MAX® FOR LATHE CONTROL

Preliminary
NC Programming Manual
The information in this document is subject to change without notice and does not represent a commitment on the part of Hurco Companies, Inc. (Hurco). The software described in this document is furnished under the License Agreement to customers. It is against the law to copy the software on any medium except as specifically allowed in the license agreement. The purchaser may make copies of the software for backup purposes. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, for any purpose without the express written permission of the Hurco machine tool owner.

Hurco Manufacturing Company reserves the right to incorporate any modification or improvements in machines and machine specifications which it considers necessary, and does not assume any obligation to make any said changes in machines or equipment previously sold.

Hurco products and services are subject to Hurco’s then current prices, terms, and conditions, which are subject to change without notice.

© 2004 Hurco Companies, Inc. All rights reserved.

**Patents:** U.S. Patents B14,477,754; 5,453,933; Canadian Patent 1,102,434; Japanese Patents 1,649,006 and 1,375,124; other Patents pending.

Hurco, Max, Ultimax, and WinMax are Registered Trademarks of Hurco Companies, Inc.

Conversational Probing, UltiDraw, UltiNet, UltiPocket and AutoSave are trademarks of Hurco Companies, Inc.

AutoCAD, Autodesk, and DXF are registered trademarks of Autodesk, Inc.

Fanuc is a registered trademark of Fanuc LTD.

IBM and PC/AT are registered trademarks of International Business Machines Corporation.

MS-DOS, Microsoft, and Windows are registered trademarks of Microsoft Corporation.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Hurco has listed here all trademarks of which it is aware. For more information about Hurco products and services, contact:

**Hurco Companies, Inc.**
One Technology Way
P.O. Box 68180
Indianapolis, IN 46268-0180
Tel (317) 293-5309 (products)
(317) 298-2635 (service)
Fax (317) 328-2812 (service)

For Hurco subsidiary contact information, go to Hurco’s Web site:
www.hurco.com
# TABLE OF CONTENTS

Max® for Lathe Control .......................................................... i

Table of Contents ................................................................. v

List of Figures ................................................................. vii

List of Tables ................................................................. 1

## Introduction
- Options/Package ............................................................. 1 - 1
- Max Classic Package ......................................................... 1 - 2
- Tool Presetter ................................................................. 1 - 2
- NC ................................................................. 1 - 2
- Bar Feed Interface ............................................................. 1 - 2

## Max for Lathe Control ..................................................... 2 - 1
- Calibration ................................................................. 2 - 1
- Part Setup – Work Offsets .................................................. 2 - 1
- Tool Setup ................................................................. 2 - 2
- Coolant Control ............................................................. 2 - 7

## NC Programming ............................................................. 3 - 1
- Definitions ................................................................. 3 - 1
- Command Overview .......................................................... 3 - 2
- How Your Integrator Can Customize Max® Software .................. 3 - 3
- Format for Part Programs Prepared Off-line (CAM System) ........ 3 - 3
- General Format for Creating Part Programs .......................... 3 - 7
- Coordinate System Introduction ........................................ 3 - 8
- Absolute and Incremental Dimensions .................................. 3 - 10
- Drill Cycles on Any Axis .................................................. 3 - 14
- G58 - Automatic Withdraw ................................................ 3 - 34
- G90 - Absolute Programming (default) .................................. 3 - 49
- G91 - Incremental Programming ......................................... 3 - 49
- G92 - Work Coordinate Offsets or Spindle Max Speed ............. 3 - 49
- G93 - Inverse Time Feed .................................................. 3 - 49
- G94 - Feed per Minute (default) ......................................... 3 - 50
- G95 - Feed per Revolution ................................................ 3 - 50
- G96 - Constant Surface Speed (CSS) .................................... 3 - 51
- G97 - Direct Spindle Speed (default) .................................... 3 - 51
- M Code Descriptions ........................................................ 3 - 52
- M00 - Program Stop ......................................................... 3 - 52
- M01 - Optional Stop ......................................................... 3 - 53
- M02 - End of Program (no rewind) ...................................... 3 - 53
- M03 - Spindle Clockwise ................................................... 3 - 53
- M04 - Spindle Counterclockwise ......................................... 3 - 53
- M05 - Spindle Off ........................................................... 3 - 54
- M07 - Secondary Coolant On .............................................. 3 - 54
- M08 - Primary Coolant On ................................................. 3 - 54
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>M09</td>
<td>Coolant Off</td>
<td>3 - 54</td>
</tr>
<tr>
<td>M30</td>
<td>End of Program (rewind)</td>
<td>3 - 54</td>
</tr>
<tr>
<td>M48</td>
<td>Use feedrate override</td>
<td>3 - 54</td>
</tr>
<tr>
<td>M49</td>
<td>Ignore feedrate override</td>
<td>3 - 55</td>
</tr>
<tr>
<td>M200</td>
<td>Block Delete Synchronization</td>
<td>3 - 55</td>
</tr>
<tr>
<td>Block Delete Code</td>
<td></td>
<td>3 - 56</td>
</tr>
<tr>
<td>Programming a Tool Change and Activating Tool Offsets</td>
<td>3 - 59</td>
<td></td>
</tr>
<tr>
<td>E Codes</td>
<td></td>
<td>3 - 62</td>
</tr>
</tbody>
</table>

Glossary ....................................................................... GL - 1
LIST OF FIGURES

Figure 2–1. Tool Setup Screen ..................................................... 2 - 2
Figure 2–2. Tool X Axis Offset .................................................... 2 - 5
Figure 2–3. Tool Orientation ....................................................... 2 - 6

Figure 3–1. Coordinate System for Typical Back Turret Lathe .................. 3 - 8
Figure 3–2. Absolute Dimensions ............................................... 3 - 10
Figure 3–3. Incremental Dimensions ............................................ 3 - 12
Figure 3–4. Arc with Endpoint and Center Coordinates ......................... 3 - 15
Figure 3–5. G02/G03 Arcs Example ............................................. 3 - 16
Figure 3–6. Arc with Endpoint and Radius ..................................... 3 - 17
Figure 3–7. G07 Radius Programming ........................................... 3 - 18
Figure 3–8. G03 Diameter Programming ....................................... 3 - 18
Figure 3–9. Turret Probe Calibration—S0 Parameters .......................... 3 - 21
Figure 3–10. Fixture Offset (X,Z) - S1 Parameters ............................ 3 - 24
Figure 3–11. Measure Diameter or Part Length - S2 Parameters ............... 3 - 26
Figure 3–12. Adjust Tool Offsets (X, Z) - S3 Parameters ..................... 3 - 28
Figure 3–13. Activate and Exit Actions; Circular Joining Actions ............. 3 - 30
Figure 3–14. Cutter Compensation Right Turned Off (G40) .................... 3 - 32
Figure 3–15. Cutter Compensation Turned On (G42) ........................ 3 - 32
Figure 3–16. G58 Automatic Withdraw ......................................... 3 - 34
Figure 3–17. Thread Parameters Example ..................................... 3 - 37
Figure 3–18. OD Thread with Offsets U and W Example ....................... 3 - 38
Figure 3–19. OD Thread with Chase in (A) and Chase out (C) Example .... 3 - 38
Figure 3–20. OD with multiple starts at 120°, 170°, and 180° Example ........ 3 - 39
Figure 3–21. OD with 3 Evenly Spaced Threads Example ..................... 3 - 39
Figure 3–22. Tapered Thread Example .......................................... 3 - 40
Figure 3–23. No Programmed Finish Pass Example .......................... 3 - 41
Figure 3–24. Three Programmed Finished Passes Example ................... 3 - 41
Figure 3–25. Straight OD Thread Example .................................... 3 - 43
Figure 3–26. Straight OD Thread with Lead In/Lead Out Angles .............. 3 - 44
Figure 3–27. Tapered OD Thread Example .................................... 3 - 45
Figure 3–28. Straight ID Thread Example ..................................... 3 - 46
Figure 3–29. Tool Offset Behaviors .............................................. 3 - 61
Figure 3–30. Fixture Offset: Immediate Activation ............................. 3 - 62
Figure 3–31. Fixture Offset: Deferred Activation ............................... 3 - 63
Figure 3–32. Tool Motion for G73 Chip Break Drill Cycle ..................... 3 - 64
Figure 3–33. G81 Drill Cycle ....................................................... 3 - 65
Figure 3–34. G82 Drill Cycle with Dwell ....................................... 3 - 66
Figure 3–35. G83 Peck Drill Cycle ................................................ 3 - 67
Figure 3–36. G84 Right Hand Tapping and Float Tapping ..................... 3 - 68
Figure 3–37. Rigid Tapping for Non-positioning Spindle ....................... 3 - 70
Figure 3–38. General Drill Cycle Tool Motion .................................. 3 - 72
Figure 3–39. Drill Cycle Motion with G98 ..................................... 3 - 73
LIST OF TABLES

Table 3–1. Order of Block Code Processing .............................................. 3 - 4
Table 3–2. Most Common Block Data Formats ........................................... 3 - 5
Table 3–3. Other Block Data Formats .......................................................... 3 - 6
Table 3–4. Effects of Using G07 and G08 .................................................... 3 - 9
Table 3–5. Cancel G58 Options ................................................................. 3 - 35
Table 3–6. Face Parameter ................................................................. 3 - 36
Table 3–7. Turn Thread Parameters .......................................................... 3 - 37
Table 3–8. Thread Length Parameters ......................................................... 3 - 37
Table 3–9. Clearance and Chase Parameters ............................................ 3 - 38
Table 3–10. Multiple Start Parameters .................................................... 3 - 39
Table 3–11. Tapered Thread Parameter ...................................................... 3 - 40
Table 3–12. Thread Cutting Parameters ...................................................... 3 - 40
Table 3–13. Threading Equivalents between Turn and Face Thread Macros 3 - 47
Table 3–14. Face Threading Parameters .................................................... 3 - 48
Table 3–15. R Values and G96 Programming ............................................ 3 - 51
Table 3–16. M Code Options for Look-Ahead Program Execution ............. 3 - 58
Table 3–17. Tool Offset Options ............................................................... 3 - 60
INTRODUCTION

This document describes the general operation and programming for the Lathe control.

- The Introduction chapter identifies the option packages available with the control.
- The Max for Lathe Control chapter describes the general operation of the control, i.e. calibrating axis, setting tools and part, etc.
- The NC Programming chapter describes G-code part programming, i.e. G and M code descriptions. In addition, this chapter describes the special profiling and grooving macros that can be called from within a M&G code program.
- The Glossary contains definitions for terms used in this manual.

Options/Package

- Overview Package
  - Graphics
    - Wireframe to be standard (shows program path)
    - Shaded graphics to be optional (shows tool path) – Like the Classic package.
  - Memory
    - Standard will be minimum required.
    - 256 MB is optional in Classic
  - Hard Drive
    - Listed as 2 GB standard
    - Maximum of drive is optional in Classic
  - 3.5” floppy
    - Included on all system
    - Part of Optional Classic package
  - DXF is optional with OptiKey
  - Tool Touch Setter is optional with OptiKey
  - UntiNet is optional
  - The NC editor is standard.
  - Options controlled by Parameter Only (no optikey)
• Chip Conveyor
• Parts Catcher
• Bar Feed

• Control Notes & Security
  • A USB Drive should be used for software updates rather than a CD in a CD Drive.
  • OptiKey security should be switched to a “Dallas” USB key in the near future.

Max Classic Package

• 256 MB RAM
• 2 GB Hard Drive
• 3.5” Floppy disk
• Touch Screen Shaded Graphics with Stylus

Tool Presetter

Automatic System (motorized arm for up and down) Optional

NC

• No ISNC
• Only Hurco Lathe NC – included as Standard.
• NCPP – If, While, For statements and math functions – optional.

Bar Feed Interface

• Available with Hurco’s Bar Feeder brand.
• Available for 3rd party bar feeders with PLC’s; initially limited to LNS and SMW.
• Hurco provides connectors
• Hurco provides wiring diagram
MAX FOR LATHE CONTROL

This chapter explains Max for Lathe control operations.

Calibration

The axis must be calibrated each time the machine is powered up to determine the machine coordinate zero location. X zero will be towards the back of the machine. Z zero will be to the right away from the spindle. The X axis will calibrate first and then the Z axis. This is done to avoid collision with the tail stock. X Positive direction is towards the back of the machine. Z positive is the direction away from the base of the spindle.

Calibrated Z value will be the Z travel limit. Calibrated X value will be the distance to the Spindle Center Line.

To calibrate the machine press the Manual button on the console. The Manual screen will appear. Press the Calibrate menu item. The Start Cycle lamp will flash. Press the Start Cycle lamp to calibrate the axis.

Part Setup – Work Offsets

To setup your part you need to define the X and Z offsets that define the zero location for the part.

- **X Offset – Z Offset** —There are up to 99 work offsets that can be programmed. Each offsets is defined by storing the machine position for that axis. Typically this is done be jogging a tool with a known X and Z offsets to the part and storing the Machine Position for that tool.

- **Active Offset**—You can specify which offset is active when you run your program by entering the index value (1 – 99) in the Active Offset field.

  You can also define a new Active Offset by inserting a Change Part Setup data block within your part program.
Tool Setup

Tool Setup allows you to define tools for use in conversational and M&G code part programs. The insert shape and size information is used to graphically simulate the part program. The max depth of cut and speed and feed values are used in conversational programs but are not required for M&G code programs since you can specify S and F codes within the G-code program. Geometry Offsets are always required to run a conversational or G-code program.

Figure 2–1. Tool Setup Screen
The Tool Setup fields are defined as follows (note that not all fields appear on the screen for all tools):

- **Tool**—identify the number of the programmed tool.

