Controls, Measurement and Sensing
A Division of Giddings & Lewis, LLC
A Thyssen Production Systems Business

MeasureMax+™

Measurement Software

For Sheffield®Coordinate Measurement Machines

Reference Manual
for
Measurement Library (MLB)

p/n M.1300.9732
file 58010343
15-Feb-2001

Giddings & Lewis reserves the right to make changes to the product which may not be reflected in this manual.

At Giddings & Lewis, our business is quality and precision. To help us maintain a high level of quality, we invite you to send us your comments.

Preface

This manual is for persons who use the Measurement Library (MLB) to write part programs for MeasureMax+™ and MaxLite™ systems. You should be familiar with the Microsoft® Visual Basic programming language when you write part programs for MeasureMax+™.

Using MLB

You may use this manual with the *System Orientation and Operations* manual.

This MLB Reference Manual includes:

- Explanations of the purpose of each MLB command
- Command format and examples for each MLB command
- Results of all measurements and calculations, in addition to the affects of each command on the coordinate measuring system and inspection
- Result of a ‘skip action’ for each command
- Error messages which might occur due to improper use of the command
- Supplementary information

This manual does not include:

- Programs for specific types of parts
- Instructions for using the Visual Basic programming language
- Detailed information on the CMM hardware
- Detailed information on the computer system

Using This Manual

This manual is divided into task-oriented chapters. The MLB commands that are used to complete certain types of tasks are grouped together in a chapter. At the beginning of each group, you will find a listing of the MLB commands included in that group. This alphabetical listing of MLB commands helps you find specific commands faster. We have included the following information in the final chapters of the manual.

- Chapter 25 includes the types of data storage used by the Measurement Library. It includes the Dimensional Array, MLB variables, feature storage, summation arrays, and parametric variables. It also includes a description of tolerancing modes and their affect on the Dimensional Array.
- Chapter 26 includes an alphabetical listing of all the MLB commands and their definitions.

- Chapter 27 is a glossary of terms used in this manual.
- An Index is located at the back of the manual for your assistance.

Skip Action

We use the term “Skip Action” throughout the MLB manual. Skip Action defines what happens when a command that is in a program block gets skipped during a partial program run. There are four possible results of a Skip Action:

1. The system pushes one or two null features into the Feature Stack. These null features fill the place that would have been occupied by a feature if the command had not been skipped. A null feature only occupies a place in the Feature Stack to keep the sequence of numbers consistent. The measurement processor registers an error condition if a null feature is used for feature construction.
2. The system ignores the command. These are functions that normally do not affect the Feature Register, Feature Stack or Feature Storage. They are also commands relating to inspection reporting or logging. In this case, the sequence number is advanced to retain the report or log order.
3. The system generates an error for commands that are not allowed in skip sequences.
4. The system executes the command anyway. These are commands that directly access the Feature Register, Feature Stack or Feature Storage. Commands that set global states (such as measurement units) or modify the current reference frame by a constant value (such as change offset angle) are also processed.

Special Note

Some MLB commands have been enhanced and require the current version of the Measurement Processor. This manual has been written specifically for:

- MP30/35/30-II, version 19
- SMP-3xx, version 19
- SMP-400, version 104

Table of Contents

1. Basic Measurement Group

MLB commands in the Basic Measurement Group collect, create, and move single point data. They can also set system constants for

- probe tip diameter,
- surface compensation,
- measurement units,
- part and machine scales, and
- temperature compensation.

The Basic Measurement Group includes the following MLB commands.

CM	**Probe Compensation** specifies the probe compensation for round features.
CP	**Determine Probe Compensation** corrects the computed feature results for probe radius after the feature is measured.
IC	**Immediate Read of MEA Correction** returns the current position based on the MEA correction data.
II	**Read CMM Axis Positions**
IM	**Read Probe Position**
IN	**Inch** sets measurement units to inches.
IT	**Initiate Temperature Compensation** causes all measurements to be corrected to appear as though both the machine and the part were at 20 degrees Celsius (68° F).
ME	**Metric** sets measurement units to millimeters.
PB	**Probe Type** identifies a probe's type (hard or touch) and status (deflected or not deflected since installation).
PR	**Probe Diameter** sets or returns a probe's diameter.
QP	**Query CMM Probe Position**
QT	**Query Temperature** returns the current temperature reading for a Temperature Compensation temperature sensor.
QU	**Measure Point** prompts the operator to record a single point on a part's surface.
QV	**Query Touch Point Surface Vector**
R3	**Get Averaged Position Reading**
RE	**Read Point**

RX	**Reverse Axis for Mirror Imaging Part**
SM	**Scale Machine** scales machine coordinates for the X, Y and Z axes.
SP	**Scale Part** scales part coordinates for the X, Y and Z axes.
ST	**Set Point / Feature**
WT	**Define Thermistors for TempComp**

CM - Feature-of-size Probe Compensation

The CM command is used to define the desired probe compensation of features which have a well-defined inside and outside, such as circles, ellipses, cylinders, cones, spheres, and slots/webs. It typically indicates whether the feature surface is internal or external and consequently, whether to compensate the feature's computed size by adding or subtracting the probe's radius.

On MP30 systems all fits are least squares. On SMP-400 systems or MeasureMax 6+ (version 6 or later) systems the CM command can be used to specify the type of fit algorithm to be used in certain feature computations.

Command Format

- MP30: MLString "CM *p1*"
- SMP-400 / MeasureMax+ 6.00: MLString "CM *p1*,*p2*"

Examples

1. MLString "CM INT"
2. MLString "CM EXT"
3. MLString "CM INT,MIC"

Input

- *p1* = INT or EXT or OFF (String; optional)
- *p2* = LSQ or MCC or MIC or MRS (String; optional)

p1 indicates whether the probe is inside the feature (INT) or outside the feature (EXT) during point collection. Setting *p1* to either INT or EXT will cause the computed feature size (position for cones) to be corrected for the probe size. If *p1* is set to OFF then the feature calculations are not affected by the probe size.

<u>SMP-400 / MeasureMax+ 6.00 systems only:</u>
p2 indicates which type of fit is to be used for computation of circles, cylinders, and spheres. LSQ requests a least-squares best-fit; MCC requests a minimum circumscribed fit; MIC requests a maximum inscribed fit; and MRS requests a minimal radial separation fit. LSQ is the default at part program initialization and it is selected automatically if *p1* is specified without *p2*.

If *p1* and *p2* are omitted then system variables S7 and S8 are set to reflect the current system settings.

Results

If *p1* and/or *p2* are specified, the system sets the desired compensation and fit type.

Whether *p1* and/or *p2* are specified, a value reflecting the current compensation is set in system variable S7 (1 -> INT, -1 -> EXT, 0 -> OFF) and (SMP-400 only) a value reflecting the current fit type is set in system variable S8 (0 -> LSQ, 1 -> MCC, 2 -> MIC, and 3 -> MRS).

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-043 A segmented measurement is in progress.

CP - Determine Probe Compensation

The CP command is used after a feature has been measured with a touch probe without probe compensation. It corrects the computed feature results for probe radius. The feature results are based on the location of the surface relative to the probe center using the touch vector of the last touch point. A character string is returned in S0D indicating what compensation was performed. The feature cannot be a 3D line or a distance feature.

Command Format

MLString "CP *p1*"

Examples

1. MLString "CP"
2. MLString "CP P"

Input

- *p1* = P (String; optional).

p1 can only be specified if the feature is a point. If it is specified, probe compensation is to be based on polar coordinates in the active working plane assuming that the point was either on a round surface centered at the datum or on a flat surface parallel to a polar radius line.

In the first example (MLString "CP"), the feature in the feature register is corrected for probe compensation based upon the last touched point. If the feature is a 2- dimensional line, a plane, or a cone, the feature position is corrected. If the feature is a circle, sphere, cylinder, ellipse, or slot or web, the feature size is corrected. If the feature is a point, the surface is assumed to be parallel to one of the major working planes and feature position is moved parallel to the axis normal to that plane.

In the second example (MLString "CP P"), "P" indicates that the last point in the feature register is corrected for probe compensation. The surface is assumed to be either normal or perpendicular to a radial line in the working plane. The point's position is moved along or perpendicular to the radial line, based on the polar coordinates in the active working plane.

Results

The feature position and/or size is updated in the feature register and the Dimensional Array as required. S0D may be any of the following:

- **off** - Compensation cannot be determined because a hard probe is in use or because the touch vector is zero.
- **int** - Feature is compensated as internal feature of size (that is, I.D.)
- **ext** - Feature is compensated as external feature of size (that is, O.D.)
- **+X** - Feature is a two-dimensional line or plane. Feature position is compensated based on +X part axis pointing generally away from the surface.
- **-X,+Y,-Y,+Z or -Z** - Same as +X except for the axis identified.
- **pr,0,0 or npr,0,0** - Feature is point, move in the +X or -X direction by probe radius.
- **0,pr,0 or 0,npr,0** - Feature is point, move in +Y or -Y direction by probe radius.
- **0,0,pr or 0,0,npr** - Feature is point, move in +Z or -Z direction by probe radius.
- **spr or snpr** - Feature is a point, move in + or - polar angle direction by probe radius.
- **rpr or rnpr** - Feature is a point, moved away from or toward the datum in the active plane by a probe radius.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-021 First parameter invalid. "P" cannot be specified if feature is not a point.
- MLB-030 The feature register does not contain a feature.
- MLB-036 The feature in the feature register is a null feature.
- MLB-037 Invalid feature type. Distance or 3-dimensional line in feature register.

IC - Immediate Read of MEA Correction Data

Get a current reading for the CMM X, Y, and Z axes of either: (1) MEA corrected position as number of counts offset from MEA zero switches; or, (2) the current MEA correction values in counts. Note that this command cannot be executed prior to setting the MEA zeroes.

Command Format

MLString "IC *p1*"

Examples

1. MLString "IC"
2. MLString "IC C"

Input

- *p1* = C (String; optional).

The parameter *p1* can only be the character 'C'. If *p1* is specified, then the return values are the MEA / laser correction factors in counts. If *p1* is not specified, then the return values are the MEA / laser corrected position in counts relative to the MEA zero switch locations.

The first example captures the corrected XYZ axis position data.

The second example captures the current XYZ axis MEA/laser correction offsets.

Results

If *p1* is not specified then the current position data for X, Y and Z in counts, including MEA and laser corrections is returned. This data is stored in the following variables:

- S7 CMM X-axis reading, offset from MEA zero switch
- S8 CMM Y-axis reading, offset from MEA zero switch
- S9 CMM Z-axis reading, offset from MEA zero switch

If *p1* is specified, the current MEA and laser correction data for X, Y and Z in counts is returned. This data is stored in the following variables:

- S7 X-axis correction
- S8 Y-axis correction
- S9 Z-axis correction

The X, Y and Z position data or correction data is expressed in machine axis counts and is not affected by changes in the active part reference frame.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

II - Read CMM Axis Positions

The II command gets a current position reading for the CMM linear axes.

Command Format

MLString "II"

Examples

1. MLString "II"

Input

None.

Results

The current position data in millimeters is returned to system variables for all the CMM linear axes. This data is stored in the following variables:

- S7 - CMM X-axis reading, offset from MEA zero switch.
- S8 - CMM Y-axis reading, offset from MEA zero switch.
- S10 - CMM Z-axis reading, offset from MEA zero switch.
- S11 - ATLAS Y2-axis reading, if present.
- S12 - ATLAS Y3-axis reading, if present.

The X, Y, and Z readings are offsets from the MEA zero-switch positions and are not affected by changes in the active part reference frame. The ATLAS Y2 and Y3 readings are relative to the Y-axis zero-switch position (these counters are reset to zero when the Y-axis MEA zero switch is initially tripped).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

None.

IM - Read Probe Position

The IM command reads the current probe tip's position and records a point feature.

Command Format

MLString "IM"

Examples

1. MLString "IM"

Input

None.

Results

The system stores a point feature with the current probe position's X, Y and Z coordinates in the Feature Register. The contents of the Feature Register move into the Feature Stack. In the Dimensional Array, the system stores :

- Feature data in X, Y, Z, R, and A,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

None.

IN - Inch

The IN command sets the units of measurement to inches.

Command Format

MLString "IN"

Examples

1. MLString "IN"

Input

None.

Results

Axis position displays and the report linear data changes so that CMM-generated data is expressed in inches. The number of digits shown after the decimal point will also typically change since inch values are usually shown to one more place than metric values. The number of places shown will depend on the configured resolution of the CMM system. The probe diameter, part coordinates and actual values in the Dimensional Array convert to inches. User-entered nominal and tolerance values are not affected.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-043 A segmented measurement is in progress.

IT - Initiate Temperature Compensation

The IT command initiates temperature compensation on systems supporting this option. This command is also used to determine whether any Temperature Compensation warnings occur during a part inspection on systems with Temperature Compensation hardware.

This command should immediately follow the **IL** command. It causes all measurement results to be corrected to appear as though both the machine and the part were at a temperature of 20 degrees Celsius (68 degrees Fahrenheit). The **IL** command cancels Temperature Compensation.

Command Format

MLString "IT P1,P2"

Examples

1. MLString "IT 12.7F"
2. MLString "IT 22.9,23.5"

Input

- **P1** = Part thermal coefficient (Numeric; optional).

- **P2** = Temperature of part and machine (Numeric; optional).

The optional parameter P1 defines the thermal coefficient of expansion of the part being inspected. The thermal coefficient is expressed as units per million units per degree. If the value specified is in Fahrenheit, the value must be followed by the letter "F." If no letter follows the value specified, the value is treated as per degree Celsius. The letter "C" may be appended to identify Celsius but it is not required since this is the default.

The optional parameter P2 specifies the current temperature in the units (degrees Fahrenheit or Celsius) defined by P1. This parameter is ignored on systems with Temperature Compensation hardware. On systems without Temperature Compensation hardware, this temperature is used as the temperature of both the machine and the part.

The first example (MLString "IT 12.7F") returns temperature values in Fahrenheit. The thermal coefficient indicates that the part material expands at a rate of 12.7E-6 inches per inch (or millimeters per millimeter) per degree Fahrenheit. This is a typical value for an aluminum alloy. This form of the command assumes that Temperature Compensation hardware is available.

Example 2 (MLString "IT 22.9,23.5") returns temperature values in Celsius. The thermal coefficient indicates that the part material expands at a rate of 22.9E-6 millimeters per millimeter (or inches per inch) per degree Celsius. This is a typical value for an aluminum alloy. The current temperature is to be assumed to be 23.5 degrees Celsius for both machine and part unless Temperature Compensation hardware is available.

Results

If P1 is specified, new scale factors are computed for the X, Y, and Z machine axes as well as the part. These scale factors correct actual measured and displayed coordinate values to reflect the difference in temperature between the machine axes and the part (if Temperature Compensation hardware is present). They also account for the difference between the current temperature and the nominal temperature of 20 degrees Celsius (68 degrees Fahrenheit). This scaling remains in effect until one of the following commands is executed:

- **IS** (System Initialization)
- **IL** (Part Program Initialization)
- Another **IT** command

When the IT command with parameter P1 is executed, the following text is output to the inspection report and the inspection log:

Temperature compensation requested. Thermal coefficient = n.n(F).

n.n is the specified thermal coefficient. (F) appears if the temperature is being measured in Fahrenheit. This text may be followed by the message:

- WARNING 200 : CMM does not support temperature (n)

Here, if "n" equals 0 then the system is not configured to support TempComp; if "n" equals -1 then a serious system error has disabled TempComp; if "n" equals -2 then no part thermistors are active; and a value in the range 1 to 28 indicates a particular transducer has generated an invalid reading.

Otherwise, the following message prints:

- Machine type :. ttt - Temperatures :. (X :.t.t, Y :.t.t, Z :.t.t, P :.t.t)

Here, ttt is the configured machine type and (X :.t.t, Y :.t.t, Z :.t.t, P :.t.t) give the average X, Y, Z, and part temperatures. These temperatures are used for rescaling purposes. The temperatures are reported in Celsius or Fahrenheit to agree with the specified thermal coefficient.

Note that the temperature specified by P2 is printed for all of the above if Temperature Compensation hardware is not available. The temperature values for X, Y, Z, and the part are returned to variables S7, S8, S10, and S11 respectively.

If P1 is not specified, a numeric value is returned to S7. This value indicates whether any Temperature Compensation warnings have been generated. This value is zero (0) if Temperature Compensation hardware is not available or if no warnings have occurred. A value ranging from one (1) to thirty-one (31) is returned to S7 if a warning has occurred. The returned value is a coded number that indicates which type or types of warnings occurred during the part inspection. It is the sum of five values corresponding to the different warning types:

- A one (1) is returned if a violation of the absolute temperature limit (Warning 201) has occurred.
- A two (2) is returned if a violation of gradient temperature limit (Warning 202) occurred.
- A four (4) is returned if a violation of transient temperature limit (Warning 203) occurred.
- An eight (8) is returned if a violation of the probe data (Warning 205) occurred.
- A sixteen (16) is returned if a violation of the table center coordinates (Warning 206) occurred.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

ME - Metric

The ME command sets the units of measurement to millimeters.

Command Format

MLString "ME"

Examples

1. MLString "ME"

Input

None.

Results

Axis position displays and the report linear data change so that CMM generated data is expressed in millimeters. The number of digits shown after the decimal point will also typically change since metric values are usually shown to one less place than inch values. The number of places shown will depend on the configured resolution of the CMM system. The probe diameter, part coordinates and actual values in the Dimensional Array convert to millimeters. User entered nominal and tolerance values are not affected.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-043 A segmented measurement is in progress.

PB - Probe Type

The PB command identifies the type of probe in use by returning a single-digit code. The code also shows the probe's status.

Command Format

MLString "PB"

Examples

1. MLString "PB"

Input

None.

Results

The system puts the probe status code into variable S7. The following list shows the possible codes and the status each code signifies:

- 0 = Hard probe installed.
- 1 = Touch probe installed. It has not been deflected since the last measurement was read.
- 2 = Touch probe installed. It has been deflected, but not read.

SMP-400 systems use variable S8 to indicate the presence of an SP600 analog probe. If the active probe is an SP600 analog probe, then variable S8 is set to one (1); otherwise, it is set to zero (0).

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

PR - Probe Diameter

The PR command sets or returns the active probe's diameter in current units. The probe diameter does not need to be set with PR if the probe is calibrated using a calibration sphere of known size. The initial default value for probe diameter is 4 mm.

Command Format

MLString "PR P1"

Examples

1. MLString "PR 3"

2. MLString "PR"

Input

- **P1** = Probe diameter (Numeric; optional).

P1 specifies the probe's diameter. In the first example, the probe is 3 units in diameter.

The second example omits P1. The system stores the probe's diameter in S7.

Results

If P1 is specified, the active probe's diameter is set to the specified value. If P1 is omitted, the system returns the current probe diameter in variable S7.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

QP - Query CMM Probe Position

The QP command copies the X, Y, and Z coordinates into variables S8, S10, and S11 at the time a probe was touched, or the Record button on the RCU was pressed, or the footswitch was pressed.

Command Format

MLString "QP P1"

Examples

1. MLString "QP"

   Capture current CMM probe position.

2. MLString "QP Q"

   Wait for operator to press Record button on RCU or footswitch, then capture CMM position or last touch point.

Input

- *p1* **= 'Q' or 'N'** (String; optional)

If *p1* is specified as 'Q' then the position reading is to be qualified by an operator pressing either the Record button on the RCU or the footswitch.

If *p1* is specified as 'N', then current CMM probe position data is returned with no averaging. This option is reserved for internal use only by Sheffield Measurement for utilities such as motion tunneling tests.

Results

Depending on the type of probe mounted, the command argument, and the operator actions, an X, Y, Z part coordinate probe position is placed in variables S8, S10, and S11.

- If *p1* is specified as 'Q' and a touch probe is in use, and the touch probe is tripped after the execution of this command but before the Record button (or footswitch) is pressed, then S7 is 1.0 and S8, S10, and S11 reflect the coordinates of the touched point.
- If *p1* is specified as 'Q' and a touch probe is in use, but the touch probe is not tripped after the execution of this command and before the Record button (or footswitch) is pressed, then S7 is 0.0 and S8, S10, and S11 reflect the coordinates of the probe position at the time the Record button (or footswitch) was pressed.
- If *p1* is specified as 'Q' but a touch probe is not in use, then S7 is 1.0 and S8, S10, and S11 reflect the coordinates of the probe position at the time the Record button (or footswitch) was pressed.
- If *p1* is not specified, then S7 is 0.0 and S8, S10, and S11 reflect the coordinates of the probe position at the time the command was executed.

Skip Action

Ignored. Not executed if inside a block of code being skipped.

Error Messages

None.

QT - Query Temperature

The QT command retrieves the current temperature reading for a particular Temperature Compensation temperature sensor.

Command Format

MLString "QT P1"

Examples

1. MLString "QT 4F"

Input

- **P1** = 1 to 28 (Integer; required)

P1 specifies a particular temperature sensor. Temperature sensor locations may vary from one CMM type to another, so a map of the temperature sensor locations for the particular CMM must be obtained before using **QT**. The letter "F" must follow P1 if the temperature is to be reported in degrees Fahrenheit.

The example above retrieves the current temperature reading of temperature sensor #4 in Fahrenheit degrees.

Results

Variable S7 is set to the current temperature reading for the specified temperature sensor. If the hardware for temperature compensation is not present or is not operational, S7 is set to zero (0).

Skip Action

Executes even when inside a block of code being skipped.

Error Messages

None.

QU - Measure Point

The QU command prompts the operator to record a single point on a part's surface. The command provides control over probe compensation, prompt display, storage of point data, and whether footswitch qualification of touch probe data is required.

Command Format

MLString "QU P1,P2,P3"

Examples

1. MLString "QU"
2. MLString "QU @MEASURE POINT ON TOP SURFACE@,$CQ"
3. MLString "QU 0,0,NPR,$RC"
4. MLString "QU RPR,$C"
5. MLString "QU $CQU"

Input

- **P1** = @operator prompt@ (String; optional)
- **P2** = S, R, or set of three numbers (Numeric/string; optional).
- **P3** = $ and one or more letters C, R, Q, U, I, D and S (String; optional).

P1 is used to specify a message that is displayed when this command is executed. This prompt should help the operator identify the correct feature to be measured.

P2 allows for compensation of the measured point. P2 takes one of three forms:

- The letter "S” followed by a signed number or parametric variable. This specifies that the measured point is moved by the designated amount in a direction perpendicular to the polar radius line.
- The letter "R” followed by a signed number or parametric variable. This specifies that the recorded point is moved by the specified amount along the polar radius line.
- A set of three signed numbers or parametric variables. These specify that the measured point is to be moved by the designated amounts parallel to the X, Y and Z axis respectively.

P3 allows control of the footswitch. P3 is a character string consisting of the $ sign followed by one or more of the characters C, R, Q, U, I, D and S. These are operation control characters which may be specified in any order and are defined as follows:

- C - Indicates that the operator prompt is not sent to the host computer.
- R - Indicates that the operator prompt is not sent to the RCU.

- Q - Indicates that the touch probe hits do not require operator qualification. If specified, a touch hit that occurs after this command is received is to be accepted without qualification as the desired measurement point. If specified and a touch probe is active, pressing the Record button on the RCU (or footswitch) will terminate this command with an error #91.
- U,I - These letters indicate that the measured point is to be stored in either the User (U) or the Internal (I) Summation Array rather than being pushed into the Feature Register. Only one of these may be specified. Inclusion of either of these letters causes the probe compensation parameter P2 to be ignored; that is, no compensation for probe size will be made to the point before storage into the Summation Array. If U or I is specified, then the coordinates of the measured point are returned to the variables S7, S8 and S10 as well as added to the specified array.
- D, S - These control characters indicate that the QU command is to be terminated if either the Drive button or the Return-to-Screen button is pressed on the RCU-II. Note that these require SMP-3xx firmware of at least Version 18.0 or SMP-400 firmware of at least Version 103.0 in addition to RCU-II firmware of at least Version 2. If any of these are incorrect then specification of D and/or S will result in a syntax error (#-21, -22, or -23). Otherwise, if D is specified then pressing the RCU-II Drive button causes the command to terminate with Error #-88; if S is specified then pressing the RCU-II Return-to-screen button causes the command to terminate with Error #-89.

Example 2 sends the default prompt to the RCU but not to the host. If a touch probe is in use, the next touch probe hit is accepted without qualification. No compensation is performed. The measured point is pushed into the Feature Register.

Example 3 does not generate any operator prompts. Probe compensation is performed by subtracting the probe radius from the Z-axis coordinate. The measured point is pushed into the Feature Register. Qualification is always required.

Example 4 sends the default operator prompt to the RCU but the message is not sent to the host. Probe compensation is performed by adding the probe radius to the polar radius of the recorded point based on the active working plane. The measured point is pushed into the Feature Register. Qualification is required even if a touch probe is in use.

Example 5 generates no operator prompt. No probe compensation is performed. The measured point is stored in the User Summation Array. If a touch probe is in use, the next touch probe hit is accepted without qualification as the measured point.

Results

The operator is prompted as specified to measure a single point. If a touch probe is in use and qualification is not required then this measurement is accomplished by simply deflecting the touch probe. Otherwise it is necessary to deflect the touch probe (if any) and press the Record button on the RCU (or footswitch). If a touch probe is in use, the coordinates of the most recent touch are recorded; otherwise, the current machine coordinates are recorded. The coordinates of the measured point are either stored in the specified Summation Array and returned to variables S7, S8 and S10, or compensated for probe size, pushed into the Feature Register, and stored in the Dimensional Array.

Skip Action

Not executed if part of a skipped block. A null feature is pushed onto the Feature Register/Stack unless storage of point to Summation Array is requested.

Error Messages

- MLB-021 First parameter is invalid.

- MLB-022 Second parameter is invalid. Either it is syntactically incorrect or at least one of the control characters D or S has been specified but either an RCU-I is in use, an MP30/35 is in use, the SMP-3xx firmware is less than Version 19.0, the SMP-400 firmware is less than Version 104.0, or the RCU-II firmware is less than Version 2.
- MLB-023 Third parameter is invalid. Either it is syntactically incorrect or at least one of the control characters D or S has been specified but either an RCU-I is in use, an MP30/35 is in use, the SMP-3xx firmware is less than Version 18.0, the SMP-400 firmware is less than Version 103.0, or the RCU-II firmware is less than Version 2.
- MLB-088 D was specified and the RCI-II Drive button was pressed.
- MLB-089 P3 specified 'S' and the RCI-II Return-to-screen button was pressed.
- MLB-091 P3 specified 'Q' for a non-qualified touch probe reading, a touch probe is in use, but the RCU Record button or the footswitch was pressed before a point was recorded by deflecting the touch probe.

QV - Query Touch Point Surface Vector

Query the surface vector associated with the last recorded touch.

Command Format

MLString "QV"

Examples

1. MLString "QV"

   Query touch vector and return a unit vector associated with the most recently recorded touch.

Input

None.

Results

A unit vector in part coordinates for the estimated surface normally associated with the most recently recorded touch probe 'hit' is returned to variables S7, S8 and S10. Note that this vector may be zero (0,0,0) or meaningless if no touches have been recorded or if a touch probe is not in use. Note that this vector is a rough estimate based either on the direction of travel before surface contact or on the analog probe's deflection vector at the computed touch point.

Skip Action

Ignored.

Error Messages

None.

R3 - Get Averaged Position Reading

The R3 command is used to capture a current position reading for the CMM X, Y, and Z axes as well as the three axes of an analog probe such as the SP3.

Command Format

MLString "R3"

Examples

1. MLString "R3"

Input

None.

Results

The current position in part coordinates is returned to system variables for the CMM linear axes together with the analog probe axes. The reading is the average of the position data over the last 128 milliseconds. This data is stored in the following variables:

- S7 - Part coordinate X-axis reading.
- S8 - Part coordinate Y-axis reading.
- S10 - Part coordinate Z-axis reading.
- S11 - Analog probe first axis reading (counts).
- S12 - Analog probe second axis reading (counts).
- S13 - Analog probe third axis reading (counts).

Skip Action

Ignored; not executed when inside a skipped block.

Error Messages

None.

RE - Read Point

The RE command retrieves a hard probe's current position. If a touch probe is in use, the last recorded location that the probe touched is retrieved.

Command Format

MLString "RE"

Examples

1. MLString "RE"

Input

None.

Results

The point is stored in the Feature Register. The contents of the Feature Register move into the Feature Stack. In the Dimensional Array, the system stores:

- Feature position data in X, Y, Z, R, and A,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

None.

RX - Reverse Axis for Mirror Imaging Part

The RX command reverses the direction of count for a specified coordinate axis. This command is intended for use in inspecting mirror-imaged parts with a minimum of part program modification. Axis reversal is canceled by part program initialization (**IL**) and by executing a no-level (**NO**) command. Recalling a part reference frame (**RR**) sets the axis reversal status to the state active when that reference frame was saved.

Command Format

MLString "RX P1"

Examples

1. MLString "RX"

   The system returns status information as to whether any axis is currently reversed.

2. MLString "RX X"

   Modify the active reference frame by adding reversing the X-axis.

3. MLString "RX N"

   Modify the active reference frame by ensuring that no coordinate axes are reversed. This command can be used to cancel a previous axis reversal.

Input

- P1 (String; optional) = Can be only **N**, **X**, **Y** or **Z**.

  This parameter designates which coordinate axis is to be reversed with **N** indicating that no axes are to be reversed. If P1 is not specified, then the system returns status information as to whether any axis is currently reversed.

Results

If P1 is not specified, then the current axis reversal state is returned in variable S7. A value of zero (0) indicates no axes are reversed, a value of one (1) indicates the X-axis is reversed, a value of two (2) indicates the Y-axis is reversed, and a value of three (3) indicates that the Z-axis is reversed.

If P1 is specified then the reference frame is updated to reflect the fact that the specified axis has been reversed from its normal positive direction. In effect this mirrors the active reference frame about the datum point along the specified part axis. Only one axis may be reversed. If this command is executed for a second axis, then the reversal of the first axis is canceled. If it is re-executed for an axis that is already reversed, then it has no affect. The data in the feature register is updated to reflect the modified reference frame.

This axis reversal is saved and restored with the part reference frame data.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-010 Either the host software or the - firmware does not support this capability. Firmware newer than V20 (SMP-3xx) or V104.1 (SMP-400) is required.

SM - Scale Machine

Note: Use Scale Machine before establishing the part reference frame or measuring features.

This command compensates for changes in the size of the machine X-, Y-, and Z-axes. The default scale factor is 1-to-1. The scale factors can be changed for all axes or for each axis independently.

Command Format

MLString "SM P1,P2,P3"

Examples

1. MLString "SM .99,.99,.99"
2. MLString "SM 1.01,1.01,1"

Input

- **P1** = X Scale Factor (Numeric; required).
- **P2** = Y Scale Factor (Numeric; required).
- **P3** = Z Scale Factor (Numeric; required).

P1, P2 and P3 represent the machine scale factors for each of the 3 axes.

In the first example, the first .99 sets up a .99-to-1 ratio for the X machine axis. The second .99 sets up a .99-to-1 ratio for the Y machine axis. The third .99 sets up a .99-to-1 ratio for the Z machine axis. The CMM has contracted one percent along each of the X-, Y-, and Z-axes.

In example 2, the first 1.01 sets up a 1.01-to-1 ratio for the X machine axis. The second 1.01 sets up a 1.01-to-1 ratio for the Y machine axis. The CMM has expanded one percent along the X- and Y-axes. The third factor, 1, sets up a 1-to-1 ratio for the Z machine axis. This indicates that the Z-axis has not changed.

Results

The system deletes all features that were in the Feature Register and the Feature Stack.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-043 A segmented measurement is in progress.

SP - Scale Part

Note: Establish the part reference frame before using the **SP** command.

The SP command compensates for changes in the scaling of the X, Y, and Z part axes. Use the **SP** command to compensate for scale models. The default scale factor is 1-to-1. The scale factors can be changed for all axes or for each axis independently.

To inspect mirror-imaged parts with a minimum of part program modification, use the RX command. The RX command reverses the direction of count for a specified coordinate axis.

Command Format

MLString "SP P1,P2,P3"

Examples

1. MLString "SP 2,2,2"
2. MLString "SP 1.8,1.5,1"

Input

- **P1** = X Scale Factor (Numeric; required).
- **P2** = Y Scale Factor (Numeric; required).
- **P3** = Z Scale Factor (Numeric; required).

P1, P2 and P3 represent the part scale factors for each of the 3 axes.

In the first example above, the first 2 sets up a 2-to-1 ratio for the X part axis. The second 2 sets up a 2-to-1 ratio for the Y part axis. The third 2 sets up a 2-to-1 ratio for the Z part axis. The part being measured is a one-half scale model. At a ratio of 2-to-1, the system displays and reports full-scale dimensions.

In example 2, 1.8 is the scale factor for the X part axis. The part is expected to shrink .8 along the X-axis. The 1.5 is the scale factor for the Y part axis. The part is expected to shrink .5 along the Y-axis. The third factor, 1, sets up a 1-to-1 ratio for the Z part axis. This signifies that the Z-axis is at full scale.

The probe diameter must be appropriately rescaled when the part is expected to shrink along each axis.

Results

Until the part scale is changed, all measurement results will be multiplied by the specified scale factors. Note that the specified part scale factors are associated with the X, Y, and Z part axes.

Warning

The SP command rotates with the part axes if a leveling or offset operation is performed. This may cause unexpected results if different part scale factors were specified for the three axes and then the coordinate reference frame is changed.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-043 A segmented measurement is in progress.

ST - Set Point / Feature

The ST command stores an artificial feature in the Feature Register.

Command Format

MLString "SS P1,P2,P3,P4,P5,P6,P7,P8,P9,P10"

Examples

1. MLString "ST 0,0,0"

   Push the point (0,0,0) into the Feature Register.

2. MLString "ST X,Y1,10"

   Push the point (X, Y1, 10) into the Feature Register. "X" identifies the X-coordinate of the feature currently in the Feature Register. "Y1" identifies the Y-coordinate of the feature currently in F1.

3. MLString "ST 0,0,0,3,1,1,0"

   Push the line through the point (0,0,0) and with direction vector (1,1,0) into the Feature Register. The direction vector is normalized before it is stored.

4. MLString "ST 1,2,3,5,0,0,0,4.75"

   Push the circle with center (1,2,3) and radius (4.75) into the Feature Register. Since the direction vector is (0,0,0), the working plane of the circle defaults to the active working plane.

5. MLString "ST 10.75,21,0,17,0,0,0,4.75,5.25,22.5"

   Push the ellipse with center (10.75,21,0), minor radius 4.75, major radius 5.25, and with its major axis inclined at an angle of 22.5 degrees from the major axis of the plane into the Feature Register. Since the direction vector is (0,0,0), the plane of the ellipse will default to the active working plane.

Input

- **P1** = feature position X-coordinate (numeric/parametric; required)
- **P2** = feature position Y-coordinate (numeric/parametric; required)
- **P3** = feature position Z-coordinate (numeric/parametric; required)
- **P4** = feature type (numeric; optional; default = 1)
- **P5** = X component of direction vector (Numeric; optional)
- **P6** = Y component of direction vector (Numeric; optional)
- **P7** = Z component of direction vector (Numeric; optional)
- **P8** = feature size (Numeric; optional)
- **P9** = major radius (Numeric; optional) (or length for cparln features)
- **P10** = inclination of ellipse (numeric; optional) (or ellipse inclination vector) (or normal direction vector for slot/web, cparln features)

P1, P2, and P3 together define the XYZ position of the feature being created.

P4 sets the type for the feature being created. It must be one of the following:

- 1 = point (default)
- 2 = plane
- 3 = line
- 4, 5, or 6 = circle, O.D.; circle; circle I.D.
- 7, 8, or 9 = sphere, O.D.; sphere; sphere I.D.
- 10, 11, or 12 = cylinder, O.D.; cylinder; cylinder I.D.
- 13, 14, or 15 = cone, O.D.; cone; cone I.D.
- 16, 17, or 18 = ellipse, O.D.; ellipse; ellipse I.D.
- 19 = distance
- 20, 21, or 22 = web, O.D.; slot/web; slot.
- 23, 24, or 25 = cparlnf, O.D.; cparlnf; cparlnf I.D.
- 26, 27, or 28 = cparlnr, O.D.; cparlnr; cparlnr I.D.

P5, P6, and P7 together define the feature's orientation. They are normalized to a unit vector. If the direction vector is omitted or set to (0,0,0) for features requiring direction cosines (line, plane, cylinder, cone, slot/web), then it defaults to the normal to the currently selected working plane.

P8 specifies the feature size and must be specified as a positive value for all features with size. P8 sets the radius for circles, spheres, and cylinders; the minor radius for ellipses; the apex angle for cones; and the width for slots, webs, or cparln features. P8 is ignored if it is included for other feature types.

P9 specifies the major radius for ellipses. For ellipses, the specified value must be greater than P8. P9 specifies the length for cparln features. P9 is ignored if included for other feature types.

P10, if a single value, specifies the inclination in degrees of the major diameter of an ellipse relative to the first named axis of the active working plane. P10, if specified as a vector (three values separated by commas), is interpreted as the direction vector of the major axis in the plane of the ellipse. If P10 is included for slot/web and cparln features, this parameter MUST be three values separated by commas, and will be interpreted as the plane normal vector of the feature. P10 is ignored for all other feature types.

The meaning of these parameters is order-dependent. All parameters prior to a given parameter must be given.

Results

The feature described by the function parameters is pushed into the Feature Register.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-021 First parameter is invalid.
- MLB-022 Second parameter is invalid.
- MLB-023 Third parameter is invalid.
- MLB-024 Fourth parameter is invalid.
- MLB-025 Invalid parameter specified.

WT - Define Workpiece Thermistors for TempComp Scaling Corrections

The WT command identifies a list of thermistors to be used to determine the temperature of the workpiece being inspected. The thermistors specified replace the configured list of part thermistors until the system is reinitialized (see the **IL** or **IS** commands). If this command is not executed before the **IT** command is used to initiate temperature compensation, then the configured thermistors are used to determine workpiece temperature.

Command Format

MLString "WT P1"

Examples

1. MLString "WT 12"
2. MLString "WT 5,7"
3. MLString "WT"

Input

- **P1** = list of thermistors (Numeric, optional)

P1 defines the list of thermistors to be used to determine the temperature of the workpiece being inspected. Each thermistor number must be in the range of 1 to 28 and must correspond to a properly configured and active thermistor. A maximum of eight thermistors may be specified. A thermistor may be specified only once in the list. If more than one thermistor is specified then the average of the specified thermistors is used as the workpiece temperature. If no parameter is specified then the currently active list of workpiece thermistors is returned.

In the first example above, thermistor 12 is to be used to determine workpiece temperature when executing the **IT** command.

In the second example, the average of the readings for thermistors 5 and 17 is to be used to determine workpiece temperature when executing the **IT** command.

The third example is a request for the active list of workpiece thermistors.

Results

If P1 is specified, the specified list of workpiece thermistors is stored for use by the **IT** command. **This** list remains active until the next **IL**, **IS** or **WT** command or until power is cycled. Note: This command is ignored if the system is not configured as a TempComp system or if TempComp is inactive due to a hardware failure.

If P1 is not specified then the currently active list of workpiece thermistors is returned to host variables S7, S8, S10, S11, S12, S13, S14, and S15. The first zero in this list indicates the end of the list.

Skip Action

Ignored, not executed if part of a skipped block.

Error Messages

- MLB-021 First parameter invalid. The thermistor number is outside the 1 to 28 range or does not correspond to a properly configured thermistor.
- MLB-022 Second parameter invalid. The thermistor number is outside the 1 to 28 range or does not correspond to a properly configured thermistor.
- MLB-023 Third parameter invalid. The thermistor number is outside the 1 to 28 range or does not correspond to a properly configured thermistor.
- MLB-024 Fourth parameter invalid. The thermistor number is outside the 1 to 28 range or does not correspond to a properly configured thermistor.
- MLB-025 Invalid parameter specified. The thermistor number is outside the 1 to 28 range or does not correspond to a properly configured thermistor.

2. Part Reference Frame Group

MLB commands in the Part Reference Frame Group provide an exact reference from which the location, orientation, and geometric relationships of part features can be derived.

The Part Reference Frame Group includes the following MLB commands.

DR	**Delete Reference Frame**
EO	**Establish Offset Angle** establishes the offset angle based on a feature.
EP	**Establish Datum Plane** establishes the primary datum plane or working plane based upon the feature in the Feature Register.
ES	**Establish Reference** establishes the XYZ datum.
EX	**Establish X Reference** establishes the X datum.
EY	**Establish Y Reference** establishes the Y datum.
EZ	**Establish Z Reference** establishes the Z datum.
LV	**Level** orients the working plane.
MO	**Move Offset Angle** modifies the offset angle for the current reference frame.
MR	**Move Reference** moves a datum along one or more of the axes.
MX	**Move X Reference** moves a datum along the X-axis.
MY	**Move Y Reference** moves a datum along the Y-axis.
MZ	**Move Z Reference** moves a datum along the Z-axis.
NO	**Cancel Level** cancels leveling and aligns the part reference frame with the machine reference frame.
PS	**Preset Datum** establishes the XYZ datum relative to the position of the probe.
PX	**Preset X Datum** establishes a datum along the X axis relative to the position of the probe.
PY	**Preset Y Datum** establishes a datum along the Y axis relative to the position of the probe.
PZ	**Preset Z Datum** establishes a datum along the Z axis relative to the position of the probe.
RC	**Reference Frame Temporary Change**
RO	**Read Offset Angle** retrieves the offset angle of the current reference frame.
RR	**Recall Reference Frame** recalls a previously-saved reference frame.

RT	**Stored Reference Frame Status** determines the status of reference frame storage locations.
SL	**Set Level** specifies autolevel direction cosines for the working plane.
SO	**Set Offset Angle** specifies offset angle.
SR	**Save Reference Frame** saves the current reference frame.
XI	**LOM Inch Increment / Decrement** updates the MP30 transformations when the layout machine (LOM) is relocated along the machine base.
XM	**LOM Metric Increment / Decrement** updates the MP30 transformations when the layout machine (LOM) is relocated along the machine base.
XY	**XY Plane** selects the XY plane as the working plane.
YZ	**YZ Plane** selects the YZ plane as the working plane.
ZX	**ZX Plane** selects the ZX plane as the working plane.

DR - Delete Reference Frame

The DR command deletes a previously saved part reference frame (see the **SR** command).

Command Format

MLString "DR P1"

Examples

1. MLString "DR 5"
2. MLString "DR ALL"

Input

- **P1** = 1 to 48, label or 'ALL' (Numeric/string; required).

  Reference Frames 17 to 48 are available for MP30/35, SMP-3xx Version 19.0 and later, and SMP-400 Version 104.0 and later. With MeasureMax+ 6.00 and later P1 can be a label (an alphanumeric string; see **SR** command).

P1 is either an identifier telling which stored reference frame is to be cleared or the string 'ALL' indicating that data for all saved reference frames is to be cleared.

Example 1 above clears the saved data for reference frame 5. Example 2 clears the saved data for all reference frames.

Results

If P1 specifies a particular reference frame, then any saved reference frame data for that reference frame is deleted. If P1 is 'ALL', then all saved reference frame data is deleted. Subsequent references to the deleted reference frame(s) via the **RR** command will be invalid until another reference frame is saved there using the **SR** command.

Note: Data for all reference frames other than those with integer labels in the range 1 to 48 is deleted by execution of an **IL** command. Data for all reference frames is deleted by execution of an **IS** command.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

EO - Establish Offset Angle

The EO command rotates the active working plane to align it so that either the first- or second-named axis of the working plane is parallel to the line/axis of a specified line-type feature, parallel to a specified plane, or parallel to the line defined by two specified point-type features.

Command Format

MLString "EO P1,P2,P3,P4"

Examples

1. MLString "EO"
2. MLString "EO Y,+X"
3. MLString "EO Y,+X,CIRCLE13,CIRCLE14"

Input

- **P1** = Which machine axis to make parallel to the feature. Use X, Y, or Z (String; optional).
- **P2** = The machine direction indicating positive direction. Use -X, +X, -Y, +Y, -Z, +Z. (String; optional).
- **P3** = Name of a saved feature (String; optional)
- **P4** = Name of a saved feature (String; optional)

P1 and P2 allow control of the orientation of the part independent of the order of points or features taken.

P1 must identify one of the axes of the working plane. The identified axis is to be aligned. P2 identifies a signed machine axis that represents the general machine direction the positive end of the part axis should point after rotation. Both P1 and P2 are specified if either is. If P1 and P2 are omitted, then the first named axis of the working plane is aligned with the feature and counts positively from first point to last point.

P3 is the name of a saved feature to be used to establish the axis alignment. If not specified, the feature in the feature register is used. P4 may only be specified if the feature P3 is a point, circle, sphere, or ellipse. If P4 is not specified then the feature in the first feature stack position (F1) is used.

Unless otherwise specified, the recorded order of the points determines the positive direction aligned axis. The direction goes from the first to second points recorded. Recording the two points in the opposite order reverses the positive direction of both axes of the working plane.

Results

The position and/or orientation data for the feature in the feature register and in the Dimensional Array is updated to reflect the new orientation of the working plane.

If the specified feature is a line, cylinder, cone, or slot/web, the aligned axis is rotated so it is parallel to the orientation vector of the feature.

If the specified features are a pair of point-type features (point, circle, sphere, or ellipse), the aligned axis is rotated so it is parallel to the vector between the two features.

If the specified feature is a plane, the aligned axis is rotated so that it is normal to the direction vector of the plane. In this case, if the parameters P1 and P2 have not been specified, the positive end of the aligned axis is at a polar angle of +90 degrees from the plane's direction vector.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 The Feature Stack or Feature Register does not contain the required features.
- MLB-031 The first parameter specifies an invalid type.
- MLB-032 The Feature Register does not contain an appropriate feature
- MLB-035 Specified direction is perpendicular to working plane.
- MLB-036 One of the required features is a null feature.
- MLB-041 The points do not define a line or define a line perpendicular to the working plane.
- MLB-043 A segmented measurement is in progress.

EP - Establish Datum Plane

The EP command establishes the orientation and location of the working plane based upon a specified planar feature.

Command Format

MLString "EP P1,P2"

Examples

1. MLString "EP"
2. MLString "EP -Z"
3. MLString "EP PLANE13"
4. MLString "EP -X,RIGHTSIDE"

Input

- **P1** = X or -X or Y or -Y or Z or -Z or OFF or EXT or INT. (String; optional).
- **P2** = Name of a saved feature (String; optional).

P1 is used to control the direction that is positive (up) relative to the working plane after its rotation. It may be X, -X, Y, -Y, Z, or -Z to designate a machine axis that points generally in the direction desired for the positive end of the axis normal to the working plane. In the second example, -Z is a machine axis. The system mathematically rotates the working plane so that the positive end of the axis of rotation points in the -Z direction.

If P1 is OFF or EXT then the positive direction is to be defined by the plane's direction vector. This assumes that appropriate probe compensation was specified when the plane was measured so that the direction vector points away from the surface. If P1 is INT then the positive direction is to point into the surface rather than away from it.

If P1 is not specified then the positive direction is defined by the orientation of the machine axis corresponding to the axis normal to the working plane.

P2 is the saved feature used to establish the datum plane. If P2 is not specified, the feature in the feature register is used. In either case this must be a plane. In example 3, the **EP** command levels to the plane named "PLANE13" and sets the third axis datum to this plane.

Results

The working plane is rotated to be parallel to the specified plane and is translated so that the third axis coordinate value of this plane's position is zero. The position and/or orientation data for the feature in the feature register and in the Dimensional Array is updated to reflect the new orientation and location of the working plane.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 The specified feature does not exist.
- MLB-031 The feature in the feature register is not a plane.
- MLB-043 A segmented measurement is in progress.

ES - Establish Reference

Th ES command establishes the XYZ datum. It sets the datum (zero point) of the part reference frame to the coordinates defined by a specified feature position.

Command Format

MLString "ES P1"

Examples

1. MLString "ES"
2. MLString "ES FEATURE13"

Input

- **P1** = Name of a saved feature (String; optional)

P1 is the name of a saved feature to be used to establish the axis references. If P1 is not specified, this defaults to the feature in the Feature Register.

Results

The part reference frame is translated so that the datum (zero point) is coincident with the position of the specified feature. The position related information in the Dimensional Array is updated to reflect this change. Corresponding data in the Feature Register, the Feature Stack and Feature Storage changes to reflect the new datum.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 The specified feature does not exist.
- MLB-043 A segmented measurement is in progress.

EX - Establish X Reference

The EX command establishes the X axis datum. It sets the X-axis datum (zero point) to the location defined by a specified feature position.

Command Format

MLString "EX P1"

Examples

1. MLString "EX"
2. MLString "EX FEATURE13"

Input

- **P1** = Name of a saved feature (String; optional)

P1 is the name of a saved feature to be used to establish the X-axis reference. If P1 is not specified, this defaults to the feature in the feature register. The second example shows the system moving the active reference frame X-axis reference so that the X-axis coordinate of the feature named "FEATURE13" is zero.

Results

The part reference frame is translated so that the X-axis datum (zero point) is coincident with the position of the specified feature. The position related information in the Dimensional Array is updated to reflect this change. Corresponding data in the Feature Register, the Feature Stack and Feature Storage changes to reflect the new datum.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 The Feature Register does not contain the required feature.
- MLB-043 A segmented measurement is in progress.

EY - Establish Y Reference

The EY command establishes the Y axis datum. It sets the Y-axis datum (zero point) to the location defined by a specified feature position.

Command Format

MLString "EY P1"

Examples

1. MLString "EY"
2. MLString "EY FEATURE13"

Parameters

- **P1** = Name of a saved feature (String; optional)

P1 is the name of a saved feature to be used to establish the Y-axis references. If P1 is not specified, this defaults to the feature in the feature register. The second example shows the system moving the active reference frame Y-axis reference so that the Y-axis coordinate of the feature named "FEATURE13" is zero.

Results

The part reference frame is translated so that the Y-axis datum (zero point) is coincident with the position of the specified feature. The position related information in the Dimensional Array is updated to reflect this change. Corresponding data in the Feature Register, the Feature Stack and Feature Storage changes to reflect the new datum.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 The Feature Register does not contain the required feature.
- MLB-043 A segmented measurement is in progress.

EZ - Establish Z Reference

The EZ command establishes the Z axis datum. It sets the Z-axis datum (zero point) to the location defined by a specified feature position.

Command Format

MLString "EZ P1"

Examples

1. MLString "EZ"
2. MLString "EZ FEATURE13"

Input

- **P1** = Name of a saved feature (String; optional)

P1 is the name of a saved feature to be used to establish the Z-axis references. If P1 is not specified, this defaults to the feature in the feature register. The second example shows the system moving the active reference frame Z-axis reference so that the Z-axis coordinate of the feature named "FEATURE13" is zero.

Results

The part reference frame is translated so that the Z-axis datum (zero point) is coincident with the position of the specified feature. The position related information in the Dimensional Array is updated to reflect this change. Corresponding data in the Feature Register, the Feature Stack and Feature Storage changes to reflect the new datum.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 The Feature Register does not contain the required feature.
- MLB-043 A segmented measurement is in progress.

LV - Level

The LV command establishes the orientation of the working plane based upon a specified feature. If the specified feature is a plane, this command rotates or "tilts" the active working plane so that it is parallel to that plane. If the feature is a line or centerline of a cylinder or cone, this command rotates the active working plane so that it is perpendicular to the line or axis.

Command Format

MLString "LV P1,P2"

Input

- **P1** = X, -X, Y, -Y, Z, -Z, EXT, INT, or OFF (String; optional).

- **P2** = Saved feature name (String; optional)

P1 is used to control the direction that is positive (up) relative to the working plane after its rotation. It may be X, -X, Y, -Y, Z, or -Z to designate a machine axis that points generally in the direction desired for the positive end of the axis normal to the working plane. In the second example, -Z is a machine axis. The system mathematically rotates the working plane so that the positive end of the axis of rotation points in the -Z direction.

If P1 is OFF or EXT then the positive direction is to be defined by the plane's direction vector. This assumes that appropriate probe compensation was specified when the plane was measured so that the direction vector points away from the surface. If P1 is INT then the positive direction is to point into the surface rather than away from it.

If P1 is not specified then the positive direction is defined by the orientation of the machine axis corresponding to the axis normal to the working plane.

P2 is the name of a saved feature to be used to establish the alignment angles. If P2 is not specified, this defaults to the feature in the feature register. The feature must be a plane, line, cylinder or cone. If the feature is a plane, the resultant active plane will be parallel to it. If the feature is a line, cylinder, or cone, the resultant active plane will be perpendicular to the line (or feature axis).

Results

The working plane is rotated to be parallel to the specified plane. The position and/or orientation data for the feature in the feature register and in the Dimensional Array is updated to reflect the new orientation of the working plane.

Note: The only point in space whose coordinates will not be affected by this operation is the datum point (0,0,0).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 The feature register does not contain the required feature and P2 isn't specified.
- MLB-031 The feature specified is of an invalid type.
- MLB-043 A segmented measurement is in progress.

MO - Move Offset Angle

The MO command rotates the current reference frame active working plane about the axis normal to that plane. The amount and direction of rotation is controlled by specifying an angle (in degrees) or a set of point coordinates.

Command Format

1. MLString "MO P1,P2"

Examples

1. MLString "MO -35"

2. MLString "MO 25.235,47.468"

Input

- **P1** = Size and direction when used alone; or 1st-named axis value when used with P2 (Numeric; required).
- **P2** = Second-named axis value (Numeric; optional).

P1 is either an angle in degrees or a coordinate value for the first axis of the working plane. In the first command format, P1 specifies a positive or negative direction and amount to rotate the working plane. In the first example, rotation is in the negative direction. The amount of rotation is 35 degrees.

In the second command format, P1 and P2 together specify the coordinates of a point in the working plane. When P1 and P2 are used, P1 is the first-named axis and P2 is the second-named axis of the working plane. In the second example, assuming that the active plane is XY, P1 is an X-coordinate value of 25.235 and P2 is a Y-coordinate value of 47.468. The working plane is rotated so that the point (25.235,47.468) will lie on the new X-axis.

Results

The active coordinate reference frame is updated to reflect the modified offset angle. The actual data in the feature register and the dimensional array is updated to reflect the new reference frame orientation.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-043 A segmented measurement is in progress.

MR - Move Reference

The MR command moves the datum point along one or more of the axes. In affect, the system subtracts the amount the datum moves from the previous actual coordinates.

Command Format

MLString "MR P1,P2,P3"

Examples

1. MLString "MR 9.765,27.535,-1.000"
2. MLString "MR X2,Y2,Z2"

Input

- **P1** = Distance to move X (Numeric/parametric; Required).
- **P2** = Distance to move Y (Numeric/parametric; Required).
- **P3** = Distance to move Z (Numeric/parametric; Required).

P1 specifies the distance to move the X-coordinate. In the first example, the coordinate moves 9.765 units. In the second example, the X2 specifies the X- coordinate in the second position in the Feature Stack.

P2 specifies the distance to move the Y-coordinate. In the first example, the coordinate moves 27.535 units. The Y2 specifies the Y-coordinate in the second position in the Feature Stack.

P3 specifies the distance to move the Z-coordinate. In the first example, the coordinate moves -1.000 units. In the second example, the Z2 specifies the Z- coordinate in the second position in the Feature Stack.

Results

The system updates the position data in the coordinate display, the Dimensional Array, the Feature Register, the Feature Stack, and Feature Storage to reflect the new datum position.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-030 The Feature Stack or Feature Register does not contain the required features.
- MLB-043 A segmented measurement is in progress.

MX - Move X Reference

The MX command moves the datum point along the X-axis.

Command Format

MLString "MX P1"

Examples

1. MLString "MX 1.492"

Input

- **P1** - numeric/parametric - required

P1 specifies the distance that the datum point moves along the X-axis. In the example, the datum moves 1.492 units along the X-axis.

Results

The datum moves by the specified amount. In affect, the system subtracts the specified amount from the previous X-axis coordinate. The system updates the actual position data in the Dimensional Array.

Corresponding data in the Feature Register, the Feature Stack, and Feature Storage changes to reflect the new datum.

Skip Action

Not executed when inside a block of code being skipped.

Error Messages

- MLB-030 The Feature Stack or Feature Register does not contain the required features.
- MLB-043 A segmented measurement is in progress.

MY - Move Y Reference

The MY command moves the datum point along the Y-axis.

Command Format

MLString "MY P1"

Examples

1. MLString "MY -7.368"

Input

- **P1** - numeri/parametric - required

P1 specifies the distance that the datum point moves along the Y-axis. In the example, the datum moves -7.368 units along the Y-axis.

Results

The datum moves by the specified amount. In affect, the system subtracts the specified amount from the previous Y-axis coordinate. The system updates the actual position data in the Dimensional Array.

Corresponding data in the Feature Register, the Feature Stack, and Feature Storage changes to reflect the new datum.

Skip Action

Not executed when inside a block of code being skipped.

Error Messages

- MLB-030 The Feature Stack or Feature Register does not contain the required features.
- MLB-043 A segmented measurement is in progress.

MZ - Move Z Reference

This command moves the datum point along the Z-axis.

Command Format

MLString "MZ P1"

Examples

1. MLString "MZ 87.5702"

Parameters

- **P1** - numeric/parametric - required

P1 specifies the distance that the datum point moves along the Z-axis. In the example, the datum moves 87.5702 units along the Z-axis.

Results

The datum moves by the specified amount. In affect, the system subtracts the specified amount from the previous Z-axis coordinate. The system updates the actual position data in the Dimensional Array.

Corresponding data in the Feature Register, the Feature Stack, and Feature Storage changes to reflect the new datum.

Skip Action

Not executed when inside a block of code being skipped.

Error Messages

- MLB-030 The Feature Stack or Feature Register does not contain the required features.
- MLB-043 A segmented measurement is in progress.

NO - Cancel Level

The NO command cancels part leveling and aligns the part reference frame with the machine reference frame.

Command Format

MLString "NO"

Examples

1. MLString "NO"

Input

None.

Results

The position and orientation data in the Dimensional Array reflect the new orientation.

Error Messages

- MLB-043 A segmented measurement is in progress.

PS - Preset Datum

This command establishes the XYZ datum relative to the current position of the probe. For undeflected touch probes, this is based upon the position of the probe's center. For deflected touch probes, this is based upon the point where deflection occurred.

Command Format

MLString "PS P1,P2,P3"

Examples

1. MLString "PS 25.445,14.879,-11.134"
2. MLString "PS X3,Y3,Z5"

Input

- **P1** = numeric/parametric - required
- **P2** = numeric/parametric - required
- **P3** = numeric/parametric - required

P1 specifies the X-coordinate for the current/touched probe position. In the example, the X-coordinate is 25.445.

P2 specifies the Y-coordinate at the current/touched probe position. In the example, the Y-coordinate is 14.879.

P3 specifies the Z-coordinate at the current/touched probe position. In the example, the Z-coordinate is -11.134.

Results

The reference frame is updated to reflect the new XYZ datum. Any data in the Feature Register is updated to reflect the modified reference frame. Actual positional data in the Dimensional Array is also updated.

Skip Action

Ignored. Not executed if inside a block of code being skipped.

Error Messages

- MLB-043 A segmented measurement is in progress.

PX - Preset X Datum

The PX command establishes the X axis datum relative to the current position of the probe. For undeflected touch probes, this is based upon the position of the probe's center. For deflected touch probes, this is based upon the point where deflection occurred.

Command Format

MLString "PX P1"

Examples

1. MLString "PX 56.235"

Input

- **P1** = numeric/parametric - required

P1 specifies the X-coordinate for the current/touched probe position. In the example, the X coordinate is 56.235.

Results

The reference frame is updated to reflect the new X axis datum. Any data in the Feature Register is updated to reflect the modified reference frame. Actual positional data in the Dimensional array is also updated.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-043 A segmented measurement is in progress.

PY - Preset Y Datum

The PY command establishes the Y axis datum relative to the current position of the probe. For undeflected touch probes, this is based upon the position of the probe's center. For deflected touch probes, this is based upon the point where deflection occurred.

Command Format

MLString "PY P1"

Examples

1. MLString "PY 46.235"

Input

- **P1** = numeric/parametric - required

P1 specifies the Y-coordinate for the current/touched probe position. In the example, the Y coordinate is 46.235.

Results

The reference frame is updated to reflect the new Y axis datum. Any data in the Feature Register is updated to reflect the modified reference frame. Actual positional data in the Dimensional Array is also updated.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-043 A segmented measurement is in progress.

PZ - Preset Z Datum

The PZ command establishes the Z axis datum relative to the current position of the probe. For undeflected touch probes, this is based upon the position of the probe's center. For deflected touch probes, this is based upon the point where deflection occurred.

Command Format

MLString "PZ P1"

Examples

1. MLString "PZ 35.235"

Input

- **P1** - numeric/parametric - required

P1 specifies the Z-coordinate for the current/touched probe position. In the example, the Z coordinate is 35.235.

Results

The reference frame is updated to reflect the new Z axis datum. Any data in the Feature Register is updated to reflect the modified reference frame. Actual positional data in the Dimensional Array is also updated.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-043 A segmented measurement is in progress.

RC - Reference Frame Temporary Change

The RC command will temporarily change the part reference frame so as to align the actual data of a plane with the nominal data.

Command Format

MLString "RC P1,P2,P3"

Examples

1. MLString "RC Plane5,2.5,3,5.2,0,0,1"

2. MLString "RC F3,3,3,3,0,1,0"
3. MLString "RC"

Input

- **P1** = String (Optional)).
- **P2** = 3 Numbers (Optional).
- **P3** = 3 Numbers (Optional)

Optional parameter P1 specifies a stored feature's name as set by the **SF** command, or by its position in the feature stack. In the first example, Plane5 is the stored feature name.

The optional parameter P2 is the nominal position or X,Y,Z of the feature.

The optional parameter P3 are the nominal I,J,K direction cosines normal to the plane.

If P1 is specified, then P2 and P3 must be specified.

If all parameters are omitted, the original reference frame will be restored.

Results

This command modifies the reference frame transformation data so that the transformed orientation matches the supplied nominal orientation (P3), and the position of the transformed plane is normal to the nominal position (P2). When applied, the Dimensional Array will reflect this new transformation. The original reference frame in effect at this time is saved.

Invoking an "RC" with no parameters will restore the original reference frame (as in example 3).

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-032 The feature is not a plane.
- MLB-070 Attempt to restore a reference frame that was not previously saved.

RO - Read Offset Angle

The RO command retrieves the offset angle of the current reference frame.

Command Format

MLString "RO"

Examples

1. MLString "RO"

Input

None.

Results

The offset angle of the current reference frame returns to variable S7.

Skip Action

Executed normally.

Error Messages

None.

RR - Recall Reference Frame

The RR command recalls a previously saved part reference frame (see the **SR** command).

Command Format

MLString "RR P1"

Examples

1. MLString "RR"

   Recall saved reference frame "1" and make it the active part coordinate reference frame.

2. MLString "RR 16"

   Recall saved reference frame "16".

Input

- P1 = 1 to 48 or an alphanumeric label (Numeric/string; optional).

  Reference Frames 17 to 48 are available for MP30/35, SMP-3xx Version 19.0 and later, and SMP-400 Version 104.0 and later. Reference frames may be saved by name for MeasureMax+ 6.0 and later systems.

P1 identifies reference frame that is to be retrieved. Omitting P1 selects reference frame 1. In the second example, the system retrieves part reference frame 4.

Results

The data for the active reference frame is recalled and made active. Data in the feature stack and named storage is updated to reflect the reference frame change. Actual data in the dimensional array is also updated.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-043 A segmented measurement is in progress.
- MLB-070 The reference frame was not previously saved.

RT - Stored Reference Frame Status

The RT command determines if a set of part reference frame data has been saved and, if so, to retrieve the description (if any) saved with the reference frame data.

Command Format

MLString "RT P1"

Examples

1. MLString "RT 9"

Input

- P1 = 1 to 48 or alphanumeric label (Numeric/string; optional).

  Reference Frames 17 to 48 are available for MP30/35, SMP-3xx Version 19.0 and later, and SMP-400 Version 104.0 and later. Reference frames may be saved by name for MeasureMax+ 6.0 and later systems.

P1 identifies which storage location to examine. Omitting P1 selects reference frame 1. In the example, the reference frame 9 is examined.

Results

The status of the specified reference frame returns to variable S7. S7=1 indicates that it is a valid saved reference frame. S7=0 indicates that the reference frame is invalid.

If a reference frame has been saved and if a description was save with that reference frame (see the **SR** command), then the description is returned to system variable S0D.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

SL - Set Level

The SL command specifies autolevel direction cosines for aligning the working plane.

Note: Do not use the **SL** command if any other level command can be used. Entering for the autolevel direction cosines, instead of using commands to calculate the autolevel based upon actual feature measurements, can cause errors throughout the program.

Command Format

MLString "SL P1,P2,P3"

Examples

1. MLString "SL .707106781,.5,.5"

Input

- **P1** = X direction cosine (Numeric; required).
- **P2** = Y direction cosine (Numeric; required).
- **P3** = Z direction cosine (Numeric; required).

P1, P2, and P3 together describe the orientation of the axis perpendicular to the working plane. At least one of the three values must be non-zero.

In the example, .707106781 is the direction cosine of the vector relative to the X-axis. The value is the cosine of the angle between the vector and the X-axis.

The first .5 in the example is the direction cosine of the vector relative to the Y-axis. The value is the cosine of the angle between the vector and the Y-axis.

The second .5 in the example is the direction cosine of the vector relative to the Z-axis. The value is the cosine of the angle between the vector and the Z-axis.

Results

The specified vector is normalized to a unit vector and the reference frame is modified so that the working plane is normal (perpendicular) to the specified vector. The position and orientation data in the Dimensional Array is updated to reflect the change in the part reference frame.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-043 A segmented measurement is in progress.

SO - Set Offset Angle

The SO command specifies offset angle to rotate the working plane about the third coordinate axis.

Note: Do not use the **SO** command if any other offset command can be used. Entering values for the offset angle, instead of using commands to calculate the offset angle based upon actual feature measurements, can cause errors throughout the program.

Command Format

MLString "SO P1"

Examples

1. MLString "SO 45.225"

Input

- **P1** = Offset angle in degrees (Numeric; required).

P1 specifies the offset angle in degrees. In the example, the positive angular value of the offset angle is 45.225 degrees.

Results

The actual position and orientation data in the Dimensional Array is updated to reflect the change to the active part reference frame.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-043 A segmented measurement is in progress.

SR - Save Reference Frame

The SR command saves the current reference frame.

Command Format

MLString "SR P1,P2"

Examples

1. MLString "SR"

   Save the current reference frame data as reference frame 1.

2. MLString "SR 16"

   Save the current reference frame data as reference frame 16.

3. MLString "SR ABC,@Front face of engine block@"

   Save the current reference frame data as reference frame ABC and save the string "Front face of engine block" with it.

Input

- P1 = 1 to 48 or alphanumeric label (Numeric/string; optional).

  Reference Frames 17 to 48 are available for MP30/35, SMP-3xx Version 19.0 and later, and SMP-400 Version 104.0 and later. Reference frames may be saved by name for MeasureMax+ 6.0 and later systems. Reference frame names can be no more than 64 characters long and can consist only of letters (A, ... , Z), digits (0, ..., 9), underline, and hyphen characters.

- P2 = @Maximum length of 30 characters@ (String; optional).

P1 is the identifier used to save the data for the current reference frame. If P1 is omitted then P1 defaults to "1".

P2 is a string to be saved with the reference frame data that describes the reference frame. P2 must be omitted if P1 is omitted.

Results

The data for the active reference frame is stored using the specified label and is then available for later recall using the RR command. The descriptor string p2 (if any) is also stored and is available for later recall using the RT command.

NOTE: Reference frame labels 1 through 48 are treated as semi-permanent reference frame storage. Data saved with these labels is not deleted at a part program initialization. Data saved under any other label is treated as temporary and is deleted at a part program initialization.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

XI - LOM X Inch Increment / Decrement

The XI command updates the MP transformations when the layout machine's (LOM) linear bar is relocated along the machine base. This command assumes that the linear bar is moved a specific number of multiples of 10.0", the spacing between grid lines on an "inch" LOM table.

Command Format

MLString "XI P1"

Examples

1. MLString "XI 1"
2. MLString "XI -4"

Input

- **P1** = Number of table slots machine has been moved (Numeric; required).

P1 is the number of table slots that the machine has been moved. A positive value indicates that the machine was relocated in the positive direction. A negative value indicates that the machine was relocated in the negative direction. In the first example, the machine was moved one slot in the positive direction. In the second example, the machine was moved four slots in the negative direction.

Results

The MP transformation is updated by adding the specified multiple of 10.0" to the X-axis table translation vector. Features measured after this operation reflect this motion while previously measured features are not affected.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

XM - LOM X Metric Increment / Decrement

The XM command updates the MP transformation when the layout machine's (LOM) linear bar is relocated along the machine base. This command assumes that the linear bar is moved a specific number of multiples of 200.0 mm, the spacing between grid lines on a "metric" LOM table.

Command Format

MLString "XM P1"

Examples

1. MLString "XM 1"
2. MLString "XM -4"

Input

- **P1** = Number of table slots machine has been moved (Numeric; required).

P1 is the number of table slots that the machine has been moved. A positive value indicates that the machine was relocated in the positive direction. A negative value indicates that the machine was relocated in the negative direction. In the first example, the machine was moved one slot in the positive direction. In the second example, the machine was moved four slots in the negative direction.

Results

The MP transformation is updated by adding the specified multiple of 200.0 mm to the X-axis table translation vector. Features measured after this operation reflect this motion while previously measured features are unaffected.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

XY - XY Plane

The XY command defines the XY plane as the working plane. At part program initialization the XY plane is the active working plane.

Command Format

MLString "XY"

Examples

1. MLString "XY"

Input

None.

Results

The X axis becomes the reference for polar angles. Z is the axis normal to the working plane about which the offset angle is measured. X is the first-named axis of the working plane and Y is the second-named axis. Polar radius and polar angle values in the Dimensional Array reflect the new working plane.

Variables S1, S2 and S3 are active working plane indices to the Dimensional Array. They change so that:

- S1=2 indicating X
- S2=3 indicating Y
- S3=4 indicating Z

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-043 A segmented measurement is in progress.

YZ - YZ Plane

The YZ command defines the YZ plane as the working plane.

Command Format

MLString "YZ"

Examples

1. MLString "YZ"

Input

None.

Results

The Y axis becomes the reference for polar angles. X is the axis normal to the working plane about which the offset angle is measured. Y is the first-named axis of the working plane and Z is the second-named axis. Polar radius and polar angle values in the Dimensional Array reflect the new working plane.

Variables S1, S2 and S3 are active working plane indices to the Dimensional Array. They change so that:

- S1=3 indicating Y
- S2=4 indicating Z
- S3=2 indicating X

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-043 A segmented measurement is in progress.

ZX - ZX Plane

The ZX command defines the ZX plane as the working plane.

Command Format

MLString "ZX"

Examples

1. MLString "ZX"

Input

None.

Results

The Z axis becomes the reference for polar angles. Y is the axis normal to the working plane about which the offset angle is measured. Z is the first-named axis of the working plane and X is the second-named axis. Polar radius and polar angle values in the Dimensional Array reflect the new working plane.

Variables S1, S2 and S3 are active working plane indices to the Dimensional Array. They change so that:

- S1=4 indicating Z
- S2=2 indicating X
- S3=3 indicating Y

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-043 A segmented measurement is in progress.

3. Probe Calibration Group

MLB commands in the Probe Calibration Group let you establish a probe tip's center. A probe tip's center is relative to the coordinate measuring machine and to other probes. A probe tip's center also varies according to the probe shaft's length and the tip's diameter and attitude. If you change probe tips or alter a probe tip's attitude, recalibrate the probe tip. If the system loses power, you do not need to recalibrate probe tips.

