ACLRAT

Function Used to set the acceleration for measurements, safe moves, and rotary tables.
Default DME internally stored default value.
Input ACLRAT / var1, var2, var3
Output None.
Where
var1 MESACL or: POSACL or: ROTACL
var2 MPMM or: IPMM or: RPMM or: PCENT or: HIGH or: LOW
var3 ,n or: does not exist
MESACL signifies that the measurement acceleration, or the acceleration of the sensor for measurement/contact moves, is to be set.
POSACL signifies that the positional acceleration, or the acceleration of the sensor for positioning/safe moves, is to be set.
ROTACL signifies that the rotary table's rotational acceleration is to be set.
HIGH is the DME's internally stored high value.
LOW is the DME's internally stored low value.
n is the acceleration value.
MPMM signifies meters per minute per minute.
IPMM signifies inches per minute per minute.
RPMM signifies revolutions per minute per minute.
PCENT signifies the percent of maximum, i.e. , 0.75 = 75%.
Note: When var2 is HIGH or LOW, var3 does not exist. The following statements are interrelated in the use of rotary tables: ROTDEF, ROTSET, ROTAB, FEDRAT, ACLRAT, and CALIB.

ALGDEF

Function Used to define an algorithm and assign to it a label.
Default None.
Input VA(label) = ALGDEF / CODE, n
Output VA(label) = ALGDEF / CODE, n
Where
VA(label) is an alphanumeric label assigned to the algorithm, and is up to 10 characters in length.
CODE signifies that the algorithm is being defined with a numeric code.
n is any integer(S) representing the previously coded algorithm.
Note: The characterization file contains the various algorithms that are supported by the particular DME. Each of these algorithms is assigned an integer code in the characterization file. The intent of this statement is to support those DMEs (particularly video devices), that maintain several alternate algorithms for use in the inspection or evaluation process. Codes for algorithms can, for example, specify counting, intersections, line fits, circle fits, and maximum, or minimum points, etc. The ALGDEF statement is passed on to the output file when activated by the SNSET statement.

BOUND

Function Applies boundaries to features and tolerances that are unbounded by definition.

Default None.

Input BOUND / var1, var2 var2 ...

Output BOUND / var1, var2 var2 ...

Where

var1 F(label) or: FA(label) (for output only)

var2 F(labeln) or: FA(labeln)

F(label) is the previously defined feature to be bounded. A Sphere, Cone, Line or a Cylinder bound.

F(labeln) is a previously defined bounding plane that is to be one of the bounds for F(label). Plane reducible features such as an arc, circle, plane or an elongated hole used as the bounding planes.

FA(labeln) is a previously measured bounding plane that is to be one of the bounds for F(label). Plane reducible features such as an arc, circle, plane or an elongated hole used as the bounding planes.

Note: There are no limits in DMIS for the number of planes that used to bound a feature; however, the characterization file identifies the limitation for the DME. In the following example, there are six (6) bounding planes for the cone. BOUND / F(cone), F(pln1), F(pln2), FA(pln1), FA(pln2), F(pln3), FA(pln3). Features that are unbounded by definition include: planes, lines, cones, and cylinders. The BOUND command is passed to the output file along with the bounding plane definitions by the DME. When boundary information is used by the DME in feature measurement or tolerance evaluation, the actuals are output in the BOUND statement. When the DME ignores boundary information, the nominals are output in the bound statement.

CALIB

Function Used to calibrate a sensor element or rotary table prior to taking measurements.

Default None.

Input CALIB / var1

Output None.

Where

var1 SENS, S(label1), var2 or: RTAB, RT(label1), var3

var2 FA(label1), n or: FA(label1), 'text'

var3 FA(label2), n or: FA(label2), FA(label3)

SENS signifies that a sensor is to be calibrated.

S(label1) is the name of the sensor to be calibrated.

RTAB signifies that a rotary table is to be calibrated.

RT(label1) is the name of the rotary table to be calibrated.

FA(label1) is a previously defined feature of known dimensions or characteristics to be used for the calibration.

n is the number of measurements to be taken for the calibration.

'text' is the name of an algorithm or subroutine, resident to the DME, that will be used for the calibration.

FA(label2) is a previously defined feature, typically the center of rotation of the rotary table, (center bung).

FA(label3) is a previously defined or constructed feature. For example, in the case where var3 is FA(label2), FA(label3): A ball mounted on the rotary table, and measurements taken to determine the ball's center in at least 3 different rotational positions. FA(label2) the center of the constructed circle that passes through the ball centers, and FA(label3) the constructed plane in which the ball centers lie.

Note:

The objective for the CALIB statement is to provide a means by which the location and orientation of sensors and rotary tables accurately established relative to the machine coordinates. The location and orientation specified in the nominal definition for the feature to be used in calibration is irrelevant for manual calibration (MODE/MAN). They are significant, however, if calibration is to occur in AUTO or PROG mode. In this case, they are positioned with respect to the active coordinate system. The size specified is the actual size, which is known prior to calibration. When MODE/AUTO is in effect, the calibration is done automatically using the DME's algorithms. When MODE/PROG is in effect, the move and measure commands following CALIB are used to measure the calibrating feature. The calibration sequence is terminated with an ENDMES. When MODE/MAN is in effect, the calibration is complete when the operator has measured n points on the feature. If the scan mode is off, the trigger mode of the laser sensor is calibrated. In AUTO mode, the program can limit the surface area of the sphere using the BOUND and DMESW/COMAND,’CLSPH=<num>‘ commands. If the scan mode is on, the scan mode of the laser sensor is calibrated. The mode must be AUTO. Currently only probe type is supported.
POLARIS SPECIFIC CALIBRATION:. The method of Probe calibration on POLARIS is significantly different from a regular machine. Probe calibration begins by measuring enough locations on a calibrated ball to fit a sphere ‘A’. The saddle is moved to the other side of the center and a new set of locations is measured to fit to a sphere ‘B’. Probe deltas are optimized such that these two spheres yield to same location **.**In MODE/MAN or MODE/PROG, ENDMES block expects equal number of measured points to fit to two spheres. The part program should have appropriate PTMEAS commands to collect data. When MODE/AUTO is in effect, the calibration is done automatically using the DME's algorithms.

CALL

Function Calls and invokes execution of a macro.

Default None.

Input CALL/M(label) var1

Output None.

Where

var1 var2 or: var2, var3, ... , varn or: does not exist. var2 is the value of the argument to be passed to the routine. var2,var3, varn are the values of the arguments in a list to be passed to the routine.

M(label) is the name of a previously defined macro routine which is being invoked.

Note: The number of values in the CALL statement argument list must be exactly equal to the number of dummy parameters in the MACRO statement. Ordering is significant; the first value is used in place of the first dummy parameter; the second value is used in place of the second dummy parameter, and so on. No values are required in the CALL statement when the dummy parameter list in the MACRO statement is empty. Refer to Section 3.12 for further information.

CALL/EXTERN

Function Executes an external DMIS part program file or an Executable file.

Default None

Input CALL/EXTERN,var1,’name’,’args’

Output None

Where

var1 DMIS or: SYS.

DMIS signifies that the file is a DMIS part program file.

SYS signifies that the file is an executable.

‘name’ is the complete file name.

‘args’ comma separated argument list passed if required.

Note: Upto 10 levels of nested DMIS CALL/EXTERNs are allowed.

CLMPID

Function Defines the identification of a part holding clamp.

Default None.

Input CI(label) = CLMPID / 'text'

Output None. (Activated with the REPORT statement).

Where

CI(label) is the name associated with this definition.

'text' is an alphanumeric field, enclosed with apostrophes that identifies the part holding clamp.

Note:

CLMPSN

Function Defines the identification of a part holding clamp's serial number.

Default None.

Input CS(label) = CLMPSN / 'text'

Output None. (Activated with the REPORT statement).

Where

CS(label) is the name associated with this definition.

'text' is an alphanumeric field, enclosed with apostrophes that identifies the part holding clamp's serial number.

Note:

CONST

Function

Default CONST / var1, F(label1), BF, FA(label2), var2 var3 ...

Input None.

Output

Where

var1 ARC or: CIRCLE or: ELLIPS or: LINE or: PLANE or: SPHERE

var2 FA(label3) or: F(label3)

var3 **,**var2 or: does not exist

ARC signifies that an arc is to be constructed.

CIRCLE signifies that a circle is to be constructed.

ELLIPS signifies that a ellipse is to be constructed.

LINE signifies that a line is to be constructed.

PLANE signifies that a plane is to be constructed.

SPHERE signifies that a sphere is to be constructed.

F(label1) is the feature to be constructed.

BF signifies that the constructed feature is a best fit through the features that follow.

FA(label2) are the previously defined features to be used for the
FA(label3) construction, the first of which must be a previously measured
F(label3) feature.

Note: Any feature that reduces to a point (arc, circle, cone, ellipse, point, or sphere) used with BF; the point to which the feature reduces is used in the construction. Specifically, for arcs, circles and spheres, the centerpoint is taken; for cones, the vertex is used; and for ellipses, the intersection point of the major and minor axes is taken. The minimum number of previously defined features required for these constructions is given as follows:

ARC	3
CIRCLE	3
ELLIPSE	5
LINE	2
PLANE	3
SPHERE	4

In the case of an ARC construction, the end points will be defined as the points where the best-fit arc is intersected by radial lines drawn through FA(Label2) and the last feature specified in the format list. All points are coplanar and lie on the arc. In the case of a CIRCLE construction, the points are coplanar and lie on the circle. In the case of an ELLIPS construction, the points are coplanar and lie on the ellipse. In the case of a SPHERE construction, all points lie on the sphere. Since all constructed features have a nominal feature definition, i.e. , F(label1) sent down in the program, there should be no ambiguities in the construction. When more than one result is possible from a given construction, the desired result is that which most closely agrees with the nominal feature definition.

CONST

Function

Default CONST / LINE, F(label1), var1, FA(label2)

Input None.

Output

Where

var1 MIDLI, FA(label2),var2 or: PROJLI, FA(label2)

var2 FA(label3) or: F(label3)

LINE signifies that a line is to be constructed.

F(label1) is the feature to be constructed.

MIDLI signifies that the feature to be constructed is to be the midline of the two previously defined features.

PROJLI signifies that the feature to be constructed is to be the projection of the previously defined feature into the current work plane.

FA(label2)
FA(label3)
F(label3) are the previously defined features to be used for the construction, the first of which must be a previously measured feature.

Note: The MIDLI bisects intersecting lines and is parallel to, and half way between, parallel lines. In the case of two skew lines or axes (A) and (B), one or both of which are unbounded, the unbounded MIDLI described in the following way: Lines A and B have a midplane which is defined as perpendicular to the line of closest approach between A and B and passing through the midpoint of that line. The MIDLI between A and B, then, lies in the midplane and is the bisector of the angle between the respective projections of A and B into the midplane. In the case of two skew lines or axes (A) and (B), both of which are bounded, the bounded MIDLI defined in the following way. Two line segments C and D defined which connect the corresponding endpoints of line A and B. The MIDLI is the line segment connecting the midpoints of the lines C and D. Since all constructed features have a nominal feature definition, i.e. , F(label1) sent down in the program, there should be no ambiguities in the construction. When more than one result is possible from a given construction, the desired result is that which most closely agrees with the nominal feature definition.

CONST

Function

Default CONST / PLANE, F(label1), MIDPL, FA(label2), var1

Input None.

Output

Where

var1 FA(label3) or: F (label3)

PLANE signifies that a plane is to be constructed.

F(label1) is the feature to be constructed.

MIDPL signifies that the plane to be constructed is to be a midplane between the two features that follow.

FA(label2) are the previously defined planes, lines or feature axes, to be
FA(label3) used for the construction, the first of which must be a
F(label3) previously measured feature.

Note: The midplane bisects intersecting planes, lines or axes; is half way between, and parallel to parallel planes, lines or axes. Two skew lines (A) and (B) have a midplane which is defined as perpendicular to the line of closest approach between A and B passing through the midpoint of that line. Since all constructed features have a nominal feature definition, i.e. , F(label1) sent down in the program, there should be no ambiguities in the construction. When more than one result is possible from a given construction, the desired result is that which most closely agrees with the nominal feature definition.

CONST

Function

Default CONST / POINT, F(label1), var1

Input None.

Output

Where

var1 MIDPT, FA(label2) var2 or: VERTEX, FA(label2) or: PROJPT, FA(label2) or: MOVEPT, FA(label2),dx,dy,dz

var2 ,FA(label3) or: ,F(label3)

POINT signifies that a point is to be constructed.

F(label1) is the feature to be constructed.

MIDPT signifies that the feature to be constructed is to be the midpoint of the two previously defined features that follow. All point-reducible features used.

MOVEPT signifies that the feature to be constructed is to be offset from a previously measured point designated by the delta distance given. Almost all features used to obtain a moved feature except a GCURVE, GSURF and elongated hole features.

VERTEX signifies that the feature to be constructed is to be the vertex of the previously defined feature that follows. If a feature argument supplied does not have a vertex, then the feature reduced to it's point is used.

PROJPT signifies that the feature to be constructed is to be the projection of the previously defined feature into current work plane.

FA(label2) are the previously defined features to be used for the
FA(label3) construction, the first of which must be a previously measured
F(label3) feature.

dx,dy,dz delta distance for MOVEPT.

Note: When MIDPT is specified, the FA(label2) and FA(label3) or F(label3) are two features which reduce to points, the first of which is measured. When VERTEX is specified, FA(label2) is a cone. When PROJPT is specified, FA(label2) is a feature that reduces to a point. When MOVEPT is specified, the FA(label2) is a point.

Since all constructed features have a nominal feature definition, i.e. , F(label1) sent down in the program, there should be no ambiguities in the construction. When more than one result is possible from a given construction, the desired result is that which most closely agrees with the nominal feature definition.

CONST

Function

Default CONST / var1, F(label1), PROJCT, FA(label2)

Input None.

Output

Where

var1 ARC or: CIRCLE

ARC signifies that an arc is to be constructed.

CIRCLE signifies that a circle is to be constructed.

F(label1) is the feature to be constructed.

PROJCT signifies that the feature to be constructed is to be the projection of the previously defined and measured feature into the current work plane.

FA(label2) specifies the previously defined and measured feature to be used for the construction.

Note: This construction is useful for further constructions that require features to be coplanar. The nominal i,j,k, vector of the feature being projected must be perpendicular to the current work plane. Since all constructed features have a nominal feature definition, i.e. , F(label1) sent down in the program, there should be no ambiguities in the construction. When more than one result is possible from a given construction, the desired result is that which most closely agrees with the nominal feature definition.

CONST

Function

Default CONST / var1, FA(label2), var3

Input None.

Output

Where

var1 CIRCLE, F(label1), var2 or: LINE, F(label1), var2 or: POINT, F(label1), INTOF

var2 TANTO or: INTOF

var3 FA(label3) or: F(label3)

CIRCLE signifies that a circle is to be constructed.

LINE signifies that a line is to be constructed.

POINT signifies that a point is to be constructed.

F(label1) is the feature to be constructed.

TANTO signifies that the constructed feature is to be tangent to the features that follow; at least the first of which must be a previously measured feature.

INTOF signifies that the constructed feature is given by the inter- section of the features that follow; at least the first of which must be a previously measured feature.

FA(label2) are the previously defined features to be used for the FA(label3) construction, the first of which must be a previously measured F(label3) feature.