There are several ways to add a tool:

- Enter a new number in the Tool field and move to another field.
- With the cursor in the Tool field, press the PAGE DOWN key to display the New Tool field. Enter a new tool number and press the ENTER key. The remaining Tool Setup fields appear.
- With the cursor in the Tool field, press CTRL + The Right Arrow Key. Enter a new tool number and press the ENTER key. The remaining Tool Setup fields appear.
- The Tool Review screen displays a list of programmed tools. Use the Add Tool or Delete Tool softkey on this screen to add or delete tools.

The range of possible tool numbers is 0 through 99.

The Next Tool and Previous Tool softkeys become available when more than one tool has been programmed. Use these softkeys as appropriate to access tool setup screens within a program.

**Fields for All Tool Types**

All tool types use the following input fields:

- **Tool**—This field defines the tool number that will be used in the part program. Up to 99 tools can be defined.
- **Type**—identify the type of tool to use for this tool number. The tool type determines the need for parameters required to define that tool. Use the appropriate softkey or drop-down list to select a tool type. If you do not see the type of tool that you wish to use, click the More softkey to display more tool types.
- **Comment**—The comment field allows you to enter up to twenty characters to help describe the tool.
- **Default Offset**—This field defines the tool geometry and wear offsets used for this tool.
- **Speed**—Speed may be programmed as an RPM value or as a Constant Surface Speed value in Feet Per Minute for Inch programs or Meters Per Minute for Metric programs. The speed will be copied into each new data block in the part program that uses this tool. If necessary, this parameter can be changed within the new data block when programming the part.
- **Feed**—You may define the programmed feed rate for this tool as Inches or Millimeters Per Minute or as Inches or Millimeters Per Revolution.
- **Coolant**—Coolant may be programmed as OFF, Primary, Secondary or Both.
Fields for Turning, Boring, Threading, Back Turning and Back Boring Tools

- Max Depth of Cut—This defines the maximum depth of cut that you can program with this tool. When entering a tool number in a data block the data block’s Depth of Cut field will automatically be filled with this value.

- Insert Shape—You may select from 17 standard shapes to define the geometry of the insert. This field is used only for graphical part program verification.

- Insert Size—Insert Size defines the size of the insert. This field is used only for graphical part program verification.

Fields for Drills and Center Drills

- Diameter—This field specifies the tool’s diameter. It is used primarily for part program graphical verification. The diameter range is from 0 to 9.999.

- Max X Offset—This field defines the maximum amount a tool can be offset from the spindle center line when boring a hole. This is useful for cases where you need to bore a hole larger than the tool diameter. Not all tools are capable of this.

- Insert Size—Insert Size defines the size of the insert. This field is used only for graphical part program verification.

- Surface Speed (FPM)—The Surface Speed, Flutes, and Feed/Flute fields appear. Enter a Surface Speed in feet per minute (or millimeters per minute), and the Flutes and Feed/Flute fields appear. Enter values that indicate the number of the flutes (teeth) on the tool and the feedrate per flute. The system uses the values entered in Surface Speed, Flutes, and Feed/Flute to automatically calculate the feeds and speeds for the tool.

- Flutes—Enter the number of cutting flutes for the tool. This entry will be used to automatically calculate the Mill Feed in all data blocks for this part program using this tool.

- Feed/Flute (Tooth)—Enter the tool's chipload. This entry will be used to automatically calculate the Mill Feed in all data blocks for this part program using this tool.

Fields for Tap Tools

- Pitch—Appears when the tool type is Tap. Pitch is the distance between two points on adjacent screw threads. Divide 1” (or 25.4 mm) by the number of threads to calculate this number. This field can be used to calculate the TPI or TPMM field.

- TPI/TPMM (Threads Per Inch/Threads Per Millimeter)—This field lets you enter the number of threads per inch or millimeter and then calculates the Pitch for you.
Fields for Grooving and Cut Off Tools

- **Max Step Over**—Defines the distance the tool will move laterally before making the next plunge into the part.

**Tool X Axis Offset**

![Figure 2–2. Tool X Axis Offset](image)

**X/Z Tool Offset**

X offset from machine supplier defined point on face of turret. It is calibrated by touching the tool off on the stock material in the X axis. You must know the diameter of the stock to store the calibrated tool offset value.

**Tool Z Axis Offset**

Calibrating the Tool Z Offset requires that you touch off the face of the turret to a fixed location. Store the Z position and then touch off each tool to the same location and subtract the turret's touch off value from the tool's.

**Tool Nose Radius & Orientation**

- The intersection of the X and Z offset define a “virtual” tool tip. Graphic displays and position displays show position based on the virtual tool tip.
- Tool Nose Radius is used to compensate the tool path.
Tool Orientation

The orientation of the tool is necessary to determine where the virtual tool tip (if there is one) is in relation to the center of the tool. This diagram shows eight possible orientations. Orientations 5-8 are for tools that do not have a radius compensation value. They cut in one direction and are calibrated at the cutting tip.

Tool Probing

Manual and automatic methods for calibrating the tool diameter and length will be provided. Tool Nose Radius will be probed as well and Tool X and Z Offsets.

Diameter/Length Wear Compensation

Incremental value added to the tool diameter or length.

These will be taken from a diameter and length offsets table.

The programmer will be able to specify an index into the offset table for any tool. This gives the operator the ability to have multiple offsets from nominal for machining at different diameters on the part.

Tool Library

Tool Orientations will have an associated graphics image to aid in selection.

The programmer will be able to select the cutter shape/type from a menu of standard shapes. (this is cutter insert shape not orientation of tool)

Leading and trailing angles for the tool (this includes holder) will be defined for each tool. This will be necessary if we choose to provide recessed profiles.

Maximum Depth Of Cut (Used as default value for conversational programs).
Coolant Control

Features for commanding coolant on demand and based on position will be required. Each tool can be programmed with or without automatic coolant control. When automatic coolant control is defined for a tool the coolant will turn on at initial X, Z positions for conversational programming.

The programmer may also command coolant directly using an M Code data block.

The programmer may also override coolant commands directly using the console coolant buttons.
NC PROGRAMMING

This document describes the programming codes supported by the Max® software and how to use them in a part program. Descriptions and examples show how to use G codes, M codes, and T codes with Max® software.

Definitions

- **Block**—a line of code in a part program. Several blocks of code make up an entire part program. To continue a block onto the next line, put a `/` (back slash) at the end of the line that is being continued.

- **Modal**—a programmed command that Max® remembers until it is canceled by another programmed command from the same modal group. For example, programming G20 Inch mode and G90 Absolute mode in the first block of a part program tells Max® to remain in inch mode until G21 Millimeter mode is programmed, and to remain in Absolute mode until G91 Incremental mode is programmed. A modal command stays in effect until another command from the same group is programmed.

- **Integrator**—supplier of Max® software, including machine builders, machine suppliers, soft logic programmers.

- **Machine envelope**—the boundaries established by the axis overtravel switches. If you ever move an axis onto one of these switches, Max® hardware generates an emergency stop.

- **Reference position (or Machine home)**—the position where all axes position at the end of a sync cycle. The coordinates of this position establish the coordinates from which all machine coordinates are measured.
**Command Overview**

All part programs consist of a combination of the following items:

- **M Code**—Performs miscellaneous functions such as turning on the spindle, turning on coolant, specifying a program stop, or an end of program.
- **S Code**—Set the spindle speed.
- **F Code**—Sets the modal feed rate for cutting moves.
- **T Code**—Identifies the active tool and activates the offsets for the current tool.
- **E Code**—Specifies a fixture offset number to move part zero to a position that is convenient for the machine operator.
- **G Code**—G codes have two basic functions:
  - specify a modal condition (example: G20 establishes Inch mode, G21 establishes Millimeter mode, G90 establishes Absolute mode and G91 establishes Incremental mode).
  - specify the type of tool motion (example: G00 programs a rapid move, G01 programs a linear feed move, G02 and G03 program circular moves).
- **X, Z, I, K**—Coordinates for programming geometric information that Max® needs to determine the endpoint of a motion command.
- **N Numbers and O Numbers**—Each block may contain a nine-digit integer N Number and O Number. N and O numbers are not required, as Max® does not use these numbers. However, part programmers often use them as sequence numbers to order the part program. However, Max® does not require that the numbers appear in order within the part program.
- **Comment Statements**—Comment statements provide information to any person who reads the part program. You can insert comment statements at the end of any part program block by enclosing the comment within parentheses. You may make an entire block of code a comment statement by enclosing it within parentheses. Default behavior for Max® is to display each comment statement at the top of the Max® message window as its block is executed. This feature can be disabled by setting an Max® variable.
- **Loop (or Repeat) commands**—Data enclosed in parentheses is normally treated as a comment by Max®. The only exception to this is the Loop (or Repeat) command. The Loop command, which must be enclosed in parentheses, causes a specified part of the part program to repeat a specific number of times. Loop commands are not available on all controls; the feature must be enabled by your integrator.
- **Block Delete**—Block Delete codes specify blocks of code that are skipped when the Block Delete control feature is enabled. The specified block, or portion of a block, begins with a / (forward slash).
  The Block Delete control is activated from the Operator Control window.
- **Subroutines**—The subroutine feature allows you to call subroutines from either the same part program or a separate file.
• **Macros**—The macro feature allows you to pass information from a part program to a separate macro program. Macro programs are written in Visual Basic® language and then compiled into executable programs.

• **Spindle Designators**—Spindle designators are used in programs that control multiple spindles. They allow you to program a command for a specific spindle.

### How Your Integrator Can Customize Max® Software

The numeric value of each function performed by each G and M code in Max®'s programming system can be modified or redefined by your integrator in order to make the control more compatible with part programs that have been written for a CNC built by another manufacturer.

The most common example of this modification is changing the G codes that select inch and millimeter modes from G20/G21 to G70/G71.

⚠️ Your integrator is responsible for informing you about any changes they have made to Max®'s part programming system. This manual only describes the Standard G and M codes that have been assigned by Hurco Companies, Inc.

### Format for Part Programs Prepared Off-line (CAM System)

This section describes how a person writing an NC post processor should configure part program code for Max®. The two items that are important to a post processor are: the order in which Max® processes a block of code and the data format for each code.
How Max® Processes a Block of Code

A block of code can contain 0 to 10 G codes and 0 to 10 M codes. The table below shows the order in which Max® will process a block of part program code. Except for block delete codes, which are processed at the point they appear in a block, the order in which each entry appears within a block of code does not affect the order in which the data is processed.

<table>
<thead>
<tr>
<th>Description</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>N and O Numbers</td>
<td>N123400000</td>
</tr>
<tr>
<td>Comment Statements</td>
<td>(this is a message)</td>
</tr>
<tr>
<td>Block Delete Codes</td>
<td>/., /0, /4</td>
</tr>
<tr>
<td>Subroutines, Macros</td>
<td>SUBCALL, SUBFILE, MACROFILE</td>
</tr>
<tr>
<td>Job Syncing M Codes</td>
<td>M100 - M199</td>
</tr>
<tr>
<td>Specific G Codes</td>
<td>G201, G202, G40-42, G07-08, G20-21, G90-91, G59, G53, G58, G92, G93-95, G96-97</td>
</tr>
<tr>
<td>Feedrate Override</td>
<td>M48, M49</td>
</tr>
<tr>
<td>Beginning of Block M Codes</td>
<td>M07, M08</td>
</tr>
<tr>
<td>(configured by integrator with defMBef())</td>
<td></td>
</tr>
<tr>
<td>Tool and Fixture Commands</td>
<td>T code, E code</td>
</tr>
<tr>
<td>After Tool Change, Before Motion M Codes</td>
<td>M03, M04</td>
</tr>
<tr>
<td>(configured by integrator with defMRpm())</td>
<td></td>
</tr>
<tr>
<td>Spindle Speed Command</td>
<td>S code</td>
</tr>
<tr>
<td>Probing Setup</td>
<td>G07, G08</td>
</tr>
<tr>
<td>Dwell Command</td>
<td>G04</td>
</tr>
<tr>
<td>Feedrate or Dwell</td>
<td>F code</td>
</tr>
<tr>
<td>Motion Commands</td>
<td>G00, G01, G02, G03, G33</td>
</tr>
<tr>
<td>Exact Stop</td>
<td>G09</td>
</tr>
<tr>
<td>End of Block M Codes (configured by machine</td>
<td>M09</td>
</tr>
<tr>
<td>integrator with defMAft())</td>
<td></td>
</tr>
<tr>
<td>Spindle Stop</td>
<td>M05</td>
</tr>
<tr>
<td>End of Block System M Codes</td>
<td>M00, M01, M02, M30</td>
</tr>
<tr>
<td>Block Delete Synchronization</td>
<td>M200</td>
</tr>
</tbody>
</table>

Table 3–1. Order of Block Code Processing
How Max® Handles Unused Data

Any code that is not recognized by Max® will cause the control to stop execution and display an error message in the Message window.

Formats for Part Programs

Block data formats provide numerical values for the commands in a part program.

Distance and Feedrate Formats

Distance block data formats describe distances for axes and feedrates. The decimal point in linear distance formats can shift 1 or 2 places to the right or left of the default setting, depending on how a variable has been set by the machine integrator. This variable determines the range of numbers you can use when writing a program for a particular machine tool. Your machine integrator will tell you which range to use. The table below shows the different ranges for shifting distance formats.

<table>
<thead>
<tr>
<th>Blocks</th>
<th>Optionala (inch)</th>
<th>Default (inch)</th>
<th>Optional (mm)</th>
<th>Default (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometric Entries X Z I K R (range)</td>
<td>(±84.50000)</td>
<td>3.4 (±845.0000)</td>
<td>4.4 (±2146.3000)</td>
<td>5.3 (±21463.0000)</td>
</tr>
<tr>
<td>Feedrate (feed/min) and (feed/rev) F (range)</td>
<td>2.5 (00.000001 - 84.50000)</td>
<td>3.4 (000.0001 - 845.0000)</td>
<td>4.4 (0000.001 - 2146.3000)</td>
<td>5.3 (0000.001 - 21463.0000)</td>
</tr>
<tr>
<td>Thread Lead (distance/rev) K (range)</td>
<td>1.7 (0.0000001 - 1.000000)</td>
<td>2.6 (0.0000001 - 10.000000)</td>
<td>2.6 (0.0000001 - 24.000000)</td>
<td>3.5 (0.0000001 - 254.000000)</td>
</tr>
</tbody>
</table>

a. The decimal point in linear distance formats can shift 1 or 2 places to the right or left of the default setting, depending on how a variable has been set by the machine integrator. This option shows the decimal point moved 1 place to the left.