The Measurement Library provides three calibration commands for Direct Computer-Controlled (DCC) machines. They are DCC Calibration Tip 0, DCC Calibrate Tip and DCC Locate Calibration Sphere. These commands include automatic and semi-automatic sequences for DCC measurement of the calibration sphere.

The number of tips that can be calibrated is unlimited. This allows you to calibrate all the tips you will use in a measurement task before you begin the task. A part program can use any calibrated probe tip without interrupting measurement tasks. Each time a tip is calibrated, the form of the sphere is stored in the S7 variable.

To calibrate a probe tip, you define a calibration fixture. If the calibration fixture is a sphere, and if the part defined the sphere's diameter, then the system automatically determines and stores the diameter of each probe tip you calibrate. The system uses this data when it measures a feature that requires compensation for probe diameter.

The Probe Calibration Group includes the following MLB commands.

AH	**DCC Calibrate Tip 0** directs the automatic or semi-automatic calibration of Tip 0.
AP	**Auto-sphere Measurement Point**
AT	**DCC Calibrate Additional Tips** directs the automatic or semi-automatic measurement of a calibration sphere with a tip other than Tip 0.
C6	**Calibrate PH6** calibrates a PH6A or PH9A probe head adapter using a probe changer sphere.
CA	**Calibrate Using Best-fit Data** calibrates probe tips from data in a summation array.
CE	**Calibrate Extension** calibrates a probe extension.
CF	**Calibrate Fixture** defines the type, location and size of a calibration fixture.
CH	**Calibrate Tip 0** lets you calibrate Tip 0 manually.
CT	**Calibrate Additional Tips** calibrates multiple probe tips and identifies tip positions by measuring a calibration fixture.
D9	**Define PH9 Safe Position** establishes a position where the PH9 can safely change probe tip attitudes prior to loading or unloading a station.
LA	**DCC Locate Calibration Sphere** directs the automatic or semi-automatic measurement of a moved calibration sphere after an initial calibration at the sphere's MEA location.

LC	**Locate Calibration Sphere** locates the calibration sphere on the worktable for a manual measurement after an initial calibration at the sphere's MEA location.
M9	**Mover PH9** rotates the PH9 probe in A or B axis or both.
RI	**Read Probe Data** tells you a probe's offset and radius.
SI	**Station Information** tells you if a station is loaded or unloaded. A station is loaded if the probe or probe extension from that station is currently mounted on the CMM probe shaft. Station Information also tells you whether or not a station is calibrated.
SQ	**Calibration Setup** tells the measurement processor to use the measurements recorded on the calibration sphere to calibrate a probe tip.
TI	**Tip Status** tells you if a probe tip is calibrated.

AH - DCC Calibrate Tip

The AH command directs the automatic or semi-automatic calibration of Tip 0. It requires that a touch probe and a calibration sphere be used. This command only applies to DCC operations. Note: Use the **AH** command only for the first calibration of Tip 0 or if the calibration sphere is in its original position on the table. If the calibration fixture is moved, use the **LA** command.

SEMI-AUTOMATIC CALIBRATION:

For semi-automatic calibration, the operator must define two points in space. Define the first point approximately 150 millimeters or 6 inches from the calibration sphere. Define the second point approximately 5 millimeters or .30 inches from the calibration sphere. These points define a line that is parallel to the probe shaft. When this line is extended, it should pass through the approximate center of the calibration sphere. Measuring the calibration sphere determines the displacement between old and new Tip 0.

AUTOMATIC CALIBRATION:

The automatic calibration sequence assumes that the probe has been positioned (either manually or under DCC control) at the first position. You define the drive vector in the part program. If a calibration sphere with a diameter other than the standard 38.1 millimeter (1.5 inches) is in use, use the **CF** command to specify the sphere's diameter. If the system-calculated probe size is not desired, use the **PR** command immediately after the probe is calibrated to specify the probe's diameter.

If Tip 0 has been used to measure the calibration sphere since the system was turned on, using the **AH** command indicates either a new tip or a new position for Tip 0.

____

If:

- Tip 0 has not been used to measure the calibration sphere since the system was turned on, and
- The sphere is defined as SP, SPON, or SPOF,

then:

- The **AH** command only determines the location of the sphere relative to Tip 0.

If:

- The sphere is defined as FX, FXON, or FXOF,

then:

- The **AH** command uses the location of the sphere relative to the end of the probe shaft as calibrated at the factory.
- The **AH** command determines the offset between Tip 0 and the end of the probe shaft.
- The **AH** command sets the sphere position relative to Tip 0 for later use.

Probe calibration using DCC operations includes special instructions for semi-automatic calibration. The semi-automatic calibration sequence instructs the recording of two points defining the drive vector. The system then instructs the operator to switch to AUTO mode to complete the calibration.

Command Format

MLString "AH P1,P2,P3,P4"

Examples

1. MLString "AH 5,L17,D12,2,-5,R2.5"

   Calibrate tip #0 by measuring 5 points on the calibration sphere under DCC control. The probe has an approximate tip radius of 2.5 units and a shaft length of 17 units. Calibration is completely under DCC control beginning with a seek along the part coordinate vector X12,Y2,Z-5 from the current position towards the center of the calibration sphere.

2. MLString "AH 6,R1.775,@TP2 with 20 mm X 4mm Stylus@"

   Calibrate tip #0 by measuring 6 points on the calibration sphere under DCC control. The probe has an approximate tip radius of 1.775 units and a shaft length of at least one half of the sphere radius. Calibration begins by prompting the operator to record two probe positions, one 150 mm (6 inches) and the other 6 mm (0.25 inches) away from the calibration sphere. Thereafter probe calibration is completely under DCC control. Save the string "TP2 with 20mm X 4mm Stylus" as a probe descriptor.

Inputs

These arguments must be specified in numerical order.

- **P1** = Number of points (Numeric; optional).
- **P2** = L(ength) and probe shaft length (Numeric; optional).
- **P3** = D(irection) or M followed by X, Y, Z (Numeric; optional).
- **P4** = R(adius) and probe tip radius (Numeric; optional).
- **P5** = @Maximum of 30 characters@ (String; optional).

P1 specifies the number of points to record on the spherical fixture. If P1 is omitted, the system uses the value defined by the **CF** command. If the **CF** command does not define the number of points, the system defaults to 4 points or 10 points. In the first example, [MLSTRING("AH 5,L17,D12,2,-5,R2.5")] the system records 5 points. In the second example [MLSTRING("AH 6,R1.775")], the system records six points.

P2 consists of the letter L followed by the length of the probe shaft. The shaft's length is the distance from the center of the probe tip to the first bend or obstruction along the probe shaft.

If the shaft length is less than the calibration sphere radius, include P2. In the first example, the length is L17.

If P2 is omitted, the system defaults to the calibration sphere radius. The second example omits P2.

P3 starts automatic probe calibration. P3 consists of either the letter D or the letter M, followed by a signed direction vector. The direction vector consists of three numbers separated by commas. In the example, the vector is D 12, 2, -5. The D indicates that the vector is in the currently active reference frame (e.g., D 0, 0, -1) while the letter M indicates that the letter is in the machine reference frame (e.g., M 0, 1, 0). If present, P3 invokes the automatic probe calibration and causes the first touch to be made by moving from the current probe position towards the calibration sphere using the specified vector. If P3 is not present, then the operator is prompted to define this drive vector.

P4 specifies the approximate radius of the probe tip in current measurement units. P4 consists of the letter R followed by the radius. The system uses the radius to plan the automatic motions of the probe around the calibration sphere. In the first example, the radius is 2.5 units. Example 2 shows a radius of 1.775 units. If P4 is omitted, the system uses a default probe radius of 1 millimeter (0.03937 inches).

P4 does not change the probe size defined by the **PR** command or computed by the calibration sequence.

P5 is a string of any printable characters enclosed in @ signs. P5 is saved by the system as a descriptor for the calibrated tip. If P5 is greater than 30 characters, it is truncated by deletion of the rightmost characters. Trailing blanks are ignored.

Results

Tip 0 is calibrated, calibration data is stored and added to the inspection report.

For AUTOMATIC CALIBRATION, **(AM)** commands must have positioned the probe near the sphere before executing the **AH** command. P3 starts automatic calibration. The drive vector defines the direction that the probe is to move from its last position in seeking towards the sphere. Sphere measurement then continues as in semi-automatic operation.

The results of these measurements determine either the location of the calibration sphere or the amount of change in attitude of Tip 0.

As a result of the calibration, the following information is output to the inspection report: tip name; probe changer stations (if applicable); PH9/PH10 angles (if applicable); tip diameter; tip offsets; form of measured sphere. After the command is complete the measured sphere form is stored in variable S7.

Later tip calibrations use this information to determine the relative offsets between Tip 0 and other probe tips.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-080 The system does not have the required probe.
- MLB-090 DCC is not available.
- MLB-091 The probe tip does not make contact with the required object.
- MLB-111 The calibration fixture is not a sphere.

AP - Auto-sphere Measurement Point Control

The AP command control of the distribution of points measured during auto-sphere probe calibrations.

Command Format

MLString "AP P1"

Example

- MLString "AP 4"
- MLString "AP"

Input

- **P1** = Number of points per level (Numeric, optional).

The optional parameter P1 defines the number of points to be measured around the calibration sphere during auto-sphere operations (see **AH**, **AT**, and **LA**). If P1 is not specified or is less than 3, the system uses a default value of three. When the calibration sphere is subsequently measured for probe calibration (using one of the above commands) the specified number of points are measured by taking one point on top and then by measuring 'P1' points around the sphere at levels equally spaced between the lowest level available and the top of the sphere. Fewer than 'P1' points are measured at the last level measured to obtain the correct total number of points.

Note: This setting defaults to three at power-up and full initialization. Execution of this command does not change the total number of points to be measured during probe calibration.

In the first example, four points are to be measured at each level on the calibration sphere during **AH**, **AT**, and **LA** commands. The total number of points to be measured can be set by **CF**, **AH**, **AT**, or **LA** commands. For example, if nine total points are to be measured, one point is taken on top, then a group of four spaced ninety degrees apart around the sphere at the latitude closest to the equator, and finally a second group of four halfway between that level and the top of the calibration sphere.

In the second example, three points (the default) are to be measured) at each level on the calibration sphere during **AH**, **AT**, and **LA** commands.

Results

Subsequent automatic probe calibration operations **(AH**, **AT**, and **LA**) are performed by measuring the specified total number of points, taking one point on top of the sphere and then 'P1' points around the sphere at levels equally spaced between the lowest level available and the top of the sphere. Fewer than 'P1' points may be measured at the last level to obtain the correct total number of points. At each level the points are spaced equally round the sphere based upon the setting of 'P1'. That is, they are separated by 360 divided by 'p1' degrees. The first point at the "equator" will be measured 45 degrees away from the assumed sphere support (-Y or -Z) unless the user has specified the exact sphere support vector using the **CF** command. In the latter case the first "equator" touch is positioned so that touches at the equator are equally displaced around the sphere relative to the sphere support.

Skip Action

This command is ignored (not executed) if it occurs within a block of code being skipped.

Error Messages

- MLB-021 First parameter is invalid.

AT - DCC Calibration of Tips Other Than Tip #0

The AT command directs the automatic or semi-automatic measurement of a calibration sphere with a tip other than Tip 0. This measurement determines the position of the new tip relative to Tip 0. The **AT** command requires the use of a touch probe and a calibration sphere. This command is applicable only to DCC operations.

SEMI-AUTOMATIC CALIBRATION:

For semi-automatic calibration, the operator must define two points in space. Define the first point approximately 150 millimeters or 6 inches from the calibration sphere. Define the second point approximately 5 millimeters or .30 inches from the calibration sphere.

The system prompts for AUTO mode when necessary and then measures the calibration sphere with the designated tip.

AUTOMATIC CALIBRATION:

The automatic calibration sequence assumes that the probe has been positioned (either manually or under DCC control) so that the probe tip is located near and directed at the sphere. A vector is defined in the **AT** command indicating the direction that the probe is to move to contact the sphere.

The difference between the results of the measurements and the previous measurement of the sphere with Tip 0 determines the XYZ offsets between Tip 0 and the calibrated tip. The system stores this difference vector and uses it whenever the part program selects the tip for measurement.

The most recently calibrated tip remains in the current tip position until the part program selects another tip. See the **GT** command for information regarding selecting the probe tips. If a calibration sphere is used with a diameter other than the standard 38.1 millimeter (1.5 inches), use the **CF** command to specify the sphere's diameter. If the system-calculated probe size is not desired, use the **PR** command after the probe is calibrated to specify the probe's diameter.

Command Format

MLString "AT P1,P2,P3,P4,P5,P6"

Examples

1. MLString "AT 12,5,L17,D12,2,-5,R2.5"

Calibrate tip #12 by measuring 5 points on the calibration sphere under DCC control. The probe has an approximate tip radius of 2.5 units and a shaft length of 17 units. Calibration is completely under DCC control beginning with a seek along the part coordinate vector X12,Y2,Z-5 from the current position towards the center of the calibration sphere.

2. MLString "AT 154,6,R1.775,@TP2 with 20mm X 4mm Stylus@"

   Calibrate tip #154 by measuring 6 points on the calibration sphere under DCC control. The probe has an approximate tip radius of 1.775 units and a shaft length at least one half of the sphere radius. Calibration begins by prompting the operator to record two probe positions, one 150 mm (6 inches) and the other 6 mm (0.25 inches) away from the calibration sphere. Thereafter probe calibration is completely under DCC control. Save the string "TP2 with 20mm X 4mm Stylus" as a probe descriptor.

3. MLString "AT Tip_X5,@TP2 with 20mm X 4mm Stylus@"

   Calibrate tip "Tip_X5" by measuring the default number of points on the calibration sphere under DCC control. Moves around the sphere are to be planned using a estimated probe radius of 1 mm. The probe has a shaft length at least one half of the sphere radius. Calibration begins by prompting the operator to record two probe positions, one 150 mm (6 inches) and the other 6 mm (0.25 inches) away from the calibration sphere. Thereafter probe calibration is completely under DCC control. Save the string "TP2 with 20mm X 4mm Stylus" as a probe descriptor.

Inputs

If more than one of these arguments is used, use them in numerical order.

- **P1** =Tip I.D. (String (maximum of ten characters); required).
- **P2** = Number of points (Numeric; optional).
- **P3** = L(ength) and shaft length (Numeric; optional).
- **P4** = D(irection) or M followed by X, Y, Z (Numeric; optional).
- **P5** = R(adius) and probe tip radius (Numeric; optional).
- **P6** = @Maximum of 30 characters@ (String; optional).

The required parameter p1 defines the identifier of the tip to be calibrated. The parameter p1 must be a string label (maximum length 10; only numeric, alphabetic, and underline characters permitted). If p1 is greater than 10 characters than it will be truncated by deletion of the rightmost characters. Note that the label '0' is reserved for tip #0.

The optional numeric parameter p2 is the number of points to be measured on the calibration sphere. If omitted the default number of points (see CF command) is used.

The optional parameter p3 consists of the letter 'L' followed by the length of the probe shaft. The length of the probe shaft is the distance from the center of the probe tip to the first bend or obstruction along the probe shaft. This must be specified if it is less than one-half of the calibration sphere radius.

The optional parameter p4 consists of either the letter 'D' or the letter 'M' followed by a signed direction vector. The 'D' indicates that the vector is in the currently active part reference frame (e.g., D0,0,-1) while the 'M' indicates that the vector is in the machine reference frame (e.g., M0,1,0). If present p4 invokes automatic probe calibration and causes the first touch to be made by moving from the current probe position towards the calibration sphere using the specified vector. If p4 is not present then the operator is prompted to define this drive vector.

The optional parameter p5 consists of the letter 'R' and defines the approximate radius of the probe tip in current measurement units. If omitted, the system will plan moves around the calibration sphere using a default radius of one millimeter.

The optional parameter p6 is a string (maximum length 30; any printable character except '@' allowed) delimited by '@' characters which is to be saved as a descriptor for the calibrated tip. Trailing blanks will be ignored. If p6 is greater than 30 characters than it will be truncated by deletion of the rightmost characters. Note that this descriptor will be stored only if the host software is Controls II compatible.

Results

FOR AUTOMATIC CALIBRATION:

AM commands should have located the probe near the sphere before executing the **AT** command. P3 invokes automatic calibration. The drive vector defines the direction for the probe to move from its last position towards the sphere to make its first touch. Sphere measurement then proceeds as in semi-automatic operation.

The results of the measurements determine the relative offsets between this tip and Tip 0. The system stores this offset and uses it whenever the tip performs a measurement. The calibrated tip becomes the selected tip.

As a result of the calibration, the following information is output to the inspection report: tip name; probe changer stations (if applicable); PH9/PH10 angles (if applicable); tip diameter; tip offsets; form of measured sphere. After the command is complete the measured sphere form is stored in variable S7.

This output is done internally using **ID** commands. If the logging mode is ON, the results are also directed to the inspection log. The results are stored as an **ID** record and cannot be used for statistics.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-080 The system does not have the required probe.
- MLB-090 DCC is not available.
- MLB-091 The probe tip does not make contact with the required object.
- MLB-110 Tip 0 has not been calibrated.
- MLB-111 The calibration fixture is not a sphere.

C6 - Calibrate PH6

The C6 command prompts the operator to measure the position of the autochange rack relative to the PH6A or PH9/PH10 probe heads.

Command Format

MLString "C6"

Examples

1. MLString "C6"

Input

None.

Results

The user is instructed to take 4 points with the sphere of the autochange rack. Those four points lie on the rim of the PH6A, PH9A, or PH-10 probe adapter. The user is then instructed to take a point on the base of the adapter with the probe changer sphere. Any flat area on the bottom of the probe adapter is suitable for this touch.

Note: A **C6** command need not be executed each time power is cycled on the MP. Locate the MEA zero switches with the **ZZ** command. Then, execute a **C6** command. The vector from the probe changer sphere to the MEA zero switches is stored in nonvolatile memory in the MP once the **ZZ** and **C6** are executed. When the **ZZ** command is executed again after cycling power, the information stored in memory is used to define the location of the rack. Thus, it is not necessary to perform the **C6** manual measurements again.

Skip Action

Error; this command can not be used in a skip block.

Error Messages

- MLB-080 Improper probe installed.
- MLB-150 The APC rack is not connected.
- MLB-151 APC firmware or interface are not present.

CA - Calibrate Using Best-Fit Data

Note: Use the **SQ** command before recording any points to use with the **CA** command.

This command computes probe tip calibration data from points in a summation array. Use a sphere as the calibration fixture.

When calibrating Tip 0, the results of this command depend on the state of the system. If Tip 0 has measured the calibration sphere using one of the functions in the Probe Calibration Group since the system was turned on, use of this command indicates a new probe or a new probe position for Tip 0. Measuring the calibration sphere determines the displacement between the new and old Tip 0. Changes in the MEA Tip 0 offsets and the machine-to-part coordinate transformations reflect the new data.

If Tip 0 has not measured the calibration sphere using one of the commands in the Probe Calibration Group since the system was turned on and if the sphere is defined as SP, SPON or SPOF, this command only determines the location of the sphere relative to Tip 0.

If Tip 0 has not measured the calibration sphere using one of the commands in the Probe Calibration Group since the system was turned on, and if the sphere is defined as FX, FXON or FXOF, this command:

- uses the location of the sphere relative to the end of the probe shaft as calibrated at the factory. Failure to have the sphere at this location will result in erroneous measurement data.
- determines the offset between Tip 0 and the end of the probe shaft. It sets the sphere position relative to Tip 0 for later use.

Note: This command assumes that the calibration fixture is in the same position on the machine as originally measured. If you move the calibration fixture, then use the **LC** command.

Command Format

MLString "CA P1"

Examples

1. MLString "CA INTERNAL"

Input

- **P1** = INTERNAL (String; optional).

P1 can only be "INTERNAL". If P1 is included, the system uses the internal summation array. If P1 is omitted, the system uses the user summation array.

In the example, INTERNAL indicates that the system uses the points summed into the internal summation array to compute the required information.

Results

The system computes and stores the X, Y, and Z offsets for the probe tip. If the **CF** is used to set the size of the calibration sphere, the system computes and stores the effective probe size. If the PH-9 is present, the A and B axis positions become the A and B indices for the specified probe tip.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-040 The summation array does not contain at least four points.
- MLB-041 The points do not define a sphere.
- MLB-042 The summation array is invalid.
- MLB-043 A segmented measurement is in progress.

CE - Calibrate Extension

The CE command calibrates a probe extension.

Command Format

MLString "CE"

Examples

1. MLString "CE"

Input

None.

Results

The part program prompts the user to take 4 points on the calibration sphere of the probe changer with the rim of the extension. The part program then prompts the user to take a point on the same sphere with the base of the extension. The MP stores the extension length in the configuration data.

Skip Action

Error; this command can not be used in a skip block.

Error Messages

- MLB-150 The APC rack is not connected.
- MLB-151 APC firmware or interface are not present.
- MLB-152 The APC rack is not calibrated.
- MLB-157 No extension is loaded.
- MLB-158 Too many extensions are loaded.

CF - Calibration Fixture

The CF command defines the type, location, and size of a calibration fixture. The default value of the fixture is a sphere. If a calibration sphere and its diameter is specified, the system calculates a probe's effective diameter whenever a probe tip is calibrated. Using this command does not invalidate previously-calibrated probe tips but does require that the calibration fixture be measured by tip #0 before any additional tips can be calibrated.

Command Format

MLString "CF P1,P2,P3,P4"

Examples

1. MLString "CF SP,19.0501,P7,V0,1,0"

   The calibration fixture is a sphere of diameter 19.0501 supported by a shaft pointing in the positive Y direction. It is to be measured using seven points during probe calibration unless otherwise specified.

2. MLString "CF"

   Return information about the current calibration fixture status.

Input

- **P1** = Sphere (String; required).
- **P2** = Sphere's diameter (Numeric; optional).
- **P3** = Number of points (Numeric; optional).

- **P4** = Vector X, Y, and Z (String; optional)

If no parameters are specified then this command is a request for current status information (see OUTPUT section below). Otherwise the following rules govern the use and/or meaning of the various parameters.

P1 specifies the type and location of a calibration fixture. P1 must be one of the following:

For machines without rotary tables:

- CN Using a cone.
- SP Using a sphere.
- For MEA machines without rotary tables :
- FX Using the fixed sphere. (MEA sphere at factory calibrated position.)

Machines with rotary tables:

- CN Using a cone on the rotary table.
- CNON Using a cone on the rotary table.
- CNOF Using a cone not on the rotary table.
- SP Using a sphere on the rotary table.
- SPON Using a sphere on the rotary table.
- SPOF Using a sphere not on the rotary table.

MEA machines with rotary tables:

- FX Using the fixed sphere on the rotary table.
- FXON Using the fixed sphere on the rotary table.
- FXOF Using a fixed sphere not on the rotary table.
- SP is the default at initialization. SP denotes that the calibration fixture is a sphere on the rotary table. The sphere specified in the example is on the rotary table.

CN, CNOF, and CNON apply only to hard probes with conical fixtures. Do not use further arguments with CN, CNON, or CNOF.

FX, FXON, and FXOF apply only to machines with Microprocessor Enhanced Accuracy (MEA). These arguments specify that the calibration fixture is a sphere, mounted at a fixed position. The sphere's location was defined and calibrated in the MEA data at the factory.

The optional numeric parameter p2 is the calibration sphere's diameter in current units. If p2 is not specified then the system is unable to compute the probe size during calibration. The size of the calibration sphere must be specified to at least the degree of accuracy required for determining probe tip size. The specified sphere diameter must be at least 2 mm (0.0787 inches) but no more than 50.8 mm (2.0 inches).

The optional parameter p3 is the letter 'P' followed by the number of points to be measured on the calibration sphere. This must be at least four. The default at initialization is four for manual operations and ten for DCC operations.

The optional parameter p4 is the letter 'V' followed by three numbers that define the X, Y, Z components of a vector pointing towards the sphere along the shaft supporting the sphere.

Results

If parameters are specified then the system stores the calibration fixture information for later use during probe calibration. NOTE: After execution of this command with parameters the calibration fixture is assumed to have been changed and must be measured by tip #0 before any additional tips may be calibrated.

If no parameters are specified then current status information is returned in variables S7, S8, and S10 as follows:

- S7 = 1 if the calibration fixture has been measured with tip #0 and additional tips may now be calibrated; otherwise 0.
- S8 = sphere diameter in current units if previously specified by CF command; 0.0 otherwise.
- S10 = number of points for calibration fixture measurement specified by CF command; a value of 0 indicates that system default values (4 for manual, 10 for DCC) are being used.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

CH - Manual Calibrate Tip 0

The CH command commands the manual calibration of Tip 0. Any probe tip calibrated using **CH** becomes Tip 0. When calibrating multi-tip probes, Tip 0 establishes the probe tip that determines the relative position of the other tips.

Use this command only for the first calibration of tip 0 or if the calibration fixture is in its original position on the machine. If the calibration fixture is moved, use the **LC** command to determine the sphere's location on the machine.

Use the **CH** command before using the **CT** or **AT** commands to calibrate additional tips. Use the **CH** command to recalibrate Tip 0 when the tip size or the attitude of Tip 0 changes. If the **CH** command is used to change Tip 0, other tips calibrated in relation to the old tip 0 must also be recalibrated. If other tips remain calibrated to the old Tip 0, invalid measurement can result.

With all machines if:

- The calibration fixture has not been measured using the **CH** or **AH** commands since the system was turned on or since a **CF** command was executed, and
- the sphere is defined as SP, SPON, SPOF, CN, CNON, or CNOF,

then the command:

- **CH** determines the location of the calibration fixture relative to Tip 0.

With MEA machines, if:

- the sphere is defined as FX, FXON or FXOF,

then:

- **CH** determines the offset between Tip 0 and the end of the probe shaft,
- **CH** sets the sphere position relative to Tip 0 for later use.

For a calibration sphere with a defined diameter, the **CH** command computes the effective diameter of the probe tip. Use a sphere to calibrate and recalibrate hard probes when probe diameters change during measurement. Any change in probe diameters invalidates the use of conical seats.

Command Format

MLString "CH P1"

Examples

1. MLString "CH 6"

2. MLString "CH @TIP2 with 20mm X 4mm Stylus@"

Inputs

- **P1** = Number of points (Numeric; optional).
- **P2** =@Maximum of 30 characters@ (String; optional)

P1 specifies the number of points to record on the spherical fixture. In the example, the system records 6 points.

If the fixture is a sphere and P1 is omitted, the system uses the value defined by the **CF** command. Do not specify P1 for conical seats.

P2 is a string of any printable characters except @, that are a description of Tip 0. This probe description is saved with the probe data. Example 2 has the string "TP2 with 20mm X 4mm Stylus" saved along with the probe data.

Results

If the calibration fixture is a sphere, the part program tells the user to measure the sphere:

CALIBRATION SPHERE, MEAS n PT, TIP 0

The n specifies the number of points to record. Each time a point is recorded, the number displayed in the prompt message decreases by one. Points taken as far apart on the sphere as possible, and not in the same plane, produce the most accurate results.

If the calibration fixture is a cone, the part program tells you to measure the cone:

CALIBRATION CONE, MEAS 1 PT, TIP 0

The system records only one point while the probe is seated in the conical fixture.

As a result of the calibration, the following information is output to the inspection report:

- tip number
- probe changer stations (if applicable)
- PH-9 angles (if applicable)
- tip diameter
- tip offsets
- form of measured sphere

This output is done internally using **ID** commands. If the Logging mode is ON, the results are also directed to the inspection log.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-115 System unable to store tip data. Indicates host file system failure.

CT - Calibrate Additional Tips

Note: Use the **CH** (or **AH**) command before you use this command.

This command calibrates multiple probe tips and identifies tip positions by measuring a calibration fixture. The **CT** command establishes the relationship of the selected probe tip to Tip 0. The calibrated tip is automatically selected feature measurement. See the **GT** command for further information on tip selection. Note: The **CT** command is the only way to calibrate a multiple tip hard probe. The **AT** command can be used to calibrate a multiple tip touch probe on a DCC system.

When a calibration sphere with a defined diameter is used (see the **CF** command), the system computes and uses the effective diameter of the probe tip.

Command Format

MLString "CT P1,P2,P3"

Examples

1. MLString "CT 39,6"
2. MLString "CT 199,@TP2 with 20mm X 4mm Stylus@"
3. MLString "CT Tip_Num_X5,@TP2 with 20mm X 4mm Stylus@"

Input

- **P1** = Tip I.D. (String (maximum of 10 characters); required).
- **P2** = Number of points (Numeric; optional).
- **P3** = @Maximum of 30 characters@ (String; optional)

P1 specifies the identification name of the probe tip being calibrated. In the first example, the probe tip being calibrated is tip 39. A maximum of 10 characters may be used for the tip name, as shown in example 3 (tip equals "Tip_Num_X5"). Note: It is easier to use probe tips later in the program if they are named consistently when calibrated.

P2 specifies the number of points to record. In the example, the system records 6 points. If P2 is omitted, the system uses the value specified in the **CF** command or the default value of 4 points. Do not specify P2 for conical seats.

P3 is a string of any printable characters except @, that allows the entry of a description of the tip. This description is stored along with the probe data. Examples 2 and 3 show the tip description "TP2 with 20mm X 4mm Stylus".

Results

If the calibration fixture is a sphere, the user is prompted to measure the sphere:

CALIBRATION SPHERE, MEAS n PT, TIP #

The n specifies the number of points remaining to be recorded. Each time a point is recorded, the number displayed in the prompt message decreases by one. Points taken as far apart on the sphere as possible, and not in the same plane, produce the most accurate results.

If the calibration fixture is a cone, the user is prompted:

CALIBRATION CONE, MEAS 1 PT, TIP #

The system records only one point while the probe is seated in the conical fixture.

As a result of the calibration, the following information is output to the inspection report:

- tip number

- probe changer stations (if applicable)
- probe changer stations (if applicable)
- PH9 angles (if applicable)
- tip diameter
- tip offsets
- form of measured sphere
- This output is done internally using **ID** commands. If the Logging mode is ON, the results are also directed to the inspection log.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-110 Tip 0 has not been calibrated.
- MLB-115 System unable to store tip data. Indicates host file system failure.

D9 - Define PH9 Safe Position

The D9 command prompts the user to move the PH9 or PH10 to a position where the PH9 or PH10 can safely change probe tip attitudes before loading or unloading a station from either the probe changer or one of two stylus changers. The system stores the coordinates of the indicated position. The coordinates are preserved when power is turned off. Use the D9 command if the rack is moved or if a different rotation position must be used to avoid a collision with a part.

Command Format

MLString "D9 P1"

Examples

1. MLString "D9"
2. MLString "D9 2"

Input

- **P1** = 0 or 1 or 2; (Numeric, optional)

P1 can only be 0, 1, or 2. Not specifying P1 is equivalent to setting it to zero. (Note that MP firmware versions prior to 15.2 do not allow P1). A value of 0 indicates that a safe position is being defined for the automatic probe changer (APC). A value of 1 or 2 indicates that a safe position is being defined for stylus changer #1 or stylus changer #2 respectively. Note that the system maintains a different safe position for each of these three devices.

Results

The user is instructed to position the PH9A or PH10 to a safe position and press the Record button on the RCU. After this, the PH9A or PH10 will move to this point before every load or unload station command, including tip selections which require station load / unload.

Skip Action

Error; this command can not be used in a skip block.

Error Messages

- MLB-150 APC rack is not connected.
- MLB-151 APC firmware or interface are not present.
- MLB-152 APC rack is not calibrated.

LA - DCC Locate Calibration Sphere

The LA command directs the automatic or semi-automatic measurement of a calibration sphere to locate the sphere on the worktable. Use this command to move the calibration sphere after beginning probe calibration.

Measurement of the calibration fixture using DCC operations includes special instructions to the user to begin semi-automatic measurement. The semi-automatic measurement sequence instructs the user to record two points defining the drive vector. Instructions then tell the user to switch to AUTO mode so the system can complete the calibration.

The automatic calibration sequence assumes that the probe has been positioned (either manually or under DCC control) at the correct position near the sphere. The **LA** command then contains a drive direction vector.

Use this command with DCC systems only. This command requires the use of an installed touch probe and a calibration sphere.

Command Format

MLString "LA P1,P2,P3,P4"

Examples

1. MLString "LA 5,L17,D12,2,-5,R2.5"

   Locate the calibration sphere by measuring 5 points on the calibration sphere using tip #0 under DCC control. The probe has an approximate tip radius of 2.5 units and a shaft length of 17 units. Calibration is completely under DCC control beginning with a seek along the part coordinate vector X12, Y2, Z-5 from the current position towards the center of the calibration sphere.

2. MLString "LA 6,R1.775"

   Locate the calibration sphere using tip #0 by measuring 6 points on the calibration sphere under DCC control. The probe has an approximate tip radius of 1.775 units and a shaft length at least one half of the sphere radius. Calibration begins by prompting the operator to record two probe positions, one 150 mm (6 inches) and the other 6 mm (0.25 inches) away from the calibration sphere. Thereafter probe calibration is completely under DCC control.

Input

If more than one of these arguments is used, use them in numerical order.

- **P1** = Number of points (Numeric; optional).
- **P2** = L(ength) and probe shaft length (Numeric; optional).

- **P3** = D(irection) or M followed by X, Y, Z (Numeric; optional).
- **P4** = R(adius) and probe tip radius (Numeric; optional).

The optional numeric parameter p1 is the number of points to be measured on the calibration sphere. If p1 is omitted the default number of points (see CF command) is used.

The optional parameter p2 consists of the letter 'L' followed by the length of the probe shaft. The length of the probe shaft is the distance from the center of the probe tip to the first bend or obstruction along the probe shaft. This distance must be specified if this distance is less than one-half of the calibration sphere radius.

The optional parameter p3 consists of either the letter 'D' or the letter 'M' followed by a signed direction vector. The 'D' indicates that the vector is in the currently active part reference frame (e.g., D0,0,-1) while the 'M' indicates that the vector is in the machine reference frame (e.g., M0,1,0). If present p3 invokes automatic probe calibration and causes the first touch to be made by moving from the current probe position towards the calibration sphere using the specified vector. If p3 is not present then the operator is prompted to define this drive vector.

The optional parameter p4 consists of the letter 'R' and defines the approximate radius of the probe tip in current measurement units. If omitted, the system moves around the calibration sphere using a default radius of one millimeter.

Results

FOR AUTOMATIC CALIBRATION:

For automatic calibration (P3 given), the probe moves from its last position parallel to the given vector to make the first touch on top of the sphere. For semi-automatic calibration the user is prompted to record two positions in space describing a line pointing towards the center of the sphere; the system then prompts for AUTO and drives along this line to make the first touch on top of the sphere. Sphere measurement then proceeds the same in both modes. The results of these measurements determine the location of the calibration sphere relative to probe Tip 0.

Later tip calibrations use this information to determine the relative offsets between Tip 0 and other probe tips. As a result of the calibration, the following information is output to the inspection report:

- tip number
- probe changer stations (if applicable)
- tip diameter
- tip offsets
- form of measured sphere (also stored in variable S7).

This output is done internally using **ID** commands. If the Logging mode is ON, the results are also directed to the inspection log.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-080 The system does not have the required probe.
- MLB-090 DCC is not available.
- MLB-091 The probe tip does not make contact with the required object.
- MLB-111 The calibration fixture is not a sphere.

LC - Locate Calibration Fixture

The LC command prompts the operator to take manual measurements to locate the calibration fixture on the work table. This location is relative to Tip 0. Use this command when it is necessary to move the calibration fixture after beginning probe calibration.

Command Format

MLString "LC P1"

Examples

1. MLString "LC 6"

Input

- **P1** = Number of points (Numeric; optional).

Use P1 only if the calibration fixture is a sphere. It specifies the number of points to record on the sphere. In the example, the system records 6 points.

If the calibration fixture is a sphere and if P1 is omitted, the part program uses the value previously set by the **CF** command. If the **CF** command did not specify the number of points to record, the system defaults to 4 points.

Results

The part program prompts the user to measure the calibration fixture manually with Tip 0. This measurement determines the location of the calibration fixture on the machine relative to Tip 0. Later tip calibrations use this information to determine the relative offsets between Tip 0 and other probe tips. As a result of the calibration, the following information is output to the inspection report:

- tip number
- probe changer stations (if applicable)
- tip diameter
- tip offsets
- form of measured sphere (also stored in variable S7).

This output is done internally using **ID** commands. If the Logging mode is ON, the results are also directed to the inspection log.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

None.

M9 - Move PH9

The M9 command causes the PH9/PH10 probe systems to rotate to a new position.

Command Format

MLString "M9 P1,P2"

Examples

1. MLString "M9 A90, BI7.5"
2. MLString "M9 B-120"

Input

- **P1** = A, optional I, and angle (Numeric; optional).
- **P2** = B, optional I, and angle (Numeric; optional).

P1 and P2 specify the axis position for the probe. The format of the argument consists of an axis letter (A or B), an optional I (for incremental positioning) and the position or target angle in degrees. Note that the A-axis position can range from 0 to 105 degrees while the B-axis can range from -180 to +180 degrees.

If no arguments are specified then the current PH9/PH10 axis positions are returned to system variables.

In the first example, the A-axis of the probe moves to the +90 degree mark. The B-axis moves, using incremental positioning, 7.5 degrees in the positive direction.

In the second example, the B-axis of the probe moves to the -120 degree mark.

Results

The target angle is rounded to the nearest 7.5 degree increment. If any DCC moves are in the DCC queue, the CMM completes all the moves before moving the probe. If the system is in manual mode but moves are in the DCC queue, the following prompt appears: AUTO MODE REQUIRED. Once the DCC queue is empty the probe head is rotated to the new position.

If no arguments are specified then the current PH9/PH10 axis positions are returned to variables S7 (the A-axis) and S8 (the B-axis).

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-080 The system does not have a PH9 or PH10.
- MLB-100 The PH9 is not seated properly.
- MLB-101 The probe head is obstructed.

RI - Read Probe Data

The RI command retrieves a probe's offsets and radius. If the system has a PH9, the system returns A and B indexes for the tip. If the system has a probe changer, the system also returns station numbers associated with the tip.

Command Format

MLString "RI P1"

Examples

1. MLString "RI 15"
2. MLString "RI Tip_X5"

Input

- **P1** = Tip I.D. (String (maximum of ten characters); required).

P1 specifies the tip for which data is to be retrieved. The tip I.D. is assigned when the probe tip is calibrated. In the first example, data is retrieved for the tip named 15. The returned probe offsets are in current measurement units relative to the machine coordinate reference frame. The system uses the machine scale factors, input through the **SM** command, in calculating the coordinates returned through the **RI** command.

Results

The X offset returns to S7. The Y offset returns to S8. The Z offset returns to S10. The probe tip radius returns to S11.

The A index on the PH9 returns to S12. The B index on the PH9 returns to S13. If the system does not have a PH9 installed, S12 and S13 contain a 0. The probe changer station numbers return to S14, S15, S16 and S17. A 0 indicates that no station is specified for that position. If the probe changer is not present on the system, S14 through S17 are 0. The probe tip name is returned in variable S0D.

Skip Action

Executes even if in a skipped block.

Error Messages

- MLB-026 The specified tip is invalid.

SI - Station Information

The SI command tells if a station is loaded or unloaded. A station is loaded if the probe or probe extension from that station is currently mounted on the CMM probe shaft. The SI command also tells whether or not a station is calibrated.

Command Format

MLString "SI P1"

Examples

1. MLString "SI 4"

Input

- **P1** = Station number (Numeric; required).

P1 is the station number. The station number in the example is 4. Station numbers range from 1 to 8.

Results

The status returns to S7 as follows:

- 0 Calibrated and not loaded
- 1 Calibrated and loaded
- -1 Uncalibrated and unloaded

The user-defined label assigned by the **GS** command returns to variable S0D.

Skip Action

Executed even if part of a skipped block.

Error Messages

None.

SQ - Calibration Setup

The SQ command directs the MP to begin collecting measurement data for the purposes of calibrating a probe tip. The **CA** command completes the probe calibration operation begun by the **SQ** command.

Command Format

MLString "SQ P1,P2"

Examples

1. MLString "SQ 0"
2. MLString "SQ 199,@TP2 with 20mm X 4mm Stylus@"
3. MLString "SQ Tip_X5,@TP2 with 20mm X 4mm Stylus@"

Input

- **P1** = Tip I.D. (String (maximum of ten characters); required).
- **P2** = @Maximum of 30 characters@ (String; optional).

P1 specifies the name of the tip to calibrate. Tip assignment names are a string with a maximum length of 10 characters and can include only letters, digits, and the underline character. In the third example, tip Tip_X5 is calibrated.

P2 is a description for the calibrated tip.

Results

The MP saves the tip name and description. Internal flags are set indicating that measured points are for probe calibration until the **CA** command is executed. Note that this results in special MEA correction and only probe calibration data should be recorded until the sequence is complete.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-110 Occurs if P1 is not 0 and tip 0 has not been calibrated.

TI - Tip Status

The TI command checks whether the probes in a specified list are calibrated.

Command Format

MLString "TI P1,P1,...P1"

Examples

1. MLString "TI 13"
2. MLString "TI 1,3,4,5,Tip_X5"

Input

- **P1** = Tip I.D. (String (maximum ten characters); required).

P1 is the identifier of a tip to examine. Tip identifiers are assigned by tip calibration operations (see the **AT** and **CT** commands). In the second example:

- 1 is the first tip to examine.
- 3 is the second tip to examine.
- 4 is the third tip to examine.
- 5 is the fourth tip to examine.
- "Tip_X5" is the fifth tip to examine.

Results

The status of the probe tip calibration returns to variables S7 and S8 as follows:

- S7 = 1 means the specified tips are calibrated.
- S7 = 0 means at least one of the specified tips is not calibrated.
- S8 = 1 means that any automatic probe changer stations required for the specified tips are loaded.
- S8 = 0 means that the automatic probe changer stations required for at least one of the specified tips are not loaded.

Skip Action

Executes even if in a skipped block.

Error Messages

- MLB-026 One or more of the specified tips is invalid.

4. Tip Selection Group

MLB commands in the Tip Selection Group support feature measurement with probes that have more than one tip or more than one tip position. These probes include the 5-tip star probe and the PH9/PH10. Identify and calibrate probe tips before you select them. Selecting an uncalibrated probe tip results in an error.

The Tip Selection Group includes the following MLB commands.

FP	**Define the Active Probe Tip Data Set**
GS	**Get Station** loads the specified probe changer stations onto the probe adapter.
GT	**Select Tip** selects the probe tips.
NT	**Delete Tip** deletes the calibration vectors for all probe tips except Tip 0.
RZ	**Remove Station** deletes all calibrated tips associated with a specified station.

FP - Define the Active Probe Tip Data Set

The FP command stores and retrieves a string used to identify the active probe tip data set.

Typically, you use only one probe tip data set per part program. If a part program uses multiple data sets, then the position of tip zero must be identical in all data sets. Deviation from this rule will not maintain reference to the datum established with the tips used in the previous data set.

Command Format

MLString "FP P1"

Examples

1. MLString "FP"
2. MLString "FP @Probe tip data set #13.@"
3. MLString "FP @PRDCTN@"
4. MLString "FP @@"

Input

- **P1** = @Maximum of 81 characters@ (String; optional)

P1 is used by the system to identify the active probe tip data set.

Results

If P1 is omitted (example 1), the system returns the current probe tip data set identifier to system variable S0D. If P1 is a null string (example 4), the system uses probe data from the EEPROM in the MP. If P1 is present (examples 2 and 3), the probe tip data set identifier is equal to the P1 string.

If the FP command changes to a different tip set than the one that had been active, then tip #0 must be calibrated before any additional tips can be calibrated.

Skip Action

Executed normally.

Error Messages

None.

GS - Get Station

The GS command loads equipment from the probe changer station(s) and/or stylus changer stations onto the probe adapter. The loaded probe extensions and tip can then perform tip calibration or measurements.

Command Format

MLString "GS P1,P1,P1,P1,P2"

Examples

1. MLString "GS"
2. MLString "GS 4,5,1,STAR-POBE"
3. MLString "GS 4,5,1"
4. MLString "GS 0"

Input

- **P1** = Station number (Numeric; optional).
- **P2** = User-defined station name (String; optional).

P1 is a station number. Legal station numbers are 1 to 8 for the probe changer, 21 to 26 for stylus changer #1, and 31 to 36 for stylus changer #2. The command "GS 0" is a special form of the command used to unload equipment without loading more. A maximum of four stations may be specified but only the last station may correspond to one of the stylus changers. The order in which the stations are listed determines the order in which they are loaded. The last specified probe changer station may contain a probe tip but all previous stations must contain extensions.

If P1 is omitted, the currently selected station or combination of stations returns to variables for user verification. The first example shows the command with P1 omitted.

For example, if the commands "GS 4,5,1" and "GS" were executed, the values in variables S7, S8, S10, and S11 would be 4, 5, 1, and 0.

P2 is a 20-character, user-defined label that names the last specified probe changer station. It can be any alphanumeric label and can contain embedded spaces. In the second example, the probe loads stations 4 and 5 (which contain extensions) and station 1, and the label "Star-Probe" is given to station 1.

P2 cannot be used alone. For example, the command "GS STAR-PROBE" produces an error. P2 can only be used to label a specified station.

Results

If P1 is specified, the probe unloads the active station or combination. Then the probe loads the specified station or combination. That probe combination becomes active. Note: It is the responsibility of the programmer to allow clearance for loading and unloading probe stations.

If P1 is omitted:

- The currently active station combination returns to temporary variables S7, S8, S10, and S11 in their loading order.
- A zero in S7 indicates no station was loaded.
- If a probe changer is not present on the system, S7, S8, S10, and S11 contain 0.

The command "GS 0" causes the in-use probe to be unloaded to its station and the unarmed PH9/PH10 to go to the safe position.

Skip Action

Ignored; not executed if part of a block of code being skipped.

Error Messages

- MLB-150 The APC rack is not connected.
- MLB-151 APC firmware or interface is not present.
- MLB-152 The APC and or SCR rack is not calibrated.
- MLB-153 An APC/MP communication error has occurred.
- MLB-154 APC rack timeout or lock failure.
- MLB-155 APC rack overtravel.
- MLB-156 APC and/or SCR station or safe position is not calibrated.

GT - Select Tip

The GT command selects a calibrated probe tip or tips for measurement. When the system evaluates the **GT** command, the first tip in the selection list immediately becomes active. The command can specify one tip or several tips for multiple measurements.

Command Format

MLString "GT P1,P1,...,P1" or MLString "GT n*P1,n*P1,...,P1"

Examples

1. MLString "GT 4"
2. MLString "GT 3*Tip1,TP7_2,10*Tip_X5,Tip199"

Input

- **P1** = Tip I.D. (String (maximum ten characters); optional).

P1 is any calibrated tip identifier. Most typical use is to specify a single tip (as in example 1) which is used for all measurements until another **GT** command is executed or anther tip is calibrated (tip calibration results in the calibrated tip being selected as the active tip).

However, as shown above, the **GT** command may be used to specify a series of tips to be used in sequence during an automatic circle measurement (see the AC command) and a count may be specified with each tip to indicate how many touches are to be made with that tip before selecting the next tip. This form of the command is especially useful when using a star probe to measure a circle automatically.

If P1 is omitted, the **GT** command returns the identifier of the currently selected tip.

Damage can result if the **GT** command occurs in a DCC part program and the selected tips are no longer in the positions expected by the part program.

When more than one tip is specified, separate the tip identifiers by commas. For consecutive multiple measurements that use the same tip, the number of measurements with that tip and an asterisk (*) can precede the tip identifier.

The second example, MLString "GT 3*Tip1,TP7_2,10*Tip_X5,Tip199", selects probe tip "Tip1" for three single point measurements, then probe tip "TP7_2" for one point measurement, then probe tip "Tip_X5" for ten point measurements, and finally "Tip199" as the active probe tip for subsequent measurements.

The last tip used from a list of several tips remains active for subsequent measurements. Note: All measurement commands except for AC use the currently selected tip. They do not change the selected tip even though a list of tips occurs in the **GT** command.

Results

If a probe changer is in use, and the selected tip requires a different station, stations are automatically changed. There must be a clear path from the previous probe position to the probe changer.

If a PH9 is used and a different A and B position is required, the PH9 automatically changes A and B position. There must be enough room to rotate.

During operations, measurement prompts to the user display the identifier of the currently selected probe tip. Tips are automatically selected during an AC command in the P1 order until the list is exhausted or until another **GT** command occurs in the part program.

Each time the program executes an automatic circle (AC) measurement, it uses the next set of tips from the P1 list.

The last tip used for an automatic circle (AC) measurement remains selected until another multi-point measurement occurs.

The last selected tip in the list remains selected for subsequent measurements. If a **GT** command occurs with a new P1 parameter before exhausting the previous list, the old list is discarded. The new one is immediately in use.

If P1 is omitted, the **GT** command returns the selected tip identifier to S0D.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-026 One of the specified tips is invalid.

NT - Delete Tip

The NT command deletes the calibration data for previously calibrated probe tips.

Command Format

MLString "NT P1"

Examples

1. MLString "NT"
2. MLString "NT 13"
3. MLString "NT TP7_2"

Input

- **P1** = Tip I.D. (String (maximum ten characters); optional)

If P1 isn't used, the data for all calibrated tips is deleted. If P1 is used, the data for tip P1 is deleted. Tip 0 cannot be deleted unless all tips are being deleted.

The first example deletes data for all calibrated tips, the second example deletes data for tip 13, and the third example deletes data for tip TP7_2.

Results

The data for the specified tip(s) is deleted. If all tips are deleted, tip 0 is returned to its default state of zero offsets and a probe diameter of 2 mm. Deletion of other tips means that those tips can no longer be selected for measurement purposes.

Note: If a probe rack is present and the deleted probe tip active (in the PH9/PH10), the **NT** command results in the active probe tip being returned to its station and the unarmed PH9/PH10 moving to the safe position.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-026 Invalid tip identifier specified. This error occurs if an attempt is made to delete tip 0.

RZ - Remove Station

The RZ command deletes all the calibrated tips associated with the specified station.

Command Format

MLString "RZ P1"

Examples

1. MLString "RZ 5"

Input

- **P1** = Station number (Numeric; required).

P1 is the station number. Station numbers range from 1 to 8. In the example, the station number is 5.

Results

The system deletes all calibrated tips that involve the specified station.

Skip Action

Error; this command can not be used in a skipped block.

Error Messages

None.

5. Feature Stack Group

MLB commands in the Feature Stack Group let you move data within the Feature Register and the Feature Stack. When the system measures or computes a feature, it stores the resulting data in the Feature Register. This data includes:

- feature position
- orientation data relative to the part reference frame
- feature size of circles, spheres, cylinders, ellipses, and cones
- feature number
- feature type

When new data enters the Feature Register, the former contents of the Feature Register move into the Feature Stack. At the same time, the data in each Feature Stack position moves down to the next position.

The Feature Stack holds up to 8 features. If the Feature Stack if full when the system records new feature data, the data in the last position drops from the Feature Stack and the data is lost.

The Feature Stack Group includes the following MLB commands.

PO	**Pop Stack** moves the features in the Feature Stack up one position. It puts the data from the first position of the Feature Stack into the Feature Register.
PU	**Push Stack** moves the features in the Feature stack down one position.
US	**Unstack** moves the features in the Feature Stack up one position. It does not affect the Feature Register.
XC	**Exchange Feature** exchanges the features in the Feature Register with the one in the first position of the Feature Stack.

PO - Pop Stack

The PO command moves the features in the Feature Stack up one position. The **PO** command puts the data from the first position of the Feature Stack into the Feature Register. All features that remain in the Feature Stack move up one position.

Command Format

MLString "PO P1"

Examples

1. MLString "PO D"

Input

- **P1** = D (String; optional).

P1 can only be the letter 'D'. If P1 is specified then the feature number to be assigned to the next feature is decremented. This is used when it is necessary to temporarily create a feature that is immediately discarded.

Results

The system deletes the contents of the Feature Register and pops the remaining contents of the Feature Stack up one position. If P1 is specified, the next feature number to be assigned is decremented. The Dimensional Array stores the part coordinates for the new contents of the Feature Register. It also stores the feature type and the feature number.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

PU - Push Stack

The PU command moves all of the features in the Feature Stack down one position, duplicating the feature data from the Feature Register into F1.

Command Format

MLString "PU"

Examples

1. MLString "PU"

Input

None.

Results

The feature at the bottom of the Feature Stack drops out of the Feature Stack and is discarded. The system duplicates the data in the Feature Register into the first position of the Feature Stack after all feature data in the Feature Stack has been pushed down one position. A new feature number is assigned to the data in the Feature Register.

Skip Action

Executes even when inside a block of code being skipped.

Error Messages

- MLB-030 The Feature Register does not contain any data.

US - Unstack

The US command moves all of the features in the Feature Stack up one position, discarding the previous data from stack position number one.

Command Format

MLString "US"

Examples

1. MLString "US"

Input

None.

Results

The system discards the feature in the first Feature Stack position. The feature in the Feature Register does not change.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

XC - Exchange Feature

The XC command exchanges the feature in the first position of the Feature Stack with the feature in the Feature Register.

Command Format

MLString "XC"

Examples

1. MLString "XC"

Input

None.

Results

The data for the feature in the Feature Register (FR) is swapped with the data for the feature in the first position of the Feature Stack (F1). The Dimensional Array stores the part coordinates for the new contents of the Feature Register. It also stores the feature type and the feature number.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-030 The Feature Stack first position or the Feature Register does not contain a feature.

6. Feature Storage Group

MLB commands in the Feature Storage Group let you store and retrieve features by name. You can store features for future use. For example, you can use stored features to construct part reference frames or compute feature relationships. Feature Storage holds an unlimited number of features. Users assign the feature names. The names consist of up to 64 alphanumeric characters. Do not use punctuation or spaces in feature names, however you may use underscore characters. If you use lowercase letters in a name, the system converts them to uppercase.

The Feature Storage Group includes the following MLB commands.

DF	**Delete Feature** deletes a feature from Feature Storage and frees that storage location.
RF	**Recall Feature** moves the Feature Stack down and puts the data from the specified feature storage area into the Feature Register.
SF	**Save Feature** copies data from the Feature Register into Feature Storage.

DF - Delete Feature

The DF command deletes feature(s) from Feature Storage and frees the storage location(s).

Command Format

MLString "DF P1"

Examples

1. MLString "DF Line5"

Input

- **P1** = string (Optional).

P1 specifies a stored feature's name. The feature was assigned when saved with the **SF** command. In the example, Line5 is the name of the Feature Storage location to delete.

If P1 is omitted the system deletes all features in Feature Storage.

Results

The system deletes the specified feature(s) from Feature Storage.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-072 The feature is not found.

RF - Recall Feature

The RF command moves the Feature Stack down and puts the data from the specified feature storage area into the Feature Register.

Command Format

MLString "RF P1,P2,P3"

Examples

1. MLString "RF Line5"
2. MLString "RF Line5,RefLine"
3. MLString "RF Line5,RefLine,M"

Input

- **P1** = String (Required).
- **P2** = String or Number (Optional).
- **P3** = M or I (Optional: only allowed if P2 is present)

P1 specifies a stored feature's name as set by the **SF** command. In the example, Line5 is the stored data name.

The optional parameter P2 is the identifier of a previously saved reference frame. If P2 is specified then the data for the recalled feature is to be expressed in the specified reference frame when moved into the Dimensional Array.

The optional parameter P3 is either the letter I or the letter M. This parameter can only be used if P2 has been specified and indicates whether the feature data in the Dimensional Array is to be expressed in inches (I) or millimeters (M). If P3 is not specified then the data for the recalled feature is expressed in currently active units.

Results

Data in the Feature Register and the Feature Stack moves down one position. The Feature Register accepts the recalled feature. Feature Storage does not change. The Dimensional Array stores the feature's position, orientation, and size if applicable. The Dimensional Array also stores the feature type and the feature number.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-072 The feature is not found.

SF - Save Feature

The SF command copies data from the Feature Register into Feature Storage and assigns a name to that feature.

Command Format

MLString "SF P1"

Examples

1. MLString "SF Line5"

Input

- **P1** = string (Required).

P1 specifies the stored feature's name. Feature names can be no more than 64 characters long and can consist only of letters (A, ... , Z), digits (0, ..., 9), underline characters, hyphens, parentheses, and square brackets. The system has predefined names for the Feature Register (FR), the Feature Stack (F1, ..., F8), and parts of the reference frame (XA, YA, ZA, XY, YZ, ZX, and DP). Do not give a feature one of these names. In the example, the Line5 is the name assigned to the saved feature.

Results

The data for the feature in the Feature Register is added to the saved feature storage under the specified name. The data in the Feature Register does not change.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-030 The Feature Register does not contain a valid feature.

7. Feature Measurement Group

MLB commands in the Feature Measurement Group measure features manually.

The Feature Measurement Group includes the following MLB commands.

M2	**Manual Measure 2D Line** constructs a line by recording points and computing a best-fit line. It projects the line onto a plane parallel to the work plane.
M3	**Manual Measure 3D Line** constructs a best-fit line defined by a set of recorded points.
MB	**Manual Measure Slot/Web** lets you measure a slot or web.
MC	**Manual Measure Circle** constructs a circle by recording points and computing a best-fit to a circle.
MD	**Manual Measure Ellipse** constructs an ellipse by recording points and computing the best-fit of an ellipse to those points.
MF	**Manual Measure Offset Plane** lets you measure an offset plane.
MI	**Manual Measure Cylinder** constructs a cylinder by recording points and computing the best-fit of a cylinder to those points.
MJ	**Manual Measure Step Cylinder** lets you measure a step cylinder.
MP	**Manual Measure Plane** computes the best-fit of a plane to the measured points.
MQ	**Manual Measure Cone** constructs a cone by recording points and computing the best-fit of a cone to those points.
MS	**Manual Measure Sphere** constructs a sphere by recording points and computing the best-fit of a sphere to those points.
QZ	**Manual Single Point Measurement** lets you select prompting at the RCU or the host computer.

M2 – Manual Measure 2D Line

The M2 command prompts the operator to measure at least two points and then constructs a best-fit line based on the set of recorded points as projected into a plane parallel to the work plane.

Command Format

MLString "M2 P1,P2,P3,P4"

Examples

1. MLString "M2"
2. MLString "M2 INT,10"
3. MLString "M2 @Measure a line on the right side@,+X,5,0,1,0"
4. MLString "M2 .7,0,.7"

Input

- **P1** = @operator prompt@ (String; optional)
- **P2** = X, -X, Y, -Y, Z, -Z, OFF, EXT, or INT (String; optional).
- **P3** = Number of points (Numeric; optional).
- **P4** = Set of thee numbers (Numeric; optional).

P1 is used to specify a message to be shown in the manual measure line window that is displayed when this command is executed. This prompt should help the operator identify the correct feature to be measured.

P2 specifies the desired probe compensation. Omitting P2 means no probe compensation occurs. The first example shows measurement of two points describing a line parallel to the active working plane. A specified axis indicates the part axis that most nearly points in a direction away from the measured surface. In the third example, the +X is the desired probe compensation. The example measures five points describing a line parallel to the active working plane approximately parallel to the Y-axis. The positive end of the X-axis points in a direction away from the measured surface.

The possible P2 arguments are :

- X,-X,+Y,-Y,+Z,-Z The specified axis points away from the measured surface.
- OFF No compensation,
- EXT The measured surface is facing away from the part datum.
- INT The measured surface is facing towards the part datum.

P3 defines the number of points to measure. At least two points are required for measurement. Omitting P3 selects the automatic default of 2 points. For segmented measurements, P3 can equal one. In the third example, the 5 defines the number of points to record.

P4 is a set of three numbers defining the approximate orientation of the line in the active part reference frame. This may define a unit vector but this is not required. At least one of the two numbers corresponding to the active plane coordinates must be non-zero.

Results

The user is prompted through the manual measure line window to measure the specified number of points. After the points are measured the system computes the data for the line. If the recorded points do not define a line then the operator is prompted to remeasure the feature.

The current contents of the Feature Register move down to Feature Stack Position #1. The Feature Register stores the line feature data. The computed line position is modified to reflect any requested probe compensation. Form (planar straightness) is computed as specified by the **FC** command (BFD: the sum of the distances from the computed line to the points furthest away on opposite sides of the line; MRS: the minimal separation between two parallel lines in the working plane which contains the set of points). The form results are stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the computed line in the S(14,0), S(15,0) and S(16,0).

The Dimensional Array also stores :

- Base point (average of the measured points) of the line,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skip block but a null feature is pushed onto the Feature Register.

Error Messages

None.

M3 - Manual Measure 3D Line

The M3 command prompts the operator to measure at least two points describing a line in space and then computes the best-fit position and attitude of a line through the points.

Command Format

MLString "M3 P1,P2,P3"

Examples

1. MLString "M3 @Measure line 17A@"
2. MLString "M3 6"
3. MLString "M3 1,0,0"
4. MLString "M3 12,.7,0,.7"

Input

- **P1** = @operator prompt@ (String; optional)
- **P2** = Number of points (Numeric; optional).
- **P3** = Set of thee numbers (Numeric; optional).

P1 is used to specify a message to be shown in the manual measure line window that is displayed when this command is executed. This prompt should help the operator identify the correct feature to be measured.

P2 defines the number of points to measure. At least two points are required for measurement. Omitting P2 selects the automatic default of 2 points. For segmented measurements, P1 can equal one. In the second example, the 6 defines the number of points to be measured.

P3 defines the approximate orientation of the line in the active part reference frame. This may define a unit vector but this is not required. At least one of the three numbers must be non-zero. Example 3 is measuring two points describing a line approximately in line with the X-axis. Example 4 is measuring 12 points describing a line whose direction vector is approximately .7,0,.7. If P3 is specified then the line is computed to point in the same general direction as the given vector.

Results

The user is prompted through the manual measure line window to measure the specified number of points. After the points are measured the system computes the data for the line.

The current contents of the Feature Register move down to Feature Stack Position #1. The Feature Register stores the line feature data. The computed line position is modified to reflect any requested probe compensation. Form (planar straightness) is computed as specified by the **FC** command (BFD: twice the maximum distance between the best-fit line and the individual points; MRS: the diameter of the smallest cylinder which contains the set of points). The form results are stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the computed line in the S(14,0), S(15,0) and S(16,0).

The Dimensional Array also stores:

- Position of the line,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

None.

MB - Manual Measure Slot / Web

The MB command prompts the operator to measure at least three points describing a slot or web in the active working plane and then computes the best-fit position, attitude, and size of a slot or web through these points. Measure the slot or web by recording two or more points on one side of the feature and one or more points on the other side. The system applies the probe compensation specified by the **CM** command (INT when measuring a slot and EXT when measuring a web).

Command Format

MLString "MB P1,P2,P3,P4"

Examples

1. MLString "MB 2,1"

Input

- **P1** = @operator prompt@ (String; optional)
- **P2** = Number of points (Numeric; optional).
- **P3** = Number of points (Numeric; optional).
- **P4** = AP (String; optional).

A **CM** command should be executed before the "MB" command. The argument specified in the **CM** command specifies the type of feature (slot or web) to be measured and the desired probe compensation:

- INT - Indicates that a slot is to be measured.

- EXT - Indicates that a web is to be measured.
- OFF - Indicates no probe compensation.

P1 is used to specify a message to be shown in the manual measure slot/web window that is displayed when this command is executed. This prompt should help the operator identify the correct feature to be measured.

P2 and P3 specify the number of points to be measured on each side of the slot or web. P2 represents the number of points to be measured on the first side and must be greater than or equal to 2. P3 represents the number of points to be measured on the second side and must be greater than or equal to 1. If either P2 or P3 is specified, both must be present. If P2 and P3 are not specified, the default is two points on the first side and one on the second.

The optional parameter P4 can only be "AP". This specifies that the feature is to be computed in the major plane (XY, YZ, or ZX) which is most parallel to the set of measured points rather than in the active working plane. If P4 is not specified then the slot/web is computed in the active working plane.

Results

The user is prompted through the manual measure slot/web window to measure the specified number of points. After the points are measured the system computes the data for the slot/web.

The current contents of the Feature Register move down to Feature Stack Position #1. The Feature Register stores the slot/web feature data. The computed size is modified to reflect any requested probe compensation. Form is computed as range of the absolute distances from the center line of the feature to the set of points. The form results are stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the computed center line in the S(14,0), S(15,0) and S(16,0).

The Dimensional Array also stores:

- Position of the center line of the feature,
- Feature size (radius and diameter),
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

None.

MC - Manual Measure Circle

The MC command prompts the operator to measure at least three points on a circle. It then computes the best-fit to a circle that is parallel to the measurement plane. Use the **CM** command to specify compensation for the probe radius.

Command Format

MLString "MC P1,P2,P3"

Examples

1. MLString "MC"
2. MLString "MC 10"
3. MLString "MC AP"

Input

- **P1** = @operator prompt@ (String; optional)
- **P2** = Number of points (Numeric; optional).
- **P3** = AP (String; optional)

P1 is used to specify a message to be shown in the manual measure circle window that is displayed when this command is executed. This prompt should help the operator identify the correct feature to be measured.

P2 defines the number of points to measure. At least three points are required for measurement. The first example measures three points on a circular feature in the active working plane. In the second example, the 10 is the number of points for the user to record. Omitting P2 selects the automatic default of three points. For segmented measurements, P2 can be 1 or 2.

P3 can only be "AP". This specifies that the circle is to be computed in the major plane (XY, YZ, or ZX) which is most parallel to the set of measured points rather than in the active working plane. The third example prompts the operator to measure three points on a circle in a major plane (XY, YZ, or ZX).

Results

The user is prompted through the manual measure circle window to measure the specified number of points. As points are measured they are accumulated in the internal best-fit summation array. After the points are measured the system computes the data for the circle. If the points do not define a circle in the working plane then the user is prompted to remeasure the feature.

The current contents of the Feature Register move down to Feature Stack Position #1. The Feature Register stores the circle feature data. The computed circle size is modified to reflect any requested probe compensation (see **CM** command). Form is computed as specified by the **FC** command (BFD: the range of the distances from the computed feature center to the individual points; MRS: the separation between the two concentric circles of minimal separation which contain the points). The results are stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the vector normal to the active working plane in S(14,0), S(15,0) and S(16,0). The feature orientation vector (I, J, K) is set based on the plane of computation of the circle.

The Dimensional Array also stores:

- X, Y, and Z center coordinates,
- Size (radius and diameter),
- Feature type, and
- Feature number

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

None.

MD - Manual Measure Ellipse

The MD command prompts the operator to measure an ellipse. It computes the best-fit to an ellipse that is parallel to the measurement plane. Use the **CM** command to specify compensation for the probe radius.

Command Format

MLString "MD P1,P2,P3"

Examples

1. MLString "MD"
2. MLString "MD 7"
3. MLString "MD AP"

Input

- **P1** = @operator prompt@ (String; optional)
- **P2** = Number of points (Numeric; optional).
- **P3** = AP (String; optional)

P1 is used to specify a message to be shown in the manual measure ellipse window that is displayed when this command is executed. This prompt should help the operator identify the correct feature to be measured.

P2 defines the number of points to measure. At least five points are required for measurement. The first example measures five points on an elliptical feature in the active working plane. In the second example, the 7 is the number of points for the user to record. Omitting P2 selects the automatic default of five points. For segmented measurements, P1 can be 1 or 2.

P3 can only be "AP". This specifies that the ellipse is to be computed in the major plane (XY, YZ, or ZX) which is most parallel to the set of measured points rather than in the active working plane. The third example prompts the operator to measure five points on an ellipse in a major plane (XY, YZ, or ZX).

Points should be distributed evenly over as large a portion of the elliptical surface as possible give the best results.

Use the **CM** command to specify compensation for the probe radius. This adjusts calculation of the maximum and minimum radii of the ellipse. An INT compensation flag indicates compensation for measurements made inside an elliptical hole. An EXT compensation flag indicates measurements made outside a solid ellipse. A compensation flag set to OFF means no compensation occurs.

Results

The user is prompted through the manual measure ellipse window to measure the specified number of points. As points are measured they are accumulated in the internal best-fit summation array. After the points are measured the system computes the data for the ellipse. If the points do not define an ellipse in the working plane then the user is prompted to remeasure the feature.

The current contents of the Feature Register move down to Feature Stack Position #1. The Feature Register stores the ellipse feature data. The computed ellipse size is modified to reflect any requested probe compensation (see **CM** command). Form is computed as the range of the distances from the computed feature surface to the individual points. The results are stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the vector normal to the active working plane in S(14,0), S(15,0) and S(16,0). The feature orientation vector (I, J, K) is set based on the plane of computation of the ellipse.

The Dimensional Array also stores:

- Working plane coordinates of the center of the ellipse and the average of the third axis readings in X, Y, and Z,
- Minimum radius in the Radius row, S(6,0),
- Maximum radius in the Major Radius row, S(17,0),
- Angle between the axis of the ellipse and the primary axis of the work plane in the Angularity slot, S(7,0),
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

None.

MF - Manual Measure Offset Plane

The MF command prompts the operator to measure three points on surfaces defining an offset plane. An offset plane is a surface made up of three pads at two or three different levels. The location and orientation of the plane are computed and pushed into the feature stack. The location is compensated for probe size if so specified.

Command Format

MLString "MF P1,P2,P3,P4,P5,P6"

Examples

1. MLString "MF OFF,19.8,52.9,36.4,-Z"

Input

- **P1** = @operator prompt@ (String; optional)
- **P2** = X, -X, Y, -Y, Z, -Z, INT, EXT, or OFF (String; optional).
- **P3** = Offset from first measured point to nominal plane (Numeric; required).
- **P4** = Offset from second measured point to nominal plane (Numeric; required).
- **P5** = Offset from third measured point to nominal plane (Numeric; required).
- **P6** = X, -X, Y, -Y, Z, -Z (String; required).

P1 is used to specify a message to be shown in the manual measure offset plane window that is displayed when this command is executed. This prompt should help the operator identify the correct feature to be measured.

P2 specifies the desired probe compensation. Omitting P2 means no probe compensation occurs. A specified axis indicates the part axis that most nearly points in the direction away from the measured surface. The OFF in the example specifies that probe compensation does not occur.

The possible P2 arguments are:

- X,-X,+Y,-Y,+Z,-Z The specified axis points away from the measured surface.
- OFF No compensation.
- EXT The measured surface is on the side of the plane opposite the part datum.
- INT The measured surface is facing the part datum.

P3 states the offset from the first measured point to the nominal plane. In the example, the 19.8 is the amount of offset from the first measured point to the nominal plane.

P4 states the offset from the second measured point to the nominal plane. In the example, the 52.9 is the amount of offset from the second measured point to the nominal plane.

P5 states the offset from the third measured point to the nominal plane. In the example, the 36.4 is the amount of offset from the third measured point to the nominal plane.

P6 indicates the plus direction for the offset values of P3, P4, and P5. It identifies a part axis (X, -X, Y, -Y, Z, or -Z) that points from the nominal plane toward the measured points. In the example, the -Z indicates that the minus end of the Z-axis points from the nominal plane toward the measured points.

Results

The user is prompted through the manual measure offset plane window to measure the three points. As points are measured they are accumulated in the internal best-fit summation array. After the points are measured the system computes the data for the plane. If the points do not define a plane then the user is prompted to remeasure the feature.

The current contents of the Feature Register move down to Feature Stack Position #1. The Feature Register stores the plane data. The computed plane position is modified to reflect any requested probe compensation. Form is set to zero and stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the vector normal to the computed plane in S(14,0), S(15,0) and S(16,0).

The Dimensional Array also stores:

- Feature type
- Feature number

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

None.

MI - Manual Measure Cylinder

The MI command prompts the operator to measure a cylinder. Use the **CM** command to specify compensation for the probe radius.