Note When CIRCLE, TANTO are specified, FA(label2) is a previously defined and measured line or circle, and var3 is a previously defined, measured, or constructed line or circle, coplanar with FA(label2). In the case of constructing a circle tangent to two coplanar lines, the lines must not be parallel. The constructed circle's diameter will be that specified in the nominal feature definition. When required to construct a circle tangent to two parallel lines, use format 7, where the thru point is both the circle's tangent point, and the point on the second line. In the case of constructing a circle tangent to two coplanar circles, the circles must not be circumscribed. The constructed circle's diameter will be that specified in the nominal feature definition. When the nominal diameter is less that the minimum tangent circle, the minimum tangent circle will be constructed. When CIRCLE, INTOF are specified, FA(label2) is a previously defined and measured plane, and var3 is a previously defined, measured, or constructed cone. The cone's axis must be perpendicular to the plane of FA(label2). When LINE and TANTO are specified then, FA (label2) is a previously defined and measured circle, and var3 is a previously defined, measured, or constructed circle, coplanar with FA(label2). When LINE and INTOF are specified then, FA(label2) is a previously defined and measured plane, and var3 is a previously defined or measured plane. When POINT, INTOF are specified then: When FA(label2) is a previously defined and measured line then, var3 a previously defined line, coplanar with FA(label2). or: a previously defined circle, coplanar with FA(label2). or: a previously defined plane. When FA(label2) is a previously defined and measured circle then, var3 a previously defined, measured, or constructed line, coplanar with FA(label2). or: a previously defined measured, or constructed circle, coplanar with FA(label2). Since all constructed features have a nominal feature definition, i.e. , F(label1) sent down in the program, there should be no ambiguities in the construction. When more than one result is possible from a given construction, the desired result is that which most closely agrees with the nominal feature definition.

CONST

Function

Default CONST / var1, var3

Input None.

Output

Where

var1 CIRCLE, F(label1), TANTO or: LINE, F(label1), var2 or: PLANE, F(label1), var2

var2 PERPTO or: TANTO or: PARTO

var3 FA(label2), THRU, var4 or: F(label2), THRU, FA(label3)

var4 FA(label3) or: F(label3)

CIRCLE signifies that a minimum diameter circle is to be constructed tangent to two features.

LINE signifies that a line is to be constructed.

PLANE signifies that a plane is to be constructed.

F(label1) is the feature to be constructed.

PERPTO signifies that the constructed feature is to be perpendicular to the following features.

TANTO signifies that the constructed feature is to be tangent to the following features.

PARTO signifies that the constructed feature is to be parallel to the following features.

FA(label2) is a previously defined feature to be used in the construction.

F(label2) Note that it need not be previously measured.

THRU signifies that the feature being constructed passes through the following point.

FA(label3) is a previously defined point or feature reducing to a point.

F(label3)

Note: When CIRCLE, TANTO are specified: var3 is: a previously defined, measured, or constructed line or circle, in which case the constructed circle will lie in the plane defined by var3 and the through point. Note that var3 and the through point must be coplanar. A previously defined plane, in which case the through point must not lie in the plane of var3. The constructed circle will lie in a plane perpendicular to the plane of var3 the through point. When LINE and PERPTO are specified, var3 is a previously defined, measured, or constructed line or plane. When LINE and TANTO are specified, var3 is a previously defined, measured, or constructed circle, and the through point is in the plane of the circle but not within the circle. When LINE and PARTO are specified, var3 is a previously defined, measured, or constructed line. When PLANE and PERPTO are specified, var3 is a previously defined, measured, or constructed line. When PLANE and TANTO are specified, var3 is a previously defined, measured, or constructed circle, in which case the constructed plane will be perpendicular to the plane of the circle. The through point must not lie within the cylindrical drive volume produced by a projection of the circle along the circle's vector. When PLANE and PARTO are specified, var3 is a previously defined, measured, or constructed plane. Since all constructed features have a nominal feature definition, i.e. , F(label1) sent down in the program, there should be no ambiguities in the construction. When more than one result is possible from a given construction, the desired result is that which most closely agrees with the nominal feature definition.

CUTCOM

Function Defines a compensation, or process adjustment for a manufacturing device.

Default None.

Input CC(label) = CUTCOM / MD(label), var1

Output None, (activated with the REPORT statement)

Where

var1 LEFT or: RIGHT

var3 XYPLAN or: YZPLAN or: ZXPLAN

CC(label) is the name assigned to this definition.

MD(label) is the name of the manufacturing device to be adjusted.

ADJUST signifies that a cutter compensation adjustment for a specific tool is to follow.
PARAM signifies that the compensation to follow is in parameter form.
MATRIX signifies that the compensation to follow is in matrix form.
USERDF signifies that the compensation to follow is user defined.
TL(label) is the name of the tool to be compensated.
LEFT signifies that cutter compensation is directed in the left direction.
RIGHT signifies that cutter compensation is directed in the right direction.
XYPLAN signifies the XYPLAN orientation.
YZPLAN signifies the YZPLAN orientation.
ZXPLAN signifies the ZXPLAN orientation.
x,y,z are the x,y,z coordinates for the parameter adjustment.
a,b,c are the vectors for the parameter adjustment.
dx,dy,dz are the delta coordinates for the matrix adjustment.
ix,iy,iz are the i vectors for the x,y,z adjustment.
jx,jy,jz are the j vectors for the x,y,z adjustment.
kx,ky,kz are the k vectors for the x,y,z adjustment.
amt is the value for the amount of adjustment.
'text' is an alphanumeric field, enclosed with apostrophes that identifies the compensation or process adjustment. This parameter is particularly useful for applications in the electronics industry.
Note: This statement is associated with the MFGDEF and TOOLDF statements for applications to adjust the manufacturing process based on inspection results. The CUTCOM statement is passed on to the output file when executed.

DATDEF

Function Assigns a datum label to a previously measured feature(s).
Default None.
Input DATDEF / var1
Output DATDEF / var1
Where
var1 FA(label1), DAT(x) or: FA(label2), DAT(x-x)
FA(label1) is the previously measured feature to be associated with the datum.
FA(label2) is the previously constructed feature to be associated with the compound datum.
DAT(x) is the datum label assigned to the feature. X is one or two upper case alpha characters.
DAT(x-x) is the compound datum label assigned to a CONSTRUCTED FEATURE. Each x is one or two upper case alpha characters separated by a dash, (ASCII 45).
Note: Datums are referenced in the DATSET, ROTATE, TRANS, and TOL statements. This statement simply assigns a datum label to a previously measured feature, or features. The DATSET, ROTATE, TRANS, and TOL statements control the orientation and alignment for the application of the datum labels. The DATDEF statement is passed through to the output file at the time it is executed. The compound datum, DAT(x-x), is commonly used with the tolerances for runout, i.e. , TOL/CRNOUT, and TOL/TRNOUT.

DATSET

Function Defines and activates a datum set, or part coordinate system, and assigns to it a label.

Default None.

Input D(label) = DATSET / var1 var7

Output D(label) = DATSET / var1 var7

Where

var1 MCS or: var2 var3

var2 DAT(x), var5 var6

var3 var2 var4 or: Does not exist

var4 DAT(x), var6 or: Does not exist

var5 XDIR -XDIR YDIR -YDIR ZDIR -ZDIR

var6 ,XORIG YORIG ZORIG Does not exist

var7 ,CR(label)

D(label) is the name assigned to the part coordinate system. A new label is required for each DATSET.

MCS signifies that the DME's machine coordinate system is to be reset and activated, causing the previous part coordinate system (if any), to be canceled.

DAT(x) is the datum label used to define the part coordinate system axis that follows.

XDIR signifies that the positive x direction is given by the direction vector of the preceding feature. **-**

XDIR signifies that the negative x direction is given by the direction vector of the preceding feature.

YDIR signifies that the positive y direction is given by the direction vector of the preceding feature. **-**

YDIR signifies that the negative y direction is given by the direction vector of the preceding feature.

ZDIR signifies that the positive z direction is given by the direction vector of the preceding feature. **-**

ZDIR signifies that the negative z direction is given by the direction vector of the preceding feature.

CR(label) signifies a previously defined carriage when more than one exists.

Note: that this is optional for systems limited to one active part coordinate system. This label provides the required association between part coordinate systems and carriages when programming parallel or dual systems.

XORIG signifies that the x-component of the datum is used to establish the origin.

YORIG signifies that the y-component of the datum is used to establish the origin.

ZORIG signifies that the z-component of the datum is used to establish the origin.

Note: Since the features of parts are defined and toleranced in reference to datums, a part coordinate system must be established reflecting those datums before the features measured. Also, since a DME has its own coordinate system, -- a system of three mutually orthogonal axes of motion known as the machine coordinate system, the part coordinate system must be created within the machine coordinate system. A complete part coordinate system consists of three mutually orthogonal planes whose paired intersections represent the axes, and whose mutual intersection depict the origin. DMEs establish part coordinate systems by measuring datum features or references and specifying them as the required elements of the part coordinate system. The DATSET statement provides for the establishment of the orientation, alignment, and origin, of the part coordinate system. The TRANS statement can also establish the origin of the part coordinate system, or translate it to establish a new one. The ROTATE statement can also establish the alignment of the part coordinate system, or rotate it to establish a new one. Together, the DATSET, ROTATE, and TRANS statements establish part coordinate systems in three steps: 1) ORIENTING_the_PRIMARY_AXIS, 2) ALIGNING_the_SECONDARY_AXIS, 3) ESTABLISHING_the_ORIGIN. The first_step in creating a part coordinate system is to establish the direction of the primary axis based on the primary datum reference of the part. This must be determined with a minimum of three points, and could be the X, Y, or Z axis. Once determined, this axis and the plane normal to it become the primary axis and primary plane establishing the ORIENTATION of the part coordinate system. This is commonly the Z axis. For example:

```
D(mcs) = DATSET / MCS
F(plane_1) = FEAT / PLANE,CART,0,0,0,0,0,1
MEAS / PLANE, F(plane_1), 3
PTMEAS / CART, 0,1,0,0,0,1 ...
ENDMES
DATDEF / FA(plane_1), DAT(A)
D(orient 1) = DATSET / DAT(A), ZDIR, ZORIG
```

With the primary axis and plane established, the directions of the two remaining part coordinate system axes may be OPTIONALLY determined, (in some cases, a primary axis and plane is sufficient). These remaining axes lie in the primary plane and, are perpendicular to each other and with the primary axis. The second_step in creating a part coordinate system is to establish the direction of the secondary datum referenced on the part for the ALIGNMENT of the secondary axis. This must be established with a minimum of two points and could be the X, Y, or Z axis as long as it does not conflict with the primary axis, (must be orthogonal to the primary axis). This is commonly either the X or Y axis. For example: ... .

```
D(orient_1) = DATSET / DAT(A), ZDIR, ZORIG
$$ Measure two circles, construct a line through their centers and
$$ align the Y axis to the vector of the line. The alignment
$$ in this case will be established with the ROTATE and DATSET
$$ statements to illustrate their application.
F(circle_1) = FEAT / CIRCLE, INNER, CART,0,0,0,0,0,1,10
MEAS / CIRCLE, F(circle_1), 3
PTMEAS / CART, ... .
ENDMES
F(circle_2) = FEAT / CIRCLE, INNER, CART,0,20,0,0,0,1,10
MEAS / CIRCLE, F(circle_2), 3
```

```
PTMEAS / CART, ... ... . .
ENDMES
F(line_1) = FEAT / LINE, CART, 0,0,0,0,1,0
CONST / LINE, F(line_1), BF, FA(circle_1), FA(circle_2)
DATDEF / FA(line_1), DAT(B)
D(align_1) = ROTATE / ZAXIS, DAT(B), YDIR
```

Once the secondary axis is established, the direction of the third or tertiary part coordinate system axis is automatically locked in at the same time since it is perpendicular to the primary and secondary axes by definition. With the directions of the three part coordinate system axes established, the final step is to determine the zero point or origin. The third_step in establishing a part coordinate system is to determine the zero starting point or ORIGIN. This final step is accomplished with the TRANS or DATSET statements. For example:

```
$$ Set the X and Y axis origin to the center of FA(circle_1)
$$ and the Z axis origin on FA(plane_1) DATDEF / FA(circle_1), DAT(C)
D(REF_SYS_1) = TRANS / XORIG, DAT(C), YORIG, DAT(C), ZORIG, DAT(A)
$$ Save the part coordinate system for later re-use.
```

SAVE / D(REF_SYS_1) The part coordinate systems established by the DATSET, ROTATE, and TRANS statements saved with the SAVE statement and recalled for later use with the RECALL statement. They can also be rotated or translated to establish new coordinate systems. When creating a coordinate system, it is fair to assume that the following general rule applies: 1. The primary is the first DATUM assigned.2. The secondary is always perpendicular to the primary datum. The DME will force perpendicularity. 3. The tertiary is always perpendicular to both the primary and secondary datums. Here again, the DME will force perpendicularity. The DATSET statement is passed through to the output file at the time it is executed. No measurements accomplished, unless they are required to create the part coordinate system, until the initial part coordinate system is established. XORIG, YORIG and ZORIG can only apply once in any given DATSET. An alternative to the previously described approach is to establish the part coordinate system using one DATSET command. It is important to note, however, that when the following procedure is utilized, the DATSET must remain totally intact. The use of the tertiary portion of the DATSET is optional and is totally dependent upon the specific application. For example:

```
D(mcs) = DATSET/MCS
F(plane_1) = FEAT/PLANE,CART,0,0,0,0,0,1
MEAS/PLANE, F(plane_1),3 ... ...
ENDMES
DATDEF/FA(plane_1),DAT(A)
F(circle_1) = FEAT/CIRCLE,INNER,CART,0,0,0,0,0,1,10
MEAS/CIRCLE,F(circle_1),3 ... ...
ENDMES
DATDEF/FA(circle_1),DAT(c)
F(circle_2) = FEAT/CIRCLE,INNER,CART,0,20,0,0,0,1,10
```

```
MEAS\CIRCLE,F(circle_2),3 ... ...
ENDMES
F(line_1) = FEAT/LINE,CART,0,0,0,0,1,0,1,0,0
CONST/LINE,F(line_1),BF,FA(circle_1),FA(circle_2)
DATDEF/FA(line_1),DAT(B)
D(ref_sys_1) = DATSET/DAT(A), ZDIR,ZORIG,DAT(B),
XDIR,YORIG,DAT(C), YDIR,XORIG
```

DECL

Function Declares variables of several data types to be used in the program.

Default None.

Input DECL / var1, var2, ... varn

Output None.

Where

var1 CHAR, n or: INTGR or: REAL

var2 is the variable name of the declared data type.

varn is the nth variable name in a list of variables of the declared data type.

CHAR signifies a character data type.

n is the number of character spaces assigned to the variable.

INTGR signifies an integer data type.

REAL signifies a real data type.

Note: All DMIS variables are declared. Variable names consist of a string of letters, digits, and underscores. The first character of a name must be a letter, and names may have up to 6 characters. Variable names should not use DMIS reserve words (major words, minor words, definitions, etc.) and are not case sensitive. All variables declared in the main program are global to the entire program. Variables declared in a routine are local to that routine. String variables are used as dummy labels in macro routines. When a character data type is declared, the data assigned to the variable name is enclosed with apostrophes. The 'n' value does not include the apostrophes. Any number of variables declared with a single DECL statement.

DELETE

Function Deletes Feature Nominal and Actual.

Default None

Input DELETE/var1

Output None

Where

var1 FA(label)

FA(label) Is the name of a previously measured feature actual to be deleted.

Note:

DIRNAM

Function This command sets the directory path name.

Default Current working directory.
Input DIRNAM/var1,'name',var2
Output None
Where
var1 COORD or: SENS or: DATA or: FEAT.
COORD signifies that the path is for coordinate system files.
SENS signifies that the path is for sensor files.
DATA signifies that the path is for part program files.
FEAT signifies that the path is for feature files.
var2 INPUT or: OUTPUT.
INPUT signifies that the path is for recalling the type of information that is specified in the rest of the command.
OUTPUT that the path is for storing the information that is specified in the rest of the command.
'name' is the complete path name; it must begin and end with single apostrophe.
Note: The directory name is limited to 80 characters. Axiom places a "\" Or "/" as required between the directory name and the file name. The DIRNAM command specifies the path for the next FILNAM command with the same file type.