Table 3–2. Most Common Block Data Formats

Format: n.m means n number of places to left of decimal point and m number of places to right of decimal point.

For example, 3.4 is 3 places to the left of the decimal point and 4 places to the right.
Other Block Data Formats

Other block data formats are for programming G and M codes, rate of component motion or dwell, sequence numbers, tool codes, and fixture offsets.

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
<th>Format (range)</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>G Code</td>
<td>5 (0-65535)</td>
</tr>
<tr>
<td>M</td>
<td>M (Miscellaneous) Code Job Syncing M Codes</td>
<td>5 (0-65535) 3 (100-199)</td>
</tr>
<tr>
<td>S</td>
<td>Spindle Speed (rpm) and (css)</td>
<td>9 (0 - 2 x 10⁹) (no practical limit)</td>
</tr>
<tr>
<td>T</td>
<td>Tool Code + Offset T00 cancels tool offsets</td>
<td>0 (0-999999999)</td>
</tr>
<tr>
<td>E</td>
<td>Fixture Offset Code E00 cancels any fixture offsets</td>
<td>4 (0-9999)</td>
</tr>
<tr>
<td>N O</td>
<td>N and O Numbers</td>
<td>9 (0-999999999)</td>
</tr>
<tr>
<td>F</td>
<td>G93 inverse mode G94 FPM Mode G95 FPR Mode</td>
<td>4.3 inverse minutes (0.001 - 9999.999) See previous table See previous table</td>
</tr>
<tr>
<td>F</td>
<td>Dwell in G93 inverse mode Dwell in G94 fpm mode Dwell in G95 fpr mode</td>
<td>3.4 seconds (0.0001 - 999.9999) 3.4 seconds (0.0001 - 999.9999) 3.4 revs (0.0001 - 999.9999)</td>
</tr>
</tbody>
</table>

*Table 3–3. Other Block Data Formats*

**Format:** n means number of digits. n.m means n number of places to left of decimal point and m number of places to right of decimal point. For example, 3.4 is 3 places to the left of the decimal point and 4 places to the right.

Specifying Spindles

Max® allows you to specify whether a spindle G or M command applies to one spindle, or all spindles currently in use. To specify a command for a specific spindle, include a spindle designator with the G or M code.
General Format for Creating Part Programs

The code below shows what a typical part program looks like when you enter it using a text editor.

```
N20 G90 G40 G94 T0000          (absolute mode, CRC off, reset tool offset)
N30 G00 X2.5 T0101            (rapid to X, tool 1 offset 1)
N40 Z3                       (rapid to Z)
N60 X.5                      (rapid to X)
N70 G96 G07 R.5 S350 M03      (CSS, radius programming, X axis at .5in.)
                               (350 SFM and clockwise)
N80 G01 G95 F.01 X0           (face, IPR 0.01)
N120 G00 Z4                   (rapid to Z)
N130 X.5 Z3                   (rapid to XZ)
N160 G01 X1 Z2                (taper)
N170 X1.7                    (face)
N180 Z0.3                   (turn)
N190 X1.9                    (face)
N200 Z0                     (turn)
N230 G00 X2.5                (rapid to X)
N240 Z3                     (rapid to Z)
N900 M30                    (end of program, rewind)
```
Coordinate System Introduction

The coordinate system for a typical back turret lathe is shown below.

The most important part of the illustration is the outer most rectangle, which shows boundaries defined by the position of the axes overtravel limit switches. A high limit and low limit switch establishes the maximum travel for each axis. The term **machine envelope** refers to the boundaries established by these switches. If you ever move an axis onto one of these switches, Max® will generate an emergency stop.

The **reference position** shows the point where the axes position at the end of a **reference zero** cycle (also known as "syncing"). The axes may reference anywhere within the machine's envelope. Your machine integrator will show you how to reference your machine.

The dashed lines in the following drawing show the location of the machine's **software travel limits**. These limits are set by the machine integrator and are measured from the machine's reference position. Attempting to move an axis past a software travel limit will cause Max® to stop all axes motion, but it will not generate an emergency stop.

![Coordinate System for Typical Back Turret Lathe](image)

Your machine integrator may provide a third level of protection by defining slowdown limits located inside of the software travel limits. Moving an axis within its slowdown limit will cause Max® to restrict the move rate to a limit established by the machine integrator.

The point labeled **E01** (recorded in the fixture offset table) in drawing above is a point that you define to tell Max® where you want part zero to be located. You can move the fixture offset anywhere within the machine envelope. You may store up to 9999 different fixture offset positions in Max® and activate the one that you need to use.
G07/G08 Radius and Diameter Programming

The control has the ability of programming X axis dimensions using the radius or diameter of the part. Program G07 when you wish to program radius dimensions; program G08 when you wish to program diameter dimensions.

The table below provides more information about how the modal setting of will affect your part programs.

<table>
<thead>
<tr>
<th>Programmed Value</th>
<th>G07 Modal</th>
<th>G08 Modal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint of linear move</td>
<td>X use radius</td>
<td>X use diameter</td>
</tr>
<tr>
<td>Endpoint of circular move</td>
<td>X use radius</td>
<td>X use diameter</td>
</tr>
<tr>
<td>Center and radius of circular move</td>
<td>IKR use radius</td>
<td>IKR use radius</td>
</tr>
<tr>
<td>X axis infeed distances</td>
<td>infeed distance use radius</td>
<td>infeed distance use radius</td>
</tr>
</tbody>
</table>

Table 3–4. Effects of Using G07 and G08

Fixture Offset E Code

There are three options for positioning Z0 at a convenient position along the Z axis. The option used depends on your machine-tool configuration:

- All machines can use G92 to establish the part program coordinates at the current position without generating any tool motion.
- Depending on how the machine integrator has set variables:
  - Fixture Offset Tables will be fully functional.
  - Operator will not be able to directly enter data into the Fixture Offset table. However, the machine integrator will provide some method that allows the operator to enter a Z axis offset value into Fixture Offset #1 (E1). Once this value has been entered, Max® will automatically activate the offset whenever a nonzero Tool offset is active. Max® will remove the Z axis offset when the Tool offset is cancelled.
Absolute and Incremental Dimensions

Before starting to create your own part programs, you should be familiar with programming absolute (G90) or incremental (G91) dimensions. Max® provides both methods so the part programmer can enter part program dimensions from the blueprint with a minimal amount of effort.

Absolute Dimensions

As shown in the figure below, absolute dimensions are measured from part zero. The figure shows the path that the tool center will follow; the actual part will be smaller by the radius of the tool.

⚠️ Do not run this part on your machine; it is only intended to demonstrate the use of absolute dimensions.

Select absolute dimension programming by entering G90 in the part program. This setting remains active until you program G91 to select incremental dimension programming.
Example—G90 Absolute Dimensions

N10  (msg, lathe doc sample, abs G42, tool radius 0.1)
N20  G90 G40 G94 T0000  (absolute mode, CRC off, offset 0)
N30  G07 G00 X2.5 T0101  (radius programming,)
N40  Z3  (rapid to Z)
N60  X.5  (rapid to X)
N70  G96 R.5 S350 M03  (CSS, X axis at .5in,)
N80  G01 G95 F.01 X0  (face, IPR 0.01)
N120 G00 Z4  (rapid to Z)
N125 G42  (CRC right, skip if block del on)
N130 X.5 Z3  (rapid to XZ)
N160 G01 X1 Z2  (taper)
N170 X1.7  (face)
N180 Z0.3  (turn)
N190 X1.9  (face)
N200 Z0  (turn)
N230 G00 G40 X2.5  (CRC off, rapid to X, skip if)
N240 Z3  (rapid to Z)
N900 M30  (end of program, rewind)
Incremental Dimensions

As shown in the figure at below, incremental dimensions specify the distance that the tool must move during each block (i.e. the distance from the start of the move to the end of the move).

⚠️ Do not run this part on your machine; it is only intended to demonstrate the use of absolute dimensions.

![Figure 3–3. Incremental Dimensions](image)

You select incremental dimension programming by entering G91 in the part program. This setting will remain active until you program G90 to select absolute dimension programming.

Most part programs should begin with a **G90** Absolute command that moves the tool to a fixed position on the machine. Once the start point is defined, the part programmer can study the part blueprint and determine whether to use **G90** Absolute or **G91** Incremental dimensions.
Example: G91 Incremental Dimensions

N10  (msg, lathe doc sample inc G42 tool rad 0.1)
N20  G90 G40 G94 T0000  (absolute mode, CRC off, offset 0)
N30  G07 G00 X2.5 T0101  (radius programming,)
     (rapid to X, tool 1 offset 1)
N40  Z3  (rapid to Z)
N60  G91 X-2  (incremental mode, rapid to X)
N70  G96 R.5 S350 M03  (CSS, X axis at .5in,)
     (spindle speed 350 and clockwise)
N80  G01 G95 F.01 X-.5  (face, RPM 0.01)
N120 G00 Z1  (rapid to Z)
N125 G42  (CRC right, skip if block del on)
N130 X.5 Z-1  (rapid to XZ)
N160 G01 X.5 Z-1  (taper)
N170 X.7  (face)
N180 Z-1.7  (turn)
N190 X.2  (face)
N200 Z-.3  (turn)
/N230 G00 G40 X.6  (CRC off, rapid to X, skip if)
     (block del on)
N240 Z3  (rapid to Z)

N900 M30  (end of program, rewind)
Drill Cycles on Any Axis

Max® can perform drill cycles along any linear axis. Drill cycles axes may be configured to allow only cycles with negative drill axis feed, positive drill axis feed, or both.

Drill Cycle Axes

The machine integrator specifies in the Tune file which axes are allowed as drill cycle axes. The machine integrator can also specify the default drill cycle axis. If the default axis is not specified, and the drill cycle axes are not configured in the Tune file, the default drill cycle axis for job streams with 3 or more axes is the 3rd axis. Job streams with less than 3 axes use their highest numbered axis (i.e. Z on a lathe) as the default drill cycle axis.

If a job stream has one or more axes as allowed drill cycle axes, and a default drill cycle axis is not specified, the highest numbered allowed axis will be the default drill cycle axis. The drill cycle axis will be restored to its default value at the end of a program (M2 or M30), and by a complete sequencer reset.

If two or more axes can be assigned, one and only one of their axis letters must appear in the first drill cycle block (following a G80).

Drill cycle blocks following the first block may specify any axis. Allowed drill cycle axes, which are not the active drill cycle axis, will be treated as positioning axes and will rapid to their programmed positions before the drill cycle axis is moved.

On a 2-axis lathe, when no axes are specified in the Tune file as allowed drill cycle axes, the default drill cycle axis is the 2nd axis, typically Z. It will remain the active drill cycle axis, since no other axes can be assigned.

Change Active Drill Cycle Axis

The G80 code cancels the active drill cycle and modal parameters, and allows the drill cycle axis to be changed. The first drill cycle block following a G80 will change the active drill cycle axis if the block contains an axis letter of an allowed drill cycle axis. This drill cycle axis will remain active until the next G80.

Drill Cycle Axes and the Active Plane

Typically, drill cycles on a lathe are run unrestricted by the active plane. The machine integrator configures each allowed drill cycle axis to be either unrestricted or restricted by the active plane. Restricted by the active plane means that only a drill cycle axis perpendicular to the active plane can be used. An axis is perpendicular to the active plane when it is not parallel to either of the two axes which define the plane. The machine integrator assigns which axes are parallel in the Tune file.

The first drill cycle block can assign a drill cycle axis which is either unrestricted, or restricted but perpendicular to the active plane. If no allowed drill cycle axis can be assigned, the previous drill cycle axis is retained.
**G00 - Linear Motion at Rapid (default)**

**G00** sets rapid linear motion as the machine's modal condition. Any block that executes while **G00** is modal will cause the axes to rapid to the programmed endpoint. However, Max® will control the rate of each axis to ensure that all programmed axes reach their respective endpoints simultaneously.

**G00** is cancelled by any G code in modal group 1.

**G01 - Linear Motion at Feed**

**G01** sets feed linear motion as the machine's modal condition. Any block that executes while **G01** is modal will cause the programmed axes to feed directly to their programmed endpoint. Max® uses the programmed rate to adjust the feedrate of each axis to ensure that all axes reach their endpoints simultaneously.

**Example:**

Assume G94 is active, if the X axis is programmed at a feedrate of 10 inches per minute (G01 X2 F10), the X axis feedrate will be 10 inches per minute. If both X and Z axes are programmed to move an equal distance at 10 inches per minute (G01 X2 Z2 F10), the X and Z axes will each move at 7.071 inches per minute to ensure that the feedrate along the cutting path is 10 inches per minute.

**G01** is cancelled by any G code in modal group 1.

**G02/G03 - Clockwise/Counterclockwise Circular Motion at Feed**

**G02** sets clockwise circular interpolation as the modal condition. **G03** sets counterclockwise circular interpolation as the modal condition. There are two methods for defining an arc; program the arc's endpoint and center coordinates, or program the arc's endpoint and radius.

**Program Arc with Endpoint and Center Coordinates**

This method requires you to define the arc's endpoint and center coordinates. Dimensions for the arc's endpoint (X, Z) may be defined with absolute or incremental dimensions, depending on the modal G90/G91 condition. However, all arc center dimensions (I, K) must be defined using signed incremental dimensions (the distance from the arc's start point to the arc's center).
If **G07** Radius Programming is active, the arc's endpoint (X) and center (I) coordinates must be programmed as radius values.

