

# Command Reference for the ShopBot Control Software

## Introduction

All operations are accomplished in ShopBot with simple 2-letter commands (e.g. “MX” for “move in the X axis”). These commands may be followed by one or more parameters that define the operation (e.g. how far to move). This document is a full listing and explanation of ShopBot Commands and their related parameters. Each command description shows the 2-letter command and full command name at the top, followed by a list of parameters for the command as short abbreviations. This list shows the order that the parameters, separated by commas, must be entered in the Parameter Line or in a part file. After this short listing, a full description of the command and its parameters follows.

When using the 2-letter commands in the ShopBot Control Software, you have the option of entering the parameters for the command in the Parameter Line in the Console Screen or of using helpful Fill-In sheets that provide explanatory information to enter the parameters. [Sample Parameter Line](#); [Sample Fill-In Sheet](#).

The software is configured so that simple commands such as [MX; Move X Axis] automatically expect just a value or two in the Parameter Line and that complex commands such as [CC; Cut Circle] will automatically bring up a Fill-In sheet for you to complete. To turn off display of Fill-In sheet, click on the display box on the bottom of the Fill-In sheet to remove the check mark. To use the Fill-In sheet if it is not displayed, click on the icon for the Fill-In sheet located to the right of the command line in the ShopBot Control Software.

## For Fast Access to a Command

[\[CA\]](#) [\[CC\]](#) [\[CG\]](#) [\[CP\]](#) [\[CR\]](#)  
[\[FC\]](#) [\[FE\]](#) [\[FG\]](#) [\[FP\]](#) [\[FS\]](#)  
[\[HA\]](#) [\[HB\]](#) [\[HC\]](#) [\[HE\]](#) [\[HN\]](#) [\[HQ\]](#) [\[HR\]](#) [\[HT\]](#) [\[HU\]](#) [\[HW\]](#)  
[\[J2\]](#) [\[J3\]](#) [\[J4\]](#) [\[J5\]](#) [\[JA\]](#) [\[JB\]](#) [\[JH\]](#) [\[JS\]](#) [\[JX\]](#) [\[JY\]](#) [\[JZ\]](#)  
[\[M2\]](#) [\[M3\]](#) [\[M4\]](#) [\[M5\]](#) [\[MA\]](#) [\[MB\]](#) [\[MD\]](#) [\[MH\]](#) [\[MO\]](#) [\[MS\]](#) [\[MX\]](#) [\[MY\]](#) [\[MZ\]](#)  
[\[RA\]](#) [\[RI\]](#) [\[RP\]](#) [\[RR\]](#) [\[RS\]](#) [\[RZ\]](#)  
[\[SA\]](#) [\[SF\]](#) [\[SI\]](#) [\[SK\]](#) [\[SL\]](#) [\[SM\]](#) [\[SO\]](#) [\[SP\]](#) [\[SR\]](#) [\[ST\]](#) [\[SW\]](#)  
[\[TC\]](#) [\[TD\]](#) [\[TF\]](#) [\[TH\]](#) [\[TS\]](#) [\[TT\]](#) [\[TU\]](#)  
[\[UD\]](#) [\[UL\]](#) [\[UN\]](#) [\[UR\]](#) [\[UU\]](#) [\[UV\]](#) [\[UZ\]](#)  
[\[VA\]](#) [\[VB\]](#) [\[VC\]](#) [\[VD\]](#) [\[VI\]](#) [\[VL\]](#) [\[VN\]](#) [\[VP\]](#) [\[VR\]](#) [\[VS\]](#) [\[VU\]](#)  
[\[Z2\]](#) [\[Z3\]](#) [\[Z4\]](#) [\[Z5\]](#) [\[ZA\]](#) [\[ZB\]](#) [\[ZT\]](#) [\[ZX\]](#) [\[ZY\]](#) [\[ZZ\]](#)

## Access by Menu Listing Selection

[Circle](#) [File](#) [Help](#) [Jog](#) [Move](#) [Record](#) [Set](#)  
[Utilities](#) [Values](#) [Virtual Tools](#) [Zero](#)

## Table of Contents

Introduction .....	1
For Fast Access to a Command .....	1
Access by Menu Listing Selection .....	1
Table of Contents .....	2
Commands .....	4
CA - Cut Arch .....	4
CC - Cut Circle .....	5
CG - Cut G-code Circle .....	6
CP - Cut Circle from Center Point .....	8
CR - Cut Rectangle .....	9
FC - File Conversion .....	10
FE - File Edit.....	10
FG - File Load in Goto/Single-Step Mode .....	11
FP - File Load PART File .....	12
FS - File, Set the PARTS File Directory .....	13
Fill-In Sheet Sample .....	14
HA - Help About.....	15
HB - Help: ShopBot website Technical Bulletins .....	15
HC - Help: Command Reference.....	15
HE - Help: Quick Reference Page (editing) .....	15
HN - Help: Send a request for help to ShopBot Technical Support .....	15
HQ - Help: Quick Reference Page .....	15
HR - Help: Quick Reference Page.....	15
HT - Help: Troubleshooting & Maintenance .....	15
HU - Help: ShopBot User Guide .....	15
HW - Help ShopBot Website .....	15
J2 - Jog 2 Dimensions .....	15
J3 - Jog 3 Dimensions .....	16
J4 - Jog 4 Dimensions .....	16
J5 - Jog 5 Dimensions .....	16
JA - Jog A axis .....	16
JB- Jog B axis .....	16
JS - Jog Speed.....	16
JX - Jog X-axis .....	17
JY - Jog - Y axis .....	17
JZ - Jog Z-axis.....	17
JH - Jog Home .....	17
M2 - Move 2 Dimensions.....	17
M3 - Move 3 Dimensions.....	17
M4 - Move 4 Dimensions.....	18
M5 - Move 5 Dimensions.....	18
MA - Move A axis .....	18
MB - Move B axis .....	18
MD - Move Distance Move by specifying Distance and Angle.....	18
MH - Move Home .....	19
MO - Motors Off.....	19
MS - Move Speed Set .....	19
MX - Move X axis .....	19
MY - Move Y axis .....	20
MZ - Move Z axis.....	20
Parameter Line Sample.....	20
Ramping - Detailed Explanation .....	21
RA - Record Activate .....	23
RI - Record Inactivate.....	23
RP - Record Play all Commands in Storage.....	23
RR - Record Replay of the Last Commands .....	24
RS - Record Save Stored Commands to a File and Optionally Edit.....	24
RZ - Record Zero the Memory.....	24
SA - Set to Absolute Distances .....	24
SF - Sets File and Move Limit Checking .....	25

SI - Send Command Line(s) .....	25
SK - Set to Keyboard Arrows to Move Tool .....	25
SL - Set to Clear all Variables in Memory .....	26
SM - Set to Move/Cut Mode .....	26
SO - Set Output Switch .....	26
SP - Set to Preview Mode .....	26
SR - Set to Relative Distances .....	27
ST - Set Location to Table Base Coordinates .....	27
SW - Set Warning Duration .....	28
Virtual Tools .....	28
TC - Tools Copy Machine .....	28
TD - Tools Drill Press .....	28
TF - Tools Forney Fluter .....	28
TH - Tools Header Writer .....	29
TI - Tools Indexer .....	29
TS - Tools ShopBot Setup .....	29
TT - Tools Typesetter .....	29
TU - Tools Tabletop Surfacers .....	29
UD - Utilities Diagnostic Display .....	29
UI - Utilities Install Control Box Firmware .....	30
UL - Utilities List Variables .....	30
UN - Utilities Name of Editor Program .....	30
UR - Reset software to Default Settings, Load a Custom Settings File, or Clear the System Log .....	30
US - Save Current Settings to a Default Settings File .....	30
UU - Utilities Calculator .....	31
UV - Utilities Values and Settings Display .....	31
UZ - Utilities Zero all Current Locations and Table Base Coordinates .....	31
VA - Values for Axis Locations .....	31
VB - Values for Tabbings Feature .....	32
VC - Value for Cutter-related Parameters .....	32
VD - Values for Display Settings .....	33
VH - Values for Z-axis Height Controller .....	34
VI - Values for Communications Port and Other Addresses .....	35
VL - Values for Limits for Table .....	35
VN - Value Input Switch Assignment .....	36
VP - Values for Preview Screen .....	37
VR - Values for Ramps .....	37
VS - Values for Speeds .....	38
VU - Values for Calibration Units .....	39
Z Commands Note .....	40
Z2 - Zero 2D (X & Y) axes .....	40
Z3 - Zero 3D (X, Y & Z) axes .....	40
Z4 - Zero 4D (X, Y, Z, & A) axes .....	40
Z5 - Zero 5D (X, Y, Z, A, and B) axes .....	40
ZA - Zero A axis .....	40
ZB - Zero B axis .....	40
ZT - Zero Table Base Coordinates .....	40
ZX - Zero X axis .....	41
ZY - Zero Y axis .....	41
ZZ - Zero Z axis .....	41

## Commands

### CA - Cut Arch

*arch length, arch height{ , [O]ut-[I]n-[T]rue, direction, alignment-angle, plunge, repetitions, proportion-X, proportion-Y, 1-TAB feature ON, 1-no pull up at end, 1-do plunge from Z-axis 0}*

It is often useful to be able to quickly cut an arch that will span a particular distance. This is straightforward to do for a 180 degree arch (where the span length equals twice the height), but involves some calculation for any other arc length because you don't know how many degrees of arc need to be spanned. Making any arch is easy using this built-in routine because you only need to enter the basic parameters of length and height. You could accomplish the same cut with one of the other circle functions, but by using [\[CA\]](#) you can quickly set up the arch without doing any calculations. Just specify the length of the span, and the height you want the arch to rise to, and the rest is taken care of ... though you can further modify the arch by specifying additional parameters. Some of the instructions sound more complicated than they are. Try this one in preview to get a feel for how it works. For general purpose circles or arcs, use [\[CC\]](#) or [\[CP\]](#).

#### **Arch length**

The span of the arch. Cutting starts at the current cutter position and moves to the other end of the arch as defined by arch height and arch length, taking into account optional parameters. Arch length is required.

#### **Arch height**

How tall the arch will be. The degrees of arc represented by the arch are derived from arch height. If arch height is less than twice the length the arc will be some angle less than 180. See proportion, below, for modifying shape. Arch height is required.

#### **[O]ut-[I]n-[T]rue (default = T)**

How the cutter size will be figured in defining the path of the cut. 'O' means that the specified lengths refer to the outside of the cut. 'I' specifies that the lengths refer to the inside. Note that this means that for a given length, 'O' will make a bigger part than 'I'. Thus, to cut -out- an arch of a specified size, use the 'O' feature. To cut an arch shaped hole of a specified size, use the 'I' feature. 'T' moves the cutter on the path of the lengths specified and is the default.

#### **Direction (default = 1)**

Direction of the path followed. Use 1 to specify clockwise and -1 to specify counter-clockwise.

#### **Alignment-angle (default = 0)**

The angle at which the arch will be created on the table. Default (0) is horizontal, which means the arch will be cut to the right from the start point and oriented horizontally. Orientation can be changed by using an alignment angle that is oriented clockwise from horizontal. An alignment angle of 90 would produce an arc oriented vertically and cut from top to bottom on the Y axis.

#### **Plunge (default = 0)**

The depth of plunge for each repetition of the cut. Plunges down are specified with negative numbers. You do not need to plunge if the tool is already at cutting depth from a previous move. You can also plunge before calling the circle function by using the commands [\[MZ\]](#) or [\[JZ\]](#). Plunge is most useful if you are making the circle or arc in several passes as specified with Repetitions.

#### **Repetitions (default = 1)**

Number of passes to be made. This is useful if you're cutting thick material and need to make repeated passes, increasing the depth of the cutter each time, to achieve the full depth of cut. This typically is done with full circles. When you're cutting arcs/arches that are connected to other segments and need to make multiple passes, it's probably better to make multiple repetitions of the whole cut rather (e.g. in the [\[FP\]](#) instruction) than cutting the smaller segments separately.

#### **Proportion X, proportion Y (default = 1,1)**

Scaling values for the circle or arc/arches in the X and Y axes. You can alter the X or Y proportion to create ellipses or elliptical arcs/arches. For example, a circle with a diameter of 16, executed with an X proportion of 3 and a Y proportion of 1, will produce an ellipse that is 16 inches wide in the Y dimension and 48 inches wide in the X dimension. Cutting this shape from a start of 270 degrees to an end of 90 degrees would yield a wide arc/arch.

**1-TAB feature ON (default = 0)**

Cutting a circle/arc/arch with the TAB feature on will create small attachment tabs that will hold the circle being cut and prevent it from flying off. The circle is removed later. The tabs are executed as continuous smooth 3D moves that slope up then back down. The parameters defining the size of the TAB and distances between tabs can be set with [\[VB; Values for tabbing\]](#). If TABbing is turned on for a File, it remains on for individual circles within the file without turning it on for the circle/arc/arch.

**1-No pull up at end (default = 0).**

If this parameter is set to 1, the tool will remain at its plunge depth when it moves on to the next instruction.

**1-Offset plunge from Z-Axis 0 (default = 0).**

If this parameter is set to 1, the cutter will move to the Z-axis location 0 before the plunge instruction is executed and plunge depth will be taken from 0. This allows a plunge or repeated plunges to be done from the surface of the material or from a given location.

## CC – Cut Circle

```
diameter{ , [O]ut-[I]in-[T]rue, direction, begin-angle, end-angle, plunge,
repetitions, proportion-X, proportion-Y, [1-TAB feature ON; 2-POCKET feature
ON; 3-SPIRAL feature ON; 4-SPIRAL & Bottom Pass feature ON], 1-no pull up at
end, 1-do plunge from Z-axis 0 }
```

This is a very flexible routine for cutting circles, arcs, ellipses, etc.. If you simply specify a diameter, a circle will be transcribed from the current location, with the cutter following a path ('[T]rue') specified by the diameter and with the cut starting from the top of the circle (12 O'clock as you face a clock). But, there's a lot more you can do here ... have a look

Technical Note on ShopBot Circles Functions. (This note applies to all circles and arcs created by ShopBot [C] commands.) ShopBot converts your instructions for circles and arcs to a series of very small vectors/segments. When circle resolution is set to the lowest value (see [\[VU\]](#)), the vectors are so close to the size of steps themselves, that there is little actual difference between executing the circle as a series of small vectors or as the output of a 'circular interpolation' process. Because the vectors are all buffered in a movement block, they are executed efficiently and smoothly. There may be occasions where controlling the resolution of the circle or the size of the movement block is important and in this case you will find the discussion of 'Circle Segment Resolution' under [\[VU\]](#) of interest. When cutting a very small circle that would require a resolution smaller than the current segment resolution, the software automatically downshifts to a lower resolution and then returns to the resolution that existed before the small circle.

**Diameter**

The diameter of the circle or arc. See the next paragraph for how this is measured. ShopBot circle functions are ultimately generated from very small line segments. The resolution - size of the line segments - can be set with the [\[VU\]](#) Command. Diameter is the only required parameter for a circle.

**The [O]ut-[I]in-[T]rue (default = T)**

How the cutter size will be figured in defining the path of the cut. 'O' means that the specified diameter refers to the outside of the cut. 'I' specifies that the diameter refers to the inside. Note that this means that for a given diameter, 'O' will make a bigger circle than 'I'. Thus, to cut a wheel of a specified diameter, use the 'O' feature. To make a hole of a specified diameter, use the 'I' feature. 'T' moves the cutter on the path of the diameter specified and is the default setting.

**Direction (default = 1)**

The direction of the path followed. Use 1 to specify clockwise and -1 to specify counter-clockwise.