Command Format

MLString "MI P1,P2,P3"

Examples

1. MLString "MI @Measure bore number 13-A"
2. MLString "MI 10,AXIS"
3. MLString "MI 12,PLNE"
4. MLString "MI 12,1,0,0"
5. MLString "MI .7,0,.7"

Input

- **P1** = @operator prompt@ (String; optional)
- **P2** = Number of points (Numeric; optional).
- **P3** = AXIS, PLNE, or a set of three numbers (String/numeric; optional).

P1 is used to specify a message to be shown in the manual measure cylinder window that is displayed when this command is executed. This prompt should help the operator identify the correct feature to be measured.

P2 defines the number of points to measure. At least five points are required. Omitting P2 selects the automatic default of 5 points. For segmented measurements, P1 can equal one through four. In examples 2 and 3, the 10 and 12 define the number of points for the user to record.

P3 is used to define the approximate orientation of the cylinder. Omitting P3 means that the cylinder is approximately perpendicular to the active working plane. P3 can be one of the following:

- the word "AXIS", indicating that the axis of the cylinder is approximately parallel to the line joining the first and last points measured;
- the word "PLNE", indicating that the axis of the cylinder is approximately perpendicular to the plane defined by the first three points measured; or
- a set of three numbers defining the approximate orientation of the axis of the cylinder in the active part reference frame. At least one of the three numbers must be non-zero.

Example 2 commands the operator to measure 10 points for a cylinder whose axis is approximately parallel to the line defined by the first and last points measured.

Example 3 commands the operator to measure twelve points for a cylinder. The first three points are to be measured at the same level around the axis of the cylinder so that they describe a plane perpendicular to the axis of the cylinder.

Example 4 commands the operator to measure twelve points for a cylinder whose axis is approximately in line with the X-axis.

Example 5 commands the operator to measure five points for a cylinder whose axis has direction vector approximately .7,0,.7.

Results

The user is prompted through the manual measure cylinder window to measure the specified number of points. As points are measured they are accumulated in the internal best-fit summation array. After the points are measured the system computes the data for the cylinder. If the points do not define a cylinder then the user is prompted to remeasure the feature.

The current contents of the Feature Register move down to Feature Stack Position #1. The Feature Register stores the cylinder data. The computed cylinder size is modified to reflect any requested probe compensation (see **CM** command). Form is computed as specified by the **FC** command (BFD: the range of the distances from the computed axis to the individual points; MRS: the separation between the two coaxial cylinders of minimal separation which contain the points). The results are stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the cylinder axis in S(14,0), S(15,0) and S(16,0).

The Dimensional Array also stores:

- X, Y, and Z coordinates of the centroid of the points as projected on the computed cylinder axis,
- Radius and diameter,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

None.

MJ - Manual Measure Step Cylinder

The MJ command prompts the operator to measure six points defining a step cylinder. A step cylinder is a surface made up of two coaxial cylinders with different radii. Six points are measured: three at one level on one cylinder and three at another level on the other cylinder. Use the **CM** command to specify compensation for the probe radius.

Command Format

MLString "MJ P1"

Examples

1. MLString "MJ @Measure the two levels in the central bore@"

Input

- **P1** = @operator prompt@ (String; optional)

P1 is used to specify a message to be shown in the manual measure step cylinder window that is displayed when this command is executed. This prompt should help the operator identify the correct feature to be measured.

Six points are required for the measurement. The six points must be located in two groups of three. Each group must be at a different level of the step cylinder.

Use the **CM** command to specify compensation for the probe radius. This corrects the computed radii of the measured surfaces. An INT compensation flag indicates compensation for measurements made inside a cylindrical hole. An EXT compensation flag indicates measurements made outside a solid cylinder. A compensation flag set to OFF means no compensation occurs.

Results

The user is prompted through the manual measure step cylinder window to measure the six points. As points are measured they are accumulated in the internal best-fit summation array. After the points are measured the system computes the data for the cylinders. If the points do not define a cylinder then the user is prompted to remeasure the feature.

The current contents of the Feature Register move down to Feature Stack Position #2. Data for the first level measured is moved into Feature Stack Position #1. The Feature Register stores the feature data for the second level measured. The computed cylinder sizes are modified to reflect any requested probe compensation (see **CM** command). Form is set to zero and stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines for the cylinder axis in S(14,0), S(15,0) and S(16,0).

The Dimensional Array also stores:

- Feature type, which is 10 (cylinder, O.D. or boss), 11(cylinder), or 12 (cylinder, I.D. or bore) depending on the compensation flag
- Feature number for the second level measured.

Skip Action

Not executed if part of a skipped block but two null features are pushed onto the Feature Register/Feature Stack.

Error Messages

None.

MP - Manual Measure Plane

The MP command prompts the operator to measure at least three points and then constructs a best-fit plane based on the set of recorded points.

Command Format

MLString "MP P1,P2,P3,P4"

Examples

1. MLString "MP"
2. MLString "MP INT,10"
3. MLString "MP +Y,5,0,1,0"
4. MLString "MP .7,0,.7"

Input

- **P1** = @operator prompt@ (String; optional)

- **P2** = X, -X, Y, -Y, Z, -Z, OFF, EXT, or INT (String; optional).
- **P3** = Number of points (Numeric; optional).
- **P4** = Set of thee numbers (Numeric; optional).

P1 is used to specify a message to be shown in the manual measure plane window that is displayed when this command is executed. This prompt should help the operator identify the correct feature to be measured.

P2 specifies the desired probe compensation. Omitting P2 means no probe compensation occurs. The first example shows measurement of three points describing a plane. A specified axis indicates the part axis that most nearly points in a direction away from the measured surface. In the third example, the +Y axis points generally away from the measured surface (towards the probe center). This example measures five points describing a plane approximately perpendicular to the Y-axis. The positive end of the Y-axis points in a direction away from the measured surface.

The possible P2 arguments are :

- X,-X,+Y,-Y,+Z,-Z The specified axis points away from the measured surface.
- OFF No compensation.
- EXT The measured surface faces away from the part datum.
- INT The measured surface is facing toward the part datum.

P3 defines the number of points to measure. At least three points are required for measurement. Omitting P3 selects the automatic default of 3 points. For segmented measurements, P3 can equal one or two. In the third example, the 5 defines the number of points to record.

P4 is a set of three numbers defining the approximate orientation of the plane in the active part reference frame. This may define a unit vector but this is not required. At least one of the three numbers must be non-zero.

Results

The user is prompted through the manual measure plane window to measure the specified number of points. After the points are measured the system computes the data for the plane. If the recorded points do not define a plane then the operator is prompted to remeasure the feature.

The current contents of the Feature Register move down to Feature Stack Position #1. The Feature Register stores the plane data. The computed plane position is modified to reflect any requested probe compensation. Form (flatness) is computed as specified by the **FC** command (BFD: the sum of the distances from the computed line to the points furthest away on opposite sides of the computed plane; MRS: the minimal separation between two parallel planes which contains the set of points). The form results are stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the normal to the computed plane in the S(14,0), S(15,0), and S(16,0).

The Dimensional Array also stores :

- Position of the plane,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

None.

MQ - Manual Measure Cone

The MQ command prompts the operator to measure at least six points and then constructs a best-fit cone based on the set of recorded points. Use the **CM** command to specify compensation for the probe radius.

Command Format

MLString "MQ P1,P2,P3"

Examples

1. MLString "MQ"
2. MLString "MQ 12"
3. MLString "MQ 1,0,0"
4. MLString "MQ 15,.7,0,.7"

Input

- **P1** = @operator prompt@ (String; optional)
- **P2** = Number of points (Numeric; optional).
- **P3** = Set of thee numbers (Numeric; optional).

P1 is used to specify a message to be shown in the manual measure cone window that is displayed when this command is executed. This prompt should help the operator identify the correct feature to be measured.

P2 defines the number of points to measure. At least six points are required for measurement. Omitting P2 selects the automatic default of 6 points. For segmented measurements, P2 can equal one. In the second example, the 12 defines the number of points to record.

P3 is a set of three numbers defining the approximate orientation of the axis of the cone in the active part reference frame. This may define a unit vector but this is not required. At least one of the three numbers must be non-zero. If P3 isn't specified, then the first three points must be measured at one level on the conical surface and the fourth, fifth and sixth points must be measured at a different level. Any other points can be distributed at random on the conical surface. Evenly distribute points over as large a portion of the conical surface as possible to ensure the best results. If P3 is specified then the points may be measured in any order but should still be distributed as evenly as possible over the complete feature surface.

The **CM** command is used to specify compensation for the probe radius. This adjusts the computed location of the cone's apex based upon whether the probe center was inside or outside the conical surface. An INT compensation flag indicates compensation for measurements made inside a conical hole. An EXT compensation flag indicates measurements made outside a solid cone. A compensation flag set to OFF means no compensation occurs.

Results

The user is prompted through the manual measure cone window to measure the specified number of points. After the points are measured the system computes the data for the cone. If the recorded points do not define a cone then the operator is prompted to remeasure the feature.

The current contents of the Feature Register move down to Feature Stack Position #1. The Feature Register stores the cone data. The computed cone apex position is modified to reflect any requested probe compensation. Form is computed as the sum of the distances from the computed cone surface to the points furthest away on opposite sides of the computed surface. The form results are stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the axis of the computed cone in the S(14,0), S(15,0), and S(16,0).

The Dimensional Array also stores:

- The coordinates of the apex of the cone in X, Y, and Z,
- The included apex angle in S(7,0),
- The feature type, and
- The feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

None.

MS - Manual Measure Sphere

The MS command prompts the operator to measure at least four points on a sphere and then computes the best-fit to a sphere based upon the measured points. Use the **CM** command to specify compensation for the probe radius.

Command Format

MLString "MS P1,P2"

Examples

1. MLString "MS 5"

Input

- **P1** = @operator prompt@ (String; optional)
- **P2** = Number of points (Numeric; optional)

P1 is used to specify a message to be shown in the manual measure sphere window that is displayed when this command is executed. This prompt should help the operator identify the correct feature to be measured.

P2 defines the number of points to measure. At least four points are required for measurement. The example measures five points. Omitting P2 selects the automatic default of four points. For segmented measurements, P2 can be less than four.

Results

The user is prompted through the manual measure sphere window to measure the specified number of points. As points are measured they are accumulated in the internal best-fit summation array. After the points are measured the system computes the data for the sphere. If the points do not define a sphere then the user is prompted to remeasure the feature.

The current contents of the Feature Register move down to Feature Stack Position #1. The Feature Register stores the sphere feature data. The computed sphere size is modified to reflect any requested probe compensation (see the **CM** command). Form is computed as specified by the **FC** command (BFD: the range of the distances from the computed feature center to the individual points; MRS: the separation between the two concentric spheres of minimal separation which contain the points). The results are stored in the form deviation slot, S(13,4), of the Dimensional Array. The direction cosines are set to zero in S(14,0), S(15,0), and S(16,0).

The Dimensional Array also stores:

- X, Y, and Z center coordinates
- Size (radius and diameter),
- Feature type, and
- Feature number

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

None.

QZ - Manual Single Point Measurement

The QZ command requests a single point measurement, allowing for selective operator prompting at the RCU and host computer.

Command Format

MLString "QZ P1,P2,P3"

Examples

1. MLString "QZ @MEASURE POINT@,ON,OFF"
2. MLString "QZ @MEASURE POINT@ ,OFF"
3. MLString "QZ"

Input

- **P1** = @operator prompt@ (String; optional)
- **P2** = ON or OFF (String; optional).
- **P3** = ON or OFF (String; optional).

The optional parameter P1 is a user prompt, bracketed by @ signs. If no user prompt is selected, the system default (#nn, MEAS 1 PT, TIP#) is used.

The optional parameter P2 may be either "ON" or "OFF", indicating whether the prompt is to be sent to the RCU. "ON" indicates that it is to be sent.

The optional parameter P3 may be either "ON" or "OFF", indicating whether the prompt is to be sent to the host computer. "ON" indicates that it is to be sent. P2 must be specified if P3 is specified. The default for both P2 and P3 is "ON".

Results

If specified, the operator is prompted. The WAIT light on the MP-30/35 is turned on. The MP waits for a touch probe deflection or for the RCU Record button (or footswitch) to be pressed. If a touch probe deflection occurs, the touched point is recorded. If the Record button (or footswitch) is pressed, the current probe position is recorded.

If a touch occurs, a second set of position data is captured 0.032 seconds before contact is made. This is used to compute a unit-length vector directed away from the touched surface by taking into account whether the probe is rearmed when the delayed point is captured. If the points are identical (or if no touch occurs), the vector is set to (0,0,0). This vector should only be used as a general indication of the orientation of the surface and should not be used directly for probe compensation.

The recorded point is pushed into the Feature Register. The feature type is point (1). The surface vector as described above is stored as the orientation (I,J,K). The feature number is stored in the Feature Register. The feature form is set to zero. The WAIT light is turned off, the prompt (if any) is erased from the RCU, and the recorded data is returned to the host computer. The variable S7 is set to zero (0) if a touch probe is in use but a non-touch point is recorded. This variable is set to one (1) in all other cases.

Skip Action

Not executed if part of skipped block but a null feature is pushed onto the Feature Register / Stack.

Error Messages

None.

8. Feature Reduction Group

MLB commands in the Feature Reduction Group provide the computational algorithms used to construct geometric features.

The Feature Reduction Group includes the following MLB commands.

CN	**Cone** computes the location, orientation and included angle of a cone from five points in the stack.
CR	**Circle** computes a circle that is parallel to the measurement plane. The command uses three points in the stack.
CY	**Cylinder** computes a cylinder from five points in the stack.
EL	**Ellipse** computes the location, orientation, and maximum and minimum radii of an ellipse parallel to the working plane from five points in the stack.
L2	**2D Line** computes the position and attitude of a line parallel to the working plane from two points in the stack.
L3	**3D Line** computes the position and attitude of a line in three-dimensional space from two points in the stack.
OP	**Offset Plane** computes the parameters from a plane described by three points. The points can exists in the stack or one of the summation arrays.
PN	**Plane** computes the centroid and orientation of a plane from three points in the stack.
SC	**Step Cylinder** computes the location, orientation and radii of the step cylinder from six points. The six points can be in the stack or one of the summation arrays.
SH	**Sphere** computes a sphere from four points in the stack.

CN - Cone

The CN command computes the location, orientation, and included angle of a cone using six points: the point in the Feature Register and the top five points in the Feature Stack. Use the **CM** command to specify compensation for the probe radius.

Command Format

MLString "CN"

Examples

1. MLString "CN"

Input

The points in FR, F1, and F2 must define one level on the conical surface. The points in F3, F4, and F5 must lie at a different level on the cone. Use the **CM** command to specify compensation for the probe radius. This adjusts the computations of the cone's apex. The **CM** argument INT indicates the points were taken inside a conical hole; EXT indicates the points were taken outside a solid cone; and OFF indicates that the calculations are not to compensate for the probe size. Compensation is based upon the size of the currently active probe (not necessarily the probe active when the points were measured).

Results

The current contents of the Feature Register move into F1 of the Feature Stack. The command computes the cone parameters and stores them in the Feature Register. The apex angle of the cone is the total angle between the sides of the cone at its apex. Measurement of the angle occurs in a plane containing the cone's axis. The system stores this included angle in the angularity slot (S(7,0)) of the Dimensional Array. The command sets conicity to zero in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines describing the orientation of the cone's axis in S(14,0), S(15,0), and S(16,0).

The Dimensional Array also stores:

- Coordinates of the apex in X, Y, and Z,
- Apex angle in S(7,0),
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skip block but null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register does not contain the required six points.
- MLB-033 One of the six features is a plane.
- MLB-041 The points do not define a cone. This may occur if the points are not properly ordered (see Input section above).

CR - Circle

The CR command computes a circle that is parallel to the active working plane using three points: the point in the Feature Register and the top two points in the Feature Stack. Use the **CM** command to specify compensation for the probe radius.

Command Format

MLString "CR"

Examples

1. MLString "CR"

Input

The points in FR, F1, and F2 must define a circle in the working plane. Use the **CM** command to specify compensation for the probe radius. This adjusts the computations of the circle's size. The **CM** argument INT indicates the points were taken inside a hole; EXT indicates the points were taken outside a boss; and OFF indicates that the calculations are not to compensate for the probe size. Compensation is based upon the size of the currently active probe (not necessarily the probe active when the points were measured).

Results

The current contents of the Feature Register are pushed down. The Feature Register stores the circular feature data. The command sets roundness to zero in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the vector normal to the active working plane in S(14,0), S(15,0), and S(16,0).

The Dimensional Array also stores:

- Coordinates of the circle center in X, Y, and Z,
- Feature radius and diameter,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skip block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register does not contain the required three points.
- MLB-033 One of the three features is a plane.
- MLB-041 The points do not define a circle in the active working plane.

CY - Cylinder

The CY command computes the location, orientation, and size of a cylinder using five points: the point in the Feature Register and the top four points in the Feature Stack. Use the **CM** command to specify compensation for the probe radius.

Command Format

MLString "CY P1"

Examples

1. MLString "CY"
2. MLString "CY AXIS"

Input

- **P1** = AXIS or PLNE; (String, optional)

P1 is used to indicate the approximate alignment of the cylinder to be computed. If P1 is not specified then the cylinder is approximately perpendicular to the active working plane. If P1 is

AXIS then the cylinder's axis is approximately parallel to the line defined by the two points FR and F4. If P1 is PLNE then the axis of the cylinder is approximately perpendicular to the plane defined by points F2, F3, and F4 (i.e., these three points all lie at the same cross-section along the cylinder).

Use the **CM** command to specify compensation for the probe radius. This adjusts the computations of the cylinder's size. The **CM** argument INT indicates the points were taken inside a cylindrical hole; EXT indicates the points were taken outside a solid cylinder; and OFF indicates that the calculations are not to compensate for the probe size. Compensation is based upon the size of the currently active probe (not necessarily the probe active when the points were measured).

Results

The current contents of the Feature Register move down. The Feature Register stores the cylinder data. Cylindricity is set to zero in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the computed cylinder axis in S(14,0), S(15,0), and S(16,0).

The Dimensional Array also stores:

- Coordinates of the centroid of the measurements projected onto the cylinder axis,
- Cylinder diameter and radius,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skip block but null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register does not contain the required five points.
- MLB-033 One of the five features is a plane.
- MLB-041 The points do not appear to define a cylinder. This may occur if the points are not consistent with the P1 setting (see Input section above).

EL - Ellipse

The EL command computes an ellipse that is parallel to the measurement plane using five points: the point in the Feature Register and the top four points in the Feature Stack. Use the **CM** command to specify compensation for the probe radius.

Command Format

MLString "EL"

Examples

1. MLString "EL"

Input

The points in FR, F1, F2, F3, and F4 must define an ellipse in the working plane. Use the **CM** command to specify compensation for the probe radius. This adjusts the computations of the ellipse's size. The **CM** argument INT indicates the points were taken inside a hole; EXT indicates the points were taken outside a boss; and OFF indicates that the calculations are not to compensate for the probe size. Compensation is based upon the size of the currently active probe (not necessarily the probe active when the points were measured).

Results

The current contents of the Feature Register are pushed down. The Feature Register stores the ellipse feature data. The command sets form to zero in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the vector normal to the active working plane in S(14,0), S(15,0), and S(16,0).

The Dimensional Array also stores:

- Coordinates of the ellipse center in X, Y, and Z,
- Feature major radius, minor radius and diameter (twice the minor radius),
- Angle between the axis of the ellipse and the first-named axis of the working plane in S(7,0),
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register does not contain the required five points.
- MLB-033 One of the five features is a plane.
- MLB-041 The points do not define an ellipse in the active working plane.

L2 - 2D Line

The L2 command computes a line that is parallel to the active working plane using two points: the point in the Feature Register and the top point in the Feature Stack. The location of the line can be compensated for the probe radius.

Command Format

MLString "L2 P1"

Examples

1. MLString "L2 +Z"

Input

- **P1** = X, -X, Y, -Y, Z, -Z, OFF, EXT, or INT (String; optional).

P1 specifies the desired probe compensation. In the example, the +Z indicates the desired probe compensation. It means that the positive end of the Z-axis points away from the measured surface. Omitting P1 means no probe compensation occurs. A specified axis should be the part axis that most nearly points in the direction away from the measured surface. The possible arguments are:

- X, -X, Y, -Y, Z, -Z The specified axis points away from the measured surface.
- OFF No compensation.
- EXT The measured surface is on the side of the line opposite the part datum.

- INT The measured surface faces the part datum.

Results

The current contents of the Feature Register move down. The Feature Register stores the line feature data. The computed line position is modified to reflect requested probe compensation. The feature form is set to zero in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the computed line in S(14,0), S(15,0), and S(16,0).

The Dimensional Array also stores :

- Coordinates of the midpoint of the line as base point of the line,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-021 The first parameter is invalid.
- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-033 At least one of the required features is a plane.
- MLB-041 The points do not define the feature.

L3 - 3D Line

The L3 command computes a line in space using two points: the point in the Feature Register and the top point in the Feature Stack.

Command Format

MLString "L3"

Examples

1. MLString "L3"

Input

The Feature Register and the first position in the Feature Stack must contain valid features.

Results

The current contents of the Feature Register move down. The Feature Register stores the line feature data. The command sets spatial straightness to zero in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the computed line in S(14,0), S(15,0), and S(16,0). The Dimensional Array also stores:

- Coordinates of the midpoint of the line as the base point of the line,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-033 One of the feature positions contains a plane.
- MLB-041 The points do not define the feature.

OP - Offset Plane

The OP command computes the parameters for the plane described by the point in the Feature Register and the top two points in the Feature Stack or from three points in a specified summation array. An offset plane is a surface made up of three pads at two or three different levels. The location and orientation of the plane are computed and pushed into the feature stack. The location is compensated for probe size if so specified.

Command Format

MLString "OP P1,P2,P3,P4,P5,P6"

Examples

1. MLString "OP INTERNAL,-Z,65.524,36.666,132.989,+Z"

Input

- **P1** = USER or INTERNAL (String; optional)
- **P2** = X, -X, Y, -Y, Z, -Z, INT, EXT, or OFF (String; optional).
- **P3** = Offset from first measured point to nominal plane (Numeric; required).
- **P4** = Offset from second measured point to nominal plane (Numeric; required).
- **P5** = Offset from third measured point to nominal plane (Numeric; required).
- **P6** = X, -X, Y, -Y, Z, -Z (String; required).

P1 specifies the location of the point data used for the computation. If P1 is omitted, the command uses data from the stack. If P1 is USER, the command uses data from the user summation array. If P1 is INTERNAL, the command uses data from the internal summation array. In the example, INTERNAL means that point data has been summed into the internal summation array.

P2 specifies the desired probe compensation. Omitting P2 means no probe compensation occurs. A specified axis indicates the part axis that most nearly points in the direction away from the measured surface. The -Z in the example specifies that probe compensation occurs based upon the probe being of the negative Z side of the surface.

The possible P2 arguments are:

- X,-X,+Y,-Y,+Z,-Z The specified axis points away from the measured surface.
- OFF No compensation.
- EXT The measured surface is on the side of the plane opposite the part datum.

- INT The measured surface is facing the part datum.

P3 states the offset from the first point to the nominal plane. In the example, the 65.524 is the amount of offset from the first point to the nominal plane.

P4 states the offset from the second point to the nominal plane. In the example, the 36.666 is the amount of offset from the second point to the nominal plane.

P5 states the offset from the third point to the nominal plane. In the example, the 132.989 is the amount of offset from the third point to the nominal plane.

P6 indicates the plus direction for the offset values of P3, P4, and P5. It identifies a part axis (X, -X, Y, -Y, Z, or -Z) that points from the nominal plane toward the measured points. In the example, the -Z indicates that the minus end of the Z-axis points from the nominal plane toward the measured points.

Results

The system computes the data for the plane. The current contents of the Feature Register move down to Feature Stack Position #1. The Feature Register stores the plane data. The computed plane position is modified to reflect any requested probe compensation. Form is set to zero and stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the vector normal to the computed plane in S(14,0), S(15,0) and S(16,0).

The Dimensional Array also stores:

- Feature type, and
- Feature number

Skip Action

Not executed if part of a skip block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-033 One of the feature positions contains a plane.
- MLB-040 The summation array does not contain the desired feature.
- MLB-041 The points do not define the feature.

PN - Plane

The PN command computes a plane using three points: the point in the Feature Register and the top two points in the Feature Stack.

Command Format

MLString "PN P1"

Examples

1. MLString "PN +Z"
2. MLString "PN OFF"

Input

- **P1** = X, -X, Y, -Y, Z, -Z, OFF, EXT, or INT (String; optional).

P1 specifies the desired probe compensation. In the first example, the +Z indicates the desired probe compensation. It means that the positive end of the Z-axis points away from the measured surface. Omitting P1 means no probe compensation occurs. A specified axis should be the part axis that most nearly points in the direction away from the measured surface. The possible arguments are:

- X, -X, Y, -Y, Z, -Z The specified axis points away from the measured surface.
- OFF No compensation.
- EXT The measured surface is on the side of the line opposite the part datum.
- INT The measured surface faces the part datum.

Results

The current contents of the Feature Register move down. The Feature Register stores the plane feature data. The computed plane position is modified to reflect requested probe compensation. The feature form is set to zero in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the computed plane in S(14,0), S(15,0), and S(16,0).

The Dimensional Array also stores :

- The coordinates of the centroid of the three points as the position of the plane,
- The feature type, and
- The feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-033 At least one of the specified features is a plane.
- MLB-041 The points do not define the feature.

SC - Step Cylinder

The SC command computes the parameters for the two levels of a step cylinder described by the point in the Feature Register and the top five points in the Feature Stack or from six points in a specified summation array. A step cylinder is made up of two coaxial cylindrical surfaces with different radii. The size data is compensated for probe size as specified by the **CM** command.

Command Format

MLString "SC P1"

Examples

1. MLString "SC USER"

Input

- **P1** = USER or INTERNAL (String; optional)

P1 specifies the location of the point data. If P1 is omitted, the system uses data from the stack to compute the step cylinder. If P1 is USER, data from the user summation array is used to compute the step cylinder. If P1 is INTERNAL, data from the internal summation array is used to compute the step cylinder. In the example, the USER indicates that six points from the user summation array is to be used to compute the step cylinder.

The six points must be located on a step cylinder: the first three points on one level and the second three points on a different level.

Use the **CM** command to specify compensation for the probe radius. This adjusts calculation of the radii of the measured surfaces. An INT compensation flag indicates compensation for measurements made inside a cylindrical hole. An EXT compensation flag indicates measurements made outside a solid cylinder. A compensation flag set to OFF means no compensation occurs.

Results

The data for the two levels of the step cylinder is computed. The current contents of the Feature Register move down to Feature Stack Position #2. Data for the first level is moved into Feature Stack Position #1. The Feature Register stores the feature data for the second level. The computed cylinder sizes are modified to reflect any requested probe compensation (see **CM** command). Form is set to zero and stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines for the cylinder axis in S(14,0), S(15,0) and S(16,0).

The Dimensional Array also stores:

- Feature type, which is 10 (cylinder, O.D. or boss), 11(cylinder), or 12 (cylinder, I.D. or bore) depending on the compensation flag
- Feature number for the second level.

Skip Action

Not executed if part of a skipped block but two null features are pushed onto the Feature Register/Feature Stack.

Error Messages

- MLB-030 The Feature Stack or Feature Register does not contain the required features.
- MLB-033 One of the feature positions contains a plane.
- MLB-040 The summation array does not contain at least six points.
- MLB-041 The points do not define a step cylinder.

SH - Sphere

The SH command computes a sphere using four points: the point in the Feature Register and the top three points in the Feature Stack. Use the **CM** command to specify compensation for the probe radius.

Command Format

MLString "SH"

Examples

1. MLString "SH"

Input

The points in FR, F1, F2, and F3 must define a sphere. Use the **CM** command to specify compensation for the probe radius. This adjusts the computations of the sphere's size. The **CM** argument INT indicates the points were taken inside a hole; EXT indicates the points were taken outside a sphere; and OFF indicates that the calculations are not to compensate for the probe size. Compensation is based upon the size of the currently active probe (not necessarily the probe active when the points were measured).

Results

The current contents of the Feature Register are pushed down. The Feature Register stores the computed sphere data. The command sets form to zero in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array sets the direction cosines in S(14,0), S(15,0), and S(16,0) to zero.

The Dimensional Array also stores:

- Coordinates of the sphere center in X, Y, and Z,
- Feature radius and diameter,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-033 One of the feature positions contains a plane.
- MLB-041 The points do not define a sphere.

9. Best-fit Reduction Group

MLB commands in the Best-fit Reduction Group provide the computational algorithms used to construct best-fit geometric features.

The Best-fit Reduction Group includes the following MLB commands.

B2	**Best-fit 2D Line** computes the best-fit position and attitude of a line from points in the selected summation array. The line computation projects the line on a plane parallel t the work plane.
B3	**Best-fit 3D Line** computes the best-fit position and attitude of a line from points in the selected summation array. The line is computed in three-dimensional space.
BC	**Best-fit Circle** computes the best-fit to a circle from points in the specified summation array.
BD	**Best-fit Ellipse** computes the best-fit location, orientation, and maximum and minimum radii for an ellipse. It uses points from the specified summation array to perform the calculations.
BI	**Best-fit Cylinder** computes the best-fit to a cylinder from points in the selected summation array.
BK	**Best-fit Cone** computes the best-fit location, orientation, and included angle for a cone. It uses points from the specified summation array to perform the calculations.
BM	**Best-fit Summation** sums the position of the specified feature into the specified summation array.
BP	**Best-fit Plane** computes the best-fit to a plane from points in a specified summation array.
BS	**Best-fit Sphere** computes the center point, diameter, and best-fit to a sphere from points in the specified summation array.
BT	**Compute Best-fit Point**
BW	**Compute Best-fit Slot / Web**
BX	**Compute Best-fit Cparln Feature** computes the best-fit Cparlnf (Closed parallel lines - flat end) or Cparlnr (Closed parallel lines - round end) in the working plane to a previously accumulated set of points.
CB	**Clear Summation Array** clears and initializes the summation array.
FF	**Fetch Form Feature Data**
HL	**High/Low Point Retrieval** retrieves the high and low points determined when best-fit feature form is computed.
QB	**Query Best-fit Summation Status**

RB	**Recall Best-fit Point** loads the Feature Register with the specified point data from one of the summation arrays.
UM	**Unsum a Point from the Summation Array**

B2 - Best-fit 2D Line

The B2 command computes the best-fit position and attitude of a line through the points in the selected summation array. The command computes the line as projected on to a plane parallel to the working plane and through the centroid of the points. The P1 argument specifies the summation array. The P2 argument compensates for the probe radius. The P3 argument defines the approximate orientation of the line in the active part reference frame.

Command Format

MLString "B2 P1,P2,P3"

Examples

1. MLString "B2"
2. MLString "B2 INTERNAL,INT"
3. MLString "B2 +X,0,1,0"
4. MLString "B2 INTERNAL,.7,0,.7"

Input

The selected summation array must contain at least 2 points.

- **P1** = INTERNAL (String; optional).
- **P2** = X, -X, Y, -Y, Z, -Z, OFF, EXT, or INT (String; optional).
- **P3** = vector defining approximate orientation (Numeric; optional).

If you include P1, it must equal INTERNAL. This specifies that this command uses the points found in the internal summation array. In the second example, INTERNAL indicates that the points for the computation exist in the internal summation array. Omitting P1 causes the command to use the points in the user summation array.

P2 specifies the desired probe compensation. If P2 is omitted, no probe compensation occurs. A specified axis indicates the part axis that most nearly points in the direction away from the measured surface. OFF indicates that no probe compensation takes place.

The possible values for P2 are :

- X,-X,+Y,-Y,+Z,-Z The specified axis points away from the measured surface.
- OFF No compensation.
- EXT The measured surface is on the side of the plane opposite the part datum.
- INT The measured surface is facing the part datum.

The optional parameter P3 is a set of three numbers defining the approximate orientation of the line in the active part reference frame. This may define a unit vector but is not required. At least one of the two numbers corresponding to the active plane coordinates must be non-zero. The third example is computing a line parallel to the active working plane approximately parallel to the Y-axis using the points in the user summation array. It is also compensating for the probe in the +X direction. The fourth example is computing a line parallel to the active working plane whose direction vector is approximately .7,0,.7 using the points in the internal summation array.

Results

The contents of the Feature Register move into the Feature Stack. The position of the computed line reflects any requested probe compensation. The Feature Register stores the data for the line. Form (planar straightness) is computed as specified by the **FC** command (BFD: the sum of the distances from the computed line to the points furthest away on opposite sides of the line; MRS: the minimal separation between two parallel lines in the working plane which contains the set of points). The form results are stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the computed line in S(14,0), S(15,0) and S(16,0).

The Dimensional Array also stores:

- The coordinates of the centroid in X, Y, and Z,
- The feature type, and
- The feature number

Skip Action

Not executed if part of a skip block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-021 First parameter is invalid.
- MLB-040 The summation array does not contain sufficient features.
- MLB-041 The points do not define the feature.
- MLB-042 The summations are invalid.
- MLB-043 A segmented measurement is in progress.

B3 – Best-fit 3D Line

The B3 command computes the best-fit position and attitude of a line through the points in the selected summation array. This command computes the line in three-dimensional space.

Command Format

MLString "B3 P1,P2"

Examples

1. MLString "B3"
2. MLString "B3 INTERNAL"
3. MLString "B3 1,0,0"
4. MLString "B3 INTERNAL,.7,0,.7"

Input

The selected summation array must contain at least 2 points.

- **P1** = INTERNAL (String; optional).
- **P2** = A set of three numbers (Numeric; optional).

If P1 is specified, it must equal INTERNAL. This specifies that this command uses the points found in the internal summation array. The INTERNAL in the second example indicates that the points the command uses for the computation are in the internal summation array. Omitting P1 causes this command to use the points in the user summation array.

The optional parameter P2 defines the approximate orientation of the line in the active part reference frame. This may define a unit vector but is not required. At least one of the three numbers must be non-zero. If P2 isn't specified, the line computed is assumed to run approximately from the first point to the last point in the summation array.

Example 3 computes a line approximately in line with the X-axis using the points in the user summation array. Example 4 computes a line whose direction vector is approximately .7,0,.7, using the points in the internal summation array.

Results

The contents of the Feature Register move into the Feature Stack. The Feature Register stores the data for the line. Form (spatial straightness) is computed as specified by the **FC** command (BFD: twice the maximum distance between the best-fit line and the individual points; MRS: the diameter of the smallest cylinder which contains the set of points). The form results are stored in the form deviation slot, S(13,4), in the Dimensional Array. The Dimensional Array stores the direction cosines of the computed line in S(14,0), S(15,0) and S(16,0).

The Dimensional Array also stores:

- Coordinates of the centroid in X, Y, and Z,
- Feature type, and
- Feature number

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-040 The summation array does not contain sufficient features.
- MLB-041 The points do not define the feature.
- MLB-042 The summations are invalid.
- MLB-043 A segmented measurement is in progress.

BC - Best-fit Circle

The BC command uses the points in the summation array to compute the best-fit to a circle. The computed circle is parallel to one of the major planes, or optionally, a user-specified plane. Use the probe compensation command **CM** to specify compensation for the probe radius.

Command Format

MLString "BC P1,P2,P3,P4"

Examples

1. MLString "BC"
2. MLString "BC INTERNAL"
3. MLString "BC AP"
4. MLString "BC AP,R11.5"
5. MLString "BC V1,1,0"

Input

The selected summation array must contain at least 3 points.

- **P1** = INTERNAL (String; optional).
- **P2** = AP (String; optional)
- **P3** = R followed by positive number (Optional)
- **P4** = V followed by 3 values defining a direction vector normal to the plane the circle fit is to be performed in, rather than the active working plane. (Optional)

If P1 is included, it must equal INTERNAL. This specifies that the command uses the points found in the internal summation array. Omitting P1 causes the command to use the points in the user summation array. Omitting P1 in the example means that the computation uses the points found in the user summation array.

P2 can only be "AP". This specifies that the circle is to be computed in the major plane (XY, YZ, or ZX) which is most parallel to the set of measured points rather than in the active working plane. The third example computes a circle in a major plane (XY, YZ, or ZX) based on the set of points in the user summation array.

P3 can only be the letter R followed by a positive value specifying the radius of the arc described by the points. Note that this value is the size of the arc after probe compensation (i.e, the nominal feature size). If P3 is specified then the fit type is least-squares (**CM** fit specification ignored) and the form calculation is based upon the least-squares computed center (**FC** specification ignored). The fourth example computes the center of an arc of radius 11.5 units in a major plane (XY, YZ, or ZX) based on the set of points in the user summation array.

P4 can only by the letter V followed by three values defining a direction vector in the current reference frame. This vector need not be a unit vector, but must be non-zero. If P4 is specified then the circle fit is performed in the plane normal to this vector rather than in the active working plane. Note that P2 and P4 cannot both be specified.

Results

The current contents of the Feature Register move into the Feature Stack. The Feature Register stores the circular feature data. The feature form is computed as specified by the **FC** command (BFD: the range of the distances from the computed feature center to the individual points; MRS: the separation between the two concentric circles of minimal separation which contain the points). The results are stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the vector normal to the active working plane in

S(14,0), S(15,0) and S(16,0). The feature orientation vector (I,J,K) is set based on the plane of computation of the circle.

The Dimensional Array also stores:

- Center coordinates,
- Feature radius,
- Feature diameter,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skip block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-040 The summation array does not contain at least three points.
- MLB-041 The points do not define a circle.
- MLB-042 The summations are invalid.
- MLB-043 A segmented measurement is in progress.

BD - Best-fit Ellipse

The BD command computes the best-fit location, orientation, and maximum and minimum radii parameters for an ellipse. At least five points must be in the specified summation array to define the ellipse. Use the probe compensation command **CM** to specify compensation for the probe radius.

Command Format

MLString "BD P1,P2,P3"

Examples

1. MLString "BD"
2. MLString "BD INTERNAL"
3. MLString "BD AP"
4. MLString "BD V1,1,0"

Input

The summation array must contain at least 5 points. The measured points must be located on an elliptical surface in the active working plane. Points distributed evenly over as large a portion of the elliptical surface as possible ensure the best results.

Use the **CM** command to specify compensation for the probe radius. This adjusts computation of the radii of the ellipse. An INT compensation flag indicates compensation for measurements made inside a elliptical hole. An EXT compensation flag indicates measurements made outside a solid ellipse. A compensation flag set to OFF means no compensation occurs.

- P1 = INTERNAL (String; optional)
- P2 = AP (String; optional)

- P3 = V#,#,# (V followed by a vector; optional)

If you include P1, it must equal INTERNAL. This specifies that the command uses the points found in the internal summation array. The INTERNAL in the example indicates that the internal summation array is the array that contains the points for the best-fit computation. Omitting P1 causes this command to use the points in the user summation array.

P2 can only be "AP". This specifies that the ellipse is to be computed in the major plane (XY, YZ, or ZX) which is most parallel to the set of measured points rather than in the active working plane. The third example computes an ellipse in a major plane (XY, YZ, or ZX) based on the set of points in the user summation array.

P3 (optional) can only be the letter V followed by three values defining a direction vector in the current reference frame. This vector need not be a unit vector but must be non-zero. If p3 is specified then the ellipse fit is performed in the plane normal to this vector rather than in the working plane. Note that P2 and P3 cannot both be specified.

Results

The current contents of the Feature Register moves into Feature Stack Position # 1. The Feature Register stores the best-fit ellipse parameters. Feature form is computed as the sum of the maximum distances from the best-fit ellipse to points inside and outside the ellipse. The result is stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the vector normal to the active working plane of points defining the ellipse in S(14,0), S(15,0), and S(16,0). The feature orientation vector (I, J, K) is set based on the plane of computation of the ellipse.

The Dimensional Array also stores:

- Working plane coordinates of the center of the ellipse and the average of the third axis readings in X, Y, and Z,
- Minimum radius in S(6,0),
- Maximum radius in S(17,0),
- Angle between the axis of the ellipse and the first-named axis of the working plane in the angularity slot, S(7,0),
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-040 The summation array does not contain sufficient features.
- MLB-041 The points do not define the feature.
- MLB-042 The summations are invalid.
- MLB-043 A segmented measurement is in progress.

BI - Best-fit Cylinder

The BI command uses the points in the summation array to compute the best-fit to a cylinder. Use the **CM** command to specify compensation for the probe radius.

Command Format

MLString "BI P1,P2"

Examples

1. MLString "BI"
2. MLString "BI INTERNAL,AXIS"
3. MLString "BI INTERNAL,PLNE"
4. MLString "BI 1,0,0"
5. MLString "BI INTERNAL,.7,0,.7"

Input

The summation array must contain at least 5 points.

- P1 = INTERNAL (String; optional).
- P2 = AXIS, PLNE, or set of three numbers (String; optional).

If P1 is included, it must equal INTERNAL. This specifies that this command uses the points found in the internal summation array. Omitting P1 causes this command to use the points in the user summation array. Example 1 omits the P1 argument, indicating that the command uses the points in the user summation array by default.

P2 must equal AXIS, PLNE, or a set of three numbers defining the approximate orientation of the axis of the cylinder in the active part reference frame. If a set of three numbers, at least one of the three numbers must be non-zero.

P2 equal to AXIS indicates that the cylinder axis is approximately parallel to the line defined by the first and last points measured. The AXIS in the second example indicates that the cylinder axis is approximately parallel to the line formed by the first and last recorded points in the internal summation array. The cylinder does not have to be perpendicular to the working plane.

P2 equal to PLNE indicates that the axis of the cylinder is approximately perpendicular to the plane defined by the first three points. The PLNE in the third example computes a cylinder using the points in the internal summation array whose axis is approximately perpendicular to the plane defined by the first three points in the array.

The set of numbers in the fourth example computes a cylinder whose axis is approximately in line with the X-axis using the points in the user summation array.

The set of numbers in the fifth example computes a cylinder whose axis has direction vector approximately .7,0,.7 using the points in the user summation array.

Omitting P2 indicates that the cylinder is approximately perpendicular to the working plane.

Results

The current contents of the Feature Register move into the Feature Stack. The Feature Register stores the cylinder data. The feature form is computed as specified by the **FC** command (BFD: the sum of the maximum distances from the best-fit cylinder surface to points on the inside and outside of the cylinder; MRS: the radial separation between the two coaxial cylinders of minimal radial separation which contain the set of points). The result is stored in the form deviation slot, S(13,4), in the Dimensional Array. The Dimensional Array stores the direction cosines of the computed cylinder axis in S(14,0), S(15,0) and S(16,0).

The Dimensional Array also stores:

- The coordinates of the centroid of the points as projected on the cylinder axis,
- The feature radius,
- The feature diameter,
- The feature type, and
- The feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-040 The summation array does not contain sufficient features.
- MLB-041 The points do not define the feature.
- MLB-042 The summations are invalid.
- MLB-043 A segmented measurement is in progress.

BK - Best-fit Cone

The BK command computes the best-fit location, orientation, and included angle parameters for a cone. The command uses at least six points from the specified summation array to perform these computations. Use the **CM** command to specify compensation for the probe radius.

Command Format

MLString "BK P1,P2"

Examples

1. MLString "BK"
2. MLString "BK INTERNAL"
3. MLString "BK 1,0,0"
4. MLString "BK INTERNAL,.7,0,.7"

Input

The summation array must contain at least 6 points. The command requires that the first three measured points be located at one level on the conical surface. The location of the fourth, fifth and sixth points must be at a different level. Any other points can be distributed at random on the conical surface. Evenly distributed points over as large a portion of the conical surface as possible ensure the best results.

The previously specified **CM** command corrects the coordinates of the cone's apex for probe radius. An INT compensation flag indicates compensation for measurements made inside a conical hole. An EXT compensation flag indicates measurements made outside a solid cone. A compensation flag set to OFF means no compensation occurs.

- **P1** = INTERNAL (String; optional).
- **P2** = Set of three numbers (Numeric; optional).

P1 can only be INTERNAL. This specifies that the command uses the points found in the internal summation array. Omitting P1 causes this command to use the points in the user summation array. The example omits P1, indicating that the command uses the points found in the user summation array for the computations.

P2 is a set of three numbers defining the approximate orientation of the axis of the cone in the active part reference frame. This may define a unit vector but is not required. At lease one of the three numbers must be non-zero.

The third example computes a cone whose axis is approximately in line with the X-axis using the points in the user summation array.

The fourth example computes a cone whose axis has direction vector approximately .7,0,.7 using the points in the internal summation array.

If P2 isn't specified, the first three points must be at one level on the conical surface and the second group of three points must be at a different level. Additional points should be distributed evenly over the measured surface. If P2 is specified then there's no requirement on point distribution other than taking them evenly over the measured surface.

Results

The current contents of the Feature Register move into Feature Stack Position # 1. The Feature Register stores the best-fit cone parameters. The included angle of the cone is the total angle between the sides of the cone at its apex in a plane containing the cone's axis. The Dimensional Array stores this included angle in the angularity slot, S(7,0). Feature form is computed as the sum of the maximum distances from the best-fit cone to points inside and outside the cone. The result is stored in the form deviation slot, S(13,4), of the Dimensional Array.

The Dimensional Array also stores:

- Direction cosines of the cone's axis in the S(14,0), S(15,0), and S(16,0),
- Coordinates of the apex of the cone in X, Y, and Z,
- Feature type (in S4), and
- Feature number (in S5).

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-040 The summation array does not contain at least six points.
- MLB-041 The points do not define a cone.
- MLB-042 The summations are invalid.
- MLB-043 A segmented measurement is in progress.

BM - Best-fit Summation

The BM command sums the specified point into the user or internal summation array.

Command Format

MLString "BM P1,P2,P3"

Examples

1. MLString "BM F8"
2. MLString "BM INTERNAL,POINT17"
3. MLString "BM INTERNAL,-7.5,12.325,5.0"
4. MLString "BM -7.5,12.325,5.0,0,0,1"

Input

- P1 = INTERNAL (String; optional).
- P2 = FR, F1, F8 or Feature Storage Name or X,Y,Z location (Parametric or numeric; optional).
- P3 = I, J, K orientation vector (Numeric; optional).

P1 can only be INTERNAL. This specifies that the system sums the point into the internal summation array. Omitting P1 causes this command to add the point to the user summation array.

P2 identifies the contents of the Feature Register (FR), one position of the Feature Stack (F1, F2, F3, etc.), or a Feature Storage name. In the first example, F8 indicates that the command use the contents of position 8 in the Feature Stack. In the second example, the name POINT17 refers to a previously saved feature. Omitting P2 causes the system to use the contents of the Feature Register. If P2 is the name of the feature stack position or of a saved feature, the point to be summed is the position data for that feature. If P2 is a set of three numeric values, these define the X, Y, and Z part coordinates of the point to be summed.

P3 is a set of three numeric values which define the I, J, and K components of the approximate surface normal at the specified point in part coordinates. If P3 is not specified then this vector defaults to (0,0,0).

Results

The part program sums the position coordinates of the feature in the P2 argument into the declared summation array. This command does not affect the contents of the Feature Register and Feature Stack.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 The Feature Stack, Feature Register, or Feature Storage does not contain the required feature.
- MLB-042 The summations are invalid.
- MLB-043 A segmented measurement is in progress.

BP - Best-fit Plane

The BP command computes the best-fit to a plane from the points in the selected summation array.

Command Format

MLString "BP P1,P2,P3"

Examples

1. MLString "BP"
2. MLString "BP INTERNAL,INT"
3. MLString "BP +X,1,0,0"
4. MLString "BP INTERNAL,.7,0,.7"

Input

The selected summation array must contain at least 3 points.

- **P1** = INTERNAL (String; optional).
- **P2** = X, -X, Y, -Y, Z, -Z, OFF, EXT, or INT (String; optional).
- **P3** = Set of three numbers (Numeric; optional).

P1 can only be INTERNAL. This specifies that the command uses the points found in the internal summation array. Omitting P1 causes the use of the points in the user summation array.

P2 specifies the desired probe compensation. If P2 is omitted, no probe compensation occurs. A specified axis should indicate the part axis that most nearly points in the direction away from the measured surface. The +Y in the example indicates that the positive end of the part's Y-axis points away from the measured surface.

The possible P2 arguments are:

- X,-X,+Y,-Y,+Z,-Z The specified axis points away from the measured surface.
- OFF No compensation.
- EXT The measured surface is facing away from the part datum.
- INT The measured surface is facing towards the part datum.

P3 is a set of three numbers defining the approximate orientation of the normal of the plane in the active part reference frame. This may define a unit vector but is not required. At least one of the three numbers must be non-zero. Example 4 computes a plane approximately perpendicular to the X-axis using the points in the user summation array. No probe compensation is requested in this example but the resultant plane normal will point generally in the +X-axis direction from the measured surface.

Results

The contents of the Feature Register move into the Feature Stack. The Feature Register stores the data for the plane. Feature form is computed as specified by the **FC** command (BFD: the sum of the maximum distances from the computed plane to points on opposite sides of the plane; MRS: the minimal separation between two parallel planes which contain the set of points). The results are stored in the form deviation slot, S(13,4), in the Dimensional Array. The Dimensional Array stores the direction cosines of the vector normal to the computed plane in S(14,0), S(15,0), and S(16,0).

The Dimensional Array also stores :

- The coordinates of the centroid of the plane,
- The feature type, and
- The feature number.

Skip Action

Not executed if part of a skip block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-040 The summation array does not contain at least three points.
- MLB-041 The points do not define a plane.
- MLB-042 The summations are invalid.
- MLB-043 A segmented measurement is in progress.

BS - Best-fit Sphere

The BS command uses the points in the summation array to compute the best-fit to a sphere. Use the **CM** command to specify compensation for the probe radius.

Command Format

MLString "BS P1"

Examples

1. MLString "BS INTERNAL"

Input

The summation array must contain at least 4 points.

- **P1** = INTERNAL (String; optional).

P1 can only be INTERNAL. This specifies that the command uses the points found in the internal summation array. The INTERNAL in the example indicates that the internal summation array holds the points used in the summation. Omitting P1 causes the command to use the points in the user summation array.

Results

The current contents of the Feature Register move into the Feature Stack. The Feature Register stores the sphere data. Feature form is computed as specified by the **FC** command (BFD: the range of the distances from the best-fit sphere center to the set of individual points; MRS: separation between the two concentric spheres of minimal separation which contain the set of points). The system stores the result in the form deviation slot, S(13,4), in the Dimensional Array. The Dimensional Array sets the direction cosines to zero in S(14,0), S(15,0) and S(16,0).

The Dimensional Array also stores:

- The center coordinates,
- The feature radius,
- The feature diameter,
- The feature type, and
- The feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-040 The summation array does not contain at least four points.
- MLB-041 The points do not define a sphere.
- MLB-042 The summations are invalid.
- MLB-043 A segmented measurement is in progress.

BT - Compute Best-fit Point to Set of Points

Compute a best-fit or centroid point to a previously accumulated set of points. At least one point must have been accumulated. The location of the best-fit point is computed and pushed into the feature stack. There is no compensation for probe size.

Command Format

MLString "BT p1"

Examples

1. MLString "BT"

   Compute a point using the points in the user summation array.

2. MLString "BT INTERNAL"

   Compute a point using the points in the internal summation array.

Input

The optional string parameter p1 can only be "INTERNAL". This specifies that the point data has been accumulated in the internal summation array. If not specified then the default is the user summation array.

Results

The average of the points in the specified array is computed. The resulting point feature is pushed into the feature register.

The feature orientation vector (I,J,K) is to (0,0,0). The feature form is set to 0 (zero). The feature number is stored in the feature register. The feature type is set to 1.

Skip Action

Push of null feature.

Error Messages

- MLB-040 - The summation array does not contain any points.
- MLB-042 - The specified summation array is invalid.
- MLB-043 - A segmented feature measurement is in progress.

BW - Best-fit Slot / Web

The BW command uses the points in the summation array to compute the best-fit slot or web in the working plane to a previously accumulated set of points. Use the CM command to specify compensation for the probe radius.

Command Format

MLString "BW *p1*,*p2*,*p3*"

Examples

1. MLString "BW"
2. MLString "BW INTERNAL,AP"
3. MLString "BW 0,1,0"
4. MLString "BW INTERNAL,.7,0,.7"

Input

The selected summation array must contain at least 3 points.

- *p1* = INTERNAL (String; optional).
- *p2* = AP (String; optional)
- *p3* = Set of three numbers (Numeric; optional).

p1 can only be INTERNAL. This specifies that this command uses the points found in the internal summation array. Omitting *p1* causes this command to use the points in the user summation array. Example 1 omits the *p1* argument, indicating that the command uses the points in the user summation array.

p2 can only be AP. This specifies that the slot/web is to be computed in the major plane (XY, YZ, or ZX) which is most parallel to the set of measured points rather than in the active working plane.

p3 is a set of three numbers defining the approximate orientation of the center line of the slot /web in the active part reference frame. This may define a unit vector but is not required. At least one of the two numbers corresponding to the plane of the feature must be non-zero. Omitting **p3** indicates the feature is approximately parallel to the first two points summed.

Results

The current contents of the Feature Register move into the Feature Stack. The Feature Register stores the feature data. Feature form is computed as the difference between the maximum and minimum distances from the computed slot / web centerline to the individual points. The feature size is returned to the radius and the diameter rows of the dimensional array. The third axis coordinate is the average of the third axis coordinates of the set of measured points.

The Dimensional Array also stores:

- The feature type in S4. Feature type is 20 (web), 21 (slot-no compensation) or 22(slot), depending on specified probe compensation, and
- The feature number in S5.

Skip Action

Not executed if part of a skip block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-040 The summation array does not contain at least three points.
- MLB-041 The points do not define a slot/web in the working plane.
- MLB-042 The summations are invalid.
- MLB-043 A segmented measurement is in progress.

BX - Best-fit Cparln Feature

The BX command uses the points in the summation array to compute the best-fit Cparlnf (Closed parallel lines - flat end) or Cparlnr (Closed parallel lines - round end) in the working plane to a previously accumulated set of points. These features are similar to the slot / web, except they have length as well as width. Unlike the slot / web, the location of the Cparln feature is at the center of its diameter and length. Only one point from each end is allowed / required. Use the CM command to specify compensation for the probe radius.

Command Format

MLString "BX P1,P2,P3,P4"

Examples

1. MLString "BX"
2. MLString "BX INTERNAL,AP"
3. MLString "BX R,0,1,0"
4. MLString "BX INTERNAL,R,.7,0,.7"

Input

The selected summation array must contain at least 5 points.

- **P1** = INTERNAL (String; optional).
- **P2** = AP (String; optional)
- **P3** = R (String; optional)
- **P4** = Set of three numbers (Numeric; optional).

P1 can only be INTERNAL. This specifies that this command uses the points found in the internal summation array. Omitting P1 causes this command to use the points in the user summation array. Example 1 omits the P1 argument, indicating that the command uses the points in the user summation array.

P2 can only be AP. This specifies that the Cparln is to be computed in the major plane (XY, YZ, or ZX) which is most parallel to the set of measured points rather than in the active working plane.

P3 can only be R. This specifies that the Cparln has round (semi-circle) ends, and will generate a feature of type Cparlnr. If this parameter is omitted, flat ends are assumed, generating a feature of type Cparlnf.

P4 is a set of three numbers defining the approximate orientation of the center line of the Cparln in the active part reference frame. This may define a unit vector but is not required. At least one of the two numbers corresponding to the plane of the feature must be non-zero. Omitting P3 indicates the feature is approximately parallel to the first two points summed.

Results

The current contents of the Feature Register move into the Feature Stack. The Feature Register stores the feature data. Feature form is computed as the difference between the maximum and minimum distances from the computed slot/web centerline to the individual points (the end points are deliberately omitted from this calculation). The feature size is returned to the radius and the diameter rows of the dimensional array. The third axis coordinate is the average of the third axis coordinates of the set of measured points. The orientation vector will be returned to the I, J, and K rows of the dimensional array. The normal vector will be returned in S variables SI2, SJ2, and SK2.

The Dimensional Array also stores:

- The feature type in S4. Feature type for Cparlnf is 23 (OD), 24 (no compensation) or 25 (ID), depending on specified probe compensation. Feature type for Cparlnr is 26 (OD), 27 (no compensation) or 28 (ID), depending on specified probe compensation, and
- The feature number in S5.

Note: The order of measured points will be reordered such that the two end points will appear at the end of the list.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-040 The summation array does not contain at least five points.
- MLB-041 The points do not define a Cparln feature in the working plane.
- MLB-042 The summations are invalid.
- MLB-043 A segmented measurement is in progress.

CB - Clear Summation Array

The CB command clears and initializes the summation array. Use this command before initial use of the array and in preparation for collecting a new set of data. It can also be used to discard the set of points used to compute the feature.

Command Format

MLString "CB P1"

Examples

1. MLString "CB INTERNAL"

2. MLString "CB"
3. MLString "CB _CIRCLE5"

Input

- **P1** = INTERNAL (String; optional).

P1 can be INTERNAL or it may be the name of a saved feature or a stack position. INTERNAL specifies that the system is to clear the internal summation array. Omitting P1 causes the system to clear the user summation array. In the first example, INTERNAL indicates that the system clears the internal summation array. Since there is no P1 argument in the second example, the system clears the user summation array. If P1 specifies a feature then any point data saved when that feature was computed is discarded.

Results

The specified summation array is cleared or the specified set of data points is discarded. This command does not affect the Feature Register or Feature Stack.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-043 A segmented measurement is in progress.

FF - Fetch Form Feature Data

Pushes data onto the feature stack corresponding to that used to compute form for the last best-fit feature. Different algorithms may be used to compute feature form than are used to compute best-fit feature data (e.g., see FC and CM commands). The position, orientation, size, and form data used to compute form for the last best-fit feature are pushed into the feature stack.

Command Format

MLString "FF"

Example

1. MLString "FF" - push feature data onto the feature stack as used to compute form for the most recently computed multi-point (best-fit) feature.

Input

There are no input parameters for this command.

Results

A new feature is created using data as used to compute form for the most recently computed multi-point (best-fit) feature. Feature type is set to the basic (uncompensated) feature type. Feature size, orientation, and/or position reflects the center of the form zone. Data is not corrected for probe size.

If no best-fit feature has been computed since the last part program initialization (see IL command), a null feature is pushed onto the feature stack.

NOTE: If this command is used only for temporary retrieval and use of this data it should be followed by a 'PO D' command in order to restore the feature stack to its previous state.

Skip Action

Push of null feature.

Error Messages

None.

HL - High/Low Point Retrieval

The HL command is used to retrieve the high and low points determined when best-fit feature form is computed.

Command Format

MLString "HL"

Examples

1. MLString "HL"

Input

The **HL** command returns the extreme points determined while computing feature form for the feature in the feature register. The **HL** command should be executed immediately after feature measurement or construction.

Results

The two extreme points are pushed into the Feature Stack--first the "low" point and then the "high" point. For example, if the feature is a circle (or a sphere), the low point is the point closest to the center point. The high point is the point furthest from the center point. For a 3D line, both points are the same, corresponding to the point furthest from the line.

Skip Action

Not executed if part of a skip block but two null features are pushed onto the Feature Register.

Error Messages

- MLB-042 The point data is no longer available for the feature in the feature register.

QB - Query Best-fit Summation Status

Query the best-fit arrays to determine whether point data for a feature is available.

NOTE: With MeasureMax+ 6.00 and later this command is effectively obsolete since the point data is saved with each feature and is always available unless it has specifically been deleted using the **CB** command.

Command Format

MLString "QB"

Examples

1. MLString "QB"
   Return feature numbers for features last solved using data in the summation arrays.

Input

None.

Results

The feature numbers corresponding to features computed from point data in the two summation arrays are returned to variables S7 (user array) and S8 (internal array). If the value returned for an array is zero then either the summation array no longer contains valid data or no feature has yet been computed using that data.

Skip Action

Executed normally.

Error Messages

None.

RB - Recall Best-Fit Point

Recalls data from one of the best-fit summation arrays or from a saved/stacked feature.

Command Format

MLString "RB P1,P2,P3"

Examples

1. MLString "RB"

   Return the number of points in the user summation array.

2. MLString "RB 15,INTERNAL"

   Recall data for point number 15 in the internal summation array and push it into the feature register.

3. MLString "RB -15,_LINE17"

   Recall data for point number 15 saved with feature _LINE17 and return the X, Y, Z coordinates into variables S7, S8, and S10.

4. MLString "RB ALL,_CIRCLE5"

   Return the data for all points saved with feature _CIRCLE5 in array S9 and the number of points to variable S7.

Input

- **P1** = Point number (Numeric; optional).
- **P2** = INTERNAL or feature identifier (String; optional).
- **P3** = C (String; optional).

The optional alphanumeric parameter p1 is either the index of the point to be recalled or the string 'ALL' indicating that all point data is to be returned to the S9 array. If p1 is not specified then the number of points is returned. If p1 is specified as a positive number then the point data is pushed onto the feature stack as a new feature. If p1 is specified as a negative number then the point data is simply returned to the variables as described below.

The optional string parameter p2 can be "INTERNAL" which specifies that the point data is to be retrieved from the internal summation array or a feature identifier (FR, F1, ..., F8 or the name of a saved feature). If not specified then the point data is retrieved from the user summation array.

The optional string parameter p3 may only be "C" and may only be used when p2 is the identifier of a feature in the stack or which has been previously saved. If p3 is specified then the point data is compensated for the probe radius to return a surface point. If p3 is not specified then the point data returned is uncompensated probe center data.

Results

If p1 is a positive number the data for the specified point is recalled and pushed into the feature register. If p1 is a negative number then the data for the point indexed by the absolute value of p1 is recalled and stored into variables S7, S8 and S10. If p1 is 'ALL', then data for all saved points (up to the maximum that can be stored in array S9) is stored in array S9 and the number of points stored is returned to variable S7. If p1 is not specified then the number of points is returned to variable S7.

Note that whenever a feature is computed as a best-fit to a set of points, the points are saved with the computed feature and are available for later recall as described above unless the feature has been deleted or the set of points has been cleared using the **CB** command.

Skip Action

If p1 is positive then a null feature is pushed into the feature register; otherwise ignored.

Error Messages

- MLB-072 The specified feature does not exist
- MLB-140 The specified point is not available

UM - Unsum (delete) a Point from the Summation Array

The UM command causes a point to be deleted from one of the summation arrays.

Command Format

MLString "UM P1,P2"

Examples

1. MLString "UM 48"
2. MLString "UM 15,INTERNAL"
3. MLString "UM INTERNAL"

Input

- **P1** = Index of point to delete (Numeric; optional).
- **P2** = INTERNAL (String; optional)

P1 is index of the point to delete. If it is not specified then the most recently recorded point is to be deleted.

P2 can only be the string "INTERNAL". If it is specified then the point is deleted from the internal array, otherwise it is deleted form the user array.

In the first example point number 48 is to be deleted. The second example deletes point 15 from the internal summation array. The third example deletes the most recently recorded point from the internal summation array.

Results

The designated point is deleted from the specified best-fit array. After deletion, the summation data is recomputed to reflect the remaining points. The point number specified corresponds to the position of the point data in the current recorded point list. Deletion of a point other than the last will effectively renumber all points later in the list.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

10. Feature Construction Group

MLB commands in the Feature Construction Group construct a feature from other features.

The Feature Construction Group includes the following MLB commands.

CC	**Compute Circle Tangent to Two Lines**
FI	**Feature Intersection** computes the point of intersection of two circles, a circle and a line, or a cylinder and a line.
KK	**Cone Cross-section** computes the circular cross-section of a cone.
MV	**Move Point** moves a feature in the Feature Register a specific point.
NL	**Normal Line** constructs a line using features in its function command string.
NP	**Normal Plane** constructs a plane using features in its function command string.
OL	**Offset Line**
PL	**Parallel Line** constructs a line using features in its function command string.
PP	**Parallel Plane** constructs a plane using features in its function command string.
PT	**Point** constructs a point using features in its function command string.
WB	**Construct Slot/Web** computes the parameters of a slot or web in the active working plane.

CC - Compute Circle Tangent to Two Lines or Other Features

The CC command computes the circle which lies in the midplane of two given lines, is tangent to both lines (as projected onto the midplane), is of a specified size and (of the four possible such circles) is located closest to the specified location. This command may also compute a circle tangent to a given line and circle, to two given circles, or to a given circle and a given point. All computations given a circle as one feature are assumed to be in the plane of the circle (or P5 in the case of 2 circles).

Command Format

MLString "CC P1,P2,P3,P4,P5,P6"

Examples

1. MLString "CC 13.7,-2.35,0,15.75"

2. MLString "CC 13.7,-2.35,0,15.75,_LINE3,_LINE17"

Input

- **P1**, **P2**, **P3** = the nominal XYZ position of the desired circle (Numeric; required).
- **P4** = Diameter of circle (Numeric; required).
- **P5**, **P6** = F1 to F8 or Feature Name (String/parametric; optional).

P1, P2, P3 together describes the nominal location of the desired circle.

P4 defines the diameter of the desired circle. This must be a positive number.

P5 and P6 specifies the two lines or additional features. P5 and P6 may specify a feature stack location or the name of a saved feature. If P5 and P6 are not specified, then they defaults to "FR" and "F1". The features P5 and P6 must be two lines, a line and a circle, two circles, or a circle and a point. If one of these features is a circle, and the features are so far apart that no circle of the specified size is tangent to both, then the size of the nominal circle is adjusted to the minimal value to allow such tangency.

Results

A feature of type circle is computed and pushed into the feature register. Data for the circle defined by the input parameters is computed and stored in the Feature Register.

The data placed in the Feature Register includes:

- The center point,
- Radius, and
- Diameter.

Feature form is set to zero. The feature type is circle (5). The orientation of the midplane of the two lines is stored as the orientation (I, J, K) of the circle.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The referenced features are not defined.
- MLB-032 P5 or P6 identifies a feature which is not a line.
- MLB-041 A circle satisfying the constraints cannot be computed. This may occur if the two lines are parallel.

FI - Feature Intersection

The FI command computes the point of intersection of two circles, a line and a circle, a line and a cylinder, or a line and a cone.

Command Format

MLString "FI P1,P2,P3"

Examples

1. MLString "FI ID3,OD6,+X"

Input

- **P1** = Feature in FR, Stack or Storage (Parametric/string; optional).
- **P2** = Feature in FR, Stack or Storage (Parametric/string; optional).
- **P3** = X, -X, Y, -Y, Z, -Z (String; optional).

P1 and P2 identify the two features that intersect. These can be in the Feature Register, Feature Stack or Feature Storage. In the example, ID3 and OD6 refer to features in Feature Storage. If you omit P1 and P2, the command uses the data from FR and F1.

P3 must identify a signed axis as +X, -X, +Y, -Y, +Z or -Z that distinguishes between the two possible intersection points. The selected point of intersection is the one furthest in the direction specified by P3. If P3 is -X, the command computes and stores the point of intersection with the most negative X-coordinate. In the example, the +X distinguishes between the two points of intersection. The command stores the point that has the most positive X-axis coordinate.

Note: If the requested intersection is of two circles or of a line with a circle then the intersection is computed in the active working plane. In this case the specified axis (P3) must correspond to one of the working plane axes.

Results

The current contents of the Feature Register move into the Feature Stack. The Feature Register stores the specified point of intersection. If the two specified features do not intersect, the Feature Register stores the point of closest approach of the first feature to the second instead. Feature form equals zero in the form deviation slot, S(13,4), of the Dimensional Array.

The Dimensional Array stores:

- Coordinates of the point in X, Y and Z,
- Feature type (point, 1), and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-021 The first parameter is invalid.
- MLB-023 The third parameter is invalid.
- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-031 The first parameter specifies an invalid type.
- MLB-032 The second parameter specifies an invalid type.
- MLB-041 The points do not define the feature.

KK - Cone or Sphere Cross Section

The KK command computes the circular cross-section of a cone or sphere. The circular cross-section of a cone is defined by:

- Intersection of a plane with the cone or sphere,
- Offset from a major datum plane,
- Offset of the cross-section from the apex or center of the sphere, or
- Radius of the cross-section.

Command Format

MLString "KK P1,P2,P3"

Examples

1. MLString "KK I,XY"
2. MLString "KK 0,10.5,CONE15"
3. MLString "KK R,17.5,F2"
4. MLString "KK Z,-15"

Input

- **P1** = I, O, R, X, Y, or Z (String; required).
- **P2** = Radius at cross section (Numeric/parametric; required).
- **P3** = F1 to F8 or Feature Name (String/parametric; optional).

P1 describes the type of cross-section desired and the meaning of P2. P1 takes one of four forms:

- The letter "I". If P1 is "I", the desired cross-section is the intersection of the plane P2 with the cone or sphere. P2 is a stack location, the name of a saved feature, or the name of one of the working planes. The feature specified by P2 must be a plane. The center of the circle is the intersection of the axis of the cone with the specified plane or the point on the plane closest to the center of the sphere. The radius and diameter of the circle are the size of the circular cross-section of the cone or sphere at the center point.
- The letter "O". If P1 is "O", the desired cross-section is at an offset from the apex of the cone or the center of the sphere. The offset is specified in P2. P2 must be a positive distance in the current measurement units for a cone and may be either position or negative for a sphere. If the feature is a sphere then the computed center point is in a plane parallel to the active working plane and distance P2 from the sphere center in the third axis direction.
- The letter "R". If P1 is "R", the desired cross-section is at the point where the cross-section of the cone or sphere has the radius specified in P2. P2 must be a positive value in the current measurement units. If the feature is a sphere then the computed center point is in a plane parallel to the active working plane and located in the positive third axis direction from the sphere center.
- The letter "X", "Y", or "Z". If P1 is X, Y, or Z, the desired cross-section is at a distance of P2 from the datum plane perpendicular to the specified axis (X, Y, or Z). When P1 is X, Y, or Z, P2 may be positive or negative.

The first example shows the "I" form. This example computes the circle defined by the intersection of the XY plane with the cone in FR.

The second example shows the "O" form. This example computes the circle at an offset of 10.5 units from the apex of the cone saved as feature "CONE15."

The third example shows the "R" form. This example computes the circular cross-section of the cone in stack position F2 which has a radius of 17.5 units.

The fourth example shows the X/Y/Z form. This example computes the circular cross-section of the cone in FR at a distance of 15 below the XY datum plane.

P3 specifies the cone or sphere. P3 may specify a feature stack location or the name of a saved feature. If P3 is not specified, it defaults to "FR".

Results

A feature of type circle is computed and pushed into the feature register. Data for the circular cross-section defined by the input parameters is computed and stored in the Feature Register.

The data placed in the Feature Register includes:

- The center point,
- Radius, and
- Diameter.

Feature form is set to zero. The feature type is circle, O.D. (4), circle (5), or circle, I.D. (6) depending on the feature type of the cone or sphere. The orientation of the cone or intersecting plane for spheres is stored as the orientation (I, J, K) of the circle.

The offset between the cross-section and the cone apex or sphere center is computed and stored in the distance actual column in the host data array. The offset value is not saved with the other data for the circle if it is saved to named feature storage.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register does not contain the required features.
- MLB-032 The second parameter specifies an invalid type.
- MLB-033 The third parameter specifies an invalid type. Or, the third parameter is not specified and the feature register does not contain a cone or sphere.
- MLB-041 The points do not define the feature. This can happen if the given plane does not intersect the sphere, or the specified distance/radius is greater than the sphere radius.

MV - Move Point

The MV command moves the feature in the Feature Register a specific distance. The values defined by the arguments are added to the respective X-, Y-, and Z-coordinates of the feature in the Feature Register. After the system moves the feature, it gives the feature a new feature number.

Command Format

MLString "MV P1,P2,P3"

Examples

1. MLString "MV S-1.2"
2. MLString "MV RY4"
3. MLString "MV 0,PR,0"

Input

- **P1** = Direction (Numeric/parametric; required).
- **P2** = Direction (Numeric/parametric; required if P1 is a signed number, "PR" or "NPR").
- **P3** = Direction (Numeric/parametric; required if P1 is a signed number, "PR" or "NPR").