DISPLY

Function Specifies the current device to which output data will be sent and the format in which it will be sent.
Default None.
Input DISPLY/var1
Output None
Where
var1 PRINT,var2,var3,var3,var3 or: TERM,var2,var3,var3,var3 or: STOR,var2,var3,var3,var3 or: COMM,var2,var3,var3,var3 or: None
var2 DMIS or: V(label) or: DMIS,V(label) or: SG(label)
var3 ,var1,var2 or: does not exist
PRINT signifies that the output format(s) specified by the following minor word(s) will be output as a PRINTed report.
TERM signifies that the output format(s) specified by the following minor word(s) will be output to the video TERMinal.
STOR signifies that the output format(s) specified by the following minor word(s) will be output to magnetic STORage.
COMM signifies that the output format(s) specified by the following minor word(s) will be output to the auxiliary COMMunications port.
DMIS signifies that the output will be in DMIS format.
V(label) is the name of the output label in vendor format, as specified by the VFORM statement.
SG(label) is the name of the scanning graphics settings for output, as specified by the SCNGRF statement.
Note: The DISPLY statement may be used more than once in a DMIS input program.

VFORM and SCNGRF must be previously defined when used in a DISPLY statement. Refer to Section 3.13 Output Data for additional information.

DMEID

Function Defines the identification of a Dimensional Measuring Device.
Default None.
Input DI(label) = DMEID / 'text'
Output None, (Activated with the REPORT statement).
Where
DI(label) is the name associated with this definition.
'text' is an alphanumeric field, enclosed with apostrophes that identifies the DME.
Note:

DMESW

Function Used to control data or the processing of data sent in the input file to the DME.
Default None
Input DMESW/var1
Output DMESW/var1
Where
var1 COMAND, 'command' or: PAUSE
COMAND signifies that the string of alphanumeric characters that follow will be interpreted as a DME-specific command. This is used to code DME-specific instructions that are not within the capability of the DMIS interface.
command is the command sent to the receiving system. The following is the list commands supported.

‘MCU=val’	send val to MCU
‘ACC2=val’	where val is “C” :or “A2 C” :or “XYZ”. Tool changer commands.
‘AUDIO=val’	where val is “EXTERNAL” :or “INTERNAL” :or “OFF”.
‘CCU=val’	where val is the RS232 port to communicate with motion controller.
‘PRINTER=val’	where val is the port number where the printer is connected.
‘NAME=val’	Field for a name.
‘ADDRESS=val’	Field to specify address
‘CITY=val’	Field to specify city
‘STATE=val’	Field to specify state
‘ZIP=val’	Field to specify the zipcode.
‘PHONE=val’	Field to specify phone number
’CONTACT=val’	Contact supervisor etc.
‘DIGITS=val’.	Number of significant digits used in reports.
‘TPP=val1 val2 val3’	Primary sensor mount axis.
‘TPS=val1 val2 val3’	Secondary sensor mount axis.
‘TCOMP=val1’.	Room temperature.
Or ’TCOMP= val1 val2 val3’.	val1, val2 and val3 are the temperature values from x, y and z axes.
‘PECOMP=val’	Part expansion coefficient.

‘PTCOMP=val’. Part Temperature.

‘MODEL=val’. Machine model number.

‘SERIAL=val’. Serial number.

‘DUMCS’. Define User’s MCS.

‘CLUMCS’. Clear User’s MCS.

‘SYMMETRY=val’. Initialize symmetry option. val is the plane’s feature label to be used in symmetry.

‘DESYMM’. Disable symmetry option.

‘SMP=val’. val is the complete path name for the SMP file.

‘CLSPH=val’ val is always < 1.0. Typically used for AUTO mode point generation. Specifies that (1.0-val) percent of the feature should be used. A typical value of .9 implies that, only 10 percent of the feature must be used.

‘ACCDOCK=val1 val2 val3’. Vector perpendicular to the face of the rack. This signifies the vector along which the tool approaches the rack.

‘ACCLENGTH=val1 val2 val3’.A Vector from port 1 to port n.

‘RESET’. Reset Text Query Labels.

‘PHC=val’. val is the RS232 port to communicate with probe head controller.

‘ACC2=val’. val is the RS232 port to communicate with Auto Tool changer.

‘FAST=val’. val is the fast serial port for scanned data.

‘KPD=val’. Port for External numeric keypad.

‘BLOCK=val’. val ON or OFF. Toggles blocking feature on/off.

PAUSE signifies that the DME is to stop processing data. All data received after the DMESW/PAUSE statement is encountered and before the next DMESW/CONTIN statement is encountered are ignored by the DME. The DMESW statement is passed on to the output file when executed.

‘LASER=val1 val2 val3’ Start laser calibration of axis.
Currently laser calibration of rotary axis Y is not supported.
val1is the axis id. (0) for X-Axis Or (1) for Y-Axis Or (2) for Z-Axis.
val2 is the laser calibration type. It ranges between 0 and 5.
val3 is 0 if calibration. This generates compensation tables. And is 1 if verification.
Note: Refer to LASER.DMS sample program.
Example: If the axis id(val1) is 0(i.e. X-Axis).
For Linear val2=0; Straightness X-Y, val2=1; Straightness X-Z, val2=2; Roll, val2=3; Pitch, val2=4; Yaw, val2(5);

‘LASER=’ Stop/Finish laser calibration. This completes with generating compensation tables

Polaris Specific:

‘BALLBAR=val’ Start/Stop Ballbar data capture. Refer to Ballbar.dms sample program.
val=1 Starts Ballbar data capture.

val=0 Ends Ballbar data capture.

‘CMDTOXMAX=val’ Toggles to an alternate X-Axis position.
val=0 Default. All moves that are commanded by AXIOM are such that the saddle is on one side of the center only. i.e. the saddle never moves past the center.
val=1 X-Axis is commanded such that it moves past center.

‘MCU=val PATH_DIR’ Set a flag to Control the direction of rotation of Y-Axis.
val=CCW Y always rotates Counter-Clockwise
val=CW Y always rotates Clockwise.
val=SHORT Y axis follows shortest direction of rotation.

‘MCU= val NREV_MOVE_Y’
Move Y-Axis through N revolutions. val is a floating point number which specifies the number of revolutions. The sign controls the direction of rotation (+=CCW). Ex. val=0.5 rotates Y-axis 180.0 degrees counter clockwise.

‘MCU=COMPUTE_TOTAL’ Diagnostic routine to check total counts on Y-Axis scale for one revolution.

‘MCU=HIT_HOME’ Diagnostic routine to locate home switch on the Y-Axis.

Note:

DMESWV

Function Defines the identification of the DME's software version.
Default None.
Input DV(label) = DMESWV / 'text'
Output None, (Activated with the REPORT statement).
Where
DV(label) is the name associated with this definition.
'text' is an alphanumeric field, enclosed with apostrophes that identifies the DME's software version.
Note:

DMISMN

Function External program identification for a DMIS input program. Assigned by the receiving system.
Default
Input DMISMN/'text'
Output None
Where
'text' is a string of printable ASCII characters.
Note: This string of ASCII characters must begin and end with an apostrophe. When the string of characters must extend to another line, use a single dollar sign, '$', at the end of the line. DMISMN designates the beginning of the main input program and it must be the first line of executable code in the DMIS input program. For

example:

```
System File Name (DMIS INPUT PROGRAM)
ANC101. DMI
DMISMN/'ANC101 TEST PROGRAM'
V(vendor)= VFORM/ALL
DISPLY/PRINT,DMIS,STOR,V(vendor)
FILNAM/'ANC101 Test Program Results'
CALL/EXTERN,DME,'9W3309,SUB'
ENDFIL
(DMIS MODULE)   9W3309. SUB
DMISMD/'Bolt_Circle_Routine'
ENDFIL
(DMIS OUTPUT FILE)
ANC101. DMO
FILNAM/'ANC101 Test Program Results'
DMISMD/'Bolt_Circle_Routine'
ENDFIL
```

DO

Function	To provide the capability of repeating a sequence of instructions based on an initial and limit value at a specified increment.
Default	None
Input	DO/(label),index=initial value, limit, value, var1
Output	None
Where	
var1	is the increment value or doesn’t exist.
index	is the DO loop index variable. It is a previously DECL/INTGR variable.
initial	is the integer initial value of the DO loop index variable.
limit	is the integer limit value of the DO loop index variable.
increment	is the integer increment value. If omitted, the increment is 1.
Note:	The variables above must be an integer or integer variable. When DO loops are nested, care should be exercised to include corresponding ENDDO statements for every DO statement.

ELSE

Function	Controls a branch in a conditional.
Default	None.
Input	ELSE
Output	None.

Note: ELSE follows an associated IF statement. If the conditional test in the IF statement fails, program control is transferred to the line following the ELSE. ELSE is optional, but when used, is the first and only word on a line.

ENDAT

Function Signifies the end of a data stream.
Default None.
Input None.
Output ENDAT Used to terminate data in a raw data listing.

Note:

ENDDO

Function Indicates the end of a DO.
Default ENDDO
Input None
Output

Note:

ENDFIL

Function Specifies the end of the program.
Default None.
Input ENDFIL
Output

ENDIF

Function Signifies the end of a conditional.
Default None.
Input ENDIF
Output None.

Note:

ENDMAC

Function Terminates a macro sequence definition.
Default None.
Input ENDMAC
Output
None.
Note:

ENDMES

Function Signifies the end of a measurement sequence.

Default None.

Input ENDMES

Output None.

Note:

ERROR

Function Used to convey the handling of DME error codes.

Default None.

Input ERROR / (label), ercode

Output ERROR / (label), ercode

(label) is a statement label for control to branch to (JUMPTO), upon encountering an error condition.

ercode is a previously defined variable assigned to the error code. Error codes are identified in the characterization file.

Note: This statement provides the capability of handling errors that would otherwise suspend processing when error conditions are encountered while in automatic or program mode. Typically, this statement is placed in the beginning of a program. When an error condition is encountered that is associated with the error code in this statement, control of the program is passed to the line having the statement (label). The appropriate ERROR statement is passed on to the output file when an error condition is encountered.

EVAL

Function Executes the evaluation of a feature(s) to a tolerance(s).

Default None.

Input EVAL/var1

Output None

Where

var1 FA(label), var2 or: FA(label1), FA(label2), T(label1)

var2 T(label) or: T(label2),T(label3),... T(labeln) or: does not exist

FA(label) is the name of the feature to be evaluated.

FA(label1) is the name of the first feature to be associated with a relationship tolerance, i.e. , TOL/ANGLB or TOL/DISTB.

FA(label2) is the name of the second feature to be associated with a relationship tolerance, i.e. , TOL/ANGLB or TOL/DISTB.

T(label) is the name of the tolerance associated with the FA(label).

T(label1) is the name of the relationship tolerance, i.e. , TOL/ANGLB or TOL/DISTB, associated with the measured features.

T(label2) is the name of the first tolerance in a list associated with the measured feature.

T(labeln) is the name of the nth tolerance in a list associated with the measured feature.

Note: The execution of the EVAL statement results in the computation of the tolerance actual(s), i.e. , TA(label). The labels of these actuals can then be used in conjunction with the following statements: OBTAIN, IF, CUTCOM, or OUTPUT. The EVAL statement is useful to generate actuals required for selective processing, or IPV, but not necessarily intended for output. For example:

```
EVAL / FA(cir1), T(siz1)
DECL / REAL, cir1_dia
cir1di = OBTAIN / FA(cir1),10
IF (cir1di . LT. .245)
MD(jig_bore) = MFGDEV / '6673 JIG BORE'
TL(bore) = TOOLDF / MD(jig_bore), '22470 TOOLNO BORE TOOL'
CC(bore1) = CUTCOM/MD(jig_bore),ADJUST,TL(bore),LEFT,XYPLAN,lotol1
ENDIF
GOHOME.
```

The EVAL statement is passed on to the output file when executed.

FEAT/ARC

Function Defines a nominal arc and assigns to it a label.

Default None.

Input F(label)=FEAT/ARC,var1,var2,i,j,k,rad,ang1,ang2

Output FA(label)=FEAT/ARC,var1,var2,i,j,k,rad,ang1,ang2

Where

var1 INNER or: OUTER

var2 CART,x,y,z or: POL,r,a,h

label is an alphanumeric label assigned to the feature, and is up 10 characters in length.

ARC signifies that the feature is an arc.

INNER signifies that the inside of the arc is to be measured (i.e. , a fillet).

OUTER signifies that outside of the arc is to be measured (i.e. , a round).

CART signifies that the center is given by cartesian coordinates.

POL signifies that the center is given by polar coordinates.

x,y,z are the cartesian coordinates of the centerpoint of the arc.

r,a,h are the polar coordinates of the centerpoint of the arc.

i,j,k is the direction vector of the plane that the arc lies in.

rad is the radius of the arc.

ang1 is the start angle of the arc lying on, or parallel to, the WKPLAN of the current part coordinate system, and relative to the major axis of the WKPLAN. Use the right hand rule for sign conventions.

ang2 is the positive included angle of the arc relative to ang1.

Note: The x,y,z point coordinates are given relative to the origin of the active part coordinate system. The r and a values are also given relative to the active part coordinate system. The h value is the perpendicular distance of the point from the WKPLAN.

FEAT/CIRCLE

Function Defines a nominal circle and assigns to it a label.

Default None.
Input F(label)=FEAT/CIRCLE,var1,var2,i,j,k,diam
Output FA(label)=FEAT/CIRCLE,var1,var2,i,j,k,diam
Where
var1 INNER or: OUTER
var2 CART,x,y,z or: POL,r,a,h
label is an alphanumeric label assigned to the feature, and is up to 10 characters in length.
CIRCLE signifies that the feature is a circle.
INNER signifies that the inside of the circle is to be measured (i.e. , a hole).
OUTER signifies that outside of the circle is to be measured (i.e. , a boss).
CART signifies that the center is given by cartesian coordinates.
POL signifies that the center is given by polar coordinates.
x,y,z are the cartesian coordinates of the centerpoint of the circle.
r,a,h are the polar coordinates of the centerpoint of the circle.
i,j,k is the direction vector of the plane that the circle lies in.
diam is the diameter of the circle.
Note: The x,y,z point coordinates are given relative to the origin of the active part coordinate system. The r and a values are also given relative to the active part coordinate system. The h value is the perpendicular distance of the point from the WKPLAN.

FEAT/CONE

Function Defines a nominal cone and assigns to it a label.
Default None.
Input F(label)=FEAT/CONE,var1,var2,i,j,k,ang
Output FA(label)=FEAT/CONE,var1,var2,i,j,k,ang
Where
var1 INNER or: OUTER
var2 CART,x,y,z or: POL,r,a,h
label is an alphanumeric label assigned to the feature, and is up to 10 characters in length.
CONE signifies that the feature is a cone.
INNER signifies that the inside of the cone is to be measured (i.e. , a conical hole).
OUTER signifies that the outside of the cone is to be measured (i.e. , a conical boss).
CART signifies that the coordinates of the vertex are stated in cartesian coordinates.
POL signifies that the coordinates of the vertex are stated in polar coordinates.
x,y,z are the cartesian coordinates of the vertex.
r,a,h are the polar coordinates of the vertex.
i,j,k is the direction vector associated with the cone, which points along the cone's axis from the vertex to the open end of the cone.
ang is the included angle of the cone.
Note: The x,y,z point coordinates are given relative to the origin of the active part coordinate system. The r and a values are also given relative to the active part

coordinate system. The h value is the perpendicular distance of the point from the WKPLAN.

FEAT/CYLNDR

Function Defines a nominal cylinder and assigns to it a label.

Default None.

Input F(label)=FEAT/CYLNDR,var1,var2,i,j,k,diam

Output FA(label)=FEAT/CYLNDR,var1,var2,i,j,k,diam

Where

var1 INNER or: OUTER

var2 CART,x,y,z or: POL,r,a,h

label is an alphanumeric label assigned to the feature, and is up to 10 characters in length.

INNER signifies that the insides of the cylinder is to be measured (i.e. , a cylindrical hole).

CYLNDR signifies that the feature is a cylinder.

OUTER signifies that outside of the cylinder is to be measured (i.e. , a cylindrical boss).

CART signifies that the coordinates of a point on the axis are stated in cartesian coordinates.