**Begin-angle, end-angle (default = , )**

Allows you to specify the starting and ending point of a circle or to define an arc of less than 360 degrees. All cuts begin at the current location of the tool. Angles are specified in degrees as measured with 0 degrees at the top of the circle, that is, at the 12 O'clock position. A cut that starts at 0 degrees and ends at 360 degrees will be a full circle that starts at the top of the circle (assuming a clockwise direction). This circle will be cut below the start point. A cut that starts at 180 degrees and ends at 540 degrees will be a circle that starts and ends at the bottom (i.e., 6 O'clock; assuming a clockwise direction). This circle will be cut above the starting location. Starting at 270 degrees and ending at 630 will produce a circle drawn to the right of the starting point. Similarly, begin-angle, end-angle are used to define the starting point and end point for an arc. Starting at 90 degrees and ending at 180 will produce a 90-degree arc in the lower right quadrant when the direction setting is 1 (clockwise). A direction setting of -1 (counter-clockwise) will produce a 270-degree arc

filling all but the lower right quadrant of a circle (an end angle of -180 degrees will also work in this case). For example, if you were cutting a box and wanted to create a rounded corner after having just cut the bottom edge left to right, you would use a start angle of 180 degrees, an end angle of 90 degrees, and a direction of -1 (counter-clockwise) to produce the lower right rounded corner. The default start and end angles are both blanks. A blank in the start angle, causes the angle to be treated as 0 if a clockwise circle and 360 if a counter-clockwise circle. A blank in the end angle is treated as the end of a full circle, whatever the start point and direction.

#### **Plunge (default = 0)**

The depth of plunge for each repetition of the cut. Plunges down are specified with negative numbers. You do not need to plunge if the tool is already at cutting depth from a previous move. You can also plunge before calling the circle function by using the commands [\[MZ\]](#) or [\[JZ\]](#). Plunge is most useful if you are making the circle or arc cut in several passes as specified with Repetitions.

#### **Repetitions (default = 1)**

The number of passes to be made. This is useful if you're cutting thick material and need to make repeated passes, increasing the depth of the cutter each time, to achieve the full depth of cut. This typically is done with full circles. When you're cutting arcs that are connected to other segments and need to make multiple passes, it's probably better to make multiple repetitions of the whole cut rather (e.g. in the [\[FP\]](#) instruction) than cutting the smaller segments separately.

#### **Proportion X, proportion Y (default = 1,1)**

Scaling values for the circle or arc in the X and Y axes. You can alter the X or Y proportion to create ellipses or elliptical arcs. For example, a circle with a diameter of 16, executed with an X proportion of 3 and a Y proportion of 1, will produce an ellipse that is 16 inches in the Y dimension and 48 inches in the X dimension. Cutting this shape from a start of 270 degrees to an end of 90 degrees would yield a wide arc/arch.

#### **[1-TAB feature ON; 2-POCKET feature ON; 3-SPIRAL plunge feature ON; 4-SPIRAL & Bottom Pass feature ON]. (Default = 0,0,0)**

Cutting a circle/arc with the TAB feature on will create small attachment tabs on the last pass that will hold the circle/arc being cut and prevent it from flying off. The circle is removed later. Starting with Version 2, the tabs are executed as continuous smooth 3D moves that slope up then back down. The parameters defining the size of the TAB and distances between tabs can be set with [\[VB\]](#); Values for tabbing]. If TABbing is turned on for a File, it remains on for individual circles/arcs within the file without turning it on for the circle/arc.

Activating the POCKETing feature will cause the circle to be 'hollowed out' with increasingly smaller circles being cut inside the initial circle. TAB and POCKET are not used at the same time, and each is designed to work only with full circles.

Activating the SPIRAL feature causes the defined plunge to be made gradually as the cutter is circling down. If repetitions is 1 then the plunge distance is distributed over the whole circle. If repetitions is larger than 1 then the plunge distance is moved for each repetition of the circle (e.g. repeated spiraling). SPIRAL with Bottom Pass causes a final pass at the terminal depth. By default, TAB, POCKET, and SPIRAL are OFF (0).

#### **1-No pull up at end (default = 0).**

If this parameter is set to 1, the tool will remain at its plunge depth when it moves on to the next instruction.

#### **1-Offset plunge from Z-Axis 0 (default = 0).**

If this parameter is set to 1, the cutter will move to the Z-axis location 0 before the plunge instruction is executed and plunge depth will be taken from 0. This allows a plunge or repeated plunges to be done from the surface of the material.

## CG - Cut G-code Circle

```
diameter, X-end-point, Y-end-point, *X-center-offset(I), *Y-center-offset(J)
{,[O]ut-[I]n-[T]rue, direction, plunge, repetitions, proportion-X,
proportion-Y, [1-TAB feature ON; 2-POCKET feature ON; 3-SPIRAL feature ON; 4-
SPIRAL & Bottom Pass feature ON], 1-no pull up at end, 1-do plunge from Z-
axis 0}
```

This command is provided primarily for compatibility with G-Code format and G-Code conversions. It has all the capabilities of [\[CC\]](#) but it makes the specification of the circle/arc dependent on its start point (the current

tool position) and the specified end-point. There are two methods for the circle or arc to be computed. Either the diameter can be specified (this works only for arcs less than 180 degrees) or the circle/arc can be derived by entering the location of the center-point of the circle. Here, center-point is indicated by its offset in X and Y from the current location. \*Note that only the diameter or the center-offsets may be used. If a diameter is entered, the X and Y offsets will be ignored.

### **Diameter**

The diameter of the circle or arc. See the next paragraph for how this is measured. For this function, diameter is not required. If it is provided, a subsequent X and Y center-point offset will be ignored.

### **X-end-point, y-end-point**

The end point of the circle/arc. If no values are provided, it is assumed the end point is the same as the start point. A circle can not be created from only a diameter in this command.

### **[O]ut-[I]n-[T]rue (default = T)**

How the cutter size will be figured in defining the path of the cut. 'O' means that the specified diameter refers to the outside of the cut. 'I' specifies that the diameter refers to the inside. Note that this means that for a given diameter, 'O' will make a bigger circle than 'I'. Thus, to cut a wheel of a specified diameter, use the 'O' feature. To make a hole of a specified diameter, use the 'I' feature. 'T' moves the cutter on the path of the diameter specified and is the default setting.

### **Direction (default = 1)**

The direction of the path followed. Use 1 to specify clockwise and -1 to specify counter-clockwise.

### **Offset-to-x-center-point(I), offset-to-y-center-point(J)**

The location of the center of the circle from which diameter and arc angle are calculated. If no diameter is given, these parameters are required. \*Note that only the diameter or the center-offsets may be used. If a diameter is entered, the X and Y offsets will be ignored. Also, for compatibility with 'modal' G-Code operation, the X or Y offset will be maintained for subsequent calls of the function if no new X or Y value is given (i.e. if no diameter and no offset is specified, the arc will be computed from the previous X and Y center-point offsets).

### **Plunge (default = 0)**

Specifies the depth of plunge for each repetition of the cut. Plunges down are specified with negative numbers. You do not need to plunge if the tool is already at cutting depth from a previous move. You can also plunge before calling the circle function by using the commands [\[MZ\]](#) or [\[JZ\]](#). Plunge is most useful if you are making the circle or arc cut in several passes as specified with

### **Repetitions.**

X indicates the number of passes to be made. This is useful if you're cutting thick material and need to make repeated passes, increasing the depth of the cutter each time, to achieve the full depth of cut. This typically is done with full circles. When you're cutting arcs that are connected to other segments and need to make multiple passes, it's probably better to make multiple repetitions of the whole cut rather (e.g. in the [\[FP\]](#) instruction) than cutting the smaller segments separately.

### **Proportion X, proportion Y (default = 1,1)**

specify scaling values for the circle or arc in the X and Y axes. You can alter the X or Y proportion to create ellipses or elliptical arcs. For example, a circle with a diameter of 16, executed with an X proportion of 3 and a Y proportion of 1, will produce an ellipse that is 16 inches wide in the Y dimension and 48 inches wide in the X dimension. Cutting this shape from a start of 270 degrees to an end of 90 degrees would yield a wide arc/arch. Default proportions are 1 and 1.

### **[1-TAB feature ON; 2-POCKET feature ON; 3-SPIRAL plunge feature ON; 4-SPIRAL & Bottom Pass feature ON]. (Default = 0,0,0)**

Cutting a circle/arc with the TAB feature on will create small attachment tabs on the last pass that will hold the circle/arc being cut and prevent it from flying off. The circle is removed later. Starting with Version 2, the tabs are executed as continuous smooth 3D moves that slope up then back down. The parameters defining the size of the TAB and distances between tabs can be set with [\[VB; Values for tabbing\]](#). If TABbing is turned on for a File, it remains on for individual circles/arcs within the file without turning it on for the circle/arc.

Activating the POCKETing feature will cause the circle to be 'hollowed out' with increasingly smaller circles being cut inside the initial circle. TAB and POCKET are -not- used at the same time, and each is designed to work only with full circles.

Activating the SPIRAL feature causes the defined plunge to be made gradually as the cutter is circling down. If repetitions is 1 then the plunge distance is distributed over the whole circle. If repetitions is larger than 1 then the plunge distance is moved for each repetition of the circle (e.g. repeated spiraling). SPIRAL with Bottom Pass causes a final pass at the terminal depth. By default, TAB, POCKET, and SPIRAL are OFF (0).

**1-No pull up at end (default = 0).**

If this parameter is set to 1, the tool will remain at its plunge depth when it moves on to the next instruction.

**1-Offset plunge from Z-Axis 0 (default = 0).**

If this parameter is set to 1, the cutter will move to the Z-axis location 0 before the plunge instruction is executed and plunge depth will be taken from 0. This allows a plunge or repeated plunges to be done from the surface of the material.

## CP - Cut Circle from Center Point

*diameter, X-center-point, Y-center-point {,[O]ut-[I]in-[T]rue, direction, start-angle, end-angle, plunge, repetitions, proportion-X, proportion-Y, [1-TAB feature ON; 2-POCKET feature ON; 3-SPIRAL feature ON; 4-SPIRAL & Bottom Pass feature ON], 1-no pull up at end, 1-do plunge from Z-axis 0}*

This command has all the capabilities of [\[CC\]](#) but it adds the specification of the center-point of the circle to the parameters. When this command is executed the tool moves at Jog speed to the start point of the circle from its current location (\*it is up to you to make sure the bit is out of the material). The start point is computed from the given center-point of the circle before the move is started. The [\[CC\]](#) command is usually the most convenient to use if you are creating a circle from a known location of the tool, or moving from a line into an arc segment. However, this [\[CP\]](#) command is very useful when you just want to specify circles in a cutting file. With [\[CP\]](#) you won't need to worry about getting the cutter to the right spot on the circumference to start the cut - this is computed for you. This command thus requires diameter and an X,Y center-point location.

**X-center-point, y-center-point**

Coordinates for the center of the circle. The tool does not actually ever move to this location. The center-point parameters are required to use this [\[CP\]](#) function.

**Diameter**

The diameter of the circle or arc. See the next paragraph for how this is measured. ShopBot circle functions are ultimately generated from very small line segments. The resolution - size of the line segments - can be set with the [\[VU\]](#) command. Diameter is third required parameter for a [\[CP\]](#) circle.

**The [O]ut-[I]in-[T]rue (default = T)**

How the cutter size will be figured in defining the path of the cut. 'O' means that the specified diameter refers to the outside of the cut. 'I' specifies that the diameter refers to the inside. Note that this means that, for a given diameter, 'O' will make a bigger circle than 'I'. Thus, to cut a wheel of a specified diameter, use the 'O' feature. To make a hole of a specified diameter, use the 'I' feature. 'T' moves the cutter on the path of the diameter specified and is the default setting.

**Direction (default = 1)**

The direction of the path followed. Use 1 to specify clockwise and -1 to specify counter-clockwise.

**Begin-angle, end-angle (default = 0,0)**

Allows you to specify the starting and ending point of a circle or to define an arc of less than 360 degrees. Angles are specified in degrees as measured with 0 degrees at the top of the circle, that is, at the 12 O'clock position. A cut that starts at 0 degrees and ends at 360 degrees will be a full circle that starts at the top of the circle. This circle will be cut below the start point. A cut that starts at 180 degrees and ends at 540 degrees will be a circle that starts and ends at the bottom (i.e., 6 o'clock). This circle will be cut above the starting location. Similarly, begin-angle, end-angle are used to define the starting point and end point for an arc. Starting at 90 degrees and ending at 180 will produce a 90-degree arc in the lower right quadrant when the direction setting is 1 (clockwise). A direction setting of -1 (counter-clockwise) will produce a 270 degree arc filling all but the lower right quadrant of a circle. See discussion of default values for the [\[CC\]](#) command, which are handled similarly here.

**Plunge (default = 0)**

The depth of plunge for each repetition of the cut. Plunges down are specified with negative numbers. You do not need to plunge if the tool is already at cutting depth from a previous move. You can also plunge



before calling the circle function by using the commands [\[MZ\]](#) or [\[JZ\]](#). Plunge is most useful if you are making the circle or arc cut in several passes as specified with Repetitions.

#### **Repetitions (default = 1)**

The number of passes to be made. This is useful if you're cutting thick material and need to make repeated passes, increasing the depth of the cutter each time, to achieve the full depth of cut. This typically is done with full circles. When you're cutting arcs that are connected to other segments and need to make multiple passes, it's probably better to make multiple repetitions of the whole cut rather (e.g. in the [\[FP\]](#) instruction) than cutting the smaller segments separately.

#### **Proportion X, proportion Y (default = 1,1)**

Scaling values for the circle or arc in the X and Y axes. You can alter the X or Y proportion to create ellipses or elliptical arcs. For example, a circle with a diameter of 16, executed with an X proportion of 3 and a Y proportion of 1, will produce an ellipse that is 16 inches in the Y dimension and 48 inches in the X dimension. Cutting this shape from a start of 270 degrees to an end of 90 degrees would yield a wide arc/arch.

#### **[1-TAB feature ON; 2-POCKET feature ON; 3-SPIRAL plunge feature ON; 4-SPIRAL & Bottom Pass feature ON]. (Default = 0,0,0)**

Cutting a circle/arc with the TAB feature on will create small attachment tabs on the last pass that will hold the circle/arc being cut and prevent it from flying off. The circle is removed later. Starting with Version 2, the tabs are executed as continuous smooth 3D moves that slope up then back down. The parameters defining the size of the TAB and distances between tabs can be set with [\[VB; Values for tabbing\]](#). If TABbing is turned on for a File, it remains on for individual circles/arcs within the file without turning it on for the circle/arc.

Activating the POCKETing feature will cause the circle to be 'hollowed out' with increasingly smaller circles being cut inside the initial circle. TAB and POCKET are -not- used at the same time, and each is designed to work only with full circles.

Activating the SPIRAL feature causes the defined plunge to be made gradually as the cutter is circling down. If repetitions is 1 then the plunge distance is distributed over the whole circle. If repetitions is larger than 1 then the plunge distance is moved for each repetition of the circle (e.g. repeated spiraling). SPIRAL with Bottom Pass causes a final pass at the terminal depth. By default, TAB, POCKET, and SPIRAL are OFF (0).

#### **No pull up at end (default = 0).**

If this parameter is set to 1, the tool will remain at its plunge depth when it moves on to the next instruction.

#### **1-Offset plunge from Z-Axis 0 (default = 0).**

If this parameter is set to 1, the cutter will move to the Z-axis location 0 before the plunge instruction is executed and plunge depth will be taken from 0. This allows a plunge or repeated plunges to be done from the surface of the material.