If **G08** Diameter Programming is active the arc endpoint (X) must be programmed as a diameter value, but the arc center (I) must be programmed as a radius value.

**Example—G02/G03 Arcs**

The following example part program shows arcs programmed with X, Z, I, and K.

```
N1 G90 G20 G40
N2 G00 G07 X.0 Z.1 (G07 = radius programming)
N3 G01 Z0 F10
N4 X.2
N5 G03 X.8 Z-.6 I0 K-.6
N6 G01 G91 Z-1.0
N7 G90 G08 G02 X1.9512 Z-2.0243 R.6 (G08=diameter programming)
N8 G01 X2.2 Z-2.1487
N9 Z-2.5
N10 G00 X2.5
N11 Z.1
N12 M30
```

![Figure 3–5. G02/G03 Arcs Example](image_url)
Program Arc with Endpoint and Radius
The second method of programming a G02 or G03 block is to define the arc's endpoint and radius. As shown in the following Figure, there are two possible arcs when you program X, Z, and R. The sign of the R radius value determines whether you want the arc that spans less than 180° (by programming R as a positive value), or the arc that spans more than 180° (by programming R as a negative value).

Figure 3–6. Arc with Endpoint and Radius

G04 - Dwell

G04 programs a pause in part program execution.

<table>
<thead>
<tr>
<th>If G code active is:</th>
<th>Then F in the G04 block programs:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>G93</strong> Inverse Time Feed or G94 Feed Per Minute</td>
<td>The number of seconds to dwell. For example, the block <strong>G04</strong> F2.1 will cause Max® to pause for 2.1 seconds before it executes the next block of code.</td>
</tr>
<tr>
<td><strong>G95</strong> Feed Per Revolution</td>
<td>The number of spindle revolutions to dwell. For example, the block <strong>G04</strong> F10 will cause Max® to pause for 10 spindle revolutions before it executes the next block of code.</td>
</tr>
</tbody>
</table>

F is the only entry allowed in a G04 block.
G07/G08 - Radius Programming/Diameter Programming

**G07**, Radius programming, establishes a modal setting that tells the control to interpret all cross-slide dimensions in radius values. **G08**, Diameter programming, specifies that all cross-slide positions are interpreted in diameter values. In the following Figure, the arc center dimension (I) stays the same for both commands, but the cross-slide dimension (X) changes.

**G07 Radius Programming**

![Figure 3–7. G07 Radius Programming](image)

**G08 Diameter Programming**

![Figure 3–8. G03 Diameter Programming](image)

The program may switch between G07 Radius and G08 Diameter at any time.
**G09 - Exact Stop**

**G09** commands Max® to make sure that the tool reaches its programmed endpoint before it transfers to the next block in the part program. Since **G09** is non-modal, you must program it in every block that requires the tool to come to a complete stop.

**G20 - Inch Mode**

This code puts the control into inch mode. If your part is dimensioned in inches, this command should be programmed in the first block of each part program to ensure that your part program dimensions are interpreted in the correct units.

**G21 - Millimeter Mode**

This code puts the control into millimeter mode. If your part is dimensioned in millimeters, this command should be programmed in the first block of each part program to ensure that your part program dimensions are interpreted in the correct units.

**Unit Display Modes**

**G20** and **G21** do not effect the Unit Display Mode as viewed in the position or offset windows. Their use is for being able to correctly interpret the correct units from the part program.

**G33 - Threading**

This code allows you to program a threading cycle. To program a threading move, program the thread’s endpoint (using Z only, X only, or a combination of X and Z for a taper thread). Next program the lead of the thread using the letter K (the lead along the Z axis) or I (the lead along the X axis). If you are programming a tapered thread (both X and Z axes move) you must program the lead (I or K) for the axis that has the longest move.

You can also start a thread at a given angle. The thread start angle specified by a 'P' value in a **G33** block, such as:

```
G0 X3.5 Z4   (Position for thread)
G33 Z1 K.2 P180 (Cut thread)
G1 X3.6   (Pullout of thread)
```

The angle value may range from -360.0000 to 360.0000. A zero angle starts the thread at the index mark of the spindle encoder, and is the default. The 'P' value is not modal and must appear in the same block as the **G33**.
G38 - Turret Probe

There are four cycles within Turret Probing. To differentiate between the table probe cycles, the parameter S (CycSel) is used.

- S0 Datum X or Z
- S1 Set/Adjust Fixture Offset (X, Z)
- S2 Measure Diameter (ID, OD:X) or Part Length (Z)
- S3 Adjust Tool Offsets (X, Z)

All turret probe cycles are defined by a common G-code defined in the application tune file. Throughout the rest of this document, we will use G38 to specify a turret probe cycle.

The X and Z positions passed to any turret probe cycle must be in part coordinates to get the desired results.

Turret Probe Calibration - S0

- The purpose of the turret probe datum cycle is to define the datum offsets to be used for X and Z when measuring part surfaces in other turret probe cycles for most accurate results.
- The turret probe datum cycle will permit the user to specify which axis and direction (-X, -Z, +X, and/or +Z) to datum. In the other turret probe cycles, axis measurements are only permitted for those axes/directions that have been successfully calibrated.
- For the turret probe datum cycle, program CycSel0 (S0).
- One G38 macro call per axis and direction must be programmed on the Lathe to datum the probe. The surface touch point for the probe must be known and specified in the macro. X and Z must be pre-positioned before the datum macro call. Only the "probe align axis" moves during the cycle. When the cycle is finished, this "probe align axis" will be back at its macro start point.

Format of Sample Recorded Datum Cycle Data

hh:mm:ss Probe Datum: Probe #14; -X Offset 0.0827 at X2.5247, Z7.5000
### Parameters

<table>
<thead>
<tr>
<th>Letter Parameter</th>
<th>Definition</th>
<th>Alternative Word Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S</strong></td>
<td>0 - Specifies Turret probe calibration cycle</td>
<td>CycSel</td>
</tr>
</tbody>
</table>

**Must select one of the following: X, Z**

<table>
<thead>
<tr>
<th>Letter Parameter</th>
<th>Definition</th>
<th>Alternative Word Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>X</strong></td>
<td>Known datum contact point in part program coordinates for X</td>
<td>XDtmX</td>
</tr>
<tr>
<td><strong>Z</strong></td>
<td>Known datum contact point in part program coordinates for Z</td>
<td>ZDtmZ</td>
</tr>
</tbody>
</table>

**Must be specified in either tune file as the default or within the G38 block: A,F,T,V**

<table>
<thead>
<tr>
<th>Letter Parameter</th>
<th>Definition</th>
<th>Alternative Word Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
<td>Approach distance from expected contact point at which feed rate changes from programmed rate F to probe rate V</td>
<td>ApDst</td>
</tr>
<tr>
<td><strong>F</strong></td>
<td>Feedrate from approach point until contact</td>
<td>FRPrb</td>
</tr>
<tr>
<td><strong>T</strong></td>
<td>Tolerance about expected contact point</td>
<td>Tol</td>
</tr>
<tr>
<td><strong>V</strong></td>
<td>Feedrate to approach point and away from contact point</td>
<td>FRAp</td>
</tr>
</tbody>
</table>

**Optional**

<table>
<thead>
<tr>
<th>Letter Parameter</th>
<th>Definition</th>
<th>Alternative Word Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>P</strong></td>
<td>Probe Number</td>
<td>PrbNum</td>
</tr>
</tbody>
</table>

*Figure 3–9. Turret Probe Calibration—S0 Parameters*

To datum in X, **XDtmX (X)** is programmed. The current X position determines which direction the X datum is to occur.

To datum in Z, **ZDtmZ (Z)** is programmed. The current Z position determines which direction the Z datum is to occur.

If Probe Number is neither defined in the tune file nor programmed in the macro call, a value of P-1 is recorded.
**Macro Operation**

**X-Datum**

- X feeds (FRAp) to known contact point ± approach position (XDtmX ± ApDst).
- X is commanded to feed (FRPrb) to known contact point ± specified tolerance (XDtmX - Tol). Fault is flagged with the macro aborted unless probe contact is detected between (XDtmX - Tol) and (XDtmX + Tol). `axProbeSpnOffPos[X]` (datum done in +X direction) or `axProbeSpnOffNeg[X]` (datum done in -X direction) is set to (XDtmX - actual contact position for X) when valid contact detected.
- X feeds (FRAp) back to its start position.

**Z-Datum**

- Z feeds (FRAp) to known contact point ± approach position (ZDtmZ ± ApDst).
- Z commanded to feed (FRPrb) to known contact point ± specified tolerance (ZDtmZ ± Tol). Fault is flagged with the macro aborted unless probe contact is detected between (ZDtmZ - Tol) and (ZDtmZ + Tol). `axProbeSpnOffPos[Z]` (datum done in +Z direction) or `axProbeSpnOffNeg[Z]` (datum done in -Z direction) is set to (ZDtmZ - actual contact position for Z) when valid contact detected.
- Z feeds (FRAp) back to its start position.

**Datum in -X and -Z example:**

```
H00 E00
G00 G40 X10 Z5
G38 S0 X5 A.5 T.125 V20 F10
G0 Z10
G38 S0 Z7.6 A.5 T.125 V20 F10
```

**In the above example:**

If default values had been specified in the tune file that were appropriate for A, T, V, F for this datum cycle, the first G38 call block above could have been shortened to:

```
G38 S0 X5
```

- After contact is made during the X-datum at Z5, X returns to its start position (X10; F20).
- After contact is made during the Z-datum at X10, Z returns to its start position (Z10; F20).
- Start point when part program continues after the macro call is at X10, Z10 for this example.
Set/Adjust Fixture Offset (X,Z) - S1

- This cycle incrementally adjusts \((\text{Incr} = 1)\) or absolutely sets \((\text{Incr} \text{ not programmed or } 0)\) the fixture offset for the specified axis and E-code based on the measured surface of a part for that axis.

- The **absolute** fixture offset value stored will be equal to the measured surface for the given axis in **part coordinates** altered by the adjusted offset \((\text{XAdj or ZAdj})\) specified for the axis in the macro call block, if any.

- The **incremental** adjustment to the specified axis fixture offset table value equals the difference between the expected and actual contact points on the surface of a part for the designated axis.

- The programmer must pre-position the "other axis" to the probe align position. This "other" axis is not moved within the macro.

- The \(X\) or \(Z\) expected contact point is programmed in the macro call block. The corresponding "axis offset" parameter \((\text{I or K may})\) also be programmed if fixture offset entry is absolute. This "axis offset" parameter is the incremental distance from the contact point to the program zero point for the axis. For example, if you want to call the probe contact position 5 inches in program coordinates, program an \(I\) or \(K\) value of -5 inches.

- To measure in \(X\), program \(X\text{Exp} (X)\) and, optionally, \(X\text{adj} (I)\). \(X\text{adj} (I)\) may only be used with absolute fixture offset entries.

- To measure in \(Z\), program \(Z\text{Exp} (Z)\) and, optionally, \(Z\text{adj} (K)\). \(Z\text{adj} (K)\) may only be used with absolute fixture offset entries.

- If the cycle is aborted due to premature or no probe contact, the probing axis is fed back to its start position at the approach feed rate \((\text{FRAp, V})\).

**Format of "fixture offsets" cycle recorded data:**

\[hh:mm:ss \text{ Probe Fixture X: E19 new: 3.2629 old: 3.2600}\]

⇒ The expected contact point \((X, X\text{Exp})\) and fixture offset adjust number \((I, X\text{Adj})\) for \(X\) must always be programmed in the same units (radius or diameter) as is modally active \((G7, G8)\) in the control at the time the macro is called.
### Parameters

<table>
<thead>
<tr>
<th>Letter Parameter</th>
<th>Definition</th>
<th>Alternative Word Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>1 - Find part edge to adjust fixture offset</td>
<td>CycSel</td>
</tr>
</tbody>
</table>

**Must select one of the following: X, Z**

<table>
<thead>
<tr>
<th>X</th>
<th>X expected contact point in part program coordinates</th>
<th>XExp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>Z expected contact point in part program coordinates</td>
<td>ZExp</td>
</tr>
</tbody>
</table>

**Required**

| E                | E-code for which designated axis entry is to be set or altered | FixNum                    |

**Must be specified in either tune file as the default or within the G38 block: A,F,T,V**

<table>
<thead>
<tr>
<th>A</th>
<th>Approach distance from expected contact point at which feed rate changes from programmed rate F to probe rate V</th>
<th>ApDst</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Feedrate from approach point until contact</td>
<td>FRPrb</td>
</tr>
<tr>
<td>T</td>
<td>Tolerance about expected contact point</td>
<td>Tol</td>
</tr>
<tr>
<td>V</td>
<td>Feedrate to approach point and away from contact point</td>
<td>FRAp</td>
</tr>
</tbody>
</table>

**Optional**

<table>
<thead>
<tr>
<th>I</th>
<th>X fixture offset adjust number if absolute entry</th>
<th>XAdj</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>Z fixture offset adjust number if absolute entry</td>
<td>ZAdj</td>
</tr>
<tr>
<td>Q</td>
<td>Incremental fixture offsets if 1</td>
<td>Incr</td>
</tr>
</tbody>
</table>

*Figure 3–10. Fixture Offset (X,Z) - S1 Parameters*
**Macro Operation**

- The designated axis is programmed to move to the approach point (expected contact position + approach distance) at the approach feed rate \((\text{FRAp}, \text{V})\).
- This axis is programmed to feed at the probe feed rate \((\text{FRPrb}, \text{F})\) to the expected contact point + tolerance band \((\text{Tol}, \text{T})\).
- **Absolute entries:** When valid contact is made, the actual contact point in part coordinates plus the given axis adjust value \((\text{XAdj, I} \text{ or } \text{ZAdj, Z})\), if any, is stored in the fixture offset table for the specified E-code value and axis letter.
- **Incremental entries:** When valid contact is made, the measured contact point – expected contact point \((\text{XExp, X} \text{ or } \text{ZExp, Z})\) is added to or subtracted from the specified axis fixture offset table value for the specified E-code value and axis letter.
- When the cycle is finished, the measuring axis is back at its macro start position.
- The altered fixture offset does not become effective until the designated select code value is reactivated.