POL signifies that the coordinates of a point on the axis are stated in polar coordinates.

x,y,z are the cartesian coordinates of a point on the cylinder's axis; the point is the centerpoint for bounded cylinders.

r,a,h are the polar coordinates of a point on the cylinder's axis; the point is the centerpoint for bounded cylinders.

i,j,k is the direction vector associated with the cylinder, and points along the cylinder's axis; from the first end measured to the last end measured.

diam is the diameter of the cylinder.

Note: The x,y,z point coordinates are given relative to the origin of the active part coordinate system. The r and a values are also given relative to the active part coordinate system. The h value is the perpendicular distance of the point from the WKPLAN.

FEAT/EHOLE

Function Defines a nominal elongated hole and assigns to it a label.

Default None.

Input F(label)=FEAT/EHOLE,var1,var2,var3,i,j,k,i1,j1,k1,len,wid

Output FA(label)=FEAT/EHOLE,var1,var2,var3,i,j,k,i1,j1,k1,len,wid

Where

var1 INNER or: OUTER.

var2 ROUND or: SQUARE.

var3 CART,x,y,z or: POL,r,a,h

label is an alphanumeric label assigned to the feature, and is up 10 characters in length.

EHOLE signifies that the feature is an elongated hole.

CART signifies that the center is given by cartesian coordinates.

POL signifies that the center is given by polar coordinates.

x,y,z are the cartesian coordinates of the centerpoint of the ehole.

r,a,h are the polar coordinates of the centerpoint of the ehole.

i,j,k is the direction of the ehole.

i1,j1,k1 is the direction of the ehole orientation.

wid is the width of the ehole.

len is the length of the ehole.

Note:

FEAT/ELLIPS

Function Defines a nominal ellipse and assigns to it a label.

Default None.

Input F(label)=FEAT/ELLIPS,var1,var2,var3,i,j,k,diam

Output FA(label)=FEAT/ELLIPS,var1,var2,var3,i,j,k,diam

Where

var1 INNER or: OUTER

var2 CART,f1x,f1y,f1z,f2x,f2y,f2z or: POL,f1r,f1a,f1h,f2r,f2a,f2h

var3 MAJOR or: MINOR

label is an alphanumeric label assigned to the feature, and is up to 10 characters in length.

ELLIPS signifies that the feature is an ellipse.

INNER signifies that the inside of the ellipse is to be measured (i.e. , an elliptical hole).

OUTER signifies that outside of the ellipse is to be measured (i.e. , an elliptical boss).

CART signifies that the coordinates of the foci are stated in cartesian coordinates.

POL signifies that the coordinates of the foci are stated in polar coordinates.

f1x,f1y,f1z
f2x,f2y,f2z are the cartesian coordinates of the two foci.

f1r,f1a,f1h
f2r,f2a,f2h are the polar coordinates of the two foci.

MAJOR signifies that the major diameter is to be defined.

MINOR signifies that the minor diameter is to be defined.

i,j,k is the direction vector of the plane that the ellipse lies in.

diam is the diameter of the specified MAJOR or MINOR parameter.

Note: The x,y,z point coordinates are given relative to the origin of the active part coordinate system. The r and a values are also given relative to the active part coordinate system. The h value is the perpendicular distance of the point from the WKPLAN. The TOL/DIAM is associated with the MAJOR or MINOR diameter defined here.

FEAT/GCURVE

Function Assigns a label to a general curve.

Default None.

Input F(label)=FEAT/GCURVE,var1,i,j,k,var3

Output FA(label)=FEAT/GCURVE,var2,i,j,k /var4 /var4 /... ENDAT

Where

var1 CART,x,y,z or: POL,r,a,h

var2 CART then var4 is: x,y,z

var2 POL or then var4 is: r,a,h

var3 xs, ys, zs, is, js, ks Or None.

label is an alphanumeric label assigned to the feature, and is up to 10 characters in length.

GCURVE signifies that the feature is a general curve.

CART signifies that the coordinates of the point are stated in cartesian coordinates, and that the output data will be in cartesian coordinates.

POL signifies that the coordinates of the point are stated in polar coordinates, and that the output data will be in polar coordinates.

x,y,z are the cartesian coordinates of a point on the plane in which the curve lies, whether actual or nominal.

r,a,h are the polar coordinates of a point on the plane in which the curve lies, whether actual or nominal.

i,j,k is the direction vector of the plane in which the curve lies.

xs,ys,zs are the cartesian coordinates of the starting point of Adaptive scan.

is,js,ks is the approach vector of the start point.

r,a,h are the polar coordinates of a point on the plane in which the curve lies, whether actual or nominal.

i,j,k is the direction vector of the plane in which the curve lies.

ENDAT is the major word signifying the end of the data defining GCURVE.

Note: The x,y,z point coordinates are given relative to the origin of the active part coordinate system. The r and a values are also given relative to the active part coordinate system. The h value is the perpendicular distance of the point from the WKPLAN. GCURVE is used to assign a label to a curve, and used when digitizing a part. The nominal feature definition specifies the plane in which the feature should lie. Output data from a measured GCURVE can represent points on the curve if PRCOMP is "ON" or can represent raw data, x,y,z, or r,a,h, data for center of probe, when PRCOMP is "OFF. "

Adaptive Scanning:

Adaptive scanning of a GCURVE is activated by AUTO mode measurement of a gcurve. **var3** specifies the starting point and it’s approach vector.

Example:

```
MODE/AUTO,PROG,MAN
F(GC01)=FEAT/GCURVE,x,y,z,i,j,k,xs,ys.zs,is,js,ks
MEAS/GCURVE,F(GC01),n
ENDMES
```

FEAT/GSURF

Function Assigns a label to a general surface.

Default None.

Input F(label)=FEAT/GSURF

Output FA(label)=FEAT/GSURF /x,y,z /x,y,z /... ENDAT

Where

label is an alphanumeric label assigned to the feature, and is up to 10 characters in length.

GSURF signifies that the feature is a general surface.

x,y,z are the actual measured x,y,z coordinates.

ENDAT is the major word signifying the end of the data defining GSURF.

Note: GSURF is used to assign a label to a surface, and used when digitizing a part. Output data from a measured GSURF can represent points on the surface if PRCOMP is "ON" or can represent raw data, x,y,z, or r,a,h, data for center of probe, when PRCOMP is "OFF. " An example of the FEAT/GSURF definition follows:

```
PRCOMP/ON F(label)=FEAT/GSURF
MEAS/GSURF,F(label),5
PTMEAS/CART,x,y,z,i,j,k
PTMEAS/CART,x,y,z,i,j,k
PTMEAS/CART,x,y,z,i,j,k
PTMEAS/CART,x,y,z,i,j,k
PTMEAS/CART,x,y,z,i,j,k
ENDMES
```

When PRCOMP/ON is initiated, the i,j,k vectors surface normals of each PTMEAS is required.

FEAT/LINE

Function Defines a nominal line and assigns to it a label.

Default None.

Input F(label)=FEAT/LINE,var1 var4

Output FA(label)=FEAT/LINE,var1 var4

Where

var1 UNBND, var2 or: BND, var3

var2 CART,x,y,z,i,j,k, or: POL,r,a,h,i,j,k,

var3 CART,e1x,e1y,e1z,e2x,e2y,e2z, or: POL,e1r,e1a,e1h,e2r,e2a,e2h, var4 is: ni,nj,nk

label is an alphanumeric label assigned to the feature, and is up to 10 characters in length.

LINE signifies that the feature is a line.

BND signifies that a bounded line is to be defined.

UNBND signifies that an unbounded line is to be defined.

CART signifies that the coordinates of the point on the line are stated in cartesian coordinates.

POL signifies that the coordinates of the point on the line are stated in polar coordinates.

x,y,z are the cartesian coordinates of a point on the line.

r,a,h are the polar coordinates of a point on the line.

i,j,k is the direction vector which lies along the line and points from the first point measured to the second point measured.

ni,nj,nk is the normal vector of the plane in which the line lies.

e1x,e1y,e1z

e2x,e2y,e2z are the cartesian coordinates of the two end points of the line.

e1r,e1a,e1h

e2r,e2a,e2h are the polar coordinates of the two end points of the line.

Note: The x,y,z point coordinates are given relative to the origin of the active part coordinate system. The r and a values are also given relative to the active part coordinate system. The h value is the perpendicular distance of the point from the WKPLAN.

FEAT/PLANE

Function Defines a nominal plane and assigns to it a label.

Default None.

Input F(label)=FEAT/PLANE,var1,i,j,k

Output FA(label)=FEAT/PLANE,var1,i,j,k

Where

var1 CART,x,y,z or: POL,r,a,h

label is an alphanumeric label assigned to the feature, and is up to 10 characters in length.

PLANE signifies that the feature is a plane.

CART signifies that the coordinates of the point on the plane are stated in cartesian coordinates.

POL signifies that the coordinates of the point on the plane are stated in polar coordinates.

x,y,z are the cartesian coordinates of a point on the plane.

r,a,h are the polar coordinates of a point on the plane.

i,j,k is the direction vector of the plane pointing away from the part.

Note: The x,y,z point coordinates are given relative to the origin of the active part coordinate system. The r and a values are also given relative to the active part coordinate system. The h value is the perpendicular distance of the point from the WKPLAN.

FEAT/POINT

Function Defines a nominal point and assigns to it a feature label.

Default None.

Input F(label)=FEAT/POINT,var1 var2

Output FA(label)=FEAT/POINT,var1 var2

Where

var1 CART,x,y,z or: POL,r,a,h

var2 ,i,j,k

label is an alphanumeric label assigned to the feature, and is up to 10 characters in length.

POINT signifies that the feature is a general point.

CART signifies that the coordinates of the point are stated in cartesian coordinates.

POL signifies that the coordinates of the point are stated in polar coordinates.

x,y,z are the cartesian coordinates of the point itself.

r,a,h are the polar coordinates of the point itself.

i,j,k is a vector normal to, and pointing away from, the surface in which the point lies.

Note: The x,y,z point coordinates are given relative to the origin of the active part coordinate system. The r and a values are also given relative to the active part coordinate system. The h value is the perpendicular distance of the point from the WKPLAN.

FEAT/SLOT

Function Defines a nominal slot and assigns to it a label.

Default None.

Input F(label)=FEAT/SLOT,var1,i,j,k,i1,j1,k1,len,wid

Output FA(label)=FEAT/SLOT,var1,i,j,k,i1,j1,k1,len,wid

Where

var1 CART,x,y,z or: POL,r,a,h

label is an alphanumeric label assigned to the feature, and is up 10 characters in length.

SLOT signifies that the feature is a slot.

CART signifies that the center is given by cartesian coordinates.

POL signifies that the center is given by polar coordinates.

x,y,z are the cartesian coordinates of the centerpoint of the slot.

r,a,h are the polar coordinates of the centerpoint of the slot.

i,j,k is the direction of the slot.

i1,j1,k1 is the direction of the slot orientation.

wid is the width of the slot.

len is the length of the slot.

Note:

FEAT/SPHERE

Function Defines a nominal sphere and assigns to it a label.

Default None.

Input F(label)=FEAT/SPHERE,var1,var2,diam

Output FA(label)=FEAT/SPHERE,var1,var2,diam

Where

var1 INNER or: OUTER

var2 CART,x,y,z or: POL,r,a,h

label is an alphanumeric label assigned to the feature, and is up to 10 characters in length.

SPHERE signifies that the feature is a sphere.

INNER signifies that the inside of the sphere is to be measured.

OUTER signifies that outside of the sphere is to be measured.

CART signifies that the center is given by cartesian coordinates.

POL signifies that the center is given by polar coordinates.

x,y,z are the cartesian coordinates of the centerpoint of the sphere.

r,a,h are the polar coordinates of the centerpoint of the sphere.

diam is the diameter of the sphere.

Note: The x,y,z point coordinates are given relative to the origin of the active part coordinate system. The r and a values are also given relative to the active part coordinate system. The h value is the perpendicular distance of the point from the WKPLAN.

FEDRAT

Function Used to set the velocities for measurements, safe moves, and rotary tables.

Default DME internally stored default value.

Input FEDRAT / var1, var2 var3

Output None.

Where

var1 MESVEL or: POSVEL or: ROTVEL or: SCNVEL

var2 MPM or: IPM or: RPM or: PCENT or: HIGH or: LOW

var3 ,n or: does not exist

MESVEL signifies that the measurement velocity, or the velocity of the sensor for measurement/contact moves, is to be set.

POSVEL signifies that the positional velocity, or the velocity of the sensor for positioning/safe moves, is to be set.

ROTVEL signifies that the rotary table's rotational velocity is to be set.

HIGH is the DME's internally stored high value.

LOW is the DME's internally stored low value.

n is the velocity value.

MPM signifies meters per minute.

IPM signifies inches per minute.

RPM signifies revolutions per minute.

PCENT signifies the percent of maximum, i.e. , 0.75 = 75%.

Note: When var2 is HIGH or LOW, var3 does not exist. The following statements are interrelated in the use of rotary tables: ROTDEF, ROTSET, ROTAB, FEDRAT, ACLRAT, and CALIB.

FILNAM

Function This command sets the file name.

Default COMERO.ext where ‘ext’ is the appropriate extension for the file type.

Input FILNAM/var1,’name’,var2,var3

Output None

Where

var1 COORD or: SENS or: DATA or: FEAT.

COORD signifies that it is a coordinate system file.
SENS signifies that it is a sensor file.
DATA signifies that it is a part program file.
FEAT signifies that it is a feature file.
var2 INPUT or: OUTPUT.
INPUT signifies that the file is for recalling the type of information that is specified in the rest of the command.
OUTPUT siginifies that the file is for storing the information that is specified in the rest of the command.
‘name’ is the complete path name; it must begin and end with single apostrophe.
var3 APPEND or: OVERWR if var2 = OUTPUT.
Note: The file names are limited to 40(UNIX) and 13(Windows) characters.

FINPOS

Function Enables or disables the fine positioning feature. When enabled, the sensor is positioned with high resolution accuracy at a low velocity following the normal positioning move.
Default FINPOS/OFF
Input FINPOS/var1
Output None.
Where
var1 ON or: OFF
ON signifies that the fine positioning is to be enabled until the FINPOS/OFF statement is issued.
OFF signifies that the fine positioning is to be disabled until the FINPOS/ON statement is issued.
Note:

FIXTID

Function Defines the identification of a part holding fixture.
Default None.
Input FI(label) = FIXTID / 'text'
Output None, (Activated with the REPORT statement).
Where
FI(label) is the name associated with this definition.
'text' is an alphanumeric field, enclosed with apostrophes that identifies the part holding fixture.
Note:

FIXTSN

Function Defines the identification of a part holding fixture's serial number.
Default None.
Input FS(label) = FIXTSN / 'text'

Output None, (Activated with the REPORT statement).
Where
FS(label) is the name associated with this definition.
'text' is an alphanumeric field, enclosed with apostrophes that identifies the part holding fixture's serial number.
Note:

FROM

Function Defines the home position to be used by the GOHOME statement.
Default None.
Input FROM/x,y,z
Output None.
Where
x,y,z are the cartesian coordinates of the home position in the present part coordinate system.
Note:

GOHOME

Function Used to position the sensor at the coordinates defined in the last FROM statement. The home position is relative to the active part coordinate system in effect when FROM was executed.
Default None.
Input GOHOME
Output None.

Note:

GOTO

Function Executes a sensor move and defines the endpoint of the move.
Default None.
Input GOTO/var1
Output None.
Where
var1 x,y,z or: CONTIN,x,y,z
x,y,z are the cartesian coordinates of the endpoint to which the sensor will travel relative to the origin of the active part coordinate system.
CONTIN indicates that the GOTO is part of a continuos path. This is always issued inside a measurement block when scanning is on. During this path, no scan data is collected. See MEAS for more details.
Note: All parameters must be included in the statement, or an error condition will result, i.e. , GOTO/,,1 is not valid.

IF

Function Tests a variable in a logical expression to determine whether it is true or false, and causes the control of the program to branch based on the result.

Default None.