## CR - Cut Rectangle

```
Length X{ ,length Y, [O]ut-[I]in-[T]rue, direction, start, plunge,
repetitions, [1-TAB feature ON; 2-POCKET feature ON], 1-do plunge from Z-axis
0}
```

This is a flexible routine for cutting rectangles. If you specify only an X length, a square will be cut, starting from the current location, with the cutter following a path ('[T]rue') specified by the X length, and with the cut starting from the top left hand corner of the square. But MUCH more is possible.

#### **Length X , length Y**

The size of the rectangle in terms of the X and Y dimensions. If only X is specified, a square is assumed. An X value is required for the [\[CR\]](#) function. See the next paragraph for how lengths are measured.

#### **The [O]ut-[I]in-[T]rue (default = T)**

How the cutter size will be figured in defining the path of the cut. 'O' means that the specified lengths refer to the outside of the cut. 'I' specifies that the lengths refer to the inside. Note that this means that for a given length, 'O' will make a bigger rectangle than 'I'. Thus, to cut a block of a specified size, use the 'O' feature. To cut a rectangular hole of a specified size, use the 'I' feature. 'T' moves the cutter on the path of the lengths specified and is the default.

**Direction (default = 1)**

The direction of the path followed. Use 1 to specify clockwise and -1 to specify counter-clockwise.

**Start (default = 4)**

The corner of the rectangle from which the cut starts. Use 1 to specify the upper left, 2 for upper right, 3 for bottom right, and 4 for bottom left. Cutting always starts at the current position of the tool. Thus, a rectangle cut with a starting position of 3 will be placed to the upper left of the starting point. Default start is the bottom left (4).

**Plunge (default = 0)**

The depth of plunge for each repetition of the cut. Plunges down are specified with negative numbers. You do not need to plunge if the tool is already at cutting depth from a previous move. You also can plunge by using the commands [\[MZ\]](#) or [\[JZ\]](#) before calling the rectangle function. Plunge is most useful if you're making the rectangle cut in several passes as specified with Repetitions.

**Repetitions (default = 1)**

The number of passes to be made. This is useful if you're cutting thick material and need to make repeated passes, increasing the depth of the cutter each time, to achieve the full depth of cut.

**[1-TAB feature ON; 2-POCKET feature ON] (default = 0).**

Cutting a rectangle with the TAB feature on will create small attachment tabs on the last cut that will hold the rectangle being cut and prevent it from flying off. The rectangle is removed later. The parameters defining the size of the TAB and distances between can be set with [\[VB, Values for tabbing\]](#). Activating the POCKETing feature will cause the rectangle to be 'hollowed out' with increasingly smaller rectangles being cut inside the initial rectangle. TAB and POCKET are not used at the same time. By default, both TAB and POCKET are OFF.

**1-Offset plunge from Z-Axis 0 (default = 0).**

If this parameter is set to 1, the cutter will move to the Z-axis location 0 before the plunge instruction is executed and plunge depth will be taken from 0. This allows a plunge or repeated plunges to be done from the surface of the material.

## FC - File Conversion

*filename{.dxf, .plt, .hpg, .tif, .tap, etc.}*

This command calls-up ShopBot's file conversion functions. Files of the following type can be converted to ShopBot Part Files (.sbp): 'dxf' files (drawing exchange format); '.plt' or '.hpg' files (HP graphics language plotter); '.nc' or other formats of CNC files (standard G-code files); and '.tif' files (a raster type file bitmap file in which gray scale is extracted as 3D). If ShopBot is provided a filename with one of these listed extensions, it will convert the file using the appropriate converter and create a '.sbp' Part File. Depending on the type of conversion, the user may be prompted for additional information. If the file that is submitted for conversion does not have a recognized file type extension, the user will be prompted for the type of conversion to attempt.

Description of how each type of file is converted and the entities that are supported for that file type is provided in the full Users Guide, under 'Format Conversions.'

## FE - File Edit

**Select or type in a file name and click "Open" or hit <Enter>.**

This utility calls up a text editor to edit the designated filename. If no filename is given, the software will try to load the latest file you've loaded with [\[FE\]](#) or [\[FP\]](#); otherwise it will start the editor with a blank file. By default, ShopBot will use the ShopBot editor SbEdit. You can define an alternate editor using [\[UN\]](#) if you prefer.

Help with the ShopBot commands is available under the Help menu by clicking on ShopBot Command Reference. [\[FE\]](#) can be called from within a Part File along with a file name in order to display information to the operator of the tool. After the editor is exited, the Part File continues.

See Also: [\[UN\]](#)

## FG - File Load in Goto/Single-Step Mode

```
filename{ , proportion-X, proportion-Y, proportion-Z, repetitions, [1-3D  
Offset ON; 2-2D Offset ON], plunge, 1-TAB feature ON, 1-do plunge from Z-axis  
0}
```

This command allows the file to be executed on a line-by-line basis. It can be used in either MOVE or PREVIEW Mode. When file execution begins with [\[FG\]](#), the first line of the file is read and displayed. Then you are offered a selection of additional commands that will allow you to single-step or to position yourself in the file and single-step or Run from that location.

By using the [\[FG\]](#) command you are able to execute as much or as little of a file as you would like, and to study the cutting action while you are executing a Part File one line at a time. Also, by moving to a line number you can return to the place in a file where previous cutting was disrupted (e.g. by a broken bit or power failure).

Part Files are text files, usually having the filename extension '.sbp.' Part files are the key to productivity and creativity with your tool and they can be created or generated in a number of different ways. No matter how they are created, all Part files contain lists of instructions for controlling and moving your ShopBot. The 'commands' in a Part file are identical to the commands you enter from the keyboard. PART files also can contain other programming instructions that increase the flexibility and usefulness of your ShopBot. See the later section on creating files and programming your tool to learn how to use programming instructions and regular ShopBot commands together.

### **Filename**

The name of the file you want to load. If the file is not in the current PART directory you can browse to a higher or lower level directory using the file display in the first Fill-In Sheet. The filename parameter is required, but if you do not enter a parameter and just hit <Enter>, the last filename used will be inserted.

The Fill-In sheet provides a page for entering parameters related to executing the file. It appears after a file has been selected.

### **Proportion-X, proportion-Y, proportion-Z (default = 1,1,1)**

Set to scale to a different size the cuts made by commands contained in the file.

### **Repetitions (default = 1)**

The number of times that the total contents of the file will be executed from the current location.

### **1-TAB feature ON (Default = 0)**

Cutting a part with the TAB feature ON will create small attachment tabs (on the last pass if there are more than 1 repetition) that will hold the part being cut and prevent it from flying off. Then you punch out the part later. Starting with Version 2, the tabs are executed as continuous smooth 3-D moves that slope up then back down. The parameters defining the size of the TAB and distances between tabs can be set with [\[VB\]](#); Values for tabbing]. If TABbing is turned on for a File, it remains on for individual sub-files within the file.

### **Offset**

If set to 1 will produce a 3D offset by which all the moves in the file will be executed from the current X,Y, Z location (i.e. as if the current location were a temporary 0,0,0); with offset set to 2 the file will be executed from the current X,Y location; the Z moves will not be offset but will be cut at the location indicated in the file. OFFSET is the feature that allows you to cut a single Part file at many locations on your workpiece, allowing easy cutting of multiple copies of a part. If no offset is turned on (OFF=0 or blank; the default), the file is executed using the specific values and locations in the file.

### **Offset plunge from Z-Axis 0 (default = 0).**

If this parameter is set to 1, the cutter will move to the Z-axis location 0 before the plunge instruction is executed and plunge depth will be taken from 0. This allows a plunge or repeated plunges to be done from the surface of the material.

See Also: [\[FP\]](#) for standard execution of files

## FP - File Load PART File

```
filename{ , proportion-X, proportion-Y, proportion-Z, repetitions, [1-3D  
Offset ON; 2-2D Offset ON], plunge, 1-TAB feature ON, 1-do plunge from Z-axis  
0}
```

This is the command that loads/runs/executes/cuts a ShopBot Part File. Part Files are text files (saved in Windows, DOS, or Unix format), usually having the filename extension '.sbp.' Part files are the key to productivity and creativity with your tool and they can be created or generated in a number of different ways. No matter how they are created, all Part Files contain lists of instructions for controlling and moving your ShopBot. The 'commands' in a Part File are identical to the commands you enter from the keyboard. Part Files also can contain other programming instructions that increase the flexibility and usefulness of your ShopBot. See the ShopBot "User Guide" or "Programming Handbook" to learn about creating files and programming your tool to learn how to use programming instructions and regular ShopBot commands together.

[FP] will also run G-code files containing most standard G-code instructions (.tap files and .nc files; select 'ALL' to show all file types when selecting file; see the Users Guide for a list of currently supported G and M codes; these are the same codes as supported by the G-code converter in [FC]). You can thus run legacy G-code in ShopBot in 3 ways. 1) With the [FC] converter – recommended for cases where you have just a few older G-code files that you regularly use. Once converted to ShopBot code (which only takes a second or two) you will be able to read, edit, and update the code more easily. 2) With [FP] – recommended for cases where you have a legacy program that only generates G-code and where you must frequently generate new G-code files. Because you can intermix ShopBot and G-code, you can still enhance these files by adding ShopBot Commands or routines. 3) With [SI] – G-code programmers can enter G-code commands straight from the keyboard, again they can be intermixed with ShopBot Commands to add more flexibility or capability.

The Fill-In sheet provides the page for entering parameters related to executing the file. It appears after a file has been selected.

This command offers the option of TABBING, a system for leaving small pieces of material to hold a part in place during cutting. As noted below, the Tabbings system will work in the presence of 3D moves in the file.

The [FP] command also allows for cutting of a 'template' type file with repeated plunges. Plunge values and number of repetitions is assigned when the file is called up. For multi-pass template cutting to work correctly, the file must contain only 2D moves; that is, there should be no Z moves in the file. If you need to do multiple passes of several different entities, create a file for each, and then call them with a master file that activates the multi-pass function for each.

If you have a file with Z moves that you want to cut in several passes, just call the file several times and use the Z proportion parameter to control increasing depth (e.g. cut at .50, .85, and 1.00). This operation can be automated by creating a master file that will do the calls of [FP].

### **Filename**

The name of the file you want to load. If the file is not in the current Part directory you can browse to a higher or lower level directory using the file display in the first Fill-In Sheet. The filename parameter is required, but if you do not enter a parameter and just hit <Enter>, the last filename used will be inserted.

### **Proportion-X, proportion-Y, proportion-Z (default = 1,1,1)**

Scale to a different size the cuts made by commands contained in the file.

### **Repetitions (default = 1)**

The number of times that the total contents of the file will be executed from the current location.

### **Offset**

If set to 1 will produce a 3D offset by which all the moves in the file will be executed from the current X,Y,Z location (i.e. as if the current location were a temporary 0,0,0); with offset set to 2 the file will be executed from the current X,Y location; the Z moves will not be offset but will be cut at the location indicated in the file. OFFSET is the feature that allows you to cut a single Part file at many locations on your workpiece, allowing easy cutting of multiple copies of a part. If no offset is turned on (OFF=0 or blank; the default), the file is executed using the specific values and locations in the file.

**Plunge (default = 0)**

The depth of plunge for each repetition of the file. Plunges down are specified with negative numbers. You do not need to plunge if the tool is already at cutting depth from a previous move. You also can plunge by using the commands [\[MZ\]](#) or [\[JZ\]](#) before calling the file. With a file, plunge would typically be used to cut a template type of file that did not have plunges in the file itself. It might further be of use if the file were being cut in several passes specified with Repetitions.

**TAB feature ON (Default = 0)**

Cutting a part with the TAB feature ON will create small attachment tabs (on the last pass if there are more than 1 repetition) that will hold the part being cut and prevent it from flying off. Then you punch out the part later. From Version 2, the tabs are executed as continuous smooth 3-D moves that slope up then back down. The parameters defining the size of the TAB and distances between tabs can be set with [\[VB\]](#); Values for tabbing]. If TABbing is turned on for a File, it remains on for individual sub-files within the file.

**1-Offset plunge from Z-Axis 0 (default = 0)**

If this parameter is set to 1, the cutter will move to the Z-axis location 0 before the plunge instruction is executed and plunge depth will be taken from 0. This allows a plunge or repeated plunges to be done from the surface of the material.

See also: [\[FG\]](#) for single-stepping through files

## FS - File, Set the PARTS File Directory

*path and directory name{, search mask}*

Sets the current Part File directory.

**Path and directory name**

Use the full path and directory name (e.g., C:\SHOPBOT\PARTS). Entering a blank will restore the program directory (the one you started the ShopBot Control Software from) as the current PARTS directory. You can see what ShopBot considers the current PARTS directory by looking in the display provided by [\[UV\]](#). Note that when the directory is changed during the file selection process of [\[FP\]](#) or [\[FE\]](#) or other functions that include file selection, the change that is made stays in effect until altered by [\[FS\]](#) or another change made during file selection.

**Search mask**

The file types, designated by filename extensions, that will be made visible when the file selector Fill-In sheet is called up during a function such as [\[FP\]](#) or [\[FE\]](#). The default is '\*.sb?', which means that all filenames having an extension with an 'sb' as the first two letters will be displayed. [As a matter of standard practice, we use the '.sb' in the first two positions of the filename extension to indicate that a file has something to do with ShopBot, and we use '.sbp' to indicate a standard PART file.] '\*' is a general purpose wild card that means any number of any set of characters, while ? is a wild card that means any single character. Using the mask '\*.\*', for example, will cause all filenames in a directory/folder to be displayed. Using the mask 'proj??.sbp', for example, will result in only filenames that begin with 'proj' and have just two additional characters before an '.sbp' to be displayed.

The [\[FS\]](#) command can be used within a Part File to set the new PARTS directory for a group of sub-Part Files. It can be used by giving the full path or the relative location (e.g. just the sub directory if the new path is below the current directory). At the end of the Part File, [\[FS\]](#) with the DOS up-level shorthand ".." can be used to return to the original directory.

## Fill-In Sheet Sample

On all commands that have a large or complex number of parameters, the Fill-In Sheets make it easier to provide the values that you want to set. The Fill-In Sheet lays out the various parameters with a more complete explanation of their use than is shown on the parameter line. You can move between fields by tabbing and can select from available options with the pull down menu.

This is a sample Fill-In Sheet for the [\[CC\]](#) Command. It displays all the parameters available for you to enter to control the execution of the command. Required parameters are marked with a Green Asterisk. If there are more parameters than fit on a single sheet, you can scroll down to them.

Parameter Name:	Value:	Required
Diameter		*
Outside-Inside-True	T = true path	
Direction	1 = clockwise	
Begin Angle		
End Angle		
Plunge (per repetition)	-0.00	
Repetitions	1	
Proportion X	1.00	
Proportion Y	1.00	
No Pull-Up	0 - Off	
Plunge from 0	0 - Off	
Special: Tabbing, Pocketing, & Spiral Plunge		
Tab, Pocket, Spiral	0 - No feature	
Related Commands:		

Reset  Display Recall Last OK

Passing the mouse cursor over the Parameter Name will display a 'Tool Tip' that explains the parameter. Passing the cursor over the value box will display the default setting for the parameter and it's range.

Clicking 'Reset' at the bottom left (or ALT-R) will reset all parameters to the their default values. Checking or unchecking the 'Display' box (or ALT-D) will turn off or on the display of Fill-In Sheets for this Command. Clicking "Recall Last" will set all parameters to those used the last time this command was executed from a Fill-In sheet.

Click OK/START to continue when you have completed any of the Fill-Ins.

Note that for a few commands the order of parameters on a Fill-In Sheet may not be the order that should be used in a Parameter Line of in a ShopBot Part File. The Fill-In Sheet provides the parameters in the simplest order, but for compatibility with earlier versions of ShopBot software, this may not be the order required for Part Files. The Part File order is shown in the abbreviated listing, just under the Command Name. It is also provided in a simple format on the QuickRef Edit page [\[HE\]](#).