**Sample blocks**

```
T01 E00 H01
G00 X--- Z---
G38 S1 E3 Z0 K-1 A.5 T.125 V20 F10
```

**Measure Diameter or Part Length - S2**

- This cycle measures the outer or inner diameter \((X)\) or length \((Z)\) of a part and record the results (as specified by \text{defSEVProbeUse}).
- The software determines the align direction from the macro start position and expected contact point.
- The programmer must pre-position the "other axis" to the probe align position. This other axis is not moved within the macro.
- The expected contact point \((\text{XExp, X} \text{ or } \text{ZExp, Z})\) for the measuring axis \((X, Z)\) is programmed in the macro call block.
- If the cycle is aborted due to premature or no probe contact, the probing axis is fed back to its start position at the approach feed rate \((\text{FRAp}, \text{V})\).

**Format of "measure diameter or part length" cycle recorded data:**

```
hh:mm:ss Probe Measure X: machine: 3.2629 part: 3.2600
```
### Program Parameters

<table>
<thead>
<tr>
<th>Letter Parameter</th>
<th>Definition</th>
<th>Alternative Word Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S</strong></td>
<td>2 - Measure diameter (OD, ID: X) or length (Z) and record the results</td>
<td>CycSel</td>
</tr>
</tbody>
</table>

**Must select one of the following: X, Z**

<table>
<thead>
<tr>
<th>Letter Parameter</th>
<th>Definition</th>
<th>Alternative Word Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>X</strong></td>
<td>X expected contact point in part program coordinates</td>
<td>XExp</td>
</tr>
<tr>
<td><strong>Z</strong></td>
<td>Z expected contact point in part program coordinates</td>
<td>ZExp</td>
</tr>
</tbody>
</table>

**Must be specified in either tune file as the default or within the G38 block: A,F,T,V**

<table>
<thead>
<tr>
<th>Letter Parameter</th>
<th>Definition</th>
<th>Alternative Word Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
<td>Approach distance from expected contact point at which feed rate changes from programmed rate <strong>F</strong> to probe rate <strong>V</strong></td>
<td>ApDst</td>
</tr>
<tr>
<td><strong>F</strong></td>
<td>Feedrate from approach point until contact</td>
<td>FRPrb</td>
</tr>
<tr>
<td><strong>T</strong></td>
<td>Tolerance about expected contact point</td>
<td>Tol</td>
</tr>
<tr>
<td><strong>V</strong></td>
<td>Feedrate to approach point and away from contact point</td>
<td>FRAp</td>
</tr>
</tbody>
</table>

---

**Figure 3–11. Measure Diameter or Part Length - S2 Parameters**

The macro will automatically determine whether an OD or ID (X) is to be measured based on the direction of the expected contact point from the macro start point for X. The programmer must make sure that X and Z have been properly positioned before the macro is called.

The expected contact point \((X, X_{Exp})\) for X must always be programmed in the same units (radius or diameter) as is modally active \((G7, G8)\) in the control at the time the macro is called.
Macro Operation

- The designated axis is programmed to move to the approach point (expected contact position ± approach distance) at the approach feed rate \((\text{FRAp, V})\).
- This axis is programmed to feed at the probe feed rate \((\text{FRPrb, F})\) to the expected contact point ± tolerance band \((\text{Tol, T})\).
- When contact is made, the specified axis contact point in both machine and part coordinates is recorded as defined by \text{defSEVProbeUse}.
- When the cycle is finished, the given axis is returned to its macro start position \((\text{FRAp, V})\).

Sample blocks
T01 E00 H01
G00 X--- Z---
G38 S2 X0 A.5 T.125 V20 F10

Adjust Tool Offsets \((X, Z)\) - S3

- The purpose of this cycle is to measure the outer or inner diameter \((X)\) or length \((Z)\) of a part, adjust the specified tool offset table value, and record the results (as specified by \text{defSEVProbeUse}).
- The software determines the align direction from the macro start position and expected contact point.
- The programmer must pre-position the "other axis" to the probe align position. This other axis is not moved within the macro.
- The expected contact point \((X_{\text{Exp}}, X\) or \(Z_{\text{Exp}}, Z)\) for the measuring axis \((X, Z)\) is programmed in the macro call block.
- If the cycle is aborted due to premature or no probe contact, the probing axis is fed back to its start position at the approach feed rate \((\text{FRAp, V})\).

Format of "adjust tool offset" cycle recorded data:

hh:mm:ss Probe Offset Z: T13 new: 0.9468 old: 0.9437
Parameters

<table>
<thead>
<tr>
<th>Letter Parameter</th>
<th>Definition</th>
<th>Alternative Word Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>3 - specifies adjust tool offset (X, Z)</td>
<td>CycSel</td>
</tr>
</tbody>
</table>

Must select one of the following: \(X, Z\)

<table>
<thead>
<tr>
<th>Letter Parameter</th>
<th>Definition</th>
<th>Alternative Word Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>(X) expected contact point in part program coordinates</td>
<td>XExp</td>
</tr>
<tr>
<td>Z</td>
<td>(Z) expected contact point in part program coordinates</td>
<td>ZExp</td>
</tr>
</tbody>
</table>

Required

<table>
<thead>
<tr>
<th>Letter Parameter</th>
<th>Definition</th>
<th>Alternative Word Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>(H)-code for which the designated axis entry is to be altered</td>
<td>TONum</td>
</tr>
</tbody>
</table>

Must be specified in either tune file as the default or within the G38 block: \(A, F, T, V\)

<table>
<thead>
<tr>
<th>Letter Parameter</th>
<th>Definition</th>
<th>Alternative Word Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Approach distance from expected contact point at which feed rate changes from programmed rate (F) to probe rate (V)</td>
<td>ApDst</td>
</tr>
<tr>
<td>F</td>
<td>Feedrate from approach point until contact</td>
<td>FRPrb</td>
</tr>
<tr>
<td>T</td>
<td>Tolerance about expected contact point</td>
<td>Tol</td>
</tr>
<tr>
<td>V</td>
<td>Feedrate to approach point and away from contact point</td>
<td>FRAp</td>
</tr>
</tbody>
</table>

Figure 3–12. Adjust Tool Offsets \((X, Z)\) - S3 Parameters

The macro will automatically determine whether an OD or ID \((X)\) is to be measured based on the direction of the expected contact point from the macro start point for \(X\). The programmer must make sure that \(X\) and \(Z\) have been properly positioned before the macro is called. The expected contact point \((X, XExp)\) for \(X\) must always be programmed in the same units (radius or diameter) as is modally active \((G7, G8)\) in the control at the time the macro is called.
Macro Operation

- This cycle operation is identical to CycSel2 (S2). At the end of the cycle, however, this cycle incrementally adjusts the tool offset table entry (by difference between the expected and measured contact points). Also, the recorded message formats of the two cycles differ as noted above.

Sample blocks
T01 H00 E00
G00 X--- Y---
G38 S3 X1.5 H8 A.5 T.125 V20 F10

G40 - Cutter Radius Compensation Off (default)

This code turns off cutter compensation. When G40 is active, Max® will move the tool center along the path defined in the part program.

G41 - Cutter Radius Compensation Left

This code turns cutter compensation on to the left of the programmed part profile. When G41 is active, Max® will generate a tool path that is offset by the radius value stored in the tool table. The offset will be to the left side of the part profile.

G42 - Cutter Radius Compensation Right

This code turns cutter compensation on to the right of the programmed part profile. When G42 is active, Max® will generate a tool path that is offset by the radius value stored in the tool table. The offset will be to the right side of the part profile.

=G41 and G42 are similar to those used for milling a better approach may be to use Tool Orientations within the tool offset file. These are activated by using the tool offset call.
**Cutter Radius Compensation Activation**

The path taken by the tool varies depending on whether the cutting path is inside or outside the part, and then by the variables set by the machine integrator. The table below shows the most common use of the variables (both Activate and Exit Normal and Circular Joining are activated). The following Table shows the possible actions that can be performed by the machine tool.

For more information about the cutter radius compensation variables, see the "Max® Variable Dictionary".

<table>
<thead>
<tr>
<th>Max® Variable Sets</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate and Exit Normal on</td>
<td>Max® ensures that the tool feeds to a point that is perpendicular to the programmed start and end points.</td>
</tr>
<tr>
<td>Activate and Exit Normal off</td>
<td>The start and end points for the next tool motion are the current location of the tool.</td>
</tr>
<tr>
<td>Circular Joining on</td>
<td>Tool applies circular joining around outside corners.</td>
</tr>
<tr>
<td>Circular Joining off</td>
<td>Tool follows a series of linear blocks around outside corners.</td>
</tr>
</tbody>
</table>

*Figure 3–13. Activate and Exit Actions; Circular Joining Actions*
Examples

The following examples both use the part program below; the only difference is whether or not cutter compensation is turned on. The Figure on the left shows the tool path with cutter compensation turned off. The Figure on the right shows the tool path with **G42** (Cutter Compensation Right) turned on.

```
N10 (msg, lathe doc sample, abs G42, tool rad 0.1)
N20 G90 G40 G94 T0000 (absolute mode, CRC off, offset 0)
N30 G07 G00 X2.5 T0101 (radius programming, (rapid to X, tool 1 offset 1)
N40 Z3 (rapid to Z)
N60 X.5 (rapid to X)
N70 G96 R.5 S350 M03 (CSS, axis at .5in.,) (350 SFM and clockwise)
N80 G01 G95 F.01 X0 (face, IPR 0.01)
N120 G00 Z4 (rapid to Z)
N125 G42 (Cutter Comp Right)
N130 X.5 Z3 (rapid to XZ)
N160 G01 X1 Z2 (taper)
N170 X1.7 (face)
N180 Z0.3 (turn)
N190 X1.9 (face)
N200 Z0 (turn)
N230 G00 G40 X2.5 (CRC off, rapid to X)
N240 Z3 (rapid to Z)
N900 M30 (end of program, rewind)
```
Figure 3–14. Cutter Compensation Right Turned Off (G40)

Figure 3–15. Cutter Compensation Turned On (G42)
G53 - Program Machine Coordinates

This code allows you to program the axes to a position that you define using machine coordinates. This code causes the part program to ignore the following offsets for one block:

- active work coordinate offsets (G92)
- active tool offset
- active fixture offset

Before you program G53 in a block, the program must meet the following conditions:

- Cutter radius compensation (CRC) must be off.
- Absolute mode (G90) must be active.
- Either rapid (G00) or linear (G01) mode must be active.
G58 - Automatic Withdraw

This programs a move that executes when the machine operator presses the withdraw button. The machine integrator must configure your machine to use this feature. The withdraw feature is useful if you want to check a tool, or immediately stop cutting a part.

**G58** does not generate axis motion; instead it programs the values that will be used to move an axis immediately away from the part. After the withdraw move occurs, the control will be in feedhold. You can also program **G58** with **G53** in the same block so Automatic Withdraw will go to an absolute machine coordinate.

After using the automatic withdraw feature, the machine operator can use the withdraw feature, as described in the "Max® Machine Operator Manual", to continue moving the axis away from the part.

```
N10 G90 G00 Z0 X0
N50 G58 X3 (enable G58)
.
activate auto withdraw during program
.
N200 G00 X3
N210 G58 (cancel G58)
N220 M30
```

**Figure 3–16. G58 Automatic Withdraw**
When the operator starts the auto withdraw sequence, Max® will only move axes programmed in the G58 block. The only codes a G58 block may contain are:

- G90 (absolute mode)—program a move to absolute coordinates
- G91 (incremental mode)—program a move to incremental coordinates
- G53 (machine coordinates)—program a move to programmed machine coordinates (use G53 only with G90 mode)
- axis moves—only the programmed axes will move
- feedrate for the move

To cancel a G58, choose one of the following options:

<table>
<thead>
<tr>
<th>Source</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller</td>
<td>G58 block without any programmed axis moves or M30 End of program code.</td>
</tr>
<tr>
<td>soft PLC logic</td>
<td>soft PLC performs a complete reset on the controller.</td>
</tr>
<tr>
<td>Other conditions set by Integrator</td>
<td>Integrator can cancel G58 as part of the programmed action in M codes or soft PLC logic. For example, G58 can be canceled when the operator switches to jog mode.</td>
</tr>
</tbody>
</table>

**Table 3–5. Cancel G58 Options**

**G59 - Cancel Work Coordinate Offsets**

This code cancels any active offset set by G92 Work Coordinate Offsets.

**G78 - Threading Cycle**

This feature provides a multi-pass Threading Cycle for OD and ID threads. This cycle supports taper threads, face threads, lead in and lead out angles, multiple spring passes, and various methods of specifying depth of cut

To make the part program easier to read, you may program this cycle using single letter entries (e.g. A30, Z3.1) or multi-letter entries (angIn30, ZEnd3.1). The second method is easier to read, while the first method is typically output by CAM systems.

Prior to executing this cycle, the spindle must be running at speed in G97 mode and the tool must be positioned so it can rapid directly to the thread’s X and Z clearance position.
**Turn Vs. Face Threading**

The type of threading is determined by the **M** (Face) parameter.

<table>
<thead>
<tr>
<th>Parameter Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M=0</td>
<td>Indicates a Turn Thread.</td>
</tr>
<tr>
<td>M=1</td>
<td>Indicates a Face Thread.</td>
</tr>
</tbody>
</table>

Default Value: M=0

*Table 3–6. Face Parameter*

**Turn Threading - Inner Diameter Vs. Outer Diameter**

The type of thread cut (Inner or Outer Diameter) is defined by the signs of the **X** and **J** parameters. The **X** parameter (X, minR, minD) specifies the radius or diameter at the start of the final thread pass. The **J** parameter (J, TDepth) specifies the total depth of the thread.