P1 is required and specifies the type of change to be made to the coordinates of the feature:

- If P1 is the letter 'S' followed by a signed number or parametric variable, then the feature position is to be moved in the active working plane perpendicular to the polar radius line (the sign of the value corresponds to the sign of the polar angle direction).
- If P1 is the letter 'R' followed by a signed number or parametric variable, then the feature position is to be moved in the active working plane along the polar radius line (positive is away from the datum point, negative is towards it.
- If P1 does not begin with the letter 'R' or 'S' then it must be a signed number or parametric variable specifying the amount of change for the X-axis coordinate. In this case P2 and P3 must also be specified as signed numbers or parametric variables, specifying the amount of change for the Y and Z axis coordinates respectively.

The first example shows the S form. The feature is moved by 1.2 units perpendicular to the polar radius line in the positive polar angle direction.

The second example shows the R form. The feature is moved in the working plane towards or away from the datum point by the amount defined by Y4, the Y coordinate of the feature in the fourth position in the Feature Stack.

The third example shows the XYZ form. The first 0 indicates that the X-coordinate does not change, the PR indicates that a value equal to the active probe radius is added to the Y- coordinate, and the second 0 indicates that the Z- coordinate does not change.

Results

The position coordinates of the feature in the feature register are updated as specified. The feature number increases by one. The Dimensional Array reflects the changed position information.

Skip Action

The feature number is incremented. The feature number for the feature in the feature register is incremented and its feature name is cleared.

Error Messages

- MLB-030 The Feature Register does not contain any feature or the parametric argument refers to an empty stack position.

NL - Normal Line

The NL command uses the features specified in the command string arguments to construct a line.

Command Format

MLString "NL P1,P2"

Examples

1. MLString "NL LN1,DP"

Input

- **P1** = Feature in FR, Stack or Storage (Parametric; optional).
- **P2** = Feature in FR, Stack or Storage (Parametric; optional).

P1 and P2 can identify features in the Feature Register, Feature Stack or Feature Storage. P1 and P2 can be any measured or constructed feature. P2 may refer to a reference frame axis (XA, YA or ZA), a reference frame plane (XY, YZ, or ZX), or the reference frame datum point (DP). If you omit P1 and P2, the command uses the data from FR and F1 by default.

In the example, the LN1 is the label of a feature saved in Feature Storage. If LN1 is a line, the command constructs a line through the datum point and perpendicular to the line.

Results

A line is constructed and pushed into the Feature Register/Stack based upon the types of the two input features as follows:

- If the two features are point-type features (points, circles, ellipses, or spheres) then the line containing the two feature positions is constructed.
- If one of the features is a point-type feature and the other is a line-type feature (line, cylinder, cone, slot, or web) then the line through the point-type feature position and normal (perpendicular) to the line-type feature is constructed.
- If one of the features is a point-type feature and the other is a plane then the line through the position of the point-type feature and normal (perpendicular) to the plane is constructed.
- If both of the features are line-type features then the line through the midpoint of the two lines and normal (perpendicular) to both lines is constructed.

The current contents of the Feature Register move down. The Feature Register stores the constructed line. Feature form equals 0 in the form deviation slot, S(13,4), of the Dimensional Array.

The Dimensional Array stores:

- Coordinates of a point on the line,
- Direction cosines describing the orientation of the line,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-032 The second parameter specifies an invalid type.
- MLB-041 The points do not define the feature.

NP - Normal Plane

The NP command uses the features specified in the command string to construct a plane.

Command Format

MLString "NP P1,P2"

Examples

1. MLString "NP F5,PT7"

Input

- **P1** = Feature in FR, Stack or Storage (Parametric; optional).
- **P2** = Feature in FR, Stack or Storage (Parametric; optional).

P1 and P2 can identify features in the Feature Register, Feature Stack or Feature Storage. P1 and P2 can be any measured or constructed feature. P2 may refer to a reference frame axis (XA, YA or ZA), a reference frame plane (XY, YZ, or ZX), or the reference frame datum point (DP). If both P1 and P2 are omitted, the command uses the data from FR and F1 by default.

In the example, the F5 refers to the feature in the fifth position in the Feature Stack. PT7 is the label of a feature saved in Feature Storage.

Results

A plane is constructed and pushed into the Feature Register/Stack based upon the types of the two input features as follows:

- If the two features are point-type features (points, circles, ellipses, or spheres) then the plane normal (perpendicular) to the line containing the two feature positions is constructed.
- If one of the features is a point-type feature and the other is a line-type feature (line, cylinder, cone, slot, or web) then the plane through the point-type feature position and normal (perpendicular) to the line-type feature is constructed.
- If one of the features is a line-type feature and the other is a plane then the plane containing the line and normal (perpendicular) to the plane is constructed.
- If both of the features are line-type features then the plane normal (perpendicular) to the shortest line between the lines and passing through its midpoint is constructed.

The current contents of the Feature Register move down. The Feature Register stores the constructed plane. Feature form equals 0 in the form deviation slot, S(13,4), of the Dimensional Array.

The Dimensional Array stores:

- Coordinates of a point on the plane,
- Direction cosines describing the orientation of the plane,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-032 The second parameter specifies an invalid type.
- MLB-041 The points do not define the feature.

OL - Offset Line

The OL command computes the parameters for the line in the active plane described by the point in the Feature Register and the top point in the Feature Stack or from two points in a specified summation array. An offset line is a surface made up of two pads at two different levels. The location and orientation of the line are computed and pushed into the feature stack. The location is compensated for probe size if so specified.

Command Format

MLString "OL P1,P2,P3,P4,P5"

Examples

1. MLString "OL INTERNAL,-Y,65.524,36.666,+Y"

Input

- **P1** = USER or INTERNAL (String; optional)
- **P2** = X, -X, Y, -Y, Z, -Z, INT, EXT, or OFF (String; optional).
- **P3** = Offset from first measured point to nominal line (Numeric; required).
- **P4** = Offset from second measured point to nominal line (Numeric; required).
- **P5** = X, -X, Y, -Y, Z, -Z (String; required).

P1 specifies the location of the point data used for the computation. If P1 is omitted, the command uses data from the stack. If P1 is USER, the command uses data from the user summation array. If P1 is INTERNAL, the command uses data from the internal summation array. In the example, INTERNAL means that point data has been summed into the internal summation array.

P2 specifies the desired probe compensation. Omitting P2 means no probe compensation occurs. A specified axis (which must not be the axis normal to the active plane) indicates the part axis that most nearly points in the direction away from the measured surface. The -Y in the example specifies that probe compensation occurs based upon the probe being of the negative Y side of the surface.

The possible P2 arguments are:

- X,-X,+Y,-Y,+Z,-Z The specified axis points away from the measured surface.
- OFF No compensation.
- EXT The measured surface is on the side of the plane opposite the part datum.
- INT The measured surface is facing the part datum.

P3 states the offset from the first point to the nominal plane. In the example, the 65.524 is the amount of offset from the first point to the nominal plane.

P4 states the offset from the second point to the nominal plane. In the example, the 36.666 is the amount of offset from the second point to the nominal plane.

P5 indicates the plus direction for the offset values of P3 and P4. It identifies a part axis (X, -X, Y, -Y, Z, or -Z) that points from the nominal line toward the measured points. Note that this axis must not be the axis normal to the active plane. In the example, the +Y indicates that the plus end of the Y-axis points from the nominal line toward the measured points.

Results

The system computes the data for the line. The current contents of the Feature Register move down to Feature Stack Position #1. The Feature Register stores the line data. The computed line position is modified to reflect any requested probe compensation. Form is set to zero and stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the orientation vector of the computed line in S(14,0), S(15,0) and S(16,0).

The Dimensional Array also stores:

- Feature type
- Feature number

Skip Action

Not executed if part of a skip block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-033 One of the feature positions contains a plane.
- MLB-040 The summation array does not contain at least two points.
- MLB-041 The points and offsets do not define a line in the active plane.

PL - Parallel Line

The PL command uses the features specified in the command string arguments to construct a line.

Command Format

MLString "PL P1,P2"

Examples

1. MLString "PL F2,LN3"

Input

- **P1** = Feature in FR, Stack or Storage (Parametric; optional).
- **P2** = Feature in FR, Stack or Storage (Parametric; optional).

P1 and P2 can identify features in the Feature Register, Feature Stack or Feature Storage. P1 and P2 can be any measured or constructed feature. P2 may refer to a reference frame axis (XA, YA or ZA), a reference frame plane (XY, YZ, or ZX), or the reference frame datum point (DP). If both P1 and P2 are omitted, the command uses the data from FR and F1 by default.

In the example, F2 refers to the feature in the second position in the Feature Stack and LN3 is the label of a feature saved in Feature Storage.

Results

A line is constructed and pushed into the Feature Register/Stack based upon the types of the two input features as follows:

- If one of the features is a point-type (points, circles, ellipses, or spheres) feature and the other is a line-type feature (line, cylinder, cone, slot, or web) then the line through the point-type feature position and parallel to the line-type feature is constructed.
- If one of the features is a line-type feature and the other is a plane then the line formed by projecting the line into the plane is constructed.
- If the two features are planes then the line of intersection of the two planes is constructed.
- If the two features are line-type features then the midline of the two lines is constructed.

The current contents of the Feature Register move down. The Feature Register stores the constructed line. Feature form equals 0 in the form deviation slot, S(13,4), of the Dimensional Array.

The Dimensional Array stores:

- The coordinates of a point on the line,
- The direction cosines describing the orientation of the line,
- The feature type, and
- The feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-032 The second parameter specifies an invalid type.
- MLB-041 The points do not define the feature.

PP - Parallel Plane

The PP command uses the features specified in the command string arguments to construct a plane.

Command Format

MLString "PP P1,P2"

Examples

1. MLString "PP F2,LN3"

Input

- **P1** = Feature in FR, Stack or Storage (Parametric; optional).
- **P2** = Feature in FR, Stack or Storage (Parametric; optional).

P1 and P2 can identify features in the Feature Register, Feature Stack or Feature Storage. P1 and P2 can be any measured or constructed feature. P2 may refer to a reference frame axis (XA, YA or ZA), a reference frame plane (XY, YZ, or ZX), or the reference frame datum point (DP). If both P1 and P2 are omitted, the command uses the data from FR and F1 by default.

In the example, F2 refers to the feature in the second position in the Feature Stack and LN3 is the label of a feature saved in Feature Storage.

Results

A plane is constructed and pushed into the Feature Register/Stack based upon the types of the two input features as follows:

- If one of the features is a point-type (points, circles, ellipses, or spheres) feature and the other is a line-type feature (line, cylinder, cone, slot, or web) then the plane containing the two features is constructed.
- If one of the features is a point-type feature and the other is a plane then the plane containing the point and parallel to the plane is constructed.
- If the two features are both line-type features then the plane containing the first line and parallel to second is constructed.
- If the two features are both planes then the midplane of the two planes is constructed.

The current contents of the Feature Register move down. The Feature Register stores the constructed plane. Feature form equals 0 in the form deviation slot, S(13,4), of the Dimensional Array.

The Dimensional Array stores:

- Coordinates of a point on the plane,
- Direction cosines describing the orientation of the plane,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-032 The second parameter specifies an invalid type.

- MLB-041 A plane cannot be uniquely computed from the specified features. (For example, if two identical lines are specified, every plane containing the line satisfies the conditions described above.)

PT - Point

The PT command uses the features specified in the command string arguments to construct a point.

Command Format

MLString "PT P1,P2"

Examples

1. MLString "PT F4,YA"

Input

- **P1** = Feature in FR, Stack or Storage (Parametric; optional).
- **P2** = Feature in FR, Stack or Storage (Parametric; optional).

P1 and P2 can identify features in the Feature Register, Feature Stack or Feature Storage. P1 and P2 can be any measured or constructed feature. P2 may refer to a reference frame axis (XA, YA or ZA), a reference frame plane (XY, YZ, or ZX), or the reference frame datum point (DP). If both P1 and P2 are omitted, the command uses the data from FR and F1 by default.

In the example, F4 refers to the feature in the fourth position in the Feature Stack and YA refers to the Y axis of the part reference frame.

Results

A point is constructed and pushed into the Feature Register/Stack based upon the types of the two input features as follows:

- If the two features are both point-type (points, circles, ellipses, or spheres) features then the midpoint of the two features is constructed.
- If one of the features is a point-type feature and the other is a line-type feature (line, cylinder, cone, slot, or web) then the point on the line closest to the point is constructed.
- If one of the features is a point-type feature and the other is a plane then the point on the plane closest to the point is constructed.
- If the two features are both line-type features then the midpoint of the shortest line between the two features is constructed.
- If one of the features is a line-type feature and the other is a plane then the point of intersection of the line with the plane is constructed.

The current contents of the Feature Register move down. The Feature Register stores the constructed point. Feature form equals 0 in the form deviation slot, S(13,4), of the Dimensional Array.

The Dimensional Array stores:

- Coordinates of the point,

- Direction cosines are set to zero,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-032 The second parameter specifies an invalid type.
- MLB-041 A point cannot be uniquely computed from the specified features. (For example, if a line and plane are specified and the line is parallel to the plane, then no intersection point exists.)

WB - Construct Slot / Web

Construct Slot/Web computes the parameters of a slot or web in the active working plane as defined by two specified features: a point and a line or two lines.

Command Format

MLString "WB P1,P2"

Examples

1. MLString "WB F2,LN1"

Input

- **P1** = Feature in FR, Stack or Storage (Parametric; optional).
- **P2** = Feature in FR, Stack or Storage (Parametric; optional).

P1 and P2 identify the two input features to be used in constructing the slot or web. Both features may be lines or one of the features may be a point while the other feature is a line.

P1 and P2 may be parametric variables which identify feature register or stack positions, or they may be names of features (points and lines) in named feature storage. The default values for P1 and P2 are FR and F1, respectively. P2 may identify an axis of the current part reference frame (XA, YA, or ZA) or the datum point DP.

The **CM** command executed before the WB command specifies what probe compensation (if any) is to be performed.

Results

Data for the slot/web is computed. The current contents of the Feature Register move down to Feature Stack Position #1. The Feature Register stores the slot/web feature data. The computed size is modified to reflect any requested probe compensation. Form is set to zero and stored in the form deviation slot, S(13,4), of the Dimensional Array. The Dimensional Array stores the direction cosines of the computed center line in the S(14,0), S(15,0) and S(16,0).

The Dimensional Array also stores:

- The position of the center line of the feature,
- The feature size (radius and diameter),
- The feature type, and
- The feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register does not contain the required features.
- MLB-031 The first feature is an invalid type.
- MLB-032 The second feature is an invalid type.

11. Geometric Evaluations Group

MLB commands in the Geometric Evaluations Group calculates angular and distance relationships between features.

The Geometric Evaluations Group includes the following MLB commands.

A1	**ANSI Angularity** computes the ANSI angularity between a best-fit feature and a specified datum axis or plane.
A2	**ANSI Parallelism** computes the ANSI parallelism between a best-fit feature and a specified datum axis or plane.
A3	**ANSI Perpendicularity** computes the ANSI perpendicularity between a best-fit feature and a specified datum axis or plane.
A4	**ANSI Runout** computes the ANSI total runout between a best-fit feature and a specified datum axis.
A5	**ANSI / ISO / DIN Positional Deviation** uses the best-fit points to compute positional deviation of a point, circle, ellipse, sphere, line, cylinder and cone.
A6	**ANSI / ISO / DIN Concentricity / Coaxiality** uses best-fit points to computer concentricity or coaxiality deviation of a point, circle, ellipse, sphere, cylinder, or cone
A7	**ANSI / ISO / DIN Circular Runout** uses best-fit points to compute runout deviation for circles, ellipses, cylinders, planes, and cones.
A8	**ANSI / ISO / DIN Symmetry Deviation** computes symmetry deviation for the feature in the Feature Register.
AI	**Included Angle** computes the obtuse or acute angle between two features, including cones, cylinders, slots and webs.
AN	**Angularity** computes the angle between lines and planes.
CV	**Changing Sign or Orientation Vector**
DA	**Axial Distance** computes the three-dimensional distance between two points, two lines, two planes, a line and a plane, a point and a line, or a point and a plane. It also derives the axial components of the computed distance and pushes these as a 'distance' feature onto the feature stack.
DS	**Distance** computes the three-dimensional distance between two features.
FC	**Form Calculation Method**
I0	**ANSI / ISO / DIN Form Reporting** identifies and reports the form of the feature in the Feature Register.
I1	**ANSI / ISO / DIN Angularity Reporting** computes angularity deviation as a width of a zone containing the feature in the Feature Register and prints a report showing this result.

I2	**ANSI / ISO / DIN Parallelism Reporting** computes parallelism deviation as a width of a zone containing the feature in the Feature Register and prints a report showing this result.
I3	**ANSI / ISO / DIN Perpendicularity Reporting** computes perpendicularity as a width of a zone containing the feature in the Feature Register and prints a report showing this result.
I4	**ANSI / ISO / DIN Total Runout Deviation Reporting** computes total runout deviation between the best-fit feature in the Feature Register and a datum axis and prints a report showing the result.
I5	**ANSI / ISO / DIN Position Deviation Reporting** computes and reports position deviation for the feature in the Feature Register.
I6	**ANSI / ISO / DIN Concentricity / Coaxiality Reporting** computes and reports concentricity or coaxiality deviation for a feature.
I7	**ANSI / ISO / DIN Circular Runout Deviation Reporting** computes and reports circular runout deviation for the feature in the Feature Register.
I8	**ANSI / ISO / DIN Symmetry Deviation Reporting** computes and reports symmetry deviation for the feature in the Feature Register.
JA	**Projected Angularity** projects two lines onto a selected plane and computes the angle between them.
JD	**Projected Distance** projects two features onto a plane and computes the distance between them.
JI	**Projected Included Angle** projects two features onto the selected plane and computes the included angle between them.
JL	**Projected Parallelism** projects two lines onto the selected plane and computes the deviation from parallelism between them.
JP	**Projected Perpendicularity** projects two lines onto the selected plane and computes the deviation from perpendicularity between them.
JX	**Projected Axial Distance** projects two features onto the selected plane before computing the distance between them. It also derives the axial components of the computed distance.
PA	**Parallelism** computes the deviation from parallelism between two fedatues.
PE	**Perpendicularity** computes the deviation from perpendicularity between two features.

A1 - ANSI Angularity

The A1 command computes angularity deviation as a width of a zone containing the feature in the feature register.

Command Format

MLString "A1 **p1**,**p2**,**p3**"

Example

1. MLString "A1 XA,30,ZX"

   The feature in the feature register is to be checked for conformance to a basic angle of 30 degrees relative to the X-axis when viewed in the ZX plane of the active reference frame.

2. MLString "A1 ZX,15"

   The feature is to be checked for conformance to a basic angle of 15 degrees to the ZX plane of the active reference frame.

Input

The Feature Register must contain a best-fit feature that is either:

- Line,
- Plane,
- Cone, or
- Cylinder.

The points used to derive this feature must still be available.

- **p1** = XA, YA, ZA or XY, YZ, ZX, or the name of a saved feature. (Parametric or string; required).
- **p2** = Nominal angle (Numeric; required).
- **p3** = XY, YZ, or ZX (Parametric; optional).

p1 identifies the datum feature. This is either a parametric feature identifier (XA, YA, ZA, XY, YZ, or ZX) identifying a datum axis or a datum plane, or the name of a saved feature. If a named feature is specified, this is interpreted as a datum plane if a plane is specified or as a datum axis if a line, cylinder, or cone is specified.

p2 states the nominal angle in degrees between the feature and the datum specified in **p1**. In the example, the 30 represents the size of the nominal angle between the feature and the **p1** datum.

p3 can only be XY, YZ, or ZX. This defines the engineering view that is to be used to compute angularity. This parameter must be specified for feature types line, cylinder, or cone; it may not be specified for feature type plane.

Results

For cones and cylinders the measured surface points are projected onto the feature axis to generate a set of axial points which are then used in these calculations. The system computes the minimum width of a zone between two parallel planes at the specified angle to the specified datum feature that is needed to hold all of the measured points. The computed value is stored in the deviation column of the dimensional array for angularity s(7,4). If the feature is a line, cylinder, or cone, then this zone is based only on the points or axis in the view specified.

Skip Block

Ignored; not executed if part of a skip block.

Error Messages

- MLB-023 The third parameter is invalid. May indicate that the feature is a plane but a projection plane **p3** was specified.
- MLB-025 The feature is a line, cone, or cylinder, but no projection plane **p3** was specified.
- MLB-037 The feature type is invalid. The feature type is not line, plane, cone, or cylinder.
- MLB-042 Summation invalid. The best-fit point data for the feature is required to make the requested evaluation but is no longer available.

A2 - ANSI Parallelism

The A2 command computes parallelism deviation as a width of a zone containing the feature in the feature register.

Command Format

MLString "A2 **p1**,**p2**"

Example

1. MLString "A2 XA,ZX"

   The feature in the feature register is to be checked for conformance to parallelism to the X-axis when viewed in the ZX plane of the active reference frame.

2. MLString "A2 ZX"

   The feature is to be checked for parallelism to the ZX plane of the active reference frame.

Input

The Feature Register must contain a best-fit feature that is either:

- Line,
- Plane,
- Cone, or
- Cylinder.

The points used to derive this feature must still be available.

- **p1** = (XA, YA, ZA, XY, YZ, or ZX) or the name of a saved feature. (Parametric or string; required).
- **p2** = XY, YZ, or ZX (Parametric; optional).

p1 identifies the datum feature. This is either a parametric feature identifier (XA, YA, ZA, XY, YZ, or ZX) identifying either a datum axis or a datum plane, or the name of a saved feature. If a named feature is specified, this is interpreted as a datum plane if a plane is specified or as a datum axis if a line, cylinder, or cone is specified.

p2 can only be XY, YZ, or ZX. This defines the engineering view that is to be used to compute parallelism. This parameter may be specified for feature types line, cylinder, or cone; it may not be specified if the feature type is plane.

Results

For cones and cylinders the measured surface points are projected onto the feature axis to generate a set of axial points which are then used in these calculations.

If the features is a line, cone, or cylinder, the datum specified is an axis, and a projection plane is not specified, then parallelism is computed as the diameter of the smallest cylinder which includes all of the points and is parallel to the datum axis.

In all other cases, parallelism is computed as the minimum width of a zone between two parallel planes which are parallel to the specified datum feature and holds all of the measured points.

If a projection plane is specified, then the point data is projected into that plane before parallelism is computed.

The computed value is stored in the deviation column of the dimensional array for parallelism s(11,4).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-022 The second parameter is invalid. Will occur if the feature is a plane and a projection plane **p2** is specified.
- MLB-037 Invalid feature type. Feature type is not line, plane, cone, or cylinder.
- MLB-042 Summation invalid. The best-fit point data for the feature is required to make the request evaluation but is no longer available.

A3 - ANSI Perpendicularity

The A3 command computes perpendicularity deviation as a width of a zone containing the feature in the feature register.

Command Format

MLString "A3 **p1**,**p2**"

Examples

1. MLString "A3 ZX"

   The feature is to be checked for perpendicularity to the ZX plane of the active reference frame.

Input

The Feature Register must contain a best-fit feature that is either:

- Line,
- Plane,
- Cone, or
- Cylinder.

The points used to derive this feature must still be available.

- **p1** = (XA, YA, ZA, XY, YZ, or ZX) or the name of a saved feature. (Parametric or string; required).
- **p2** = XY, YZ, or ZX (Parametric; optional)

p1 identifies the datum feature. This is either a parametric feature identifier (XA, YA, ZA, XY, YZ, or ZX) identifying a datum axis or a datum plane, or the name of a saved feature. If a named feature is specified, this is interpreted as a datum plane if a plane is specified or as a datum axis if a line, cylinder, or cone is specified.

p2 can only be XY, YZ, or ZX. The defines the engineering view that is to be used to compute perpendicularity. This parameter may be specified for feature types line, cylinder, or cone. It may not be specified if the feature type is a plane.

Results

For cones and cylinders the measured surface points are projected onto the feature axis to generate a set of axis points which are then used in these calculations.

If the feature is a line, cone, or cylinder and the datum specified is a plane, then perpendicularity is computed as the diameter of the smallest cylinder which includes all of the points and is perpendicular to the datum plane.

In all other cases, perpendicularity is computed as the minimum width of a zone bounded by two parallel planes which are perpendicular to the specified datum feature and holds all of the measured points.

If a projection plane is specified, then the point data is projected into that plane before perpendicularity is computed.

The computed value is stored in the deviation column of the dimensional array for perpendicularity s(10,4).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-037 The feature type is invalid. Feature type is not line, plane, cone, or cylinder.
- MLB-042 Summation invalid. The best-fit point data for the feature is required but is no longer available.

A4 - ANSI Runout

The A4 command computes total runout deviation between a best-fit feature in the feature register and a datum axis.

Command Format

MLString "A4 **p1**,**p2**"

Example

1. MLString "A4 XA,0"

The feature in the feature register is to be checked for total runout relation to the X-axis of the active reference frame.

2. MLString "A4 ZA,30"

   The feature is to be checked for total runout relative to a basic angle of 30 degrees to the Z-axis of the active reference frame.

Input

The Feature Register must contain a best-fit feature that is either:

- Circle,
- Ellipse,
- Plane,
- Cone, or
- Cylinder.

The points used to derive this feature must still be available.

- **p1** = (XA, YA, or ZA) or the name of a saved feature (Parametric or string; required).
- **p2** = Nominal angle (Numeric; optional).

p1 identifies the datum feature. This is either a parametric feature identifier (XA, YA, or ZA) identifying the datum axis or the name of a saved feature. If a named feature is specified, it must be a line, cylinder, or cone to be used as the datum axis.

p2 is the nominal angle in degrees between the feature and the datum axis. If the feature surface is nominally parallel to the axis, **p2** must be 0. If the feature surface is nominally perpendicular to the datum axis, **p2** must be 90. If not specified, **p2** defaults to 90 for feature type plane and to 0 for all other feature types.

Results

Total runout is computed based upon the variation of the set of measured points from a true surface of rotation at the specified angle from the datum axis.

The computed value is stored in the deviation column of the dimensional array for total runout s(12,4).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-037 Invalid feature type. Feature type is not circle, ellipse, plane, cone, or cylinder.
- MLB-042 Summation invalid. The best-fit point data for the feature is required to make the request evaluation but is no longer available.

A5 - Position Deviation

The A5 command computes position deviation for the feature in the feature register.

Command Format

MLString "A5 **p1**,**p2**,**p3**"

Examples

1. MLString "A5 15,-7.5,0,V0,0,1"

   The feature has a theoretically exact position defined by the XYZ coordinates (15,-7.5,0) and the IJK direction cosines (0,0,1), i.e., parallel to the Z-axis. The shape of the tolerance zone is either a cylinder or a pair of parallel planes based upon the type of feature.

2. MLString "A5 15,-7.5,0"

   The feature has a theoretically exact position defined by the XYZ coordinates (15,-7.5,0). Since no direction cosines are specifies, the feature must be a point, circle, ellipse, or sphere. The shape of this tolerance zone is either a circle (if the feature is a point, circle, or ellipse) or is a sphere (if the feature is a sphere).

3. MLString "A5 15,-7.5,0,V0,0,1,YZ"

   The feature has a theoretically exact position defined by the XYZ coordinates (15,-7.5,0) and the IJK direction cosines (0,0,1). The shape of this tolerance zone is a pair of parallel planes perpendicular to the YZ plane. The feature must be a line, cylinder, or cone.

Input

Except for point type features (points, circles, ellipses and spheres), the points used to derive this feature must still be available.

- **p1** = X, Y, Z location of feature (numeric; required).
- **p2** = orientation of feature position zone as V and the I,J,K direction cosines (numeric; required for feature types line, plane, cylinder, and cone).
- **p3** = XY, YZ, or ZX (Parametric; optional).

p1 gives the X, Y, Z coordinates of the theoretically exact feature location in the active reference frame.

p2 defines the orientation of the feature position zone. **p2** is required for feature types line, plane, cylinder, and cone and is not allowed for other feature types. **p2** begins with the letter "V" and gives the direction cosines in I,J,K order defining the theoretically exact feature orientation in the active reference frame. The actual feature data is used to determine the extent of the tolerance zone.

p3 can only be XY, YZ, or ZX. This defines the engineering view that is to be used. Specification of an engineering view is required for feature types point, circle, and ellipse and defaults to the active working plane (points) or the plane of measurement (circles and ellipses) if parameter **p3** is not specified. Specification of an engineering view is allowed for feature types sphere, line, cylinder, and cone. For lines, cylinders, and cones the tolerance zone is a cylinder if **p3** is not included, and a pair of parallel planes if it is. For spheres the tolerance zone is a sphere if **p3** is not included, a circle otherwise. Specification of an engineering view is not allowed for feature type plane.

Results

The positional deviation is computed as follows :

- For features that are measured or constructed as points or to the center coordinates of two dimensional features such as circles and ellipses, position deviation is computed as the diameter of the smallest cylinder containing the point and whose axis passes through the nominal feature position **p1** and whose axis is normal to the specified projection plane **p3**.

- For spherical features where no projection plane is specified, position deviation is computed as the diameter of the smallest sphere containing the center point and whose center is at the nominal feature position **p1**.
- For line features where a projection plane is specified, position deviation is computed as the smallest distance between two parallel planes containing the set of points measured for the line; these two planes symmetrically located with respect to the nominal line position **p1**, each parallel to the nominal line **p2**, and each normal to the specified projection plane **p3**.
- For features with axes (cylinders and cones) where a projection plane is specified, position deviation is computed as the smallest distance between two parallel planes containing that portion of the computed feature axis bounded by the set of points measured for the feature; these two planes symmetrically located with respect to the nominal axis position **p1**, each parallel to the nominal axis **p2**, and each normal to the specified projection plane **p3**.
- For line features where a projection plane is not specified, position deviation is computed as the diameter of the smallest cylinder containing the set of points measured for the line; the axis of the cylinder being defined by the nominal line position vector **p1** and orientation vector **p2**.
- For features with axes (cylinders and cones) where a projection plane is not specified, position deviation is computed as the diameter of the smallest cylinder that contains that portion of the computed feature axis bounded by the set of points measured for the feature; the axis of this cylinder being defined by the nominal line position vector **p1** and orientation vector **p2**.
- For plane features, position deviation is computed as the smallest distance between two parallel planes containing the measured set of points; these two planes symmetrically located with respect to the nominal plane position **p1** and each parallel to the nominal plane **p2**.

The computed value in all cases is stored in the deviation column of the dimensional array row for position s(9,4). Note that this data is maintained in the array only until the next operation which updates the feature register position data.

Skip Action

Ignored; not executed if part of a skip block.

Error Message

- MLB-042 Summation invalid. The best-fit point data for the feature is required to make the requested evaluation but is no longer available.

A6 – Concentricity / Coaxiality Deviation

The A6 command computes concentricity or coaxiality deviation for a feature.

Command Format

MLString "A6 **p1**,**p2**"

Examples

1. MLString "A6 DP"

The feature is to be checked for concentricity to the datum point of the active reference frame.

2. MLString "A6 ZA"

   The feature is to be checked for coaxiality to the Z-axis of the active reference frame.

3. MLString "A6 DP,XY"

   The feature is to be checked for concentricity to the datum point in the XY plane of the active reference frame.

Input

Except for point type features (points, circles, ellipses and spheres), the points used to derive this feature must still be available.

- **P1** = (DP, XA, YA, or ZA) or the name of a saved feature (Parametric or string; required).
- **P2** = XY, YZ, or ZX (Parametric; optional).

p1 identifies the datum feature. This is either a parametric feature identifier (DP, XA, YA, or ZA) for the datum point or a datum axis; or the name of a saved feature. If a named feature is used it must be either a datum point (point, circle, sphere, or ellipse) or a datum axis (line, cylinder or cone).

p2 can only be XY, YZ, or ZX. This defines the engineering view to be used in computing concentricity. An engineering view is required for feature types point, circle, and ellipse and defaults to the active working plane (points) or the plane of measurement (circles and ellipses) if **p2** is not specified. An engineering view may be specified for feature type sphere; if not specified, evaluation defaults to a spherical zone.

Results

The concentricity or coaxiality deviation value is computed as follows:

- For points, circles, and ellipses, concentricity is computed as the diameter of the smallest cylinder whose axis is coincident with the datum feature **p1**, which is normal to the projection plane **p2**, and which contains the computed feature location.
- For spheres where a projection plane is specified, concentricity is computed as the diameter of the smallest cylinder whose axis is the datum feature **p1**, which is normal to the projection plane **p2**, and which contains the computed feature location.
- For spheres where a projection plane is not specified, concentricity is computed as the diameter of the smallest sphere located by the datum feature **p1** and which contains the computed feature location.
- For lines, coaxiality is computed as the diameter of the smallest cylinder whose axis is the datum axis **p1** and which contains the set of measured points.
- For cylinders and cones, coaxiality is computed as the diameter of the smallest cylinder whose axis is the datum axis **p1** and which contains that potion of the computed feature axis bounded by the set of measured points.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-037 Invalid feature type.

- MLB-042 The best-fit point data for the feature is required to make the requested evaluation but is no longer available.

A7 - Circular Runout Deviation

This command computes circular runout deviation for the feature in the feature register.

Command Format

MLString "A7 **p1**,**p2**,**p3**"

Examples

1. MLString "A7 ZA"

   The feature is to be checked for circular runout to the Z-axis of the active reference frame. The inclination angle and the zone width are to be determined from the feature data as described below.

2. MLString "A7 XA,15,W.1"

   The feature is to be checked for circular runout at an angle of 15 degrees relative to the X-axis. All points that fall within a circular cross-section 0.1 wide are to be treated as a subgroup for evaluating circular runout.

Input

The points used to derive this feature must still be available.

- **p1** = (XA, YA, or ZA) or the name of a saved feature (Parametric or string; required).
- **p2** = Inclination direction (Numeric; optional).
- **p3** = W and width of cross-sectional zones (Numeric; optional).

p1 identifies the datum feature. This is a parametric feature identifier (XA, YA, or ZA) identifying a datum axis; or the name of a saved feature. If a named feature is specified, it must be a line, cylinder, or cone to be used as the datum axis.

p2 is the amount of inclination of the nominal surface relative to the datum axis. The inclination angle defaults to 0 degrees (radial runout) for circles, ellipses, cylinders, and cones. The inclination angle defaults to 90 degrees (axial runout) for planes.

p3 is a the letter "W" followed by a numeric value. This is the width of the cross-sectional zones to be used in checking circular runout. The zone width is infinite for circles and ellipses whether **p3** is specified or not. For cylinders and cones the zone width defaults to three times the feature length divided by the number of points measured (i.e., averages three points per default cross-section). For planes, the zone width defaults to three times the maximum distance between any of the measured points and the datum axis divided by the number of points measured.

Results

The circular runout deviation is computed as follows:

- For cylindrical features or circles (treated as single cross-section), circular runout is the maximum of the circular runout deviations for the set of circular cross-sections defined by the set of measured points and the width **p3**. The circular runout deviation for each cross-section is the smallest distance between two coaxial cylinders containing the points in that cross-section; these cylinders having axis **p1**.

- For planes, circular runout is the maximum of the axial runout deviations for the set of annular cross-sections defined by the set of measured points and the width **p3**. The axial runout deviation for each cross-section is the smallest distance between two parallel planes containing the points in that cross-section; these planes being normal to the datum axis **p1**.
- For cones, circular runout is the maximum of the circular runout deviations for the set of conical cross-sections defined by the set of measured points, the width **p3**, and the nominal angle **p2**. The circular runout deviation for each cross-section is the smallest distance between two coaxial cones containing the points in that cross-section; these cones have axis **p1** and included apex angle twice the nominal angle **p2**.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-037 Invalid feature type. Circular runout can't be computed for points, lines, or spheres.
- MLB-042 Summation invalid. The best-fit point data for the feature is required to make the requested evaluation but is no longer available.

A8 - Symmetry Deviation

The A8 command computes symmetry deviation for the feature in the feature register.

Command Format

MLString "A8 **p1**,**p2**"

Examples

1. MLString "A8 XY"

   Check the feature for symmetry to the XY datum plane of the active reference frame.

2. MLString "A8 ZA,ZX"

   Check the feature for symmetry to the Z-axis in the ZX plane of the active reference frame.

Input

The points used to derive this feature must still be available.

- **p1** = (XA, YA, ZA, XY, YZ, or ZX) or the name of a saved feature (Parametric or string; required).
- **p2** = XA, YA, or ZA (Parametric; optional).

p1 identifies the datum feature. This is either a parametric identifier (XA, YA, ZA, XY, YZ, or ZX) or the name of a saved feature identifying a datum axis or plane. If a named feature is specified, this is treated as a datum plane if a plane is specified or as a datum axis if a line, cylinder, cone, or width is specified.

p2 can only be XY, YZ, or ZX. This is the engineering view to be used to compute symmetry. Specification of an engineering view is required for line and axis feature types but is not allowed for feature type plane.

Results

The symmetry deviation is computed as follows:

- For line features, symmetry is computed as the minimum distance between two parallel lines, each parallel to the datum axis **p1**, disposed symmetrically with respect to the datum axis, and which contain the set of measured points as projected into the engineering view **p2**.
- For features with axes, symmetry is computed as the minimum distance between two parallel lines, each parallel to the datum axis **p1**, disposed symmetrically with respect to the datum axis, and which contain the set of measured points as projected onto the best-fit feature axis in the engineering view **p2**.
- For plane features, symmetry is computed as the minimum distance between two parallel planes, each parallel to the datum plane **p1**, disposed symmetrically with respect to the datum plane, and which contain the set of measured points.

The computed deviation is stored in the deviation column of the dimensional array for position s(9,4). This value is maintained in the array only until the next operation that updates the feature register position data.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-037 Invalid feature type. Symmetry can't be computed for points, circles, ellipses, or spheres.
- MLB-042 Summation invalid. The best-fit point data for the feature is required to make the requested evaluation but is no longer available.

AI - Included Angle

The AI command computes the included angle between two features. The two features must be lines (including cones, cylinders, slots, and webs) or planes.

Command Format

MLString "AI P1,P2"

Examples

1. MLString "AI"
2. MLString "AI F6,LINE1"
3. MLString "AI FR,XA"

Input

- **P1** = Parametric; optional.

- **P2** = Parametric; optional.

P1 and P2 must identify features, either in the feature stack or in named feature storage. P2 may also be a parametric variable designating a coordinate frame axis (XA, YA, ZA) or plane (XY, YZ, ZX). The parameters P1 and P2 must be both specified or both omitted. If both P1 and P2 are omitted, they default to FR and F1 respectively, as shown in Example 1. The features specified may be two planes, two lines, or a line and a plane. Note that features with axes such as cones, cylinders, slots, and webs are treated as lines by this command.

In Example 2, AI computes the included angle between the feature and stack position #6 and the feature "LINE1" in named feature storage.

Example 3 computes the included angle between the feature in FR and the X- axis of the active reference frame.

Results

The result is an angle that is greater or equal to 0 and less than 180 degrees. It is stored in the Dimensional Array, position S(7,0).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 Referenced features not defined.
- MLB-031 P1 is not a line or a plane.
- MLB-032 P2 is not a line or a plane.

AN - Angularity

The AN command computes the angle between lines and planes. The angle is an acute angle, expressed in degrees.

Command Format

MLString "AN P1,P2"

Example

- MLString "AN LN5,PL9"

Input

- **P1** = Data in FR, Stack or Storage (Parametric; optional).
- **P2** = Data in FR, Stack or Storage or a datum axis or plane (Parametric; optional).

P1 and P2 identify features in the Feature Register, Feature Stack or Feature Storage. P2 may also designate a datum axis (XA, YA, or ZA) or datum plane (XY, YZ, or ZX). The types of features specified by the arguments must be:

  - Two lines or features with axes (cylinders, cones, widths),
  - A line or a feature with an axis and a plane, or

- Two planes.

In the example, LN5 refers to a feature saved in Feature Storage. The PL9 refers to a feature saved in Feature Storage. AN computes the angle between LN5 and PL9. If P1 and P2 are omitted, the command uses the data from FR and F1 by default.

Results

The Dimensional Array stores the angle in S(7,0).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-031 The first parameter specifies an invalid type.
- MLB-032 The second parameter specifies an invalid type.

CV - Changing Sign or Orientation Vector

The CV command is used to correct the sign of the orientation vector of the feature in the Feature Register to align with a given nominal vector.

This command is useful when checking the orientation of a feature, such as a cylinder. The actual direction vector (I, J, K) can be forced to point in the same general direction as the nominal direction vector using **CV**.

Command Format

MLString "CV P1"

Examples

1. MLString "CV .57,0,.45"

Input

- **P1** = Three numbers representing I, J, and K nominal values (Numeric; required).

P1 is the nominal orientation vector for the feature in the feature register. In the example, the nominal vector is (.57,0,.45).

Results

If the magnitude of the angle between the actual and the nominal orientation vectors is greater than ninety degrees, then the actual orientation vector is negated. The orientation vector is updated in the feature register and the host S(14,0), S(15,0), and S(16,0) Dimensional Array as required.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 The feature register does not contain the required feature.

DA - Axial Distance

The DA command computes the three-dimensional distance between two points, two lines, two planes, a line and a plane, a point and a line, or a point and a plane. It also derives the axial components of the computed distance and pushes these as a 'distance' feature onto the feature stack.

Normal geometric calculations are used to calculate most distances. The following exceptions apply:

- The distance from a line to a plane, skewed less than five degrees, is the shortest possible distance from the base point of the first feature specified (P1) to the second feature (P2).
- The distance between two planes, skewed less than five degrees, is the shortest possible distance from the base point of the first plane specified (P1) to the other plane (P2).
- The distance between two lines, skewed less than five degrees, is the shortest possible distance from the base point of the first line specified (P1) to the other line (P2).

Command Format

MLString "DA P1,P2"

Examples

1. MLString "DA F4,OD5"

Input

- **P1** = Data in FR, Stack or Storage (Parametric; optional).
- **P2** = Data in FR, Stack or Storage (Parametric; optional).

P1 and P2 identify features in the Feature Register, Feature Stack or Feature Storage. P2 may also be used to identify a coordinate frame axis (XA, YA, or ZA), plane (XY, YZ, or ZX), or datum point (DP). If both P1 and P2 are omitted then they default to FR and F1 respectively.

The types of features specified must be :

- Two point type (point, circle, ellipse, sphere) features,
- Two line type (line, cone, cylinder, slot/width) features,
- Two planes,
- One line type feature and a plane,
- One point type feature and one line type feature,
- One point type feature and one plane.

In the example, F4 refers to the feature in the fourth position of the Feature Stack. The OD5 refers to a feature saved in Feature Storage. DA computes the distance between F4 and OD5. It also derives the axial components for that distance and creates an appropriate distance feature.

Results

The contents of the Feature Register are moved into the Feature Stack. Then, the axial components are stored in the Feature Register.

The Dimensional Array stores:

- Distance in S(8,0),
- Axial components of the distance in the X, Y and Z coordinate rows.

The feature type is set to distance (19). The command sets the signs of the axial components for the vector from the first feature to the second feature. Data is valid only if the reference remains unmoved after calculating distance.

Skip Action

Not executed if part of a skip block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required features.

DS - Distance

The DS command computes the three-dimensional distance between two points, two lines, two planes, a line and a plane, a point and a line, or a point and a plane.

Normal geometric calculations are used to calculate most distances. The following exceptions apply:

- The distance from a line to a plane, skewed less than five degrees, is the shortest possible distance from the base point of the first feature specified (P1) to the second feature (P2).
- The distance between two planes, skewed less than five degrees,is the shortest possible distance from the base point of the first plane specified (P1) to the other plane (P2).
- The distance between two lines, skewed less than five degrees,is the shortest possible distance from the base point of the first line specified (P1) to the other line (P2).

Command Format

MLString "DS P1,P2"

Examples

1. MLString "DS F4,OD5"

Input

- **P1** = Data in FR, Stack or Storage (Parametric; optional).
- **P2** = Data in FR, Stack or Storage (Parametric; optional).

P1 and P2 identify features in the Feature Register, Feature Stack or Feature Storage. P2 may also be used to identify a coordinate frame axis (XA, YA, or ZA), plane (XY, YZ, or ZX), or datum point (DP). If both P1 and P2 are omitted then they default to FR and F1 respectively.

The types of features specified must be:

- Two point type (point, circle, ellipse, sphere) features,
- Two line type (line, cone, cylinder, slot/width) features,
- Two planes,
- One line type feature and a plane,

- One point type feature and one line type feature,
- One point type feature and one plane.

In the example, F4 refers to the feature in the fourth position of the Feature Stack. The OD5 refers to a feature saved in Feature Storage. DA computes the distance between F4 and OD5.

Results

The Dimensional Array stores the computed distance in S(8,0).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required features.

FC - Form Calculation Method

The FC command allows the choice between two alternate methods for computing feature form deviation during best-fit calculations. The default method for computing feature form deviation for lines, planes, circles, spheres, and cylinders is referred to as MRS or "Minimal Radial Separation". In this method the form is computed as the minimal separation between two parallel feature surfaces of perfect form which contain the set of points. The second method (always used by MP30 and SMP-3xx systems with MeasureMax+ versions prior to 6.00) is to compute form as the deviation of the individual points from the computed feature surface.

Note that the first method is consistent with national and international dimensioning and tolerancing standards while the second is the traditional method used by CMM systems and may be more useful for analyzing form deviations.

Command Format

MLString "FC P1"

Examples

1. MLString "FC BFD"

Input

- **P1** = BFD or MRS; (String, optional).

P1 may only be BFD or MRS. "MRS" specifies that form calculations for lines, planes, circles, spheres, and cylinders is to be done by the MRS method independent of the method used to compute feature size and position. "BFD" specifies that form calculations are to be based upon point deviations from the computed feature size and orientation.

The current state of the form calculation setting is requested by not specifying P1.

Results

If P1 is specified then feature form calculations for lines, planes, circles, spheres, and cylinders is done using the specified method until another **FC** command is executed or until a part program initialization. Note that form for cones and slots/webs is always computed by the best-fit surface deviation method.

If P1 is not specified then either "BFD" or "MRS" is returned to system variable S0D to indicate the active form calculation method.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

I0 - Reporting Form

The I0 command identifies and reports the form of the feature in the feature register. **I0** prints an ISO or ANSI or DIN identification line with each report that reflects which standard the program is running under.

Command Format

MLString "I0"

Example

1. MLString "I0"

Input:

Dimensional Array.

Results

The command generates a report of form. In the following sample report the ‘feature form’ line is shown three times; however, only one of the lines will appear in your report.

ISO FEATURE FORM: FLATNESS
or
ANSI FEATURE FORM: FLATNESS
or
DIN FEATURE FORM: FLATNESS

#3
Feature #10
Type: PLANE
Name:

	ACTUAL	NOMINAL	+ TOL	- TOL	DEV	OUT-TOL
FM			0.100		0.025	0.025

The identification line (ISO / ANSI / DIN FEATURE FORM: FLATNESS) above reflects the type of feature in the Feature Register. Other identification lines are:

- 2-dimensional line (based on active plane):

ISO / ANSI / DIN FEATURE FORM: STRAIGHTNESS IN XY PLANE
ISO / ANSI / DIN FEATURE FORM: STRAIGHTNESS IN YZ PLANE
ISO / ANSI / DIN FEATURE FORM: STRAIGHTNESS IN ZX PLANE

- 3-dimensional line:

  ISO / ANSI / DIN FEATURE FORM: STRAIGHTNESS IN SPACE

- Plane:

  ISO / ANSI / DIN FEATURE FORM: FLATNESS

- Circle, I.D., O.D.:

  ISO / ANSI / DIN FEATURE FORM: CIRCULARITY

- Cylinder:

  ISO / ANSI / DIN FEATURE FORM: CYLINDRICITY

- Ellipse:

  ISO / ANSI / DIN FEATURE FORM: ELLIPTICITY

- Sphere:

  ISO / ANSI / DIN FEATURE FORM: SPHERICITY

- Cone:

  ISO / ANSI / DIN FEATURE FORM: CONICITY

Skip Block

Ignored; not executed if part of a skip block.

Error Messages

None.

I1 - Reporting Angularity

The I1 command computes angularity deviation as a width of a zone containing the feature in the feature register and prints a report showing this result. **I1** prints an ISO, ANSI, or DIN identification line with each report that reflects which standard is active.

Command Format

MLString "I1 **p1**,**p2**,**p3**,**p4**"

Examples

1. MLString "I1 XA,30,ZX"

   The feature in the feature register is to be checked for angularity with a nominal angle of 30 degrees to the Z-axis after it has been projected into the ZX plane of the active reference frame.

2. MLString "I1 XA,45,2.2"

   The feature is checked for angularity to the X-axis with a nominal angle of 45 degrees and with the calculated results projected across a length of 2.2 units.

Input

The Feature Register must contain a best-fit feature that is either:

- Line,
- Plane,
- Cone, or
- Cylinder.

The points used to derive this feature must still be available.

- **p1** = XA, YA, ZA, XY, YZ, ZX, or the name of a saved feature. (Parametric or string; required).
- **p2** = Nominal angle (Numeric; required).
- **p3** = XY, YZ, or ZX (Parametric; optional).
- **p4** = Length of tolerance zone (Numeric; optional).

p1 identifies the datum feature. This is either a parametric feature identifier (XA, YA, ZA, XY, YZ, or ZX) identifying a datum axis or a datum plane; or the name of a saved feature. If a named feature is specified, this is treated as a datum plane if a plane is specified or as a datum axis if a line, cylinder, or cone is specified.

p2 is the nominal angle in degrees between the feature and the datum **p1**.

p3 can only be XY,YZ, or ZX. This defines the engineering view that is to be used to compute angularity. This parameter must be specified for feature types line, cylinder, or cone; it may not be specified if the feature type is plane.

p4 specifies the length of a tolerance zone for length dependent calculation of the angularity deviation.

Results

The tolerance value is stored in the plus tolerance column of the dimensional array for angularity (s(7,2) variable).

If a length of tolerance zone is specified, the angle between the best-fit feature and the datum feature (in the viewing plane **p3** if specified) is used to compute a units-per-unit deviation value which is then multiplied by the specified length.

Otherwise, for cones and cylinders the measured points are projected onto the feature axis to generate a set of axial points. If a projection plane is specified then the measured / projected points are projected into that plane before angularity is computed. The angularity deviation value is computed as the minimum width of a zone between two parallel planes at the specified angle to the specified datum feature that holds all of the measured / projected points.

The computed value is stored in the deviation column of the dimensional array for angularity (s(7,4) variable).

An inspection report such as the following is generated, with the initial comment modified to reflect the command arguments and the second line generated only if a tolerance zone length has been specified:

ISO ANGULARITY: To Z-axis in ZX plane
Projected Tolerance Zone: Length=2.2
#3

Feature #10
Type: CYLINDER
Name:

	ACTUAL	NOMINAL	+ TOL	- TOL	DEV	OUT-TOL
AM			0.100		0.011	0.001

Skip Block

Ignored; not executed if part of a skip block.

Error Messages

- MLB-023 The current feature is a plane and a projection plane **p3** is specified.
- MLB-025 The current feature is a line, cone, or cylinder and a projection plane **p3** is not specified.
- MLB-037 Invalid feature type. Feature type is not a line, plane, cone, or cylinder.
- MLB-042 The best-fit point data for the feature is required to make the requested evaluation but is no longer available.

I2 - Reporting Parallelism

The I2 command computes parallelism deviation as a width of a zone containing the feature in the feature register and prints a report showing this result. **I2** prints an ISO, ANSI, or DIN identification line with each report that reflects which standard is active.

Command Format

MLString "I2 **p1**,**p2**,**p3**,**p4**"

Examples

1. MLString "I2 ZA,ZX"

   The feature in the feature register is to be checked for parallelism to the X-axis when viewed in the ZX plane of the active reference frame.

2. MLString "I2 ZX"

   The feature is to be checked for parallelism to the ZX plane of the active reference frame.

3. MLString "I2 XY,50,100"

   The feature plane is to be checked for parallelism to the XY datum plane with the calculation based upon a projected plane of length 50 and width 100.

Input

The Feature Register must contain a best-fit feature that is either:

- Line,
- Plane,

- Cone, or
- Cylinder.

The points used to derive this feature must still be available.

- **p1** = XA, YA, ZA or XY, YZ, ZX, or the name of a saved feature (parametric variable or string; required).
- **p2** = XY, YZ, or ZX (parametric variable; optional).
- **p3** = Length of projected tolerance zone (Numeric; optional).
- **p4** = Width of project tolerance zone (Numeric; optional)

p1 identifies the datum feature. This is either a parametric feature identifier (XA, YA, ZA, XY, YZ, or ZX) identifying either a datum axis or a datum plane, or the name of a saved feature. If a named feature is specified, this is interpreted as a datum plane if a plane is specified or as a datum axis if a line, cylinder, or cone is specified.

p2 specifies a projection plane corresponding to the engineering view calling out the parallelism tolerance. You may include this argument if the feature is a line, cone, or cylinder and the specified datum is an axis. Omit **p2** if the feature is a plane.

p3 specifies the length of the projected tolerance zone for length dependent calculations of the deviation.

p4 specifies the width of the projected tolerance zone. **p4** can only be specified is the measured feature is a plane, **p1** specifies a datum plane (not a named feature), and **p3** has also been specified.

Results

The tolerance value is stored in the plus tolerance column of the dimensional array for parallelism (s(11,2) variable).

If a length of tolerance zone is specified then parallelism is computed as follows:

- If the feature is a line or axis, the parallelism between the feature and the datum is computed as a units-per-units deviation value and then multiplied by the specified length.

- If the feature is a plane and only a length of tolerance zone is specified, the parallelism between the feature and datum feature is computed as a units-per-unit deviation value and then multiplied by the specified length.

- If the feature is a plane and both a length and a width of tolerance zone are specified, then the datum must be a plane. The parallelism between the feature and a planar patch of the specified length (along the first named axis of the datum plane) and of the specified width (along the second named axis of the datum plane) is computed as a units-per-unit deviation value and multiplied by the width of the planar patch in the direction of maximum deviation.

Otherwise, for cones and cylinders the measured surface points are projected onto the feature axis to generate a set of axial points. If a projection plane is specified, the point data is projected into that plane before parallelism is computed. Then:

- If the feature is a line, cone, or cylinder, the datum specified is an axis, and a projection plane is not specified, parallelism is computed as the diameter of the smallest cylinder which includes all of the points and is parallel to the datum axis.

- In all other cases, parallelism is computed as the minimum width of a zone between two parallel planes that are parallel to the specified datum feature and holds all of the measured points.

The computed value is stored in the deviation column of the Dimensional Array for parallelism (s(11,4) variable).

An inspection report such as the following is generated with the initial comment modified to reflect the command arguments and the second line generated only if a tolerance zone length has been specified:

ISO PARALLELISM: To XY Plane
Projected Tolerance Zone: Length=50, Width=100
#3
Feature #10
Type: PLANE
Name:

	ACTUAL	NOMINAL	+ TOL	- TOL	DEV	OUT-TOL
PA			0.010		0.011	0.001

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-022 occurs if the feature is a plane and a projection plane **p2** is specified.
- MLB-037 The feature type is invalid. Feature type is not a line, plane, cone, or cylinder.
- MLB-042 Best-fit point data for the feature is required to make the requested evaluation but is no longer available.

I3 - Reporting Perpendicularity

The I3 command computes perpendicularity as a width of a zone containing the feature in the Feature Register and prints a report showing this result. **I3** prints an ISO, ANSI, or DIN identification line with each report that reflects which standard is active.

Command Format

MLString "I3 **p1**,**p2**,**p3**,**p4**"

Examples

1. MLString "I3,ZX"

   The feature is to be checked for perpendicularity to the ZX plane of the active reference frame.

2. MLString "I3 XA,50,100"

   The feature plane is to be checked for perpendicularity to the X datum axis with the calculation based upon a projected plane of length 50 and width 100.

Input

The Feature Register must contain a best-fit feature that is either:

- Line,
- Plane,

- Cone, or
- Cylinder.

The points used to derive this feature must still be available.

- **p1** = XY, YZ, ZX, XA, YA, or YZ or the name of a saved feature. (parametric variable or string; required).
- **p2** = XY, YZ, or ZX (parametric variable; optional) .
- **p3** = Length of the projected tolerance zone (Numeric; optional).
- **p4** = Width of the projected tolerance zone (Numeric; optional).

p1 identifies the datum feature. This is either a parametric feature identifier (XA, YA, ZA, XY, YZ, or ZX) identifying a datum axis or a datum plane, or the name of a saved feature. If a named feature is specified, this is treated as a datum plane if a plane is specified or as a datum axis if a line, cylinder, or cone is specified.

p2 can only be XY, YZ, or ZX. This defines the engineering view that is to be used to compute perpendicularity. This parameter may be specified for feature types line, cylinder, or cone; it may not be specified if the feature type is a plane.

p3 specifies the length of the projected tolerance zone for length dependent calculation of the deviation.

p4 specifies the width of the projected tolerance zone. **p4** can only be specified if the measured feature is a plane, **p1** specifies a datum axis (not a named feature), and **p3** has also been specified.

Results

The tolerance value is stored in the plus tolerance column of the dimensional array for perpendicularity (s(10,2) variable).

If a length of tolerance zone is specified then perpendicularity is computed as:

- If the feature is a line or axis, then perpendicularity between the feature and the datum feature is computed as a units-per-unit deviation value and then multiplied by the specified length.
- If the feature is a plane and only a length of tolerance zone is specified, the perpendicularity between the feature and the datum feature is computed as a units-per-unit deviation value and then multiplied by the specified length.
- If the feature is a plane and both a length and a width of tolerance zone are specified, then the datum must be an axis. Then parallelism is computed between the feature and a planar patch of the specified length (along the first named axis of the datum plane normal to the specified axis) and of the specified width (along the second named axis of the datum plane normal to the specified axis) as a units-per unit deviation value and multiplied by the width of the plane in the direction of maximum deviation.

Otherwise, for cones and cylinders the measured surface points are projected onto the feature axis to generate a set of axial points. If a projection plane is specified, then the point data is projected into that plane before perpendicularity is computed. Then:

- If the feature is a line, cone, or cylinder, the datum specified is a plane, and a projection plane is not specified, perpendicularity is computed as the diameter of the smallest cylinder which includes all of the points and is perpendicular to the datum plane.

- In all other cases, perpendicularity is computed as the minimum width of a zone that holds all the measured points and is between two parallel planes that are perpendicular to the specified datum feature.

The computed value is stored in the deviation column of the dimensional array for perpendicularity (s(10,4) variable).

An inspection report such as the following is generated, with the initial comment modified to reflect the command arguments and the second line is generated only if a tolerance zone length has been specified:

ISO PERPENDICULARITY: To X-axis
Projected Tolerance Zone: length=50, width=100
#3
Feature #10
Type: PLANE
Name:

	ACTUAL	NOMINAL	+ TOL	- TOL	DEV	OUT-TOL
PE			0.010		0.011	0.001

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-037 Invalid feature type. Feature type is not line, plane, cone, or cylinder.
- MLB-042 Summation invalid. The best-fit point data for the feature is required to make the requested evaluation but is no longer available.

I4 - Reporting Total Runout Deviation

The I4 command computes total runout deviation between the best-fit feature in the Feature Register and a datum axis and prints a report showing the result. **I4** prints an ISO, ANSI, or DIN identification line with each report that reflects which standard is active.

Command Format

MLString "I4 **p1**,**p2**,**p3**,**p4**,**p5**"

Examples

1. MLString "I4 ZA,T.01"

   The feature in the feature register is to be checked for total runout to the Z-axis of the active reference frame. The width of the tolerance zone is 0.01. The inclination angle is to be determined from the feature type as described below.

2. MLString "I4 XA,15,T.01"

The feature is to be checked for total runout at an angle of 15 degrees to the X-axis of the active reference frame. The width of the tolerance zone is 0.01.

3. MLString "I4 XA,T.01,1.0,2.0"

   The feature is to be checked for total radial runout to the X-axis of the active reference frame. The diameter of the tolerance zone is 0.01. The inclination angle defaults to 90 degrees. The start position along the axis is 1.0. The length of the projected tolerance zone is 2.0.

Input

The Feature Register must contain a best-fit feature that is either:

- Circle,
- Ellipse,
- Plane,
- Cone, or
- Cylinder.

The points used to derive this feature must still be available.

- **p1** = XA, YA, ZA or the name of a saved feature (Parametric variable or string; required).
- **p2** = Nominal angle (Numeric; optional).
- **p3** = T and width of tolerance zone (Numeric; required).
- **p4** = Nominal position (Numeric; optional).
- **p5** = Length of the tolerance zone (Numeric; optional).

p1 identifies the datum feature. This is either a parametric feature identifier (XA, YA, or ZA) identifying a datum axis, or the name of a saved feature. If a named feature is specified, it must be a line, cylinder, or cone to be used as the datum axis.

p2 is the nominal angle in degrees between the feature surface and the datum axis. This angle defaults to 90 degrees (radial runout) for circles, ellipses, cylinders, and cones; it defaults to 0 degrees (axis runout) for planes. If **p4** and **p5** are used to compute length dependent radial runout then **p2** must be omitted or set to 90 degrees.

p3 is a value preceded by the letter "T" defining the width of the tolerance zone.

p4 specifies a nominal position along the named datum axis for length dependent calculation of runout deviation. The parameter may only be specified for lines, cones, or cylinders and when the datum is not a named feature. If **p4** is given, **p5** must also be provided.

p5 specifies the length of the tolerance zone for length dependent calculation of radial runout.

Results

The tolerance value is stored in the plus tolerance column of the Dimensional Array for runout (s(12,2) variable).

If **p4** and **p5** are not specified, total runout is computed based upon the variation of the set of measured points from a true surface of rotation at the specified angle from the specified datum axis.

If **p4** and **p5** are specified, the calculation is for length-dependent total radial runout. The deviation in this case is computed based upon the axis for the best-fit feature, at the specified nominal start position and projected across the given length. The computed value is the diameter of the cylindrical deviation.

The computed value is stored in the deviation column of the Dimensional Array for total runout (s(12,4) variable).

An inspection report such as the following is generated, with the initial comment modified to reflect the command arguments and the second line generated only if a tolerance zone length has been specified:

ISO TOTAL RUNOUT: To X-axis
Projected Tolerance Zone: nominal position=1.0, length=2.0
#3
Feature #10
Type: CYLINDER
Name:

	ACTUAL	NOMINAL	+ TOL	- TOL	DEV	OUT-TOL
RO			0.010		0.011	0.001

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-037 Invalid feature type. Runout can't be computed for points or spheres and can be computed for lines only if the calculation is length dependent.
- MLB-042 The best-fit point data for the feature is required to make the requested evaluation but is not longer available.

I5 - Reporting Position Deviation

The I5 command computes and reports position deviation for the feature in the feature register. **I5** prints an ISO, ANSI, or DIN identification line with each report that reflects which standard is active.

Command Format

MLString "I5 **p1**,**p2**,**p3**,**p4**,**p5**,**p6**"

Examples

1. MLString "I5 15,-7.5,0,V0,0,1"

   The feature has a theoretically exact position defined by the XYZ coordinates (15,-7.5,0) and the IJK direction cosines (0,0,1), i.e., parallel to the Z-axis. The shape of the tolerance zone is either a cylinder or a pair of parallel planes based upon the type of feature. The diameter of the tolerance zone is 0.01.

2. MLString "I5 15,-7.5,0,T.01"

   The feature has a theoretically exact position defined by the XYZ coordinate (15,-7.5,0). Since no direction cosines are specified, the feature must be a point, circle, ellipse, or sphere. The shape of this tolerance zone is either a circle (if the feature is a point, circle, or ellipse) or a sphere (if the feature is a sphere). The diameter of the tolerance zone is 0.01.

3. MLString "I5 15,-7.5,0,V0,0,1,YZ,T.01"

   The feature has a theoretically exact position defined by the XYZ coordinate (15,-7.5,0) and the IJK direction cosines (0,0,1). The shape of this tolerance zone is a pair of parallel planes perpendicular to the YZ plane. The feature must be a line, cylinder, or cone. The width of the tolerance zone is 0.01.

4. MLString "I5 15,-7.5,0,V0,0,1,T.01,2.0,2.5"

   The feature plane has a theoretically exact position defined by the XYZ coordinates (15,-7.5,0) and the IJK direction cosines (0,0,1), i.e., parallel to the Z-axis. The shape of the tolerance zone is a pair of parallel planes since the feature must be plane (both **p5** and **p6** specified). The thickness of the tolerance zone is 0.01.

Input

- **p1** = X, Y, Z location of feature (numeric; required).
- **p2** = orientation of feature position zone as V and the I,J,K direction cosines (numeric; required for feature types Plane, Cylinder,and Cone).
- **p3** = XY, YZ, or ZX (parametric variable; optional).
- **p4** = letter T with tolerance zone width (Numeric; required).
- **p5** = length of tolerance zone (Numeric; optional).
- **p6** = width of tolerance zone (Numeric; optional).

p1 gives the X, Y, Z coordinates of the theoretically exact feature location in the active reference frame.

p2 defines the orientation of the feature position zone. **p2** is required for feature types line, plane, cylinder, and cone. **p2** begins with the letter "V" and gives the direction cosines in I,J,K order defining the theoretically exact feature orientation in the active reference frame. The actual feature data is used to determine the extent of the tolerance zone.

p3 can only be XY, YZ, or ZX. This defines the engineering view that is to be used. Specification of an engineering view is required for feature types point, circle, and ellipse and defaults to the active working plane (points) or the plane of measurement (circles and ellipses) if parameter **p3** is not specified. Specification of an engineering view is allowed for feature types sphere, line, cylinder, and cone. For lines, cylinders, and cones the tolerance zone is a cylinder if **p3** is not included, and a pair of parallel planes if it is. For spheres the tolerance zone is a sphere if **p3** is not included, a circle otherwise. Specification of an engineering view is not allowed for feature type plane.

p4 is a value preceded by the letter "T" defining the width (diameter) of the tolerance zone.

p5 specifies the length of a tolerance zone for length dependent calculation of the deviation.

p6 specifies the width of a projected tolerance zone. **p6** can only be specified if the measured feature is a plane and **p5** has also been specified.

Results

Nominal feature position and orientation information is stored in the Dimensional Array for X, Y, Z, I, J, and K. The tolerance value is stored in the plus tolerance dimensional array for position (s(9,2) variable).

The positional deviation is computed as follows:

- For points and for the center coordinates of circles and ellipses, position deviation is computed as the diameter of the smallest cylinder containing the point and whose axis passes through the nominal feature position **p1** and whose axis is normal to the specified projection plane **p3**.
- For spherical features where no projection plane is specified, position deviation is computed as the diameter of the smallest sphere containing the center point and whose center is at the nominal feature position **p1**.
- For line features where a projection plane is specified, position deviation is computed as the smallest distance between two parallel planes containing the set of points measured for the line; these two planes symmetrically located with respect to the nominal line position **p1**, each parallel to the nominal line **p2**, and each normal to the specified projection plane **p3**. If a length of projected tolerance zone is specified **p5**, then the calculated value is the distance between two such planes containing the best-fit line over the region defined by the nominal position and the specified length.
- For features with axes (cylinders and cones) where a projection plane is specified, position deviation is computed as the smallest distance between two parallel planes containing that portion of the computed feature axis bounded by the set of points measured for the feature; these two planes symmetrically located with respect to the nominal axis position **p1**, each parallel to the nominal axis **p2**, and each normal to the specified projection plane **p3**. If a length of projected tolerance zone is specified **p5**, the calculated value is the distance between two such planes containing the best-fit feature axis over the region defined by the nominal position and the specified length.
- For line features where a projection plane is not specified, position deviation is computed as the diameter of the smallest cylinder containing the set of points measured for the line; the axis of the cylinder being defined by the nominal line position vector **p1** and orientation vector **p2**. If a length of projected tolerance zone is specified **p5**, then the calculated value is the diameter of the smallest cylinder containing the best-fit line over the region defined by the nominal position and the specified length.
- For features with axes (cylinders and cones) where a projection plane is not specified, position deviation is computed as the diameter of the smallest cylinder that contains that portion of the computed feature axis bounded by the set of points measured for the feature; the axis of this cylinder being defined by the nominal line position vector **p1** and orientation vector **p2**. If a length of projected tolerance zone is specified **p5**, the calculated value is the diameter of the smallest cylinder containing the best-fit feature axis over the region defined by the nominal position and the specified length.
- For plane features, position deviation is computed as the smallest distance between two parallel planes, containing the measured set of points; these two planes symmetrically located with respect to the nominal plane position **p1** and each parallel to the nominal plane **p2**. If a length and a width of projected tolerance zone is specified (**p5** and **p6**), the calculated value is the distance between two such planes containing the best-fit plane over the region defined by the nominal position and the specified dimensions. Width is measured toward the position end of the first named axis of the working plane (XY, YZ, or ZX) that is "most parallel" to the specified normal plane. Length is measured toward the positive end of the second named axis

of this working plane. If more than one working plane is equally parallel to the nominal, planes will be selected in the following order: XY then YZ then ZX.

The computed deviation is stored in the deviation column of the dimensional array for position (s(9,4) variable). This value is maintained in the array only until the next operation that updates the feature register position data.

For position is a specific engineering view, the following Inspection Report output is generated:

ISO POSITION: In XY plane
#3
Feature #10
Type: CIRCLE
Name:

	ACTUAL	NOMINAL
X	14.998	15.000
Y	-7.250	-7.500

#4

	ACTUAL	NOMINAL	+ TOL	- TOL	DEV	OUT-TOL
PS			0.010		0.012	0.002

For position in space, the following Inspection Report is generated:

ISO POSITION
#3
Feature #10
Type: PLANE
Name:

	ACTUAL	NOMINAL
X	14.998	15.000
Y	-7.250	-7.500
Z	0.002	0.000
I	0.000	0.000
J	0.001	0.000
K	1.000	1.000

#4

	ACTUAL	NOMINAL	+ TOL	- TOL	DEV	OUT-TOL
PS			0.010		0.012	0.002

For length dependent position in space the following inspection report is generated:

ISO POSITION
Projected Tolerance Zone: length=2.0, width=2.5
#3
Feature #10
Type: PLANE
Name:

	ACTUAL	NOMINAL
X	14.998	15.000
Y	-7.250	-7.500
Z	0.002	0.000
I	0.000	0.000
J	0.001	0.000
K	1.000	1.000

#4

	ACTUAL	NOMINAL	+ TOL	- TOL	DEV	OUT-TOL
PS			0.010		0.012	0.002

If MMC tolerancing has been specified, the first section of reported data includes diameter actual and nominal if the feature type is circle, cylinder, or sphere. In this case, the report also includes the following line:

MMC BONUS TOLERANCE = 0.010

Note that tolerance, deviation, and out-of-tolerance information is only printed for the position row of the report. The feature location and orientation actuals and nominals are printed for information purposes. Orientation values are not printed for feature types point, circle, ellipse, or sphere.

Skip Action

Ignored; not executed if part of a skip block.

Error Message

- MLB-042 The best-fit point data for the feature is required to make the requested evaluation but is no longer available.

I6 - Reporting Concentricity / Coaxiality Deviation

The I6 command computes and reports concentricity or coaxiality deviation for a feature. **I6** prints an ISO, ANSI, or DIN identification line with each report that reflects which standard is active.

Command Format

MLString "I6 **p1**,**p2**,**p3**,**p4**,**p5**"

Examples

1. MLString "I6 DP,T.01"

   Check the feature for concentricity to the datum point of the active reference frame. The diameter of the tolerance zone is 0.01.

2. MLString "I6 ZA,T.01"

   Check the feature for coaxiality to the Z-axis of the active reference frame. The diameter of the tolerance zone is 0.01.

3. MLString "I6 DP,XY,T.01"

   Check the feature for concentricity to the datum point in the XY plane of the active reference frame. The diameter of the tolerance zone is 0.01.