## HA – Help About

This command provides information about the particular version of ShopBot Control Software that is running.

## HB - Help: ShopBot website Technical Bulletins

Takes you to the Technical Bulletin area of the ShopBot website if you are connected to the internet.

## HC - Help: Command Reference

This command takes you to the contents of the full Command Reference Manual (what you are reading now). The Command Reference provides complete information on all ShopBot commands. See the User's Guide [\[HU\]](#) for information on how to use your ShopBot and the accompanying software.

## HE - Help: Quick Reference Page (editing)

This command opens a Quick Reference page for ShopBot Commands laid out with parameters in columns to be useful when you are writing or editing Part Files with the ShopBot editor.

## HN - Help: Send a request for help to ShopBot Technical Support

This command helps you easily ask ShopBot Tech Support for help if you are connected to the internet on the computer running the ShopBot software. Fill in the request as well as you can and someone will reply by email or a phone call as soon as possible.

## HQ - Help: Quick Reference Page

This command opens a Quick Start page to help new users run their first few Part Files at the tool. If you are a new operator or haven't used your tool for a while, have a look at these instructions.

## HR - Help: Quick Reference Page

This command opens a Quick Reference page of ShopBot Commands to help you remember a command you may want to enter at the keyboard. This is one you might want to print out and keep near your tool.

## HT - Help: Troubleshooting & Maintenance

This command brings up troubleshooting help for common problems and maintenance suggestions.

## HU - Help: ShopBot User Guide

This command accesses the ShopBot User Guide. The Guide provides detailed discussion on all aspects of using your ShopBot. This is the same document as the printed Guide in ring binder you received from ShopBot.

## HW - Help ShopBot Website

This command will take you to the ShopBot Tools website [www.shopbottools.com](http://www.shopbottools.com) if you are connected to the internet.

## J2 - Jog 2 Dimensions

*{x-location/distance, y-location/distance}*

Makes an X and Y axis move to the designated location or a specified distance (if in Relative Mode) at current Jog speed. These 2-D interpolated moves are made on a diagonal from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting. If a value is not entered for one of the parameters, the value defaults to the current location assuming a comma is used as a spacer to designate the correct parameter field (cf. 'modal' in G-code programming).

## J3 - Jog 3 Dimensions

*{x-location/distance, y-location/distance, z-location/distance}*

Makes an X, Y, and Z axis move to the designated location or a specified distance (if in Relative Mode) at current Jog speed. These 3-D interpolated moves are made on a diagonal from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting. If a value is not entered for one of the parameters, the value defaults to the current location assuming a comma is used as a spacer to designate the correct parameter field.

## J4 - Jog 4 Dimensions

*{x-location/distance, y-location/distance, z-location/distance, a-location/distance}*

Makes an X, Y, Z and A axis move to the designated location or a specified distance (if in Relative Mode) at current Jog speed. These 3-D interpolated moves are made on a diagonal from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting. If a value is not entered for one of the parameters, the value defaults to the current location assuming a comma is used as a spacer to designate the correct parameter field.

## J5 - Jog 5 Dimensions

*{x-location/distance, y-location/distance, z-location/distance, a-location/distance, b-location/distance}*

Makes an X, Y, Z, A, and B axis move to the designated location or a specified distance (if in Relative Mode) at current Jog speed. These 3-D interpolated moves are made on a diagonal from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting. If a value is not entered for one of the parameters, the value defaults to the current location assuming a comma is used as a spacer to designate the correct parameter field.

## JA - Jog A axis

*a-location/distance*

Makes the A axis move to the designated location or a specified distance (if in Relative Mode) at Jog speed. These moves are made from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting. The A axis is sometimes used as a rotational axis. When this is done, the location/distance value may be in degrees or other arbitrary units as defined by its Unit values with [\[VU\]](#).

## JB- Jog B axis

*b-location/distance*

Makes the B axis move to the designated location or a specified distance (if in Relative Mode) at Jog speed. These moves are made from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting. The B axis is sometimes used as a rotational axis. When this is done, the location/distance value may be in degrees or other arbitrary units as defined by its Unit values with [\[VU\]](#).

## JS - Jog Speed

*{xy-Speed, z-Speed, a-Speed, b-Speed}*

Sets the speed in inches or mm per second for Jog Speed movement in the XY, Z, A, and B axes. The new values will appear on screen in the upper left panel. This speed represents the actual movement speed of the cutter tip. Because the vector speed along a diagonal represents the geometric addition of the X and Y vectors, for any given speed, the motors will operate more slowly on a diagonal move than on a vertical or horizontal move. The change you notice is the sound of the motors is as it should be. For a circle the speed



of the motors will continuously change up and down in order to achieve a constant vectored movement speed.

This command is similar to the [\[MS\]](#) command for setting Move speed. It also has the same effect as setting the 4th and 5th parameter of [\[VS; Values Speed\]](#) command, which lets you set all 6 speeds (including accessory axis) at the same time.

SEE ALSO: [\[VR\]](#) to set the acceleration ramps and deceleration ramps for each type of movement.

## JX - Jog X-axis

*x-location/distance*

Makes an X axis move to the designated location or a specified distance (if in Relative Mode) at Jog speed. These moves are made from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting.

## JY - Jog - Y axis

*y-location/distance*

Makes a Y axis move to the designated location or a specified distance (if in Relative Mode) at Jog speed. These moves are made from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting.

## JZ - Jog Z-axis

*z-location/distance*

Makes a Z axis move to the designated location or a specified distance (if in Relative Mode) at Jog speed. These moves are made from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting.

## JH - Jog Home

Moves the tool to the home (0,0) XY position at Jog speed. If the tool is positioned at a plunge location below that defined as the Safe-Z height (default .15) on the Z Axis, the Z-axis is first automatically moved to the current 'Safe-Z' height (Safe-Z is set or turned off with [\[VC\]](#)). If the tool has already been moved above Safe-Z, it is left in that position. Then an immediate move to X, Y location 0, 0 at Jog speed is executed. Action starts immediately when the [H] is entered with no pause for parameters occurring. Note that Relative mode does not apply to [\[MH\]](#) or [\[JH\]](#). The tool is always returned to Absolute 0, 0

## M2 - Move 2 Dimensions

*{x-location/distance, y-location/distance}*

Makes an X and Y axis move to the designated location or a specified distance (if in Relative Mode) at current Move or cutting speed. These 2-D interpolated moves are made on a diagonal from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting. If a value is not entered for one of the parameters, the value defaults to the current location assuming a comma is used as a spacer to designate the correct parameter field (this would be the same as 'modal' in G-code programming).

## M3 - Move 3 Dimensions

*{x-location/distance, y-location/distance, z-location/distance}*

Makes an X, Y, and Z axis move to the designated location or a specified distance (if in Relative Mode) at current Move or cutting speed. These 3-D interpolated moves are made on a diagonal from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting. If a value is not entered for one of the parameters, the value defaults to the current location assuming a comma is used as a spacer to designate the correct parameter field.

## M4 - Move 4 Dimensions

*{x-location/distance, y-location/distance, z-location/distance, a-location/distance}*

Makes an X, Y, Z, and A axis move to the designated location or a specified distance (if in Relative Mode) at current Move or cutting speed. These 3-D interpolated moves are made on a diagonal from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting. If a value is not entered for one of the parameters, the value defaults to the current location assuming a comma is used as a spacer to designate the correct parameter field.

## M5 - Move 5 Dimensions

*{x-location/distance, y-location/distance, z-location/distance, a-location/distance, b-location/distance}*

Makes an X, Y, Z, A and B axis move to the designated location or a specified distance (if in Relative Mode) at current Move or cutting speed. These 3-D interpolated moves are made on a diagonal from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting. If a value is not entered for one of the parameters, the value defaults to the current location assuming a comma is used as a spacer to designate the correct parameter field.

## MA - Move A axis

*a-location/distance*

Makes an A axis move to the designated location or a specified distance (if in Relative Mode) at Move speed. These moves are made from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting. The A axis is sometimes used as a rotational axis. When this is done, the location/distance value may be in degrees or other arbitrary units as defined by its Unit values with [\[VU\]](#).

## MB - Move B axis

*b-location/distance*

Makes a B axis move to the designated location or a specified distance (if in Relative Mode) at Move speed. These moves are made from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting. The B axis is sometimes used as a rotational axis. When this is done, the location/distance value may be in degrees or other arbitrary units as defined by its Unit values with [\[VU\]](#).

## MD - Move Distance Move by specifying Distance and Angle

*length of move, angle {,z-axis change}*

### **Length**

Make a move that is defined by the *length* of the move and the direction that it is made in, at current Move or cutting speed. The *length* is the distance the move will make, starting at the end point of the last move. The *angle* can be specified in several ways...

### **Angle**

An Absolute angle can be entered in degrees and the move will be made in that direction using ShopBot's angle system (e.g. 0 degrees at the top of a circle when viewed from front of tool).

A Relative angle can be entered in degrees with an "R" or "L" after the angle to determine if the angle is made to the Right or Left, based on the direction of the last move. For instance, if an angle of "90R" is entered in the second parameter, that move will be at 90 degrees to the right (clockwise) of the direction of the last move. If this is the first move and there is no previous segment, the "R" and "L" are ignored.

An Absolute angle can be entered in Clock notation, as if you were looking down at the hour hand of a clock face. The angle is entered as hours and minutes, separated by a semicolon ":". To move 90 degrees (

parallel to the X axis in a positive direction) you would enter "3:00" for the angle, or to move 45 degrees you would enter "1:30".

An *Absolute angle* can be entered in *Compass notation* as "N", "S", "E", or "W", using up to 3 compass direction values together. To move parallel to the X axis in a positive direction you would enter "E" for the angle, and to move at 202.5 degrees you would enter "SSW".

### Z-axis change

The third parameter is the *Z-axis change*. If only a number is entered the Z-axis move is absolute and the tool will move to that Z-axis location. A Z-axis move that's relative to the current location can be specified by adding a "U" after the number to move up that amount, or "D" to move down by that amount.

## MH - Move Home

Moves the tool to the home (0,0) XY position at Move speed. If the tool is positioned at a plunge location below that defined as the Safe-Z height (default .15) on the Z Axis, the Z-axis is first automatically moved to the current 'Safe-Z' height (Safe-Z is set or turned off with [\[VC\]](#)). If the tool has already been moved above Safe-Z, it is left in that position. Then an immediate move to X, Y location 0, 0 at move speed is executed. Action starts immediately when the [\[H\]](#) is entered with no pause for parameters occurring. Note that Relative mode does not apply to [\[MH\]](#) or [\[JH\]](#). The tool is always returned to Absolute 0, 0.

## MO - Motors Off

This command immediately turns off power to all motors on PRT tools and turns off all output switches. (Either a numeric zero or the letter 'O' can be used for the second letter of this command.) **[MO] does not turn off motor Power on PRTalpha or PRS Tools.**

The power cut-off occurs to outputs and motors occurs when the second letter of the command is entered. There is no pause for a parameter. You can use this command to de-power your stepper motors on PRT tools.

**IMPORTANT:** Because the stepper motors, when rotated manually become very powerful generators, it is possible to send large electrical pulses to the control box by moving the carriages by hand. There is protection circuitry in the box, but the pulses can be very difficult to manage. Moving the carriages by hand is thus a bad idea. You can almost always do your positioning with the arrow keys. If it is absolutely necessary to move the tool by hand, only move it for small distances and at slow speed, or disconnect the power wire to the motors that are being moved.

## MS - Move Speed Set

*{xy-Speed, z-Speed, a-Speed, b-Speed}*

Sets the speed in inches or mm per second for Move Speed movement in the XY, Z, A and B axes. The new values will appear on the left side of the control console. This speed represents the actual movement speed (feed rate) of the cutter tip. Because the vector speed along a diagonal is the geometric addition of the X and Y vectors, for any given speed, the motors will operate more slowly on a diagonal move than on a straight X or Y move, even though the vector speed of the cutter is accurate. The change you notice in the sound of the motors is as it should be. For a circle or arc the speed of the motors will continuously change up and down in order to achieve a constant vectored movement speed around the circle or arc.

This command is similar to the [\[JS\]](#) command for changing Jog Speed and both of these commands are subsets of the [\[VS; Values Speed\]](#) command, which lets you set all speeds at the same time.

SEE ALSO: [\[VR\]](#) to set the acceleration ramps and deceleration ramps for each type of movement

SEE ALSO: [\[VU\]](#) for setting extremely slow speeds

## MX - Move X axis

*x-location/distance*

Make an X axis move to the designated location or a specified distance (if in Relative Mode) at Move speed. These moves are made from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting.

## MY - Move Y axis

*y-location/distance*

Make a Y axis move to the designated location or a specified distance (if in Relative Mode) at Move speed. These moves are made from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting.

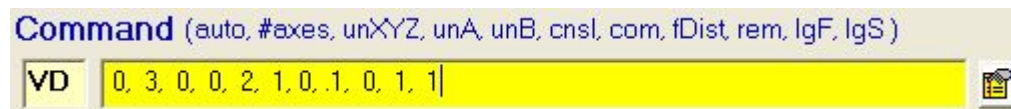
## MZ - Move Z axis

*z-location/distance*

Make a Z axis move to the designated location or a specified distance (if in Relative Mode) at Move speed. These moves are made from the current location to the end location indicated in the location/distance values, which may be absolute or relative depending on the Move Mode setting.

## Parameter Line Sample

This is a sample Parameter Line entry for a relatively complex command as it would appear if you entered the parameters without using a Fill-In Sheet.



Note that the order of the parameters is displayed in an abbreviated fashion above the line to help make entering the values easier (this is the same order documented here for each command). By clicking the Fill-In icon at the right you can also access the Fill-In Sheet for the command.

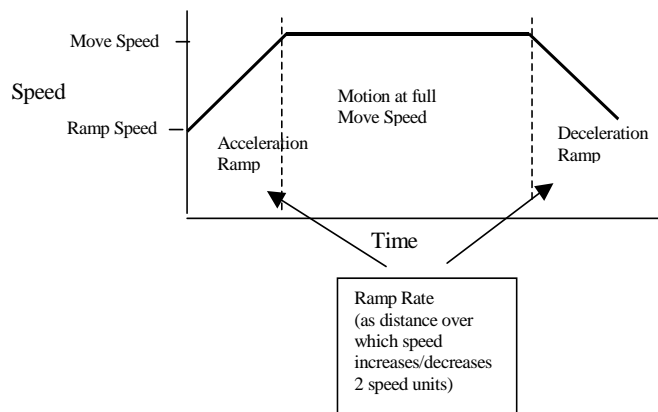
Hit ENTER when you are finished entering the parameter data

## Ramping - Detailed Explanation

By “ramping” we mean the acceleration up to speed and deceleration down to a stop in tool moves. In the same way that a car gradually accelerates to speed and decelerates, so your tool ramps up to speed and ramps down. Listen to your tool while it makes a simple MX move, and you will hear the ramping at the start and stop. It is not possible to instantly be at speed without producing abrupt, vibration-inducing motion. Thus the goal of ramping is to smoothly but quickly accelerate your tool to cutting or jogging speed and to smoothly and efficiently decelerate it to a stop. Additionally, ramping comes into play when complex movements or tight curves require slowing down to reduce cutting perturbations that would come from violent changes in direction. (Don't confuse this use of “ramping” with the entry ramps that are generated by some types of CAM or design software to lead gradually into a cut. These are 3D inclines in the tool path to help prevent cutter marks at the beginning and ending of the cut.)

### Ramping Basics

At its basic level ShopBot ramping is pretty easy to understand. Consider the speed profile of a simple move in the X axis at 2in/sec.