- if (X, minR, minD) and (J, TDepth) are of the same sign (both positive or both negative), an OD (outer diameter or external) thread cut will be implemented.
- If (X, minR, minD) and (J, TDepth) are of the opposite sign (one positive and one negative), an ID (inner diameter or internal) thread cut will be implemented.

⇒ If the sign of **X+J** is not the same as the sign of **X**, the macro is aborted with an appropriate fault message displayed. The X-axis is not allowed to cross the spindle centerline when cycling a threading macro.
Required Parameters for Turn Threads

The following entries are required to define a turn thread.

Table 3–7. Turn Thread Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>JDepth</td>
</tr>
<tr>
<td>X</td>
<td>minR</td>
</tr>
<tr>
<td></td>
<td>minD</td>
</tr>
<tr>
<td>K</td>
<td>ZStart</td>
</tr>
<tr>
<td>F</td>
<td>FpR</td>
</tr>
<tr>
<td>F-</td>
<td>TpI</td>
</tr>
</tbody>
</table>

a. The interpretation of this parameter is not affected by the Max® Radius/Dimension Mode state (G7 or G8).
b. The interpretation of this parameter is not affected by the Max® Radius/Dimension Mode state (G7 or G8).

You must program the following to specify the length of the thread.

Table 3–8. Thread Length Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>LThrd</td>
</tr>
<tr>
<td>Z</td>
<td>ZEnd</td>
</tr>
<tr>
<td>H</td>
<td>ZTEnd</td>
</tr>
</tbody>
</table>

Figure 3–17. Thread Parameters Example
**Clearance, Chase In and Chase Out Parameters**

These parameters have default values and may not need to be programmed.

<table>
<thead>
<tr>
<th>U</th>
<th>XClr</th>
<th>X return clearance. Measured from the major diameter. Default 0.1&quot; or 2.54 mm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>ZClr</td>
<td>Z clearance. Measured from the start of thread. Default 0.1&quot; or 2.54 mm.</td>
</tr>
<tr>
<td>A</td>
<td>angIn</td>
<td>Lead in Angle. This is the infeed (chase in) angle with respect to the X axis. Default 0°</td>
</tr>
<tr>
<td>C</td>
<td>angOut</td>
<td>Lead out Angle. Pullout (chase out) angle with respect to X axis. Default 0°.</td>
</tr>
</tbody>
</table>

*Table 3–9. Clearance and Chase Parameters*

*Figure 3–18. OD Thread with Offsets U and W Example*

*Figure 3–19. OD Thread with Chase in (A) and Chase out (C) Example*
**Multiple Start Parameters**

These parameters allow you to define multiple starting points at different angles from 0 - 359.999° around the thread. This parameter is optional.

<table>
<thead>
<tr>
<th>Q</th>
<th>AngStr</th>
<th>Thread cut start angle. Default 0°.</th>
</tr>
</thead>
<tbody>
<tr>
<td>+Q</td>
<td>MSNum</td>
<td>Multiple Start number. Delta angle per thread start equals 360/Q.</td>
</tr>
</tbody>
</table>

*Table 3–10. Multiple Start Parameters*

![Diagram](image)

**G78 Q-120** + other parameters

**G78 Q-170** + other parameters

**G78 Q-180** + other parameters

*Figure 3–20. OD with multiple starts at 120°, 170°, and 180° Example*

![Diagram](image)

**G78 Q3** + other parameters

*Figure 3–21. OD with 3 Evenly Spaced Threads Example*
**Tapered Thread**

For a tapered thread, you must program \( B \) to specify the taper angle with respect to the Z axis for turn thread, or X axis for face thread.

<table>
<thead>
<tr>
<th>( B )</th>
<th>( \text{angTap} )</th>
<th>Taper angle relative to the Z axis. (-45^\circ \leq B \leq 45^\circ)</th>
</tr>
</thead>
</table>

*Table 3–11. Tapered Thread Parameter*

![Figure 3–22. Tapered Thread Example](image)

**Cutting Parameters**

The following cutting parameters affect how the thread will be cut. You can specify the start and final rough depth, number of finish passes, or the number of spring passes.

<table>
<thead>
<tr>
<th>( D )</th>
<th>( \text{SDepth} )</th>
<th>Start rough depth. <strong>Required.</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>( D )</td>
<td>( \text{FDepth} )</td>
<td>Final rough depth. If ( E ) is not programmed, the value defaults to ( D ) unless ( V ) is programmed.</td>
</tr>
<tr>
<td>( L )</td>
<td>( \text{finCut} )</td>
<td>Depth of cut per finish pass. If ( L ) is not programmed, then there are no finish passes. If ( P ) is nonzero and ( L ) equals zero, then the number of spring passes equals ( P + R ).</td>
</tr>
<tr>
<td>( P )</td>
<td>( \text{finNum} )</td>
<td>Number of finish passes implemented before starting spring passes. If ( P ) is not programmed and ( L ) is nonzero, set ( L ) to 1. Otherwise, no finish passes will occur.</td>
</tr>
<tr>
<td>( R )</td>
<td>( \text{SprNum} )</td>
<td>Number of spring passes at 0 depth per pass.</td>
</tr>
<tr>
<td>( V )</td>
<td>( \text{cVol} )</td>
<td>When set to 1, the depth per pass results in constant volume removal per pass. When programming constant volume, do not program a final depth parameter (( E )) in the same block.</td>
</tr>
</tbody>
</table>

*Table 3–12. Thread Cutting Parameters*
No Finish Pass Programmed—G78 X1.05 J.45 D.1 E.05

Only the start rough depth (D) and final rough depth (E) are programmed. The tool moves progressively deeper from D to E through a number of passes calculated by Max®.

![Diagram](image)

Figure 3–23. No Programmed Finish Pass Example

Three Finished Passes Programmed—G78 X1.05 J.45 D.1 E.05 L.01 P3

Three finish passes (L) are programmed along with the D and E values. The tool moves progressively deeper from D to E through a number of passes calculated by Max®, then moves in for three consecutive finishing passes.

![Diagram](image)

Figure 3–24. Three Programmed Finished Passes Example
**Sample Part Programs**
The following sample code shows four types of threads:

- Straight OD Thread, 3 - 43
- Straight OD Thread with Lead In/Lead Out Angles, 3 - 44
- Tapered OD Thread, 3 - 45
- Straight ID Thread, 3 - 46

⚠️ Do not run these sample part programs on your machine, they are intended only to demonstrate the types of threads. The drawings are for example only and are not to scale.
**Straight OD Thread**

This example shows a straight thread cut with:

- Variable depth per pass
- Two finish passes of .02" per finish pass
- Three spring passes

(Thread cut example - straight thread)

G70 G90 M03 S120
G0 Z4.1
G78 X.9 J.2 Z1.2 K2 F.15 D.04 E.03 P2 L.02 R3
M30

*Figure 3–25. Straight OD Thread Example*
**Straight OD Thread with Lead In/Lead Out Angles**

This example shows a straight thread cut with:

- Lead In angle (A) of 30°
- Lead Out angle (C) of 5°

(Thread cut example - straight thread with lead in/lead out angles)

G70 G90 M03 S120
G0 X4.1 Z2.5
G78 X.9 J.6 Z1.2 K2 A30 C5 F.15 D.04 E.03 P2 L.02
M30

*Figure 3–26. Straight OD Thread with Lead In/Lead Out Angles*
**Tapered OD Thread**

This example shows a tapered thread cut with a Taper angle (B) of 14.7°.

(Thread cut example - tapered thread)

G70 G90 M03 S120
G0Z4.1
G78 X.9 J.6 B14.7 Z1.2 K4 F.375 D.1 E.08 P2 L.02 R3
M30

![Figure 3–27. Tapered OD Thread Example](image-url)
**Straight ID Thread**

This example shows a straight thread cut with:

- Variable depth per pass
- Two finish passes of .02" per finish pass

(Thread cut example - straight ID thread)

G70 G90 M03 S120
G0 Z4.1
G78 X.9 J-.2 Z1.2 K2 F.15 D.04 E.03 P2 L.02
M30

![Figure 3–28. Straight ID Thread Example](image)

---

**Figure 3–28. Straight ID Thread Example**
**Threading Equivalents**

The following equivalents exist between the turn and face thread macro arguments.

<table>
<thead>
<tr>
<th>Thread Axis</th>
<th>Turn Thread</th>
<th>Face Thread</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thread Start</td>
<td>K, ZStart</td>
<td>I, XStart</td>
</tr>
<tr>
<td>Thread Normal End</td>
<td>Z, ZEnd</td>
<td>X, minR, minD</td>
</tr>
<tr>
<td>Optional Thread End</td>
<td>H, ZTEnd</td>
<td>S, XEnd</td>
</tr>
<tr>
<td>Optional Thread End #2</td>
<td>Thread Start + T, LThrd</td>
<td>Thread Start + T, LThrd</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Depth Axis</th>
<th>Turn Thread</th>
<th>Face Thread</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Pass Depth</td>
<td>X, minR, minD</td>
<td>Z, ZEnd</td>
</tr>
<tr>
<td>Initial Depth</td>
<td>Last Depth + J, TDepth</td>
<td>Last Depth + J, TDepth</td>
</tr>
</tbody>
</table>

*Table 3–13. Threading Equivalents between Turn and Face Thread Macros*

1. The thread start, last pass depth, and total depth arguments (K, X, and J for turn thread; I, Z, and J for face thread) must be programmed for each thread cycle.
2. Exactly one of the three thread arguments (Z, H, or T for turn thread, X, S, or T for face thread) must be programmed for the given type of thread cycle (turn or face).
**Face Threading**

The following entries are required to define a face thread.

<table>
<thead>
<tr>
<th>M</th>
<th>Face</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M1 is required to activate the face thread. No M-value or M=0 activates the turn thread. The M default value is 0.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>J</th>
<th>TDepth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total depth of cut (End depth = Z, Start depth = Z+J.)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Z</th>
<th>ZEnd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Z-Position at the start of the final thread pass at the end of the infeed motion.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I</th>
<th>XStart</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X-position where thread starts.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>F</th>
<th>FpR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pitch. Inches/millimeters per revolution.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>F-</th>
<th>TPI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lead. Threads per inch. (F-8 is same as F.125)</td>
</tr>
</tbody>
</table>

You must program the following to specify the length of the thread.

<table>
<thead>
<tr>
<th>T</th>
<th>LThrd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Length of thread between the infeed move and pullout move.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>minR, minD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The X end position at Z Clearance (Z + J + W).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S</th>
<th>XEnd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The X end position at Z start depth (Z+J).</td>
</tr>
</tbody>
</table>

*Table 3–14. Face Threading Parameters*
G90 - Absolute Programming (default)

G90 establishes a modal setting that tells the control to interpret all XZ endpoint dimensions in absolute coordinates. G90 Absolute Programming specifies that all tool endpoint positions are measured from the current part zero position.

Most part programs should begin with a G90 command to establish part zero. A program may switch between G90 Absolute mode and G91 Incremental mode at any time.

G91 - Incremental Programming

G91 establishes a modal setting that tells the control to interpret all XZ endpoint dimensions in incremental coordinates. G91 Incremental Programming specifies that all tool endpoint positions are measured from the tool’s position at the start of the motion. A program may switch between G90 Absolute and G91 Incremental at any time within a part program.

G92 - Work Coordinate Offsets or Spindle Max Speed

G92 lets you establish the part program coordinates at the current position without generating any tool motion.

Work coordinate offsets are cancelled using G59.

G92 can also set a maximum spindle speed for constant surface speed mode.

Example

The code G92 S5000 will prevent the spindle from exceeding 5000 rpm.

G93 - Inverse Time Feed

G93 puts the control into inverse time mode. When G93 is modal, the F value represents 1/(minutes of cut).

Any dwell command that is programmed while G93 is active will be interpreted as the number of seconds to dwell (the same as G94 feed per minute mode).

When programming short times for long distances, remember that an axis will not move faster than its maximum rate as set in the Tune file.

To determine the F value, use the formula:

\[ F = \frac{1}{t} \]
**Inverse Time F value = (Feedrate)/(Distance of Move)**

**Example 1**
To determine the F value for a four-minute cut, use this equation:

\[ F = \frac{1}{(4 \text{ minutes})} = 0.25 \]

and program a block with \( F = 0.25 \):

\[ \text{N100 G93 X2 F.25} \]

**Example 2**
To determine the F value for a 20-second cut, use this equation:

\[ F = \frac{1}{(20 \text{ seconds})} = \frac{1}{0.333333} \]

and program a block with \( F = 3 \):

A variable lets the machine integrator specify whether F is modal or whether it is required in every G93 block. Most controls require an F value in every G93 block.

---

**G94 - Feed per Minute (default)**

**G94** puts the control into feed per minute mode. When **G94** is modal, all F feedrate values are interpreted in inch/minute or millimeter/minute units. Any dwell command that is programmed while **G94** is active will be interpreted as number of seconds to dwell.

---

**G95 - Feed per Revolution**

**G95** puts the control into feed per revolution mode. When **G95** is modal, all F feedrate values are interpreted in inch/revolution or millimeter/revolution units. Any dwell command that is programmed while **G95** is active will be interpreted as number of spindle revolutions to dwell.
G96 - Constant Surface Speed (CSS)

**G96** sets feet per minute or meters per minute as the modal spindle command mode. When G96 is programmed, the spindle speed will increase as the tool tip moves toward the spindle centerline and decrease as the tool moves away from the spindle centerline. When **G96** is modal, the programmed S value is read by Max® as a feet per minute or meters per minute command. If the application uses multiple spindles, CSS is applied to all spindles in the same job stream.

To specify a radius of the tool tip that corresponds to the surface speed given in S, program the R value in the same block as G96.

- If you do not program an R value or the R value is 0, Max® uses the X-axis position as the radius.
- If an R value has been previously programmed, and you program a **G96** without an R value, the previously programmed R value will be used.