Input IF (var1 var2 var3) ... . executable statement(s) ... . ELSE ... . executable statement(s) ... . ENDIF

Output None.

Where

var1 any previously declared and assigned variable.

var2 . GT. or: . GE. or: . EQ. or: . LT. or: . LE. or: . NE.

var3 any previously declared and assigned variable, or value.

GT signifies the relational operator meaning greater than.

GE signifies the relational operator meaning greater than or equal to.

EQ signifies the relational operator meaning equal to.

LT signifies the relational operator meaning less than.

LE signifies the relational operator meaning less than or equal to.

NE signifies the relational operator meaning not equal to.

Note: If the logical expression is TRUE, control of the program is passed to the statement(s) following the IF command and executed until an ELSE or ENDIF is encountered. If the logical expression is FALSE, control of the program is passed to the statements following the next occurrence of ELSE. If an ENDIF is encountered before an ELSE, the conditional is ended. A logical IF must begin with an IF and end with an ENDIF statement. ELSE statements are optional. Refer to Section 3.11 for further information. An example of the use the IF statement follows:

```
IF (A . LE. B)
MEAS / CIRCLE, F(cir_1),4
PTMEAS / CART, ... ....
ENDMES
ELSE
MEAS / LINE, ....
PTMEAS / CART, ... ....
ENDMES
ENDIF
```

JUMPTO

Function Unconditionally transfers execution of the program.

Default None.

Input JUMPTO/(label)

Output None.

Where

label is the alphanumeric label of the statement to which the program execution is transferred, and is up to ten characters in length.

Note: The statement to which program execution is transferred by the JUMPTO

statement is labeled in the program in the following manner: the label is enclosed in parentheses and appears at the beginning of the line. JUMPTOs must be forward in the program. That is, program control cannot be passed back to a statement which has already been read. For example:
.... executable statements ....
JUMPTO/(new_probe) ....
by-passed statements ....
(new_probe) SNSLCT/S(next_probe) ....

LASER

Function Initialize and turn laser on/off. The active sensor must be a laser probe.
Default None.
Input LASER/var1
Output None.
Where
var1 INIT,x,y,z or: OFF or: ON or: RESET
INIT siginifies that the machine should position to the following coordinates then initialize the laser.
x,y,z are the cartesian coordinates of the point to which the move is made relative to the active coordinate system.
OFF means that the laser should be turned off.
ON means that the laser should be turned on.
RESET Reset laser.
Note:

LOGICAL_OPERATORS

Function Perform logical test operation. The result of the operation is TRUE(1) if successful or FALSE(0) otherwise.
Default None
Input variable = variable var1 variable
Output None
Where
var1 EQ. or: .NE. or: .LT. or: .LE. or: GT. or: .GE. or: .AND. or: .OR. or: .NOT. .
EQ. Equal to .
NE. Not Equal to .
LT. Less than .
LE. Less than or equal to .
GT. Greater than .
GE. Greater than equal to .
AND. Logical AND .
OR. Logical OR .
NOT. Logical NOT.
Note: .

LOTID

Function Defines the identification of the part lot identifier.
Default None.
Input LI(label) = LOTID / 'text'
Output None, (Activated with the REPORT statement).
Where
LI(label) is the name associated with this definition.
'text' is an alphanumeric field, enclosed with apostrophes that identifies the part's lot.
Note:

MACRO

Function Defines a macro routine and assigns to it a label.
Default None.
Input M(label)=MACRO/var1
Output None.
Where
var1 var2 or: var2, var3, ... varn or: does not exist var2 is the first argument or dummy parameter associated with this macro. var3,varn are the arguments or dummy parameters in a list associated with this macro.
label is an alphanumeric label assigned to the macro being defined, and is up to 10 characters in length. This label will be used by the CALL statement to invoke the macro.
Note: When the macro is called, the actual parameters in the CALL statement will be substituted for the dummy parameters in the argument list. When a dummy parameter is to substitute a label name, it is enclosed in quotation marks. The argument list may be empty. An example of a macro definition and a CALL statement follows:

```
FROM / 0,0,2
MODE / AUTO
M(bolthole) = MACRO / x1,y1,x2,y2,"label1","label2"
GOTO / x1,y1,2
MEAS / CIRCLE,F(label1),4
ENDMES
GOTO / x1,y1,2
GOTO / x2,y2,2
MEAS / CIRCLE / F(label2),4
ENDMES
GOTO / x2,y2,2
GOHOME
ENDMAC
```

CALL / M(bolthole), 1.0, 1.5, -1.0, -1.5, (righthole), (lefthole) A macro is terminated with the ENDMAC statement, and executed with the CALL statement. Refer to Section 3.12 for further information.

MATH_OPERATORS

Function Perform mathematical operations.
Default None
Input variable = variable var1 variable
Output None
Where
var1 + or: **-** or: ***** or: / or: ******
+ Addition
- Subtraction
/ Division
***** Multiplication
****** Exponent(to the power of).
Note: .

MEAS

Function Causes the DME to measure a feature.
Default None.
Input MEAS/var1,F(label),n
Output None.
Where
var1 ARC or: CIRCLE or: CONE or: CYLNDR or: GCURVE or: GSURF or: ELLIPS or: LINE or: PLANE or: POINT or: SPHERE
ARC signifies that an arc is to be measured.
CIRCLE signifies that a circle is to be measured.
CONE signifies that a cone is to be measured.
CYLNDR signifies that a cylinder is to be measured.
GCURVE signifies that a general curve is to be measured.
GSURF signifies that a general surface is to be measured.
ELLIPS signifies that an ellipse is to be measured.
LINE signifies that a line is to be measured.
PLANE signifies that a plane is to be measured.
POINT signifies that a point is to be measured.
SPHERE signifies that a sphere is to be measured.
F(label) is the name of the previously defined feature to be measured.
n is the number of measurements to be taken in the measurement of the feature. It is only required when MODE/MAN and MODE/AUTO are in effect.
Note: **Scanning:** When x,y,z, point data are used to measure planar features, the plane in which the measured feature lies is the plane that is parallel to the primary datum plane of the active part coordinate system, best fit, through the measured points. that the defined feature's normal unit vector is perpendicular to the primary datum plane. In the case of measuring a point: when 'n is 1', the point defined by F(label) is measured. When 'n is 0', the current sensor position is recorded as the point and

assigned to the F(label) name in the 'MEAS/POINT,F(label),0' statement. The MEAS statement is usually followed by a series of PTMEAS and GOTO statements. The MEAS statement is terminated with an ENDMES statement. The DME will measure the feature with its own internal algorithm when MODE/AUTO is active. When the DME lacks this capability, it will default to the next program level. When MODE/PROG is active, the DME will follow the given PTMEAS and GOTO statements to measure the feature. If the feature type is not recognized by the DME, and if a measurement sequence is programmed, the DME will follow the measurement sequence and output raw data for the measurement result. The minimum number of points required to measure each feature is: feature type   minimum n
ARC 3
CIRCLE 3
CONE 6
CYLNDR 6
GCURVE 2
GSURF 3
ELLIPS 5
LINE 2
PLANE 3
POINT 1 (exactly)
SPHERE 4. If scanning is on, the MEAS command begins the scanning measurement block. Only GSURF and GCURVE features may be scanned. The sensor selected must be a sensor which allows scanning. Data collected during scanning are placed in label. ACT. Axiom creates a label. NOM file from the PTMEAS points in the requested path. Any existing label. NOM or label. ACT files are overwritten. Axiom determines the scan path by reading a file or by the commands inside the measurement block. The MODE (AUTO or PROG) determines where Axiom looks. In AUTO mode: Axiom looks for a label. SCN file and reads it. If the label. SCN file does not exist, Axiom issues an error. The . SCN file must contain valid DMIS commands allowed inside a measurement block. The measurement block must be empty. In PROG mode: Axiom reads in commands from the measurement block. MODE/MAN in scanning mode is not supported. The commands (either from the measurement block or from the . SCN file) determine how scanning is performed. Axiom reads the commands sequentially. GOTO/CONTIN and PTMEAS commands create a continuos path. During the part of the path defined by the PTMEAS command, the machine collects data. When a command other than GOTO/CONTIN, PTMEAS or SPLINE is given, the path ends. Axiom then executes scanning along the path. When the continuos path is done other commands are executed. This process continues until ENDMES. Scanning along a path involves creating the path from the PTMEAS and GOTO/CONTIN commands, turning on the laser, beginning scanning, turning off the laser, storing the actual data in the . ACT file.

MFGDEV

Function Defines a manufacturing device.
Default None.
Input MD(label) = MFGDEV / 'text'
Output None, (activated with the REPORT statement)
Where
MD(label) is the name assigned to this definition.
'text' is an alphanumeric field, enclosed with apostrophes that identifies the manufacturing device.
Note: This statement is associated with the CUTCOM and TOOLDF statements for applications to adjust the manufacturing process based on inspection results. The MFGDEF statement is passed on to the output file when executed.

MISC_STRING_OPERATORS

Function String Utilities
Default None
Input variable = LEN(label) or variable =INDEX(label1, label2)
Output None
Where
var1 LEN or: INDEX.
LEN Length of the string
INDEX returns the position of the first occurrence of label2 in label1
Note:

MODE

Function Sets the mode in which the DME will execute the program.
Default None.
Input MODE/var1
Output None, (Activated with the REPORT statement).
Where
var1 AUTO, var2 or: PROG, MAN or: MAN var2 PROG, MAN or: MAN
AUTO signifies the DME will execute GOTARG and MEAS commands ignoring the given intermediate moves and measurements, and will use its own algorithms in their place. DME is under servo control in this mode.
PROG signifies the DME will execute GOTARG and MEAS commands using the given intermediate moves and measurements. DME is under servo control in this mode.
MAN signifies the DME will be driven manually for the measurements and moves, and is not under servo control.
Note: The DME will execute the GOTARG and MEAS commands in the mode given by the first minor word. When the DME is unable to execute in this mode, it will default to the next minor word. When it is unable to execute in this mode, it will default to the last minor word. The decision to default to the next level is made by the DME. If the DME defaults to MAN mode, a message will be output to the operator. The DME tests each MEAS and GOTARG statement independently; if MODE/AUTO,PROG,MAN has been issued the DME can perform some MEAS

and GOTARGs in AUTO mode, some in PROG mode, and others in MAN mode.

OBTAIN

Function Sets a variable equal to any parameter of a definition.

Default None.

Input var1 = OBTAIN / var2, n

Output None.

Where

var1 is the name of a previous declared variable to which the obtained value is assigned.

var2 is a label name for a previously defined, measured, or constructed DMIS statement including: F(label) for FEAT-defined FA(label) for FEAT-actual T(label) for TOL-defined TA(label) for TOL-actual S(label) for SNSDEF

n is the ordinal value of the parameter to be retrieved from the definition list. The first argument after the '/' is ordinal value 1. Ordinal value of 0 returns true if the element is defined.

Note: This statement allows any parameter following the '/' delimiter from any DMIS statement having a label to be obtained, and assigned to a variable name. Where tolerances apply, OBTAIN will evaluate the tolerance statement(s) in the last EVAL and/or OUTPUT commands. For example: EVAL/F(label),T(label) var = OBTAIN/TA(label),3. When character type data is obtained, the string of data is preceded and followed with an apostrophe.

OPERID

Function Defines the identification of the operator running the DME.

Default None.

Input OP(label) = OPERID / 'text'

Output None, (Activated with the REPORT statement).

Where

OP(label) is the name associated with this definition.

'text' is an alphanumeric field, enclosed with apostrophes that identifies the operator.

Note:

OUTPUT

Function Causes results of a DME measurement or evaluation to be output.

Default None.

Input OUTPUT/var1

Output None

Where

var1 FA(label) var2 var4 or: FA(label1), FA(label2), TA(label1) var4 or: F(label) var3 var4 or: F(label1), F(label2), T(label1) var4 or: R(label)

var2 ,TA(label) or: ,TA(label2),TA(label3),... TA(labeln) or: does not exist

var3 ,T(label) or: ,T(label2),T(label3),... T(labeln) or: does not exist var4 ,R(label) or: does not exist

FA(label) is the name of the measured feature for which output data will be output.

FA(label1) is the name of the first measured feature to be associated with a relationship tolerance, i.e. , TOL/ANGLB or TOL/DISTB.

FA(label2) is the name of the second measured feature to be associated with a relationship tolerance, i.e. , TOL/ANGLB or TOL/DISTB.

TA(label) is the name of the actual tolerance associated with the actual feature, FA(label).

TA(label1) is the name of the actual relationship tolerance, i.e. , TOL/ANGLB or TOL/DISTB, associated with the measured features.

TA(label2) is the name of the first actual tolerance in a list associated with the measured feature.

TA(labeln) is the name of the nth actual tolerance in a list associated with the measured feature.

F(label) is the name of the nominal feature for which output data will be output.

F(label1) is the name of the first nominal feature to be associated with a relationship tolerance, i.e. , TOL/ANGLB or TOL/DISTB.

F(label2) is the name of the second nominal feature to be associated with a relationship tolerance, i.e. , TOL/ANGLB or TOL/DISTB.

T(label) is the name of the nominal tolerance associated with the nominal feature, F(label).

T(label1) is the name of the nominal relationship tolerance, i.e. , TOL/ANGLB or TOL/DISTB, associated with the nominal features.

T(label2) is the name of the first nominal tolerance in a list associated with the nominal feature.

T(labeln) is the name of the nth nominal tolerance in a list associated with the nominal feature.

R(label) is the name of the report label identifying additional report information required, as defined in the REPORT statement.

Note: Tolerances are generic, in that they have no pointers to features. Association between features and tolerances is provided for with the OUTPUT and EVAL statements. One or several tolerances associated with a feature. Two features are required to be associated with the angle between, TOL/ANGLB, and the distance between, TOL/DISTB, tolerances. DME will output results in the order in which the OUTPUT statements appear in the input program. The OUTPUT statement is passed on to the output file when executed.

PARTID

Function Defines the identification of the part to be inspected.

Default None.

Input PN(label) = PARTID / 'text'

Output None, (Activated with the REPORT statement).

Where

PN(label) is the name associated with this definition.

'text' is an alphanumeric field, enclosed with apostrophes that identifies the part.

Note:

PARTRV

Function Defines the identification of the part's revision level.
Default None.
Input PR(label) = PARTRV / 'text'
Output None, (Activated with the REPORT statement).
Where
PR(label) is the name associated with this definition.
'text' is an alphanumeric field, enclosed with apostrophes that identifies the part's revision level.
Note:

PARTSN

Function Defines the identification of the part serial number.
Default None.
Input PS(label) = PARTSN / 'text'
Output None, (Activated with the REPORT statement).
Where
PS(label) is the name associated with this definition.
'text' is an alphanumeric field, enclosed with apostrophes that identifies the part serial number.
Note:

PLANID

Function Defines the identification of the inspection plan.
Default None.
Input PL(label) = PLANID / 'text'
Output None, (Activated with the REPORT statement).
Where
PL(label) is the name associated with this definition.
'text' is an alphanumeric field, enclosed with apostrophes that identifies the inspection plan.
Note:

PRCOMP

Function Causes automatic probe compensation to be enabled or disabled.
Default PRCOMP/ON
Input PRCOMP/var1
Output PRCOMP/var1
Where
var1 ON or: OFF
ON signifies that automatic probe compensation is enabled.
OFF signifies that automatic probe compensation is disabled; therefore, output is raw

data - probe centerline.

Note: Raw point data is invoked under the following conditions: 1) when a feature is not supported by the DME, or 2) when PRCOMP is OFF. When PRCOMP/ON is initiated, the i,j,k vector of each PTMEAS is required.

PREVOP

Function Defines the identification of the previous operation.

Default None.

Input PV(label) = PREVOP / 'text'

Output None, (Activated with the REPORT statement).

Where

PV(label) is the name associated with this definition.

'text' is an alphanumeric field, enclosed with apostrophes that identifies the previous operation.

Note: This statement is required to support the inspection of electrical components, but is not limited to applications in the electronics industry.

PROCID

Function Defines the identification of the inspection procedure.