After the start of a move, speed increases during the ramp period from Ramp Speed (the starting speed) to Move Speed for the axis. Motion continues at Move Speed. Then at the end of the move, during the deceleration ramp period, speed decreases from Move Speed to Ramp Speed at which point motion stops. ShopBot's ramps are linear accelerations and decelerations. That makes the overall speed profile of a move a symmetrical trapezoid in time.

What follows are the details of how ramping can be modified, but please understand that the default ramp settings for your tool will provide optimal ramping for most situations. Special cutting situations or efficiency considerations may encourage you to experiment with altered settings, but you can always revert to the default values for relatively smooth operation if your changes to settings do not produce useful results.

There are many parameters for the ramping in the [VR] Command. However, the start and stop of action and its smoothness is primarily controlled by just 2 of these values. You can adjust the Ramp Speed (the start and end speed of the move) so that the beginning and ending speed are higher or lower. Or you can adjust the Rate of the acceleration or deceleration by increasing or decreasing the period over which the ramp and thus the steepness of the ramp.

*Working with Ramp Speed.* The most efficient starts and ends of motion are achieved by having your tool begin moving at a low but distinct starting and ending speed. Because the nature of stepper motors is inherently digital, you can reach a point of diminishing returns by setting the start speed too low. Practically speaking, it is usually not useful to set Ramp Speed below .2 in/sec. On the other hand, setting the start speed too high, in order to try and get to fast motion more quickly, can cause problems if the start speed initiates movement too abruptly and causes vibration, or worse, leads to lost steps because the motors cannot jump to speed quickly enough. Lost steps will not be a problem with closed-loop PRTalpha tools, but

even for a PRTalpha setting Ramp Speed too high will still produce cut-jarring starts. Ramps Speeds of over 1.2 in/sec are likely to cause problems on both PRT and PRTalpha tools.

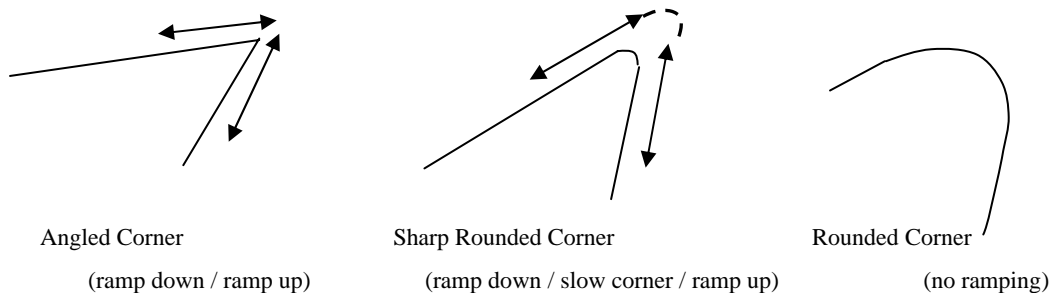
Ramp Speed also defines the speed of the slowest moves your tool makes. For example, if your tool is moving through a series of very short back and forth cut segments, the look-ahead-ramping feature of the software will have determined that there is not enough length in the moves to accelerate to speed and will execute the move at a continuous slow speed rather than wavering up and down. The reduced speed will be the Ramp Speed. The Ramp Speed also comes into play for very small arc segments or circles. If a circle or arc has a radius smaller than the "Small circle size" set in [VU] then it will be executed at Ramp Speed. The look-ahead function is described in more detail below. But in general you can think in terms of Ramp Speed defining the lowest operational speed for intricate moves.

*Working with Ramp Rate.* Ramp Rate sets a distance for acceleration and deceleration ramps. A lower value means a shorter distance and thus faster accelerations and decelerations. A higher value means longer ramps and generally smoother accelerations and decelerations. Specifically the Ramp Rate value means: the distance (in units; inches or mm) that it takes for the tool to increase or decrease 2 units of speed (inches or mm depending on your tool setting).

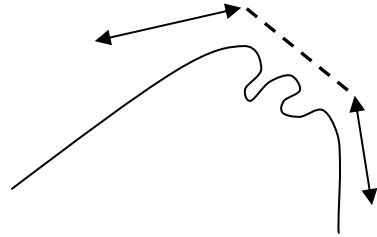
### When Does ShopBot Ramp?

ShopBot always ramps at the beginning and end of every isolated, individual move you ask the tool to make, assuming that your Move Speed is higher than the Ramp Speed. ShopBot also ramps up and down for every Jog, including Jogs in Part Files. But ramping can actually have considerable complexity because the process of 2D or 3D machining can involve very complex tool paths that require speed variation in order to produce the most efficient cutting while avoiding the violent jarring motion that comes from changes in direction that are too sharp or abrupt. It is this process of creating a fast but smooth cutting motion that is provided by the "look-ahead" ramping system.

The *look-ahead ramping system* analyses all upcoming moves in a tool path in order to determine if changes in direction or the tightness of corners is so sharp that it cannot be handled at the full move speed. If there are corners or intricate portions of cutting that require slower cutting, ramps are computed that will accelerate into and out of these regions. If the region contains many short intricate moves with changing direction, then the segment may be executed at a slower continuous speed in order to produce a smooth cut and reduce potential cutter chatter.



To make the picture slightly more complex, the *reaction to upcoming corners or curves is graded*. A sharp corner produces a full ramp with deceleration down to the Ramp Speed. But a more modest corner angle reduces speed only slightly. ShopBot defines angles sharper than  $43^\circ$  [.75 radians] (or, as a curved change of  $43^\circ$  with a radius of less than .25in) as those requiring a full ramp down, no matter what the speed. Less sharp corners will also produce a reduction in speed, but the angle at which these reductions occur is graded depending on the current speed of the tool. At 10 in/sec ramping will occur with any angle of over about  $5^\circ$ , but at slower speeds the criterion angle increases, so that at 1 in/sec there is no ramping unless the full ramp angle criterion of  $43^\circ$  is reached. The amount of speed reduction is controlled by the "Slow Corner Speed" parameter in the [VR] Command. By default this is set to 65%, which means that when look-ahead ramping determines that the slight ramp criterion is reached, speed will be reduced by 65%. Increasing this value will decrease the amount of ramping that occurs on gentle corners. Decreasing the value will cause more pronounced ramping on gentle corners.



(ramp down / slow speed / ramp up)

There is one additional parameter that comes into play for ramping, though we do not expect that you will ever need to change this one. The parameter is called “Minimum Distance to Include”. It is the amount of motion that is included in each check for angular change. To understand this one, imagine that you have a curve made up of many tiny line segments. The angular change between any small segment may not be significant, but the angular change over several segments can amount to a change that should produce ramping. The Minimum Distance to Include indicates that change over a given distance should be considered for deciding ramping, by default the distance is 0.15 inch. The default parameter should work for just about all situations. Increasing the distance will cause ramping to occur more frequently, because more distance is considered for the change in direction, decreasing the distance value will make ramping less likely to occur and can produce bumpiness in intricate files.

### Ramping in 3D

3D ramping action for the vertical (Z) axis is handled slightly differently than that described so far because the criteria for ramping decisions differs. Cutter forces in lateral movement are very different than for vertical movement because of the side loads in generated in cutting. This means that in general and whenever possible, we want to maintain smooth movement in X and Y axes of cutting, with vertical motion making accommodation. For this reason, ramping criteria are based on angular change in direction for the X and Y axis, but on changes in motor speed for the Z axis (and A and B). The Z axis does its best to accommodate to the action in the cutting plane. There are two situations where movement in the Z constrains the XY action. First, if the Z (or A or B axis) would be required to move faster than its defined Move Speed, the speed of the X and Y axis will be reduced to so that this does not happen. For this reason, you need to make sure that your Z Speed is not set too low for normal XY movement in 3D files. Second, the changes in the motion of the Z (or A or B) axis will activate a ramp if the change is greater than the Z Ramp Speed. For example, imagine that your Z Ramp Speed is set to .4 and, there is a 3D move then causes the Z to need to change speed from 2 to 1.3 in/sec. In this case, since the difference is more than .4, ramping will occur into and out of this move. The sensitivity of the 3D ramping can be adjusted with the “3D Threshold” parameter in [VR]. Increasing the percentage means that the threshold will increase and 3D ramping will be less likely. Decreasing the threshold will sensitize 3D ramping and make it more likely.

## RA - Record Activate

Activates the Record function. Recording function on is indicated by a blue dot to the left of the replay window on the Control Console. . Default is on.

Also see: [\[RI\]](#), [\[RZ\]](#)

## RI - Record Inactivate

Inactivates the Record function. Recording function off is indicated by an open ‘o’ to the left of the replay window on the Control Console.

RI does not erase stored commands. It is possible to record a few commands, then turn off recording to make moves or adjustments you do not want to record, and then re-activate recording.

Also see: [\[RA\]](#), [\[RZ\]](#)

## RP - Record Play all Commands in Storage

Immediately replays all commands in storage. This is most useful if you have recently reset [\[RZ\]](#) the command memory and just carried out a couple of specific moves that you want to repeat as entered.

Without the reset, you will replay all the commands that you have entered since the beginning of the session or since the last reset. [\[RR\]](#) and [\[RS\]](#) provide more flexibility in replaying selected commands from the list of commands that are currently stored.

Also see: [\[RA\]](#), [\[RZ\]](#), [\[RR\]](#), [\[RS\]](#)

## RR - Record Replay of the Last Commands

*{number of commands back, repetitions}*

This command allows you to selectively replay the most recent commands in storage, up to the last 100.

### **Number of commands back**

How far back in memory you want to play from. The Fill-In Sheet for this command allows you to inspect the list. The default number back is 1 -- the most recent command will be replayed. Use 'A' to indicate all commands in memory (this would be functionally the same as [\[RP\]](#)). See related commands for instructions on how to clear memory and to start and stop recording. Commands will be replayed with the parameters as they were entered.

### **Repetitions**

The number of times the designated group of previous commands is replayed.

Also see: [\[RA\]](#), [\[RI\]](#), [\[RZ\]](#), [\[RS\]](#), [\[RP\]](#)

## RS - Record Save Stored Commands to a File and Optionally Edit

*{filename, Number of commands back (blank=all)}*

Saves the most recent commands to a file in the current Part directory, up to the last one hundred. When the command is entered, the file is created and opened in the editor for inspection or modification.

### **Filename**

The name of the file to save to. If no extension is given, the file will receive the extension .sbp (i.e. a standard PART file extension). You can override this action by entering a different extension. If no filename is given, the file will be saved to SBTEMP.SBP.

### **Number of commands back**

The number of recently entered commands to Save. A blank will save all commands in memory.

Also see: [\[RA\]](#) start recording, [\[RI\]](#) stop recording, [\[RZ\]](#) clear memory, [\[RS\]](#), [\[RP\]](#)

## RZ - Record Zero the Memory

Resets the Record memory of all previous commands and starts recording over again. A typical Record sequence might involve first resetting memory with [\[RZ\]](#), then making sure that Record is active with [\[RA\]](#). You would enter the commands you want to record and then might use [\[RI\]](#) to make sure no further commands are added to the list. You could then replay the commands in memory [\[RP\]](#) or [\[RR\]](#) or save them to a file for replaying later or for editing [\[RS\]](#).

## SA - Set to Absolute Distances

This command sets your tool for Absolute Distance. All distances entered for moves will cause tool to move to the specified location measured from the 0, 0 point. In this mode you direct the tool as if it is on an XYZ graph. If you send the tool to location 3, 7 it will go to 3 inches (or mm) from 0 on the X axis and 7 inches from 0 on the Y axis from where ever it is. Once at this location, if you tell it to go to location 3, 7 it will not move, because it is already there. Your tool is always set to either Absolute or Relative distances.

We believe that for most purposes it is best to use Absolute Distances for Part Files. However, there are special situations where Relative Distances can be very useful, see [\[SR\]](#). The current Distance mode is displayed as a toggle switch in the Status Panel in the upper right portion of the display.

See also: [\[SR\]](#)



## SF - Sets File and Move Limit Checking

0-OFF; 1-ON

This command sets up 'Limit Checking'. With Limit Checking ON, before any physical movement the ShopBot Control Software will first verify that every move your tool is instructed to make is within the limits set for your tool. If a move would take the tool beyond its limits, you will be alerted to the problem and given a choice on how to proceed. Limit Checking is done in with reference to the Base Coordinate Values of the tool and not the current 'working' coordinates. Use the [\[ZT\]](#) command to set the location of the Table Base Coordinate System zero points.

Limit Checking may be turned ON or OFF using the [\[SF\]](#) command inside a Part File. However, at the time the Part File execution ends, Limit Checking is set to the same status it was at the time the Part File execution began. Limit Checking must be changed with a keyboard instruction in order to produce a persisting change. Limit Checking may slow file reading on slower computers. In such cases you may want to just use it for verifying files and then turn it off for normal operation.

See also: [\[VL\]](#), [\[ZT\]](#), [\[ST\]](#)

## SI - Send Command Line(s)

Line #1, Line #2, etc ...

[shortcut <i>]

This command creates a scratch pad for conveniently sending one or more lines of code from the keyboard without needing to create a Part File. Enter the each Command (with parameters) on a line. Then click OK when you are ready to send/run them.

The Lines can be replayed by double clicking on the Command in the 'Recorded Command List' or by entering the [\[SI\]](#) Command again and hitting Alt-L to bring up the last lines entered, at which point you can edit them, and then re-send. The commands are also saved in a Part File with the name temp\_code.sbp, which you can find in the current Part File folder.

The Command Lines can be ShopBot Part File Commands and parameters, or they can be G-code lines. Code types can be intermixed.

## SK - Set to Keyboard Arrows to Move Tool

[shortcut: <k> ]

This command activates the Arrow Key movement control that allows moving your tool around from the keyboard. We refer to this as the KeyPad mode. The KeyPad mode only works in MOVE/CUT Mode. It does not function in PREVIEW. As indicated in the screen display, the left and right arrows move the tool in the X axis while the up and down arrows move the tool in the Y axis. PageUp and PageDn move the Z axis. You can use either the keyboard or click on the keypad image with your mouse.

If you hold the <Ctrl> key down while using the arrow keys, the tool will move at Jog Speed. Holding the <Alt> key will cause the Accessory axis rather than the Z-axis to function with the <PageUp> and <PageDn> keys. The </> key moves the carriage on a +X,+Y diagonal and the <\> moves the carriage on a +Y,-X diagonal. Holding down the <Alt> key with one of the diagonal keys will move the carriage in the reverse direction. You can get the same diagonal moves by clicking on the sweet spot between two arrows on the KeyPad on the screen.

Several additional functions are provided. If you hit a <D> (upper or lower case) on your keyboard or click on the "fixed" button in the keypad window the tool will move a fixed distance. This distance is set in the Fill-In sheet by typing "VD" and changing the "KeyPad Fixed Distance" value.

You can also save a particular location as a move in the command Recorder. For example to save the current location as a J2 move, hit a < 2 >. In this way, you can move the tool around and record the

locations that you want to replay as a set of moves. Use [\[RR\]](#) to replay the commands or [\[RS\]](#) to save/edit them.

When in the KeyPad mode you can activate a remote control pendant, either your mouse, or the ShopBot Remote, by hitting the letter <e>. With the pendant active, hitting the left button causes the tool to move down an axis. Hitting the right button causes it to move up an axis. And, hitting the middle button switches between the active axis (active axis indicated on the KeyPad screen).

The keyboard arrow function can be called from within a Part File using the [\[SK\]](#) command. When the operator exits the arrow key display by hitting the < ESC > key, Part File execution resumes.

See also: [\[RR\]](#), [\[RS\]](#)

## SL - Set to Clear all Variables in Memory

This command clears all variables in memory. It deletes the variables as well as removing any values associated with them. The Control Software will retain up to 100 variables. These are defined with the [\[VV\]](#) command.