<table>
<thead>
<tr>
<th>Active Programming Mode</th>
<th>R Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>G07 Radius Programming</td>
<td>distance from the tool tip to the spindle centerline (radius)</td>
</tr>
<tr>
<td>G08 Diameter Programming</td>
<td>twice the distance from the tool tip to the spindle centerline (diameter)</td>
</tr>
</tbody>
</table>

**Table 3–15. R Values and G96 Programming**

CSS in Multiple Job Streams

**G96** controls CSS only for the job stream in which it is programmed. When CSS is active, it applies to all spindles in the job stream. Any axis in the job stream may be used to determine the CSS for the job stream, however it should be an axis that points at the centerline of the spindle.

The default axis on which CSS is based is the lowest numbered axis in the job stream. The axis may be changed by setting a value in the Tune file. Soft logic may also change the CSS axis.

G97 - Direct Spindle Speed (default)

**G97** sets direct RPM as the modal spindle command mode for the job stream in which it is programmed. When **G97** is modal, the programmed S value is read by Max® as revolutions per minute. If the application uses multiple spindles, Max® sets direct RPM for all spindles in the same job stream.

When using multiple spindles, the S code (commanded speed) applies to all spindles in a job stream unless it is used with a spindle designator. Using the spindle designator with an S code specifies an RPM for a specific spindle in the current job stream.
M Code Descriptions

The following M codes are available on most Max® systems. However, since the machine integrator is responsible for designing how each M code operates, you should refer to the documentation provided by the machine integrator.

To help ensure that your CNC performs commands as expected, review the final command definitions with the machine integrator.

M codes belong to one of four execution categories. The execution categories are listed below.

- Execute before tool change or motion (configured by machine integrator) - example: M08
- Execute after T and S codes, but before motion (configured by machine integrator) - examples: M03, M04
- Execute after tool change or motion (configured by machine integrator) - example: M09
- Execute after tool change or motion (system M codes) - examples: M00, M05, M30

Understanding each M code's execution category is important when the M code is programmed in a block that also contains a T code, S code, E code, and XZ axes motion. Within each category, M codes execute in the order defined by the machine integrator.

The M codes in this chapter perform the functions as described. However, the integrator can program additional events for some M codes, as noted in their descriptions. The integrator can also create M codes and assign them to one of the execution categories listed above.

M00 - Program Stop

**M00** causes:

- part program execution to stop
- the spindle to turn off
- the parser look-ahead program to finish processing the blocks in the parser look-ahead buffer and stop looking ahead for more blocks

The operator must press **CYCLE START** to resume program execution and restart the look-ahead program.
**M01 - Optional Stop**

M01 performs the same functions as M00, but can be toggled on or off through the OpStop button on the machine operator’s panel.

- **Off**—CNC ignores M01
- **On (locked)**—CNC acts on M01 and keeps it active
- **On (turn off)**—CNC acts on M01 then toggles it off

**M02 - End of Program (no rewind)**

M02 stops program execution and leaves the part program pointer on the block that follows the M02 command. M02 commands the spindle to turn off. M02 executes at the end of the block (after all motion occurs).

**M03 - Spindle Clockwise**

M03 tells the control to turn the spindle on in the clockwise direction. If the machine's spindle is controlled by Max®, your machine integrator is responsible for designing the logic that turns on the spindle. M03 will execute after T, S, and E codes execute, but before motion programmed in the block occurs.

An S code specifying the spindle speed must be programmed before the spindle will turn on. Max® has an advanced feature called RPM Look Ahead that reduces part cycle time by ramping the spindle up to speed before the feed move begins. This improves the cycle time because the control does not dwell while the spindle ramps up to the programmed speed. When this feature is active, it is possible to create a part program containing an M03 and spindle speed command that does not turn on the spindle (this usually occurs during start-up testing when the maintenance engineer programs a part to test the spindle but does not program a feed move). You may wish to discuss this option with your machine integrator.

Your machine integrator is responsible for designing the logic that controls the direction the spindle will turn when M03 and are programmed.

**M04 - Spindle Counterclockwise**

M04 is the same as M03 except it commands the spindle to rotate in the counterclockwise direction.
**M05 - Spindle Off**

M05 tells the control to turn the spindle off. If the machine’s spindle is controlled by Max®, your machine integrator is responsible for designing the logic that turns off the spindle. M05 will execute at the end of the block (after all motion occurs).

**M07 - Secondary Coolant On**

M07 commands the secondary coolant pump to turn on. Typically the second coolant pump is used to provide mist coolant. This command is cancelled by an M09 command. Your machine integrator is responsible for designing the logic to implement this M code. M07 is usually configured to execute at the beginning of the block (before any motion occurs).

**M08 - Primary Coolant On**

M08 commands the primary coolant pump to turn on. Typically, the primary coolant pump is used to provide flood coolant. This command is cancelled by an M09 command. Your machine integrator is responsible for designing the logic to implement this M code. M08 is usually configured to execute at the beginning of the block (before any motion occurs).

**M09 - Coolant Off**

M09 command tells the control to turn the primary and secondary coolant pumps off. Your machine integrator is responsible for designing the logic to implement this M code. M09 is usually configured to execute at the end of the block (after all motion occurs).

**M30 - End of Program (rewind)**

M30 stops program execution and moves the part program pointer to the top of the part program. M30 commands the spindle to turn off. M30 is configured to execute at the end of the block (after all motion occurs). The machine integrator may define additional events that occur when M30 executes.

**M48 - Use feedrate override**

M48 tells Max® to use the value specified by the Feedrate Override Slider (FPM %).
M49 - Ignore feedrate override

M49 tells Max® to ignore the value specified by the Feedrate Override Slider (FPR %) and use the programmed feedrate.

M200 - Block Delete Synchronization

M200 allows the operator to turn on block delete while the part program is executing. Using this code ensures that blocks marked with block delete codes are skipped after block delete has been turned on, without requiring the operator to stop and restart the program.
**Block Delete Code**

Block Delete will skip specified blocks of code when turned on.

**Programming Block Delete**

To skip a block of code, place a slash at the front of the block. Multiple block delete levels are allowed, but most controls only use one level, as shown in the example below.

```
N10 (msg, lathe doc sample, abs G42, tool rad 0.1)
N20 G90 G40 G94 T0000 (absolute mode, CRC off, offset 0)
N30 G00 X2.5 T0101 (rapid to X, tool 1 offset 1)
N40 Z3 (rapid to Z)
N60 X.5 (rapid to X)
N70 G96 G07 R2 S350 M03 (CSS, radius programming,)
   (part radius 2in.,)
   (350 SFM and clockwise)
N80 G01 G95 F.01 X0 (face, IPR 0.01)
N120 G00 Z4 (rapid to Z)
/N125 G42 (CRC right, skip if block del on)
   .
   .
   .
/N230 G00 G40 X2.5 (CRC off, rapid to X, skip if)
   (block del on)
N240 Z3 (rapid to Z)
N900 M30 (end of program, rewind)
```
To program different levels, place a slash and a single digit at the front of a block. The levels range from 0 to 9. When a level number is specified in the Operator Control window, all blocks containing the same level will be skipped.

/0N7 X3.0   (skip block if level 0 specified)
/1N7 X3.0   (skip block if level 1 specified)
/8/9N7 X3.0 (skip block if level 8 or 9 specified)

**Activating Block Delete**

The Max® parser look-ahead program processes blocks ahead of when they appear in the Part Program window. If an operator changes block delete mode from the Operator Control window while a part program is running, Max® may have already processed blocks marked for deletion. Max® will not go back and reprocess these blocks.

To ensure that block delete is applied to the marked blocks, Max® provides three M code options for synchronizing program execution with the parser look-ahead program (see below).

When programming with **M00**, **M01**, or **M200**, program the codes so they can be activated when the tool is off the part. Using these codes when in a cutting mode may cause the tool to leave dwell marks.
<table>
<thead>
<tr>
<th>M Code</th>
<th>What Max® Does</th>
<th>Required Operator Response</th>
<th>What Happens Next</th>
</tr>
</thead>
</table>
| M200 Block Delete Synchronization | - Waits for parser look-ahead buffer to empty.  
- Prevents parser look-ahead program from looking further ahead until after it processes the M200 code.                                                                 | None                        | Program execution **continues** and block deletes are recognized.                  |
| M00 Program Stop          | - Stops part program execution  
- Turns off the spindle  
- Waits for the parser look-ahead buffer to empty.  
- Prevents the parser look-ahead program from looking further ahead until program execution is re-started | Press Cycle Start           | Program execution **re-starts** and block deletes are recognized.                   |
| M01  > Optional Stop OpStop On | - Stops part program execution  
- Turns off the spindle  
- Waits for the parser look-ahead buffer to empty.  
- Prevents the parser look-ahead program from looking further ahead until program execution is re-started | Press Cycle Start           | Program execution **re-starts** and block deletes are recognized.                   |
| M01  > Optional Stop OpStop Off | - Waits for parser look-ahead buffer to empty.  
- Prevents parser look-ahead program from looking further ahead until after it processes the M01 code. | None                        | Program execution **continues** and block deletes are recognized.                   |

*Table 3–16. M COde Options for Look-Ahead Program Execution*
Programming a Tool Change and Activating Tool Offsets

This section describes how to program a tool change and activate the tool's offsets. Since the tool change sequence will vary from machine to machine, your machine integrator is responsible for informing you about the proper sequence for your machine. Basic information about Max®'s tool change sequence is provided below.

Tool Change Sequence

A typical tool change sequence is shown below.

\[ T0303 \]
\[ M03 S1200 \]

On most machines, the T code will cause the turret to index into position. It is the responsibility of the part programmer or machine operator to ensure that the turret is in a safe position to perform an index.

On some machines the part programmer must precede this sequence with the commands required to move the axes to the machine's tool change position.

The format for a T code is:

\[ T \text{mmmmmmnn} \]

where:

\[ \text{mmmmmm} \] is the tool number
\[ \text{nn} \] is the tool offset number

The last two digits identify the tool table offsets to use for the current tool. The preceding digits in the T code identify the physical tool that must be indexed into position. If your machine does not have an automatic turret indexer, or if your machine does not have an automatic tool changer, an M code is usually programmed and the T code or tool name will be listed in part program comments to inform the operator which tool must be inserted into the tool holder.

A \text{T00} cancels any tool offsets.
**Tool Motion Following a T Code**

Tool offsets are programmed with tool offset T codes. The coordinates for each tool offset are stored in the Max® Tool Offset table. All dimensions in the tool offset table represent dimensions measured from machine coordinate zero to the desired tool offset position. A tool offset value is provided for each axis that is installed on your machine. However, your machine integrator may disable the tool offset capability of any axis.

Execution of a tool offset T code does not cause tool motion unless axis motion is programmed in the block that contains the T code, or until motion is programmed in a following block. This applies to E codes as well.

To avoid incompatibilities between part programs created on different Max® controls, you should make it a part programming practice to program all axes in the block that programs a tool offset, or in the first motion block after a tool change.

Your machine integrator determines how tool motion following a tool offset will occur by selecting one of the options listed in the following Table. This selection also applies to the manner in which fixture offset E codes are implemented. The drawings below show how the tool will move depending on which option has been selected by your machine integrator.

<table>
<thead>
<tr>
<th>When this option is enabled by your machine integrator</th>
<th>It Causes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool Offset: Immediate Activation</td>
<td>All axes will move to a position relative to the new tool offset as soon as any axis is programmed.</td>
</tr>
<tr>
<td>Tool Offset: Deferred Action</td>
<td>Only the axes that are programmed will move to a position relative to the new tool offset.</td>
</tr>
</tbody>
</table>

*Table 3–17. Tool Offset Options*
Tool Offset Behaviors

![Diagram of Tool Offsets]

**Tool Offsets:***

**Tool Offset: Immediate Action**

N1 G90 T0101
N2 G01 Z0.5

**Tool Offset: Deferred Action**

N1 G90 T0101
N2 G01 Z0.5
N3 X0

**Fixture Offset**

N1 G90 T0202
N2 G01 X0 Z0

*Figure 3–29. Tool Offset Behaviors*
E Codes

Offsets from machine zero are programmed with Fixture Offset E codes. The coordinates for each fixture offset are stored in the Max® Fixture Offset table. All dimensions in the Fixture Offset table represent dimensions measured from a machine coordinate to the desired fixture offset position. E00 cancels any fixture offsets. A fixture offset value is provided for each axis that is installed on your machine. However, your machine integrator may disable the fixture offset capability of any axis.

Execution of a Fixture Offset E code does not cause tool motion unless axis motion is programmed in the block that contains the E code, or until motion is programmed in a following block. Your machine integrator determines how tool motion following a fixture offset will occur by selecting one of the options described in this chapter.

Fixture Offset Immediate Activation

When this option is selected, all axes will move to a position relative to the new fixture offset as soon as any axis is programmed.

![Figure 3–30. Fixture Offset: Immediate Activation](image)
Fixture Offset Deferred Activation

When this option has been enabled by your machine integrator, only the axes that are programmed will move to a position relative to the new fixture offset.

⚠️ To avoid incompatibilities between part programs created on different Max® controls, you should make it a part programming practice to always program all axes in the block that programs an \texttt{E} code.

\textit{Figure 3–31. Fixture Offset: Deferred Activation}
G73 - Peck Drill with Chip Break Drill Cycle

**Machine Integrator Configurable**

G73 sets chip break drilling as the modal drill cycle. The tool motion for the chip break drill cycle is shown in the Figure below. This cycle causes the tool to feed by the amount programmed in Q, then rapid back an incremental amount A, specified in the tune file using the variable `defDrillPeckClr`. Next the tool will rapid to a distance slightly above the last infeed depth (this depth is specified by your machine integrator), then feed another Q distance into the part. This sequence will repeat until the tool reaches the programmed Z depth.

Tool motion for **G73** chip break drill cycle - Tool returns to R plane between each peck.