4. MLString "I6 ZA,T.01,1.0,2.0"

   Check the feature for length dependent coaxiality to the Z-axis of the active reference frame. The tolerance zone diameter is 0.01. The nominal start position along the reference axis is 1.0. The length of the projected tolerance zone is 2.0.

Input

- **p1** = DP, XA, YA, ZA or the name of a saved feature (Parametric variable or string; required).
- **p2** = XY, YZ, or ZX (Parametric variable; optional).
- **p3** = the letter T followed by the tolerance zone width (Numeric; required).
- **p4** = Nominal position (Numeric; optional).
- **p5** = Length of the tolerance zone (Numeric; optional).

p1 identifies the datum feature. This is either a parametric feature identifier (DP, XA, YA, or ZA) for the datum point or a datum axis, or the name of a saved feature. A named feature is used either as a datum point (feature types point, circle, sphere, or ellipse) or as a datum axis (feature types line, cylinder or cone). For length dependent calculations, **p1** must be a datum axis and may not be a named feature.

p2 can only be XY, YZ, or ZX. This defines the engineering view used in computing concentricity. An engineering view is required for feature types point, circle, and ellipse and defaults to the active working plane (points) or the plane of measurement (circles and ellipses) if parameter **p2** is not specified. An engineering view may be specified for feature type sphere; if not specified, evaluation defaults to a spherical zone. Parameter **p2** may not be specified for lines, cones, or cylinders.

p3 is a value preceded by the letter "T" defining the size (diameter) of the tolerance zone.

p4 specifies a nominal position along the named datum axis for length dependent calculation of coaxiality deviation. This may be specified only for feature types line, cone, and cylinder. If **p4** is given then **p5** must also be specified.

p5 specifies the length of the tolerance zone for length dependent calculation of coaxiality.

Results

Nominal feature position and orientation information is stored in the dimensional array for X, Y, Z, I, J, and K based upon the specified datum feature. The tolerance value is stored in the plus tolerance dimensional array for position (s(9,2) variable).

The concentricity or coaxiality deviation is computed as follows:

- For points, circles, and ellipses, concentricity is computed as the diameter of the smallest cylinder whose axis is coincident with the datum feature **p1**, is normal to projected plane **p2**, and contains the computed feature location.
- For spheres where a projection plane is specified, concentricity is computed as the diameter of the smallest cylinder whose axis is the datum feature **p1**, is normal to the projection plane **p2**, and contains the computed feature location.
- For spheres where a projection plane is not specified, concentricity is computed as the diameter of the smallest sphere located by the datum feature **p1** and which contains the computed feature location.
- For lines, coaxiality is computed as the diameter of the smallest cylinder whose axis is the datum axis **p1** and which contains the set of measured points.
- For cylinders and cones, coaxiality is computed as the diameter of the smallest cylinder whose axis is the datum axis **p1** and which contains that potion of the computed feature axis bounded by the set of measured points.
- For line and axis features, length dependent calculation is based upon the best-fit feature instead of the set of measured points. Deviation is calculated beginning at the specified nominal start point along the datum axis and continuing along the best-fit line or axis for the given length.

The computed deviation is stored in the deviation column of the dimensional array for position (s(9,4) variable). This value is maintained in the array only until the next operation which updates the feature register position data.

The following inspection report output is generated for concentricity (feature type point, circle, ellipse, or sphere):

ISO CONCENTRICITY to DP in XY plane
#3
Feature #10
Type: CIRCLE
Name:

	ACTUAL	NOMINAL
X	0.002	0.000
Y	-0.005	0.000

#4

	ACTUAL	NOMINAL	+ TOL	- TOL	DEV	OUT-TOL
PS			0.010		0.011	0.001

The following inspection report is generated for coaxiality (feature type cylinder, cone, or line):

ISO COAXIALITY to Z-axis
#3
Feature #10
Type: CYLINDER
Name:

	ACTUAL	NOMINAL
X	0.002	0.000
Y	-0.005	0.000
Z	5.732	0.000
I	-0.002	0.000
J	0.001	0.000
K	-1.000	1.000

#4

	ACTUAL	NOMINAL	+ TOL	- TOL	DEV	OUT-TOL
PS			0.010		0.011	0.001

The following inspection report is generated for length dependent coaxiality (feature type cylinder, cone, or line):

ISO COAXILITY to Z-axis
Projected Tolerance Zone: nominal position=1.0, length=2.0
#3
Feature #10
Type: CYLINDER
Name:

	ACTUAL	NOMINAL
X	0.002	0.000
Y	-0.005	0.000
Z	5.732	0.000
I	-0.002	0.000
J	0.001	0.000
K	-1.000	1.000

#4

	ACTUAL	NOMINAL	+ TOL	- TOL	DEV	OUT-TOL
PS			0.010		0.011	0.001

If MMC tolerancing has been specified, the first section of reported data includes diameter actual and nominal if the feature type is circle, cylinder, or sphere. In this case, the report also includes the following line:

MMC BONUS TOLERANCE = 0.010

Note that tolerance, deviation, and out-of-tolerance information is only printed for the position row of the report. The feature location and orientation actuals and nominals are printed for information purposes. Orientation values are not printed for feature types point, circle, ellipse, or sphere.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-037 Invalid feature type. Concentricity or coaxiality can't be computed for planes or slots/webs.
- MLB-042 The best-fit point data for the feature is required to make the requested evaluation but is no longer available.

I7 - Reporting Circular Runout Deviation

The I7 command computes and reports circular runout deviation for the feature in the Feature Register. I7 prints an ISO, ANSI, or DIN identification line with each report that reflects which standard is active.

Command Format

MLString "I7 **p1**,**p2**,**p3**,**p4**"

Examples

1. MLString "I7 ZA,T.01"

   Check the feature for circular runout to the Z-axis of the active reference frame. The inclination angle and the zone width are to be determined from the feature data as described below. The width of the zone is 0.01.

2. MLString "I7 XA,15,W.1,T.01"

   Check the feature for circular runout at an angle of 15 degrees relative to the X-axis. All points that fall within a circular cross-section 0.1 wide are to be treated as a subgroup for evaluating circular runout. The width of the tolerance zone is 0.01.

Input

- **p1** = XA, YA, ZA, or the name of a saved feature (Parametric variable or string; required).
- **p2** = Inclination direction (Numeric; optional).

- **p3** = W and width of cross-sectional zones (Numeric; optional).
- **p4** = T and width of tolerance zone (Numeric; required).

p1 identifies the datum feature, which is either a datum axis (XA, YA, or ZA) or the name of a saved feature. If a named feature is specified, it must be a line, cylinder, or cone to be used as the datum axis.

p2 is the amount of inclination of the nominal surface relative to the datum axis. The inclination angle defaults to zero degrees (radial runout) for circles, ellipses, cylinders, and cones. The inclination angle defaults to 90 degrees (axial runout) for planes.

p3 is the letter "W" followed by a numeric value. This is the width of the cross-sectional zones to be used in checking circular runout. The zone width is infinite for circles and ellipses whether **p3** is specified or not. For cylinders and cones the zone width defaults to three times the feature length divided by the number of points measured (i.e., averages three points per default cross-section). For planes, the zone width defaults to three times the maximum distance between any of the measured points and the datum axis divided by the number of points measured.

p4 is the value preceded by the letter "T" defining the width of the tolerance zone.

Results

The tolerance value is stored in the plus tolerance column of the dimensional array for runout (s(12,2) variable).

The circular runout deviation is computed as follows:

- For cylindrical features or circles (treated as single cross-section), circular runout is the maximum of the circular runout deviations for the set of circular cross-sections defined by the set of measured points and the width **p3**. The circular runout deviation for each cross-section is the smallest distance between two coaxial cylinders containing the points in that cross-section; these cylinders having axis **p1**.
- For planes, circular runout is the maximum of the axial runout deviations for the set of angular cross-sections defined by the set of measured points and the width **p3**. The axial runout deviation for each cross-section is the smallest distance between two parallel planes containing the points in that cross-section; these planes being normal to the datum axis **p1**.
- For cones, circular runout is the maximum of the circular runout deviations for the set of conical cross-sections defined by the set of measured points, the width **p3**, and the nominal angle **p2**. The circular runout deviation for each cross-section is the smallest distance between two coaxial cones containing the points in that cross-section; these cones have axis **p1** and included apex angle twice the nominal angle **p2**.

The computed value is stored in the dimensional array for runout (s(12,4) variable).

An Inspection Report such as the following is generated with the initial comment modified to reflect the command argument:

ISO CIRCULAR RUNOUT: to X-axis
#3
Feature #10
Type: CIRCLE
Name:

	ACTUAL	NOMINAL	+ TOL	- TOL	DEV	OUT-TOL
RO			0.010		0.011	0.001

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-037 Invalid feature type. Circular runout can't be computed for points, lines, or spheres.
- MLB-042 The best-fit point data for the feature is required to make the requested evaluation but is no longer available.

I8 - Reporting Symmetry Deviation

The I8 command computes and reports symmetry deviation for the feature in the Feature Register. I8 prints an ISO, ANSI, or DIN identification line with each report that reflects which standard is active.

Command Format

MLString "I8 **p1**,**p2**,**p3**,**p4**,**p5**"

Examples

1. MLString "I8 XY,T.01"

   Check the feature for symmetry to the XY datum plane of the active reference frame. The width of the tolerance zone is 0.01

2. MLString "I8 ZA,ZX,T.01"

   Check the feature for symmetry to the Z-axis in the ZX plane of the active reference frame. The width of the tolerance zone is 0.01.

3. MLString "I8 ZA,YZ,T.01,1.0,2.0"

   Check the feature for length dependent symmetry to the Z-axis in the YZ plane of the active reference frame. The width of the tolerance zone is 0.01. The nominal start position along the reference axis is 1.0. The length for the projected tolerance calculation is 2.0.

Input

- **p1** = (XA, YA, ZA, YZ, or ZX) or the name of a saved feature (Parametric or string; required).
- **p2** = XY, YZ, or ZX (Parametric; optional).
- **p3** = T and the width of tolerance zone (Numeric; optional).
- **p4** = Nominal position (Numeric; optional).
- **p5** = Length of tolerance zone (Numeric; optional).

p1 identifies the datum feature. This is either a parametric identified (XA, YA, ZA, XY, YZ, or ZX) or the name of a saved feature identifying a datum axis or plane. If a named feature is specified, this is treated as a datum plane if a plane is specified or as a datum axis if a line or an axis is specified.

p2 can only be XY, YZ, or ZX. This is the engineering view to be used to compute symmetry. Specification of an engineering view is required for line and axis feature types but is not allowed for feature type plane.

p3 is a value preceded by the letter "T" defining the width of the tolerance zone.

p4 specifies a nominal position along the named datum axis for length dependent calculation of symmetry deviation. This is for feature types line or axis only. **p4** may not be specified for feature type plane or if the specified datum is a named feature.

p5 specifies the length of the tolerance zone. If **p4** is specified, then **p5** is required. **p5** may not be specified for feature type plane or if the specified datum is a named feature.

Results

Nominal feature position and orientation information is stored in the dimensional array for X, Y, Z, I, J, and K based upon the specified datum feature. The tolerance value is stored in the dimensional array as the plus tolerance for position (s(9,2) variable).

The symmetry deviation is computed as follows:

- For line or axis features when **p4** and **p5** have been specified, then length dependent calculation is based on the best-fit feature instead of the set of measured points. Deviation is calculated beginning at the specified nominal start point along the datum axis and continuing along the best-fit line or axis for the given length.
- For line features when **p4** and **p5** have not been specified, symmetry is computed as the minimum distance between two parallel lines; each parallel to the datum axis **p1**, disposed symmetrically with respect to the datum axis, and which contain the set of measured points as projected into the engineering view **p2**.
- For features with axes when **p4** and **p5** have not been specified, symmetry is computed as the minimum distance between two parallel lines; each parallel to the datum axis **p1**, disposed symmetrically with respected to the datum axis, and which contain the set of measured points as projected onto the best-fit feature axis in the engineering view **p2**.
- For plane features, symmetry is computed as the minimum distance between two parallel planes, each parallel to the datum plane **p1**, disposed symmetrically with respect to the datum plane, and which contain the set of measured points.

The computed deviation is stored in the deviation column of the Dimensional Array for position (s(9,4) variable). This value is maintained in the array only until the next operation that updates the feature register position data.

The following Inspection Report outline is generated for symmetry (feature type cylinder, cone, or line):

ISO SYMMETRY to Z-axis in ZX plane
#3
Feature #10
Type: CYLINDER
Name:

	ACTUAL	NOMINAL
X	0.002	0.000
Y	-0.005	0.000
Z	5.732	0.000
I	-0.002	0.000
J	0.001	0.000
K	-1.000	1.000

#4

	ACTUAL	NOMINAL	+ TOL	- TOL	DEV	OUT-TOL
PS			0.010		0.011	0.001

The following inspection report is generated for length dependent symmetry (feature type cylinder, cone, or line):

ISO SYMMETRY to Z-axis in YZ plane
Projected Tolerance Zone: nominal position=1.0, length=2.0
#3
Feature #10
Type: CYLINDER
Name:

	ACTUAL	NOMINAL
X	0.002	0.000
Y	-0.005	0.000
Z	5.732	0.000
I	-0.002	0.000
J	0.001	0.000
K	-1.000	1.000

#4

	ACTUAL	NOMINAL	+ TOL	- TOL	DEV	OUT-TOL
PS			0.010		0.011	0.001

If MMC tolerancing has been specified, the first section of reported data includes diameter actual and nominal if the feature type is cylinder. In this case, the report also includes the following line.

MMC BONUS TOLERANCE = 0.010

Note that tolerance, deviation, and out-of-tolerance information is only printed for the position row of the report. The feature location and orientation actuals and nominals are printed for information purposes.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-037 Invalid feature type. Symmetric can't be computed for points, circles, ellipses, or spheres.
- MLB-042 The best-fit point data for the feature is required to make the requested evaluation but is no longer available.

JA - Projected Angularity

The JA command computes the angle between two features in a specified projection plane. The angle is an acute angle, expressed in degrees.

Command Format

MLString "JA P1,P2,P3"

Examples

1. MLString "JA F2,FR,YZ"

Input

- **P1** = Feature in FR, Stack or Storage (Parametric; optional).
- **P2** = Feature in FR, Stack or Storage (Parametric; optional).
- **P3** = XY, YZ, ZX, or the feature register (FR) or a stack position (F1 to F8), or a named feature (Parametric; optional).

P1 and P2 can identify features in the Feature Register, Feature Stack or Feature Storage. The features must be lines, cylinders, cones, or slots/webs. P2 may also be a parametric variable designating a coordinate frame axis (XA, YA, ZA). In the example, F2 refers to the feature in the second position of the Feature Stack. The FR refers to the feature in the Feature Register. If P1 and P2 are omitted, the command uses the features from FR and F1 by default.

P3 defines the plane onto which the features are projected. Only features of type "plane" may be specified. Omitting the P3 argument causes the command to use the current working plane for the projection. In the example, the command projects the features in F2 and FR onto the YZ plane and computes the angle formed by their intersection.

Results

The Dimensional Array stores the computed angle in S(7,0).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-031 The first parameter specifies an invalid type.
- MLB-032 The second parameter specifies an invalid type.

JD - Projected Distance

The JD command computes the distance between two features as if they had been projected onto the selected plane. This command uses normal geometric functions to calculate most distances. However, the following exception applies:

- The distance between two lines, skewed less than 5 degrees, is the shortest possible distance from the base point of the first line specified (P1) to the other line (P2).

Command Format

MLString "JD P1,P2,P3"

Examples

1. MLString "JD F6,F1,ZX"

Input

- **P1** = Feature in FR, Stack or Storage (Parametric; optional).
- **P2** = Feature in FR, Stack or Storage (Parametric; optional).
- **P3** = XY, YZ, ZX, or the feature register (FR) or a stack position (F1 to F8), or a named feature (Parametric; optional).

P1 and P2 can identify features in the Feature Register, Feature Stack or Feature Storage. P1 and P2 can be any measured or constructed feature except planes. P2 may refer to a reference frame axis (XA, YA or ZA) or the reference frame datum point (DP). If P1 and P2 are omitted, the command uses the data from FR and F1 by default. In the example, F6 refers to a feature in the sixth position of the Feature Stack. The F1 refers to a feature in the first position of the Feature Stack.

P3 defines the plane onto which the feature data is projected before the distance is computed. Only features of type "plane" may be specified if a saved feature name is used. Omitting P3 causes the command to use the current working plane by default. In the example, ZX selects the projection plane. This command projects F1 and F6 onto the ZX working plane and then measures the distance between the two projected features.

Results

The Dimensional Array stores the distance in S(8,0).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 The Feature Stack or Feature Register does not contain the required features.

- MLB-031 First feature specified is a plane.
- MLB-032 Second features specified is a plane.
- MLB-037 P3 is not a "plane."

JI - Projected Included Angle

The JI command computes the included angle between two features as if they had been projected onto the selected plane.

Command Format

MLString "JI P1,P2,P3"

Examples

1. MLString "JI"
2. MLString "JI F6,LINE1,ZX"
3. MLString "JI FR,XA"
4. MLString "JI PLANE16"

Input

- **P1** = Feature in FR, Stack or Storage (Parametric; optional).
- **P2** = Feature in FR, Stack or Storage (Parametric; optional).
- **P3** = XY, YZ, ZX, or the feature register (FR) or a stack position (F1 to F8), or a named feature (Parametric; optional).

P1 and P2 identify features, either in the feature stack or in named feature storage. P2 may also be a parametric variable designating a coordinate frame axis (XA, YA, ZA) or plane (XY, YZ, ZX). The parameters P1 and P2 must be both specified or both omitted. If both P1 and P2 are omitted, they default to FR and F1 respectively, as shown in Example 1. The features specified may be two planes, two line type features, or a line type feature and a plane. Note that features with axes such as cones, cylinders, slots, and webs are line type features.

P3 specifies a projection plane which must be one of the three possible working planes of the active reference frame, the feature register, a feature in the feature stack, or a saved named feature. Only features of type "plane" may be specified. If P3 is omitted, it defaults to the current working plane. The fourth example computes the angle between the features in the feature register and in the top stack position as though they had been projected into the plane PLANE16.

In Example 2, **JI** computes the included angle between the feature in stack position #6 and the feature "LINE1" in named feature storage as projected into the ZX plane. Example 3 computes the included angle between the feature in FR and the X-axis of the active reference frame.

Results

The Dimensional Array stores the computed angle in S(7,0). This angle is greater or equal to 0 and less than 180.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-030 Referenced features not defined.
- MLB-031 P1 not a line or a plane perpendicular to selected plane.
- MLB-032 P2 not a line or a plane perpendicular to selected plane.
- MLB-037 P3 is not a plane.

JL - Projected Parallelism

The JL command computes the deviation from parallelism between two features as if they had been projected onto the selected plane. The command calculates the angle between the features as in the **AN** command. The tangent of this angle is the result.

Command Format

MLString "JL P1,P2,P3"

Examples

1. MLString "JL FR,F4,ZX"

Input

- **P1** = Feature in FR, Stack or Storage (Parametric; optional).
- **P2** = Feature in FR, Stack or Storage (Parametric; optional).
- **P3** = XY, YZ, ZX, or the feature register (FR) or a stack position (F1 to F8), or a named feature (Parametric; optional).

P1 and P2 can identify features in the Feature Register, Feature Stack or Feature Storage. P2 may also be a parametric variable designating a coordinate frame axis (XA, YA, ZA). The features must be lines, cylinders, cones, slots, or webs. If both P1 and P2 are omitted, the command uses the data from FR and F1 by default. In the example, FR refers to the feature in the Feature Register. The F4 refers to the feature in the fourth position of the Feature Stack.

P3 defines the plane onto which the features are projected. If P3 is a saved feature it must be a plane. Omitting P3 causes the command to use the current working plane for the projection. In the example, the command projects the lines in FR and F4 onto the ZX working plane and calculates their deviation from parallelism.

Results

The Dimensional Array stores the deviation from parallelism in units per unit in the parallelism deviation slot, S(11,4).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-027 The angle between the two features exceeds 89 degrees.
- MLB-030 The Feature Stack or Feature Register do not contain the required variables.
- MLB-031 The first parameter specifies an invalid type.
- MLB-032 The second parameter specifies an invalid type.
- MLB-037 The third parameter doesn't specify a "plane."

JP - Projected Perpendicularity

The JP command computes the deviation from perpendicularity between two features as though they were projected onto the selected plane. The command calculates the angle between the features as in the **AN** command. The tangent of an angle of 90 degrees minus the calculated angle is computed.

Command Format

MLString "JP P1,P2,P3"

Examples

1. MLString "JP LN3,FR,YZ"
2. MLString "JP PLANE16"

Input

- **P1** = Feature in FR, Stack or Storage (Parametric; optional).
- **P2** = Feature in FR, Stack or Storage (Parametric; optional).
- **P3** = XY, YZ, ZX, or the feature register (FR) or a stack position (F1 to F8), or a named feature (Parametric; optional).

P1 and P2 can identify features in the Feature Register, Feature Stack or Feature Storage. P2 may also be a parametric variable designating a coordinate frame axis (XA, YA, ZA). The features must be lines, cylinders, cones, slots, or webs. In the first example, LN3 refers to a feature saved in Feature Storage. The FR refers to a feature in the Feature Register.

If both P1 and P2 are omitted, the command uses the data from FR and F1 by default.

P3 defines the plane onto which the features are projected. Omitting P3 causes the command to use the current working plane for the projection. In the first example, the command projects the features in LN3 and FR onto the YZ plane and computes the deviation from perpendicularity of the two lines.

In the second example, the features in the feature register and in the top stack position are projected into the plane PLANE16 and then the perpendicularity is computed.

Results

The Dimensional Array stores the computed deviation from perpendicularity in units/unit in the perpendicularity deviation slot, S(10,4).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-027 The angle between the two features is less than one (1) degree.
- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-031 The first parameter specifies an invalid type.
- MLB-032 The second parameter specifies an invalid type.
- MLB-037 The third parameter isn't a plane.

JX - Projected Axial Distance

The JX command computes the distance between two features as if they had been projected onto the selected plane. The command also computes the axial components of the distance and uses them to create a distance type feature. The axial components are returned to the X, Y and Z coordinate rows of the Dimensional Array. This command uses normal geometric functions to calculate most distances. However, the following exception applies:

- The distance between two lines, skewed less than 5 degrees, is the shortest possible distance from the base point of the first line specified (P1) to the other line (P2).

Command Format

MLString "JX P1,P2,P3"

Examples

1. MLString "JX PT5,PT9,YZ"
2. MLString "JX PLANE16"

Input

- **P1** = Feature in FR, Stack or Storage (Parametric; optional).
- **P2** = Feature in FR, Stack or Storage (Parametric; optional).
- **P3** = XY, YZ, ZX, or the feature register (FR) or a stack position (F1 to F8), or a named feature (Parametric; optional).

P1 and P2 can identify features in the Feature Register, Feature Stack or Feature Storage. P1 and P2 can be any measured or constructed feature except planes. P2 may refer to a reference frame axis (XA, YA or ZA) or the reference frame datum point (DP). If both P1 and P2 are omitted, the command uses the data from FR and F1 by default. In the first example above, PT5 refers to a feature saved in Feature Storage. The PT9 refers to a feature saved in Feature Storage.

P3 defines the plane onto which the features are projected. Only features of the type "plane" may be specified if P3 is a named saved feature. Omitting P3 causes the command to use the current working plane for the projection. In the first example, the distance is computed between the saved features PT5 and PT9 as projected onto the YZ plane. The axial components of this distance are also computed and stored in the Feature Register.

In the second example, two temporary features are created by projecting the features in the feature register and in the top stack position into the plane PLANE16 and then computing the distance between these temporary features.

Results

A distance type feature is created and pushed into the Feature Register/Feature Stack. The feature type is set to distance (19). The Dimensional Array stores:

- Distance in S(8,0),
- Axial components of the distance in the X-, Y-, and Z-coordinate rows.

The signs of the axial components assume a vector from the first feature to the second feature. Data is valid only if the reference remains unmoved after calculating distance.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-030 The Feature Stack or Feature Register does not contain the required features.
- MLB-031 The first feature specified is a plane.
- MLB-032 The second feature specified is a plane.
- MLB-037 The third parameter isn't a plane.

PA - Parallelism

The PA command computes the deviation from parallelism between two features. The command calculates the angle between the features as in the **AN** command. The tangent of this angle is the result.

Command Format

MLString "PA P1,P2"

Examples

1. MLString "PA F3,LN7"

Input

- P1 = Feature in FR, Stack or Storage (Parametric; optional).
- P2 = Feature in FR, Stack or Storage (Parametric; optional).

P1 and P2 can identify features in the Feature Register, Feature Stack or Feature Storage. P2 may also be a parametric variable designating a coordinate frame axis (XA, YA, ZA) or plane (XY, YZ, or ZX). The features must be lines, cylinders, cones, slots, webs, or planes. If both P1 and P2 are omitted, the command uses the data from FR and F1 by default. In the example, F3 refers to the feature in the third position in the feature stack. The LN7 refers to a feature in named Feature Storage.

Results

The deviation of the two features from perfect parallelism is computed in terms of units per unit runout. The Dimensional Array stores the deviation in the parallelism deviation slot, S(11,4).

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-027 The angle between the two features exceeds 89 degrees.
- MLB-030 The Feature Stack or Feature Register does not contain the required features.
- MLB-031 The first parameter specifies an invalid type.
- MLB-032 The second parameter specifies an invalid type.

PE - Perpendicularity

The PE command computes the deviation from perpendicularity between two features. The command calculates the angle between the features as in the **AN** command. The tangent of 90 degrees minus this angle is the result.

Command Format

MLString "PE P1,P2"

Examples

1. MLString "PE F3,LN7"

Input

- P1 = Feature in FR, Stack or Storage (Parametric; optional).
- P2 = Feature in FR, Stack or Storage (Parametric; optional).

P1 and P2 can identify features in the Feature Register, Feature Stack or Feature Storage. P2 may also be a parametric variable designating a coordinate frame axis (XA, YA, ZA) or plane (XY, YZ, or ZX). The features must be lines, cylinders, cones, slots, webs, or planes. If both P1 and P2 are omitted, the command uses the data from FR and F1 by default. In the example, F3 refers to the feature in the third position in the feature stack. The LN7 refers to a feature in named Feature Storage.

Results

The deviation of the two features from perfect perpendicularity is computed in terms of units per unit runout. The Dimensional Array stores the deviation in the perpendicularity slot, S(10,4).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-027 The angle between the two features is less than one degree.
- MLB-030 The Feature Stack or Feature Register do not contain the required features.
- MLB-031 The first parameter specifies an invalid type.
- MLB-032 The second parameter specifies an invalid type.

12. Deviation Out-of-Tolerance Group

MLB commands in the Deviation Out-of-Tolerance Group provide a way of computing deviation and out-of-tolerance values within the Dimensional Array. You specify the tolerancing mode before you call for evaluations.

The Deviation Out-of-Tolerance Group includes the following MLB commands.

BL	**Bilateral Tolerancing** specifies that the system uses bilateral tolerancing for evaluations.
DT	**Deviation/Tolerances** computes the deviation and out-of-tolerances for a particular row in the Dimensional Array. It does not produce a printed report.
DV	**Set a Dimensional Array Value**
LM	**Limit Tolerancing** specifies that the system uses limit tolerancing for evaluations.
MM	**Maximum Material Condition** specifies that the system uses the principles of maximum material condition when computing true position.
PD	**Position Deviation** computes the deviations and out-of-tolerances for the rectangular and polar coordinates of the part position. It does not produce a printed report.
RS	**Regardless of Feature Size**
SZ	**Size Deviation/Tolerancing** sets the mode of the size deviation calculation. This setting affects how the deviation calculation is performed when bilateral tolerancing is active.
TP	**True Position** computes the deviations and out-of-tolerances for the rectangular and polar coordinates of the part position. it does not produce a printed report.

BL - Bilateral Tolerancing

The BL command specifies that dimensions are to be positive or negative values about an entered nominal. This is the default state at system initialization.

Command Format

MLString "BL"

Examples

MLString "BL"

Input

None.

Results

The system clears the limit tolerance flag (see the **LM** command). Inspection report deviation and out-of-tolerance values are computed from the specified nominal and plus- and minus-tolerance values.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

DT – Deviation / Tolerances

The DT command computes the deviation and out-of-tolerance for a particular row in the Dimensional Array or the User Array.

Command Format

MLString "DT P1,P2"

Examples

1. MLString "DT 13"

Input

- **P1** = 0 through 17 (Numeric; required).
- **P2** = U (String; optional).

P1 specifies a row in the Dimensional Array. P1 must be a number from 0 to 17. The command uses the actual, nominal and tolerance values and the appropriate internal flags and compute the corresponding deviation and/or out-of-tolerance values for the specified row. In the example, 13 specifies that the command evaluate the values in row 13, the form row, of the Dimensional Array.

P2 can only be the letter 'U'. If it is specified then the **DT** command computes deviation and/or out-of-tolerance values for the specified row in the User Array rather than in the Dimensional Array. In this case P1 is restricted to the range 0 to 14. See the **UA** command for further information.

Results

The command calculates the deviation and out-of-tolerance values for the specified row. The Dimensional Array or the User Array stores the deviation and out-of-tolerance values. An inspection report is not printed.

If P1 specified 9 (Position) and MMC is in effect then the amount of bonus tolerance is returned in variable S7.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-021 First parameter invalid.
- MLB-022 Second parameter invalid.

DV - Set a Dimensional Array Value

The DV command assigns a specified value to an element of the Dimensional Array.

Command Format

MLString "DV P1,P2,P3"

Examples

1. MLString "DV 2,1,25.4"
2. MLString "DV 8,3,.01"

Input

- **P1** = Specific row in dimensional array (Numeric; required).
- **P2** = Specific column in dimensional array (Numeric; required).
- **P3** = Value to insert in dimensional array at the location indicated by P1 and P2 (Numeric; required).

Results

The value of P3 is stored in dimensional array element (P1,P2).

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-021 First parameter invalid.
- MLB-022 Second parameter invalid.
- MLB-028 Insufficient number of parameters.

LM - Limit Tolerancing

The LM command specifies that in tolerance dimensions are to defined by maximum and minimum values. This is the default state at system initialization.

Command Format

MLString "LM"

Examples

1. MLString "LM"

Input

None.

Results

The system sets the limit tolerance flag. Subsequent evaluation of data in the Dimensional Array (such as in **RP** command execution) uses maximum and minimum values.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

MM - Maximum Material Condition

The MM command activates the principles of maximum material condition when computing true position out-of-tolerance.

Command Format

MLString "MM"

Examples

1. MLString "MM"

Input

None.

Results

The system sets the maximum material flag. Subsequent calculations of out-of-tolerance conditions for position on features-of-size will take into account any bonus tolerance derived from the feature's departure from its maximum material condition.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

PD - Position Deviation

The PD command computes deviations and out-of-tolerances for the rectangular and polar coordinates of a part's position. The values in the actuals, nominals and tolerance columns of the Dimensional Array are used for the computations.

Command Format

MLString "PD"

Examples

1. MLString "PD"

Input

None.

Results

The deviation and out-of- tolerance values for X, Y, Z, R, and A are computed and stored in the Dimensional Array. An inspection report is not printed.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

RS - Regardless of Feature Size

The RS command specifies that position out-of-tolerance is to be computed "regardless of feature size". This means that no bonus tolerance is allowed as the feature size varies.

Command Format

MLString "RS"

Examples

1. MLString "RS"

Input

None.

Results

The system resets the maximum material flag. Subsequent calculations of out-of-tolerance conditions for position on features-of-size will not take into account any bonus tolerance derived from the feature's departure from its maximum material condition.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

SZ - Size Deviation Tolerancing Mode

The SZ command sets the "size" mode of deviation calculation. This setting effects how the deviation calculation is performed when bilateral tolerancing is active.

When the "size" mode is active, the calculated deviation is further from the datum, the tolerance is a positive value; when the calculated deviation is closer to the datum, the tolerance is a negative value. Note: When **SZ** is ON, it is necessary to enter minus tolerances with a minus sign (-). For example, enter -.005 as minus five thousandths.

Command Format

MLString "SZ P1"

Example

1. MLString "SZ ON"

Input

- **P1** = ON or OFF (String; optional)

P1 specifies whether size deviation tolerancing is ON or OFF. In the example size deviation tolerancing is ON.

Results

When **SZ** tolerancing is OFF, deviations are calculated as (actual - nominal). If **SZ** tolerancing is ON and the nominal value is less than 0, then the deviation calculated as (nominal - actual). If **SZ** tolerancing is ON and the nominal value is greater than or equal to 0, then deviation is calculated as (actual - nominal).

When **SZ** tolerancing is ON, the "graphic illustration for percentage of tolerance zone" is omitted from the inspection report. Note: The system automatically issues an **ID** command every time the **SZ** command is used to set the size deviation mode of calculation. Depending on the new setting, either "Size Deviation On" or "Size Deviation Off" prints on the inspection report.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

TP - True Position

The TP command computes the deviation and out-of-tolerance values for position, radius, and diameter in the Dimensional Array. The command uses diameter deviation and tolerance mode settings in computing position.

Command Format

MLString "TP"

Example

1. MLString "TP"

Input

None.

Results

Deviation and out-of-tolerance values are computed for the position row of the Dimensional Array. This computation is affected by the maximum material mode, the feature type, and the actual and nominal data for the feature in the Feature Register.

The Dimensional Array stores:

- The deviation value for radius,
- The deviation value of diameter,
- The out-of-tolerance values for radius,
- The out-of-tolerance value for diameter, and
- Position deviation and out-of-tolerance values.

An inspection report is not printed. This same evaluation takes place during the execution of an **RP** command that reports position data.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

13. Inspection Report Group

MLB commands in the Inspection Report Group let you format and print inspection reports.

The Inspection Report Group includes the following MLB commands.

FM	**Define Format** defines and stores an Inspection Report format.
HF	**Report Header Off** skips generation of the Inspection Report header.
ID	**Identification Data** outputs a literal string to the Inspection Report and/or Log.
NEWRPT	**New Report** is used to close and print a report file.
OT	**Off TempComp** suppresses the reporting of TempComp messages.
RP	**Report** computes deviation and out-of-tolerance values in the Dimensional Array and prints a formatted report of the requested data.

FM - Define Format

The FM command defines and stores one of one hundred inspection report formats. Use this command to redefine recallable formats. When the **RP** command selects one of these formats in a program line, it is copied into format 0.

Command Format

MLString "FM P1,P2"

Examples

1. MLString "FM 9,#?&:X,Y,Z,DI,FM"

Input

- **P1** = 0 to 99 (Numeric; required).
- **P2** = Print elements (String; optional).

P1 identifies the format to store and serves as the index used when recalling that format. P1 must be a number from 0 to 99. In the example, format 9 is being saved.

P2 defines a format by listing the elements of the print format. See the **RP** command for a description of the format string structure. If P2 is omitted, the system copies Format 0 to the saved format number specified in P1. In the example, P2 is:

#?&:X,Y,Z,DI,FM

When the program later prints the inspection report using format number 9, the report contains:

- Feature type,
- Feature sequence number,
- Feature number,
- X, Y and Z axis coordinates,
- Feature diameter, and
- Feature form.

Note that this format does not contain values for the column headings. The system prints the column headings already stored in that print format. Note: If P1 is 0 and P2 defines a format, then Format 0 is redefined. If the format is redefined in this manner and then a report is printed with any other format using the **RP** command, the redefined format is lost. When P1 is 0, the format is not saved. The **RP** command copies the named format into Format 0 in order to use it.

Results

The specified format is defined.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

HF - Report Header Off

The HF command disables generation of the standard report header.

Command Format

MLString "HF"

Examples

1. MLString "HF"

Input

The **HF** command disables generation of the standard report header. The **HF** command must be executed prior to any **ID** or **RP** commands. Note that the header printout is only disabled until the next part program initialization (see **IL** command).

Results

The standard report header is not generated.

Skip Action

Executed normally.

Error Messages

None.

ID - Identification Data

The ID command sends a literal string to the inspection report. This string is also sent to the log file if logging status is ON. A special form of this command can also be used to insert a graphic image into a report.

Command Format

MLString "ID P1"

Examples

1. MLString "ID RIGHT HOLE IN FRONT"
2. MLString "ID ~BC:\BITMAPS\LOGO.BMP,C"

Input

- **P1** = Max. of 117 characters (String; optional).

P1 is either an alphanumeric string or the name of a graphics file that is sent to the printer and/or log file. The maximum length of P1 is 117 characters. A graphics file is identified by the first two characters of P1 being "~B"; all other forms of P1 are treated as simple string constants. In the first example, the part program sends the string constant RIGHT HOLE IN FRONT to the printer and/or the log file.

If the first two characters of P1 are "~B" then the rest of the string is treated as a file name together with a placement indication. The file name must specify the drive and full path. The graphics file can be a bitmap (.bmp), Windows metafile (.wmf) or enhanced metafile (emf). Horizontal position of the graphics image can be specified by following the file name with a comma and the character "L", "C" or "R" to indicate Left, Center or Right alignment. Default alignment is at the left margin. In the second example the bitmap "C:\BITMAPS\LOGO.BMP" is to be centered on the report.

A formfeed is inserted into the report when P1 is "~F". This causes the next line of the report to be printed on the next page of the printer. Any characters following "~F" are ignored.

Note that all text, including semi-colons and '@' signs that occur after the space following the **ID** command are treated as part of the P1 argument. This means that no other commands can be concatenated onto the same line as an ID command.

Results

If the current logging status is ON, the part program sends P1 to the log file. If the current printer status is ON, the part program sends P1 to the printer device. If P1 is omitted, the part program sends a blank line to the printer and the log file.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

None.

NEWRPT - New Report file

The NEWRPT command is used to close and print a report file. This command closes an open report file, sends it to the printer, and then opens a new report file.

Command Format

MLString "NEWRPT"

Examples

1. MLString "NEWRPT"

Input

None.

Results

The currently opened report file is closed and spooled to the printer. All data reported after this command goes into a new report file.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

OT - Disable TempComp Notices

The OT command disables the notices for TempComp.

Command Format

MLString "OT"

Examples

1. MLString "OT"

Input

None.

Results

TempComp notices are disabled.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

RP - Print Report

The RP command computes deviation and out-of-tolerance values in the Dimensional Array and prints a formatted report of the requested data.

Command Format

MLString "RP P1"

Examples

1. MLString "RP"
2. MLString "RP 5"
3. MLString "RP ALL"

Input

- **P1** = 0 to 99 or <format string> or ALL (Numeric/string; optional)

P1 is either a string defining the desired report format, a number (0 to 99) referring to a previously define report format (see **FM** command), or ALL (the predefined report format "#*?%ANPMDO:R,A,X,Y,Z,DI,RD,AN,DS,TP,PE,PA,RO,FM,I,J,K,MR").

In defining a new report format, the format string consists of three sections: a control section, a column definition section, and a row definition section. The control section occurs first and may contain the following characters:

- # - print the sequence number.
- ? - print the feature type.
- * - print the feature number.
- % - print the feature name.
- & - do not print the column headings.
- ! - generate the position graphic.
- & - do not print the sequence number, feature type, feature number, or feature name.

The column definition section defines the columns, including order, of data to be printed. It may contain the following characters:

- A - Actual values
- N - Nominal values
- P - Plus tolerance or maximum values (for limit tolerancing)
- M - Minus tolerance or minimum values (for limit tolerancing)
- D - Deviation values
- O - Out-of-tolerance values
- B - A blank column

The row definition section defines the rows, including order, of data to be printed. It may contain the following items separated by commas:

- X - X-axis coordinate
- Y - Y-axis coordinate
- Z - Z-axis coordinate
- R - Polar radius coordinate
- A - Polar angle coordinate
- A' - Polar angle coordinate in degrees/minutes/second format
- DI - Feature diameter (or width)
- RD - Feature radius (or minor radius)
- DS - Distance

- AN - Angularity data
- AN' - Angularity data in degrees/minutes/second format
- PA - Parallelism data
- PE - Perpendicularity data
- TP - Position data
- RO - Runout data
- I - Direction cosine relative to the X-axis
- J - Direction cosine relative to the Y-axis
- K - Direction cosine relative to the Z-axis
- MR - Major radius data (or length)
- FM - Feature form data

Format 0 is the 'active' format. This is the format used by the last **RP** command. If P1 contains an & and the P1 argument contains a $, the $ overrides the &.

Nonexistent values and zero out-of-tolerance values are blank in the printed report. The rows and columns of the inspection report print in the order they appear in the command string.

In the first example, there is no argument in the command, and the part program prints a report following the format in Format 0. In the second example, 5 identifies the stored report format. The part program generates a printed report according to this format.

If P1 is omitted, the report is printed using Format 0. If P1 identifies a stored report format with a number from 0 to 99, that report format is copied into Format 0.

Skip Action

Special processing. The command is ignored, but sequence number S6 is incremented.

Results

The deviation and out-of-tolerance values for the elements in the format are computed. The requested report is generated and sent to the Report Monitor.

Error Messages

None.

14. User Interface Group

MLB commands in the User Interface Group provide interfaces between the user and the Coordinate Measuring System.

The User Interface Group includes the following MLB commands.

BZ	**Beeper** controls the beepers in the RCU and MP.
DM	**Display Mode** returns the current status of the display update to variable S7.
DN	**Display On** turns the display update on.
DO	**Display Off** turns the display update off.
DP	**MP Display** displays alphanumeric information on the MP.
FD	**Display Format** specifies the display format used on the MP.
FT	**Fetch** returns the value of a parametric variable to a variable for examination.
G$	**Get Key** allows the user to set up RCU-I keys as acceptable responses to a prompt.
GN	**Get Number** asks the user to enter a number on the RCU-I keypad.
MT	**Monitor MP** monitors the system variables displayed on the MP.
OW	**Operator Wait** prompts the user to press the footswitch or the appropriate key on the RCU.
P$	**Panel** gives the part program access to the MP-30/35 front panel buttons.
PF	**Polar Coordinates** causes the MP to display polar part coordinates.
PM	**Prompt** displays a line of alphanumeric information on the CRT and/or RCU.
RD	**Rectangular Coordinates** causes the MP to display rectangular part coordinates.

BZ - Beeper

The BZ command controls the beeper devices in the RCU and the MP.

Command Format

MLString "BZ P1,P2"

Examples

1. MLString "BZ 3,10"

Input

- **P1** = Ignored.
- **P2** = Max. is 255 (Numeric; optional).

P1 used to specify which device to sound: 1 = MP, 2 = RCU, 3 = MP and RCU. The current system ignores this parameter and always emits a sound at the MP and RCU.

P2 specifies the number of beeps to sound. Any number less than 256 is acceptable. In the example, the RCU and the MP sound 10 beeps when the program executes this command.

Results

The beepers are sounded the specified number of beeps.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

None.

DM - Display Mode

The DM command returns the current status of the display of current CMM XYZ position update to the S7 variable. See also the **DN** and **DO** commands.

Command Format

MLString "DM"

Examples

1. MLString "DM"

Input

None.

Results

This command does not affect whether the XYZ position display is actively updated. Variable S7 is set as follows: 0 signifies that display update is off; 1 signifies that display update is on.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-020 No parameters permitted.

DN - Display On

The DN command turns the dynamic CMM position display update ON. During normal operations, coordinate information being displayed to reflect the current CMM probe position is updated eight times per second. Updates reflect probe motion, rotary table motion or changes in the reference frame. Use the **DO** (Display OFF) command to disable updating. This may result in faster part inspection by freeing system resources needed for required calculations. ON is the default state of the display update after part program initialization.

Command Format

MLString "DN"

Examples

1. MLString "DN"

Input

None.

Results

This command enables the dynamic display of CMM coordinate data. This command does not affect error message display.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-020 No parameters permitted.

DO - Display Off

The DO command turns the dynamic CMM position display update OFF. During normal operations, coordinate information being displayed to reflect the current CMM probe position is updated eight times per second. Updates reflect probe motion, rotary table motion or changes in the reference frame. The **DO** command disables this updating. This may result in faster part inspection by freeing system resources needed for required calculations. ON is the default state of the display update after part program initialization.

Command Format

MLString "DO"

Examples

1. MLString "DO"

Input

None.

Results

This command disables the dynamic display of CMM coordinate data. This command does not affect error message display. The current coordinate display "freezes." The system turns update on again when the **DN**, **IL**, or **IS** commands are used or when power is cycled on the MP.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-020 No parameters permitted.

DP - MP Display

The DP command displays alphanumeric information in place of the normal dynamic CMM position data.

Command Format

MLString "DP @P1P2@"

Examples

1. MLString "DP @2PLANE@"

Input

- **P1** = 1 through 4 (Numeric; optional).
- **P2** = Max. of 80 characters (String; optional).

P1 represents the line number on the XYZ display to be used for the prompt message. If the first character specified is not 1, 2, 3, or 4 then the line number defaults to 1 (one).

P2@ can contain as many as 80 alphanumeric characters but once the display line is full, the remaining characters are ignored. The entire P1P2 entry must be enclosed with two @ symbols. In the example, 2PLANE is the combined P1 and P2 argument. The 2 indicates that the word 'PLANE' is to be displayed on the second display line.

Results

The specified data is displayed on the specified display line. Any character that cannot be displayed is replaced with a space. After the display line is full, any remaining characters are ignored. If P2 is omitted, the display line is cleared.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

None.

FD - Display Format

The FD command specifies the format of numerical values displayed to reflect CMM position and printed on inspection reports.

Command Format

MLString "FD P1,P2"

Examples

1. MLString "FD 5,4"

Input

- **P1** = Number of digits to right of decimal for linear data (Numeric; optional).
- **P2** = Number of digits to right of decimal for angular data (Numeric; optional).

P1 specifies how many digits are to appear to the right of the decimal point for linear data. P1 must be no less than 1 and no more than 7. In the example, 5 indicates that 5 digits appear to the right of the decimal point for linear data.

P2 specifies how many digits are to appear to the right of the decimal point for angular data. P2 must be no less than 1 and no more than 7. In the example, 4 indicates that 4 digits appear to the right of the decimal point for angular data.

Results

Following execution of this command machine position and feature data is displayed and/or printed using the specified data resolution. At system initialization, the system default assigns decimal places for linear data and angular data based upon the CMM configuration. Execution of an **IN**, **ME**, **IL**, or **IS** command will reset the data resolution settings to match the CMM configuration.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

FT - Fetch

The FT command returns numeric data associated with the specified system parametric variable to the system variables. The parametric variable must contain valid data. The command does not allow part reference frame parametric variables.

Fetch returns data that contains one numeric value to the variable S7. This includes such data as a single coordinate on the Feature Stack or the probe radius. Fetch returns variables that specify a coordinate position to the Feature Register and the Dimensional Array. These can be the contents of the Feature Register, Feature Stack or Feature Storage. If the coordinate value is somewhere other than in the Feature Register, the current contents of the Feature Register move into the Feature Stack and the Feature Register stores the fetched feature data. The system overwrites the current contents of F8 if the stack is full.

Command Format

MLString "FT P1"

Examples

1. MLString "FT PR"
2. MLString "FT F2"
3. MLString "FT S5"

Input

- P1 = string - required

P1 is a string that represents:

- Data in the Feature Register,
- Position in the Feature Stack,
- Feature in Feature Storage, or
- Another parametric variable.

P1 specifies the parametric variable for which data is to be retrieved. If it represents a single value then this is returned to variable S7. If it represents a feature then the data for this feature is pushed into the feature register/stack.

The special strings SR, S1 through S8, and M1 through M100 are used to quickly check whether a specified stack or storage location has valid data. String SR represents the feature register; strings S1 through S8 represent feature stack positions 1 through 8; strings M1 through M100 represent feature storage locations. In this case, the system returns the value 0 (empty) or 1 (not empty) to the variable S7.

In the first example, the system returns the contents of the parametric variable PR to S7. In the second example, the system copies the contents of F2 into FR, and F1 through F7 shift down one stack position. And in the third example, S5 represents the data in Feature Stack position #5 (F5).

Results

Variable S7 stores single values such as probe radius PR and NPR, and single coordinate data ...X, X1, Y1, etc. The system can also return the validity (status) of data in a specified Feature Storage location to system variable S7 (empty=0, valid=1). If requested data is valid and P1 is SR, S1,...,S8 or M1,...,M100:

- S7 = 1,
- Feature name returns to the S0D variable.
- Feature number returns to the S8 variable, and
- Feature type returns to the S10 variable.

If valid data does not exist in the specified location:

- S7 is 0
- S8, S0D (S0D), and S10 remain unchanged.

If the data is valid and the argument is FR, F1,...,F8:

- Current contents of the Feature Register move down,
- Feature Register stores the feature data,

The Dimensional Array also stores the data.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-030 The Feature Stack or Feature Register do not contain the required data.

G$ - Get Operator Response from RCU-I

The G$ command is used to prompt the operator to press one of a specified set of keys at the RCU-I. The RCU-II does not support this command.

Command Format

MLString "G$ @P1@,@P2@"

Examples

1. MLString "G$ @01@,@Are probes calibrated?@"
2. MLString "G$ @0123456789,=ABCDEFGHIJKLRSV@"

Input

- **P1** = List of legal key codes; (String; required)
- **P2** = Maximum of 80 characters (String; optional)

P1 specifies a list of RCU-I keys that may be used to respond to the command. The legal key codes are 0 through 9 (number keys), comma (+/- key), = (+/Enter key), A (+/up arrow), B (Menu Fctn), C (Calc), D (Drive), E (-/Exit), F (Nom Entry), G (Print), H (Display), I (Tip select), J (Tol entry), K (Tip Cal), L (Clear), P (Record), R (Off Line), S (*/Select), and V (-/down arrow).

P2 is a prompt to be displayed for the operator. If P2 is not specified then the operator prompt defaults to "ENTER KEY".

In the first example above the message "Are probes calibrated?" is displayed at the RCU until the user presses either the 0/No or 1/Yes key. In the second example the default prompt "ENTER KEY" is displayed at the RCU until the operator presses any of the legal keys.

Results

The specified prompt is displayed at the RCU-I and operations are suspended until the operator responds by pressing one of the specified keys. When a key is pressed, the character corresponding to that key is returned to variable S0D.

Skip Action

Ignored. Not executed if within a block of code being skipped.

Error Messages

- MLB-050 RCU is not available.

GN - Get Number from RCU-I

The GN command asks the operator to enter a number on the RCU-I. The RCU-II does not support this command.

Command Format

MLString "GN @P1@"

Examples

1. MLString "GN @ENTER NUMBER OF POINTS TO RECORD@"

Input

- **P1** = Maximum of 80 characters (String; optional)

P1 is the operator prompt. It can contain as many as 80 alphanumeric characters. Enclose the P1 message with the @ symbols. If the prompt argument is not specified, the RCU-I displays the message ENTER NUMBER.

In the example, @ENTER NUMBER OF POINTS TO RECORD@ is the P1 argument. @...@ encloses P1. ENTER NUMBER OF POINTS TO RECORD is the message that scrolls on the RCU-I.

Results

The RCU-I scrolls the entered or default message. The system pauses until the user enters the number by pressing number keys, decimal point, sign and ending with the ENTER key. The entered value is returned to the part program in variable S7.

Skip Action

Ignored. Not executed if within a block of code being skipped.

Error Messages

- MLB-050 RCU is not available.

MT - Monitor MP

The MP command requests that the system's XYZ current position display show one of X, Y, Z, W, polar radius, or polar angle on a specified display row.

Command Format

MLString "MT @P1P2@"

Examples

1. MLString "MT @4X@"

Input

- **P1** = 1 or 2 or 3 or 4 (Numeric; required).
- **P2** = X or Y or Z or W or R or A (String; required).

P1 is a number in the range of 1 to 4 indicating which display row is to be modified.

P2 is a single character identifying which data is to be displayed on row P1. It may be X, Y or Z to designate a particular part coordinate axis; W to designate the rotary table angle; R for polar radius; or A for polar angle.

Results

The display contents are modified to reflect the new specification. The requested data is updated continuously to reflect new probe and/or table positions or part reference frame changes. Once selected, this display mode remains in effect until another mode is selected or a part program or system initialization occurs.

Skip Action

Not executed if part of a skipped block of code.

Error Messages

None.

OW - Operator Wait

The OW command prompts the operator to press the Record button on the RCU or the footswitch.

Command Format

MLString "OW @P1@"

Examples

1. MLString "OW @PRESS RECORD KEY WHEN READY@"

Input

- **P1** = Maximum of 80 characters (String; optional).

P1 is the message to be displayed for the operator. The prompt message can contain as many as 80 alphanumeric characters. Enclose the message with the @ symbols. If a prompt argument is not specified, the prompt WAITING FOR USER is used by default.

If the first two characters following the first '@' are "~C", then the Prompt window will be cleared before displaying the remainder of the message.

In the example, PRESS RECORD KEY WHEN READY is the P1 argument. PRESS RECORD KEY WHEN READY is the message that appears on the RCU and/or the CRT.

Results

The prompt appears on the CRT and RCU displays. Part program operations pause until the operator responds.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

P$ - Get Operator Input from MP Panel

The P$ command is used to check for operator input from the MP-30/35 front panel buttons. The SMP systems do not support this command.

Command Format

MLString "P$ P1"

Examples

1. MLString "P$ TC"
2. MLString "P$"

Input

- **P1** = List of legal terminating key codes; (String; optional)

P1 specifies a list of MP-30/35 front panel buttons that are to be used by the operator to terminate the input. The legal key codes are 0 through 9 (number keys), X, Y, Z, W, F (Command key), T (Enter key), D (D key), x (X reset), y (Y reset), z (Z reset), w (W reset), I (I/M key), and C (Clear key). At initialization the list of front panel terminators is set to 'xyzwICT'. If P1 is specified then the list of terminators is reset. If P1 is not specified then the system checks for a completed entry.

Results

If P1 is specified then the MP-30/35 resets the list of keystrokes that define a completed entry. If P1 is not specified and the operator has completed an entry then the complete entry is returned in variable S0D. Otherwise S0D is returned as a null string. Note that this command always returns immediately whether an operator keystroke entry is available or not. To wait for a user entry the part program must include a loop with repeated execution of **P$** to check for a completed entry.

Skip Action

Ignored. Not executed if within a block of code being skipped.

Error Messages

None.

PF - Polar Coordinates

The PF command causes the system XYZ probe position display to show polar part coordinates. Radius and angle appear in the top two display lines. The axis normal to the working plane appears on the third display line.

Command Format

MLString "PF"

Examples

1. MLString "PF"

Input

None.

Results

Polar data display mode is activated. This display mode is continuous until another mode is selected. The system updates the display to reflect probe and/or table motion.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

PM - Prompt

The PM command displays a line of alphanumeric information on the RCU and/or the Prompt Monitor. If the prompt exceeds the width of the RCU display, the RCU scrolls the message. A special form of this command can also be used to display a graphic image.

Command Format

MLString "PM @P1@"

Examples

1. MLString "PM @MEASURE TOP CENTER BORE@"
2. MLString "ID ~BC:\BITMAPS\LOGO.BMP,R"

Input

- **P1** = Maximum of 80 characters (String; optional).

P1 is the prompt message and can include as many as 80 alphanumeric characters. It must be enclosed by the @ symbols at each end. In the example, @MEASURE TOP CENTER BORE@ is the prompt. MEASURE TOP CENTER BORE is the message that appears on the RCU and/or the CRT. If the prompt argument is omitted, the RCU and/or the CRT display a blank line.

If the first two characters of P1 are "~B" then the rest of the string is treated as a file name together with a placement indication. The file name must specify the drive and full path. The graphics file can be a bitmap (.bmp), Windows metafile (.wmf) or enhanced metafile (emf). Horizontal position of the graphics image can be specified by following the file name with a comma and the character "L", "C" or "R" to indicate Left, Center or Right alignment. Default alignment is at the left margin. In the second example the bitmap "C:\BITMAPS\LOGO.BMP" is to be right aligned in the Prompt Monitor.

Results

The CRT (Prompt Monitor) and/or RCU display the message.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

RD - Rectangular Coordinates

The RD command causes the system XYZ probe position display to show Cartesian part coordinates. X-axis data appears in the top display line, Y-axis data in the second line, and Z-axis data in the third display line.

Command Format

MLString "RD"

Examples

1. MLString "RD"

Input

None.

Results

Cartesian data display mode is activated. This display mode is continuous until another mode is selected. The system updates the display to reflect probe and/or table motion.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

15. DCC Motion Group

MLB commands in the DCC (Direct Computer Control) Motion Group enable the part program to control the motion of the Coordinate Measuring Machine.

The DCC Motion Group includes the following MLB commands.

AC	**DCC Centerfind** directs the CMM to automatically measure a circular feature.
AM	**DCC Probe Move** allows probe motion along the X, Y and Z axes.
HO	**Move To Home** provides the capability to perform a DCC move to the home position of the probe.
PK	**DCC Park** moves the probe to the end of travel of the X, Y and/or Z axes.
SK	**DCC Seek Move** moves the probe toward the target at the slower touch speed.
SU	**DCC Summation Touch** moves the probe at the slower touch speed and sums the point into a summation array.
SX	**Save Home Position** defines a safe position for a DCC CMM to move to allow unobstructed part loading and unloading.
TC	**DCC Touch Move** moves the probe from its current position at the slower touch speed toward the specified target.
VM	**Vector Move Using Surface Vector**
VS	**DCC Vector Seek** performs a seek from the current probe position toward a clearance point.
VT	**DCC Vector Touch** performs a DCC touch along a specified vector.
WM	**DCC Wait Move** allows probe motion along the X, Y and Z axes. It is similar to the AM command except the system completes the move before processing the next command.
WO	**DCC Probe/PH9 Move** provides the capability to perform a DCC move and select a new PH9 tip as the shaft moves. The probe tip can be swiveling to a new orientation as the arm moves.
ZM	**DCC Zero Switch Move** commands a DCC machine to move to a location defined relative to the MEA zero switches. All three axes move immediately to the zero switches.
ZZ	**DCC Zero Switch Locate** causes the DCC machine to perform a series of moves to locate the MEA zero switches. The axes move one after the other to avoid workpiece contact in the measurement area.

AC - DCC Centerfind

The AC command directs the system to automatically measure a circular feature. The system moves the probe above the feature, then down to the feature. It takes the specified number of touches around the perimeter of the circular feature. The **AC** command uses the starting angle and angular increment set by the **SA** command to place the touches and the current speed (**SD**) and touch speed (**TD**) to move the probe tip. After all measurements are made, the probe is moved to the height specified by the *AL* command above the last touch.

The command computes the best-fit to a circle that is parallel to the active working plane. Compensation is made for the probe radius, as specified by the **CM** command. An INT compensation flag indicates automatic measurement and compensation of an inner diameter. An EXT compensation flag indicates automatic measurement and compensation of an outer diameter or boss. A compensation flag set to OFF means that the system measures an ID but without probe compensation.

The **AC** command can be used with the commands that control segmented measurements and probe tip selection.

Command Format

MLString "AC P1,P2,P3,P4,P5"

Example

MLString "AC P1,P2,P3,P4,P5"

Input

- **P1** = Number of points (Numeric; required).
- **P2** = X nominal (Numeric; required).
- **P3** = Y nominal (Numeric; required).
- **P4** = Z nominal (Numeric; required).
- **P5** = Nominal diameter (Numeric; required).

P1 specifies the number of points to take on the circular feature. P2, P3, and P4 define the nominal center of the circular feature for the X-, Y-, and Z- coordinates, respectively. The coordinate for the axis perpendicular to the working plane specifies the depth (or height) at which touches are made. P5 specifies the nominal diameter of the feature.

In the example, 4 specifies the number of points to take on the circular feature. The 3.75 specifies the X- coordinate for the nominal center of the circle. The -4.5 specifies the Y- coordinate for the nominal center of the circle. The 0.0 is the P4 argument. It specifies the Z- coordinate for the nominal center of the circle. The 6.775 value specifies the nominal diameter of the feature.

Results

The contents of the Feature Register move into the Feature Stack. The Feature Register stores the data for the circular feature. The Dimensional Array stores:

- Center coordinates,
- Feature radius,
- Feature diameter,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skip block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-041 The points do not define the feature.
- MLB-080 The system does not have the required probe.
- MLB-090 DCC is not available.
- MLB-091 The probe tip does not make contact with the required object.
- MLB-092 There is insufficient clearance.

AM - DCC Probe Move

The AM command directs probe motion along the X, Y, and Z axes. The probe moves from its present position to a target (an absolute move) or it moves a specified distance (an incremental move).

Command Format

MLString "AM P1"

Example

1. MLString "AM X-5,Y10"
2. MLString "AM ZI1.5,Y6"

Input

- **P1** = X target and/or Y target and/or Z target (String; required).

P1 specifies the move targets as either absolute targets in the current reference frame or incremental targets from the current probe position.

For an absolute move, P1 specifies the axis designators (X, Y, Z) and targets in current part coordinates. For an incremental move, P1 specifies the axis designator, the letter I, and the distance of the probe move. The incremental move is one from the probe's present position, parallel to the designated axes in the current part coordinate reference frame. The letter I must be included with each axis making an incremental move.

If a value is omitted for an axis, the system uses the current position (current target if the axis is moving) of the probe for that axis. Axis targets may be specified in any order.

In the first example, X-5,Y10 is the P1 argument. X is the axis designator for the first coordinate of the target. -5 is the X-coordinate for the target. Together they indicate an absolute move along the X-axis until the probe reaches -5 in current part coordinates. Y is the axis designator for the second coordinate of the target. 10 is the Y- coordinate for the target. Together they indicate an absolute move along the Y-axis until the probe reaches 10 in current part coordinates. Since there is no Z-coordinate, the system uses the current position of the probe for that axis.

In the second example, ZI1.5,Y6 is the P1 argument. Z is the axis designator for the first coordinate of the target. I indicates an incremental move. 1.5 is the distance of the move. Together they indicate an incremental move along the Z- axis of 1.5 from the current position of the probe. The move occurs parallel to the designated axis. Y is the axis designator for the second coordinate of the target. 6 is the Y- coordinate for the target. Together they indicate an absolute move along the Y-axis until the probe reaches 6 in current part coordinates. Since there is no X-coordinate, the system uses the current position of the probe for that axis.

Results

If the system is in AUTO mode and is not currently moving, then the requested motion begins immediately. If another move is in progress currently then this move is simply added to the move queue and will be executed when the preceding move is complete. If the system is not in AUTO mode, the system prompts for the operator to select AUTO mode.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-090 DCC is not available.

HO - Move to Home

The HO command positions a DCC machine at its home position as defined by SHEFFNET.

Warning

The HO command was designed for SHEFFNET systems and it is not intended to be used with MeasureMax+ systems.

Command Format

MLString "HO"

Examples

1. MLString "HO"

Input

None.

Results

The machine moves to a predefined home position. If the system does not have a CMM power-up task to define the home position, the home position must be defined manually with the **SX** command prior to using the **HO** command.

MeasureMax+ processes the HO command differently than SHEFFNET. MeasureMax+ moves the machine to the limit switches.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

None.

PK - DCC Park

The PK command moves the probe to the specified end of travel for each of the specified axes. This can be used to clear the work area for part changes or other manual operations. It is recommended that machines equipped with "MEA" use the faster command **ZM** for the same affect.

Command Format

MLString "PK P1"

Examples

1. MLString "PK X,-Y,Z"

Input

- P1 = [-]X, [-]Y, and/or [-]Z (String; required).

P1 specifies the axes to park by giving a list of signed axis designators separated by commas. A negative sign indicates to move to the negative end of that axis, otherwise the move is to the positive end of travel. Axes may be specified axes in any order. When specifying more than one axis, separate the axis designators by commas.

In the example, X,-Y, Z is the P1 argument. It specifies which axes to park. The probe moves to the end of travel in the positive direction for the X and Z axes. It moves to the negative end of travel in the Y-axis.

Results

The probe arm on the machine moves to the limit of travel on the specified axes.

If the CMM is equipped with MEA zero switches then the park operation for any of the axes is terminated when the corresponding zero switch is tripped.

If an axis is parked to an end of travel on a machine without zero switches, the park operation is not completed until the physical endstop is reached.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-090 DCC is not available.

SK - DCC Seek Move

The SK command directs probe motion along the X, Y, and Z part axes at touch speed (see the **TD** command) in expectation of encountering a surface. The move continues until either a surface is encountered or the probe has moved the overdrive distance (see the **OD** command) past the target.

Command Format

MLString "SK P1"

Example

- MLString "SK X-5,Y10"
- MLString "SK ZI1.5,Y6"

Input

- **P1** = X target and/or Y target and/or Z target (String; required).

P1 specifies the move targets as either absolute targets in the current reference frame or incremental targets from the current probe position.

For an absolute move, P1 specifies the axis designators (X, Y, Z) and targets in current part coordinates. For an incremental move, P1 specifies the axis designator, the letter I, and the distance of the probe move. The incremental move is one from the probe's present position, parallel to the designated axis in the current part coordinate reference frame. The letter I must be included with each axis making an incremental move.

If an axis target is omitted, the system uses the current position (current target if the axis is moving) of the probe for that axis. Axis targets may be specified in any order.

In the first example, X-5,Y10 is the P1 argument. X is the axis designator for the first coordinate of the target. -5 is the X-coordinate for the target. Together they indicate an absolute target along the X-axis at -5 in current part coordinates. Y is the axis designator for the second coordinate of the target. 10 is the Y- coordinate for the target. Together they indicate a target along the Y-axis at 10 in current part coordinates. Since there is no Z-coordinate, the system uses the current position of the probe as the target for that axis.

In the second example, ZI1.5,Y6 is the P1 argument. Z is the axis designator for the first coordinate of the target. I indicates an incremental move. 1.5 is the distance of the move. Together they indicate a target at an incremental distance along the Z- axis of 1.5 from the current position of the probe. Y is the axis designator for the second coordinate of the target. 6 is the Y- coordinate for the target. Together they indicate a target at an absolute position along the Y-axis of 6 in current part coordinates. Since there is no X-coordinate, the system uses the current position of the probe as the target for that axis.

Results

The probe is moved at touch speed from its previous position towards and through the specified target until either a surface is contacted or the probe moves the overdrive distance past the target. In either case a point feature is recorded. If the probe does not make a touch, the variable S7 equals 0 and the final commanded probe position is recorded. If the probe does make the touch, the variable S7 equals 1 and the touched point coordinates are recorded.

The contents of the Feature Register move into the Feature Stack. The Feature Register stores the point. The Dimensional Array stores:

- Point coordinates,

- Feature type, and
- Feature name.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-080 The system does not have the required probe.
- MLB-090 DCC is not available.

SU - DCC Summation Touch

The SU command moves the probe at touch speed (**TD**) and records a touch point. This point is summed into one of the summation arrays. If the probe tip reaches the overdrive distance past the target without contacting the part, the system generates an error.

Command Format

MLString "SU P1,P2"

Example

1. MLString "SU INTERNAL,X-29.755,Y36.250,Z-19.009"

Input

- **P1** = INTERNAL (String; optional).
- **P2** = X target and/or Y target and/or Z target (String; required).

P1 can only be INTERNAL. This specifies that the point is to be summed into the internal summation array. Omitting P1 causes the point to be added to the user summation array.

P2 specifies the touch targets as either absolute targets in the current reference frame or incremental targets from the current probe position.

For an absolute target, P2 specifies the axis designators (X, Y, Z) and targets in current part coordinates. For an incremental move, P2 specifies the axis designator, the letter I, and the distance of the probe move. The incremental move is one from the probe's present position, parallel to the designated axes in the current part coordinate reference frame. The letter I must be included with each axis making an incremental move.

If an axis target is omitted, the system uses the current position (current target if the axis is moving) of the probe for that axis. Axis targets may be specified in any order.

In the example, X-29.755,Y36.250,Z-19.009 is the P1 argument. The touch target is at the X position -29.755, the Y position 36.250, and the Z position -19.009.

Results

The probe is moved at touch speed from its previous position towards and through the specified target until either a surface is contacted or the probe moves the overdrive distance past the target. If the surface is contacted then the touch point coordinates are captured and summed into the specified summation array.

If the probe does not make a touch, then an error is generated.

Skip Action

Not executed if part of a skipped block.

Error Messages

- MLB-042 occurs if a change in reference frames or use of a measurement command has invalidated the summation array.
- MLB-043 occurs if a segmented measurement is in progress.
- MLB-080 The system does not have the required probe.
- MLB-090 DCC is not available.
- MLB-091 The probe tip did not make contact.

SX - Save Home Position

The SX command defines a safe position for a DCC CMM. This allows unobstructed part loading and unloading. The "safe position" is known as the home position and is used as the move target for the **HO** command. This command should only be used on systems that do not have a CMM power-up task.

> **Warning**
>
> The HO command was designed for SHEFFNET systems and it is not intended to be used with MeasureMax+ systems.
>
> When using the **SX / HO** method of defining and repositioning to the home position, reference frame 10 should not be used in any part program.

Command Format

MLString "SX"

Example

1. MLString "SX"

Input

None.

Results

The current probe position is read and established as an XYZ reference, and the reference frame is stored as reference frame 10. Note that when using the **SX / HO** method of defining and repositioning to the home position, reference frame 10 should not be used in any part program.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

TC - DCC Touch Move

The TC moves the probe from its current position toward the specified target at touch speed. A point feature is recorded when the probe touches a part. If the probe has not made contact with the part within the specified overdrive distance, probe movement stops and an error is reported. See the **TD** and **OD** commands for information on changing touch speeds and overdrive distances.

Command Format

MLString "TC P1"

Examples

1. MLString "TC XI.4495,YI.7676"

Input

- **P1** = X target and/or Y target and/or Z target (String; required).

P1 specifies the move targets as either absolute targets in the current reference frame or incremental targets from the current probe position.

For an absolute move, P1 specifies the axis designators (X, Y, Z) and targets in current part coordinates. For an incremental move, P1 specifies the axis designator, the letter I, and the distance of the probe move. The incremental move is one from the probe's present position, parallel to the designated axes in the current part coordinate reference frame. The letter I must be included with each axis making an incremental move.

If an axis target is omitted, the system uses the current position (current target if the axis is moving) of the probe for that axis. Axis targets may be specified in any order.

In the example, XI.4495,YI.7676 is the P1 argument. This indicates that the touch target is .4495 units from the current position in the X direction and 0.7676 units from the current position in the Y direction. Since there is no Z-coordinate, the system uses the current position of the probe as the target for that axis.

Results

The probe is moved at touch speed from its previous position towards and through the specified target until either a surface is contacted or the probe moves the overdrive distance past the target. In the first case a point feature is recorded. In the second case an error is reported.

If point is recorded then the contents of the Feature Register move into the Feature Stack. The Feature Register stores the point. The Dimensional Array stores:

- Point coordinates,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed into the Feature Register.

Error Messages

- MLB-080 The system does not have the required probe.
- MLB-090 DCC is not available.
- MLB-091 The probe tip did not make contact.

VM - Vector Move Using Surface Vector

Move from the current probe position to a point which is the clearance distance (see the **CL** command) plus the probe radius away from the specified surface point. The surface point is described as a target point and a surface normal vector.

Command Format

MLString "VM P1,P2"

Examples

1. MLString "VM 62.810,0.114,-50,0,0,1.1"

Input

- **P1** = XYZ point (Numeric/parametric; required).
- **P2** = XYZ vector (Numeric/parametric; required).

P1 specifies the X, Y, and Z coordinates of the target surface point in the current reference frame.

P2 defines the vector in the current reference frame perpendicular to the surface at the target point and pointing from the surface point out into space. P2 may be specified as a unit vector but this isn't required.

In the example, (62.810,0.114,-50) are the XYZ components of the surface target point. The (0,0,1.1) numbers are the XYZ components of the vector perpendicular to the surface at the target point, directed away from the surface. The probe is moved from the current probe position to a point that is the clearance distance plus the probe radius away from this point.