See also: [\[VV\]](#), [\[UL\]](#)

## SM - Set to Move/Cut Mode

This command sets your ShopBot to Move and Cut – as opposed to Preview – moves. All commands will be executed by your ShopBot. If your ShopBot is not attached or the control box is not on, your ShopBot Control Software will tell you that you are not connected to the tool. In this case, if you want to test a move or the commands in a file, just switch to Preview Mode. You are always in either Preview or Move/Cut Mode. The Mode setting is displayed in the Status Panel which is the upper right portion of the display.

See also: [\[SP\]](#)

## SO - Set Output Switch

*switch number, [0-OFF or 1-ON]*

This command turns OFF or ON a specified output switch. There are eight switches numbered 1-8. [\[MO\]](#) turns off all output switches as well as motors (as explained above). See the Users Guide for information on hooking up and using the output switch circuits. The [\[SO\]](#) command can be used from the keyboard or from within files. The condition of switches is displayed in the setting panel in the upper right portion of the settings display. Switches are all turned OFF when you leave the ShopBot Control program and the state of switches is not saved or available when the software is re-started.

### **Switch number**

is the number from 1 to 8 of the switch you want to turn on or off. Special Cases: Switch number 4 (default = 4,-1) is automatically activated when the tool is in motion. It is set to ON at the time the Warning Signal (see [\[SW\]](#)) begins to flash and beep, and in this way could be used to activate a relay that controls other processes, such as external warning signals or accessories such as dust collectors. This function for Switch 4 can be adjusted or disabled. If switch 4 is set to -1, the switch will come on with the Warning Signal just before movement begins and go off when the warning goes off at the end of the move; if set to -2 it will also go off if the PANIC Bar is hit; if set to -3 it will go off if the PANIC bar is hit or if the tool 'Stalls' (cable drive only); if set to -4 this function will be disabled so that the switch can be used for other purposes.

### **0-OFF; 1-ON**

Sets the condition of the switch. Entering no value turns the switch off.

See also: [\[MO\]](#), [\[SW\]](#), Part File Programming Instruction ON INPUT

## SP - Set to Preview Mode

After this command is executed, moves and file execution will be displayed on the graphical Preview Screen. If you are already in Preview Mode, using the [\[SP\]](#) command will clear the preview display.

Note that when in Preview Mode, the location of the tool will be tracked while you are executing within a file, but because the physical location of your ShopBot has not changed, when you leave Preview Mode the

location read-outs on the main display will be returned to their initial settings. This means that, unless you specify otherwise, the first move in Preview Mode will be from the current physical location of your tool and then all subsequent moves will 'track' your location changes in Preview Mode.

If, after executing a file or move and returning to the main screen, you want to have another look at the preview screen you can use the space-bar to toggle between the main screen and the view of the Preview Screen.)

See also: [\[SM\]](#), [\[VP\]](#)

## SR - Set to Relative Distances

This command sets your tool to Relative Distances. All distances entered for moves will move the tool the specified distance from its current location. If you send the tool to 3, 7 in Relative Distance, it will go 3 inches (or mm) in the positive X direction *from where it is*, and 7 inches in the positive Y direction *from where it is* (i.e. it is as if your current position is temporarily taken to be 0, 0). Once at this new location, if you tell it to go to 3, 7 while still in Relative Distance it will again move 3 inches in X and 7 inches in Y because it makes a move Relative to its current position. (The tool always is set either to absolute or relative distances.)

Relative Distance is handy when doing work from the keyboard. For example, in Relative Distance you might produce a repeated drilling pattern by first drilling the hole, then advancing say 1.5 inches in the X direction, then simply using RR to repeat the commands as many times as you wanted holes. Brackets are used here to indicate these are commands entered from the keyboard, not in a file. Explanatory comments follow the apostrophe.

```
[SA]          'in absolute distance
[M3] 1,1,.2   'move to the start point
[SR]          'set to Relative Distance
[RZ]          'clear the Command Recorder of all earlier commands
[MZ] -.45     'plunge to .25 inches below surface 0, we are going relative from
.2
[JZ] .45      'pull back up to where we started
[JX] 1.5      'move to next drilling position 1.5 inches up X axis
[RI]          'turn off recording ... this does not erase, just prevents adding
commands
[RR] A, 20    'OK, repeat 20 times to drill another 20 holes that are 1.5 inches
apart
```

Relative Distance can be handy when you're operating your tool from the keyboard, but we don't recommend using Relative Distance in Part Files for two reasons: 1) rounding errors can incrementally add up to unexpected errors in large files; 2) it is hard to look at the numbers in a Relative Distance file and know where you are in the file because none of the numbers represent real locations on the cutting table. The current Distance mode is displayed as a toggle switch in the Status Panel in the upper right portion of the display.

See also: [\[SA\]](#)

## ST - Set Location to Table Base Coordinates

ShopBot maintains a set of reference or Table Base Coordinates. At any time, the current working coordinates can be reset to what they are in terms of the base values using this [\[ST\]](#) command. The location of the Base Coordinate zeros are initially defined using the [\[ZT\]](#) command. Having base coordinates provides a system for absolute location vs working location. The Base Coordinate system also provides the information for software limit checking because it maintains an absolute relationship to the physical table.

Note, that as with the working coordinate locations, if the tool is moved manually rather than with the software, this absolute location will be lost.

See also: [\[ZT\]](#), [\[VA\]](#) and various [Z] functions

## SW - Set Warning Duration

*duration (in seconds)*

Sets the duration, in seconds, of the visual and audible warnings before any movement starts. The warning is a simulated blinking light at the top of the screen accompanied by a beeping (if your speaker is connected). Entering a 0 parameter will turn off the pre-move warning. The default warning is 2 seconds. Note that by default the output switch #4 is automatically turned on at the same time as the warning display (see [S0]) and could be used to activate a more extensive warning display. If Warning duration is set to 0, the output switch comes on when movement starts (along with the blinking light). The warning and switch remain on until the move is complete, or until a control command in a Part File turns them off.

See also: [\[S0\]](#)

## Virtual Tools

Virtual Tools are highly friendly interfaces for many standard workshop activities. These are add-in functions and will change with each issue of the software. New and/or custom Virtual Tools can be user installed. All Virtual Tool commands begin the letter T. Virtual Tools will make your ShopBot appear as a more familiar tool and allow you to control it in a more intuitive manner than from the basic interface. For example, optional software may make your ShopBot appear as a tool such as a drill press or miter saw, allowing you to set it up for a task more like you would a physical tool (though much more efficiently). As another example, a Virtual Tool might be a system for producing a particular type of project such as shelving or signs. We view such enhancements to your robotic capabilities as one of the greatest potentials for ShopBot, and hope not only to develop such enhancements ourselves but to encourage 3rd party developers as well. The Virtual Tools listed here are the ones that are typically installed.

## TC - Tools Copy Machine

Copy Machine is a Virtual Tool that allows you to use your ShopBot and a digitizing probe to copy an existing piece and generate either a cutting file or a file that you can import into a CAD program for further editing.

Once the Copy Machine opens there are 3 choices that need to be made before you can begin the probing process to create your file. On the Pattern tab you have to select either 2D (trace the outline of an object) or 3D (probe the entire object). On the Probe tab you have to select the type of probe that you are going to be using - normally closed probe such as the ShopBot probe or a normally open probe. On the last tab you have to select from 3 different types of output file formats. You can go directly to an .sbp cutting file, or you can choose either a .dxf set of points or a .dxf polyline. The latter two will allow you to import your file into a CAD program.

## TD - Tools Drill Press

The Drill Press allows you to easily define a hole or series of holes to drill in your material. At the top of the settings tab you define the size of your blank. Next you select your bit diameter, safe Z pull up and the number of passes you want to make. Below this there is a checkbox for pocketing. Check this box if you want the toolpath to pocket from inside to outside. This would be checked in situations where the waste piece is large and you don't want it to go flying once it is cut free. You can get more information by clicking on the ShopBot icon with the question mark below the settings tab. In the bottom section you define whether the Z-Zero position is at the material surface or the table top and the last choice is units.

## TF - Tools Forney Fluter

The Forney Fluter allows you to set up the ShopBot to cut flutes by entering a few parameters. When you open this virtual tool the first tab is the Setup tab where you define your table size, the units you will be using (millimeters or inches) and whether you will be cutting the flutes in one or two passes. You can get more information by clicking on the ShopBot icon directly above the Setup tab This tab also allows for control of the zoom level of the preview screen by clicking on the + or - at the bottom right.

The next tab is the Blank tab which allows you define the size of your blank, its position on the table, the placement of the flutes on the blank and whether you want the panel and flutes to run vertical or horizontal.

The third and final tab Flutes is for defining parameters for the flutes themselves. Here is where you specify the width of the groove, the type of bit and its diameter or degrees if you are using a V-bit. The bottom section of this tab allows for specifying the distance between the flutes and the distance from the edge of the panel to the first flute.

Finally click on "I'm done....make my file", and then you can choose to preview it, cut it or exit the software.

## TH - Tools Header Writer

The Header Writer is a handy tool for including information at the beginning of your cutting file. This information can be viewed by opening the file in a text editor or by clicking on the yellow exclamation point on the Fill-In sheet when you load the part file. You can include things such as contact information and use restrictions on the General Information tab. You can also call up files to run at the beginning or end of the program using the FP command. The Part File and Material Information tab is where you record information about the size of you blank, type of material, bit size and type, units being used and whether the file should be zeroed to the top of the material or the tabletop. There is also a section for noting any offset that may have been used in the file and a box for any additional notes that you may want to add.

Once you have filled in all of the information click on the green "Write the information" button. In the box that pops up select the file that you want to write the header to and click "Open" and then "Yes" to write the header.

## TI - Tools Indexer

## TS - Tools ShopBot Setup

The Setup Virtual Tool enables t to enter some general information about your ShopBot that may be helpful in the event that you require technical support. This tool is self explanatory and you simply enter the information when prompted to do so. In the last section you are given the opportunity to set up and calibrate zeroing on all three axes. Your options are dependant on whether or not you have a zero plate and/or proximity switches.

## TT - Tools Typesetter

Type Setter enables you to V-carve certain fonts that are on your computer.

The important thing to realize here is that you will not be able to successfully v-carve every font that is in your Windows font folder. You will have to go through a trial and error method to determine which fonts work and which ones don't. For a more detailed explanation click on the help icon next to the "File Settings" tab.

## TU -Tools Tabletop Surfacer

The Tabletop Surfacer provides an easy way for you to set up your ShopBot to ensure that your tabletop is parallel to your cutter. Make sure that you have completed the instructions for leveling your Z axis before running the Tabletop Surfacer; otherwise, there will be ridges between each pass that the bit makes. This Virtual Tool is simply a front end to the CR command method of surfacing the tabletop that makes entering the information a little easier. After the information is entered and you follow the prompts you will see the file you created in preview mode. The file is saved in the Sbparts folder and you can now run it whenever needed.

## UD – Utilities Diagnostic Display

This utility displays information that can be helpful in diagnosing problems with the operation of your tool. Values from this display will be of interest to ShopBot Technical Support in solving a problem you may be having. The information provided by [\[UD\]](#) and [\[UV\]](#) is saved in a file called PROBLEM.LOG and should be e-mailed to ShopBot if you are contacting Tech Support for help with any problem.

## UI - Utilities Install Control Box Firmware

Use this command anytime that you need to install or update the firmware in the control box. Follow the prompts and wait until you receive a message that says the load was successful.

## UL – Utilities List Variables

This command displays in the editor all currently defined variables and their values. The variable may be edited and saved and new variables may be added. Remember that all variables begin with the ampersand '&' character. In this variable listing they are assigned using the format: &var = value

Variables may also be defined from the main screen using the [VV] command.  
See also: [VV], [\[SL\]](#)

## UN - Utilities Name of Editor Program

*path and filename*

With [\[UN\]](#) you can designate the program that will be used to edit files. This program should be an editor or word processor that can save files as text files (the normal file extension for such file is .txt though for Part Files we use .sbp). Unless you have a strong preference for another editor, we recommend using the default editor.

If you are running your ShopBot from WindowsXP/98/ME, you may find it convenient to also use NOTEPAD to edit your Part Files. This is made particularly easy if you "associate" .SBP files with NOTEPAD so that by just double-clicking on them they will open. (The first time you double-click the file you will be given the option to "Open with" a choice of programs. Pick Notepad from this list and the association will be created.) The default editor used by ShopBot is Sbedit.exe.

See also: [\[FE\]](#)

## UR – Reset software to Default Settings, Load a Custom Settings File, or Clear the System Log

The [UR] Command allows you to Reset all the major settings in the ShopBot software to the specific values recommended as defaults for your tool.

After starting the command, you will be given the option of deleting the System.Log file. The system log is a record of all the files that have been run on your ShopBot and can be inspected to see what has recently been done with the tool. If the file get's excessively large, it is sometimes helpful to delete it.

Then, you will be given the option of loading a Settings File. The settings folder contains files with default settings for all standard sizes of ShopBot. Just select the file that is appropriate for your tool and all settings in the software will be returned to appropriate default conditions.

If you do not have a standard ShopBot (e.g. unusual gearing or size), pick the file that is closest in type and size to your tool, and then modify the necessary specific settings after you've loaded the defaults.

You can use the related [US] Command to save a Custom Settings File with your current settings. So, at any time, you may make a copy of your current set up. This saves your custom settings in a new file in the settings folder. After you've saved one or more custom settings files, you can use the [UR] Command to load any of the custom settings files and re-instate the settings you had in effect when you saved it.

See also: [\[US\]](#), [\[VU\]](#), [\[VS\]](#), etc...

## US – Save Current Settings to a Default Settings File

The [US] Command allows you to save the current setting of all the major values in the ShopBot software to a Custom Setting File. This settings file can be loaded at anytime using [UR] to re-instate the settings in effect at the time it was saved.

[US] provides a convenient way to save settings for special tools or setups or special types of projects.

See also: [\[UR\]](#), [\[VU\]](#), [\[VS\]](#), etc...

## UU - Utilities Calculator

This command opens the ShopBot Calculator. This lets you do a quick calculation or make sure that the parameter line calculator (see below) supports a function that you wish to use. F10 can also be used to activate the Calculator at any time and allows the pasting of values into a parameter list. Speaking of calculators, you'll notice that the ShopBot PARAMETER line is also an automatic calculator. When entering parameters, you may enter simple values like 1.5, or you may enter a complex calculation as a formula, e.g. '1.5 \* SQR(4.2)/(23-12.23)'. After you've entered the formula, the Control Software will calculate the result and automatically substitute the value in the command being executed. The calculator supports most mathematical and trigonometric functions. {\*\*\* BUT NOTE, that the Calculator expects angles to be expressed in radians and not degrees. To convert to radians, multiply degrees by .017453; to convert to degrees, multiply radians by 57.2958.} Having a calculated parameter is useful in many situations. It also means that you can easily give your ShopBot fractional locations for moves. For example, to move to 12 and 11/16th inches, just enter the distance as 12+11/16. The 11/16 will automatically be converted to a decimal value before the move is executed. (Note, though, that for negative RELATIVE moves or moves below zero, you would use 2 negatives [e.g., '-2-7/32'] or put the value in parentheses [e.g. -(2+7/32)] to accomplish the fractional move).

Both the Utility Calculator and the PARAMETER line calculator will utilize variables that have been defined during the session. Calculator functions can also be called in PART files.

## UV - Utilities Values and Settings Display

Displays current values and settings that are not displayed in the White Settings Panel of the main screen. Current values for most parameters are also displayed on the Fill-In Sheets when their Value command is called up. The information provided by [\[UD\]](#) and [\[UV\]](#) is saved in a file called PROBLEM.LOG and should be e-mailed to ShopBot if you are contacting Tech Support for help with any problem.