Default None.

Input PC(label) = PROCID / 'text'

Output None, (Activated with the REPORT statement).

Where

PC(label) is the name associated with this definition.

'text' is an alphanumeric field, enclosed with apostrophes that identifies the inspection procedure.

Note: The inspection procedure identifier can also be use in the label of the feature being defined.

PTMEAS

Function Signifies that an automatic point measurement is to be performed.

Default None.

Input PTMEAS/var1 var2

Output None.

Where

var1 CART,x,y,z or: POL,r,a,h

var2 ,i,j,k or: does not exist

CART signifies cartesian coordinates.

POL signifies polar coordinates.

x,y,z are the nominal cartesian coordinates of the point to be measured.

r,a,h are the nominal polar coordinates of the point to be measured.

i,j,k is the direction vector pointing away from the surface of the feature used in making the measurement.

Note: When PRCOMP/ON is initiated, the i,j,k vector of each PTMEAS is required. When scanning is on and the mode is AUTO or PROG, Axiom collects PTMEAS coordinates and creates the scanning path from them. The coordinates must specify surface data in the current units and part coordinate system. The vector is normalized. Using PTMEAS and GOTO/CONTIN commands, the programmer must specify a path that limits slope changes and avoids inflection points along a spline section.

RECALL

Function Enables data stored with the SAVE statement to be recalled.

Default None.

Input RECALL/var1

Output RECALL/var1

Where

var1 D(label) or: S(label) or: FA(label)

D(label) is the name of the part coordinate system to be recalled.

S(label) is the name of the sensor to be recalled.

FA(label) is the name of the measured feature to be recalled

Note: In the case of the part coordinate system, the RECALL statement reactivates the recalled datum set, to become the active part coordinate system. In the case of the sensor, the RECALL statement recalls the previously saved calibration data for that sensor. It is not necessary to RECALL calibration data when utilizing different sensors, providing the data resides in memory. It is acceptable to SAVE calibration data to a mass storage device for subsequent RECALL when the amount of memory is a concern. RECALL/D(label) is passed through to the output file at the time it is executed.

REPORT

Function Specifies additional information to be put in the DME output file.

Default None.

Input R(label) = REPORT/var1 var2 var2, ... var2

Output None

Where

var1 DATE or: DV(label) or: TIME or: FI(label) or: TEMPF or: FS(label) or: TEMPC or: LI(label) or: HUMID or: OP(label) or: ALGOR or: PN(label) or: MODE or: PR(label) or: CI(label) or: PS(label) or: CS(label) or: PL(label) or: DI(label) or: PV(label) or: DS(label) or: PC(label)

var2 ,var1 or: does not exist

label is an alphanumeric label assigned to the report, and is up to 10 characters in length.

DATE signifies that the date at the end of the measurement is to be output.

TIME signifies that the time at the end of the measurement is to be output.

TEMPF signifies that the temperature, in degrees Fahrenheit, at the end of measurement is to be output.

TEMPC signifies that the temperature, in degrees Centigrade, at the end of the measurement is to be output.

HUMID signifies that the relative humidity at the end of the measurement is to be output.

ALGOR signifies the DME will output the type of algorithm used in calculating the feature. This will be a text message.

MODE signifies the DME will output the mode (AUTO,MAN,PROG) in which the feature measurement was made.

CI(label) is the name of the part holding clamp.

CS(label) is the name of the part holding clamp serial number.

DI(label) is the name of the DME. REPORT

DS(label) is the name of the DME software.

DV(label) is the name of the DME software version.

FI(label) is the name of the part holding fixture.

FS(label) is the name of the part holding fixture serial number.

LI(label) is the name of the part lot.

OP(label) is the name of the DME operator.

PN(label) is the name of the part.

PR(label) is the name of the part revision level.

PS(label) is the name of the part serial number.

PL(label) is the name of the inspection plan.

PV(label) is the name of the previous operation.

PC(label) is the name of the inspection procedure.

Note: The additional information requested in the R(label) statement are those data which are relevant to the measurement but cannot be calculated from the feature and tolerance output data. REPORT does not trigger output by itself. The elements specified in REPORT are output through the OUTPUT command when the R(label) is referenced. The specific Output_for each element is as follows:

DATE = YYYY/mm/dd

TIME = hh:mm:ss

TEMPF = INTEGAR

HUMID = INTEGAR

ALGOR = 'text'

MODE AUTO or: MAN or: PROG

CI(label) = 'text'

CS(label) = 'text'

DI(label) = 'text'

DS(label) = 'text'

DV(label) = 'text'

FI(label) = 'text'

FS(label) = 'text'

LI(label) = 'text'

OP(label) = 'text'

PN(label) = 'text'

PR(label) = 'text'

PS(label) = 'text'

PL(label) = 'text'
PV(label) = 'text'
PC(label) = 'text'.
The REPORT statement is passed on to the output file when executed.

RMEAS

Function

Default

Input

Output None.

Note: To be updated later.

ROTAB

Function Explicitly controls the motion of a rotary table on a DME.

Default None.

Input ROTAB / RT(label1), var1, var2, var3, n

Output None

Where

var1 INCR or: ABSL

var2 CW or: CCW or: SHORT

var3 ROTTOT or: ROTORG or: ROTNUL

RT(label1) signifies which previously defined rotary table is to be used.

INCR signifies that the rotation is to be from the current position.

ABSL signifies that the rotation is to an absolute position.

CW signifies clockwise rotation.

CCW signifies counter-clockwise rotation.

SHORT signifies that the rotary table will rotate to the desired position via the shortest direction.

ROTTOT signifies that total updating of the part's coordinate system established by the DATSET command (i.e. , datums, part origin, and axis alignment) will be executed with the desired rotation.

ROTORG signifies that a partial updating of the part's coordinate system, including only the origin, will be executed with the desired rotation. The axis alignment will remain as established from the DATSET command.

ROTNUL signifies that no updating of the part's coordinate system is to occur.

n is the amount of rotation, in angular units of DEG:MIN:SEC, or decimal form, as specified in the UNITS statement. Also, n is always positive.

Note: The following statements are interrelated in the use of rotary tables: ROTDEF, ROTSET, ROTAB, FEDRAT, ACLRAT, and CALIB. The ROTAB statement is passed on to the output file when executed. For an illustration of part coordinate system updating, refer to Figure 7.3.

ROTATE

Function Rotates a part coordinate system about an axis, and assigns to it a label.

Default None.

Input D(label) = ROTATE / var1, var2 var4

Output D(label) = ROTATE / var1, var2 var4

Where

var1 XAXIS or: YAXIS or: ZAXIS

var2 ang or: F(label1), var3 or: FA(label1), var3 or: DAT(x), var3

var3 XDIR or: -XDIR or: YDIR or: -YDIR or: ZDIR or: -ZDIR

var4 ,CR(label) var4 var4 or: does not exist

D(label) is the name of the new part coordinate system.

XAXIS signifies that the rotation occurs about the X axis.

YAXIS signifies that the rotation occurs about the Y axis.

ZAXIS signifies that the rotation occurs about the Z axis.

ang is the angle of rotation in units described in the UNITS statement. Positive angles of rotation about the primary axis (orientation axis), are counterclockwise. Negative angles of rotation about the primary axis are clockwise. For example: When the primary axis is the positive Z axis, the WKPLAN is the XYPLAN. A positive angle of rotation is about the Z axis counterclockwise and relative to the positive X axis.

F(label1) is the nominal feature to be aligned with the following axis direction for the new part coordinate system.

FA(label1) is the measured feature to be aligned with the following axis direction for the new part coordinate system.

DAT(x) is the previously defined datum label to be aligned with the following axis direction for the new part coordinate system.

XDIR signifies that the positive x direction is given by the direction vector of the preceding feature.

-XDIR signifies that the negative x direction is given by the direction vector of the preceding feature.

YDIR signifies that the positive y direction is given by the direction vector of the preceding feature.

-YDIR signifies that the negative y direction is given by the direction vector of the preceding feature.

ZDIR signifies that the positive z direction is given by the direction vector of the preceding feature.

-ZDIR signifies that the negative z direction is given by the direction vector of the preceding feature.

CR(label) signifies a previously defined carriage when more than one exists. Note that this is optional for systems limited to one active part coordinate system. This label provides the required association between part coordinate systems and carriages when programming parallel or dual systems.

Note: There are some implied limitations when rotating to features. For example, it is not possible to rotate the alignment about an axis to a plane that is perpendicular to the axis. Therefore, common sense and good judgment shall prevail. A SAVE statement must be issued prior to the ROTATE statement if the current part coordinate system is to be used again with the RECALL statement. The new part coordinate system is activated when the ROTATE statement is executed. ROTATE is passed through to the output file at the time it is executed.

ROTDEF

Function Used to define a rotary table, and assign to it a label.

Default None.

Input RT(label1) = ROTDEF / x,y,z, i,j,k, var1

Output None

Where

var1 RT(label2) or: does not exist

RT(label1) is the label assigned to the rotary table being defined.

x,y,z signifies the x,y,z coordinates from the DME's zero position to the rotary table's center point.

i,j,k signifies the i,j,k unit vectors for the rotary table's axis of rotation in a direction away from the rotary table's workpiece locating surface.

RT(LABEL2)is another previously defined rotary table that will support the currently defined rotary table.

Note: The rotary table's center point is defined as the point of intersection of the axis of rotation with the plane being used as the rotary table's workpiece locating surface (i.e. , the center point of rotation on the surface of the rotary table). The x, y, z, coordinates and i, j, k, vectors are given relative to the machine coordinate system. The following statements are interrelated in the use of rotary tables: ROTDEF, ROTSET, ROTAB, FEDRAT, ACLRAT, and CALIB. Refer to Figure 7.0 for an illustration of a rotary table program applying the use of several related statements. For an illustration of a sensor rotation vs a rotary table rotation, refer to Figure 7.1. For an illustration of rotary table positioning, refer to Figure 7.2. The ROTDEF statement is passed on to the output file when executed.

```
$$ Define rotary table relative to the DME home position.
RT(ROTARY1)=ROTDEF/40.0,15.0,-10.0, 0,0,1
$$ Set some rotary table parameters.
FEDRAT/LOW ACLRAT/LOW ROTSET/RT(RTAB1),0
$$ Select probe for calibration of rotary table.
SNSLCT/S(PROBE1)
$$ Perform calibration of rotary table.
F(BUNG)=FEAT/CIRCLE/INNER,CART,40.0,15.0,-10, 0,0,1,1.000
CALIB/RTAB,RT(ROTARY1),FA(BUNG),4
MEAS/CIRCLE/F(BUNG),4
PTMEAS/POL,... ...
ENDMES
$$ Mount the part, a disk, on the rotary table and establish the
$$ orientation and alignment for the part coordinate system... ....
$$ Select hooked probe for inspection.
```

```
SNSLCT/S(HOOKPROBE)
$$ Position hooked probe for measurement taking.
GOTO/0,-15.0.2.0
GOTO/0.0,-9.0,-1.5
$$ Now perform measurement on the 16.000 inch diameter.
F(pt1)=FEAT/POINT,POL,8.0,270,-1.5
MEAS/POINT,F(pt1),1
PTMEAS/POL,8.0,270,-1.5, 0,1,0
ENDMES
ROTAB/RT(ROTARY1),INCR,CW,ROTTOT,15.0 ... ...
CONST/CIRCLE,F(16DIA),BF,FA(pt1), FA(pt2),....
```

ROTSET

Function Used to reset the angular counter value for a rotary table.

Default None.

Input ROTSET / RT(label1), val

Output None

Where

RT(label1) is the label of the rotary table to be reset.

val is the reset value ranging from 0 to 359:59:59 degrees.

Note: The angular measurement used must be consistent with the angular units established in the UNITS statement. The ROTSET statement is passed on to the output file when executed. The following statements are interrelated in the use of rotary tables: ROTDEF, ROTSET, ROTAB, FEDRAT, ACLRAT, and CALIB.

SAVE

Function Stores part coordinate system datum sets, sensor calibration data, and measured feature actual data for later recall.

Default None.

Input SAVE/var1

Output None.

Where

var1 D(label) or: S(label) or: FA(label)

D(label) is the name of the part coordinate system to be saved.

S(label) is the name of the sensor to be saved.

FA(label) is the name of a previously measured feature actual to be saved.

Note: When the D(label) or S(label) are required for later use in the input program, they are recalled with the RECALL statement. Prudent use of the SAVE/FA(label) should be exercised because of memory limitations in DMEs. Whenever practical, measured features should be output immediately following a measurement sequence. If an FA(label) has been SAVE'd one only needs to use the feature to recall it. Recall is not needed for features.

SCAN

Function To toggle scan mode ON/OFF. This affects the measurement and calibration blocks.

Default OFF.

Input SCAN/var1

Output None.

Where

var1 ON or: OFF

ON scanning is on.

OFF scanning is off.

Note:

SCIENTIFIC_OPERATORS

Function Some common scientific functions.

Default None

Input variable =var1(label)

Output None

Where

var1 ABS or:ACOS or:ASIN or:ATAN or:COS or:DBLE or:EXP or:INT or:LOG or:LOG10 or:REAL or:NINT or:SIN or:SQRT or:TAN.

ABS Absolute value

ACOS Arc-Cosine

ASIN Arc-Sine

ATAN Arc-Tangent

COS Cosine of angle

DBLE Double value

EXP Exponential

INT Integer number

LOG Natural Logarithm

LOG10 Logarithm to base 10

REAL A real number

NINT Integer number

SIN Sine of angle

SQRT Square-root of number

TAN Tangent of number

Note: .

SNSDEF

Function

Default None.

Input S(label) = SNSDEF / NONCON, var1, var-2, proben

Output None.

Where

var1 FIXED or: INDEX

var2 CART, dx,dy,dz, ti,tj,tk or: POL, tilt, rot, ti,tj,tk or: VEC, i,j,k, ti,tj,tk

S(label) is an alphanumeric label assigned to the sensor, and is up to 10 characters in length.

NONCON signifies that a non-contact sensor is being defined. For example, an electrical capacitance probe having several sensors.

FIXED signifies that a fixed sensor is being defined.

INDEX signifies that an indexable sensor is being defined.

CART signifies that the sensor tip location is to be defined in cartesian coordinates.

POL signifies that the sensor tip location is to be defined in polar coordinates.

VEC signifies that the sensor tip location is to be defined in unit vectors.

dx,dy,dz is the distance in X, Y, and Z, between the sensor tip center and the sensor reference point in machine coordinates.

ti,tj,tk is the unit vector of the sensor mount socket's axis.

i,j,k is the unit vector from the sensor reference point to the probe tip center point using machine coordinates.

tilt is the sensor's angle of tilt with respect to the fully extended position where the tilt angle is zero. The tilt angle is always positive.

rot is the sensor's angle of rotation with respect to the positive X axis. Apply the right hand rule for sign convention.

proben is the number of sensors, or probes, that this device provides.

Note:

SNSDEF

Function Defines sensor orientation, type and parameters

Default None.

Input S(label) = SNSDEF / PROBE, var1, var2

Output None.

Where

var1 FIXED or: INDEX

var2 CART, dx,dy,dz, ti,tj,tk, diam or: POL, tilt, rot, ti,tj,tk, len, diam or: VEC, i,j,k, ti,tj,tk, len, diam

S(label) is an alphanumeric label assigned to the sensor, and is up to 10 characters in length.

PROBE signifies that a CMM probe is being defined.

FIXED signifies that a fixed probe is being defined.

INDEX signifies that a motorized indexable probe is being defined.

CART signifies that the probe tip location is to be defined in cartesian coordinates.

POL signifies that the probe tip location is to be defined in polar coordinates.

