56. Line feed move XY on Leadin.
n,cc,feed_move,x_f,y_f,feed_rate
57. Line feed move XY on Leadout.
n,cc,feed_move,x_f,y_f,feed_rate
64. Arc move.
n,g_arc_move,x_f,y_f,arc_center,feed_rate

71. End of 2axis cutting.
“(END OF OPERATION)”
73. High speed peck drill canned cycle.
n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,peck_drill_increment,canned_feed_rate
74. Left handed tapping canned cycle.
rigid_tapping_start
n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,canned_feed_rate
76. Fine boring canned cycle.
n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,dwell,canned_feed_rate
80. Drill canned cycle cancel.
rigid_tapping_end
n,cancel_drill_cycle
81. Standard drill canned cycle.
n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,canned_feed_rate
82. Standard drill canned cycle with dwell.
n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,dwell,canned_feed_rate
83. Peck drill canned cycle.
n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,peck_drill_increment,dwell,canned_feed_rate
84. Tapping canned cycle.
rigid_tapping_start
n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,dwell,canned_feed_rate
85. Boring cycle #1 canned cycle.
n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,canned_feed_rate
86. Boring cycle #2 canned cycle.
n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,canned_feed_rate
87. Back boring cycle canned cycle.
n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,dwell,canned_feed_rate
88. Boring cycle #1 canned cycle with dwell.
n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,dwell,canned_feed_rate
89. Boring cycle #2 canned cycle with dwell.
n,g_canned_cycle,g98_g99,x_f,y_f,drill_depth,reference_plane,dwell,canned_feed_rate
90. Canned cycle drill point format WITH SUBPROGRAMS.
n,x_f,y_f,z_f,canned_feed_rate
113. Canned cycle drill point format for high speed peck drill canned cycle NO SUBPROGRAMS.
n,x_f,y_f,z_f,canned_feed_rate
114. Canned cycle drill point format for left handed tapping canned cycle NO SUBPROGRAMS.
n,x_f,y_f,z_f,canned_feed_rate
116. Canned cycle drill point format for fine boring canned cycle NO SUBPROGRAMS.
n,x_f,y_f,z_f,canned_feed_rate

91. Canned cycle drill point format for standard drill canned cycle NO SUBPROGRAMS.
n,x_f,y_f,z_f,canned_feed_rate

92. Canned cycle drill point format for standard drill canned cycle with dwell NO SUBPROGRAMS.
n,x_f,y_f,z_f,canned_feed_rate

93. Canned cycle drill point format for peck drill canned cycle NO SUBPROGRAMS.
n,x_f,y_f,z_f,canned_feed_rate

94. Canned cycle drill point format for tapping canned cycle NO SUBPROGRAMS.
n,x_f,y_f,z_f,canned_feed_rate

95. Canned cycle drill point format for boring cycle #1 canned cycle NO SUBPROGRAMS.
n,x_f,y_f,z_f,canned_feed_rate

96. Canned cycle drill point format for boring cycle #2 canned cycle NO SUBPROGRAMS.
n,x_f,y_f,z_f,canned_feed_rate

97. Canned cycle drill point format for back boring cycle canned cycle NO SUBPROGRAMS.
n,x_f,y_f,z_f,canned_feed_rate

98. Canned cycle drill point format for boring cycle #1 canned cycle with dwell NO SUBPROGRAMS.
n,x_f,y_f,z_f,canned_feed_rate

99. Canned cycle drill point format for boring cycle #2 canned cycle with dwell NO SUBPROGRAMS.
n,x_f,y_f,z_f,canned_feed_rate

100. Absolute coordinate output.
absolute_coord

101. Incremental coordinate output.
incremental_coord

110. Tool list format.
"(MTOOL T",list_tool_number,"S1 D",tool_diameter," H",tool_length,")"

200. Is X modal? y

201. Is Y modal? y

202. Is Z modal? y

203. Are the g codes modal? y

204. Are the g codes (G02 and G03) modal in arc milling? n

205. Are the xy (or yz or xz) coordinates modal in arc milling? n

206. Are work coordinates modal ? n

207. Output sequence numbers in subprograms ? y

208. Output sequence numbers? y

209. Number of places for sequence numbers? 2

210. Delete the decimal point? n

211. Delete leading zeros? y

212. Delete trailing zeros? y

213. English or Metric format (E/M)? E

214. Places before decimal point for reals (X, Y, Z)? 1

215. Number of places for G and M codes? 1

216. Places after decimal for feedrate ? 4

217. Scale factor for feedrate ? 1

219. Add spaces to the program? y

221. Break arcs into quadrants? n

222. Arc center a=absolute, b=incremental, d=unsigned inc., e=radius? b

223. Break arcs into two pieces if greater than 180 degrees? n

227. Output G40 after, rather than with, the last linear or arc move? y

230. Use Standard drilling canned cycle? y

231. Use peck drill canned cycle? y

232. Use High speed peck drill canned cycle? y

233. Use tapping canned cycle? y

234. Use boring cycle #1 canned cycle? y

235. Use boring cycle #2 canned cycle? y

236. Use back boring cycle canned cycle? y

237. Use left hand tap cycle canned cycle? y

238. Use fine boring cycle canned cycle? y

240. Amount to add to t to obtain t1? 0
241. Amount to add to t to obtain t2? 0
242. Value of t1 at t = 0? 0
243. Value of t2 at t = 0? 0
258. Maximum number of work offsets? 26
262. Add sign to all coordinate values? n
267. Amount to add to tool # for tool register value? 0
272. Rigid tapping? y
273. Output programmable rotary axis codes? y
313. Z clearance for auto Z programming for XY move? 0.20000
314. Z clearance for auto Z programming for cutting? 0.00000
315. Minimum part height? 0.00000
414. Number of decimal places for metric numbers ? 3
415. Number of decimal places for english numbers? 4
425. Number of decimal for angles? 3
426. Number of leading decimal places for angles? 2
427. Tapping feed rate (1=ipm 2=ipr)? 2