See also: [\[VU\]](#), [\[VS\]](#), [\[VR\]](#), [\[VK\]](#) etc...

## UZ - Utilities Zero all Current Locations and Table Base Coordinates.

This command zeros all axes and locations, including table base coordinates. You would typically only use this command if you were not using file checking or if you are using the software only for preview purposes.

## VA - Values for Axis Locations

*{x-location, y-location, z-location, a-location, b-location, tableBase-x-location, tableBase-y-location, tableBase-z-location, tableBase-a-location, tableBase-b-location}*

### **(x,y,z,a or b)-location**

Sets a new location value for any axis. That is, if you use [\[VA\]](#) to set the x-axis location to, say, 5.3, then no matter where your tool is positioned, the location will now be considered to be 5.3 and this is what the location display will be changed to. Like the 'Z'ero commands, this command should be used carefully because you will lose the information about your current position on any of the axes that you change. This function is very convenient, though, and if you've added distance-indicating tapes to your tool you can just type in your current location anytime you need to set-up the tool. You can also use this command to define a new working region of your tool. Note that when you change working coordinates with [\[VA\]](#) you do not change the Table Base Coordinates.

### **The tableBase-coordinate parameters**

This command allows specifically setting the Base coordinates independently of moving the tool. Normally, the Base coordinates would be set using the [\[ZT\]](#) command at the point the tool is at its Base (home, or 0,0) location, and reset using the [\[ST\]](#) command. Having TableBase-location parameters as a component of this command allows their convenient storage in a Part File and in the configuration file SHOPBOT.INI. You will normally never need this feature.

See also: [\[ST\]](#) and [\[Z\]](#) commands.

## VB - Values for Tabbing Feature

*{distance-to-pull-UP, distance-to-next-TAB, size of TAB, distance of leads, 1-ON 0-OFF, number, start}*

Your ShopBot Control Software has a unique feature which can be used to hold parts in position while they are being cut or machined -- parts that might otherwise fly out of position and hit the cutter. In some situations this feature can be used instead of clamps or vacuum hold-downs. We call this feature 'Tabbing' - leaving small tabs of material at the bottom of a cut to hold the part in place. On the last pass for a particular part, these 'tabs' are created by pulling the cutter up slightly for a short distance then returning to cutting depth, and doing this slight pull-up at standard intervals around the perimeter of the part. The tabs are executed as smooth 3-D moves, with the pull-ups and put-back-downs executed on inclines. This smooth 3-D action minimizes tooling marks on the side of the part where the tab is executed. This feature is useful as an alternative to physically clamping the cut-out parts for a project or product, especially when you are making only a few of the parts and want to avoid jiggling-up. It has the disadvantage of leaving you to trim the tabs off the part. [An alternative that can also work is to adjust the depth of your cut so that the cut-out part is held in place by a thin amount of material all around the part. This thin layer can be removed to release the part with a utility, with a sander, or by running the material through a thickness planer. This technique works well with a vacuum hold-down since it leaves the air seal intact and does not damage the vacuum manifold.]

The [\[VB\]](#) command and its parameters define how the tabbing feature works and take effect during the last pass of a cut.

### **distance-to-pull-UP (default = .1)**

Amount the cutter pulls up from cutting depth or the thickness of the TAB.

### **distance-to-next-TAB (def = 4)**

Defines how far apart the tabs are spaced.

### **Size of TAB (def = 1)**

Defines width of TAB

### **Distance of leads (default = .5)**

Specifies the length of the 3-D 'incline' of the cutter pull-up and the put-down. This means you should view the total tab size as the size of the TAB plus the length of the lead-up and lead-down. (Doing so assumes that the final pass is deep enough to otherwise cut through and release the part.) These distances are measured from the center of the cutter, so you will need to take cutter diameter into consideration. The values can be re-defined anytime you want to change the way that a part is held by tabs. Probably, the most practical way to use tabs is to try your part with the default setting, then depending on the nature of the material and the type of cutting you are doing, make slight modifications to optimize the shape of the tab to your application.

The 5th, 6th and 7th parameters of this command are for controlling the activation of tabbing within a file.

### **1-ON 0-OFF**

Can be used to insert the beginning or ending of tab action in a file

### **number**

Specifies how many tabs to insert after an ON (in the case of a specific number of tabs, OFF is not used). If you use the ON function to turn on tabbing within a file, you need to be careful to subsequently turn it OFF (or specify the number of tabs) so that tabbing will work correctly later in the file.

### **dist**

This parameter can be used to specify the distance to the first tab in a series that is turned ON in a file.

See also: [\[FP\]](#) and [\[C.\]](#) commands

## VC - Value for Cutter-related Parameters

*{diameter, -obsolete-, -obsolete-, safe-Z-pull-up, plunge direction, % pocket overlap, safe-A-pull-up}*

This command sets the values for several cutter-related functions including the Safe-Z-pull-up feature.



**Diameter (default = .25)**

The diameter of the current cutter in your tool. This diameter is used with the built-in 'Cut' functions where measurement is based on inside, outside, or true diameter or width (e.g. in [\[CC\]](#) or [\[CRI\]](#)). Cutter diameter can also be accessed from within a Part file using the system variable `%(20)`. The default is .25 inch.

**-obsolete parameters-**

Leave empty as space-holder for parameters no longer used.

**Safe-Z-pull-up (default = 1)**

The height that the Z axis will move up to in several operations intended to pull the bit out of the material and to a safe transit height. These operations include the commands [\[MH\]](#) and [\[JH\]](#) as well as movement block reloads where the bit is withdrawn temporarily from the material. The safe-Z height is also the height the Z axis must be lifted to in order to disable the tabbing function in a file that has several entities to be tabbed.

**Plunge (default = -)**

Sets the direction the tool is moving on a plunge into the material. Use '-' to indicate that the normal plunge direction is negative, use '+' to indicate that a positive value causes a plunge.

**%Pocket overlap (default = 10)**

overlap tells the pocketing feature in [\[CC\]](#), [\[CR\]](#) and related commands, how much overlap should be computed for cutter path when the tool is carrying out 'pocketing functions'.

**Safe-A-pull-up (default = 1)**

The height that the A axis will move up to in several operations intended to pull the bit out of the material and to a safe transit height. These operations include the commands [\[MH\]](#) and [\[JH\]](#).

See also: [\[C..\]](#) commands and [\[MH\]](#), [\[JH\]](#)

## VD - Values for Display Settings

```
{Auto entry, number of axes, display units XYZ, display units A, display units B, show control console, display file comments, keypad fixed distance, keypad remote, write part file log, write system log}
```

Note: The parameters for this command are different in Version 3 than in Version 2. If you are previewing in Version 3, but running the ShopBot in Version 2, read the VD-Values for Display (Version 2) also.

This command sets features of your ShopBot display.

**Auto Entry (default = ON)**

Determines whether the ShopBot accepts the command and starts moving after the first <Enter> (1), or requires a second <Enter> to truly accept the command (0). Turning the Auto Entry Off (0) is more similar to the way the Version 2 software worked and may be a safer mode in schools and other production situations.

**Number of Axes (default = 3)**

Turns on the number of axes currently active in the ShopBot Software. A standard ShopBot with a single Z would have 3 axes displayed (X, Y & Z)

A ShopBot with two Z's set up in independent formation (only one working with the X and Y) at any time would have 4 axes displayed (X, Y, Z & A). Indexers are typically put on axis A or B and so would have 4 or 5 axes visible. See [\[VI\]](#) Values Input for information on switching between the Z and Accessory axes.

**Display Units XYZ (default = inches)**

Sets the X, Y and Z axes to run in inches (0) or millimeters (1). One inch = 25.4mm. The default is in inches, so the values given in the Unit Values for Different Gear Ratios assume that you are working in inches. If you routinely work in mm and do not wish to convert back to inches, multiply the given unit values by the conversion factor (25.4).

Many of the installed sample programs have values that only make sense in inches. Examples include `S_Zzero.sbp` that uses the Z-Zero plate to zero the Z axis and `S_Home.sbp` which uses the proximity switches to zero the X and Y axes. See tutorials in the manual for customizing those programs to work in metric units rather than inches.

When the unit value is changed, all system values are recalculated according to the new unit type and saved. That means that locations, calibrations, tool sizes, speeds, ramps and so on will automatically be converted to the correct value in the alternate unit system. If you are in inches, all other measurement values need to be entered in inches. If you are in mm, all other values need to be entered in mm.

**Display Units A Axis (default = inches)**

Sets the Accessory Axis (A) (the fourth axis) to run in inches (0), mm (1) or degrees (2). An indexer would run in degrees of rotation rather than inches or mm.

**Display Units B Axis (default = degrees)**

Sets the Accessory Axis (B) (the fifth axis) to run in inches (0), mm (1) or degrees (2). An indexer would run in degrees of rotation rather than inches or mm.

**Show Control Console (default = ON)**

Makes this and other Fill-In sheets available. When automatic Fill-Ins are deactivated by setting this parameter to 0, you may reinstate the Command Console by clicking on the console icon on the red position and screen. If you are using Fill-Ins, they can be skipped at any time by just hitting <Esc> to return to the parameter line, or you can hit <Enter> at any time to paste the values into the parameter line. You can also edit values in the parameter line even if they have initially been transferred from the Fill-In Sheet.

See comments on [Fill-In Sheet](#) and [Parameter Line](#) for more information.

**Display File Comments (default = Off)**

Enables a Message Screen (see below) that displays all "comment" lines in a file as the file is being executed. The Message Screen will also be used to display the information provided by any "Print" statement in a file.

**KeyPad Fixed Distance (default = .1)**

This is used when the ShopBot KeyPad [\[SK\]](#) is in 'Fixed Distance' Mode. It tells the software how far to move each time you give an arrow key or remote button a single tap. Note that this value changes automatically from inches (0.1) to mm (2.54) when you change the Display Units from inches to metric. Note that this value is displayed and can also be edited on the KeyPad after clicking on Fixed Distance. When making a fixed distance move, make sure you hold the key down until the move is executed.

**KeyPad Remote (default = 0)**

Tells the software to look for a mouse (0), ShopBot's remote pendant wired into the input switches (1) or another source (2) when using a remote device.

**KeyPad Switch AutoOff (default = 0)**

Setting to ON (1) will cause all Output Switches to be automatically turned off when the KeyPad function is closed.

**Write Part File Log (default = Write [1])**

Sets whether a log file is (1) or is not (0) written every time an .sbp file is run. These log files are used to remember information about the last time the .sbp file was cut.

**Write System File (default = Write [1])**

Sets whether a system file is (1) or is not (0) generated. The system files are useful for diagnostics and maintaining records on the use of the ShopBot.

**Message Screen Location X, Message Screen Location Y, Message Screen Size X, Message Screen Size Y**

These parameters set the size and location (in Twips) of the Message Screen. The Message Screen is displayed when a Part File uses a "Print" statement to display data or information, or it can be enabled by the 'Display File Comments' parameter, above. These parameters are typically used in a Part File to control where a Message Screen Displays. When a Message Screen is visible it can be moved and resized and its settings will be saved if the screen is closed.

## VH - Values for Z-axis Height Controller

```
{Height control mode, Speed(%XY step rate)}
```

This command activates height control system for the Z-axis that is used for automatic control of the height of a plasma cutter head based on a voltage feedback signal. This function might also be of interest for other types of application. The feature, enabled with the [VE], works during any XY move and causes the Z axis to

move down when the #1 Input line is ON and pull up if the #1 Input line is OFF. These Z moves happen without any disruption to the ongoing XY movement.

#### **Height control mode (default = 0)**

This parameter is used to activate the Height Control function. Entering a 1 turns the height control on. It will remain active until it is turned off, 0-OFF, though it will only be operational during an XY move.

#### **Speed (default = .25)**

Scales the rate of pull up in steps. The default here for 25% the speed of the XY axis is about right for plasma cutting.

## VI - Values for Communications Port and Other Addresses

{Comm port number, -obsolete-, -obsolete-, -obsolete-, Driver Channel 1, Driver Channel 2, Driver Channel 3, Driver Channel 4, Speed, Driver Channel 5}

This command sets various communications port and driver channel parameters.

#### **Comm Port Number**

Sets the number of the communication port used to connect your computer to your ShopBot Control Box. Normally, this number will be set by the software at the time of your initial connection and saved for use when you run your tool the next time. Communication Ports 1 - 16 can be used by your ShopBot software. See Trouble Shooting [\[HT\]](#) "Connection Problems" for more details on the steps to take if your computer is not connecting with the ShopBot Control Box.

#### **-obsolete parameters-**

Leave empty as space-holder for parameter no longer used.

#### **Driver/Channel Definition**

Every motor on the ShopBot has a driver in the control box that determines the direction and step of that motor. Each driver is associated with a driver channel.

In a standard configuration, the axes associated with the driver channels are as follows:

Driver Channel 1 – X axis.  
 Driver Channel 2 – Y axis.  
 Driver Channel 3 – Z axis.  
 Driver Channel 4 – A axis.  
 Driver Channel 5 – B axis.

#### **ComSpeed (4=460.8, 2=115.2Kb) (PRTalpha default = 4, PRT default = 2)**

This is the speed at which your computer communicates with your control box. The communication speed is determined by the software in the control box and this parameter is set automatically at startup depending on whether you have a PRT or PRTalpha.

## VL - Values for Limits for Table

{low-X , high-X, low-Y , high-Y, low-Z , high-Z, low-Acc , high-Acc, [1-file limit checking ON or 0-limit checking OFF], low-B, high-B}

This command sets Table Limits for your tool. When a command that would take your tool out of the defined movement range is encountered, an error message is generated and you are offered the opportunity to 'Continue' or to end the process that is currently executing. Table Limits are referenced to the Table Base Coordinates not to the current working location or to the location in an 'offset' file. Limits that don't match the actual physical layout of the tool could be set up for special purposes.

#### **Low-X,Y,Z,A or B**

Low values are the smallest value for a table axis -- the smallest location point to which travel can go on that axis. In the Z-axis, the low value would be the deepest plunge value accepted

#### **High-X,Y,Z,A or B**

High values are for the largest value to which a tool can move in an axis. In the Z axis the high value would be the highest pull up value accepted.

### **File limit checking**

Turns limit checking ON or OFF. This parameter has the same effect as the [\[SF\]](#) command. If used within a Part File, when the file finishes execution, Limit Checking will revert to its status at the time the Part File was started.

## VN - Value Input Switch Assignment

```
{Limit Switches Active/Disabled, Stop switch number 4 mode, Enable Torch Height Controller, Input Switch mode (for Inputs 1-3; 5-8),etc,, }
```

This command turns on or off the sensing of Limit Switches, also known as Proximity or Prox Switches.. It also sets up the operation (Normally –Open or Normally-Closed) of Limit Switches (optionally Inputs 1-3 and 5-8) and your Stop Switch (Input 4).

### **Limit Switches Active (default = 1, ON)**

In the default mode (1), Switches designated in further commands as Limit Switches will be activated to stop the tool anytime one of them is triggered. The switches can be over-ridden in keypad mode by releasing the arrow key or button and then pressing the arrow key or button down again. If a Limit Switch is encountered in a Part File or move, the tool will stop and a message will be displayed to alert you. You will have the option quitting the file or of overriding the Limits for the duration of the file or move.

### **Input Switch #4 mode (default = 0, Normally-Closed)**

Input switch number 4 is permanently defined as an “Emergency Stop” on ‘alpha’ model ShopBots and as a “Remote Stop Switch” on ‘standard’ model Shopbots (PRTs or PRSs).

PRSalph/PRTalpha – Activating the Emergency Stop on an ‘alpha’ ShopBot will immediately stop the tool and disconnect power to the Spindle/Router. After approximately 5 seconds (to allow time for the cutter to slow down), power will also be disconnected from the stepper motors. After the Emergency Stop has been activated on an ‘alpha’, the tool must be homed in the XY axis and zeroed in the Z to re-establish accurate position before cutting is re-started. This is because position can be lost when power is removed from the stepper motors.