VEC signifies that the probe tip location is to be defined in unit vectors.

dx,dy,dz is the distance in X, Y, and Z, between the probe tip center and the sensor reference point in machine coordinates. See note below.

ti,tj,tk is the unit vector of the sensor mount socket's axis.

i,j,k is the unit vector from the sensor mount to the probe tip center point using machine coordinates.
tilt is the sensor's angle of tilt with respect to the fully extended position where the tilt angle is zero. The tilt angle is always positive.
rot is the sensor's angle of rotation with respect to the positive X axis. Apply the right hand rule for sign convention.
len is the total length between the sensor mount and the probe tip center point.
diam is the diameter of the sensor tip.
Note: The sensor mount is the initial extension connect point. It is the intersection of the axis of each sensor socket with the plane of the tool holder face or axis of rotation of motorized indexable sensors. For mechanical probes, dx,dy,dz,len, and diam, are updated with actual values when the probe is calibrated with the CALIB statement. The actual values of these parameters are saved and recalled with the SAVE and RECALL commands or deleted with the DELETE command.

SNSDEF

Function
Default
Input S(label) = SNSDEF / VIDEO, var1, var2, focal, mag, apert
Output None.
Where
var1 FIXED or: INDEX
var2 CART, dx,dy,dz, ti,tj,tk or: POL, tilt, rot, ti,tj,tk or: VEC, i,j,k, ti,tj,tk
S(label) is an alphanumeric label assigned to the sensor, and is up to 10 characters in length.
VIDEO signifies that a video camera is being defined.
FIXED signifies that a fixed camera is being defined.
INDEX signifies that a motorized indexable camera is being defined.
CART signifies that the sensor location is to be defined in cartesian coordinates.
POL signifies that the sensor location is to be defined in polar coordinates.
VEC signifies that the sensor location is to be defined in unit vectors.
dx,dy,dz is the distance in X, Y, and Z, between the sensor's axis orientation point, and the sensor reference point in machine coordinates.
ti,tj,tk is the unit vector of the sensor mount socket's axis.
i,j,k is the unit vector of the optical axis.
tilt is the sensor's angle of tilt with respect to the fully extended position where the tilt angle is zero. The tilt angle is always positive.
rot is the sensor's angle of rotation with respect to the positive X axis. Apply the right hand rule for sign convention.
focal is the focal distance in millimeters.
mag is the magnification of the lens in decimal form.
apert is the aperture setting.
Note: The sensor definition is repeated with a new label if any of the parameters are adjustable. In the following example, a camera with a zoom lens requires a new

label to be associated with each zoom position.
S(zoom_10x) = SNSDEF / VIDEO, FIXED, vec, 0,0,-1, 0,0,-1, 20, 10.0, 5.6
S(zoom_20x) = SNSDEF / VIDEO, FIXED, vec, 0,0,-1, 0,0,-1, 20, 20.0, 5.6
A camera with a motorized tiltable optical axis would require a label to be associated with each tilt position:
S(tilt_45) = SNSDEF / VIDEO, INDEX, vec,.707,0,-.707, 0,0,-1, 20, 20.0, 5.6
Additional VIDEO sensor settings defined with other DMIS statements: DMIS_statement Sensor_setting LITDEF Video lighting WINDEF Video viewing windows FILDEF Video filters ALGDEF Video algorithms These additional sensor settings, along with scaling, automatic focus, and intensity, are activated with the SNSET statement. The sensor is selected with the SNSLCT statement prior to taking a measurement with the MEAS statement. For example:
S(camera_10x) = SNSDEF / VIDEO, INDEX, vec, 0,0,-1, 0,0,-1, 20, 10.0, 5.6
VW(box_1) = WINDEF/... . VL(bright) = LITDEF/....
SNSET / VW(box_1), VL(bright), .75, FOCUSY
SNSLCT / S(camera_10x)
MEAS / CIRCLE, F(sm_hole),8
PTMEAS /...
SNSDEF

SNSDEF

Function

Default None.

Input S(label) = SNSDEF / LASER, var1, var2

Output None.

Where

var1 FIXED or: INDEX

var2 POL, tilt, rot, dx,dy,dz,si,sj,sk,length,diam,scale,range

S(label) is an alphanumeric label assigned to the sensor, and is up to 10 characters in length.

LASER signifies that a laser sensor is being defined.

FIXED signifies that a fixed laser is being defined.

INDEX signifies that an indexable laser is being defined.

CART signifies that the sensor location is to be defined in cartesian coordinates.

POL signifies that the sensor location is to be defined in polar coordinates.

tilt is the sensor's angle of tilt with respect to the fully extended position where the tilt angle is zero. The tilt angle is always positive.

rot is the sensor's angle of rotation with respect to the positive X axis. Apply the right hand rule for sign convention.

dx,dy,dz is the distance in X, Y and Z, between the sensor's axis orientation point and the sensor**'s** focal point in machine coordinates when the tilt and rot angles are zero. Typically it is (0.0, 0.0, (-)sensor length)

si,sj,sk is the unit vector of the probing direction. Typically it is (0.0, 0.0, (-)1.0).

length is the length of the sensor. It is updated during calibration.

diam is the diameter of the sensor. It is updated during calibration.

scale is the scale value of the sensor. Represents number of laser units per unit distance. Updated during calibration.

range is the measurement range for the sensor in the same linear distance units used to compute the scale.

Note:

SNSDIR

Function This command sets the angles for the rotating head for video and non-contact probes.

Default None

Input SNSDIR/var1,var2

Output None

Note: To be updated.

SNSET

Function Specifies and activates sensor settings used on a DME.

Default DME internally programmed values, except for DEPTH. SNSET/DEPTH,0

Input SNSET / var1

Output None.

Where

var1 APPRCH, dist1 or: RETRCT, dist1 or: SEARCH, dist1 or: CLRSRF, var2 or: DEPTH, var2 or: SCAN, var3

var2 F(label) var5 or: FA(label) var5 or: DAT(x) var5 or: dist2

var3 THRESH,<val> or: TIME,<val>

APPRCH signifies the approach distance. The distance that the probe will begin its measurement taking sequence to the feature being measured.

RETRCT signifies the retract distance. The distance that a probe will retract after a triggering measurement.

SEARCH signifies the search distance. The distance that a probe will continue its measurement taking sequence beyond the APPRCH distance. If no point is found, the DME will retract to the point where the search was initiated and flag an error. SNSET

CLRSRF signifies the clearance distance away from the feature for safe moves. Note that when F(label1), FA(label1), or DAT(x) are used, that they are previously defined/measured planes that are parallel to the plane in which the feature to be measured, lies. Also, CLRSRF and DEPTH are perpendicular to APPRCH, RETRCT, and SEARCH.

DEPTH signifies the distance the probe will penetrate into the feature when measuring holes, slots, etc. Note that when F(label1), FA(label1), or DAT(x) are used, that they are previously defined/measured planes that are parallel to the plane in which the feature to be measured, lies. Also, CLRSRF and DEPTH are perpendicular to APPRCH, RETRCT, and SEARCH.

SCAN signifies that scanning parameters are to be set.

dist1 is the distance value assigned.

dist2 is the distance value assigned from the nominal feature and is to be measured along its vector. (see note)

F(label) is a previously defined feature (plane) used as a clearance plane, or a depth measuring plane.

FA(label) is a previously measured feature (plane) used as a clearance plane, or a depth measuring plane.

DAT(x) is a previously measured and assigned datum (plane) used as a clearance plane, or a depth measuring plane.

Note: When F(label), FA(label), or DAT(x) are used with CLRSRF and DEPTH, they are previously defined/measured planes that are parallel to the plane in which the feature to be measured, lies. Also, CLRSRF and DEPTH are perpendicular to APPRCH, RETRCT, and SEARCH. When utilizing CLRSRF and DEPTH on features such as cones, cylinders, GSURF, GCURVE and spheres it is important to note that the "dist" is relative to an imaginary plane tangent to one of the afore mentioned features. Caution should be exercised when dealing with piercing features that do not penetrate normal to the surface. In these cases, GOTO points should be used to guarantee access. DEPTH is used by a CMM when executing a measurement sequence. For features such as lines, circles and ellipses, the nominal definition specifies the surface within which the feature lies. In AUTO mode, the probe will penetrate this surface, in a direction perpendicular to the working plane, by the amount specified in SNSET/DEPTH before the CMM executes its automatic measurement routine. In PROG mode, the PTMEAS x,y,z value will be offset by the DEPTH value, in a direction perpendicular to the working plane. The DEPTH distance positive or negative, and the signed value is added to the programmed measurement points. Apply the right hand rule for sign conventions. The characterization file identifies which sensor settings are supported.

SNSLCT

Function Selects the sensor(s) to be used for measurement.

Default None.

Input SNSLCT / var1 var3

Output None.

Where

var1 S(label1) or: S(label1), var2

var2 S(label2) or: S(label2), var1

var3 ,CR(label) or: does not exist

S(label1) is a previously defined sensor.

S(label2) is a previously defined sensor(s) used for machine vision systems when more than one sensor is used to take measurements.

CR(label) signifies a previously defined carriage when more than one exists. Note that this is optional for systems limited to one active part coordinate system. This label provides the required association between part coordinate systems and carriages when programming parallel or dual systems.

Note: The number of sensors that selected simultaneously is identified in the characterization file. When programming parallel or dual systems, the association between sensors and carriages is provided with this statement.

SPLINE

Function Used to define the spline id.

Default SPLINE/1

Input SPLINE/var1

Output None.

Where

var1 is an integer ID for the spline.

Note: The SPLINE command gives the spline ID number. SPLINE commands do not cause an interruption in the movement of the machine. SPLINE is active when scanning is on. The spline ID is written along with the x, y and z coordinates to the . ACT and . NOM files to indicate which data points belong to which spline.

TECOMP

Function Causes temperature compensation to be on or off at the DME.

Default TECOMP/OFF

Input TECOMP/var1

Output TECOMP/var1

Where

var1 ON or: OFF

ON signifies temperature compensation to be turned on at the DME.

OFF signifies temperature compensation to be turned off at the DME.

Note: The TECOMP statement is passed on to the output file when executed.

TEXT

Function Specifies various forms of text to be sent to the operator and/or, the output file.

Default None.

Input TEXT/var1, 'text'

Output TEXT/var4, 'text'

Where

var1 OPER or: OUTFIL or: MAN or: QUERY,(label),length, var2, var3

var2 A or: N or: AN

var3 L or: R

var4 OUTFIL or: RES,(label),length, var2, var3

OPER signifies that the message will be printed on the screen during program execution. The message any printable ASCII character.

OUTFIL signifies that the message will be inserted into the output file. The message is inserted in the output at the point at which the TEXT/OUTFIL statement occurred in the input program.

MAN signifies that the message will be output to the operator, but only if the DME is in manual mode (if MODE/MAN has been issued or if the DME has automatically defaulted to manual during a GOTARG, MEAS, or CALIB).

QUERY signifies that a response is required from the operator. The response will be output into the output file with the same label as the query. Note that the output format for the TEXT/QUERY statement is TEXT/RES. (label) is an alphanumeric label of up to 10 characters in length.

length is the field length allowed for the response. The field length is the same for both the TEXT/QUERY and the corresponding TEXT/RES statements.

A signifies that the response should consist of all alpha characters.

N signifies that the response should consist of character representations of real numeric data.

AN signifies that the response will be any printable ASCII characters.

L signifies a left justified response.

R signifies a right justified response.

RES signifies the response to the QUERY.

'text' is the text string consisting of printable ASCII characters, and must be started and terminated with an apostrophe. $ signifies that the text extends to the following line.

Note: Use two apostrophes, one before the one required, when an apostrophe is required within a text string. or example: TEXT/OPER,'Use Paul''s setup instructions from the last job. 'This results in the following message sent to the screen: Use Paul's setup instructions from the last job. The TEXT statement is passed on to the output file when executed.

THLDEF

Function Defines an automatic tool or sensor holder/changer in terms of the sensors it carries.

Default None.

Input TH(label) = THLDEF / S(label1), n, var1

Output None.

Where

var1 S(labeln),n or: S(labeln),n, S(labeln),n,...

TH(label) is the name assigned to the sensor holder being defined.

S(label1) is the first sensor name associated with the sensor holder being defined.

S(labeln) is the nth sensor name in a list that is associated with the sensor holder being defined.

n is a one to two integer character identifying the nth position of the sensor in the holder.

Note: This statement associates a set of sensors to a sensor holder. There are no restrictions on the type or number of sensors in the sensor holder, unless they are identified in the characterization file. Although the label of the sensor is not currently used, i.e. , referenced later in the program, it is included in the format to allow a standard means of relating sensors in the DMIS environment. The

SNSLCT statement will cause a sensor change when the selected sensor is not already in place. Safe positioning prior to an automatic sensor change is encouraged to avoid sensor collision.

TOL/ANGL

Function Specifies an angular tolerance and assigns to it a label.
Default None.
Input T(label)=TOL/ANGL,lotol,uptol
Output TA(label)=TOL/ANGL,dev,var1
Where
var1 INTOL or: OUTOL
label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.
ANGL signifies that the tolerance is an angular tolerance.
lotol is the signed lower tolerance value applied to the angle.
uptol is the signed upper tolerance value applied to the angle.
dev is the deviation - the arithmetic difference between the actual value and the nominal value. It is positive when the nominal is less than the actual, and negative when the nominal is greater than the actual.
INTOL signifies the actual is within tolerance.
OUTOL signifies the actual is out of tolerance.
Note: If an angular tolerance such as 30 degrees +.5 degrees, -.4 degrees is given, then uptol = +.5 and lotol = -.4. Unsigned lotol and uptol values are assumed positive. An angular tolerance applied to a cone.

TOL/ANGLB

Function Specifies an angle and a tolerance, and assigns it to a label.
Default None.
Input T(label)=TOL/ANGLB,ang,lotol,uptol
Output TA(label)=TOL/ANGLB,ang,var1
Where
var1 INTOL or: OUTOL
label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.
ANGLB signifies the value and tolerance are applied to the angle between two features.
ang is the nominal value of an angle, or the measured angle.
lotol is the signed lower tolerance value assigned to the angle.
uptol is the signed upper tolerance value assigned to the angle.
INTOL signifies the actual is within tolerance.
OUTOL signifies the actual is out of tolerance.
Note: Defines the nominal angle to be used between two features that reduce to lines or planes i,e. , cylinders and cones reduce to their centerlines, slots reduce to their centerplane. If an angle and tolerance such as 30 degrees +.5 degrees, -.4 degrees

is given, then uptol = +.5 and lotol = -.4. Unsigned lotol and uptol values are assumed positive.

TOL/ANGLR

Function Specifies an angularity tolerance and assigns to it a label.
Default None.
Input T(label)=TOL/ANGLR,ang,tolzon,var2,var3
Output TA(label)=TOL/ANGLR,tolzon,var1,var2,lim,var3
Where
var1 INTOL or: OUTOL
var2 MMC or: LMC or: RFS
var3 DAT(x),var2 or: F (label2) or: FA(label)
label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.
ANGLR signifies angularity.
ang is the nominal value of an angle in the current primary datum plane relative to the specified datum, or the actual angle.
tolzon is the width of the tolerance zone defined by two parallel lines within which all points of the feature must lie, or the distance between parallel planes within which the center plane of the feature must lie, or the diameter of a cylindrical tolerance zone within which the axis of the feature must lie.
INTOL signifies the actual is within tolerance.
OUTOL signifies the actual is out of tolerance.
MMC signifies that maximum material condition is applied.
LMC signifies that least material condition is applied.
RFS signifies regardless of feature size.
F(label2) is the name of a feature to be used as a reference.
FA(label) is the name of the feature actual to be used as a reference.
DAT(x) is the datum to be used, or used as a reference.
lim is the sum of the tolerance plus the gain from MMC or LMC.
Note: The tolzon in the Output_is the zone in which the actual feature lies. MMC and LMC applied to a center plane or axis.

TOL/CIRLTY

Function Specifies a circularity tolerance and assigns to it a label.
Default None.
Input T(label)=TOL/CIRLTY,tolzon
Output TA(label)=TOL/CIRLTY,tolzon,var1
Where
var1 INTOL OUTOL
label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.
CIRLTY signifies circularity.

tolzon is the width of the tolerance zone bounded by two concentric circles within which elements of the surface of the feature must lie, or actually lies.