Results

A move command (at normal move speed) is generated from the current probe position toward the clearance point. This command causes the specified move to be placed into the DCC move queue but control is returned to the part program without waiting for the move to complete.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-025 occurs if a zero surface normal vector is specified.

- MLB-090 occurs if DCC operations are not allowed.

VS - DCC Vector Seek

The VS command performs a "seek" from the current probe position toward a point which is the clearance distance (see the **CL** command) plus probe radius away from the specified surface point. If no touch occurs during the seek to the clearance point, a second seek is performed toward the specified surface point.

The surface point is described as a target point and a surface normal vector.

Command Format

MLString "VS P1,P2,P3"

Examples

1. MLString "VS 62.810,0.114,-50,0,0,1.1"
2. MLString "VS -4.15,2.00,.01,.15,-4.25,12.000,NO"
3. MLString "VS -4.15,2.00,.01,.15,-4.25,12.000,CU"

Input

- **P1** = XYZ point (Numeric/parametric; required).
- **P2** = XYZ vector (Numeric/parametric; required).
- **P3** = String AB, where A is "N" or "C" and B is "O","U", "V" or "N" (Optional).

P1 specifies a point in the current reference frame. It represents the X, Y, and Z coordinates of the target point.

P2 specifies a vector in the current part reference frame. The vector must be perpendicular to the surface at the target point and point from the surface point out into space. Note that P2 may be specified as a unit vector but that this is not required.

P3 is an optional two-character string, default "CO". Both characters must be specified if P3 is used. The first character is either "C" or "N", indicating whether the touched point is "C"orrected or "N"ot corrected for probe radius or for displacement from the nominal vector. The second character designates what is done with the resulting touched point data and is either "O" for "O"rdinary push of the point into the feature register, "U" for putting the point into the "U"ser array, "V" for putting the point into the internal summation array, or "N" for returning the data to the MLB variables.

In the first example, (62.810,0.114,-50) is the X,Y,Z components of the surface target point. The (0,0,1.1) is the X,Y,Z vector perpendicular to the surface at (62.810,0.114,-50) pointing away from the surface. The touched point is to be corrected for probe radius and for displacement from the nominal vector.

In the second example, (-4.15,2.00,.01) is the X,Y,Z components of the surface target point. The (.15,-4.25,12.000) is the X,Y,Z vector perpendicular to the surface at (4.15,2.00,.01) pointing away from the surface. Since P3 is specified as "NO", the touched point is not to be corrected for probe radius or for displacement from the nominal vector; the touched point data is to be pushed into the Feature Register.

The third example is the same as the second except that the point is to be corrected for probe radius and for displacement from the nominal vector and is then to be added to the user summation array.

Results

A seek command is generated (at normal move speed) from the current probe position toward a clearance point. This point is the clearance distance plus probe radius away from the surface point P1 along the vector P2. If no touch occurs during this seek, a second seek is performed at touch speed toward the surface point P1.

If a touch occurs during the first seek, the touched coordinates are pushed into the Feature Register unless P3 was "xN". The variable S7 is set to a value of two (2), indicating that a touch occurred but not on the nominal surface.

If a touch occurs during the second seek and the argument "Nx" has not been specified, the touched point is projected onto, and corrected for probe radius using the nominal seek vector P2. The touched or corrected coordinates are then pushed into the Feature Register along with the direction cosines of the seek vector unless P3 was specified as "xN". The distance that contact occurred from the nominal surface point (+ if before, - if after) parallel to the nominal vector P2 is returned as a distance value in S(8,0).

The distance that the touch occurred from the planned vector is returned as a radius value in S(17,0). The variable S7 is set to a value of one (1), indicating that a touch occurred on the nominal surface.

If no touch occurs, the final coordinates of the probe position are pushed into the Feature Register as a point. The variable S7 is set to a value of zero (0), indicating that no touch has occurred.

If P3 is not "xN", the Dimensional Array stores:

- Coordinates (X,Y,Z) of the final probe position (S7=0) or
- Coordinates (X,Y,Z) of the touched point (S7=1 or S7=2),
- Nominal touch vector (i,j,k) (S7=1),
- Distance the touch occurred "above" or "below" the expected surface in S(8,0) (S7=1),
- Distance the touch occurred to the side of the planned vector in S(17,0) (S7=1),
- Feature type (point), and
- Feature number.

If the touched data is not pushed into the Feature Stack, then the data is returned as follows (note that the two distance values have no meaning if S7 equals 0 or 2.):

- S7 - touch (1 or 2) or miss (0) flag as above.
- S8 - X-coordinate of touch (or miss) data.
- S10 - Y-coordinate of touch (or miss) data.
- S11 - Z-coordinate of touch (or miss) data.
- S12 - distance above (+) or below (-) target.
- S13 - distance between touch and nominal vector.

If the data is added to the user array or internal array (when the second character of P3 is "U" or "V") and a touch occurs, the point data is added to the appropriate summation array. Variable S7 is set to indicate the final status of the seek. No other host data is changed.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register if P3 is not specified or if the second character of P3 is 'O'.

Error Messages

None.

VT - DCC Vector Touch

The VT command performs a drive move to clearance distance (see the **CL** command) plus probe radius away from a surface point. The surface point is described by a target point and a surface normal vector. It then performs a DCC touch along the specified vector and can compensate the resulting measured point for probe radius.

Command Format

MLString "VT P1,P2,P3"

Examples

1. MLString "VT 62.810,0.114,-2.050050,0,0,1.1"
2. MLString "VT -4.15,2.00,.01,1.15,-4.25,12.000,NO"
3. MLString "VT -4.15,2.00,.01,1.15,-4.25,12.000,CU"

Input

- **P1** = XYZ point (Numeric/parametric; required).
- **P2** = XYZ vector (Numeric/parametric; required).
- **P3** = String AB, where A is "N" or "C" and B is "O","U", "V" or "N" (Optional).

P1 specifies a point in the current reference frame. It represents the X, Y, and Z coordinates of the target point.

P2 specifies a vector in the current part reference frame. The vector must be perpendicular to the surface at the target point and point from the surface point out into space. Note that P2 may be specified as a unit vector but that this is not required.

P3 is an optional two-character string, default "CO". Both characters must be specified if P3 is used. The first character is either "C" or "N", indicating whether the touched point is "C"orrected or "N"ot corrected for probe radius or for displacement from the nominal vector. The second character designates what is done with the resulting touched point data and is either "O" for "O"rdinary push of the point into the feature register, "U" for putting the point into the "U"ser array, "V" for putting the point into the internal summation array, or "N" for returning the data to the MLB variables.

In the first example, (62.810,0.114,-2.050050) is the X,Y,Z components of the surface target point. The (0,0,1.1) is the X,Y,Z vector perpendicular to the surface at (62.810,0.114,-2.050050) pointing away from the surface. The touched point is to be corrected for probe radius and for displacement from the nominal vector.

In the second example, (-4.15,2.00,.01) is the X,Y,Z components of the surface target point. The (1.15,-4.25,12.000) is the X,Y,Z vector perpendicular to the surface at (4.15,2.00,.01) pointing away from the surface. Since P3 is specified as "NO", the touched point is not to be corrected for probe radius or for displacement from the nominal vector; the touched point data is to be pushed into the Feature Register.

The third example is the same as the second except that the point is to be corrected for probe radius and for displacement from the nominal vector and is then to be added to the user summation array.

Results

A move command is generated (at normal move speed) from the current probe position toward a clearance point. This point is the clearance distance plus probe radius away from the surface point

P1 along the vector P2. If no touch occurs during this seek, a second seek is performed at touch speed toward the surface point P1.

If a touch occurs during the first seek, an unexpected touch error is reported.

If a touch occurs during the second seek and the argument "Nx" has not been specified, the touched point is projected onto, and corrected for probe radius using the nominal seek vector P2. The touched or corrected coordinates are then pushed into the Feature Register along with the direction cosines of the seek vector unless P3 was specified as "xN". The distance that contact occurred from the nominal surface point (+ if before, - if after) parallel to the nominal vector P2 is returned as a distance value in S(8,0).

The distance that the touch occurred from the planned vector is returned as a radius value in S(17,0). The flag variable S7 is set to a value of one (1), indicating that a touch occurred on the nominal surface.

If no touch occurs, a surface not found error is reported.

If P3 is not "xN", the Dimensional Array stores:

- Coordinates (X,Y,Z) of the touched point,
- Nominal touch vector (i,j,k),
- Distance the touch occurred "above" or "below" the expected surface in S(8,0),
- Distance the touch occurred to the side of the planned vector in S(17,0),
- Feature type (point), and
- Feature number.

If the touched data is not pushed into the Feature Stack, then the data is returned as follows):

- S7 - 1
- S8 - X-coordinate of touch data.
- S10 - Y-coordinate of touch data.
- S11 - Z-coordinate of touch data.
- S12 - distance above (+) or below (-) target.
- S13 - distance between touch and nominal vector.

If the data is added to the user array or internal array (when the second character of P3 is "U" or "V") and a touch occurs, the point data is added to the appropriate summation array. No host data is changed.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register if P3 is not specified or if the second character of P3 is 'O'.

Error Messages

- MLB-090 DCC is not available.
- MLB-091 The probe tip did not make contact.

WM - DCC Wait Move

The WM command directs probe motion along the X, Y, Z, and/or W axes. The probe moves from its present position to a target (an absolute move) or it moves a specified distance (an incremental move). Unlike the **AM** command, this command does not return control to the part program until the probe reaches the target.

Command Format

MLString "WM P1"

Example

1. MLString "WM WI35,Z-9.5, X4.9"

Input

- **P1** = X target and/or Y target and/or Z target and/or W target (String; required).

P1 specifies the move targets as either absolute targets in the current reference frame or incremental targets from the current probe position.

For an absolute move, P1 specifies the axis designators (X, Y, Z, W) and targets in current part coordinates. For an incremental move, P1 specifies the axis designator, the letter I, and the distance of the probe move. The incremental move is one from the probe's present position, parallel to the designated axes in the current part coordinate reference frame. The letter I must be included with each axis making an incremental move.

If a value is omitted for an axis, the system uses the current position (current target if the axis is moving) of the probe for that axis. Axis targets may be specified in any order.

In the example, WI35,Z-9.5, X4.9 is the P1 argument. The WI35 indicates that the rotary table (W axis) is to be rotated 35 degrees in the positive direction from its current position. The Z-9.5 indicates that the probe is to be moved to a Z axis position of -9.5. The X4.9 indicates that the probe is to be moved to an X axis position of 4.9. Since no Y target is given, the probe's position along the Y axis is not to be changed.

Warning

The X, Y, and Z targets are defined in the coordinate system that exists before any specified table rotation occurs. This is important to note since the reference frame will be updated automatically when the rotary table move is complete and the displayed probe position will, in general, be different from that specified in the move command.

The linear axes move simultaneously with the table rotation. Because of the complexity involved in visualizing probe motion in parallel with part rotation on a rotary table, take extreme care when executing moves which involve simultaneous table and linear axis movement.

Results

If the system is in AUTO mode and is not currently moving, then the requested motion begins immediately. If another move is in progress currently then this move is added to the move queue and will be executed when the preceding move is complete. If the system is not in AUTO mode, the system prompts for the operator to select AUTO mode.

Completion of a commanded table rotation causes the system to automatically compute a new reference frame. When the move is complete, control returns to the part program.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-060 Rotary table is not available.
- MLB-090 DCC is not available.

WO - DCC Probe / PH9 Move

The WO command performs a DCC move and a new tip selection, including any required PH9 or PH10 rotation, in parallel.

Command Format

MLString "WO P1,P2"

Examples

1. MLString "WO 4,Z-9.5, X4.9"
2. MLString "WO TIP_9,XI4.2,YI2.2,ZI10.9"

Input

- **P1** = Probe tip identifier (Numeric or string; required).
- **P2** = X target and/or Y target and/or Z target (String; required).

P1 specifies the probe tip to be selected at the end of the specified move. If selection of this tip involves changing the PH9/PH10 axis positions then the PH9/PH10 will be commanded to reposition while the commanded move is in progress.

P2 specifies the move targets as either absolute targets in the current reference frame or incremental targets from the current probe position.

For an absolute move, P2 specifies the axis designators (X, Y, Z) and targets in current part coordinates. For an incremental move, P2 specifies the axis designator, the letter I, and the distance of the probe move. The incremental move is one from the probe's present position, parallel to the designated axes in the current part coordinate reference frame. The letter I must be included with each axis making an incremental move.

If a value is omitted for an axis, the system uses the current position (current target if the axis is moving) of the probe for that axis. Axis targets may be specified in any order.

In the first example, Z-9.5, X4.9 is the P1 argument. The Z-9.5 indicates that the probe is to be moved to a Z axis position of -9.5. The X4.9 indicates that the probe is to be moved to an X axis position of 4.9. Since no Y target is given, the probe's position along the Y axis is not to be changed. At the end of this move tip 4 is to be selected as the active tip.

Warning

The X, Y, and Z targets are defined relative to the tip that is active before the move begins. This is important to note since the reference frame will be updated automatically when the new tip is selected at the end of the move and the displayed probe position will (in general) be different from that specified in the move command.

Any required PH9/PH10 motion will occur simultaneously with the commanded XYZ motion. The speed and motion of this reorientation is controlled by the Renishaw hardware and cannot be synchronized with the commanded CMM axis motions. To allow for different CMM speeds (such as the result of the use of the RCU FAST/SLOW keys), there should be no obstructions anywhere along the commanded probe path that would interfere with the required PH9/PH10 motion. Take extreme care when executing moves that involve simultaneous PH9/PH10 and linear axis movement.

Results

If the system is in AUTO mode and is not currently moving, then the requested motion begins immediately. If another move is in progress then this move is added to the move queue and will be executed when the preceding move is complete. If the system is not in AUTO mode, the system prompts for the operator to select AUTO mode.

The system repositions the PH9/PH10 as the probe arm moves along the machine axes. The designated tip becomes active at the end of the move. Make sure there is a sufficient clearance to allow the probe to rotate in any direction along the entire length of probe travel. Otherwise, injury to personnel, to the probe or to the part can occur.

Completion of the move causes the system to automatically select the new tip and compute a new reference frame. When the move is complete, control returns to the part program.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-026 The specified tip is invalid.
- MLB-090 DCC is not available.
- MLB-100 The PH9 is not seated properly.
- MLB-101 The probe head is obstructed.

ZM - DCC Zero Switch Move

The ZM command causes a DCC machine to move to a location defined relative to the MEA zero switches.

Command Format

MLString "ZM P1,P2,P3"

Examples

1. MLString "ZM"
2. MLString "ZM +X50,-Y"

Input

- **P1** = "X" and location (Parametric; optional).
- **P2** = "Y" and location (Parametric; optional).
- **P3** = "Z" and location (Parametric; optional).

P1, P2, and P3 are used to specify an end of the X, Y, or Z axis, respectively. They may also specify a distance inside of the end of travel for the particular axis. The distance specified must be positive and in millimeters. The distance specified in P1, P2, and P3, along with the following information, tells the system where to position the probe shaft:

- Location where the MEA zero switches were tripped,
- Configured location of the switches on the CMM, and
- Configured lengths of the X, Y, and Z axes.

Any or all of P1, P2, and P3 may be specified. If only one or two axes are specified then the position of the remaining axis is left unchanged during the move. If none of P1, P2, and P3 are specified then all three axes are simultaneously moved to the location where the MEA zero switches were tripped.

If the zero switches were set using the "ZZ" command and P1, P2, and P3 are not specified, the probe shaft is positioned 2 mm closer to the end-of-travel than the actual trip point.

In the first example, the probe shaft is moved to the location defined by the MEA zero switches. In the second example, the probe shaft is moved to a point 50 mm from the X-axis MEA zero switch (in the + direction). The probe shaft is moved to the minus Y-axis end of travel. The plus (+) sign included in P2 is optional.

Results

The system prompts the user to set the machine zeroes if the user has not set them. If the CMM is not in "AUTO" mode, the system prompts the user that "AUTO" mode is required. Then, a normal speed DCC probe shaft move is made as defined by the MEA zero switch locations and the input parameters.

The system ignores the **ZM** command if the CMM does not have MEA zero switches.

Skip Action

Ignored; not executed if inside a skipped block.

Error Messages

None.

ZZ - DCC Zero Switch Locate

The ZZ command causes a DCC machine to perform a series of moves to exactly locate the MEA zero switches.

Command Format

MLString "ZZ"

Examples

1. MLString "ZZ"

Input

None.

Results

If the CMM is not in "AUTO" mode, the system prompts the user to set the CMM to "AUTO" mode. Then, the system:

1. Parks the Z axis to the end of travel defined to be the location of its MEA limit switch, normally +Z.
2. Parks the Y axis to the end of travel defined to be the location of is MEA zero switch, normally -Y.
3. Parks the X axis to the end of travel defined to be the location of is MEA zero switch, normally -X.
4. Resets the maximum DCC speed to 25.4 mm per second.
5. Simultaneously moves X, Y, and Z 25 mm from the zero switches towards the center of the measurement range.
6. Again, simultaneously parks the X, Y, and Z axes to the ends of travel defined to be the location of the MEA zero switches.
7. Simultaneously moves X, Y, and Z 10 mm away from the zero switches towards the center of the measurement range.
8. Resets the maximum DCC speed to 1.0 mm per second.
9. Again, simultaneously parks the X , Y, and Z axes to the ends-of-travel defined to be the location of the MEA zero switches.
10. Restores the original maximum DCC velocity.

If the system also contains an MEA'd W-axis then:

1. Performs moves from the following sequence until the W-axis zero switch is tripped:
   Move W incrementally minus ten degrees.
   Move W incrementally plus twenty degrees.
   Move W incrementally plus 340 degrees.
2. Move W to a position one degree on the plus side of the zero switch.
3. Move W incrementally minus two degrees.

4. Move W to a position of zero degrees.

The previous sequence of moves results in a very repeatable MEA zero switch location. When a **C6** command is performed after execution of the **ZZ** command, the location of the auto probe changer (APC) sphere relative to the MEA zero switch location is stored in non-volatile memory. When the **ZZ** command is executed again after cycling power, the information stored in memory is used to define the location of the rack. It is not necessary to perform the **C6** manual measurements again.

Skip Action

Ignored; not executed if inside a skipped block.

Error Messages

None.

16. DCC Control Group

MLB commands in the DCC Control Group let you change speeds and other direct computer controlled (DCC) variables controlling automatic feature measurement. Automatic feature measurement is controlled by the DCC Motion group. Refer to the DCC Motion group for relative MLB commands.

The DCC Control Group includes the following MLB commands.

AL	**Altitude** specifies the probe tip height from a nominal center just before a DCC Centerfind measurement.
AU	**Auto Status** determines the Auto or Manual status of the system as shown by the RCU.
AW	**Auto Mode Wait** executes a programmed wait until the system is placed in the Auto mode.
B0	**Disable Beep after DCC Touch** turns off the normal MP-30 beeper response to a DCC touch probe deflection.
B1	**Enable Beep after Touch** turns on the normal MP-30 beeper response to a DCC touch probe deflection.
BO	**Backoff Distance** specifies the distance that the probe moves away after it contacts a surface.
CL	**Clearance** specifies a distance from the nominal surface or circumference of a circle that the probe tip keeps to avoid contact before touches.
DW	**DCC Wait** prevents execution of any further FLB commands in a part program until all previous commands are completed.
MA	**Auto Mode** turns the Auto mode on. This is only supported on machines with special hardware.
MN	**Manual Mode** switches the CMM into Manual mode.
OD	**Overdrive Distance** sets the distance the probe can move without making contact with the target after reaching the nominal position of the target.
OS	**Servo Off** turns the servo power off.
SA	**Starting Angle** defines the starting angle and angular increment for a DCC measurement of a bore or boss. Used with the **AC** command.
SD	**Probe Speed** specifies the maximum probe velocity.
SV	**Servo Status** determines whether the servo power is on or off.
SW	**Table Speed** specifies the maximum speed for table rotation.
TA	**Touch Status** determines whether the touch mode is on or off, indicated by the PH9/PH10 red light.

TD	**Touch Speed** specifies the maximum speed for all programmed touches.
TF	**Touch Mode Off** prevents the system from sensing a deflected probe, whether the deflection is caused by a touch or vibration.
TN	**Touch Mode On** allows the system to sense a deflected probe.
TT	**Target Tolerances** sets the distance for the probe to begin its next DCC move before it completes its current move. Works in conjunction with high-speed commands such as **AM**.
TU	**Set Move Target Tolerance** specifies the X, Y and Z target tolerances used to begin the next high-speed DCC move, and to determine when that move is complete.
TV	**Set Touch Target Tolerance** specifies X, Y and Z target tolerances used to begin a DCC touch command, and to determine when that command is complete if no contact occurs. Works in conjunction with **SK** and **TC** commands.
TW	**Set Minimum Time** sets a minimum time interval that is to elapse between the end of a high-speed move and a subsequent surface touch.

AL - Altitude

The AL command controls the probe position before and after **AC** measurements. The altitude is actually the clearance height for the third axis relative to the height (or depth) of measurement. It is a distance relative to the nominal center specified in the **AC** command. The probe is moved to this altitude above the first point to be measured and then returns to this altitude above the last point measured. The default value is zero.

Command Format

MLString "AL P1"

Example

MLString "AL 3.250"

Input

- **P1** = Clearance altitude (Numeric; optional).

P1 specifies the clearance altitude in current measurement units. Changing units (such as millimeters to inches or inches to millimeters) automatically converts the programmed altitude to the new measurement units.

Altitude is a signed distance, giving the direction from the nominal position relative to the axis perpendicular to the working plane. In the example, the altitude is 3.250 units.

If P1 is not specified, the current setting for altitude is returned to S7.

Results

If a value is specified then it is stored for use with future **AC** commands. If a value is not specified then the current altitude value is returned to variable S7.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-090 DCC is not available.

AU - Auto Status

This command determines whether the system is in AUTO mode.

Command Format

MLString "AU"

Example

MLString "AU"

Input

None.

Results

Variable S7 is set equal to 0 if the machine is in MANUAL mode. If the machine is in AUTO mode, variable S7 equals 1.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-090 DCC is not available.

AW - Auto Wait Mode

The AW command executes a programmed wait until the operator places the system in AUTO mode.

Command Format

MLString "AW"

Example

MLString "AW"

Input

None.

Results

The command returns control to the part program only after the operator places the machine into AUTO mode.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-090 DCC is not available.

B0 - Disable Beep after DCC Touch

The B0 command disables the normal MP-30 beeper response to a DCC touch probe deflection.

Command Format

MLString "B0"

Example

MLString "B0"

Input

None.

Results

Following execution of this command, the MP does not sound its beeper after a DCC (AUTO mode) touch probe deflection. This mode remains active until the system is re-initialized or a **B1** command is executed.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

B1 - Enable Beep after DCC Touch

The B1 command enables the normal MP-30 beeper response to a DCC touch probe deflection.

Command Format

MLString "B1"

Example

MLString "B1"

Input

None.

Results

Following execution of this command, the MP sounds its beeper after all touch probe deflections. Note that this is the normal operational state of the MP. This command is used to reverse the affect of the **B0** command.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

BO - Backoff Distance

The BO command specifies the backoff distance for the probe tip after its contact with a surface in response to a touch in DCC AUTO mode. The backoff distance is the distance that the probe retracts, along the original path of travel, from the touched point. The default value is 1 mm (0.039 in).

Maximum backoff is 10 mm (0.39 in). The system ignores any attempt to set backoff distance over 10 mm (0.39 in) and uses 10 mm (0.39 in).

Command Format

MLString "BO P1"

Examples

1. MLString "BO 5"

Input

- **P1** = Backoff distance (Numeric; optional).

P1 specifies the distance, in current measurement units, that the machine backs away from a part after the probe tip contacts the part. The backoff distance is a positive distance. If P1 is less than zero, the system uses the absolute value of P1. In the example, the backoff distance is 5.

If P1 is not specified, the current setting for backoff distance is returned to S7.

Results

If P1 is specified, the value of the backoff distance is changed. The new value is active until the system is reinitialized (see **IL** command) or another **BO** command is executed. If P1 is not specified then the current backoff distance setting is returned to S7.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-090 DCC is not available.

CL - Clearance

The CL command sets the clearance distance which is used for planning moves for the **VM** (surface vector move), the VS (vector seek), the **VT** (vector touch) and the **AC** (DCC centerfind) commands. Machine movements are planned to avoid contact by staying at least the clearance distance away from the nominal surface except during touches. The actual surface can be no more than the clearance distance from the nominal surface. This ensures that the probe does not contact the part surface before a scheduled touch. The default clearance value is 2 mm (0.078 in).

Command Format

MLString "CL P1"

Examples

1. MLString "CL 2.5"

Input

- **P1** = Clearance distance (Numeric; optional).

P1 specifies the clearance distance in current measurement units. When measurement units change (i.e., millimeters to inches or inches to millimeters), the system automatically converts the programmed clearance. The clearance distance is a positive distance. If P1 is negative, the command uses the absolute value of P1. In the example, the clearance distance is 2.5 units.

If P1 is not specified, the current setting for clearance distance is returned to S7.

Results

If P1 is specified, the value of the clearance distance is changed. The new value is active until the system is reinitialized (see IL command) or another **CL** command is executed. If P1 is not specified then the current clearance distance setting is returned to S7.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-090 DCC is not available.

DW - DCC Wait

The DW command executes a programmed wait until the CMM completes all of the commanded motion operations.

Command Format

MLString "DW"

Examples

1. MLString "DW"

Input

None.

Results

The command returns control to the part program only after all programmed DCC moves are completed.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-090 DCC is not available.

MA - Auto Mode

The MA command allows the auto mode of a DCC system to be entered.

> **Warning**
>
> The MA command is only operational on machines configured to have this capability. These machines are equipped with special servo-controlled chassis and equipped with a special hardware interlock used to prevent operator injury.
>
> It is normally used only in situations where machine operation is completely unmanned.

Command Format

MLString "MA"

Examples

1. MLString "MA"

Input

None.

Results

If servo power is off, it is turned on. The system enters into AUTO mode.

Skip Action

Not executed if part of a skip block.

Error Messages

- MLB-090 DCC is not available.
- MLB-093 Remote Auto is not configured.
- MLB-094 Remote Auto interlock is not engaged.
- MLB-095 Servo power cannot be engaged.
- MLB-096 The system cannot enter Auto mode.

MN - Manual Mode

The MN command switches the system into the MANUAL mode after the CMM completes all of the motion operations in the DCC queue.

Command Format

MLString "MN"

Examples

1. MLString "MN"

Input

None.

Results

When the CMM completes all programmed DCC moves, the system switches to MANUAL mode. The operator must interact with the system to continue with the part program.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-090 DCC is not available.

OD - Overdrive Distance

The OD command sets a new overdrive distance. The overdrive distance is the amount that the probe drives past an expected touch target before failure to contact the part becomes an error. The default value is 8 mm (0.315 in.) along the path of travel.

Command Format

MLString "OD P1"

Examples

1. MLString "OD 3.50"

Input

- **P1** = Overdrive distance (Numeric; optional).

P1 specifies the overdrive distance in current measurement units. The overdrive distance is a positive distance. If P1 is less than zero, the absolute value of P1 is used. In the example, the overdrive distance is 3.50.

If P1 is not specified, the current setting for overdrive distance is returned to S7.

Results

The command changes the value of the overdrive distance. If P1 is not specified, the current setting for overdrive distance is returned to S7.

Skip Action

Executed normally even if within a block of code being skipped.

Error Messages

- MLB-090 DCC is not available.

OS - Servo Off

The OS command turns the servo power off after the system finishes executing the commands in the DCC queue.

Command Format

MLString "OS"

Examples

1. MLString "OS"

Input

None.

Results

The system shuts off the servo power once the DCC move queue is empty.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-090 DCC is not available.

SA - Starting Angle

The SA command defines the starting angle and angular increment for the next **AC** measurement. After the next **AC** command is complete, the system goes back to its default values. The default value for angular increment is 360 degrees divided by the number of touches specified. The default starting angular value is 0 degrees.

Command Format

MLString "SA P1,P2"

Examples

1. MLString "SA 15,30"

Input

- **P1** = Starting angle (Numeric; required).
- **P2** = Angular increment (Numeric; optional).

P1 specifies the polar angle for the first touch of the next **AC** command. The example specifies that the first touch is to occur at an angle of 15 degrees.

P2 specifies the angular increment between successive touches for the next **AC** command. The example specifies an angular increment of 30 degrees. If P2 is omitted, the touches will be evenly spaced over a full 360 degrees.

Note that P2 is a polar angle increment and may be either positive or negative. A positive value causes touches to occur successively moving in a counter-clockwise direction while a negative value causes touches to occur successively moving in a clockwise direction.

Results

The specified starting angle and/or angle of increment are saved for used by the next **AC** command.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-090 DCC is not available.

SD - Probe Speed

The SD command specifies the maximum linear probe velocity. The maximum machine speed limit for each axis restrains the maximum speed at which the machine can move the probe along any line. Any attempt to set the moving speed above the maximum limit will simply mean that each move is executed as quickly as possible based upon the axes and distances involved.

Command Format

MLString "SD P1"

Examples

1. MLString "SD 419"

Input

- **P1** = Maximum velocity (Numeric; optional).

P1 specifies the maximum linear probe velocity in current measurement units (inches or millimeters) per second. If P1 is less than zero, then the absolute value of P1 is used. In the example, maximum probe velocity is set to 419 units per second.

If P1 is not specified, the current setting for probe speed is returned to S7.

Results

If P1 is specified then the maximum probe velocity is reset to this value. If P1 is not specified then the current value is returned to variable S7.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-090 DCC is not available.

SV - Servo Status

The SV command determines whether the servo power is on or off.

Command Format

MLString "SV"

Example

1. MLString "SV"

Input

None.

Results

The variable S7 is set equal to 0 if the servo power is off. If the servo power is on, variable S7 is set equal to 1.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-090 DCC is not available.

SW - Table Speed

The SW command specifies the maximum speed for rotary table axis motions made in AUTO mode. The system accepts speed expressed in degrees per second. The default value of the maximum machine speed is 12 degrees per second; for Horizon machines, the maximum machine speed is 18 degrees per second.

Command Format

MLString "SW P1"

Example

1. MLString "SW 10"

Input

- **P1** = Maximum speed (Numeric; optional).

P1 specifies the maximum speed for table rotation in degrees per second. If P1 is less than zero, the absolute value of P1 is used. In the example, the maximum rotation speed of the table is set to 10 degrees per second.

If P1 is not specified, the current setting for table speed is returned to S7.

Results

If P1 is specified then the maximum rotary table speed is reset to that value. Note that the system will never command the rotary table to move faster than its configured hardware limits so values larger than that are effectively ignored.

If P1 is not specified then the current setting for the rotary table speed is returned in variable S7.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-090 DCC is not available.

TA - Touch Status

The TA command determines whether the touch probe active mode is on or off.

Command Format

MLString "TA"

Example

1. MLString "TA"

Input

None.

Results

Variable S7 is set equal to 0 if the touch probe active mode is off. If the touch probe active mode is on, variable S7 is set equal to 1.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-090 DCC is not available.

TD - Touch Speed

The TD command sets the speed to be used for all programmed touch operations. This includes **SK**, **TC**, **SU**, **VS**, **VT** and **AC** commands. The default touch speed at part program initialization is 0.3 in./sec (7.62 mm./sec.)

Command Format

MLString "TD P1"

Examples

1. MLString "TD 6.54"

Input

- **P1** = Touch speed (Numeric; optional).

P1 specifies the touch speed in current measurement units. If P1 is less than zero, the command uses the absolute value of P1. In the example, touch speed is set to 6.54 units per second.

If P1 is not specified, the current setting for touch speed is returned to S7.

> **Warning**
>
> The scale factors applied with the **SM** command do not apply to or affect the speed. Touches at excessive touch speeds can damage the probe.

Results

If P1 is specified then the touch speed is reset to this value. If P1 is not specified then the current value is returned to variable S7.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-090 DCC is not available.

TF - Touch Mode Off

The TF command prevents the system from sensing deflection of the touch probe except during commanded touches or seeks. In this mode, the system can ignore touches resulting from the vibration experienced by a long probe that is traveling at a high velocity. Because damage can result if the probe contacts a surface with force, use extreme caution when using this command.

Whenever this command is used in a part program the **TN** command should be used to re-enable the touch probe before terminating the part program.

Command Format

MLString "TF"

Example

1. MLString "TF"

Input

None.

Results

After this command is executed the system is does not recognize touch probe hits except during programmed DCC touch operations. Recognition of touch probe hits can be re-enabled by the **TN** command or by a part program initialization.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-090 DCC is not available.

TN - Enable Touch Probe

The TN command sets the touch mode to on, reenabling the touch probe previously disabled with the **TF** command. If a three-axis analog probe (e.g., the Renishaw SP-600) is in use, then the soft probe is disabled (see the **BF** command).

Command Format

MLString "TN"

Example

1. MLString "TN"

Input

None.

Results

The touch probe is enabled during all moves.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-090 DCC is not available.

TT - Target Tolerances

The TT command sets both the move (see the **TU** command) and the touch (see the **TV** command) target tolerances for the X, Y and Z-axes. These tolerances are used to determine whether it is OK to begin a programmed DCC move or touch and to determine when each programmed move or touch command is complete. A programmed machine move is complete when all involved axes are within specified tolerances, in machine coordinates, of their respective targets. The default value for target tolerances for the X, Y and Z-axes set at part program initialization is 1 mm (0.039 in.) for moves and touches.

Command Format

MLString "TT P1,P2,P3"

Examples

1. MLString "TT .50,.50,.25"

Input

- **P1** = X tolerance (Numeric; optional).
- **P2** = Y tolerance (Numeric; optional).
- **P3** = Z tolerance (Numeric; optional).

P1, P2, and P3 specify the machine coordinate target tolerances for the X, Y, and Z axes respectively. In the example, the tolerance for the X-axis is .50, the tolerance for the Y-axis is .50, and the tolerance for the Z axis is .25.

If any of P1, P2, and P3 are specified then all three are required. The target tolerance is a positive distance. If any of the specified values are less than zero, then the absolute value is used instead.

If P1, P2 and P3 are not specified, the current settings for move target tolerances are returned to MLB variables.

Results

If values are specified then both the move and the touch target tolerances are set to the specified values. Note that this overrides the effects of both the **TU** and **TV** commands.

If values are not specified then the current settings for the X, Y and Z move target tolerances are returned to variables S7, S8 and S10 respectively.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-090 DCC is not available.

TU - Set Move Target Tolerance

The TU command sets the move target tolerances for the X, Y and Z axes. These tolerances are used to determine whether it is OK to begin a programmed DCC move and to determine when each programmed move is complete. A programmed machine move is complete when all involved axes are within the specified tolerances, in machine coordinates, of their respective targets. The default value for target tolerances for the X, Y and Z-axes set at part program initialization is 1 mm (0.039 in.) for moves.

Command Format

MLString "TU P1,P2,P3"

Examples

1. MLString "TU 25,25,1"
2. MLString "TU"

Input

- **P1** = X tolerance (Numeric; optional).
- **P2** = Y tolerance (Numeric; optional).
- **P3** = Z tolerance (Numeric; optional).

P1, P2, and P3 specify the machine coordinate move target tolerances for the X, Y, and Z axes respectively in current measurement units. In the first example, the tolerance for the X-axis is 25, the tolerance for the Y-axis is 25, and the tolerance for the Z axis is 1.

If any of P1, P2, and P3 are specified then all three are required. The target tolerance is a positive distance. If any of the specified values are less than zero, then the absolute value is used instead.

If P1, P2 and P3 are not specified, the current settings for move target tolerances are returned to MLB variables.

Results

If values are specified then the move target tolerances are set to the specified values. Move commands thereafter are not considered complete until the machine position is within the specified tolerances of the programmed targets. A succeeding move command cannot begin until the preceding command is complete. The move target tolerances are reset to their default value at a part program initialization.

If values are not specified then the current settings for the X, Y and Z move target tolerances are returned to variables S7, S8 and S10 respectively.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-090 DCC is not available.

TV - Set Touch Target Tolerance

The TV command sets the touch target tolerances for the X, Y and Z axes. These tolerances are used to determine whether it is OK to begin a programmed DCC touch command and to determine when each programmed touch command is complete if no touch occurs. The default value for target tolerances for the X, Y and Z-axes set at part program initialization is 1 mm (0.039 in.) for touches.

Command Format

MLString "TV P1,P2,P3"

Examples

1. MLString "TV .25,.25,.1"
2. MLString "TV"

Input

- **P1** = X tolerance (Numeric; optional).
- **P2** = Y tolerance (Numeric; optional).
- **P3** = Z tolerance (Numeric; optional).

P1, P2, and P3 specify the machine coordinate touch target tolerances for the X, Y, and Z axes respectively in current measurement units. In the first example, the tolerance for the X axis is 0.25, the tolerance for the Y axis is 0.25, and the tolerance for the Z axis is 0.1.

If any of P1, P2 and P3 are specified then all three are required. The target tolerance is a positive distance. If any of the specified values are less than zero, then the absolute value is used instead.

If P1, P2 and P3 are not specified, the current settings for touch target tolerances are returned to MLB variables.

Results

If values are specified then the touch target tolerances are set to the specified values. Touch commands thereafter will not start until the machine position is within the specified tolerances of the programmed targets of the preceding move. If a touch command does not make contact with a part surface then it will be considered complete when the machine position is within the touch target tolerances of the position defined by the touch target and the overdrive distance.

If values are not specified then the current settings for the X, Y, and Z touch target tolerances are returned to variables S7, S8, and S10 respectively.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-090 DCC is not available.

TW - Set Minimum After-Move-But-Before-Touch Time Interval

The TW command sets a minimum time interval that is to elapse between the end of a high-speed move and a subsequent surface touch. This can be used to allow machine vibrations to damp out before a programmed touch probe hit.

Command Format

MLString "TW P1"

Examples

1. MLString "TW"
2. MLString "TW 1"

Input

- **P1** = Time interval in seconds (Numeric; optional).

P1 sets the minimum time interval in seconds between the end of a high speed DCC traverse move and a subsequent surface touch.

If P1 is not specified, the current setting for the time interval is returned.

Results

If P1 is not specified, the current setting for the time interval is returned to variable S7.

Otherwise the system stores the new value. On subsequent touch commands the system assumes that the touch begins the clearance (see the **CL** command) distance away from the surface. If the time required to traverse the clearance distance at touch velocity (see the **TD** command) is less than the time specified by the **TW** command then a delay automatically occurs. The length of this delay is computed so that the total of the delay plus the time required to move the clearance distance at touch speed is equal to the specified time interval.

Skip Action

Ignored; not executed if within a block of skipped code.

Error Messages

- MLB-090 DCC is not available.

17. Miscellaneous Group

MLB commands in the Miscellaneous Group contain utility and miscellaneous commands.

The Miscellaneous Group includes the following MLB commands.

AF	**Change Feature Number** allows the feature number to change within a part program.
BB	**Begin Block** marks the beginning of a code block.
BQ	**Begin Skip** marks the beginning of the coded block for a skip sequence.
E0	**Lower APC Rack on ProGage CMM**
E1	**Raise APC Rack on ProGage CMM**
EQ	**End Skip** ends a block skip sequence.
IL	**Initialize Library** initializes the Measurement Library.
IO	**IOSTAT** allows you to sample the status of one of the input lines or change the status of one of the output lines.
IS	**Initialize System** initializes the complete CMM system.
PI	**Pallet ID** reads the Pentrax pallet ID.
R0	**Remeasure Off** turns the remeasure mode off.
R1	**Remeasure On** turns the remeasure mode on for non-DCC operations.
RV	**Firmware Version and MP Type**
TL	**Tolerance Lights** allows you to set a part status variable so the system can turn on a specified tolerance light.
UA	**Label User Array Rows** allows a part program to label rows of the user array, and to define the type of processing to be performed on a row.
UL	**Shuttle** allows you to start the shuttle sequence on Pentrax systems.
VP	**Rectangular Conversion** converts nominal polar radius and angle to their equivalent nominal rectangular coordinates.
VR	**Polar Conversion** converts nominal rectangular coordinates to their equivalent nominal polar radius and angle.

AF - Change Feature Number

The AF command allows the feature number to be changed by a part program.

Command Format

MLString "AF P1"

Example

MLString "AF 100"

Input

- **P1** = New feature number; greater than zero (Numeric; required).

P1 specifies the feature number assigned to the next measured/constructed feature. The feature number must be no less than one. In the example, the part program gives the next measured feature the feature number of 100.

Results

The feature number is reset to reflect the P1 argument. Additional feature numbers will be assigned sequentially beginning with the P1 value.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-021 The first parameter is invalid.

BB - Begin Block

The BB command marks the beginning of a code block.

Command Format

MLString "BB P1"

Example

MLString "BB FRONTFACE"

Input

- **P1** = Maximum of 10 characters (String; required).

P1 gives an identifying name to the program block. The name can be 10 or fewer alphanumeric characters.

Results

The command sets the internal flags that indicate the beginning of a block.

Skip Action

An error occurs if this command occurs within a block of code being skipped.

Error Messages

- MLB-014 There is a nested code block.

BQ - Begin Skip

The BQ command marks the beginning of a code block for a skip sequence. This is a set of commands within a code block (see the **BB** and **EB** commands) which are executed when the normal code is being skipped. The skip block of code must be terminated by an **EQ** command.

Command Format

MLString "BQ"

Example

1. MLString "BQ"

Input

None.

Results

The **BQ** command sets internal flags to indicate the start of a skip sequence.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-016 There is a nested skip sequence. A second **BQ** command occurred before an **EQ** command was found.

E0 - Lower APC Rack on ProGage CMM

The E0 command lowers the APC rack on a ProGage CMM.

Command Format

MLString "E0"

Examples

1. MLString "E0"

Input

None.

Results

The APC rack is lowered if the target CMM is a ProGage; otherwise the command is accepted but ignored.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-161 occurs if the APC rack could not be lowered.

E1 - Raise APC Rack on ProGage CMM

The E1 command raises the APC rack on a ProGage CMM.

Command Format

MLString "E1"

Examples

1. MLString "E1"

Input

None.

Results

The APC rack is raised if the target CMM is a ProGage; otherwise the command is accepted but ignored.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-162 occurs if the APC rack could not be raised.

EB - End Block

The EB command marks the end of a code block that started with a **BB** command.

Command Format

MLString "EB"

Examples

1. MLString "EB"

Input

None.

Results

The command marks the end of the block of code that began with the previous **BB** command.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-015 There is no matching block beginning.

EI - Electronic Level Interface Reading

Get current reading from the ATLAS electronic levels.

Command Format

MLString "EI"

Example

1. MLString "EI"

Input

None.

Results

The character "?" is sent to the electronic level requesting a single reading. The response from the electronic level is stripped of carriage control characters and returned to the host. This string is stored in variable S0D.

Skip Action

Ignored; not executed when inside a skip block.

Error Messages

- MLB-131 The electronic level is not available.
- MLB-132 Electronic level timeout error. No response was received from the electronic level within 5 (five) seconds.

EQ - End Skip

The EQ command marks the end of a skip code block that started with an **EB** command.

Command Format

MLString "EQ"

Example

1. MLString "EQ"

Input

None.

Results

The command marks the end of the block of code that began with the previous **EQ** command.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-017 There is no matching skip sequence beginning.

IL - Initialize Library

The IL command initializes the Measurement Library. Variables revert to the default state at system initialization. This command does not clear system errors caused by a fault in the system hardware. Separate corrective action must be taken for such errors.

Typically, an IL command is automatically executed at the beginning of each part program; however, there is a setting in MeasureMax+ that can be used to disable this.

Command Format

MLString "IL"

Examples

1. MLString "IL"

Input

None.

Results

The system:

- Deletes all measured features,
- Resets the part scale ratios to 1:1,
- Cancels axis translations and rotations,
- Sets the feature number and sequence number to zero,
- Reverts to the default values for measurement units and the working plane, and
- Clears all other variables or sets them to their default values.

This command does not affect:

- Selected tip or probe diameter,

- Calibration of any probe tips,
- Calibration of the rotary table,
- Saved reference frames 1 through 48; however, other reference frames are deleted, or
- Machine scale factors.

The setting for ‘suppress IL’ in the CMM Station Properties of MeasureMax+ and MaxLite determines whether an IL is automatically generated at the beginning of each part program. This setting is normally disabled; thereby, forcing an IL for each part program. If the ‘suppress IL’ setting is selected (that is, the checkbox is checked), then it is the programmer’s responsibility to ensure safe measurement techniques. Suppressing the IL command affects not only part programs run by the MeasureMax+ Control Center, MaxLite and Single Touch Interface, but it also affects the initialization of QuickTeach and measurements and part programs run within QuickTeach.

Skip Action

This command may not occur in a section of code to be skipped. Error.

Error Messages

None.

IO - IOStat

The IO command allows testing of one of the input (read-only or read-write) bits or changing the status of one of the output (write-only or read-write) bits.

Command Format

MLString "IO P1,P2"

Examples

1. MLString "IO 3,1"
2. MLString "IO 2"

Input

- **P1** = 0 through 67 (Numeric; required).
- **P2** = 0 or 1 (Numeric; optional).

P1 is a number which corresponds to the input or output bit whose status is being viewed or changed. P1 must be a digit from 0 through 67. Some bits are read only, some are write only, and others are read and write. Consult Giddings & Lewis for details.

If P2 is specified, P1 identifies an output (write-only or read-write) bit. P2 may not be specified with a read-only bit. P2 indicates the state to which the output bit identified by P1 is to be changed. P2 must be either a zero (0) or a one (1).

The output bits are like switches which may be opened or closed. For the Control Board, when P2 is zero (0), the switch is closed, and when P2 is one (1), the switch is open. On the Auxiliary Digital Input Output Board and the Temperature Compensation Board, the switch is opened when P2 is zero (0), and the switch is closed when P2 is one (1). In the first example, the state of the write-only bit number 3 is being changed to one.

If P2 is not specified, P1 identifies an input (read-only or read-write) bit. The programmer can only view the status of a bit which has been identified as an input bit. The status of an input bit cannot be changed. In the second example, the bit whose status is to be viewed is the read-only bit #2 on the Control Board.

Results

If P2 is not specified, the current status of the input bit identified by P1 is stored in the variable S7. The variable S7 must be checked immediately after executing this command to verify the status of the input bit. If S7 is zero (0), there is no current flow into the input bit. If S7 is one (1), there is current flow into the bit. This is true for the Control Board and the Auxiliary Digital Input Output Boards. For the Temperature Compensation Board, if S7 is zero, the input bit is an open circuit (+V). If S7 is one, the input bit is ground (OV). For the read-write input bits, S7 contains the last value written to the output bit.

If P2 is specified in the **IO** command, the value of S7 is meaningless. Note that all input bits and/or output bits may not be available to the user. Availability is dependent on the MP configuration purchased.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-005 Communication error.
- MLB-006 Output mask logical translation error.
- MLB-007 Output unavailable.
- MLB-021 First Parameter Invalid.
- MLB-022 Second Parameter Invalid.

IS - Initialize System

The IS command initializes the complete system. It places the system in the same state it is in when system power is first turned on.

Command Format

MLString("IS")

Examples

1. MLString("IS")

Input

None.

Results

Variables go back to the same state they are in when system power is first turned on. The system does the following:

- Deletes all measured features,
- Deletes all saved reference frames,
- Sets the part and machine scale ratio to 1:1,
- Cancels axis translations and rotations,
- Sets the feature number and sequence number to zero,
- Goes back to the default values for measurement units and the working plane,
- Clears all other variables or sets them to the default values,
- Deletes the calibration of the rotary table and the machine scale ratio, and
- Selects tip #0 as the active tip.

Skip Action

Not permitted. Error.

Error Messages

None.

PI - Pallet ID

The PI command reads the PENTRAX pallet identification.

Command Format

MLString "PI"

Examples

1. MLString "PI"

Input

None.

Results

The pallet identification code is returned to the part program in variable S14.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-005 Communication Error.

R0 - Remeasure Off

The R0 command turns the remeasure mode off. Use the **R1** command to turn the remeasure mode on.

Command Format

MLString "R0"

Examples

1. MLString "R0"

Input

None.

Results

The remeasure mode is turned off. The operator will not be prompted to remeasure manually measured features if they are out of tolerance.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

R1 - Remeasure On

The R1 command turns the remeasure mode on. Use the **R0** command to turn the remeasure mode off.

Command Format

MLString "R1"

Examples

1. MLString "R1"

Input

None.

Results

The remeasure mode is turned on. While remeasure mode is on the results of every manual feature measurement will be checked for out-of-tolerance conditions based upon the current report format (format #0), the nominal and tolerance data in the Dimensional Array, and the current tolerance mode settings (**BL** / **LM** and **RS** / **MM**). If the feature is out of tolerance then a temporary report output will be generated and the operator will be asked whether he wants to remeasure the feature.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

RV - Firmware Version and MP Type

Determine SMP or RCU firmware version number and SMP type information.

Command Format

MLString "RV P1"

Examples

1. MLString "RV 0"

   Determine SMP firmware and host MLB software version numbers.

2. MLString "RV 8"

   Determine DCC / SMP firmware version number.

3. MLString "RV 96"

   Determine expansion ROM firmware version number. Note this is only valid if the MP firmware version number is less than 7.

4. MLString "RV"

   Return the MP / SMP firmware version number.

5. MLString "RV 1"

   Return the SMP BIOS firmware version number.

6. MLString "RV -1"

   Return the RCU firmware version number.

Input

- **P1** = 0, 1, 8, or 96 (Numeric; optional).

For the MP30/35 prior to version 17, parameter P1 is required and is a number identifying the MP firmware set for which the version number is being requested. The only valid values for P1 are:

P1	Results for MP30/35 prior to version 17
-1	RCU firmware
0	MP / SMP firmware and host MLB software
1	SMP BIOS version number
8	DCC firmware
96	PH9/Soft probe/APC expansion firmware (prior to MP version 7)

For the SMP-400, MP 30/35 versions at least 17, and SMP-3xx, parameter P1 may be omitted. If P1 is omitted then the MP/SMP firmware version number is returned.

Results

The requested version number is returned into variable S7. If the requested version number is for the RCU, the returned value is one of the following:

S7	RCU Description
-1	RCU not present
0	Original style RCU (RCU-I)
n	RCU-II, version <n>; n >= 1

The requested version number is returned into variable S7. If the version number is at least 17 but not 101, the hundredth digit (second digit to right of the decimal point) identifies the measurement processor type as follows:

S7	MP Type
0	MP30, MP35, or MP30-II
1	1 SMP-400
2	SMP-3xx

For example, a value of 17.00 would be an MP30 (or MP35 or MP30-II) version 17.0 while a value of 17.02 would be an SMP-3xx version 17.0.

Note that version numbers less than 17 correspond to MP30 series processors while version number 101.0 is the initial release of the SMP-400. For these units the hundredths digit is always returned as 0.

If the version number request is 0 (zero) then the returned value is a combination of the SMP version number and the host MLB version number. The returned value is 100,000 times the host MLB version number plus the firmware version number. Thus, a value of 100,019.02 indicates host MLB version 1.00 together with an SMP-3xx version 19.0.

Skip Action

Processed normally.

Error Messages

- MLB-160 Firmware entity corresponding to P1 not found.

TL - Shuttle

Used with Pentrax systems, the **TL** command sets a part status variable so the system turns on a specified tolerance light.

Any report command (**RP**) that reports an out-of-tolerance condition also automatically sets the system variable SOTF to TRUE. It remains TRUE until reset to FALSE again by an initialization (**IL**) command.

At the end of a part program SOTF can be checked for an out-of-tolerance condition. Then the **TL** command can be used to turn off and on different lights representing various conditions.

Command Format

MLString "TL P1"

Examples

1. MLString "TL 2"
2. MLString "TL 1"

Input

- **P1** = Numeric; required.

P1 may be any of the following:

- 0 - turn lights off
- 1 - turn accept light on
- 2 - turn reject light on
- 3 - turn caution light on

Results

The **TL** command sets an internal variable that directs an optional control task to turn on the tolerance lights. As the pallet is shuttled off the CMM, the part status lights will show the part status.

Skip Action

Executes even if in a skipped block.

Error Messages

- MLB-019 Invalid parameter.

UA - Label User Array Rows

The UA command creates and assigns a label to a data type stored in the User Array which is a separate storage area from the Dimensional Array. Functionality of the User Array is similar to the Dimensional Array.

Command Format

MLString "UA *p1*,*p2*,*p3*"

Examples

1. MLString "UA 7,0,LD"

Input

- *p1* = Row number 0 to 14 (Numeric; required).
- *p2* = Processing type 0, 1, 2, or 3 (Numeric; required).
- *p3* = New data type name with one or two characters (String; required).

p1 specifies the row number in the user array, system variable SU. *p1* may be a number from 0 to 14.

p2 specifies the processing type for the row. Processing types are represented by the numbers 0 to 3. In the table below, each processing type is listed along with the action taken when the processing type is specified. The "S" array row for which the processing type is used is also included in the table.

- Processing Type 0 is used for normal linear data such as that found in the Dimensional Array rows 0, 2, 3, 4, 5, 6, 7, 14, 15, 16, and 17. The deviation and out-of-tolerance are calculated from the first four columns.
- Processing Type 1 is used for linear data which does not have nominal or minus tolerance values such as that found in the Dimensional Array rows 10,11,12, and 13. The out-of-tolerance is calculated from columns 2 and 4 of the array and is stored in column 5.
- Processing Type 2 is used for angular data such as that found in the Dimensional Array rows 1 and 7. This is the same as type 0, except values are treated as angles. These values will print out on reports as decimal fractions of degrees.
- Processing Type 3 is used for angular data such as that in the Dimensional Array rows 1 and 7 which is to be reported in degrees/minutes/seconds format.

p3 specifies the row label to be used to reference this data in the RP command. *p3* may be 1 to 2 characters long.

Data is stored into the User Array actual, nominal, plus tolerance, and minus tolerance columns using standard BASIC assignment commands such as:

- objData.SU(7,1)=17.5
- objData.SU(7,2)=.2
- objData.SU(7,3)=.2

Note that deviation values also need to be stored programmatically for data identified as Type 1.

The data from the User Array can be included in the Inspection Report by including the identifier defined above in a format command defined by either the FM or the RP command.

User Array row labels are kept in an array called SULABEL(0:14).

Results

The row label (*p3*) is saved for use in formats defined and/or used by the FM and RP commands. The new labels are also used to label rows from the User Array on the report generated. No User Array data is logged. The processing type specified (*p2*) is used to determine the calculations performed in calculating deviation and/or out-of-tolerance values. The processing type specified also determines the form in which angular measurements are printed.

The IN and ME commands cause the same rescaling of data in the user array as they cause in the Dimensional Array.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-021 First parameter invalid.
- MLB-022 Second parameter invalid.

UL - Shuttle

The UL command starts the pallet shuttle sequence on Pentrax systems.

It is the responsibility of the part program to cause the CMM to remain at the home position once the shuttle command has been issued.

Command Format

MLString "UL"

Examples

1. MLString "UL"

Input

None.

Results

For I-Loaders: The part will be shuttled to the load/unload station when the CMM is at the home position, and all gates / doors are closed.

For T-Loaders: The part will be shuttled to the load/unload station when the CMM is at the home position, all gates/doors are closed, and the palm buttons are pressed.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

VP - Rectangular Conversion

The VP command uses the nominal polar radius and angle values in the Dimensional Array to compute the equivalent nominal rectangular coordinates and store these in the Dimensional Array.

Command Format

MLString "VP"

Examples

1. MLString "VP"

Input

None.

Results

The nominal rectangular coordinates for the active plane are computed from the nominal polar radius and angle values and stored in the Dimensional Array.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

VR - Polar Conversion

The VR command uses the nominal working plane rectangular coordinates in the Dimensional Array to compute the equivalent polar radius and angle values and store these in the Dimensional Array.

Command Format

MLString "VR"

Examples

1. MLString "VR"

Input

None.

Results

The nominal polar radius and angle values are computed from the nominal rectangular coordinates for the active plane and stored in the Dimensional Array.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

18. Special Measurement Group

MLB commands in the Special Measurement Group measure bolt circle patterns and rectangular patterns.

The Special Measurement Group includes the following MLB commands.

BA	**DCC Bolt Circle** measures and reports a bolt circle pattern under DCC control.
BN	**Manual Bolt Circle** measures and reports a bolt circle pattern under manual control.
RA	**DCC Rectangular Pattern** measures and reports a grid of circular features under DCC control.
RM	**Manual Rectangular Pattern** measures and reports a grid of circular features under manual control.

BA - DCC Bolt Circle

The BA command, under DCC control, does several commands in one step. It measures IDs or ODs in a circular pattern in the active working plane and prints inspection reports for the bilateral tolerancing mode.

Using the **CM** command, set probe compensation to INT for IDs (inner diameters) and to EXT for ODs (outer diameters). The system treats compensation set to OFF as an ID.

Each circle in the bolt circle pattern must be present, evenly spaced, and have the same diameter. The pattern does not have to cover 360 degrees. More than one pattern may overlap, sharing the same bolt circle diameter.

After the CMM measures each circle in a bolt circle pattern, the system computes its "actual" center (in rectangular and polar coordinates) and diameter. Then it stores the results in the Feature Register (FR) and prints an inspection report using the first print format you supply.

After the last circle is measured, the system computes the center (in rectangular and polar coordinates) and diameter of the bolt circle pattern from all circles measured, and stores these results in FR. This report prints using the second print format supplied.

After a measurement, the system automatically compares the actual measurement to the nominal, and computes deviation and out-of-tolerance values. Then it prints the report.

Therefore, the nominals must be defined and the two print formats created before using this command. The print format for individual circles might include:

- Center coordinates (X, Y, Z)
- Diameter

- Form
- True position (another way to report deviation and out-of-tolerance)
- Polar radius (distance from the bolt circle pattern center to the circle center)
- Polar angle (See note below)

Note: A point in polar coordinates is a radius and an angle. The radius is the distance from the point to the unnamed axis of the working plane (where the axes cross and both are datum). The angle starts with 0 at the plus end of the first-named axis (+X in XY), and increases toward the plus end of the second-named axis (+Y in XY). The center of the polar circle is the unnamed axis (Z in XY), or where both working plane axes are datum (0).

The report for the whole bolt circle might include:

- Center coordinates (X, Y, Z)
- Bolt pattern diameter
- True position (for deviation and out-of-tolerance of the centerpoint)
- Polar radius (distance from the bolt circle's center point to where both axes are datum)
- Polar angle (angular location of the bolt circle's center) See earlier note on polar coordinates.

Command Format

MLString "BA P1,P2,P3,P4,P5,P6,P7"

Examples

1. MLString "BA 6,12,90,0,2,1"
2. MLString "BA 6,12,90,0,2,1,BC17"

Input

- P1 = Number of points to take on each circle (in the examples above, 6 points). (Numeric; required.)
- P2 = Number of circles in the bolt circle (12 in the examples). (Numeric; required.)
- P3 = Polar angle of the first circle's center around the bolt circle center, as if the bolt circle center is datum for both axes. The center doesn't have to be datum for both axes. (In the examples, 90 degrees means the first circle's center would be on the +Y axis in XY, if the bolt circle center was datum for both axes). See the previous note on polar coordinates. (Numeric; required.)
- P4 = Angle between the circles, relative to the bolt circle center. (In the examples, 0 degrees means the circles cover 360 degrees. Put 30 degrees if they do not.) (Numeric; required.)
- P5 = Print format number for the individual circles. (In the examples, the format is 2.) (Numeric; required.)
- P6 = Print format number for the bolt circle. (In the examples, the format is 1.) (Numeric; required.)
- (MeasureMax+ only) P7 = Start of names for saving features, 7 characters maximum. If P7 is specified, the individual features will be saved as <P7>.1, <P7>.2, etc. In the second example, the individual features will be saved as BC17.1, BC17.2, etc. If P7 is not specified then the individual features are pushed into the feature stack but not saved in named feature storage. (String; optional)

The following information is passed through the Dimensional Array:

- S(S1,1) Bolt circle center nominal along the first-named axis (S1=2, 3, or 4).

- S(S2,1) Bolt circle center nominal along the second-named axis (S2=2, 3, or 4).
- S(0,1) Bolt circle radius nominal (half of the bolt circle pattern's diameter).
- S(5,1) Diameter nominal of the individual circles.
- S(S3,1) The coordinate along the unnamed axis S3 that describes what depth (IDs) or height (ODs) to take the points (S3=2, 3, or 4).

Other inputs that might be passed through the Dimensional Array are:

- S(0,2) Polar radius plus tolerance. Use the bolt circle pattern diameter plus tolerance.
- S(0,3) Polar radius minus tolerance. Use the bolt circle pattern diameter minus tolerance.
- S(1,2) Polar angle plus tolerance (for the individual circles).
- S(1,3) Polar angle minus tolerance (for the individual circles).
- S(5,2) Plus tolerance for the diameter of the individual ID/ODs.
- S(5,3) Minus tolerance for the diameter of the individual ID/ODs.
- S(S1,2) Plus tolerance along the 1st-named axis (S1) for the individual circle centers.
- S(S1,3) Minus tolerance along the 1st-named axis (S1) for the individual circle centers.
- S(S2,2) Plus tolerance along the 2nd-named axis (S2) for the individual circle centers.
- S(S2,3) Minus tolerance along the 2nd-named axis (S2) for the individual circle centers.

Other required inputs:

- Calibrate and select the proper probe tip.
- Set the altitude (**AL** command) to a height along the unnamed axis that makes the probe clear obstacles. Altitude is from the point taking level, not from datum!
- Make sure the probe has a clear path to the first circle. The probe starts over the first ID/OD at the height assigned with the **AL** command.
- Assign the necessary nominals and tolerances before using this command.
- Create the two print formats before using this command.

Results

First, the system saves the bolt circle's center coordinates. After the CMM measures each circle, the system stores the results in the Feature Register (FR) and moves all data in the stack down one position. If P7 is specified (MeasureMax+ only), the feature is also named and saved to feature storage as described above. Nominals for polar angle and for the axis coordinates are created from the starting angle and angular spacing specified in the **BA** command, and from the polar radius assigned to S(0,1). The circle's center point sums into the Internal Summation Array. Then an inspection report is printed using the first report format.

After the CMM measures the last circle, **CM** becomes "OFF" and a the system creates a "best fit circle" from the Internal Summation Array. The bolt circle center results store in the Feature Register (FR) and all data in the stack moves down one position.

The system returns the working plane axis nominals for the bolt circle's center to the Dimensional Array. And it computes its polar radius and angle nominals. Then it prints an inspection report using the second report format.

The tolerances for the individual circle centers and the bolt circle center may be different. There is only one place to put the tolerances. New tolerances cannot be entered into the Dimensional Array before the system prints the report for the bolt circle center with the second format. Therefore, if the bolt circle center has different tolerancing than the individual circles, use the tolerances in the Dimensional Array for the individual circles (the working plane axes tolerances and the true position plus tolerance). Then set P6 to -1. This skips the report for the bolt circle center.

When the **BA** command completes, the bolt circle's center point data is in the actual column of the Dimensional Array. After the **BA** command, set the working plane axes tolerances and true position plus tolerance to the bolt circle center's tolerances. Then print the inspection report for the bolt circle center.

Skip Action

Processes normally. When skipped, the **BA** command processes without any actual measurement. A null feature is pushed into the Feature Register/Stack for each feature in the pattern and one for the bolt circle. The sequence number and feature numbers increment once for each circle in the bolt circle, plus once for the bolt circle itself. This means if there were 4 bolt holes, the next inspection report's sequence and feature number has incremented by 5.

Error Messages

- MLB-018 Improper tolerancing mode. (Bilateral tolerancing required.)

BN - Manual Bolt Circle

The BN command is identical to the **BA** command in operation, and has the same inputs. (See the **BA** command.) The only difference is that the CMM prompts the operator to measure each circle manually instead of going into the DCC mode.

There is no need to set an altitude with the **AL** command. And it is not necessary to set a depth/height along the unnamed axis for the point taking level.

Command Format

MLString "BN P1,P2,P3,P4,P5,P6,P7"

Examples

1. MLString "BN 6,12,90,0,2,1"
2. MLString "BN 6,12,90,0,2,1,BC17"

Input

See the inputs P1 through P7 for the **BA** command.

Results

The results are identical to the **BA** command.

Skip Action

The skip action is identical to the **BA** command.

Error Messages

- MLB-018 Improper tolerancing mode. (Bilateral tolerancing required.)

RA - DCC Rectangular Pattern

The RA command, under DCC control, does several commands in one step. It measures IDs or ODs in a rectangular pattern in the active working plane and prints inspection reports.

Using the **CM** command, set probe compensation to INT for IDs (inner diameters) and to EXT for ODs (outer diameters). The system treats compensation set to OFF as an ID.

Each circle in the rectangular pattern must be present, evenly spaced, and have the same diameter.

After each circle in the rectangular pattern is measured, the system computes its "actual" center (in rectangular and polar coordinates) and diameter. Then it stores the results in the Feature Register (FR) and prints an inspection report using the print format supplied.

Command Format

MLString "RA P1,P2,P3,P4,P5,P6,P7,P8"

Examples

1. MLString "RA 5,3,4,25,30,1,3"
2. MLString "RA 5,3,4,25,30,1,3,RP17"

Input

- **P1** = Number of points to take on each circle (in the example, 5). (Numeric; required.)
- **P2** = Number of circles in the pattern along the first-named axis of the working plane. (Numeric; required.)
- **P3** = Number of circles in the pattern along the second-named axis of the working plane. (Numeric; required.)
- **P4** = Nominal spacing between circle centers along the first-named axis. (Numeric; required.)
- **P5** = Nominal spacing between circle centers along the second-named axis. (Numeric; required.)
- **P6** = Order of measurement. (Numeric; required.)
- **P7** = Report format number. (Numeric; required.)
- (MeasureMax+ only) P8 = Start of names for saving features, 7 characters maximum. (String; optional)

P1 specifies the number of points to be measured on each individual circle in the pattern. In the example, 5 points are to be measured on each feature.

P2 specifies the number of features in each row of the pattern along the first-named axis of the working plane. In the example, there are three features in each row.

P3 specifies the number of features in each column of the pattern along the second-named axis of the working plane. In the example, there are four features in each column.

P4 specifies the nominal spacing between feature centers along the first-named axis. In the example, there are 25 units between feature centers in this direction.

P5 specifies the nominal spacing between feature centers along the second-named axis. In the example, there are 30 units between feature centers in this direction.

P6 specifies the pattern that is to be followed in measuring the features in the pattern. The following list shows all possible P6 values:

- -1 = Square-by-row. Measure all features in a row, then move vertically to the next row and measure the features in that row moving in the opposite direction from the preceding row.
- 1 = Diagonal-by-row. Measure all features in a row, then move diagonally to the next row, returning to the first feature in that row, and then measure the features in that row moving in the same direction as the preceding row.
- -2 = Square-by-column. Measure all features in a column, then move horizontally to the next column and measure the features in that column moving in the opposite direction from the preceding column.
- 2 = Diagonal-by-column. Measure all features in a column, then move diagonally to the next column, returning to the first feature in that column, and then measure the features in that column moving in the same direction as the preceding column.

In the example, the features are to be measured in 'diagonal-by-row' order.

P7 is a number that describes the inspection report print format for the features. A negative number means do not print the report. In the example, the inspection reports print using format 3.

P8 is a string used to name and save (see the **SF** command) the individual features. If P8 is specified, the individual features will be saved as <P8>.1, <P8>.2, etc. In the second example, the individual features will be saved as RP17.1, RP17.2, etc. If P8 is not specified then the individual features are pushed into the feature stack but not saved in named feature storage.

Other required inputs:

- Calibrate and select the proper probe tip.
- Set the altitude (see the **AL** command) to a height along the unnamed axis that makes the probe clear obstacles between features. Altitude is from the point taking level, not from the datum!
- Make sure the probe has a clear path to the first feature. The probe starts over the first feature at the height you assign using the **AL** command.
- Assign any necessary nominals and tolerances.
- Create the print format using the **FM** command.

The following required information is passed through the Dimensional Array:

- S(5,1) Nominal diameter for the pattern circles.
- S(S1,1) Nominal location of the 1st circle's center along the first-named axis. S1=2, 3, or 4.
- S(S2,1) Nominal location of the 1st circle's center along the second-named axis. S2=2, 3, or 4.
- S(S3,1) Coordinate along the unnamed axis that describes the level at which to take the points. S3=2, 3, or 4.

If required by the report format, the following information should be passed through the Dimensional Array:

- S(5,2) Plus tolerance for the diameter of the pattern circles.
- S(5,3) Minus tolerance for the diameter of the pattern circles.
- S(S1,2) Plus tolerance for the centers of the pattern circles along the first-named axis.
- S(S1,3) Minus tolerance for the centers of the pattern circles along the first-named axis.
- S(S2,2) Plus tolerance for the centers of the pattern circles along the second-named axis.
- S(S2,3) Minus tolerance for the centers of the pattern circles along the second-named axis.

Results

The circles in the pattern are measured in the specified order using internally generated **AC** commands. After each circle measurement, the system computes the center (in rectangular and polar coordinates) and diameter. Then it stores the actual results in the Feature Register (FR) and moves all data in the stack down one position. If P8 is specified (MeasureMax+ only), the feature is also named and saved to feature storage as described above. After computing deviation and out-of-tolerance values, it prints an inspection report using the specified print format. At the end, the Feature Register contains the result of the last feature measured.

Skip Action

Processed normally. When skipped, the **RA** command processes without any actual measurement. A null feature is pushed into the Feature Register/Stack for each feature in the pattern. The sequence number and feature numbers increment once for each feature in the pattern. This means if there were 12 features in the pattern, the next inspection report's sequence and feature number has incremented by 12.

Error Messages

- MLB-018 Improper tolerancing mode. (Bilateral tolerancing required.)

RM - Manual Rectangular Pattern

The RM command is identical to the **RA** command in operation, and has the same inputs. (See the **RA** command.) The only difference is that the CMM prompts the operator to measure each circle manually instead of going into the DCC mode.

There is no need to set an altitude with the **AL** command. And it is not necessary to set a depth/height along the unnamed axis for the point taking level.

Command Format

MLString "RM P1,P2,P3,P4,P5,P6,P7,P8"

Example

1. MLString "RM 5,3,4,25,30,1,3"
2. MLString "RM 5,3,4,25,30,1,3,RP17"

Input

- **P1** = Number of points to take on each circle (in the example, 5). (Numeric; required.)
- **P2** = Number of circles in the pattern along the first-named axis of the working plane. (Numeric; required.)
- **P3** = Number of circles in the pattern along the second-named axis of the working plane. (Numeric; required.)
- **P4** = Nominal spacing between circle centers along the first-named axis. (Numeric; required.)
- **P5** = Nominal spacing between circle centers along the second-named axis. (Numeric; required.)

- **P6** = Order of measurement. (Numeric; required.)
- **P7** = Report format number. (Numeric; required.)
- (MeasureMax+ only) **P8** = Start of names for saving features, 7 characters maximum. (String; optional)

P1 specifies the number of points to be measured on each individual circle in the pattern. In the example, 5 points are to be measured on each feature.

P2 specifies the number of features in each row of the pattern along the first-named axis of the working plane. In the example, there are three features in each row.

P3 specifies the number of features in each column of the pattern along the second-named axis of the working plane. In the example, there are four features in each column.

P4 specifies the nominal spacing between feature centers along the first-named axis. In the example, there are 25 units between feature centers in this direction.

P5 specifies the nominal spacing between feature centers along the second-named axis. In the example, there are 30 units between feature centers in this direction.