428. Feed rates other than tapping (1=ipm 2=ipr)? 1
429. Maximum spindle speed for tapping? 10000
430. Maximum spindle speed? 10000
431. Spindle speed for high range? 10000
432. Add amount for tool number in tool list? 0
433. Maximum sequence number allowed (used when #534 is y)? 99999
511. Use incremental coordinates? n
512. Use block delete for stop codes (M00)? n
513. Output F feedrate values? n
515. Output G99 instead of G98 in drilling? n
516. Output G98/G99 in drilling? y
518. Output M00 codes? n
526. Start renumbering from start number for subprograms? n
527. Create subdirectory for nc file? n
531. Replace spaces in comment with commas ? n
533. Output subprograms at the beginning of the program? n
534. Re-Start sequence numbering once max sequence number is reached? n
535. Are Feed Rates modal? y
536. Force all codes to upper case ? y
605. Spindle speed prefix? "S"
606. Feedrate prefix? "F"
607. Dwell prefix? "P"
610. Miscellaneous end of file string? ""
613. Pattern contour subprogram start code? ""
614. Inch mode machining? "G20"
615. Metric mode machining? "G21"
620. Absolute coordinates? "G90"
621. Incremental coordinates? "G91"
622. Coordinate zero set? "G92"
625. End of file? "M02"
626. Stop? "M00"
627. Optional Stop? "M01"
628. Subprogram call? "M98"
629. Subprogram return? "M99"
630. Comment Start? "("
631. Comment End? ")"
639. Cancel wire offset? "G40"
641. Prefix for radius values? "R"
642. Prefix for arc X center? "I"
643. Prefix for arc Y center? "J"
645. Subprogram prefix? "O"
646. Machine maker? "HAAS"

647. Machine model? "VF"
648. Part Height prefix? "None"
649. Reference Plane prefix? "R"
650. Wire comp left? "G41"
651. Wire comp right? "G42"
656. Block delete? "/"
658. Subprogram call subnumber prefix? "P"
659. Add these characters to the end of each line? ""
670. Spindle forward string? "M03"
671. Spindle reverse string? "M04"
672. Spindle off string? "M05"
673. Coolant on string? "M08"
674. Coolant off String? "M09"
675. First peck prefix? "None"
676. Peck drill prefix? "Q"
677. Drill depth prefix? "Z"
678. Cutter offset prefix? "D"
679. Cancel drilling canned cycle? "G80"
680. GCode for Rapid Move? "G00"
681. GCode for Feed Move? "G01"
682. GCode for Arc CW? "G02"
683. GCode for Arc CCW? "G03"
684. Prefix for X Move? "X"
685. Prefix for Y Move? "Y"
686. Prefix for Z Move? "Z"
687. Prefix for drill canned cycle feed rate? "F"
688. Prefix for tool length offset? "H"
689. Prefix for initial plane? "I"

800. Standard drilling cycle no dwell #1? "G81"
801. Peck drill cycle no dwell #2? "G83"
802. High speed peck drill cycle no dwell #3? "G73"
803. Tapping cycle no dwell #4? "G84"
804. Boring cycle #1 no dwell #5? "G85"
805. Boring cycle #2 no dwell #6? "G86"
806. Back boring cycle no dwell #7? "G87"
807. Left hand tapping cycle no dwell #8? "G74"
808. Fine boring cycle no dwell #9? "G76"
809. Hole making cycle no dwell #10?""
810. Hole making cycle no dwell #11?""
811. Hole making cycle no dwell #12?""
820. Standard drilling cycle with dwell #1? "G82"
821. Peck drill cycle with dwell #2? "G83"
822. High speed peck drill cycle with dwell #3? "G73"
823. Tapping cycle with dwell #4? "G84"
824. Boring cycle #1 with dwell #5? "G88"
825. Boring cycle #2 with dwell #6? "G89"
826. Back boring cycle with dwell #7? "G87"
827. Left hand tapping cycle with dwell #8? "G74"
828. Fine boring cycle with dwell #9? "G76"
829. Hole making cycle with dwell #10?""
830. Hole making cycle with dwell #11?""
831. Hole making cycle with dwell #12?""
900. Work shift #1? "G54"
901. Work shift #2? "G55"
902. Work shift #3? "G56"
903. Work shift #4? "G57"
904. Work shift #5? "G58"
905. Work shift #6? "G59"
906. Work shift #7? "G110"

907. Work shift #8? "G111"
908. Work shift #9? "G112"
909. Work shift #10? "G113"
910. Work shift #11? "G114"
911. Work shift #12? "G115"
912. Work shift #13? "G116"
913. Work shift #14? "G117"
914. Work shift #15? "G118"
915. Work shift #16? "G119"
916. Work shift #17? "G120"
917. Work shift #18? "G121"
918. Work shift #19? "G122"
919. Work shift #20? "G123"
920. Work shift #21? "G124"
921. Work shift #22? "G125"
922. Work shift #23? "G126"
923. Work shift #24? "G127"
924. Work shift #25? "G128"
925. Work shift #26? "G129"
926. Work shift #27? ""
927. Work shift #28? ""
928. Work shift #29? ""
929. Work shift #30? ""
930. Work shift #31? ""
931. Work shift #32? ""
932. Work shift #33? ""
... to ...
993. Work shift #93? ""
994. Work shift #94? ""
995. Work shift #95? ""
996. Work shift #96? ""
997. Work shift #97? ""
998. Work shift #98? ""
999. Work shift #99? ""