![Figure 3–32. Tool Motion for G73 Chip Break Drill Cycle](image)

**Example:**

```
G90
G73 X2 Y2 Z-1.2 R.1 Q.5 G99
```
**G74 - Left Hand Tapping**

G74 executes the same as G84 Right Hand Tapping except the spindle directions are reversed.

**G80 - Cancel Drill Cycle**

This code cancels any active drilling cycle. When G80 is active X, Y, and Z motion commands will move the tool directly to the programmed endpoint without performing a drilling cycle. In addition to turning off the drilling cycle, G80 activates Linear Motion at Rapid and clears the modal drilling parameters such as R, Z, Q, and P.

If the drill cycle depth is not programmed after G80, then the drill cycle depth will default to the R plane.

G00 cannot be programmed in the same block.

**G81 - Drill Cycle**

G81 sets drilling as the modal drill cycle. The tool motion for the drill cycle is shown below.

1. Position to XY
2. Rapid to reference plane
3. Feed to depth
4. Rapid to reference plane

**Example:**

```
N010 G90
N020 G81 X2 Y2 Z-1.2 R.1 F10 G99
```

![Figure 3-33. G81 Drill Cycle](image)
**G82 - Drill Cycle with Dwell**

**G82** sets drilling with dwell as the modal drill cycle. The tool motion for this cycle is the same as the drill cycle except a dwell for \( P \) seconds occurs at the bottom of the hole. See the drawing below.

1. Position to XY
2. Rapid to reference plane
3. Feed to depth
4. Dwell for \( P \) seconds or revolutions (depending on FPM or FPR mode)
5. Rapid to reference plane

**Example:**

```
G90
G82 X2 Y2 Z-1.2 R.1 F10 P2.2 G99
```

*Figure 3–34. G82 Drill Cycle with Dwell*
G83 - Peck Drill Cycle

**Machine Integrator Configurable**

**G83** sets peck drilling as the modal drill cycle. The tool motion for the peck drill cycle is shown in the Figure below. This cycle causes the tool to feed by the amount programmed in Q, then rapid back to the R plane. Next the tool will rapid to a distance slightly above the last infeed depth (this depth is specified by your machine integrator), then feed another Q distance into the part. This sequence will repeat until the tool reaches the programmed Z depth.

Tool motion for **G83** peck drill cycle - Tool returns to R plane between each peck.

![Figure 3–35. G83 Peck Drill Cycle](image)

**Example:**

```
G90
G83 X2 Y2 Z-1.2 R.1 Q.5 G99
```
G84 - Right Hand Tapping and Float Tapping

G84 sets right hand tapping as the modal drill cycle. The tool motion for this cycle is described below. Max® will automatically set the feedrate and spindle override controls to 100% while this cycle is active.

There are two ways to stop tool motion when the tap is in the bore. One is to press emergency stop; the other is to use retract. Max® will not respond to a motion-stop command until the tool returns to the reference plane.

The drawing below shows right-hand tapping tool motion:

1. Tool rapids to R plane.
2. Tool feeds to the programmed depth.
3. Spindle reverses direction.
4. Control dwells for P seconds if G94 is active or for P revolutions if G95 is active.
5. Tool feeds to R plane.
6. Spindle reverses direction

*Figure 3–36. G84 Right Hand Tapping and Float Tapping*
Deep Hole Tapping
Standard Tapping, using a floating tap holder, has added capability for deep hole tapping. Tapping may now specify, in Q, the depth for each in-feed. If Q is zero, the in-feed is to final depth. If Q is greater than zero, the in-feed for each pass is determined by Q. When Q is greater than zero, the return position is either back to the R plane, or specified by the tune variable. The return is always to the R plane when tune variable is zero.

Rigid Tapping Cycles
Max® supports two types of rigid tapping cycles. The machine integrator can set variables in the Tune file for rigid-tapping left and rigid-tapping right functions, as well as other functions that control how the functions are performed.

Rigid Tapping for Positioning Spindle
This requires a machine positioning spindle. Tool motion for the Position Loop cycle is shown in the drawing above. In this cycle, the spindle is treated as a slave axis to the Z axis, but you must still program an S command for spindle speed. The speed may be in either IPM or IPR mode. The spindle and the Z axis change direction together. The Q value is 0.

1. Tool rapids to R plane.
2. Spindle goes in clockwise direction.
3. Tool feeds to the programmed depth.
4. Spindle reverses direction.
5. Tool feeds to R plane.
**Rigid Tapping for Non-positioning Spindle**

This is where the spindle is the master and you can only specify IPR for the spindle. Tool motion for the Rigid Tapping for Non-positioning Spindle cycle is shown in the Figure below. In this cycle, the Z axis is treated as a slave axis to the spindle. To command the speed, use the S code only in IPR mode. The machine integrator sets axis acceleration speed, which affects the distance the spindle will feed.

1. Tool rapids to R plane.
2. Spindle goes in clockwise direction.
3. Tool feeds to the programmed depth. If spindle is slow to reverse direction, tool may go past programmed depth.
4. Spindle reverses direction. If slave mode is disengaged at R plane, the tool may go past R plane. Amount by which tool passes R plane depends on programmed axis acceleration.
5. Tool returns to R plane.

![Figure 3–37. Rigid Tapping for Non-positioning Spindle](image-url)
G98 - Drill Cycle Initial Level Return (default)

When G98 is modal, all G81-G89 drill cycles will command the tool to return to the Z axis coordinate where the tool was located.

If the initial Z coordinate is changed for subsequent cycles, the Z coordinate for the first cycle after G80 is used as the reference plane.

For example, in the following two blocks, the tool will rapid to Z10 in the second block even though Z20 (init plane) is specified:

```
N010 G0 Z10
N020 G81 Z5 F60 (feed to Z5, rapid to Z10)
N030 G0 Z20
N040 G81 Z15 (rapid to Z10, feed up to Z15, rapid to Z10)
N050 G80
```

To return to the init plane, G98 requires a G80 Cancel Drill Cycle to change to the init plane. For example:

```
N010 G0 Z10
N020 G81 Z5 F60 (feed to Z5, rapid to Z10)
N030 G80 Z20
N040 G81 Z15 (rapid to Z10, feed up to Z15, rapid to Z20)
N050 G80
```
**G99 - Drill Cycle R Plane Return**

When **G99** is modal, all G81-G89 drill cycles command the tool to return to the modal R Reference plane at the end of the drill cycle.

**General Drill Cycle Information**

G codes in the G81-G89 range tell Max® to perform some form of a drilling, tapping, or boring cycle. The drilling will occur along the Z axis. The general tool motion for each cycle is shown in the Figure below. The tool motion that occurs at the end of each cycle depends on the modal status of **G98/G99**.

When **G99** is modal, the tool will return to the R Reference plane at the end of the block. When **G98** is modal, the tool will return to the Z axis coordinate where the tool was when the cycle was initiated.

![Figure 3–38. General Drill Cycle Tool Motion](image-url)
**Above - Drill Cycle Motion with G98**
Tool returns to Z axis coordinate where tool was when cycle was initiated.

If the initial Z coordinate is changed for subsequent cycles, the Z coordinate for the first cycle after **G80** is used as the reference plane.

To return to the init plane, **G98** requires a **G80** Cancel Drill Cycle to change to the init plane.

*Figure 3–39. Drill Cycle Motion with G98*

**Above - Drill Cycle Motion with G99**
Tool returns to R reference plane
GLOSSARY

The following is an alphabetical listing of the terms used throughout this and other Max® documents.

- **Absolute Position**—The absolute position is the location of the axes with respect to machine zero.
- **Append**—Command that allows you to add a line of instructions to a part program after the current line.
- **Auto Mode**—Mode that allows the operator to execute a part program.
- **Axes, Axis**—Linear or rotary device driven by a servo motor. Turning machines usually have 2 or 4 axes, while milling machines can have up to 7 axes.
- **CCW**—Counter-clockwise refers to the direction of rotation of the spindle or tool magazine.
- **CW**—Clockwise refers to the direction of rotation of the spindle or tool magazine.
- **Chuck**—A mechanism that holds the part in some machines, usually on a lathe.
- **Clamp**—A mechanism that holds the part in some machines, usually on a mill.
- **Component Controls**—Controls that govern the individual parts of the machine tool including: spindle, coolant, conveyor belt, chuck or clamp, and tool.
- **Chip Conveyor**—The belt that runs at the bottom of the machine tool to carry away the chips.
- **Coolant**—The liquid that is used to cool the part and provide lubrication to the cutting tools during the machining operation. Coolant flow can be operated manually or controlled by the part program.
- **Cycle**—The act of running a part program on a machine tool.
- **Cycle Start**—The command to begin a cycle.
- **Delta**—The incremental distance of a jog movement. The delta distance is determined using the SetDelta option, or one of the five standard selections, in the Jog menu.
- **Emergency Stop**—This button instantaneously disables the axes and the spindle. Every machine tool is required, by law, to have at least one Emergency Stop button.
- **End of Block Stop**—When this is turned on, the machine tool will enter a stop at the end of every block (line) of part program. Also known as Single Block.
- **EOB**—See End of Block Stop
• **External Chucking**—A form of workholding where the part is clamped from the outside (pushing inwards).

• **Feedhold**—Feedhold stops axes motion until the operator presses the Cycle Start button to resume program execution.

• **Feedrate**—The rate at which the tools move in relation to the part. It is programmed in inches per minute (ipm) or millimeters per minute (mmpm), or inches per revolution (ipr) or millimeters per revolution (mmpr).

• **FPM**—Feet per Minute. Unit of measurement indicating the speed of the tool on the part.

• **Hold**—See Feedhold.

• **Home**—A designated machine specific position where the axes are normally located when the machine tool is not in a cycle operation. It is the syncing location.

• **Insert**—The part of the tool that comes in contact with the part being machined. When an insert wears or breaks, it can be easily replaced. Also, an insert can be a replaceable machining surface on some tools.

• **Internal Chucking**—A form of workholding where the chuck pushes outwards. This is used when the part is being held from an inside diameter.

• **IPM**—Inches per Minute. Unit of measurement indicating the speed of the tool along its cutting path. For example: feed rate.

• **IPR**—Inches per Revolution. It is a unit of measurement indicating the speed of the tool along its cutting path.

• **Jog**—A method of manually moving an axis without writing a part program.

• **JogAbs**—This option allows you to type in the Absolute Position where you want an axis to move to. This command is found on the Jog menu.

• **JogHome**—Instructs the machine tool to move the axes to their Home position.

• **Manual Mode**—The Control Mode where you can manually operate individual functions of the machine tool, including the jog movement, axis jogging, manually operating the spindle, and changing tools.

• **Jump**—Tapping Jump will move the part program up or down three pages. The direction is determined by which direction the part program was last scrolled.

• **Machine Zero**—The point where all machine coordinates are measured from. Machine zero is established when the operator completes a syncing operation on all axes.

• **Mode**—There are two Modes in Max®: Manual and Auto. Manual Mode is for manual operation of the machine tool. Auto Mode is for automatic operation of the machine tool.

• **Offset**—An offset is the difference between the theoretical and actual length and position of a tool. An offset accounts for any variations in the tool.

• **Operator Control Window**—The Main Window displays and controls some of the important features of the operation of the machine tool. It contains the Component, Override, and Running Controls.
Override Controls—These controls allow you to manually adjust the feed rate, spindle speed, rapid rate, and jog rate. The adjustments are defined as a percent of the programmed value.

Over-Travel Limit Switch—A switch that trips before an axis moves outside its allowable limits. Once tripped, the machine tool automatically enters an Emergency Stop situation and needs to be resynced.

Rapid or Rapid Transverse—Movement that occurs at the machine's maximum velocity.

Servo—The servos control the motors that power the axes, the tool changer, and the spindle.

SetCarTool—This option allows you to reset the number of the tool that is in the transfer car. This is only available on some machines.

SetCurTool—This option allows you to reset the number of the tool that is currently in the spindle. This is only available on some machines, usually a mill that has a tool changer.

SetDelta—This Jog menu option allows you to type in a specific incremental distance that you wish to jog a particular axis.

SetTurOffset (some lathes only)—The Turret Offset must be set before running a part. It is the difference between the z-axis' physical home position and the theoretical home position assigned to the z-axis in the part program.

Single Block—See End of Block Stop.

Slide—A touch screen control bar that allows you to continuously jog an axis while having full control of its speed and direction.

Slider Bar—A touch screen control bar that allows you to continuously change an input by an amount that it is determined by how far you drag your finger and how long you keep it in contact with the touch-screen monitor.

Spindle—The part of the machine tool that rotates during operation. On a mill it holds the tool. On a lathe it holds the workpiece.

Spindle Speed—The rotational speed of the spindle programmed in revolutions per minute (rpm).

Start—A command to tell Max® to cycle a part program, or to make a jog move.

Sub-Menu—A group of commands within a menu from which you can select the appropriate option.

Sync—A procedure that coordinates the control with the actual position of the axes by moving each axis to its home position. It is necessary to do this when first starting the machine tool.

Tap—This is the recommended way of using the touch-screen monitor. Lightly touch a fingertip to the screen and quickly remove it. No pressure is needed, and you do not need to hold your finger there for any length of time.

Tool Changer—Most machines have a tool changer so that different machining functions can be performed automatically on one part. The tools are automatically changed, as specified in the part program.

Transfer Arm (only on some machines)—Part of a tool changer that moves the tool from the transfer car to the spindle.
• **Transfer Car** (only on some machines)—Part of some tool changers that moves the tool between the transfer arm and the drum.

• **Turret**—Part of the tool changer on a lathe. It holds the tools and rotates to ensure that the correct tool is lined up with the part.

• **User Interface**—The user interface is how you interact with the control.

• **Window**—A window is an organizational unit of Max® that contains information and buttons to run the control. Windows can be moved, opened, and closed.

• **Zero Reference Switch**—Limit switch that helps the control locate the proper machine zero positions during a syncing operation.