P6 specifies the pattern that is to be followed in measuring the features in the pattern. The following list shows all possible P6 values:

- -1 = Square-by-row. Measure all features in a row, then move vertically to the next row and measure the features in that row moving in the opposite direction from the preceding row.
- 1 = Diagonal-by-row. Measure all features in a row, then move diagonally to the next row, returning to the first feature in that row, and then measure the features in that row moving in the same direction as the preceding row.
- -2 = Square-by-column. Measure all features in a column, then move horizontally to the next column and measure the features in that column moving in the opposite direction from the preceding column.
- 2 = Diagonal-by-column. Measure all features in a column, then move diagonally to the next column, returning to the first feature in that column, and then measure the features in that column moving in the same direction as the preceding column.

In the example, the features are to be measured in 'diagonal-by-row' order.

P7 is a number that describes the inspection report print format for the features. A negative number means do not print the report. In the example, the inspection reports print using format 3.

P8 is a string used to name and save (see the SF command) the individual features. If P8 is specified, the individual features will be saved as <P8>.1, <P8>.2, etc. In the second example, the individual features will be saved as RP17.1, RP17.2, etc. If P8 is not specified then the individual features are pushed into the feature stack but not saved in named feature storage.

Other required inputs:

- Calibrate and select the proper probe tip.
- Set the altitude (see the AL command) to a height along the unnamed axis that makes the probe clear obstacles between features. Altitude is from the point taking level, not from the datum!

- Make sure the probe has a clear path to the first feature. The probe starts over the first feature at the height you assign using the AL command.
- Assign any necessary nominals and tolerances.
- Create the print format using the FM command.

The following required information is passed through the Dimensional Array:

- S(5,1) Nominal diameter for the pattern circles.
- S(S1,1) Nominal location of the 1st circle's center along the first-named axis. S1=2, 3, or 4.
- S(S2,1) Nominal location of the 1st circle's center along the second-named axis. S2=2, 3, or 4.
- S(S3,1) Coordinate along the unnamed axis that describes the level at which to take the points. S3=2, 3, or 4.

If required by the report format, the following information should be passed through the Dimensional Array:

- S(5,2) Plus tolerance for the diameter of the pattern circles.
- S(5,3) Minus tolerance for the diameter of the pattern circles.
- S(S1,2) Plus tolerance for the centers of the pattern circles along the first-named axis.
- S(S1,3) Minus tolerance for the centers of the pattern circles along the first-named axis.
- S(S2,2) Plus tolerance for the centers of the pattern circles along the second-named axis.
- S(S2,3) Minus tolerance for the centers of the pattern circles along the second-named axis.

Results

The circles in the pattern are measured in the specified order using internally generated AC commands. After each circle measurement, the system computes the center (in rectangular and polar coordinates) and diameter. Then it stores the actual results in the Feature Register (FR) and moves all data in the stack down one position. If P8 is specified (MeasureMax+ only), the feature is also named and saved to feature storage as described above. After computing deviation and out-of-tolerance values, it prints an inspection report using the specified print format. At the end, the Feature Register contains the result of the last feature measured.

Skip Action

Processed normally. When skipped, the **RM** command processes without any actual measurement. A null feature is pushed into the Feature Register/Stack for each feature in the pattern. The sequence number and feature numbers increment once for each feature in the pattern. This means if there were 12 features in the pattern, the next inspection report's sequence and feature number has incremented by 12.

Error Messages

- MLB-018 Improper tolerancing mode. (Bilateral tolerancing required.)

19. Segment Measurement Group

MLB commands in the Segment Measurement Group let you measure a portion of a feature, change the probe, and then measure another portion of the same feature. You can measure an unlimited number of these portions, or segments. As you measure a segment, the internal summation array stores data for that segment. After you have measured all the segments, the Measurement Library computes the feature data and stores the computations in the Feature Register.

Segmented measurement uses the internal summation array to accumulate the recorded point positions. You cannot, therefore, use commands that use the summation array while you are measuring segments. Do not change probe compensation or change the current part reference frame during segmented measurement.

The Segment Measurement Group includes the following MLB commands.

NS	**Segment** sets internal flags for the measurement of segments of a feature.
SG	**Next Segment** processes measurements for multiple segments of a feature.

NS - Next Segment

The NS command processes measurements for all segments after the first segment of a segmented measurement. After measurement of the last segment, the command analyzes the accumulated measurement data. The Feature Register stores the best-fit feature data.

Command Format

MLString "NS P1,P2"

Examples

1. MLString "NS 4"

Input

- **P1** = @operator prompt@ (String; optional)
- **P2** = Number of points (Numeric; optional)

P1 is used to specify a message to be shown in the manual measure feature window that is displayed when this command is executed. This prompt should help the operator identify the correct feature and portion of that feature to be measured.

P2 specifies the number of points to measure in this segment. If P2 is omitted, the number of recorded points is the default value specified by the measurement command. In the example, the 4 specifies the number of points to measure in this segment. Omit P2 for sweep measurements.

Results

After the last measured segment, the contents of the Feature Register move into the Feature Stack. The Feature Register stores the results of the best-fit computations. The Dimensional Array stores the computed feature's:

- Position,
- Size, if applicable,
- Orientation,
- Form,
- Feature type, and
- Feature number.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-040 The summation array does not contain sufficient features.
- MLB-041 The points do not define the feature.
- MLB-044 Segmentation is not initialized.

SG - Segment

The SG command indicates that the next feature measurement is to be segmented.

Command Format

MLString "SG P1"

Examples

1. MLString "SG 3"

Input

- **P1** = Number of segments (Numeric; required).

The required numeric parameter P1 specifies the number of segments that make up the segmented measurement and must be at least one (1). The first segment is to be measured with an appropriate measure feature or sweep feature command. Subsequent segments are measured using the **NS** command.

Results

Internal flags are set so that the next manual feature measurement will be treated as the start of a segmented measurement.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-043 A segmented measurement is in progress.

20. Sweep Measurement Group

MLB commands in the Sweep Measurement Group records hard probe measurement of:

- circles
- 2-dimensional lines
- 3-dimensional lines
- planes
- spheres
- cylinder

The system records measurements as you sweep a hard probe in a continuous motion over the surface of a feature. If you try to use Sweep Measurement commands without a touch probe, you get an error message.

During a sweep feature measurement, the Prompt and XYZ Display monitors display messages to assist you. If measured points do not define a given feature, a message instructs you to sweep the feature again.

Sweep measurements store 200 points in the Internal Summation Array. These points are recallable. The system replaces some of the points saved in the summation array if more points are taken than can be saved. The points available after the feature measurement is complete are a relatively uniform sampling of all the points measured. All other points taken during the sweep feature measurement are stored for use by the system, but are not recallable by the user.

The Sweep Measurement Group includes the following MLB commands.

W2	**Sweep 2D Line** records a sweep measurement of a line as projected into the working plane. It computes the best-fit position and attitude of the line.
W3	**Sweep 3D Line** records a sweep measurement of a line. It computes the best-fit position and attitude of the line in three-dimensional space.
WC	**Sweep Circle** records a sweep measurement of a circle. It computes the best-fit of a circle.
WI	**Sweep Cylinder** records a sweep measurement of a cylinder. It computes the best-fit to a cylinder.
WN	**Define Minimum Sweep Point**
WP	**Sweep Plane** records a sweep measurement of a plane. It computes the best-fit to a plane.
WS	**Sweep Sphere** records a sweep measurement of a sphere. It computes the best-fit to a sphere.

W2 - Sweep 2D Line

The W2 command records points as an operator sweeps a hard probe along a surface and then computes a best-fit line parallel to the active working plane through the points recorded during the sweep operation.

Command Format

MLString "W2 P1"

Examples

1. MLString "W2 INT"

Input

- **P1** = X, -X, Y, -Y, Z, -Z, OFF, EXT, or INT (String; optional).

P1 specifies the desired probe compensation. If P1 is omitted, no probe compensation occurs. A specified axis indicates the part axis that most nearly points in the direction away from the measured surface. In the example, INT is the specified probe compensation for the measured line.

The possible P1 arguments are:

- X,-X,+Y,-Y,+Z,-Z for the specified axis points away from the measured surface.
- OFF for no compensation.
- EXT for the measured surface is facing away from the part datum.
- INT for the measured surface is facing towards the part datum.

Data recording is initiated by pressing the Record button on the RCU (or footswitch) and is terminated by a second press of the Record button (or footswitch). The hard probe must be in contact with the part surface at all times during the sweep operation.

Results

During operations, the message "#m, SWEEP LINE, TIP#" is displayed where:

- m equals the new feature number and
- # equals the probe tip identification.

The operator presses the Record button on the RCU (or footswitch) to start data collection and then presses it again at the end of the sweep to stop the collection. After the points are recorded the system computes the data for the line. If the recorded points do not define a line then the operator is prompted to remeasure the feature.

The contents of the Feature Register move into F1 of the Feature Stack. The Feature Register stores the data from the line. The computed line position is modified according to any requested probe compensation. Form (planar straightness) is computed as specified by the **FC** command (BFD: the sum of the distances from the computed line to the points furthest away on opposite sides of the line; MRS: the minimal separation between two parallel lines in the working plane which contains the set of points). The form value is stored in the form deviation of the Dimensional Array (s(13,4) variable). The Dimensional Array also stores the direction cosines of the computed line in s(14,0), s(15,0) and s(16,0).

The Dimensional Array also stores:

- Coordinates of the centroid,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-080 The system does not have the required probe (hard probe required).

W3 - Sweep 3D Line

The W3 command records points as an operator sweeps a hard probe along a surface and then computes a best-fit line in space through the points recorded during the sweep operation.

Command Format

MLString "W3"

Examples

1. MLString "W3"

Input

Data recording is initiated by pressing the Record button on the RCU (or footswitch) and is terminated by a second press of the Record button (or footswitch). The hard probe must be in contact with the part surface at all times during the sweep operation.

Results

During operations, the message "#m, SWEEP LINE, TIP#" is displayed, where:

- m equals the new feature number and
- # equals the probe tip identification.

The operator presses the Record button on the RCU (or footswitch) to start data collection and then presses it again at the end of the sweep to stop the collection. After the points are recorded the system computes the data for the line. If the recorded points do not define a line then the operator is prompted to remeasure the feature.

The contents of the Feature Register move into F1 of the Feature Stack. The Feature Register stores the data from the line. The computed line position is modified according to any requested probe compensation.. Form is computed as specified by the **FC** command (BFD: twice the maximum distance between the best-fit line and the individual points; MRS: the diameter of the smallest cylinder which contains the set of points). The form value is stored in the form deviation of the Dimensional Array (s(13,4) variable). The Dimensional Array also stores the direction cosines of the computed line in s(14,0), s(15,0) and s(16,0).

The Dimensional Array also stores:

- Position of the line,
- Feature number.
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-080 The system does not have the required probe (hard probe required).

WC - Sweep Circle

The WC command records points as an operator sweeps a hard probe along a surface and then computes a best-fit circle in the working plane through the points recorded during the sweep operation. The **CM** command is used to specify compensation for the probe radius.

Command Format

MLString "WC"

Examples

1. MLString "WC"

Input

Data recording is initiated by pressing the Record button on the RCU (or footswitch) and is terminated by a second press of the Record button (or footswitch). The hard probe must be in contact with the part surface at all times during the sweep operation.

Results

During operations, the message "#m, SWEEP CIRCLE, TIP#" is displayed, where:

- m equals the new feature number and
- # equals the probe tip identification.

The operator presses the Record button on the RCU (or footswitch) to start data collection and then presses it again at the end of the sweep to stop the collection. After the points are recorded the system computes the data for the circle. If the recorded points do not define a circle in the active working plane then the operator is prompted to remeasure the feature.

The contents of the Feature Register move into F1 of the Feature Stack. The Feature Register stores the data from the circle. The computed circle size is modified according to any requested probe compensation (see **CM** command). Form is computed as specified by the **FC** command (BFD: the range of the distances from the computed feature center to the individual points; MRS: the separation between the two concentric circles of minimal separation which contain the points). The form value is stored in the form deviation of the Dimensional Array (s(13,4) variable). The Dimensional Array also stores the direction cosines of the active working plane in s(14,0), s(15,0) and s(16,0).

The Dimensional Array also stores:

- X, Y, and Z center coordinates,
- Size (radius and diameter),
- Feature type, and
- Feature number

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-080 The system does not have the required probe (hard probe required).

WI - Sweep Cylinder

The WI command records points as an operator sweeps a hard probe along a surface and then computes a best-fit cylinder through the points recorded during the sweep operation. The **CM** command is used to specify compensation for the probe radius.

Command Format

MLString "WI P1"

Examples

1. MLString "WI"
2. MLString "WI AXIS"

Input

- **P1** = AXIS (String; optional)

P1 can only be the word AXIS. Including P1 means that the cylinder axis is approximately parallel to the line defined by the first and last points measured. If P1 is not specified, the cylinder must be approximately perpendicular to the working plane. Because no P1 argument appears in the first example, the computations assume the cylinder is approximately perpendicular to the working plane. In the second example, AXIS means that the cylinder is parallel to the line defined by the first and last measured points.

Data recording is initiated by pressing the Record button on the RCU (or footswitch) and is terminated by a second press of the Record button (or footswitch). The hard probe must be in contact with the part surface at all times during the sweep operation.

Results

During operations, the message "#m, SWEEP CYLINDER, TIP#" is displayed, where:

- m equals the new feature number and
- # equals the probe tip identification.

The operator presses the Record button on the RCU (or footswitch) to start data collection and then presses it again at the end of the sweep to stop the collection. After the points are recorded the system computes the data for the cylinder. If the recorded points do not define a cylinder then the operator is prompted to remeasure the feature.

The contents of the Feature Register move into F1 of the Feature Stack. The Feature Register stores the data from the cylinder. The computed cylinder size is modified according to any requested probe compensation (see **CM** command). Form is computed as specified by the **FC** command (BFD: the range of the distances from the computed axis to the individual points; MRS: the separation between the two coaxial cylinders of minimal separation which contain the points). The form value is stored in the form deviation of the Dimensional Array (s(13,4) variable). The Dimensional Array also stores the direction cosines of the computed cylinder axis in s(14,0), s(15,0) and s(16,0).

The Dimensional Array also stores:

- X, Y, and Z coordinates of the centroid of the points as projected on the computed cylinder axis,
- Radius and diameter,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-080 The system does not have the required probe (hard probe required).

WN - Define Minimum Sweep Point Interval

The WN command defines the minimum spacing for points collected by the hard-probe sweep commands to prevent multiple points being collected while the probe is stationary. The default value is 0.5 mm (0.0197 inches) and the system resets to this value at power-up and at part program initialization (see the **IL** command). Successive points collected during a sweep will be separated by at least this distance.

The WN command is only available on the SMP-400.

Command Format

MLString "WN P1"

Example

1. MLString "WN 0.75"
2. MLString "WN"

Input

- **P1** = Minimum spacing (Numeric; optional).

P1 specifies the minimum spacing between successive points for data collected by a hard probe sweep operation. If P1 is less than or equal to zero, then points are to be collected as rapidly as possible. In the example, the minimum spacing is set 0.75 units.

If P1 is not specified, the current setting for sweep point spacing is returned to S7.

Results

If P1 is specified then the minimum sweep point spacing is set to this value. If P1 is not specified then the current value is returned to variable S7.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

WP - Sweep Plane

The WP command records points as an operator sweeps a hard probe along a surface and then computes a best-fit plane through the points recorded during the sweep operation.

Command Format

MLString "WP P1"

Examples

1. MLString "WP INT"

Input

- **P1** = X, -X, Y, -Y, Z, -Z, OFF, EXT, or INT (String; optional).

P1 specifies the desired probe compensation. If P1 is omitted, no probe compensation occurs. A specified axis indicates the part axis that most nearly points in the direction away from the measured surface. In the example, INT is the specified probe compensation for the measured plane.

The possible P1 arguments are:

- X,-X,+Y,-Y,+Z,-Z The specified axis points away from the measured surface.
- OFF No compensation.
- EXT The measured surface is facing away from the part datum.
- INT The measured surface is facing towards the part datum.

Data recording is initiated by pressing the Record button on the RCU (or footswitch) and is terminated by a second press of the Record button (or footswitch). The hard probe must be in contact with the part surface at all times during the sweep operation.

Results

During operations, the message "#m, SWEEP PLANE, TIP#" is displayed, where:

- m equals the new feature number and
- # equals the probe tip identification.

The operator presses the Record button on the RCU (or footswitch) to start data collection and then presses it again at the end of the sweep to stop the collection. After the points are recorded the system computes the data for the plane. If the recorded points do not define a plane then the operator is prompted to remeasure the feature.

The contents of the Feature Register move into F1 of the Feature Stack. The Feature Register stores the data for the plane. The computed plane position is modified according to any requested probe compensation. Form (flatness) is computed as specified by the **FC** command (BFD: the sum of the distances from the computed line to the points furthest away on opposite sides of the computed plane; MRS: the minimal separation between two parallel planes which contains the set of points). The form value is stored in the form deviation of the Dimensional Array (s(13,4) variable). The Dimensional Array also stores the direction cosines of the normal to the computed plane in s(14,0), s(15,0) and s(16,0).

The Dimensional Array also stores:

- Position of the plane,
- Feature type, and
- Feature number.

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-080 The system does not have the required probe (hard probe required).

WS - Sweep Sphere

The WS command records points as an operator sweeps a hard probe along a surface and then computes a best-fit sphere through the points recorded during the sweep operation. The **CM** command is used to specify compensation for the probe radius.

Command Format

MLString "WS"

Examples

1. MLString "WS"

Input

Data recording is initiated by pressing the Record button on the RCU (or footswitch) and is terminated by a second press of the Record button (or footswitch). The hard probe must be in contact with the part surface at all times during the sweep operation.

Results

During operations, the message "#m, SWEEP SPHERE, TIP#" is displayed, where:

- m equals the new feature number and
- # equals the probe tip identification.

The operator presses the Record button on the RCU (or footswitch) to start data collection and then presses it again at the end of the sweep to stop the collection. After the points are recorded the system computes the data for the sphere. If the recorded points do not define a sphere then the operator is prompted to remeasure the feature.

The contents of the Feature Register move into F1 of the Feature Stack. The Feature Register stores the data from the sphere. The computed sphere size is modified according to any requested probe compensation (see **CM** command). Form is computed as specified by the **FC** command (BFD: the range of the distances from the computed feature center to the individual points; MRS: the separation between the two concentric spheres of minimal separation which contain the points). The form value is stored in the form deviation of the Dimensional Array (s(13,4) variable). The Dimensional Array also sets the direction cosines to zero in s(14,0), s(15,0) and s(16,0).

The Dimensional Array also stores:

- X Y, and Z center coordinates,
- Size (radius and diameter),
- Feature type, and
- Feature number

Skip Action

Not executed if part of a skipped block but a null feature is pushed onto the Feature Register.

Error Messages

- MLB-080 The system does not have the required probe (hard probe required).

21. Scan Measurement Group

MLB commands in the Scan Group enable the part program to scan a surface to collect many data points.

The Scan Group includes the following MLB commands.

AD	**Disable Automatic Tracking**
AK	**Automatic Tracking** lets you use tracking for part program-controlled operations.
AS	**Automatic Scanning** scans a surface from one point to the next with DCC control of probe movement.
BF	**Soft Probe Off** deactivates the soft probe.
BR	**Soft Probe On** activates the soft probe.
CD	**Copy Scan Data to Best-fit Array**
CQ	**Circular Move**
CU	**Clear Scan Buffer**
CW	**Circular Wait Move**
DX	**Define Soft Probe**
F2	**Disable Independent Axis Data**
F3	**Third Axis Scan Off** deactivates data accumulation for the third axis.
KA	**Tracking Active Status**
KD	**Set Nominal Soft Probe Deflection** for open loop scan.
KS	**Set Tracking Speed**
KT	**Set Tracking Move Target Zone**
MK	**Manual Tracking** controls probe movement with the RCU.
N2	**Enable Independent Axis Data**
N3	**Third Axis Scan On** activates data accumulation for the third axis.
NW	**General Surface Sweep** collects point data while sweeping a hard probe over a surface.
QS	**Query Scan Buffer**
RN	**Return Scan Data**
S2	**Independent Axis Scan Buffer Control**
S3	**Buffer Control 3** lets you designate a storage location for third axis data.

S6	**DCC Calibration of SP600**
SB	**Scan Target** transmits the targets in the scan buffer to the MP.
SN	**Scan** scans a surface from one point to the next.
TM	**Tracking Move**
ZP	**Set Soft Probe Zero Position**

AD - Disable Automatic Tracking

The AD command directs the system to disable automatic tracking, optionally back away from the surface, and enable touch mode.

Command Format

MLString "AD *p1*"

Examples

1. MLString "AD"
   Disable surface tracking, disable the soft probe, back away from the surface along the current surface normal, and re-enable touch probe mode.
2. MLString "AD 0"
   Disable surface tracking, disable the soft probe, and re-enable touch probe mode. The probe is not removed from the surface and is consequently not armed.

Input

- *p1* = '0' to withdraw probe from surface contact (Numeric; optional)

The optional numeric parameter *p1* can only be 0. If specified this indicates that the probe is not to be withdrawn from surface contact.

Results

This command is added to the DCC command queue. When executed, the system disables the soft probe and terminates surface tracking. If *p1* is not specified and the probe is in surface contact then the probe is backed off from the surface along the current surface normal using the system backoff distance (see BO command). The touch probe mode is re-enabled. Note that if *p1* is specified and the probe is in surface contact that the probe is not removed from the surface and is consequently not armed until another command or operator action removes the probe from the surface.

Skip Action

Ignored.

Error Messages

- MLB-090 DCC is not available.

AK - Automatic Tracking

The AK command directs the system to enable automatic tracking.

Command Format

MLString "AK *p1*"

Example

1. MLString "AK Y"

   Automatically track a surface in the active part reference frame.

2. MLString "AK -Z"

   Automatically track a surface in the active part reference frame.

3. MLString "AK"

   Disable

4. MLString "AK 1,1,0"

   Automatically track a surface. The probe is to move initially in part coordinate direction (1,1,0) until the surface is contacted.

Input

- *p1* = X, Y, or Z (String; optional).

The optional parameter *p1* specifies as either a signed part axis designator (e.g., +X or -Z) or as a numeric XYZ part coordinate direction vector. If a numeric vector is specified then it must be non-zero. It may be a unit vector but this is not required. If *p1* is not specified then tracking is disabled.

In the first example, Y is the *p1* argument. It indicates to automatically track a surface in the machine Y-axis. The surface lies in the positive Y direction from the probe center.

In the second example, -Z is the *p1* argument. It indicates to automatically track a surface nominally perpendicular to the machine Z-axis. The surface lies in the negative Z direction from the probe center.

In the third example there is no argument. It indicates to disable automatic mode for soft probe tracking.

Results

This command is added to the DCC command queue. When executed, the system begins tracking. If the soft probe is not in surface contact when this command is executed then the probe is moved in the specified direction until in surface contact. The touch mode of the soft probe is disabled if active (see TF command) and the soft probe is activated (see BR command). The soft probe must have been calibrated prior to execution of this command.

Note that machine velocity is restricted to tracking speed (see KS command).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-090 DCC is not available.

AS - Automatic Scanning

The AS command sets up controls for an automatic scanning / data collection operation. Be aware that it has been redefined for SMP-400, version 104.0 and later.

Command Format

MLString "AS *p1,p2,p3,p4,p5,p6*"

Example

1. MLString "AS Y,S55.6,I10"

   Perform an automatic scan, collecting samples as a command of Y position beginning at Y equal to 55.6 and collecting a data point every 10 units. Data points are accumulated only in the scan data array.

2. MLString "AS A,C12.3,-5.75,S0,I1,USER"

   Perform an automatic scan, collecting samples as a command of polar angle relative to the working plane point (12.3, -5.75), beginning at zero degrees and collecting a data point every degree. Data points are accumulated in the scan data array and in the User Summation Array. Data returned to the host with the RN command is in Radius / Angle / Height format.

Input

- *p1* = Independent axis for scan (String; required).
- *p2* = Center point for polar scan (String / numeric; optional).
- *p3* = Starting value for independent axis (String; optional).
- *p4* = Interval between successive targets (String / numeric; optional).
- *p5* = ‘USER’ or ‘INTERNAL’ to select summation array (String; optional)
- *p6* = Number of points to collect (String / numeric; optional).

The required parameter *p1* specifies the independent "axis" for the scan. This can only be A, P, S, X, Y, or Z. A specifies a polar scan with data returned in R, A, H (radius, angle, height) format. P specifies a polar scan with data returned in normal X, Y, Z format. S indicates that targets are to be spaced along the surface with the initial probe location to be treated as 0.0. X, Y, and Z indicate a part axis to be used as the independent axis.

The optional parameter *p2* is the letter 'C' followed by two coordinate values. This parameter has meaning only for polar scans and defines the working plane nominal center of the scanned surface. The values specified in order correspond to X and Y for the XY plane; Y and Z for the YZ plane; and Z and X for the ZX plane. For polar scans when *p2* is not specified the center point is assumed to be at (0, 0).

The optional parameter *p3* is the letter 'S' followed by the independent axis starting coordinate. If *p4* (below) is specified without p3 then data is collected beginning at the current probe position. Note that *p3* cannot be specified without *p4*.

The optional parameter *p4* is the letter 'I' followed by the incremental displacement between targets. If *p3* and *p4* are both omitted then it is assumed that the scan targets are absolute rather than incremental and have been previously communicated using the SB command.

The optional string parameter *p5* can be "USER" or "INTERNAL". If *p5* is specified then the scan data is accumulated in the specified Summation Array as well as in the scan data buffer.

The optional parameter *p6* is the letter 'N' followed by the number of points to be collected. If not specified, data collection continues until an RN command is executed.

Results

The AS command puts the SMP into a data collection mode. Before this command is executed tracking should be enabled (see AK command) and the probe should be positioned ahead of the first target if doing closed loop scanning or the probe should be positioned to its nominal deflection if doing open loop scanning. Following the execution of this command, appropriate move commands are executed to move the probe through the target area. Finally, the command RN is executed to terminate data collection and/or to return the collected scan data to the host.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-080 The soft probe is not enabled or tracking is not turned on.
- MLB-090 DCC is not available.

BF - Turn Soft Probe Off

The BF command disables the soft probe. If a three-axis analog probe (e.g., the Renishaw SP-600) is in use, then the touch probe is enabled.

Command Format

MLString "BF"

Example

MLString "BF"

Input

None.

Results

The system disables the soft probe and stops including soft probe deflections as part of the tip position coordinate data.

Skip Action

Ignored.

Error Messages

None.

BR - Turn Soft Probe On

The BR command enables the soft probe. The touch probe is disabled when this command is executed.

Command Format

MLString "BR"

Example

MLString "BR"

Input

None.

Results

The system enables the soft probe and starts including soft probe deflections as part of the tip position coordinate data.

Skip Action

Executes even when inside a skip block.

Error Messages

- MLB-080 The soft probe has not been calibrated.

CD - Copy Scan Data to Best-fit Array

The CD command copies the set of points collected during a scan operation to one of the best-fit summation arrays.

Command Format

MLString "CD *p1*"

Examples

1. MLString "CD"

   Add the scan data points to the user summation array.

2. MLString "CD INTERNAL"

Add the scan data points to the internal summation array.

Input

- *p1* = ‘INTERNAL’ (String; optional)

The optional string parameter *p1* can only be "INTERNAL" which specifies that the data is to be added to the internal summation array. If not specified then the data is to be added to the user summation array.

Results

The point data collected during the latest scan operation is summed into the specified summation array. Note that this command does not clear the summation array before adding the new points.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-043 A segmented measurement is in progress.

CQ - Circular Move

The CQ command directs the system to reposition the active probe using a circular move profile. It should be noted that prior to executing this command, the machine axis must be positioned on the starting point of the specified circular motion within the current move tolerance.

There are two formats for this command. Format #1 is based on the specified plane and format #2 is based on specified vectors.

Command Format #1 (plane)

MLString "CQ *p1,p2,p3,p4,p5,p6*"

Example

MLString "CQ XY,0,0,25,0,90"

Perform a circular move in the XY plane with the center at X = 0, Y = 0 with a radius of 25 units, starting along the X axis and moving counter-clockwise 90 degrees.

Input

p1 = Working plane in which motion is to occur, must be "XY", "YZ" or "ZX".. (String; required)

p2 = Center point of circular motion for first axis. (Numeric; required)

p3 = Center point of circular motion for second axis. (Numeric; required)

p4 = Radius of circular motion. (Numeric; required)

p5 = Starting angle in degrees of circular motion. (Numeric; required)

p6 = Ending angle in degrees of circular motion. (Numeric; required)

The required string parameter *p1* specifies the current working plane in which motion is to occur. This can only be "XY", "YZ" or "ZX".

The required parameter *p2* specifies the center point of the desired circular motion in current units in the current reference frame for the first named axis specified in parameter *p1*.

The required parameter *p3* specifies the center point of the desired circular motion in current units in the current reference frame for the second named axis specified in parameter *p1*.

The required parameter *p4* specifies the radius of the desired circular motion in current units.

The required parameter *p5* specifies the starting angle specified in degrees of the desired circular motion. This value may be positive or negative. Zero degrees is defined as the angle specified as the positive direction along the first named axis in parameter *p1*.

The required parameter *p6* specifies the incremental angle specified in degrees of the desired circular motion. This value may be positive or negative. A positive value specifies counter-clockwise motion in the working plane specified in parameter *p1*. A negative value specifies clockwise motion in the working plane specified in parameter *p1*. If desired, multiple revolutions may be specified with a value greater than 360 degrees.

Command Format #2 (vector)

MLString "CQ *p1*,*p2*,*p3*,*p4*,*p5*,*p6*,*p7*"

Example

MLString "CQ S,0,0,1,0.7071,-0.7071,0,-12,22,-8,25,45,90"

Perform a circular move normal to the vector (0,0,1) with the center at (-12,22,-8) with a radius of 25 units, starting at 45° from the starting vector (0.7071,-0.7071,0) and moving counter-clockwise 90 degrees.

Input

p1 = Format #2, must be “S”. (String; required)

p2 = I, J and K of the vector normal to motion. (Numeric; required)

p3 = I, J and K of the vector defining starting motion 0° position. (Numeric; required)

p4 = Center point (X, Y and Z) of circular motion. (Numeric; required)

p5 = Radius of circular motion. (Numeric; required)

p6 = Starting angle in degrees of circular motion. (Numeric; required)

p7 = Ending angle in degrees of circular motion. (Numeric; required)

The required string parameter *p1* must be "S".

The required parameter *p2* specifies I, J and K of the vector normal to the desired motion.

The required parameter *p3* specifies I, J and K of the vector defining the desired starting motion 0° position.

The required parameter *p4* specifies X, Y and Z center point of the desired circular motion.

The required parameter *p5* specifies the radius of the desired circular motion in current units.

The required parameter *p6* specifies the starting angle specified in degrees of the desired circular motion. This value may be positive or negative. Zero degrees is defined as the angle specified as the positive direction along the vector specified by parameter *p3*.

The required parameter *p7* specifies the incremental angle specified in degrees of the desired circular motion. This value may be positive or negative. A positive value specifies counter-clockwise motion and a negative value specifies clockwise motion. Multiple revolutions may be specified with a value greater than 360 degrees.

Results

If the system is not in AUTO then the operator is prompted for AUTO. The new move command is added to the list (queue) of move commands to be executed when AUTO mode is entered.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

MLB-090 DCC is not available.

MLB-116 Not on circle at beginning of circular move.

CU - Clear Scan Buffer

The CU command directs the system to clear the scan buffer prior to starting collection of a new set of scan data.

Command Format

MLString "CU"

Examples

1. MLString "CU"

Input

None.

Results

Any points currently in the scan buffer are discarded. Note that the scan buffer is also cleared by execution of the IL or the IS commands.

Skip Action

Ignored.

Error Messages

None.

CW - Circular Wait Move

The CW command directs the system to reposition the active probe using a circular move profile. Unlike the CQ command, part program execution does not continue until this move is complete. It should be noted that prior to executing this command, the machine axis must be positioned on the starting point of the specified circular motion within the current move tolerance.

There are two formats for this command. Format #1 is based on the specified plane and format #2 is based on specified vectors.

Command Format #1 (plane)

MLString "CW *p1,p2,p3,p4,p5,p6*"

Example

MLString "CW XY,0,0,25,0,90"

Perform a circular move in the XY plane with the center at X = 0, Y = 0 with a radius of 25 units, starting along the X axis and moving counter-clockwise 90 degrees.

Input

p1 = Working plane in which motion is to occur, must be "XY", "YZ" or "ZX".. (String; required)

p2 = Center point of circular motion for first axis. (Numeric; required)

p3 = Center point of circular motion for second axis. (Numeric; required)

p4 = Radius of circular motion. (Numeric; required)

p5 = Starting angle in degrees of circular motion. (Numeric; required)

p6 = Ending angle in degrees of circular motion. (Numeric; required)

The required string parameter *p1* specifies the current working plane in which motion is to occur. This can only be "XY", "YZ" or "ZX".

The required parameter *p2* specifies the center point of the desired circular motion in current units in the current reference frame for the first named axis specified in parameter *p1*.

The required parameter *p3* specifies the center point of the desired circular motion in current units in the current reference frame for the second named axis specified in parameter *p1*.

The required parameter *p4* specifies the radius of the desired circular motion in current units.

The required parameter *p5* specifies the starting angle specified in degrees of the desired circular motion. This value may be positive or negative. Zero degrees is defined as the angle specified as the positive direction along the first named axis in parameter *p1*.

The required parameter *p6* specifies the incremental angle specified in degrees of the desired circular motion. This value may be positive or negative. A positive value specifies counter-clockwise motion in the working plane specified in parameter *p1*. A negative value specifies clockwise motion in the working plane specified in parameter *p1*. If desired, multiple revolutions may be specified with a value greater than 360 degrees.

Command Format #2 (vector)

MLString "CW *p1,p2,p3,p4,p5,p6,p7*"

Example

MLString "CW S,0,0,1,0.7071,-0.7071,0,-12,22,-8,25,45,90"

Perform a circular move normal to the vector (0,0,1) with the center at (-12,22,-8) with a radius of 25 units, starting at 45° from the starting vector (0.7071,-0.7071,0) and moving counter-clockwise 90 degrees.

Input

p1 = Format #2, must be “S”. (String; required)

p2 = I, J and K of the vector normal to motion. (Numeric; required)

p3 = I, J and K of the vector defining starting motion 0° position. (Numeric; required)

p4 = Center point (X, Y and Z) of circular motion. (Numeric; required)

p5 = Radius of circular motion. (Numeric; required)

p6 = Starting angle in degrees of circular motion. (Numeric; required)

p7 = Ending angle in degrees of circular motion. (Numeric; required)

The required string parameter *p1* must be "S".

The required parameter *p2* specifies I, J and K of the vector normal to the desired motion.

The required parameter *p3* specifies I, J and K of the vector defining the desired starting motion 0° position.

The required parameter *p4* specifies X, Y and Z center point of the desired circular motion.

The required parameter *p5* specifies the radius of the desired circular motion in current units.

The required parameter *p6* specifies the starting angle specified in degrees of the desired circular motion. This value may be positive or negative. Zero degrees is defined as the angle specified as the positive direction along the vector specified by parameter *p3*.

The required parameter *p7* specifies the incremental angle specified in degrees of the desired circular motion. This value may be positive or negative. A positive value specifies counter-clockwise motion and a negative value specifies clockwise motion. Multiple revolutions may be specified with a value greater than 360 degrees.

Results

If the system is not in AUTO then the operator is prompted for AUTO. The new move command is executed when AUTO mode is entered. Part program execution does not continue until this move is complete.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

MLB-090 DCC is not available.

MLB-116 Not on circle at beginning of circular move.

DX - Define Soft Probe Alignment / Scaling Matrix

The DX command is used to define or interrogate the contents of the matrix used to convert soft probe readings to a 3D distance vector in machine coordinates.

Command Format

MLString "DX *p1*,*p2*"

Examples

1. MLString "DX 1"

   Return the contents of row number one (1) of the soft probe alignment/scaling matrix.

2. MLString "DX 0,.0000625,0,0"

   Set the contents of row zero (0) of the soft probe matrix to .0000625,0,0.

Input

- *p1* = 0, 1 or 2 for matrix index (Numeric; required)
- *p2* = matrix data (Numeric; optional)

The required numeric parameter *p1* is the index of a row in the soft probe matrix and can only be 0, 1 or 2.

The optional numeric parameter *p2* is a set of three numbers defining the contents of the matrix row specified by *p1*. Data for all three matrix rows must be sent before the new data becomes active. If *p2* is not specified then the current settings for the specified row are returned.

Results

If *p2* is not specified then the three values currently in the specified matrix row are returned to the variables S7, S8, and S10.

If *p2* is specified then the new data is saved in a temporary matrix. Once a complete set of data has been sent, this temporary matrix becomes active for converting the soft probe deflection vector into data which is used while measuring with the soft probe. This matrix must be properly defined before attempting to contact a surface with the soft probe.

The matrix is used in the form **M * P = C**, where **M** is the matrix, **P** is the column vector defined by the current soft probe deflections in probe counts, and the result **C** is a column vector equivalent to **P** but in millimeters and with components aligned with the CMM machine axes. For the Renishaw SP600 with the probe axes parallel to the machine axes, the matrix is:

0.0000625 0.0 0.0
0.0 0.0000625 0.0
0.0 0.0 0.0000625

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-080 Improper probe installed. A soft probe is required.

F2 - Disable Independent Axis Data Collection

The F2 command disables the collection of independent axis data during scanning. It is for MeasureMax version 1.03 systems only.

Warning

The F2 command is not supported by current MeasureMax+ systems.

Command Format

MLString "F2"

Examples

1. MLString "F2"

Input

None.

The example disables the collection of independent axis data.

Results

The system disables the collection of independent axis data during scan operations.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

F3 - Third Axis Scan Off

The F3 command deactivates data accumulation for the third axis. It is for MeasureMax version 1.03 systems only.

> **Warning**
>
> The F3 command is not supported by current MeasureMax+ systems.

Command Format

MLString "F3"

Examples

1. MLString "F3"

Input

None.

Results

Third axis data accumulation becomes inactive for scanning operations.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

FS - Filter Scan Data

The FS command sets the filter to be used when processing scan data. The SMP-400 internally supports a simple averaging filter that can be used to reduce the noise content associated with scan data collected with an analog probe. This filter averages a defined number of points captured before and after each target point to produce each scan point. For data collected in polar mode this averaging is based upon the polar radius and height values while for all other data it is a simple XYZ point average.

Command Format

MLString "FS p1"

Examples

1. MLString "FS 3"

   Set the scan filter value to three. This means that each reported scan data point is the result of averaging seven points, three on each side of the target plus the point at the target value.

Input

- *p1* = number of points (Numeric; optional)

The optional numeric parameter *p1* specifies the number of points from before and after each collected scan point to be averaged to produce the resultant data point. This value can range from 0 to 7.

If *p1* is not specified then the current setting for the scan filter is returned. The default value after initialization (IL) is 2.

Results

If a value is specified, the scan filter is set to this value. If a value is not specified then the current value is returned to variable S7. The system uses this value when computing the data points resulting from a scan operation for reporting to the summation array (AS or CD command) or to the host (RN command). The affect of different scan filter values can be evaluated by collecting a single set of scan data and then modifying the filter to different values before reporting the data.

For filter values above 0: a coarser filter is used for points at the beginning and end of the collected scan data, increasing to the desired level based upon data availability.

Skip Action

Ignored.

Error Messages

None.

KA - Tracking Active Status

The KA command queries the current tracking status.

Command Format

MLString "KA"

Examples

1. MLString "KA"

   Determine whether or not tracking is currently active.

Input

None.

Results

Variable S7 is set to a one (1) if tracking is currently active and is set to zero (0) otherwise.

Skip Action

Executed normally.

Error Messages

None.

KD - Set Nominal Soft Probe Deflection for Open Loop Scan

The KD command sets the nominal soft probe deflection to be used during open loop scan.

Command Format

MLString "KD *p1*"

Examples

1. MLString "KD 0.05"

   Set the nominal soft probe deflection to 0.05 units.

Input

- *p1* = nominal soft probe deflection (Numeric; optional)

The optional numeric parameter *p1* specifies the nominal soft probe deflection in current units. This value must be positive.

If *p1* is not specified then the current setting for nominal soft probe deflection is returned. The default value after initialization (IL) is 0.05".

Results

If a value is specified, the nominal soft probe deflection is set to this value except as noted above. If a value is not specified then the current value is returned to variable S7.

Skip Action

Executed normally.

Error Messages

- MLB-090 DCC is not available.

KS - Set Tracking / Scanning Speed

The KS command sets the probe speed to be used during DCC moves while tracking or scanning.

Command Format

MLString "KS *p1*"

Examples

1. MLString "KS 0.5"

Set the tracking move speed to 0.5 units per second.

Input

- *p1* = tracking / scanning move speed (Numeric; optional)

The optional numeric parameter *p1* specifies the desired tracking / scanning move speed in current units per second. This value must be positive. If the specified value exceeds the machine-allowed maximum then that will be used rather than the specified value

If *p1* is not specified then the current setting for tracking / scanning move speed is returned. The default value after initialization (IL) is 0.1" (or 2.54 mm) per second.

Results

If a value is specified, the tracking / scanning move speed is set to this value except as noted above. If a value is not specified then the current value is returned to variable S7. The system uses this as the maximum velocity whenever a DCC or joystick move is commanded while tracking a surface with a soft probe or scanning with a soft probe.

Skip Action

Executed normally.

Error Messages

- MLB-090 DCC is not available.

KT - Set Tracking Move Target Zone

The KT command sets the size of the target zone that the DCC controller uses to determine when to stop an automatic move while tracking a surface.

Command Format

MLString "KT *p1*"

Examples

1. MLString "KT 2.5"

   Set the tracking move target zone size to 2.5.

Input

- *p1* = tracking move target zone size (Numeric; optional)

The optional numeric parameter *p1* specifies the tracking move target zone size in current units. This value must be positive.

If *p1* is not specified then the current setting for tracking move target zone size is returned. The default value after initialization (IL) is 5 mm (0.197 inches).

Results

If a value is specified, the tracking move target zone size is set to this value. While making an automatic move and tracking a surface, the probe moves along the surface until the commanded position comes within this distance of the specified target, then continues to move as long as the commanded position continues to get closer to the specified target. When the commanded position begins to move away from the target, it is brought to a stop. For AM and WM moves the distance to the target is evaluated in the specified direction of travel while for TM (tracking move) moves the distance to the target is evaluated as a 3D spherical distance. The move is not considered complete until the commanded position has stopped and the actual position is within the specified positioning tolerance (see TT or TU commands) of the commanded position. Note that if the tracking move target zone is too small and/or the move target is specified improperly, motion may continue indefinitely.

If a value is not specified then the current value is returned to variable S7.

Skip Action

Executed normally.

Error Messages

- MLB-090 DCC is not available.

MK - Manual Tracking

The MK command directs the system to enable/disable manual tracking.

Command Format

MLString "MK *p1*"

Examples

1. MLString "MK Y"

   Manually track a surface in the active part reference frame.

2. MLString "MK"

   Disable manual tracking.

3. MLString "MK -Z"

   Manually track a surface in the active part reference frame.

4. MLString "MK 1,1,0"

   Manually track a surface. The probe is to move initially in part coordinate direction (1,1,0) until the surface is contacted.

Input

- *p1* = X, Y, or Z (String; optional)

The optional parameter *p1* specifies as either a signed part axis designator (e.g., +X or -Z) or as a numeric XYZ part coordinate direction vector. If a numeric vector is specified then it must be non-zero. It may be a unit vector but this is not required. If *p1* is not specified then tracking is disabled.

Results

The touch mode of the soft probe is disabled if active (see TF command) and the soft probe is activated (see BR command). The system begins tracking. Tracking speed (see KS command).

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-090 DCC is not available.

N2 - Enable Independent Axis Data Collection

The N2 command enables the collection of independent axis data during scanning. It is for MeasureMax version 1.03 systems only.

Warning

The N2 command is not supported by current MeasureMax+ systems.

Command Format

MLString "N2"

Examples

1. MLString "N2"

   Enable the collection of independent axis data.

Input

None.

Results

The system enables the collection of independent axis data during scan operations.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

N3 - Third Axis Scan On

This command activates data accumulation for the third axis for MeasureMax version 1.03 systems only.

> **Warning**
>
> The N3 command is not supported by current MeasureMax+ systems.

Command Format

MLString "N3"

Examples

1. MLString "N3"

   Enable the collection of third axis data.

Input

None.

Results

Third axis data accumulation becomes active for scanning operations.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

NW - General Surface Sweep

The NW command collects point data while a hard probe is swept over a surface. Before a hard probe scan begins, the probe must be in contact with the part. The probe must remain in contact with the part until the scan is complete.

Command Format

MLString "NW"

Examples

1. MLString "NW"

Input

None.

Results

The operator is prompted to SWEEP SURFACE and the MP WAIT light illuminates. The MP begins collecting data into the User Summation Array when the operator presses the Record button on the RCU (or footswitch). During the data collection period, the MP WAIT light flashes. Data collection is terminated by pressing the Record button on the RCU (or footswitch) a second time. The point data can then be recalled from the Summation Array using the RB command.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-080 A touch probe is active instead of the required hard probe.

QS - Query Scan Buffer

The QS command requests the number of points in the SMP scan buffer.

Command Format

MLString "QS"

Examples

1. MLString "QS"

Input

None.

Results

The number of points currently stored in scan data buffer is returned to variable S7.

Skip Action

Ignored.

Error Messages

None.

RN - Return Scan Data

The RN command terminates collection of scan data and/or returns collected scan data to the host.

Command Format

MLString "RN *p1*"

Examples

1. MLString "RN"

   If scanning is active wait for the DCC queue to become empty and the final move to be in-tolerance, then terminate scanning. Return the number of data points to the host.

2. MLString "RN 0"

   If scanning is active wait for the DCC queue to become empty and the final move to be in-tolerance, then terminate scanning. Return the collected data points (beginning with point number 0) and the number of data points to the host.

Input

- *p1* = point index (Numeric; optional)

The optional numeric parameter *p1* cannot be a negative number. If specified, *p1* requests that scan data points be returned to the host beginning with point #*p1*. Note that the first point collected has index 0 (zero). If *p1* is not specified the collected data is not to be returned to the host computer.

Results

The RN command completes a scan data collection operation initiated by the AS command. If scanning is active it will be terminated when the DCC queue becomes empty and the final move to be in-tolerance. If *p1* is specified the collected data points are returned from the scan buffer to the host computer and stored in the S9 array. For all scan types except A (see AS command), the point data is stored in the S9 array in normal X, Y, Z order; that is, the X coordinate data values are stored in S9(*, 0), Y values in S9(*, 1), and Z values in S9(*, 2). For a scan of type A, polar radius values are stored in S9(*, 0), polar angle values are stored in S9(*, 1), and third axis values (height) are stored in S9(*, 2).

The number of collected data points is returned to variable S7.

Skip Action

Ignored.

Error Messages

None.

S2 - Independent Axis Scan Buffer Control

The S2 command designates which column of the S9 array is to be used to collect independent axis data during scanning. This is for MeasureMax version 1.03 systems only.

Warning
The S2 command is not supported by current MeasureMax+ systems.

Command Format

MLString S2 *p1*"

Examples

1. MLString "S2 1"

   Store independent axis data in array S9, column 1.

Input

- *p1* = 0, 1, or 2 (Numeric; required).

p1 specifies the S9 array column in which third axis data is to be stored. The value of *p1* may only be 0, 1 or 2.

Results

The system stores independent axis data during subsequent scan operations into the specified column of the S9 array.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

S3 - Scan Buffer Control 3

The S3 command designates which column of the S9 array is to be used for third axis data scan data. Third axis data is normally stored in column #1 of the S9 array. This is for MeasureMax version 1.03 systems only.

> **Warning**
>
> The S3 command is not supported by current MeasureMax+ systems.

Command Format

MLString "S3 *p1*"

Examples

1. MLString "S3 1"

Input

- *p1* = 0, 1, or 2 (Numeric; required).

p1 specifies the S9 array column in which third axis data is to be stored. The value of *p1* may only be 0, 1 or 2.

Results

Returned third axis scan data is stored in the column of the S9 array specified by *p1*.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

S6 - DCC Calibration of SP600

Calibrate alignment / scaling data for the SP600 probe.

Command Format

MLString "S6"

Examples

1. MLString "S6"

   Calibrate the SP600 alignment and scaling data by measuring up to 5 points on the calibration sphere under DCC control. Calibration is completely under DCC control beginning with a seek from the current position towards the center of the calibration sphere.

Input

None.

The SP600 probe is required to have a straight stylus mounted which is long enough to reach the center of the calibration sphere; to be oriented with the PH9/10 at (A0, B0); and positioned near the sphere with the shaft of the probe pointed towards the sphere.

Results

The system proceeds to measure points on the calibration sphere to determine the SP600 alignment and scaling relative to the CMM axes. The resultant data is stored in non-volatile memory in the SMP and is used during subsequent probe calibration operations. This operation must be performed before valid calibrations of the SP600 probes can be performed.

Skip Action

Ignored.

Error Messages

- MLB-080 occurs if a touch probe is not active.
- MLB-090 occurs if DCC operations are not available.
- MLB-091 occurs if the touch probe fails to contact the sphere.
- MLB-111 occurs if the calibration fixture has been specified to be a cone.

SB - Scan Target

The SB command transmits the independent axis targets from the scan buffer S9(*,0) to the MP, so that they can be used in the next scan. The part program must execute the SB command before it executes the **SN** command if it uses absolute targets.

Command Format

MLString "SB *p1*"

Examples

1. MLString "SB 100"

Input

- *p1* = Number of targets to transmit (Numeric; required).

p1 specifies the number of independent axis targets to transmit. *p1* must be an integer equal to or greater than 1. In the example, *p1* is 100. The probe scans 100 targets for this measurement. The first target is in the scan buffer location S9(0,0). If you are scanning in several phases, transfer all targets at once. S9 is a real array, so the number of digits that make up the target coordinate value is not restricted.

Results

Targets are sent to the MP starting at S9(0,0). If the program transmits 100 targets, the last target sent is at S9(99,0).

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

SN - Scan

The SN command is used to initiate a surface scan. This command is for MeasureMax version 1.03 systems only.

> **Warning**
>
> The SN command is not supported by current MeasureMax+ systems.

Command Format

MLString "SN *p1,p2,p3,p4,p5*"

Examples

1. MLString "SN 1,XY,400,12,.1"

2. MLString "SN XY,400"

Input

- *p1* = Column in scan buffer (Numeric; optional).
- *p2* = Dependent and independent axes (String; required).
- *p3* = Number of targets to scan (Numeric; required).
- *p4* = Starting position (Numeric; optional).
- *p5* = Interval between successive targets (Numeric; optional).

The optional numeric parameter *p1* is the index of the column in the S9 scan data array to be used for dependent axis scan results. If *p1* is not specified then scan results are stored in column zero (0).

The required string parameter *p2* defines the dependent and independent axes for the scan. The first letter identifies the dependent axis. Valid settings for p2 are XY, XZ, YX, YZ, ZX, ZY, XW, YW, ZW, and RA.

The required numeric parameter *p3* must be a value from 1 to 1000 and specifies the number of targets to be scanned.

The optional numeric parameters *p4* and *p5* define the first independent axis scan target and the spacing between targets for an incremental scan. If *p4* and *p5* are not specified then targets are absolute and must have been previously transmitted to the MP (see SB command).

The user must have the probe in position before the first scan target before this command is executed. When this command is executed the WAIT light on the MP comes on. The user then moves the probe along the scan path to collect data. After the first target is passed the WAIT light will flash until all scan targets have been collected, then go out.

The first example scans 400 targets along Y collecting X data in column 1 of the S9 array. The first target is at a Y reading of 12 and subsequent targets are spaced at a 0.1 interval in the positive direction.

The second example scans 400 targets along Y collecting X data in column 0 (default) of the S9 array. The Y-axis targets have been previously sent with the SB command.

Results

The system collects the scanned data and stores the results in the S9 array.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-080 Improper probe installed. Either a hard probe or an analog probe is required.

TM - Tracking Move

The TM command directs the system to reposition the active probe while tracking a surface.

Command Format

MLString "TM *p1*,*p2*,*p3*"

Examples

1. MLString "TM -X,XI-2.5"

   Move the probe along a surface, beginning to move in the minus X direction, moving parallel to the active working plane (must be XY or ZX) and ending when the X-axis is a negative 2.5 units from its starting position.

2. MLString "TM X,XY"

   Move along the surface, beginning to move in the plus X direction, moving parallel to the XY plane and moving until the probe returns to the starting position.

3. MLString "TM -Z,1,1,1,X10,Y10,Z10"

   Move along the surface, beginning to move in the minus Z direction, moving parallel to the plane normal to the vector (1,1,1), and continuing until the probe comes to the position defined by the surface, the point (X10,Y10,Z10) and the tracking move zone size.

Input

- *p1* = general direction of move (String; required)
- *p2* = plane (String or numeric; optional)
- *p3* = targets (String / numeric; optional)

The required parameter *p1* specifies the general direction in which the probe is to move when starting the move as a signed part axis designator, e.g. +X or -Z. If an axis is specified without a sign then positive is assumed.

The optional parameter *p2* indicates the plane in which the probe motion is to occur. This may be: (1) XY, YZ, or ZX specifying one of the major planes in the active reference frame; or (2) a non-zero XYZ direction vector indicating a direction normal to the desired motion plane. If *p2* is not specified then it defaults to the active working plane. The direction specified by parameter *p1* must not be perpendicular to the plane selected.

The optional alphanumeric parameter *p3* specifies the axis designators (X, Y, and/or Z) and targets in current part coordinates. Targets are absolute unless preceded by the letter I to indicate an incremental target. An incremental move is from the probe's current position or from the probe's position at the end of the previously commanded move if motion is currently in progress. If *p3* is not specified the move is treated as a move around a closed surface returning to the starting point. If a target is specified it is projected into the plane defined by *p2* and the initial probe position.

Note: The length of the commanded move must be greater than the active tracking move zone size (see KT command) because the system does not begin checking for the move to be complete until it is at least this distance from the starting position. This is different from an AM or WM move executed while tracking since these commands begin checking for completion immediately to allow the execution of short moves.

Results

If the system is not in AUTO then the operator is prompted for AUTO. The new move command is added to the list (queue) of move commands to be executed when AUTO mode is entered. When the command is executed, the probe moves along the surface, beginning to move in the direction specified by p1, moving parallel to the plane specified by p2, and continuing until the probe comes to the position defined by the surface, the point specified by p3, and the tracking move zone size. The probe moves along the surface until the commanded position comes within the tracking move zone size of the specified target, then continues to move as long as the commanded position continues to get closer to the specified target. When the commanded position begins to move away from the target, it is brought to a stop. For TM moves the distance to the target is evaluated as a 3D spherical distance. The move is not considered complete until the commanded position has stopped and the actual position is within the specified positioning tolerance (see TT or TU commands) of the commanded position. Note that if the tracking move target zone is too small and/or the move target is specified improperly, motion may continue indefinitely.

Skip Action

Ignored.

Error Messages

- MLB-022 May indicate a syntax error with parameter p2 or that the vector p1 is normal to the plane p2.
- MLB-080 Either a soft probe is not available or tracking is not active.
- MLB-090 DCC is not available.

ZP - Set Soft Probe Zero Position

The ZP command sets the zero (null) position for a soft probe.

Command Format

MLString "ZP"

Example

1. MLString "ZP"

   Set the soft probe zero position.

Input

None.

Results

The current soft probe position is interrogated as saved as the zero (or null) position used to detect soft probe deflections and "touches". This command must be executed only after the probe is properly mounted on the CMM and is in the desired orientation.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

None.

22. Logging Control Group

MLB commands in the Logging Control Group enable you to:

- set the logging status to on or off,
- select full, partial or log-only modes,
- turn the printer temporarily on or off.

The Logging Control Group includes the following MLB commands.

CO	**CSV File Off** disables logging to the CSV file.
C1	**CSV File On** enables logging to the CSV file.
L0	**Logging Off** prevents logging to the Inspection Log during the RP command.
L1	**Logging On** enables logging to the Inspection Log during the RP command.
LF	**Logging Full** sets the logging status to Full.
LP	**Logging Partial** sets the logging status to Partial.
ML	**Log Only** directly logs inspection data according to the selected report format. This command does not generate a printed report, regardless of printer status.
NEWCSV	**New CSV** closes and conditionally stores a CSV inspection record and then begins a new record with a new date and time stamp.
NEWLOG	**New Log** closes an open Inspection Log file, stores or deletes it (based on the input parameter) and then opens a new log file.
P0	**Printer Off** sets the printer status to Off.
P1	**Printer On** sets the printer status to On.
SS	**Assign Part Serial Number** assigns a part serial number to identify the part being inspected.
UK	**Specify User Keys** sets values to mark specific inspections in a Inspection Log for later retrieval and analysis.

C0 - CSV File Off

The C0 command disables logging to the CSV file; thereby, preventing the system from recording inspection results in a Comma Separated Value (CSV) format.

Command Format

MLString "C0"

Examples

1. MLString "C0"

Input

None.

Results

Inspection results are not recorded to a CSV file.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

None.

C1 - CSV File On

The C1 command enables logging to the CSV file; thereby, allowing the system to record inspection results to the specified file in a Comma Separated Value (CSV) format. CSV files are text files that are commonly used by third-party programs such as spreadsheets.

Command Format

MLString "C1 *p1*"

Examples

1. MLString "C1 CSVFILE"

Input

- *p1* = File name (String; required).

p1 specifies the file name. *p1* may be in the form of "drive:\path\filename". If only the file name is specified then it is created in the "Program Files\MeasureMax\Temp" directory. If the file doesn't exist, it is created when this command is executed. You should use "CSV" as the file extension.

Results

If the specified file doesn't exist then it is created. Subsequent inspection results are then appended to this file.

Skip Action

Ignored; not executed if part of a skip block.

Error Messages

- MLB-021 First parameter invalid.

L0 - Logging Off

The L0 command disables logging to the Inspection Log file. After this command is executed, inspection results are not sent to a log file when the commands RP or ML are executed. Logging status reverts to the system configuration value when the next part program is run because an IL command automatically executes at the beginning of the part program (See the L1 command).

The part program can use the L0 command as often as necessary within the part program and still keep the same log file active.

Command Format

MLString "L0"

Examples

1. MLString "L0"

Input

None.

Results

Inspection results are not recorded in an Inspection Log file.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

L1 - Logging On

When the L1 command is used, the part program sends inspection results to the Inspection Log file whenever the command RP or ML is executed. The log file is closed when the part program terminates.

An IL command changes the logging status back to its system configuration value. The part program can use the L1 and L0 commands as often as necessary within the part program and still keep the same log file active.

Command Format

MLString "L1"

Examples

1. MLString "L1"

Input

None.

Results

Inspection results are recorded in an Inspection Log file.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

LF - Full Logging

The LF command sets the logging mode to 'full'. Logging mode reverts to the system configuration value at the part program's completion. This command does not affect the logging status. In full logging mode, the data log stores all the data necessary to regenerate the printed report.

Command Format

MLString "LF"

Examples

1. MLString "LF"

Input

None.

Results

The mode for inspection logs is set to 'full'.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

LP - Partial Logging

The LP command sets the logging mode to ‘partial’. Logging mode reverts to the system configuration value at the part program's completion. This command does not affect logging status.

While the logging mode is partial, the part program records only the actual values in the inspection log.

Command Format

MLString "LP"

Examples

1. MLString "LP"

Input

None.

Results

The mode for inspection logs is set to ‘partial’.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

ML - Log Only

The ML command processes inspection results in a manner similar to the RP command. The difference is that ML records the results in the Inspection Log file, but the results are not included in the Inspection Report.

Command Format

MLSTRING("ML *p1*")

Examples

1. MLSTRING("ML 7")

Input

- *p1* = Report format: 0 to 99 (Numeric; optional).

p1 specifies a stored print format. The format defines which data in the Dimensional Array to log. The part program does not print the logged data; it only records that data in the inspection log. Print format numbers range from 0 to 99.

If *p1* is not specified then the current report format (format #0) is used by default.

Results

If the current logging status is on, the part program records the data defined by the inspection format in the Inspection Log file. The system records the data in accordance with the full or partial logging status. The report sequence number, S6A(0), increases by one, even if logging status is off.

Skip Action

Ignored; not executed if part of a skipped block. However, the sequence number is incremented.

Error Messages

None.

NEWCSV - New CSV record

The NEWCSV command conditionally writes a CSV inspection record and then generates a new record with the current date and time.

Command Format

MLString "NEWCSV *p1*"

Examples

1. MLString "NEWCSV 1"

   Write the current CSV record and create a new one.

2. MLString "NEWCSV 0"

   Delete the current CSV record and create a new one.

Input

- *p1* = 0 or 1 (Numeric; required)

If *p1* is a ‘0’, the inspection data for the current CSV record is discarded. If *p1* is a ‘1’, the inspection data for the current CSV record is written to the CSV file.

Results

The current record of CSV inspection data is appended or discarded. If *p1* is a 1, data is appended to the existing file. If *p1* is a 0, data is discarded. A new date and time stamp is obtained and any further inspection data is collected for a new record within the same file.

The most common method is to generate an Inspection Report that can be displayed or printed. It can also be stored in a text file for later recall by the user. Inspection Reports have a tabular format and are created strictly for the user. Each run of the part program will create a new Inspection Report.

The Inspection Log is a file that is used to transport inspection data to other tasks and computers. For example, a MeasureMax+ part program can create an Inspection Log file that is sent to a SHEFFNET system for automatic statistical analysis and storage into a relational database. Inspection Log files are also used by SHEFFNET to transfer data from the database to statistical tasks for analysis and reports. Although these are text files, they are in a format that is difficult for a user to interpret. Each run of the part program will create a new Inspection Log.

The CSV File enables inspection data to be stored in a Comma Separated Value (CSV) format that is popular with third-party programs such as spreadsheets. This is a text file that delimits data with commas instead of tabs and may exceed the width of an 80- column printer; therefore, it may be tedious for a user to read. The file is an accumulation of all runs of the part program. Every time the part program is run, it writes the actual data for that inspection into one record at the end of the CSV file.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

NEWLOG - New Log file

The NEWLOG command conditionally closes or deletes the current Inspection Log file and then generates a new Inspection Log file with the current date and time.

Command Format

MLString "NEWLOG *p1*"

Examples

1. MLString "NEWLOG 1"

   Save the current Inspection Log file and open a new one.

2. MLString "NEWLOG 0"

   Delete the current Inspection Log file and open a new one.

Input

- *p1* = 0 or 1 (Numeric; required)

If *p1* is a ‘0’, the inspection data for the current log file is discarded. If *p1* is a ‘1’, the inspection data for the current log file is written to the log file. In either case, a new log file is created.

Results

The currently opened log file is closed and stored or deleted. All data reported after this command goes into a new log file.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

P0 - Printer Off

The P0 command sets the printer status to ‘off’. This means no inspection results will be printed by an **RP** command. Printer status reverts to the system configuration value when an **IL** command executes. The printer status does not affect logging.

Command Format

MLString "P0"

Examples

1. MLString "P0"

Input

None.

Results

An internal system flag is set to indicate that printer status is OFF. The output of **RP** and ID commands will not appear in the inspection report until a **P1** command is executed or the system is re-initialized.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

P1 - Printer On

The P1 command sets the printer status to ON. This means that inspection results generated by RP and ID commands will be added to the inspection report. Printer status reverts to the system configuration value when an IL command executes. The printer's status does not affect logging.

Command Format

MLString "P1"

Examples

1. MLString "P1"

Input

None.

Results

An internal system flag is set to indicate that printer status is ON. The output of RP and ID commands will appear in the inspection report until a P0 command is executed.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

SS - Assign Part Serial Number

The SS command assigns a part serial number to identify the part being inspected. The part serial number is printed in the Inspection Report and included in the Inspection Log.

Command Format

MLString "SS *p1*"

Examples

1. MLString "SS 1P2R3T"

Input

- *p1* = Maximum of 20 characters (String; optional).

p1 specifies the serial number to be assigned to the part. The part serial number can be any combination of alphanumeric characters (this includes numbers, letters, and punctuation characters) up to twenty characters long. Embedded spaces are not permitted.

If *p1* is not present, the part program pauses and prompts the operator to input the part serial number.

Results

The statement "SERIAL NUMBER: nnn" is printed in the inspection report and logged as identification data if the logging status is ON. If this command is not issued prior to the first logging action, a null serial number is logged.

Skip Action

Executes even when inside a skipped block.

Error Messages

None.

UK - User Specific Keys

The UK command sets key values to mark specific inspections in a data log for later retrieval and analysis.

Command Format

MLString "UK *p1*,*p2*"

Examples

1. MLString "UK TEMPERATURE, 72_DEGREES"

Input

- *p1* = Maximum of 20 characters (String; required).
- *p2* = Maximum of 20 characters (String; required).

p1 specifies the name of the user key. The user key name is a previously defined label.

p2 specifies the value to be assigned to the key.

In the example, the user key, TEMPERATURE, is assigned the value 72_DEGREES.

A user key of "LATHE_NUMBER" can be associated with the information (or content) "123A." This key name and key content is associated during part program execution with measurements that are logged to the database. Later, any section of datalogs that have "LATHE_NUMBER"'s with the specific content of "123A" can be retrieved.

The contents of keys may be changed as often as needed within the part program, but the key names must first be created. Key contents can be up to 20 characters long.

Results

When the contents of a User Key are changed using **UK**, those contents are associated with all subsequent measurements logged by the program. The contents that are set with **UK** remain in effect until new contents are set for the key or until the program ends.

Initially, the User Keys names are set by the System Manager. A programmer can redefine the value assigned to a user key in a part program by using the **UK** command. The key names (and values, if any) appear below the inspection report header when the first **RP** or **ID** command executes. If the contents of the keys are changed using **UK** after the initial **RP** command executes, the key names and new contents are printed out before inspection data when the next **RP** command executes.

Note that once Userkeys are created, any part program that contains an **RP**, **ID**, or **ML** command first prints the Userkey name and value.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-021 First parameter invalid.
- MLB-022 Second parameter invalid.
- MLB-505 The User Key label is unknown.

23. Rotary Table Group

MLB commands in the Rotary Table Group control, calibrate and compensate for the rotary table. If the machine you are using has a rotary table, calibrate the rotary table before you perform any other measurement commands.

After you calibrate a rotary table, the system automatically compensates subsequent measurements for the rotation.

The Rotary Table Group includes the following MLB commands.

MW	**Move W Reference** moves the 0-degree angular reference of the rotary table.
PW	**Preset W-Axis** establishes the 0-degree angular reference for the rotary table.
RL	**Read Table** determines the status of table calibration.
RW	**Read W-Axis** reads the current position of the rotary table.
SE	**Set Rotary Table Center Vector**
TB	**Table** sets angular compensation for manually controlled rotary tables.
TS	**Set Table** determines the center of rotation of the rotary table.
TZ	**MEA Table Centerfind** establishes the MEA table center of rotation and Tip 0 offset.

MW - Move W Reference

The MW command moves the 0 degree angular reference of the rotary table. The rotary table must have a counter to use this command.

Command Format

MLString "MW P1"

Examples

1. MLString "MW 57.25"

Input

- **P1** = Amount in degrees (Numeric; required).

P1 is equal to the number of degrees that the 0 degree reference moves. In the example, 57.25 is the angular value by which the 0 degree reference moves.

Results

The 0 degree angular reference of the rotary table is moved.

Error Messages

- MLB-060 A rotary table is not available.
- MLB-061 A rotary table counter is not available.
- MLB-063 The rotary table is mapped (MEA'd) and the zero reference cannot be moved.
- MLB-064 The rotary table is a Hirth coupling table and the zero reference cannot be moved.

PW - Preset W Axis

The PW command establishes the 0-degree angular reference of the rotary table. It establishes the reference by setting the current angular position equal to the value specified. This command requires that the rotary table has a counter.

Command Format

MLString "PW P1"

Examples

1. MLString "PW 38.5"

Input

- **P1** = Table angle (Numeric; required).

P1 equals the position, in degrees, to set the current table position. In the example, the position is 38.5 degrees.

Results

The current angular position of the rotary table is set to the given value.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-060 A rotary table is not available.
- MLB-061 A rotary table counter is not available.
- MLB-063 The rotary table is mapped (MEA'd) and the zero reference position cannot be modified.
- MLB-064 The rotary table is a Hirth coupling table and the zero reference position cannot be modified.

RL - Read Table

The RL command determines the status of table calibration. If the table has been calibrated then the table center coordinates are retrieved.

Command Format

MLString "RL"

Examples

1. MLString "RL"

Input

None.

Results

Variable S7 returns 0 for an uncalibrated table, and 1 for a calibrated table.

If the table is calibrated then the X, Y, and Z components of the table calibration vector return to the S8, S10, and S11 system variables respectively. These coordinates are in current measurement units but are machine coordinates (measured parallel to the machine axes).

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-060 A rotary table is not available.

RW - Read W Axis

The RW command reads the W-axis position, with respect to the reference of the rotary table.

Command Format

MLString "RW"

Examples

1. MLString "RW"

Input

None.

Results

The current table position (in degrees) returns to S7.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-060 A rotary table is not available.

SE - Set Rotary Table Center Vector

The SE command sets the vector defining the relationship of the rotary table center to tip 0, i.e., the values that would define tip 0's position if it located at the table center of rotation.

Command Format

MLString "SE P1,P2,P3"

Examples

1. MLString "SE 165.34,-54.123,27.511"

Input

- **P1** = X direction (Numeric; required).
- **P2** = Y direction (Numeric; required).
- **P3** = Z direction (Numeric; required).

P1 specifies the x direction of the rotary table center to tip 0.

P2 specifies the y direction of the rotary table center to tip 0.

P3 specifies the z direction of the rotary table center to tip 0.

Results

The rotary table center is set to the specified vector. The table is defined to be calibrated. The system coordinate transformations are updated to reflect this change.

Skip Action

Executes even when inside a skipped block.

Error Messages

- MLB-060 Rotary table is not available.

TB - Table

The TB command sets the current W-axis table position to the specified angle. This command can only be used on a system with a rotary table that does not have a counter. The active reference frame is updated to reflect the change in table position.

Command Format

MLString "TB P1"

Examples

1. MLString "TB 37.500"

Input

- **P1** = Table angle (Numeric; required).

P1 specifies the angular position of the table in degrees. Before **TB** is used, include a message in the part program to prompt the operator as to where to position the table.

The example sets the angular rotary table position at 37.500 degrees.

Results

The active reference frame is updated to reflect the change in table position.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-060 A rotary table is not available.
- MLB-063 The rotary table is mapped (MEA'd) and the zero reference cannot be moved.
- MLB-064 The rotary table is a Hirth coupling table and the zero reference cannot be moved.

TS - Set Table

The TS command computes the rotary table's center of rotation based upon the operator's measurement of a calibration sphere at three table positions.

This command can be used with rotary tables that have counters or that do not have counters. If the rotary table does not have a counter then the table must be positioned at 0, 90, and 180 degrees.

Command Format

MLString "TS P1"

Examples

1. MLString "TS 7"

Input

- **P1** = Number points. (Numeric; optional)

P1 specifies the number of points to record at each calibration sphere position. If omitted or less than four, the default of four points is used. In the example, seven points are recorded at each calibration sphere position.

Results

The operator is prompted to measure the calibration sphere at each of three table positions. The center of rotation of the table is then computed and is used to rotate the machine reference frame whenever the rotary table position changes. This command clears the following system data items:

- Current part reference frame,
- Feature Register,
- Feature Stack,
- Reference frame storage, and
- Feature Storage

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-060 A rotary table is not available.

TZ - MEA Table Centerfind

The TZ command establishes the MEA table center of rotation and Tip 0 offsets by measuring the MEA fixed sphere.

Command Format

MLString "TZ P1,P2,P3,P4"

Examples

1. MLString "TZ 7,L.5,D0,0,-.25,R.04"

Input

- **P1** = Number of points (Numeric; optional).
- **P2** = "L"(length) and probe shaft length (Numeric; optional).
- **P3** = "D"(direction) and X, Y, Z (Numeric; optional).
- **P4** = "R"(radius) and probe tip radius (Numeric; optional).