Hitting the space bar (alternately the ENTER Key) will also Stop the movement of the carriages. If the tool is moving faster than 3ips, the stop will be ‘ramped’ and the Z axis will be pulled out of the material. Cutting can be resumed with no loss of location after a space bar stop (or the ‘S’ key which forces a ramped stop).

PRSalph/PRTalpha – #4 Input operates in the normally closed mode and requires four wires be connected to the control board (Black and green to ES-1 and ES-2, Red and white to ES-3 and ES-4).

PRStandard/PRTstandard – Activating the Remote Stop Switch will stop the movement of the carriages. The carriages cannot be moved until Switch #4 is released. Hitting the space bar will also stop the movement of the carriages. If the tool is moving faster than 3ips, the stop will be ‘ramped’ and the Z axis will be pulled out of the material. Cutting can be resumed with no loss of location after the Remote Stop is used.

PRTstandard – For a PRTstandard tool, your Stop Switch may be wired in one of two ways and this will dictate how the setting for this parameter in the software. PRTstandards may have switches that are either Normally-Open (older) or Normally-Closed (newer). When the software is configured in the Normally Closed (0) position, the carriages will not move unless the Remote Stop is correctly wired into the Control Box and the Stop Switch is released.

### **Enable Torch Height Controller (default = 0 disabled)**

This parameter can be used to enable automatic torch height control on a Plasma Cutter ShopBot equipped with a plasma Torch Height Controller.

### **Input Switch Mode [for Inputs 1-3 and 5-8] (default = 0, standard)**

Sets the Input Switch Mode for all inputs except #4 (see above). Inputs can be set as ‘standard’ and will be available for use in Part File programs (see Programming Handbook for information on how to read and use switches). Or Inputs can be set as Normally-Open or Normally-Closed Limit Switches, in which case their activation will stop the movement of the tool.

PRT Shopbots use Normally-Open switches for Limits. These are usually connected in parallel to Input #3.

PRS Shopbots use Normally-Closed switches for Limits. These are usually connected to Inputs #2 (X) and #3 (Y).

## VP - Values for Preview Screen

*{grid, table size-X, table size -Y, table size -Z, simulate, rotation arrow, automatic center, show jogs, renderer type*

Sets up the Table Top view. Allows changing the grid size as well as configuring many other aspects of your Preview Mode Display.

### **grid (default = 6)**

Sets the grid size in inches of you table top on the preview screen

### **table size X (default = 96)**

This is where you set the length of your table's X axis.

### **table size Y (default = 48)**

This is where you set the length of your table's Y axis.

### **table size Z (default = 6)**

This is where you define your table's Z travel distance

### **simulate (default = 0-Off)**

Turns the simulate function On (1) or Off (0). When this function is on, a simulation of the toolpath will be displayed on the preview screen. For very large or complex files it can take a long time to complete the simulation. Uncheck the simulate box in order to draw the screen faster.

### **rotation arrow (default = 1-On)**

Sets whether the bit rotation arrow will be displayed on the preview screen [On (1), Off(0)]

### **automatic center (default = 0-Off)**

When set to On(1) the preview of the file will be centered on the screen. Off (0) disables this function.

### **show jogs (default = 1-On)**

When this function is On (1) the jogs or rapid moves in tile will be displayed in blue on the preview screen. Jog moves will not be visible when this parameter is set to Off (0).

### **renderer type (default = 1-DirectX)**

Specifies the type of software rendering that will be used. In addition to DirectX you can choose OpenGL or R98. It is recommended that you use DirectX unless you are experiencing a problem with the preview screen.

See also: [\[FG\]](#)

## VR - Values for Ramps

*{XY-Move Ramp Speed, Z-Move Ramp Speed, A-Move Ramp Speed, B-Move Ramp Speed, XY-Jog Ramp Speed, Z-Jog Ramp Speed, A-Jog Ramp Speed, B-Jog Ramp Speed, Move Speed Ramp distance, Jog Speed Ramp distance, Threshold%, Minimum distance}*

Defines the acceleration/deceleration ramps at the beginning and end of each type of move. Ramping makes for smoother cuts and enhanced control of the tool movement. The values are the starting or ending speeds (in inches/sec or mm/sec) from which an acceleration ramp-up starts or a deceleration ramp-down ends. The default values initially set are appropriate for the default speeds. However, can explore different ramp values to determine which are best for your particular applications. Additional details regarding ramping can be found in the [Ramps – Full Explanation](#) section. Basically, with ramping you should be going for values that give smooth movement of your tool and reduce vibration caused by fast starts or stops.

### **XY, Z, A, B Move Ramp Speed (def =.4)**

The Move Ramp Speed sets the start and end speed for acceleration and deceleration ramps for each axis for cutting speeds,

**XY,Z,A,B Jog Ramp Speed (def =.4)**

The Jog Ramp Speed sets the start and end speed for acceleration and deceleration ramps for each axis for rapid transit speeds.

**Move Ramp Rate, and Jog Ramp Rate (default =.2, .2)**

The distances over which a change of 2 Units is made in Move Speed or Jog speed during ramping. This parameter thus sets the acceleration deceleration rate.

**3D Threshold (default = 100 [percent])**

This value adjusts the sensitivity of 3D ramping. Decreasing makes 3D ramping more sensitive, increasing makes it less sensitive. [Ramps – Full Explanation](#).

**Minimum distance (default =.15)**

The minimum distance considered in making ramping decision. Normally, you should not have to change this parameter.

**Slow corner speed (default = 65 [percent])**

The percentage of speed reduction when ramping is triggered for a slight change in direction. Angular changes in direction produce increasingly greater reduction in speed from this value to 100% of the full ramp reduction.

**Fast Stop Threshold (default = 3.0)**

Speed below which tool stops instantly when stop (Space Bar Stop or E-Stop) is pressed. Above this speed, there will be a ramped stop with a pull-up of the Z axis. This higher speed stop is the same as that activated when 'S' is pressed to stop the tools motion.

**KeyPad Stop Threshold (default = 1.75)**

KeyPad speed below which tool motion stops instantly on release of the an arrow key. Speed reduction is ramped above this value. Note that deceleration ramping is brisk compared to acceleration in order to limit motion on release of key.

**KeyPad Ramp Rate (default = .8)**

The distances over which a change of 2 Units is made in Move Speed or Jog speed during ramping when the KeyPad system is in use. This parameter sets the acceleration rate for KeyPad moves. The high default value (.8) assures a smooth increase in speed that is suited to manual control. Deceleration rate is automatically set much lower so that stopping will be brisk when an arrow key is released.

See also: [Ramps – Full Explanation](#)

## VS - Values for Speeds

{XY Move Speed, Z Move Speed, A Move Speed, B Move Speed, XY Jog Speed, Z Jog Speed, A Jog Speed, B Jog Speed}

This command sets speed values for every axis with a single command. It is similar to [\[MS\]](#) and [\[JS\]](#) but provides access to all 8 speeds. Speeds are defined in inches/sec or mm/sec depending on the measurement unit.

**Move Speed**

Sets the speed that will typically be used for cutting.

**Jog Speed**

Sets the speed for rapid moves. From your tool's point of view, there is no difference between move and jog, only that you have set them up to be carried out at different speeds.

Speeds are set in inches or mm per second, depending on the unit setting. The new values will appear on screen in the upper left panel. These speeds represent the actual X and Y movement speed of the cutter tip. Because the vector speed along a diagonal represents the geometric addition of the X and Y vectors, for any given speed, the motors will operate more slowly on a diagonal move than on a vertical or horizontal move. The change you notice is the sound of the motors is as it should be. For a circle the speed of the motors will continuously change up and down in order to achieve a constant vectored movement speed.

See also: [\[MS\]](#), [\[JS\]](#), see [\[VU\]](#) for setting extremely slow speeds

## VU - Values for Calibration Units

*{x-axis Unit, y-axis Unit, z-axis Unit, a-axis Unit, circle segment resolution, (obs), (obs), small circle, b-axis Unit, resM-x, resM-y, resM-z, resM-A, resM-B, StepInt divider}*

### Units

The initial parameters of this command deal with the scaling or measurement units of your tool. They adjust the software to the specific gear ratio of your tool. The number is the exact number of steps per unit (inches, mm, or degrees) for each axis. Current values can be viewed at anytime with [\[UV\]](#). The correct values for your specific tool can be found with the documentation provided with your tool. They are also available by selecting the appropriate settings file for your tool [\[UR\]](#), or you can look up Unit values in the Dr ShopBot (tech support) Wiki that can be accessed from the website.

Computing the Unit Value for an indexer would be done as follows, assuming you want to express the rotary motion in degrees. With an indexer having a 10:1 gearbox and with the alpha driver set at 500 steps/rev, we would have:  $(10 * 500) / 360 = 13.8889$  [or, 13.8889 steps per degree].

If you want to use your tool with the display and command units in millimeters rather than inches, it is easier to first enter the appropriate Unit Values while the tool is still set in inches; then you can just switch to mm and all the necessary values will be re-computed. If you were defining Unit Values from mm mode, you would enter values appropriate for mm units.

### Circle Segment Resolution (default = .05)

Sets the size that will be used for the little segments that make up the curve of the circle when a ShopBot circle or arc command is used, that is when one of the 'C'uts commands is used. Segment Resolution can be set as low as .005. Note that if you are using the tool to drill small holes by making small diameter circular movements, the resolution will automatically downshift to the highest resolution for these movements.

There are two down sides to setting Resolution to a very small value even though this might seem to be a good idea for a very smooth circle. First, on some older computers a low setting may increase the computation time or slow the tool slightly during execution. The second down side of setting the resolution to a low value, is that a circle or arc requires many more segments and thus requires more memory in order to execute. If the size of the circle is large and the Resolution is low the tool may have to pause to load the blocks of movements for the circles in pieces rather than in the single block that would allow a single smooth cut.

### -obsolete parameters-

Leave empty as space-holder for parameter no longer used.

### Small circle (default = .25[inches])

Circle diameter below which execution will occur at Ramp Speed rather than Move Speed. Small circles would be executed too abruptly if they accelerated to full Move Speed within their small diameter. This would be a particular issue with a spiral plunge.

### Resolution Multipliers (resM)

This controls the temporary shift in resolution for moves made in Jog Mode on PRT and PRS alpha tools. For PRS and PRT standard ShopBots, all resolution multipliers should be set to 1. Select the appropriate Settings file for your tool [\[UR\]](#) or see the Dr ShopBot Wiki for values for alpha tools.

### Step Interval Divider (default = 1)

The standard movement speed or federate can be set as low as .05 (inches/sec; or 1.27 mm/sec) see [\[VSI\]](#). Occasionally, it may be necessary for a specialized application to move your tool at even slower speeds. Slower speeds can be achieved by dividing the timing interval between steps so that stepping is even slower. The step interval can be increased by whole numbers so that it is 2 times longer, 3 times longer, etc up to 100 times longer. This means the speed will be ½ slower, 1/3 slower, etc ... up to 1/100 the displayed speed. Only use the step interval divider when a very slow speed is needed.

See also: [\[UV\]](#), [\[UR\]](#) and the [\[C..\]](#) commands

## Z Commands Note

This command and the other 'Z' commands like it are thus used to set the working 0 values for your tool. Note that they are very powerful and should be used with care, since they reset your position. There are no parameters for the Z commands.

[\[VA\]](#) is a command that is similar to the 'Z' commands in that it also resets the working location of your tool. With [\[VA\]](#) you can set the current location to any value, including zero.

In addition to its 'working' coordinate system, ShopBot maintains a 'base' (reference or absolute) coordinate system that can be used to re-establish your working location in terms of the base zero points (the real corner) of your tool. This system is set up with [\[ZT\]](#) (see below) and re-established with [\[ST\]](#). Note that limit checking ([\[VL\]](#), [\[SF\]](#)) is done with reference to the base coordinate system and not the current working coordinates.

See also: [\[ST\]](#), [\[VL\]](#), [\[SF\]](#) other [Z..] commands

### Z2 - Zero 2D (X & Y) axes

This command zeros out (sets to 0.000) the current location of the X and Y axes of your tool. That is, after [\[Z2\]](#) is used, the current physical location of your tool will now be considered 0 for the X and Y axes. See Z Commands Notes for more information.

### Z3 - Zero 3D (X, Y & Z) axes

This command zeros out (sets to 0.000) the current location of the X, Y, and Z axes of your tool. That is, after [\[ZA\]](#) is used, the current physical location of your tool will now be considered 0 for the X, Y, and Z axes. See Z Commands Notes for more information.

### Z4 - Zero 4D (X, Y, Z, & A) axes

This command zeros out (sets to 0.000) the current location of ALL axes of your tool. That is, after [\[Z4\]](#) is used, the current physical location of your tool will now be considered 0 for the X,Y,Z, and A axes.. See Z Commands Notes for more information.

### Z5 - Zero 5D (X, Y, Z, A, and B) axes

This command zeros out (sets to 0.000) the current location of ALL axes of your tool. That is, after [\[Z5\]](#) is used, the current physical location of your tool will now be considered 0 for all 5 axes. See Z Commands Notes for more information.

### ZA - Zero A axis

This command zeros out (sets to 0.000) the current location of the A axis of your tool. That is, after [\[ZA\]](#) is used, the current physical location of your tool will now be considered 0 for the A axis. See Z Commands Notes for more information.

### ZB - Zero B axis

This command zeros out (sets to 0.000) the current location of the B axis of your tool. That is, after [\[ZB\]](#) is used, the current physical location of your tool will now be considered 0 for the B axis. See Z Commands Notes for more information.

### ZT - Zero Table Base Coordinates

In addition to its 'working' coordinate system, ShopBot maintains a 'Base' (reference or absolute) Coordinate System that can be used to re-establish your working location in terms of the base zero points (the real physical corner) of your tool. This system is set up with [\[ZT\]](#) and re-established with [\[ST\]](#). Note that limit checking ([\[VL\]](#), [\[SF\]](#)) is done with reference to the Base Coordinate System and not the current working coordinates.



The [\[ZT\]](#) command zeros out (sets to 0.000) the base coordinate system of your tool. You would typically use this command to set the true physical 0,0,0,0 point of your tool after moving each axis to its starting or zero location. Base coordinates are maintained as you change the working coordinate system of your tool so that at anytime, you can return to this true base coordinate system.

NOTE: [\[ZT\]](#) is powerful in that it resets your Table Base Coordinate System. Only use it when necessary.

NOTE: The Base Coordinate System is maintained in your computer. If the tool is physically moved manually, the correct Base Coordinates will be lost. They may also be lost in the event of a power failure or a computer crash.

In an unusual situation, the second 4 parameters of [\[VA\]](#) can be used to set Base Coordinate System values to specific values.

See also: [\[ST\]](#), [\[VL\]](#), [\[SF\]](#) other [\[Z..\]](#) commands

## ZX - Zero X axis

This command zeros out (sets to 0.000) the current location of the X axis of your tool. That is, after [\[ZX\]](#) is used, the current physical location of your tool will now be considered 0 for the X axis. See Z Commands Notes for more information.

## ZY - Zero Y axis

This command zeros out (sets to 0.000) the current location of the Y axis of your tool. That is, after [\[ZY\]](#) is used, the current physical location of your tool will now be considered 0 for the Y axis. See Z Commands Notes for more information.

## ZZ - Zero Z axis

This command zeros out (sets to 0.000) the current location of the Z axis of your tool. That is, after [\[ZZ\]](#) is used, the current physical location of your tool will now be considered 0 for the Z axis. See [Z Commands](#) Notes for more information.