The input parameters for this command are the same as those for the **AH** command. Before executing this command, specify the calibration sphere size and number of points to be measured using the **CF** command.

P1 is the number of points to be measured on the sphere. P2 is the letter "L" followed by the length of the probe shaft. P3 is the letter "D" followed by a vector (three signed numbers separated by commas). If specified, P3 calls up the automatic sphere measurement beginning with an incremental move corresponding to the specified vector. P4 is the Letter "R" followed by the approximate probe tip radius.

The example command above moves in the minus Z direction to make the first sphere contact in measuring the MEA sphere with a probe tip having a radius of approximately 0.04 inches and a shaft length of 0.5 inches. The final measurement consists of seven points.

Results

If P3 is not specified, the operator is prompted to record two points. The first point should be 150 mm (or 6 inches) from the MEA sphere. The second point should be 6 mm (or 0.25 inches) from the MEA sphere. In either case, the system then prompts for "AUTO" mode, if necessary, and proceeds with the measurement of the user-specified number of points (ten (10) unless otherwise specified). The results of the second measurement are used to compute both the Tip 0 MEA probe offsets and the table center of rotation.

Skip Action

Ignored; not executed if part of a skipped block.

Error Messages

- MLB-060 occurs if the system does not have a rotary table with valid MEA data.

24. Error Handling Group

The MLB command in the Error Handling Group enables you to select a method of handling errors that occur during a part program run.

The Error Handling Group includes the following MLB commands.

ER	**Error Handling** selects a method of handling part program errors.

ER - Error Handling

The ER command selects either the default mode or the part program mode of error handling. In the default mode, when the system detects an error, an error message is displayed and part program processing pauses until the operator responds. In the part program mode, when the system detects an error, the S0 variable is set equal to the error number and the S0D string stores the error message. This information is returned to the part program without displaying any error message for the operator. The part program must evaluate S0 and take any required action when its value is not zero. Note that if set to "NO", part program error handling will not take place at teach time; only at run time.

Command Format

MLString "ER P1"

Examples

1. MLString "ER YES"

Input

- **P1** = YES or NO (String; Optional).

P1 must be either the string NO or the string YES. If P1 is NO, the part program error handling mode is selected. If P1 is YES, the default error handling mode is selected. In the example, YES indicates that the part program selects the default error handling mode. The initialize commands, **IL** and **IS,** reset P1 to YES.

Results

The system sets the error handling mode switch according to the parameter P1.

Skip Action

Executes even when inside a skip block.

Error Messages

None.

25. Data Storage

Measurement Library uses various types of storage to process inspection data.

- Feature Register
- Feature Stack
- Feature Storage
- Summation Arrays
- Dimensional Array
- Tolerancing
- MLB Variables
- Parametric Variables
- Probe Tip Data Set

Feature Storage

As a feature is measured or constructed, it is automatically stored in the Feature Register as the current feature. When a new measurement is taken or constructed, data in the Feature Register moves to the Feature Stack. The Feature Stack can store up to eight features. Each feature in the stack moves down one position when a new feature is stored in the Feature Register; whereby, the oldest feature is moved out of the stack. The Feature Register and positions in the Feature Stack are sometimes referred to as FR and F1 through F8 respectively.

Typically, when a feature is created, it is immediately printed in the Inspection Report and there is no further need for it. However, there are also cases where one or more features will need to be recalled later in the inspection for further reference and analysis. To meet this requirement, any feature in the Feature Register may be saved in the feature storage; likewise, any feature in feature storage may be recalled to the Feature Register.

MeasureMax+ supports unlimited feature storage and allows the feature names to be assigned any combination of alphanumeric and underscore characters with the following exceptions: 1) maximum of 64 characters; 2) the first character cannot be an underscore; 3) reserved words cannot be used.

Summation Arrays

There are two summation arrays used for storing and calculating data involved in best-fit feature calculations and some construction commands: User and Internal.

User Summation Array

The User Summation Array is completely controlled by the part program. MLB commands are used to:

- Clear the array,
- Enter data into the array, and
- Retrieve data for feature construction from the array.

Internal Summation Array

The Internal Summation Array is reserved for internal use by MLB. Some of the MLB commands allow you to specify the User or Internal Summation Array; however, you must be aware of possible conflicts should you select the internal array.

Caution
Changing reference frames causes the information in the summation arrays to become invalid. The information in the summation array remains intact, but it does not relate to the new reference frame. Be sure to clear the buffer before accessing the summation array if you have changed reference frames.

Dimensional Array

The Dimensional Array stores the current feature for the Inspection Report. When a feature is measured or constructed, or a relation is calculated, the data is temporarily placed in the Dimensional Array.

The array contains six columns that are referenced by the columns in the Inspection Report. The columns are for actual, nominal, plus tolerance, minus tolerance, deviation, and out-of-tolerance data.

There are eighteen rows in the array, one for each data type or characteristic. They are polar radius, polar angle, X coordinate, Y coordinate, Z coordinate, feature diameter, feature radius, angularity, distance, true position, perpendicularity, parallelism, total runout, form, I direction cosine, J direction cosine, K direction cosine, and major radius.

This is a 2-dimensional real array (0 to 17, 0 to 5) that can be accessed by a part program as *objData.S(n1,n2)*. Data is entered into the array by MLB commands and assignment statements in the part program.

Addresses to the individual cells of the Dimensional Array are listed below.

	0 Actual	1 Nominal	2 Plus Tol	3 Minus Tol	4 Dev	5 Out-of-Tol
0 Polar Radius	S(0,0)	S(0,1)	S(0,2)	S(0,3)	S(0,4)	S(0,5)
1 Polar Angle	S(1,0)	S(1,1)	S(1,2)	S(1,3)	S(1,4)	S(1,5)
2 X	S(2,0)	S(2,1)	S(2,2)	S(2,3)	S(2,4)	S(2,5)
3 Y	S(3,0)	S(3,1)	S(3,2)	S(3,3)	S(3,4)	S(3,5)
4 Z	S(4,0)	S(4,1)	S(4,2)	S(4,3)	S(4,4)	S(4,5)
5 Diameter	S(5,0)	S(5,1)	S(5,2)	S(5,3)	S(5,4)	S(5,5)
6 Feature Radius	S(6,0)	S(6,1)	S(6,2)	S(6,3)	S(6,4)	S(6,5)
7 Angularity	S(7,0)	S(7,1)	S(7,2)	S(7,3)	S(7,4)	S(7,5)
8 Distance	S(8,0)	S(8,1)	S(8,2)	S(8,3)	S(8,4)	S(8,5)
9 True Position			S(9,2)		S(9,4)	S(9,5)
10 Perpendicularity			S(10,2)		S(10,4)	S(10,5)
11 Parallelism			S(11,2)		S(11,4)	S(11,5)
12 Total Runout			S(12,2)		S(12,4)	S(12,5)
13 Form			S(13,2)		S(13,4)	S(13,5)
14 I dir. cosine	S(14,0)	S(14,1)	S(14,2)	S(14,3)	S(14,4)	S(14,5)
15 J dir. cosine	S(15,0)	S(15,1)	S(15,2)	S(15,3)	S(15,4)	S(15,5)
16 K dir. cosine	S(16,0)	S(16,1)	S(16,2)	S(16,3)	S(16,4)	S(16,5)
17 Major Radius	S(17,0)	S(17,1)	S(17,2)	S(17,3)	S(17,4)	S(17,5)

Descriptions for the Dimensional Array:

Actuals Column — Actual is the measured dimension of a feature. It always contains related values that correspond to the current contents of the Feature Register.

Nominals Column — Nominal is the desired (blue-print) dimension of a feature. Either the operator or assignment statements in the part program specifies nominal data when a part is being inspected.

Tolerance Columns — Tolerances establish the acceptable range for deviation. Deviation is the amount that an actual dimension or relation can vary from its nominal. For bilateral tolerancing, column 2 is the plus tolerance and column 3 is the minus tolerance. For limit tolerancing, column 2 is the maximum limit and column 3 is the minimum limit.

Deviation Column — Deviation is the amount by which the actual value of a dimension or relation differs from the nominal value. Deviation can be either a positive or negative value. Limit tolerancing does not use deviation or nominal data.

Out-of-Tolerance Column — Out-of-tolerance is the amount by which the deviation exceeds the tolerance range, or the amount by which a measurement is off from the allowable deviation.

Polar Radius, Polar Angle, X, Y and Z Rows — When a point is measured, the system enters the actual coordinates of that point into the Polar Radius, Polar Angle, X, Y and Z rows. There are two ways to locate a point in three-dimensional space:

- Cartesian (rectangular) coordinates where the point is defined with X, Y and Z values.
- Polar coordinates where a point is defined with a polar radius and angle, and the tertiary axis.

Feature Diameter and Feature Radius Rows — When a circular feature, cylinder, sphere or ellipse is measured, the system stores the feature diameter and radius. The major radius is also stored when an elliptical feature is measured.

Angularity and Distance Rows — Angularity and distance values are stored only if the system calculates them when computing the geometric relationship between two or more features.

True Position Row — Calculation of true position is based upon the selection of RFS or MMC tolerancing.

Perpendicularity, Parallelism and Runout Rows — Calculation of these values is determined by the selected geometric relation. Perpendicularity and parallelism are calculated as units/unit or ANSI Y14.5 tolerancing.

Form Row — Features are constructed using a number of points. A best-fit feature is the one feature constructed through the points that most closely resembles a perfect feature. Form is a value that depicts how much a particular feature deviates from perfect. If a feature is perfect, its form equals zero. The higher the number in the form row, the more the feature deviates from perfect.

Form is evaluated for the following features:

Feature	Form Evaluation
2-D line	Planar straightness
3-D line	Spatial straightness
Circle	Roundness
Ellipse	Ellipticity
Plane	Flatness
Sphere	Sphericity
Cylinder	Cylindricity
Cone	Conicity

Measuring the minimum number of points for any feature sets the form value equal to zero. Form always equals zero for points, step cylinders and offset planes.

I, J and K Direction Cosine Rows — Direction cosines define an orientation in space. The value is the cosine of the angle enclosed between the directional vector and the related axis. The sum of the square of the I, J and K cosines equals one. Direction cosines specify orientation only, they do not provide information about the position of a vector in reference to a datum point.

Direction cosines are calculated for the following features:

Feature	Direction Cosines
Point	Always zero. No true orientation or direction.
Plane	Orientation of vector perpendicular to the plane.
Line	Orientation of computed line.
Circle	Orientation of vector perpendicular to the best-fit plane for the circle.
Sphere	Always zero. No true orientation because a sphere has many axes.
Cylinder	Orientation of computed cylinder axis.
Cone	Orientation of computed cone axis.
Ellipse	Orientation of vector perpendicular to best-fit plane for the ellipse.

Major Radius Row — An ellipse is the only feature that has two radii of different sizes. The Major Radius row contains the larger radius and the Feature Radius row contains the smaller radius.

Tolerancing Modes

Two modes of tolerancing are available for determining feature tolerances: bilateral and limit.

Bilateral Tolerancing

Bilateral tolerancing is the system default. It calculates the variation between the nominal value and the actual value in the plus or minus direction. The following definitions apply to bilateral tolerancing:

- Plus Tolerance is the amount that the actual value can exceed the nominal value. The operator or part program must specify nominal and tolerance values in the Dimensional Array.
- Minus Tolerance is the amount that the actual value can be less than the nominal value. The operator or part program must specify nominal and tolerance values in the Dimensional Array.
- Deviation is the difference between the actual dimension and the nominal dimension. Measurement commands calculate deviation values and store them in the Dimensional

Array. Deviation is positive if the actual dimension is larger than the nominal, and it is negative if the actual dimension is smaller than the nominal.

- Out-of-Tolerance is the amount an actual dimension falls outside of the acceptable tolerance range. If the deviation is at or within the tolerance range, the out-of-tolerance value equals zero.

For true position, perpendicularity, parallelism, runout and form, the deviation is always positive. The difference between the tolerance range limit and the deviation is used to calculate the out-of-tolerance value.

Limit Tolerancing

Limit tolerancing specifies maximum and minimum sizes for feature dimensions. Since the maximum and minimum limits apply to the actual value, nominal values are not applicable. This mode defines column 2 of the Dimensional Array as the maximum limit and column 3 as the minimum limit. Any amount an actual dimension falls outside of the acceptable tolerance range is stored in the out-of-tolerance column.

The following rules apply to out-of-tolerance calculations for limit tolerancing:

- If the actual value is within the tolerance range (that is, less than or equal to the upper limit, and greater than or equal to the lower limit), then the out-of-tolerance value equals zero.
- If the actual value is less than the minimum limit, then the out-of-tolerance value is negative and is equal to the difference between the actual value and the minimum limit.
- If the actual value is greater than the maximum limit, the out-of-tolerance value is positive and is equal to the difference between the actual value and the maximum limit.

For perpendicularity, parallelism, runout and form, the out-of-tolerance value is always positive. True position cannot be calculated with limit tolerancing.

True Position Tolerancing

Two forms of geometric tolerancing are available for calculating true position: Regardless of Feature Size (RFS) or Maximum Material Condition (MMC).

Out-of-tolerance values are calculated in the same manner for MMC and RFS where a positive deviation is compared with the plus tolerance and a negative deviation is compared with the minus tolerance. However, the tolerance may have a different value for MMC than it does for RFS. MMC uses a derived tolerance that can be affected by the previous measurement of a mating part. RFS uses the value for the plus tolerance (or minus tolerance) that is located in column 2 (or column 3) of the Dimensional Array.

Regardless of Feature Size

Regardless of Feature Size (RFS) is the system default for true position tolerancing. When RFS is in effect, the system does not adjust position tolerancing based on the measured size of a related part. RFS is used when the location of that centerpoint is critical.

Maximum Material Condition

Maximum Material Condition (MMC) is applied to mating parts. An example of mating parts is if you have one part with two holes, another part with two pins, and the parts must fit together. The principal of MMC allows the measured variance on one part to change the tolerance on the other.

When a hole is at its smallest, the part contains the maximum acceptable material (i.e., an ID meets MMC at its lower limit). The opposite is true of a pin. When a pin is at its largest, the part contains the maximum acceptable material (i.e., an OD meets MMC at its upper limit). Both parts are at MMC when the largest pin is in the smallest hole and the least clearance is between the parts.

When both mating parts are at MMC, it is the least favorable condition. It is unrealistic to only accept parts that are manufactured with a perfect fit. The MMC principal makes manufacturing easier because the variation that is measured on one part affects the tolerance for the mating part.

When the holes are at their largest instead of MMC, the pins that fit into these holes can now measure anywhere between their upper and lower limit. When the hole is at its largest, it is giving all of the total tolerance to the mating part. When the hole is not at MMC, a greater positional tolerance is acceptable on the pin spacing.

The previous case also allows the spacing of the pins to vary. In the case where both parts are at MMC, the holes and pins must be spaced exactly the same or they will not fit together. With the larger hole, the pins can be closer together or farther apart and still fit into the holes.

MMC controls true position tolerancing for circular features. It requires the feature nominals, tolerances and coordinates. The system alters positional and size tolerances for mating parts. The positional and/or size tolerance on the second measured part is increased by the amount the first measured feature differs from the MMC.

With MMC, the system uses true position tolerance without any adjustment in the following cases:

- The feature has no size. Examples of these features are points, lines and planes.
- The probe compensation flag is off during feature measurement or construction.
- The feature is an internal surface (e.g., ID) and its diameter is equal to or smaller than the minimum allowable diameter.
- The feature is an external surface (e.g., OD) and its diameter is equal to or larger than the maximum allowable diameter.

MLB Variables

MLB variables are referred to as system variables for the part program. They are used to transfer data to and from the Measurement Library. All MLB variables reside in the Visual Basic object “objData”; therefore, references to any of the variables must be prefixed with the object name.

S variable for Dimensional Array

S is an MLB variable for the Dimensional Array. It contains the current feature and is used to generate Inspection Reports. This is a 2-dimensional array (0 to 17, 0 to 5) that can be accessed by a part program as *objData.S(n1,n2)*. Data is entered into the array by MLB commands and assignment statements in the part program.

S0 variable for error number

S0 is an MLB variable for the MLB error number. A value of zero represents no error. This is an integer variable that can be accessed by a part program as *objData.S0*.

S0D variable for error text

S0D is an MLB variable for an MLB error message or miscellaneous string. This is a string variable that can be accessed by a part program as *objData.S0D*.

S1 variable for first axis

S1 is an MLB variable for the first-named axis of the working plane. For the XY plane, S1 points to the X row of the Dimensional Array; likewise, YZ plane is the Y row, and ZX plane is the Z row. This is an integer variable that can be accessed by a part program as *objData.S1*.

S1D variable for feature name

S1D is an MLB variable for the MLB feature name. This is a string variable that can be accessed by a part program as *objData.S1D*.

S2 variable for second axis

S2 is an MLB variable for the second-named axis of the working plane. For the XY plane, S2 points to the Y row of the Dimensional Array; likewise, YZ plane is the Z row, and ZX plane is the X row. This is an integer variable that can be accessed by a part program as *objData.S2*.

S3 variable for third axis

S3 is an MLB variable for the unnamed axis of the working plane. For the XY plane, S3 points to the Z row of the Dimensional Array; likewise, YZ plane is the X row, and ZX plane is the Y row. This is an integer variable that can be accessed by a part program as *objData.S3*.

S4 variable for feature type

S4 is an MLB variable for the feature type. This is an integer variable that can be accessed by a part program as *objData.S4*.

S5 variable for feature number

S5 is an MLB variable for the MLB feature number. This is an integer variable that can be accessed by a part program as *objData.S5*.

S6A(0) variable for report sequence number

S6A(0) is an MLB variable for the MLB report sequence number. This is an integer variable that can be accessed by a part program as *objData.S6A(0)*.

S6A(1) variable for linear resolution

S6A(1) is an MLB variable for the linear data resolution in the MLB report. This is an integer variable that can be accessed by a part program as *objData.S6A(1)*.

S6A(2) variable for angular resolution

S6A(2) is an MLB variable for the angular data resolution in the MLB report. This is an integer variable that can be accessed by a part program as *objData.S6A(2)*.

S7 variable for temporary data

S7 is an MLB variable for MLB temporary data. This is a real variable that can be accessed by a part program as *objData.S7*.

S8 variable for temporary data

S8 is an MLB variable for MLB temporary data. This is a real variable that can be accessed by a part program as *objData.S8*.

S9 variable for scan

S9 is an MLB variable for the scan data. It stores the scan targets and actuals. This is a 2-dimensional real array (0 to 3999, 0 to 2) that can be accessed by a part program as *objData.S9(n1,n2)*. Data is entered into the array by MLB commands and assignment statements in the part program.

S10 variable for temporary data

S10 is an MLB variable for MLB temporary data. This is a real variable that can be accessed by a part program as *objData.S10*.

S11 variable for temporary data

S11 is an MLB variable for MLB temporary data. This is a real variable that can be accessed by a part program as *objData.S11*.

S12 variable for temporary data

S12 is an MLB variable for MLB temporary data. This is a real variable that can be accessed by a part program as *objData.S12*.

S13 variable for temporary data

S13 is an MLB variable for MLB temporary data. This is a real variable that can be accessed by a part program as *objData.S13*.

S14 variable for temporary data

S14 is an MLB variable for MLB temporary data. This is an integer variable that can be accessed by a part program as *objData.S14*.

S15 variable for temporary data

S15 is an MLB variable for MLB temporary data. This is an integer variable that can be accessed by a part program as *objData.S15*.

S16 variable for temporary data

S16 is an MLB variable for MLB temporary data. This is an integer variable that can be accessed by a part program as *objData.S16*.

S17 variable for temporary data

S17 is an MLB variable for MLB temporary data. This is an integer variable that can be accessed by a part program as *objData.S17*.

Parametric Variables

Parametric variables provide an alternate method for referencing probe data, components of the reference frame, and data in the Feature Register and Feature Stack. They can only be used in MLB commands that allow parametric variables to be used as command parameters.

Parametric variables for probe data:

Variable	Reference
PR	Probe Radius
NPR	Negative Probe Radius

Parametric variables for components of the reference frame:

Variable	Reference
DP	Datum Point
XA	X-Axis
YA	Y-Axis
ZA	Z-Axis
XY	XY plane
YZ	YZ plane
ZX	ZX plane

Parametric variables for Feature Register:

Variable	Reference
FR	Feature Register
X and NX	X-axis and Negative X-axis
Y and NY	Y-axis and Negative Y-axis
Z and NZ	Z-axis and Negative Z-axis
A1 and NA1	first Axis and Negative first Axis
A2 and NA2	second Axis and Negative second Axis
A3 and NA3	third Axis and Negative third Axis

Parametric variables for Feature Stack.

Note that the following table only shows the parametric variables for positions 1 and 8 of the stack. You can access any of the eight positions be substituting the bolded number with the appropriate position number.

Variable	Reference
F**1** through F**8**	Feature Stack **positions 1 through 8**
X**1** and NX**1**	X-axis and Negative X-axis of stack **position 1**
Y**1** and NY**1**	Y-axis and Negative Y-axis of stack **position 1**
Z**1** and NZ**1**	Z-axis and Negative Z-axis of stack **position 1**
A1**1** and NA1**1**	first Axis and Negative first Axis of stack **position 1**
A2**1** and NA2**1**	second Axis and Negative second Axis of stack **position 1**
A3**1** and NA3**1**	third Axis and Negative third Axis of stack **position 1**
X**8** and NX**8**	X-axis and Negative X-axis of stack **position 8**
Y**8** and NY**8**	Y-axis and Negative Y-axis of stack **position 8**
Z**8** and NZ**8**	Z-axis and Negative Z-axis of stack **position 8**
A1**8** and NA1**8**	first Axis and Negative first Axis of stack **position 8**
A2**8** and NA2**8**	second Axis and Negative second Axis of stack **position 8**
A3**8** and NA3**8**	third Axis and Negative third Axis of stack **position 8**

26. MLB Quick Reference

This is an alphabetical listing of all of the MLB commands and a brief description.

Command	Description
A1	**ANSI Angularity** computes the ANSI angularity between a best-fit feature and a specified datum axis or plane.
A2	**ANSI Parallelism** computes the ANSI parallelism between a best-fit feature and a specified datum axis or plane.
A3	**ANSI Perpendicularity** computes the ANSI perpendicularity between a best-fit feature and a specified datum axis or plane.
A4	**ANSI Runout** computes the ANSI total runout between a best-fit feature and a specified datum axis.
A5	**ANSI / ISO / DIN Positional Deviation** uses the best-fit points to compute positional deviation of a point, circle, ellipse, sphere, line, cylinder and cone.
A6	**ANSI / ISO / DIN Concentricity / Coaxiality** uses best-fit points to computer concentricity or coaxiality deviation of a point, circle, ellipse, sphere, cylinder, or cone
A7	**ANSI / ISO / DIN Circular Runout** uses best-fit points to compute runout deviation for circles, ellipses, cylinders, planes, and cones.
A8	**ANSI / ISO / DIN Symmetry Deviation** computes symmetry deviation for the feature in the Feature Register.
AC	**DCC Centerfind** directs the CMM to automatically measure a circular feature.
AD	**Disable Automatic Tracking**
AF	**Change Feature Number** allows the feature number to change within a part program.
AH	**DCC Calibrate Tip 0** directs the automatic or semi-automatic calibration of Tip 0.
AI	**Included Angle** computes the obtuse or acute angle between two features, including cones, cylinders, slots and webs.
AK	**Automatic Tracking** lets you use tracking for part program-controlled operations.
AL	**Altitude** specifies the probe tip height from a nominal center just before a DCC Centerfind measurement.
AM	**DCC Probe Move** allows probe motion along the X, Y and Z axes.

Command	Description
AN	**Angularity** computes the angle between lines and planes.
AP	**Auto-sphere Measurement Point**
AS	**Automatic Scanning** scans a surface from one point to the next with DCC control of probe movement.
AT	**DCC Calibrate Additional Tips** directs the automatic or semi-automatic measurement of a calibration sphere with a tip other than Tip 0.
AU	**Auto Status** determines the Auto or Manual status of the system as shown by the RCU.
AW	**Auto Mode Wait** executes a programmed wait until the system is placed in the Auto mode.
B0	**Disable Beep after DCC Touch** turns off the normal MP-30 beeper response to a DCC touch probe deflection.
B1	**Enable Beep after Touch** turns on the normal MP-30 beeper response to a DCC touch probe deflection.
B2	**Best-fit 2D Line** computes the best-fit position and attitude of a line from points in the selected summation array. The line computation projects the line on a plane parallel t the work plane.
B3	**Best-fit 3D Line** computes the best-fit position and attitude of a line from points in the selected summation array. The line is computed in three-dimensional space.
BA	**DCC Bolt Circle** measures and reports a bolt circle pattern under DCC control.
BB	**Begin Block** marks the beginning of a code block.
BC	**Best-fit Circle** computes the best-fit to a circle from points in the specified summation array.
BD	**Best-fit Ellipse** computes the best-fit location, orientation, and maximum and minimum radii for an ellipse. It uses points from the specified summation array to perform the calculations.
BF	**Soft Probe Off** deactivates the soft probe.
BI	**Best-fit Cylinder** computes the best-fit to a cylinder from points in the selected summation array.
BK	**Best-fit Cone** computes the best-fit location, orientation, and included angle for a cone. It uses points from the specified summation array to perform the calculations.
BL	**Bilateral Tolerancing** specifies that the system uses bilateral tolerancing for evaluations.
BM	**Best-fit Summation** sums the position of the specified feature into the specified summation array.

Command	Description
BN	**Manual Bolt Circle** measures and reports a bolt circle pattern under manual control.
BO	**Backoff Distance** specifies the distance that the probe moves away after it contacts a surface.
BP	**Best-fit Plane** computes the best-fit to a plane from points in a specified summation array.
BQ	**Begin Skip** marks the beginning of the coded block for a skip sequence.
BR	**Soft Probe On** activates the soft probe.
BS	**Best-fit Sphere** computes the center point, diameter, and best-fit to a sphere from points in the specified summation array.
BT	**Compute Best-fit Point**
BW	**Compute Best-fit Slot / Web**
BX	**Compute Best-fit Cparln Feature** computes the best-fit Cparlnf (Closed parallel lines - flat end) or Cparlnr (Closed parallel lines - round end) in the working plane to a previously accumulated set of points.
BZ	**Beeper** controls the beepers in the RCU and MP.
C1	**CSV File On** enables logging to the CSV file.
C6	**Calibrate PH6** calibrates a PH6A or PH9A probe head adapter using a probe changer sphere.
CA	**Calibrate Using Best-fit Data** calibrates probe tips from data in a summarion array.
CB	**Clear Summation Array** clears and initializes the summation array.
CD	**Copy Scan Data to Best-fit Array**
CE	**Calibrate Extension** calibrates a probe extension.
CF	**Calibrate Fixture** defines the type, location and size of a calibration fixture.
CH	**Calibrate Tip 0** lets you calibrate Tip 0 manually.
CL	**Clearance** specifies a distance from the nominal surface or circumference of a circle that the probe tip keeps to avoid contact before touches.
CM	**Probe Compensation** specifies the probe compensation for round features.
CN	**Cone** computes the location, orientation and included angle of a cone from five points in the stack.
CO	**CSV File Off** disables logging to the CSV file.
CP	**Determine Probe Compensation** corrects the computed feature results for probe radius after the feature is measured.

Command	Description
CQ	**Circular Move**
CR	**Circle** computes a circle that is parallel to the measurement plane. The command uses three points in the stack.
CT	**Calibrate Additional Tips** calibrates multiple probe tips and identifies tip positions by measuring a calibration fixture.
CU	**Clear Scan Buffer**
CV	**Changing Sign or Orientation Vector**
CW	**Circular Wait Move**
CY	**Cylinder** computes a cylinder from five points in the stack.
D9	**Define PH9 Safe Position** establishes a position where the PH9 can safely change probe tip attitudes prior to loading or unloading a station.
DA	**Axial Distance** computes the three-dimensional distance between two points, two lines, two planes, a line and a plane, a point and a line, or a point and a plane. It also derives the axial components of the computed distance and pushes these as a 'distance' feature onto the feature stack.
DF	**Delete Feature** deletes a feature from Feature Storage and frees that storage location.
DM	**Display Mode** returns the current status of the display update to system variable S7.
DN	**Display On** turns the display update on.
DO	**Display Off** turns the display update off.
DP	**MP Display** displays alphanumeric information on the MP.
DR	**Delete Reference Frame**
DS	**Distance** computes the three-dimensional distance between two features.
DT	**Deviation/Tolerances** computes the deviation and out-of-tolerances for a particular row in the Dimensional Array. It does not produce a printed report.
DV	**Set a Dimensional Array Value**
DW	**DCC Wait** prevents execution of any further FLB commands in a part program until all previous commands are completed.
DX	**Define Soft Probe**
E0	**Lower APC Rack on ProGage CMM**
E1	**Raise APC Rack on ProGage CMM**
EL	**Ellipse** computes the location, orientation, and maximum and minimum radii of an ellipse parallel to the working plane from five points in the stack.

Command	Description
EO	**Establish Offset Angle** establishes the offset angle based on a feature.
EP	**Establish Datum Plane** establishes the primary datum plane or working plane based upon the feature in the Feature Register.
EQ	**End Skip** ends a block skip sequence.
ER	**Error Handling** selects a method of handling part program errors.
ES	**Establish Reference** establishes the XYZ datum.
EX	**Establish X Reference** establishes the X datum.
EY	**Establish Y Reference** establishes the Y datum.
EZ	**Establish Z Reference** establishes the Z datum.
F2	**Disable Independent Axis Data**
F3	**Third Axis Scan Off** deactivates data accumulation for the third axis.
FC	**Form Calculation Method**
FD	**Display Format** specifies the display format used on the MP.
FF	**Fetch Form Feature Data**
FM	**Define Format** defines and stores an Inspection Report format.
FP	**Define the Active Probe Tip Data Set**
FT	**Fetch** returns the value of a parametric variable to a system variable for examination.
G$	**Get Key** allows the user to set up RCU keys as acceptable responses to a prompt.
GN	**Get Number** asks the user to enter a number on the RCU keypad.
GS	**Get Station** loads the specified probe changer stations onto the probe adapter.
GT	**Select Tip** selects the probe tips.
HF	**Turn Standard Report Header Off**
HL	**High/Low Point Retrieval** retrieves the high and low points determined when best-fit feature form is computed.
HO	**Move To Home** provides the capability to perform a DCC move to the home position of the probe.
I0	**ANSI / ISO / DIN Form Reporting** identifies and reports the form of the feature in the Feature Register.
I1	**ANSI / ISO / DIN Angularity Reporting** computes angularity deviation as a width of a zone containing the feature in the Feature Register and prints a report showing this result.

Command	Description
I2	**ANSI / ISO / DIN Parallelism Reporting** computes parallelism deviation as a width of a zone containing the feature in the Feature Register and prints a report showing this result.
I3	**ANSI / ISO / DIN Perpendicularity Reporting** computes perpendicularity as a width of a zone containing the feature in the Feature Register and prints a report showing this result.
I4	**ANSI / ISO / DIN Total Runout Deviation Reporting** computes total runout deviation between the best-fit feature in the Feature Register and a datum axis and prints a report showing the result.
I5	**ANSI / ISO / DIN Position Deviation Reporting** computes and reports position deviation for the feature in the Feature Register.
I6	**ANSI / ISO / DIN Concentricity / Coaxiality Reporting** computes and reports concentricity or coaxiality deviation for a feature.
I7	**ANSI / ISO / DIN Circular Runout Deviation Reporting** computes and reports circular runout deviation for the feature in the Feature Register.
I8	**ANSI / ISO / DIN Symmetry Deviation Reporting** computes and reports symmetry deviation for the feature in the Feature Register.
IC	**Immediate Read of MEA Correction** returns the current position based on the MEA correction data.
ID	**Identification Data** outputs a literal string to the Inspection Report and/or Log.
II	**Read CMM Axis Positions**
IL	**Initialize Library** initializes the Measurement Library.
IM	**Read Probe Position**
IN	**Inch** sets measurement units to inches.
IO	**IOSTAT** allows you to sample the status of one of the input lines or change the status of one of the output lines.
IS	**Initialize System** initializes the complete CMM system.
IT	**Initiate Temperature Compensation** causes all measurements to be corrected to appear as though both the machine and the part were at 20 degrees Celsius (68° F).
JA	**Projected Angularity** projects two lines onto a selected plane and computes the angle between them.
JD	**Projected Distance** projects two features onto a plane and computes the distance between them.
JI	**Projected Included Angle** projects two features onto the selected plane and computes the included angle between them.

Command	Description
JL	**Projected Parallelism** projects two lines onto the selected plane and computes the deviation from parallelism between them.
JP	**Projected Perpendicularity** projects two lines onto the selected plane and computes the deviation from perpendicularity between them.
JX	**Projected Axial Distance** projects two features onto the selected plane before computing the distance between them. It also derives the axial components of the computed distance.
KA	**Tracking Active Status**
KD	**Set Nominal Soft Probe Deflection** for open loop scan.
KS	**Set Tracking Speed**
KT	**Set Tracking Move Target Zone**
L0	**Logging Off** prevents logging to the Inspection Log during the RP command.
L1	**Logging On** enables logging to the Inspection Log during the RP command.
L2	**2D Line** computes the position and attitude of a line parallel to the working plane from two points in the stack.
L3	**3D Line** computes the position and attitude of a line in three-dimensional space from two points in the stack.
LA	**DCC Locate Calibration Sphere** directs the automatic or semi-automatic measurement of a moved calibration sphere after an initial calibration at the sphere's MEA location.
LC	**Locate Calibration Sphere** locates the calibration sphere on the work table for a manual measurement after an initial calibration at the sphere's MEA location.
LF	**Logging Full** sets the logging status to Full.
LM	**Limit Tolerancing** specifies that the system uses limit tolerancing for evaluations.
LP	**Logging Partial** sets the logging status to Partial.
LV	**Level** orients the working plane.
M2	**Measure 2D Line** constructs a line by recording points and computing a best-fit line. It projects the line onto a plane parallel to the work plane.
M3	**Measure 3D Line** constructs a best-fit line defined by a set of recorded points.
M9	**Mover PH9** rotates the PH9 probe in A or B axis or both.
MA	**Auto Mode** turns the Auto mode on. This is only supported on machines with special hardware.
MB	**Measure Slot/Web** lets you measure a slot or web.

Command	Description
MC	**Measure Circle** constructs a circle by recording points and computing a best-fit to a circle.
MD	**Measure Ellipse** constructs an ellipse by recording points and computing the best-fit of an ellipse to those points.
ME	**Metric** sets measurement units to millimeters.
MF	**Measure Offset Plane** lets you measure an offset plane.
MI	**Measure Cylinder** constructs a cylinder by recording points and computing the best-fit of a cylinder to those points.
MJ	**Measure Step Cylinder** lets you measure a step cylinder.
MK	**Manual Tracking** controls probe movement with the RCU.
ML	**Log Only** directly logs inspection data according to the selected report format. This command does not generate a printed report, regardless of printer status.
MM	**Maximum Material Condition** specifies that the system uses the principles of maximum material condition when computing true position.
MN	**Manual Mode** switches the CMM into Manual mode.
MO	**Move Offset Angle** modifies the offset angle for the current reference frame.
MP	**Measure Plane** computes the best-fit of a plane to the measured points.
MQ	**Measure Cone** constructs a cone by recording points and computing the best-fit of a cone to those points.
MR	**Move Reference** moves a datum along one or more of the axes.
MS	**Measure Sphere** constructs a sphere by recording points and computing the best-fit of a sphere to those points.
MT	**Monitor MP** monitors the system variables displayed on the MP.
MW	**Move W Reference** moves the 0-degree angular reference of the rotary table.
MX	**Move X Reference** moves a datum along the X-axis.
MY	**Move Y Reference** moves a datum along the Y-axis.
MZ	**Move Z Reference** moves a datum along the Z-axis.
N2	**Enable Independent Axis Data**
N3	**Third Axis Scan On** activates data accumulation for the third axis.
NEWCSV	**New CSV** closes and conditionally stores a CSV inspection record and then begins a new record with a new date and time stamp.
NEWLOG	**New Log** closes an open Inspection Log file, stores or deletes it (based on the input parameter) and then opens a new log file.

Command	Description
NEWRPT	**New Report** is used to close and print a report file.
NO	**Cancel Level** cancels leveling and aligns the part reference frame with the machine reference frame.
NS	**Segment** sets internal flags for the measurement of segments of a feature.
NT	**Delete Tip** deletes the calibration vectors for all probe tips except Tip 0.
NW	**General Surface Sweep** collects point data while sweeping a hard probe over a surface.
OD	**Overdrive Distance** sets the distance the probe can move without making contact with the target after reaching the nominal position of the target.
OP	**Offset Plane** computes the parameters from a plane described by three points. The points can exists in the stack or one of the summation arrays.
OS	**Servo Off** turns the servo power off.
OT	**Off TempComp** suppresses the reporting of TempComp messages.
OW	**Operator Wait** prompts the user to press the footswitch or the appropriate key on the RCU.
P$	**Panel** gives the part program access the MP keypad.
P0	**Printer Off** sets the printer status to Off.
P1	**Printer On** sets the printer status to On.
PA	**Parallelism** computes the deviation from parallelism between two features.
PB	**Probe Type** identifies a probe's type (hard or touch) and status (deflected or not deflected since installation).
PD	**Position Deviation** computes the deviations and out-of-tolerances for the rectangular and polar coordinates of the part position. It does not produce a printed report.
PE	**Perpendicularity** computes the deviation from perpendicularity between two features.
PF	**Polar Coordinates** causes the MP to display polar part coordinates.
PI	**Pallet ID** reads the Pentrax pallet ID.
PK	**DCC Park** moves the probe to the end of travel of the X, Y and/or Z axes.
PM	**Prompt** displays a line of alphanumeric information on the CRT and/or RCU.
PN	**Plane** computes the centroid and orientation of a plane from three points in the stack.
PO	**Pop Stack** moves the features in the Feature Stack up one position. It puts the data from the first position of the Feature Stack into the Feature Register.

Command	Description
PR	**Probe Diameter** sets or returns a probe's diameter.
PS	**Preset Datum** establishes the XYZ datum relative to the position of the probe.
PU	**Push Stack** moves the features in the Feature stack down one position.
PW	**Preset W-Axis** establishes the 0-degree angular reference for the rotary table.
PX	**Preset X Datum** establishes a datum along the X axis relative to the position of the probe.
PY	**Preset Y Datum** establishes a datum along the Y axis relative to the position of the probe.
PZ	**Preset Z Datum** establishes a datum along the Z axis relative to the position of the probe.
QB	**Query Best-fit Summation Status**
QP	**Query CMM Probe Position**
QS	**Query Scan Buffer**
QT	**Query Temperature** returns the current temperature reading for a Temperature Compensation temperature sensor.
QU	**Measure Point** prompts the operator to record a single point on a part's surface.
QV	**Query Touch Point Surface Vector**
QZ	**Manual Single Point Measurement** lets you select prompting at the RCU or the host computer.
R0	**Remeasure Off** turns the remeasure mode off.
R1	**Remeasure On** turns the remeasure mode on for non-DCC operations.
R3	**Get Averaged Position Reading**
RA	**DCC Rectangular Pattern** measures and reports a grid of circular features under DCC control.
RB	**Recall Best-fit Point** loads the Feature Register with the specified point data from one of the summation arrays.
RC	**Reference Frame Change**
RD	**Rectangular Coordinates** causes the MP to display rectangular part coordinates.
RE	**Read Point**
RF	**Recall Feature** moves the Feature Stack down and puts the data from the specified feature storage area into the Feature Register.
RI	**Read Probe Data** tells you a probe's offset and radius.

Command	Description
RL	**Read Table** determines the status of table calibration.
RM	**Manual Rectangular Pattern** measures and reports a grid of circular features under manual control.
RN	**Return Scan Data**
RO	**Read Offset Angle** retrieves the offset angle of the current reference frame.
RP	**Report** computes deviation and out-of-tolerance values in the Dimensional Array and prints a formatted report of the requested data.
RR	**Recall Reference Frame** recalls a previously-saved reference frame.
RS	**Regardless of Feature Size**
RT	**Stored Reference Frame Status** determines the status of reference frame storage locations.
RV	**Firmware Version and MP Type**
RW	**Read W-Axis** reads the current position of the rotary table.
RX	**Reverse Axis for Mirror Imaging Part**
RZ	**Remove Station** deletes all calibrated tips associated with a specified station.
S2	**Independent Axis Scan Buffer Control**
S3	**Buffer Control 3** lets you designate a storage location for third axis data.
S6	**DCC Calibration of SP600**
SA	**Starting Angle** defines the starting angle and angular increment for a DCC measurement of a bore or boss. Used with the **AC** command.
SB	**Scan Target** transmits the targets in the scan buffer to the MP.
SC	**Step Cylinder** computes the location, orientation and radii of the step cylinder from six points. The six points can be in the stack or one of the summation arrays.
SD	**Probe Speed** specifies the maximum probe velocity.
SE	**Set Rotary Table Center Vector**
SF	**Save Feature** copies data from the Feature Register into Feature Storage.
SG	**Next Segment** processes measurements for multiple segments of a feature.
SH	**Sphere** computes a sphere from four points in the stack.
SI	**Station Information** tells you if a station is loaded or unloaded. A station is loaded if the probe or probe extension from that station is currently mounted on the CMM probe shaft. Station Information also tells you whether or not a station is calibrated.

Command	Description
SK	**DCC Seek Move** moves the probe toward the target at the slower touch speed.
SL	**Set Level** specifies autolevel direction cosines for the working plane.
SM	**Scale Machine** scales machine coordinates for the X, Y and Z axes.
SN	**Scan** scans a surface from one point to the next.
SO	**Set Offset Angle** specifies offset angle.
SP	**Scale Part** scales part coordinates for the X, Y and Z axes.
SQ	**Calibration Setup** tells the measurement processor to use the measurements recorded on the calibration sphere to calibrate a probe tip.
SR	**Save Reference Frame** saves the current reference frame.
SS	**Assign Part Serial Number** assigns a part serial number to identify the part being inspected.
ST	**Set Point / Feature**
SU	**DCC Summation Touch** moves the probe at the slower touch speed and sums the point into a summation array.
SV	**Servo Status** determines whether the servo power is on or off.
SW	**Table Speed** specifies the maximum speed for table rotation.
SX	**Save Home Position** defines a safe position for a DCC CMM to move to allow unobstructed part loading and unloading.
SZ	**Size Deviation/Tolerancing** sets the mode of the size deviation calculation. This setting affects how the deviation calculation is performed when bilateral tolerancing is active.
TA	**Touch Status** determines whether the touch mode is on or off, indicated by the PH9/PH10 red light.
TB	**Table** sets angular compensation for manually controlled rotary tables.
TC	**DCC Touch Move** moves the probe from its current position at the slower touch speed toward the specified target.
TD	**Touch Speed** specifies the maximum speed for all programmed touches.
TF	**Touch Mode Off** prevents the system from sensing a deflected probe, whether the deflection is caused by a touch or vibration.
TI	**Tip Status** tells you if a probe tip is calibrated.
TL	**Tolerance Lights** allows you to set a part status variable so the system can turn on a specified tolerance light.
TM	**Tracking Move**
TN	**Touch Mode On** allows the system to sense a deflected probe.

Command	Description
TP	**True Position** computes the deviations and out-of-tolerances for the rectangular and polar coordinates of the part position. it does not produce a printed report.
TS	**Set Table** determines the center of rotation of the rotary table.
TT	**Target Tolerances** sets the distance for the probe to begin its next DCC move before it completes its current move. Works in conjunction with high-speed commands such as **AM**.
TU	**Set Move Target Tolerance** specifies the X, Y and Z target tolerances used to begin the next high-speed DCC move, and to determine when that move is complete.
TV	**Set Touch Target Tolerance** specifies X, Y and Z target tolerances used to begin a DCC touch command, and to determine when that comand is complete if no contact occurs. Works in conjunction with **SK** and **TC** commands.
TW	**Set Minimum Time** sets a minimum time interval that is to elapse between the end of a high-speed move and a subsequent surface touch.
TZ	**MEA Table Centerfind** establishes the MEA table center of rotation and Tip 0 offset.
UA	**Label User Array Rows** allows a part program to label rows of the user array, and to define the type of processing to be performed on a row.
UK	**Specify User Keys** sets values to mark specific inspections in a Inspection Log for later retrieval and analysis.
UL	**Shuttle** allows you to start the shuttle sequence on Pentrax systems.
UM	**Unsum a Point from the Summation Array**
US	**Unstack** moves the features in the Feature Stack up one position. It does not affect the Feature Register.
VM	**Vector Move Using Surface Vector**
VP	**Rectangular Conversion** converts nominal polar radius and angle to their equivalent nominal rectangular coordinates.
VR	**Polar Conversion** converts nominal rectangular coordinates to their equivalent nominal polar radius and angle.
VS	**DCC Vector Seek** performs a seek from the current probe position toward a clearance point.
VT	**DCC Vector Touch** performs a DCC touch along a specified vector.
W2	**Sweep 2D Line** records a sweep measurement of a line as projected into the working plane. It computes the best-fit position and attitude of the line.
W3	**Sweep 3D Line** records a sweep measurement of a line. It computes the best-fit position and attitude of the line in three-dimensional space.

Command	Description
WC	**Sweep Circle** records a sweep measurement of a circle. It computes the best-fit of a circle.
WI	**Sweep Cylinder** records a sweep measurement of a cylinder. It computes the best-fit to a cylinder.
WM	**DCC Wait Move** allows probe motion along the X, Y and Z axes. It is similar to the AM command except the system completes the move before processing the next command.
WN	**Define Minimum Sweep Point**
WO	**DCC Probe/PH9 Move** provides the capability to perform a DCC move and select a new PH9 tip as the shaft moves. The probe tip can be swiveling to a new orientation as the arm moves.
WP	**Sweep Plane** records a sweep measurement of a plane. It computes the best-fit to a plane.
WS	**Sweep Sphere** records a sweep measurement of a sphere. It computes the best-fit to a sphere.
WT	**Define Thermistors for TempComp**
XC	**Exchange Feature** exchanges the features in the Feature Register with the one in the first position of the Feature Stack.
XI	**LOM Inch Increment / Decrement** updates the MP30 transformations when the layout machine (LOM) is relocated along the machine base.
XM	**LOM Metric Increment / Decrement** updates the MP30 transformations when the layout machine (LOM) is relocated along the machine base.
XY	**XY Plane** selects the XY plane as the working plane.
YZ	**YZ Plane** selects the YZ plane as the working plane.
ZM	**DCC Zero Switch Move** commands a DCC machine to move to a location defined relative to the MEA zero switches. All three axes move immediately to the zero switches.
ZP	**Set Soft Probe Zero Position**
ZX	**ZX Plane** selects the ZX plane as the working plane.
ZZ	**DCC Zero Switch Locate** causes the DCC machine to perform a series of moves to locate the MEA zero switches. The axes move one after the other to avoid workpiece contact in the measurement area.

27. Glossary

abort

Stop an in-process measurement task before it is complete.

absolute

A DCC probe move or touch using point coordinates that are a distance from zero.

absolute filepath

Complete file specification, including drive and root directory.

actual

Measured dimension of a feature.

align

To make the coordinate measuring machine axes parallel to the part you are measuring.

angularity

The measured angle in degrees between two features.

autochange rack

An optional accessory rack which holds up to eight different probe tips and extensions. You program the coordinate measuring machine to pick up the probes and extensions used in measuring a part.

autolevel

A command which mathematically rotates the coordinate measuring machine's active plane so that it is parallel with a recorded part plane or perpendicular to a center line or line.

axis, feature

The center line of a cone, cylinder, or step cylinder. The system records the midpoint of the recorded segment of this line as the feature's position. It uses the orientation of the axis as the orientation of the feature.

backoff distance

The distance that the probe moves away from a touch point after contact in order to rearm and clear the part after a DCC touch.

backup file

An extra copy of a file. A file is an organized collection of information that the computer treats as a unit. Each part program and inspection log is a file.

base point

The average position of all points recorded on a line or plane. The position of a line or plane is recorded as the coordinates of its base point.

best-fit feature

The calculated feature fitting closest to all the recorded points of the actual feature.

bilateral tolerancing

A tolerancing method in which the dimension of the feature is allowed to vary in both the positive and negative direction from its nominal.

block

A group of program lines that are marked with a beginning and an end. When you run a part program, you can tell the system to skip or execute a block.

block identifier

A 10-character name you give a block when you mark it in a part program. You use the block identifier to define which blocks to skip and which blocks to execute.

boot

To turn on a computer and load its operating system.

calibrate

To adjust or inspect a measurement instrument for accuracy. A part program may require that you calibrate a probe.

calibrate feature

A fixture used to determine relative probe tip positions during calibration. You can also specify that the system determine probe diameter during calibration.

calibration sphere

A ball-shaped fixture used to calibrate probes.

CALL statement

A BASIC statement that transfers program execution to a specified subprogram. You use the CALL statement to access commands in the Measurement Library.

CALSTA

A CMM utility program for calibrating all stations of an Autochange Rack. You must run this program if you move the rack.

Cartesian

A set of three independent, mutually-perpendicular axes (X,Y,Z). The coordinates of a point are its distances from the zero point of each of these axes as measured parallel to that axis.

Cathode Ray Tube - CRT

The part of your computer that looks like a television screen and produces a video display. Menus, prompts, and error messages can be displayed on the CRT.

clearance

The distance from the part surface at which the probe approaches a circular feature during an automatic measurement sequence.

CMM - Coordinate Measuring Machine

A machine that measures parts in three dimensions. The CMM continually tracks the position of a probe tip's position relative to the machine.

CMS - Coordinate Measuring System

A system used to inspect parts. The Sheffield CMS includes a coordinate measuring machine (CMM), a measurement processor (MP), and a system computer.

Comma Separated Value - CSV

Data is stored (logged) in a format easily used by other software. Primarily used with spreadsheet and statistical software.

conical fixture

A concave fixture used to orient hard probes. You put the tip of the currently selected probe in the cone and record one point. Unlike the calibration sphere, the conical fixture cannot be used to determine probe diameter.

construction

An operation where the system constructs a new feature based on previously measured features.

Coordinate Measuring Machine - CMM

A machine that measures parts in three dimensions. The CMM continually tracks the position of a probe tip's position relative to the machine.

Coordinate Measuring System - CMS

A system used to inspect parts. The Sheffield CMS includes a coordinate measuring machine (CMM), a measurement processor (MP), and a system computer.

coordinates

A set of points with references to each of the three axes used to define the position and dimension of a feature.

CRT - Cathode Ray Tube

The part of your computer that looks like a television screen and produces a video display. Menus, prompts, and error messages can be displayed on the CRT.

CSV - Comma Separated Value

Data is stored (logged) in a format easily used by other software. Primarily used with spreadsheet and statistical software.

cursor

An underline mark, arrow symbol, or highlighted rectangle which appears on your screen to mark the position of the next character you enter.

datum

A point, line or plane used as a reference from which the location of other features are referenced.

datum axis

A reference line from which distances or angles are measured in a coordinate measuring system. NAME=default

DCC - Direct Computer Control

A type of coordinate measuring machine (CMM) that moves without the help of an operator. The computer can control the CMM's movements.

default

The preset value assigned to a variable by the system if you do not enter a another value.

desktop system

A coordinate measuring system (CMS) consisting of a coordinate measuring machine (CMM), a measurement processor, and a computer. All the computer components can fit on a desktop.

deviation

Amount by which the actual value of a dimension differs from the nominal value.

diagnostics

Tests that tell you of any faults in the computer system.

Dimensional Array

The storage area used to create inspection reports. The columns and rows match the Inspection Report window. Data for the feature at the top of the Feature Storage window is in the Actual column. Stores actual feature measurements, nominals, tolerances, deviations, and out-of-tolerance values.

Direct Computer Control - DCC

A type of coordinate measuring machine (CMM) that moves without the help of an operator. The computer can control the CMM's movements.

direction cosine

One of the three trigonometric values recorded in the Dimensional Array which are used to record a feature's orientation in space. The feature's orientation is referenced to the measurement axes by its I, J, and K direction cosine values.

disk drive

A piece of hardware that holds the magnetic disk and allows a computer to read programs from or write programs to the disk.

DMIS

The acronym for Dimensional Measuring Interface Standard. The objective of DMIS is to provide a standard for the bi-directional communication of inspection data between computer systems and inspection equipment.

drive point

A recorded point in space which is the target for a DCC probe movement. Drive points are not used for measurement, but rather as starting points from which measurement (touch) points on the part or other drive points may be reached.

effective probe size

What is actually calculated when probe size is determined during probe calibration. Effective size takes into account actual probe diameter and the effects of the spring pressure needed to deflect touch probes.

error message

A numbered or alphabetical message displayed on the MP and/or computer, or RCU indicating a system, programming or operating malfunction.

export

This option allows you to export a part program.

extension

A metal shaft that attaches to the probe to make it longer. You use an extension when you need to measure hard-to-reach features.

FCS - Function Command String

An MLB command consisting of a string of text, which contains the measurement command's name and any parameters.

feature

Something you measure or construct. Some feature types are point, line, plane, circle, sphere, ellipse, cylinder, holes, faces, or slots, etc.

feature number

As you record features, the measurement processor numbers them sequentially. The feature number appears in the Feature Register, the Feature Stack, the Feature Storage, and Inspection Reports.

Feature Register

Data storage reserved for the most recently measured or constructed feature.

Feature Stack

A set of eight data storage areas that temporarily store data entered from the Feature Register.

Feature Storage

Assignable storage area for features. You assign an alphanumeric label to identify the data.

file

A collection of information that the computer treats as a unit.

first-named axis

The first letter (X, Y, or Z) in the name of the working plane. For example, the first-named axis in the XY working plane is the X-axis.

FLB - Function Library

The Function Library has been renamed to the Measurement Library (MLB) for the DirectInspect™ and MeasureMax+ products. It is a library of special functions, commands and variables to help you measure and inspect parts.

footswitch

A pedal for use in recording points during manual measurement. When the probe contacts the desired point, you press the footswitch to store the point coordinates in the measurement processor (MP). The MP sounds a beep to indicate that the point has been recorded.

form

The amount a feature deviates from perfect. The higher the number for form, the more the feature deviates from perfect. If form equals zero, the feature is perfect.

FORTRAN

The acronym for Formula Translator, a computer language often used in the sciences.

full logging

An inspection logging mode which stores all data according to the current report format defined in a part program. The system stores data each time an inspection report command, report heading command or identification data command appears in a part program.

function

A command you select to perform a part inspection operation. When you select this function, you call one or more commands from the Command Library located in the MP firmware and add the correct parameters.

Function Command String - FCS

An MLB command consisting of a string of text, which contains the measurement command's name and any parameters.

Function Library - FLB

The Function Library has been renamed to the Measurement Library (MLB) for the DirectInspect™ and MeasureMax+ products. It is library of special functions, commands and variables to help you measure and inspect parts.

host computer

The computer that is controlling the system. Other computers, accessories or equipment can communicate with the host computer.

ID - Inner Diameter

The distance from a point on one side of an inner circle to a point on the opposite side of an inner circle. The line that defines the diameter passes through the center of the feature. Also used to refer to a feature such as a bore which is open space surrounded by material.

import

This menu option allows you to import a part program.

incremental

A command for a DCC probe move or touch that is a distance from the current position along the specified axes.

initialize

To set starting values for hardware, software, or disks so that they are ready to use.

Inner Diameter - ID

The distance from a point on one side of an inner circle to a point on the opposite side of an inner circle. The line that defines the diameter passes through the center of the feature. Also used to refer to a feature such as a bore which is open space surrounded by material.

inspection log

A record of the data collected while a part program is running.

inspection report

A report that records inspection data that result when you use a single part program one time. An inspection report gives you a hard copy of the data in an inspection log. You define the format for an inspection report in the part program.

interactive

Interactive communication is a specific way to communicate with the computer. You tell the computer what to do by entering a command. The computer performs the task. It then responds, confirming that it performed the task.

internal summation array

Data storage area in the measurement processor for the points needed to measure or construct a feature. The system stores the data in a compressed format so you can use a nearly unlimited number of points to measure a feature.

joystick

A lever located on the Remote Control Unit (RCU). You move the joystick to control servo probe movement.

keyboard

The part of a computer that looks like a typewriter.

limit tolerancing

A tolerancing mode which specifies the maximum and minimum allowable size for a feature.

listing

A line-by-line list on the computer screen or printer of all the steps in a part program.

log directory

A list of all the inspection logs stored on a disk.

log file

A record of inspection report results from one run of a part program. Log files can be used for statistical analysis of a number of parts measured with the same part program.

logging mode

The way data is recorded in an inspection log. There are three logging modes: full logging, partial logging, and log only.

machine axes

The axes built into the coordinate measuring machine. These axes are different on vertical arm machines than they are on horizontal arm machines. After you perform part alignment, the system translates all measurements from the machine axis values to part axis values.

machine setup

Includes: selecting probes, calibrating probes, setting scale factors, setting measurement units, and selecting DCC settings.

major radius

The greatest possible distance from an ellipse to its center.

manual

A type of coordinate measuring machine (CMM) or measurement operation that requires the help of an operator to manually position the probe and record measurements while running the program.

Maximum Material Condition - MMC

A method of tolerancing which allows increased position tolerance as feature size departs from its condition of maximum material (smallest bore/largest boss).

MEA - Microprocessor-Enhanced Accuracy

Accuracy is increased by using mathematical formulas.

Measurement Library - MLB

The command language used by the system.

Measurement Processor - MP

A microprocessor which works with a coordinate measuring machine to store and use measurement data and control operations. The MP contains a portion of the library of measurement commands, processes operator commands made through the computer, reads the feature data sent from the CMM, performs all measurement calculations, compiles feature data in temporary and permanent buffers, and controls DCC operations.

measurement unit

The units, in inches or millimeters, used to display and/or print measurement results.

menu

A list of items displayed on the computer screen for selection.

Microprocessor-Enhanced Accuracy - MEA

Accuracy is increased by using mathematical formulas.

midplane

A plane located midway between two points, lines, or planes and normal to an imaginary line connecting them.

midpoint

A point at the center of a line segment.

minor radius

The smallest possible distance from an ellipse to its center.

MMC - Maximum Material Condition

A method of tolerancing which allows increased position tolerance as feature size departs from its condition of maximum material (smallest bore/largest boss).

MP configuration file

A file created using the Configuration Data program. This file sets the system default values for the coordinate measuring system.

nominal

Blue-print dimension of a feature.

normal

A surface, plane, or axis that forms a 90-degree angle with another plane or axis.

numeric variable

A value which contains only numbers.

OD - Outer Diameter

The distance from a point on the outside of a feature to a point on the opposite side of the outer surface. The line that defines the diameter passes through the center of the feature. Also used to refer to a feature such as a solid cylinder (material surrounded by space).

offset angle

An angle established which mathematically aligns the coordinate measuring machine's measurement axes with the part measurement axes.

offset plane

A plane which does not actually exist but which you create based on three part features (usually machined pads on an unmachined surface). The system calculates the plane by heights (unnamed-axis values) the operator specifies to the coordinate positions of three measured points on the pads.

orientation

Locating the probe with respect to the part feature or some other reference.

out-of-tolerance

Amount by which a dimension exceeds the tolerance range.

Outer Diameter - OD

The distance from a point on the outside of a feature to a point on the opposite side of the outer surface. The line that defines the diameter passes through the center of the feature. Also used to refer to a feature such as a solid cylinder (material surrounded by space).

overdrive distance

The distance that the probe can travel past a touch target location before the failure to touch the part becomes an error.

paginate

Code in the ‘ModMain.bas’ file is split into smaller parts to resolve most compilation problems in the program due to size.

parallelism

All points on a surface or axis are the same distance from a datum plane or axis.

parameter

A varying value at a set location in the program. Parameters affect operations. The specific value of a parameter changes depending on previous measurements or settings.

parametric variable

A data value in memory that has been labeled by the system. Although the label remains constant, the data changes as you record measurements. Parametric variables specify: the values of a part reference frame, a Feature Stack or Feature Storage position, a value such as a probe radius.

part alignment

The process of mathematically aligning the system's measurement axes with the parts axes as specified on the part print.

part axes

The axes after rotation through autolevel and offset.

part program

A list of steps used to inspect a part. These programs are created in Quick Teach, where you can enter settings, measure, construct, calibrate commands, etc.

part program directory

A list of all the part programs on a disk.

part reference frame

The zero points and alignment defined by the three part axes (X,Y,Z). The part reference frame is determined by selecting the active working plane, aligning the part for inspection, and choosing the part datum.

partial logging

Partial logging records only actual values in an inspection log. Inspection logs produced by partial logging cannot be used for statistical analysis requiring nominal and tolerance information unless you enter this information.

perpendicularity

A condition when a surface, plane or axis forms a 90-degree angle with a datum plane or axis.

PH6A

A rigid, detachable probe body mounted on the CMM's probe shaft which connects to various probes used with an autochange rack.

PH9/PH10

A two-axis motorized probe head. You can command PH9/PH10 probe movements and teach them within a part program.

polar

The polar radius and the polar angle. The polar radius is the distance from a point to the unnamed-axis of the working plane. The polar angle is the angle between the vector connecting the point to the unnamed axis and the positive end of the first-named axis of the working plane.

polarity

The orientation of an axis or line with respect to its positive and negative ends.

position

A coordinate location in space referenced to the datum.

probe

The device on the coordinate measuring machine that touches the part and records points for measurement.

probe changer

Autochange Rack. A rack that holds probes and probe changer extensions until you need them. A CMM used with a probe changer has a PH6A, PH9A or PH10 probe head. When you command the system to load a certain probe or combination of probes and extensions, the CMM moves the probe head to the rack. It unloads the currently loaded probe in its rack position. The CMM then moves to the rack position of the specified probe and loads it. If a probe is to be used with extensions, the CMM automatically builds the combination of probes and extensions.

probe compensation

The system's process of modifying a measured feature's size or position based upon adding or subtracting a probe radius to the recorded position(s) of a probe tip's center. The system performs this action during a measurement to record a true feature position or size.

probe crash

Accidental contact between the probe and the part or another object, usually in a programmed probe move. Probe crashes can cause personal injury to an operator or damage to the probe or equipment.

probe diameter

The distance from one side of a probe tip to the other side through the center of the tip.

probe extension

A metal shaft that attaches to the probe to make it longer. You use an extension when you need to measure hard-to-reach features.

probe shaft

The shank to which the probe tip is connected. During DCC probe calibration and reorientation, you specify the shaft length so the system can calculate the correct height at which to contact the calibration sphere.

probe tip

The end point of a hard or touch probe. The probe tip records the coordinates of a point on the part when it contacts the part.

projection

The way the system moves a recorded feature onto a specified plane, usually the working plane. The feature's orientation is then the same as that of the working plane.

projection plane

The plane to which a feature is moved during a projected measurement. During most projected measurements, all features are projected to the working plane. Some ANSI tolerancing commands, however, ask you to specify as a projection plane one of the three datum planes.

prompt

A message on the computer screen that asks you for information or gives you instructions.

Prompt Monitor

A monitor which displays messages and instructions to the operator of the part program.

rack sphere

The sphere on the autochange rack. It is used as a reference for locating the rack stations relative to the probe head.

Regardless of Feature Size - RFS

A method of tolerancing that does not allow increased position tolerance as feature size varies. The tolerance must be met regardless of the size of the feature.

remeasure

A setting that, when turned on, causes the system to display the dimensions of an out-of-tolerance measurement as you run a part program. The system then asks whether or not you want to remeasure the feature. If you want to remeasure, the system does not save the current measurement data. You may choose not to remeasure an out-of-tolerance feature. If you do so, the part program continues.

Remote Control Unit - RCU

A control box used for entering commands for a direct computer controlled (DCC) coordinate measuring machine (CMM). Using the RCU, you can: answer prompts from the computer, move and position the probe, control probe movement speed, and perform calculations.

reorientation

Changing the previously set orientation of Tip 0. Tip 0 is the only tip that may be reoriented. Other previously-calibrated tips must be recalibrated.

report format

The combination of rows and columns which you specify for an inspection report. You can save and recall up to nine report formats within a part program.

rotary table

An accessory on some CMM systems which allows easier access to the part by rotating it relative to the machine base.

run

When a computer uses a list of steps in a part program to inspect a part or group of parts.

runout

The deviation of a feature surface during rotation of a part on the datum axis.

S variable for Dimensional Array

S is an MLB variable for the Dimensional Array. It contains the current feature and is used to generate Inspection Reports. This is a 2-dimensional array (0 to 17, 0 to 5) that can be accessed by a part program as objData.S(n1,n2). Data is entered into the array by MLB commands and assignment statements in the part program.

S0 variable for error number

S0 is an MLB variable for the MLB error number. A value of zero represents no error. This is an integer variable that can be accessed by a part program as objData.S0.

S0D variable for error text

S0D is an MLB variable for an MLB error message or miscellaneous string. This is a string variable that can be accessed by a part program as objData.S0D.

S1 variable for first axis

S1 is an MLB variable for the first-named axis of the working plane. For the XY plane, S1 points to the X row of the Dimensional Array; likewise, YZ plane is the Y row, and ZX plane is the Z row. This is an integer variable that can be accessed by a part program as objData.S1.

S10 variable for temporary data

S10 is an MLB variable for MLB temporary data. This is a real variable that can be accessed by a part program as objData.S10.

S11 variable for temporary data

S11 is an MLB variable for MLB temporary data. This is a real variable that can be accessed by a part program as objData.S11.

S12 variable for temporary data

S12 is an MLB variable for MLB temporary data. This is a real variable that can be accessed by a part program as objData.S12.

S13 variable for temporary data

S13 is an MLB variable for MLB temporary data. This is a real variable that can be accessed by a part program as objData.S13.

S14 variable for temporary data

S14 is an MLB variable for MLB temporary data. This is an integer variable that can be accessed by a part program as objData.S14.

S15 variable for temporary data

S15 is an MLB variable for MLB temporary data. This is an integer variable that can be accessed by a part program as objData.S15.

S16 variable for temporary data

S16 is an MLB variable for MLB temporary data. This is an integer variable that can be accessed by a part program as objData.S16.

S17 variable for temporary data

S17 is an MLB variable for MLB temporary data. This is an integer variable that can be accessed by a part program as objData.S17.

S1D variable for feature name

S1D is an MLB variable for the MLB feature name. This is a string variable that can be accessed by a part program as objData.S1D.

S2 variable for second axis

S2 is an MLB variable for the second-named axis of the working plane. For the XY plane, S2 points to the Y row of the Dimensional Array; likewise, YZ plane is the Z row, and ZX plane is the X row. This is an integer variable that can be accessed by a part program as objData.S2.

S3 variable for third axis

S3 is an MLB variable for the unnamed axis of the working plane. For the XY plane, S3 points to the Z row of the Dimensional Array; likewise, YZ plane is the X row, and ZX plane is the Y row. This is an integer variable that can be accessed by a part program as objData.S3.

S4 variable for feature type

S4 is an MLB variable for the feature type. This is an integer variable that can be accessed by a part program as objData.S4.

S5 variable for feature number

S5 is an MLB variable for the MLB feature number. This is an integer variable that can be accessed by a part program as objData.S5.

S6A(0) variable for report sequence number

S6A(0) is an MLB variable for the MLB report sequence number. This is an integer variable that can be accessed by a part program as objData.S6A(0).

S6A(1) variable for linear resolution

S6A(1) is an MLB variable for the linear data resolution in the MLB report. This is an integer variable that can be accessed by a part program as objData.S6A(1).

S6A(2) variable for angular resolution

S6A(2) is an MLB variable for the angular data resolution in the MLB report. This is an integer variable that can be accessed by a part program as objData.S6A(2).

S7 variable for temporary data

S7 is an MLB variable for MLB temporary data. This is a real variable that can be accessed by a part program as objData.S7.

S8 variable for temporary data

S8 is an MLB variable for MLB temporary data. This is a real variable that can be accessed by a part program as objData.S8.

S9 variable for scan

S9 is an MLB variable for the scan data. It stores the scan targets and actuals. This is a 2-dimensional real array (0 to 3999, 0 to 2) that can be accessed by a part program as objData.S9(n1,n2). Data is entered into the array by MLB commands and assignment statements in the part program.

scale factors

Changes to either part scale or machine scale to: accommodate the effects of temperature on read heads, accommodate the effects of temperature on part scaling, adapt part programs for both sides of symmetrical parts.

screen

Computer video display, also called a Cathode Ray Tube (CRT).

scrolling

To move information on the computer screen up, down, right, or left. Scrolling is used to see more information than the screen can display at one time.

second-named axis

The second letter (X, Y, or Z) in the name of the working plane. For example, in the XY-plane, the second-named axis is the Y-axis.

SFTROTF variable for feature tolerance

SFTROTF is an MLB variable for the feature tolerance. A value of true specifies that the current feature is out-of-tolerance. This is a Boolean variable that can be accessed by a part program as objData.SFTROTF.

SIQMISC variable for miscellaneous data

SIQMISC is an MLB variable for the miscellaneous data. This is a string variable that can be accessed by a part program as objData.SIQMISC.

skewed

Not parallel to another feature. In calculating distance relationships between lines and/or planes, there are special computations for features skewed less than 5ø from each other.

skip action

What happens when a command that is in a program block is skipped during a partial program run. There are four possible results of a skip action: one or two null features enter into the Feature Stack, the system ignores the command, the system generates an error, or the system executes the command.

SOTF variable for part tolerance

SOTF is an MLB variable for the part tolerance. A value of true specifies that one or more features of the part are out-of-tolerance. This is a Boolean variable that can be accessed by a part program as objData.SOTF.

SU variable for User Array

SU is an MLB variable for an extra dimensional array that is defined by the user. This is a 2-dimensional array (0 to 14, 0 to 5) variable that can be accessed by a part program as objData.SU(n1,n2). Data is entered into the array by assignment statements in the part program. Before using the array, the rows must be labeled and the processing type defined. This is done by using the MLB command UA. The array is altered only by the MLB commands UA, DT, IN and ME.

SULABEL variable for User Array

SULABEL is an MLB variable for the User Array. It stores the label for each row of the User Array. This is a string array that can be accessed by a part program as objData.SULABEL(n).

summation array

The user's area to accumulate points for a "best fit feature" computation.

SUPROC variable for User Array

SUPROC is an MLB variable for the User Array. It stores the processing type for each row of the User Array. This is an integer array that can be accessed by a part program as objData.SUPROC(n).

tolerance

An acceptable range for deviation, or the amount a dimension can vary from its nominal. Bilateral tolerancing uses plus and minus tolerance values. Limit tolerancing uses maximum and minimum tolerance values.

touch probe

An electronically sensitive probe which deflects slightly when it contacts a part during measurement.

true position

The exact location of a point, line or plane with respect to a datum.

unnamed axis

The letter (X, Y, Z) not used in the name of the working plane. For example, in the XY plane, the unnamed axis is the Z-axis.

User Array

SU is an MLB variable for an extra dimensional array that is defined by the user. This is a 2-dimensional array (0 to 14, 0 to 5) variable that can be accessed by a part program as objData.SU. Data is entered into the array by assignment statements in the part program. Before using the array, the rows must be labeled and the processing type defined. This is done by using the MLB command UA. The array is altered only by the MLB commands UA, DT, IN and ME.

variable

A symbol whose value may change.

working plane

The active plane in which you measure part features which have a plane definition such as circles and ellipses.

X-axis

A CMM reference line (left- to right+)

Y-axis

A CMM reference line (front- to back+) on vertical CMMs; (down- to up+) on horizontal CMMs.

Z-axis

A CMM reference line (down- to top+) on vertical CMMs, (front- to back+) on horizontal CMMs.

A. Index

2

3

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z