INTOL signifies the actual is within tolerance.

OUTOL signifies the actual is out of tolerance.

Note: Regardless of Feature Size, RFS is assumed. A Circularity tolerance applied to an arc, circle, cone and sphere. Circularity when applied to cone refers to Conicity and when applied to sphere refers to spherecity.

TOL/CONCEN

Function Specifies a concentricity tolerance and assigns it to a label.

Default None.

Input T(label)=TOL/CONCEN,tolzon,var2

Output TA(label)=TOL/CONCEN,tolzon,var1,var2

Where

var1 INTOL or: OUTOL

var2 DAT(x) or: F (label2) or: FA(label)

label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.

CONCEN signifies concentricity.

tolzon is the diameter of the cylindrical tolerance zone, or the actual measured zone in which the axis of the feature lies.

INTOL signifies the actual is within tolerance.

OUTOL signifies the actual is out of tolerance.

F(label2) is the name of a feature is to be used as a reference.

FA(label) is the name of a feature actual is to be used as a reference.

DAT(x) is the datum to be used as a reference.

Note: A concentricity tolerance specifies a cylindrical tolerance zone around a datum axis within which the feature axis must lie. It applies on an RFS basis only and applied to an Arc, Circle, Cone or a Cylinder.

TOL/CORTOL

Function Specifies bi-directional positional tolerancing of features in cartesian, or polar coordinates, and assigns to it a label.

Default None.

Input T(label)=TOL/CORTOL,var1,lotol,uptol

Output TA(label)=TOL/CORTOL,var1,dev,var2

Where

var1 XAXIS or: YAXIS or: ZAXIS or: RADIUS or: ANGLE

var2 INTOL or: OUTOL

label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.

CORTOL signifies bi-directional positioning tolerancing.

XAXIS signifies that the rectangular coordinate method is to be used to tolerance the position along the X axis.

YAXIS signifies that the rectangular coordinate method is to be used to tolerance the position along the Y axis.

ZAXIS signifies that the rectangular coordinate method is to be used to tolerance the position along the Z axis.

RADIUS signifies that the polar coordinate method is used to tolerance the radial position.

ANGLE signifies the polar coordinate method is used to tolerance the angular position.

INTOL signifies the actual is within tolerance.

OUTOL signifies the actual is out of tolerance.

lotol is the signed lower tolerance value.

uptol is the signed upper tolerance value.

dev is the deviation from the nominal value.

Note: When using the rectangular coordinate method, the tolerance zone square or rectangular. Note that the feature(s) being toleranced with this method must be defined with cartesian coordinates. Likewise, when using the polar coordinate method, the feature(s) being toleranced with this method must be defined with polar coordinates. The positional tolerance values of the polar coordinate method represent distances between two concentric arc boundaries and two parallel planes , respectively, equally disposed about the true position. It applied to all bounded features.

TOL/CRNOUT

Function Specifies a circular runout tolerance and assigns to it a label.

Default None.

Input T(label)=TOL/CRNOUT,tolzon,DAT(x),var2

Output TA(label)=TOL/CRNOUT,tolzon,var1,DAT(x),var2

Where

var1 INTOL or: OUTOL

var2 ,DAT(x) or: does not exist

label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.

CRNOUT signifies circular runout.

tolzon is the width of the tolerance zone within which a single circular element i.e. , circular cross section, must lie.

DAT(x) is the datum to be used as an axis, and in the form of (x-x) for constructed compound datums.

INTOL signifies the actual is within tolerance.

OUTOL signifies the actual is out of tolerance.

Note: Circular runout is applied to circular features. The tolerance is applied independently at any circular cross section as the part is rotated 360 degrees. When applied to features with an axis, circular runout controls the cumulative variations of the circularity and coaxiality. When applied to a plane feature which is perpendicular to the datum axis, circular runout controls wobble.

TOL/CYLCTY

Function Specifies a cylindricity tolerance and assigns to it a label.

Default None.

Input T(label)=TOL/CYLCTY,tolzon

Output TA(label)=TOL/CYLCTY,tolzon,var1

Where

var1 INTOL or: OUTOL

label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.

CYLCTY signifies cylindricity.

tolzon is the width of the tolerance zone bounded by two concentric cylinders within which elements of the surface of the feature must lie, or actually lies.

INTOL signifies the actual is within tolerance.

OUTOL signifies the actual is out of tolerance.

Note: Regardless of Feature Size, RFS is assumed. Tolerance cylindricity applied to cylinders only.

TOL/DIAM

Function Specifies a diameter tolerance and assigns to it a label.

Default None.

Input T(label)=TOL/DIAM,lotol,uptol

Output TA(label)=TOL/DIAM,dev,var1

Where

var1 INTOL OUTOL

label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.

DIAM signifies that the tolerance is a diameter tolerance.

lotol is the signed lower tolerance value applied to the diam.

uptol is the signed upper tolerance value applied to the diam.

dev is the deviation - the arithmetic difference between the actual value and the nominal value. It is positive when the nominal is less than the actual, and negative when the nominal is greater than the actual.

INTOL signifies the actual is within tolerance.

OUTOL signifies the actual is out of tolerance.

Note: If a diameter tolerance such as 5 +1, -2 is given, then uptol = +1 and lotol = -2. Unsigned lotol and uptol values are assumed positive. A diameter tolerance applied to an arc, circle, cylinder, sphere or ellipse. In the case of an ellipse, the diameter specified is that referenced in the feature definition. If both diameters are to be toleranced, then two feature definitions will be required.

TOL/DISTB

Function Specifies a distance and a tolerance, and assigns it to a label.

Default None.

Input T(label)=TOL/DISTB,var2,var3,var4
Output TA(label)=TOL/DISTB,var1,var2,var3,var4
Where
var1 INTOL or: OUTOL
var2 NOMINL,dist,lotol,uptol or: LIMIT,lolimt,uplimt
var3 XAXIS or: YAXIS or: ZAXIS or: PT2PT
var4 AVG or: MAX or: MIN or: does not exist
label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.
DISTB signifies the value and tolerance are applied to the distance between two features.
INTOL signifies the actual is within tolerance.
OUTOL signifies the actual is out of tolerance.
NOMINL signifies a nominal distance with lower and upper tolerance.
LIMIT signifies a distance as a lower and upper limit.
dist is the nominal, or the actual measured distance.
lotol is the signed lower tolerance assigned to the nominal distance.
uptol is the signed upper tolerance assigned to the nominal distance.
lolimt is the lower limit tolerance, or actual low limit.
uplimt is the upper limit tolerance, or actual upper limit.
XAXIS signifies that the distance between is along the X axis.
YAXIS signifies that the distance between is along the Y axis.
ZAXIS signifies that the distance between is along the Z axis.
PT2PT signifies that the distance between is point to point, or feature to feature.
AVG signifies average or mean distance between two features.
MAX signifies maximum distance between two features.
MIN signifies minimum distance between two features.
Note: If a distance and tolerance such as 5 +1,-2 is given, then uptol = +1 and lotol = -2. Unsigned lotol and uptol values are assumed positive. In the output format, the lotol and uptol values define the zone in which the actual feature lies. When tolerancing the distance between two points, of features that resolve to points, var4 is not required.

TOL/FLAT

Function Specifies a flatness tolerance and assigns to it a label.
Default None.
Input T(label)=TOL/FLAT,tolzon
Output TA(label)=TOL/FLAT,tolzon,var1
Where
var1 INTOL or: OUTOL
label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.
FLAT signifies flatness.
tolzon is the width of the tolerance zone defined by two parallel planes within which the surface of the feature must lie, or actually lies.

INTOL signifies the actual is within tolerance.
OUTOL signifies the actual is out of tolerance.
Note: Regardless of Feature Size, RFS is assumed. Flatness applied to arc, circle, plane and elongated hole features.

TOL/LENGTH

Function Specifies a length(linear) tolerance and assigns to it a label.
Default None.
Input T(label)=TOL/LENGTH,lotol,uptol
Output TA(label)=TOL/LENGTH,dev,var1
Where
var1 INTOL or: OUTOL
label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.
RAD signifies that the tolerance is a radial tolerance.
lotol is the signed lower tolerance value applied to the radius.
uptol is the signed upper tolerance value applied to the radius.
dev is the deviation - the arithmetic difference between the actual value and the nominal value. It is positive when the nominal is less than the actual, and negative when the nominal is greater than the actual.
INTOL signifies the actual is within tolerance.
OUTOL signifies the actual is out of tolerance.
Note:

TOL/PARLEL

Function specifies a parallelism tolerance and assigns to it a label.
Default None.
Input T(label)=TOL/PARLEL,tolzon,var2,var3
Output TA(label)=TOL/PARLEL,tolzon,var1,var2,lim,var3
Where
var1 INTOL or: OUTOL
var2 MMC or: LMC or: RFS
var3 DAT(x),var2 or: F (label2) or: FA(label)
label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.
PARLEL signifies parallelism.
tolzon is the width of the tolerance zone defined by two parallel lines within which all points of the feature must lie, or the distance between parallel planes within which the center plane of the feature must lie, or the diameter of a cylindrical tolerance zone within which the axis of the feature must lie.
INTOL signifies the actual is within tolerance.
OUTOL signifies the actual is out of tolerance.
MMC signifies that maximum material condition is applied.

LMC signifies that least material condition is applied.
RFS signifies regardless of feature size.
F(label2) is the name of a feature to be used as a reference.
FA(label) is the name of a feature actual to be used as a reference.
DAT(x) is the datum to be used as a reference.
lim is the sum of the tolerance plus the gain from MMC or LMC.
Note: In the output format, the tolzon is the zone in which the actual feature lies.

TOL/PERP

Function Specifies a perpendicularity tolerance and assigns to it a label.
Default None.
Input T(label)=TOL/PERP,tolzon,var2,var3
Output TA(label)=TOL/PERP,tolzon,var1,var2,lim,var3
Where
var1 INTOL or: OUTOL
var2 MMC or: LMC or: RFS
var3 DAT(x),var2 or: F (label2) or: FA(label)
label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.
PERP signifies perpendicularity tolerance.
tolzon is the width of the tolerance zone defined by two parallel lines within which all points of the feature must lie, or the distance between parallel planes within which the center plane of the feature must lie, or the diameter of a cylindrical tolerance zone within which the axis of the feature must lie.
INTOL signifies the actual is within tolerance.
OUTOL signifies the actual is out of tolerance.
MMC signifies that maximum material condition is applied.
LMC signifies that least material condition is applied.
RFS signifies regardless of feature size.
F(label2) is the name of a feature to be used as a reference.
FA(label) is the name of a feature actual to be used as a reference.
DAT(x) is the datum to be used as a reference.
lim is the sum of the tolerance plus the gain from MMC or LMC.
Note: In the output format, the tolzon is the zone in which the actual measured feature lies.

TOL/POS

Function Specifies a position tolerance and assigns to it a label.
Default None.
Input T(label)=TOL/POS,var1,tolzon,var2, var3 var3 var3
Output TA(label)=TOL/POS,var1,tolzon,var5, var4 var3 var3 var3
Where
var1 2D or: 3D

var2 MMC or: LMC or: RFS

var3 ,DAT(x),var2 or: does not exist

var4 MMC,lim or: LMC,lim or: RFS

var5 INTOL or: OUTOL

label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.

POS signifies a position tolerance.

2D signifies a circular tolerance zone evaluated in a two dimensional plane which is parallel to the current WKPLAN.

3D signifies a cylindrical tolerance zone whose axis is perpendicular to the current WKPLAN. Note that the feature cylinder being toleranced must be bounded.

tolzon is the diameter of the circular or cylindrical tolerance zone within which the feature center point or axis must lie, or actually lies.

INTOL signifies the actual is within tolerance.

OUTOL signifies the actual is out of tolerance.

MMC signifies that maximum material condition is applied.

LMC signifies that least material condition is applied.

RFS signifies regardless of feature size.

DAT(x) is the datum to be used as a reference.

lim is the sum of the tolerance plus the gain from MMC or LMC.

Note: The material condition modifiers in var2 signify whether MMC or LMC is applied to the feature. The material condition modifiers when used within var3 signify to which datum(s), if any, MMC or LMC is applied. POS2D applied to a circle and POS3D to a cylinder respectively.

TOL/PROFL

Function Specifies a profile of a line (curve) tolerance and assigns to it a label.

Default None.

Input T(label)=TOL/PROFL,lotol,uptol var2 var2 var2

Output TA(label)=TOL/PROFL,lotol,uptol,var1 var2 var2 var2

Where

var1 INTOL or: OUTOL var2 ,DAT(x) or: does not exist

label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.

PROFL signifies that the tolerance is a profile tolerance.

lotol is the signed lower tolerance value which lies to the inside of the part.

uptol is the signed upper tolerance value which lies to the outside of the part.

INTOL signifies the actual is within tolerance.

OUTOL signifies the actual is out of tolerance.

DAT(x) is datum is to be used as a reference.

Note: The profile of a line tolerance specifies a tolerance zone which is a two dimensional band, extending along the length of the feature. If the tolerance is unbounded, it is applied along the length of the feature. The tolerance may be bounded when it is desired to apply it to only a portion of the feature. Regardless of Feature Size is assumed. The output of lotol and uptol define the tolerance zone within which the feature actually lies.

TOL/PROFS

Function Specifies a profile of a surface tolerance and assigns to it a label.

Default None.

Input T(label)=TOL/PROFS,lotol,uptol var2 var2 var2

Output TA(label)=TOL/PROFS,lotol,uptol,var1 var2 var2 var2

Where

var1 INTOL or: OUTOL var2 ,DAT(x) or: does not exist

label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.

PROFS signifies that the tolerance is an profile tolerance.

lotol is the signed lower tolerance value which lies to the inside of the part.

uptol is the signed upper tolerance value which lies to the outside of the part.

INTOL signifies the actual is within tolerance.

OUTOL signifies the actual is out of tolerance.

DAT(x) is the datum to be used as a reference.

Note: The profile of a surface tolerance specifies a three dimensional tolerance zone which extends along the length and width or circumference of the feature. As with the profile of a line tolerance, it may be unbounded or bounded. Regardless of Feature Size is assumed.

TOL/RAD

Function Specifies a radial tolerance and assigns to it a label.

Default None.

Input T(label)=TOL/RAD,lotol,uptol

Output TA(label)=TOL/RAD,dev,var1

Where

var1 INTOL or: OUTOL

label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.

RAD signifies that the tolerance is a radial tolerance.

lotol is the signed lower tolerance value applied to the radius.

uptol is the signed upper tolerance value applied to the radius.

dev is the deviation - the arithmetic difference between the actual value and the nominal value. It is positive when the nominal is less than the actual, and negative when the nominal is greater than the actual.

INTOL signifies the actual is within tolerance.

OUTOL signifies the actual is out of tolerance.

Note: If a radius tolerance such as 5 +1, -2 is given, then uptol = +1 and lotol = -2. Unsigned lotol and uptol values are assumed positive. A radial tolerance applied to an arc, circle, cylinder, sphere, and ellipse. In the case of an ellipse, the diameter specified is that referenced in the feature definition. If both diameters are to be toleranced, then two feature definitions will be required.

TOL/STRGHT

Function Specifies a straightness tolerance and assigns to it a label.
Default None.
Input T(label)=TOL/STRGHT,tolzon,var2
Output TA(label)=TOL/STRGHT,tolzon,var1,var2,lim
Where
var1 INTOL or: OUTOL
var2 MMC or: LMC or: RFS
label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.
STRGHT signifies straightness.
tolzon is the width of the tolerance zone defined by two parallel lines within which all points of the feature must lie, or the distance between parallel planes within which the center plane of the feature must lie, or the diameter of a cylindrical tolerance zone within which the axis of the feature must lie, or actually lies.
INTOL signifies the actual is within tolerance.
OUTOL signifies the actual is out of tolerance.
MMC signifies that maximum material condition is applied.
LMC signifies that least material condition is applied.
RFS signifies regardless of feature size.
lim is the sum of the tolerance plus the gain from MMC or LMC.
Note: MMC, or LMC applied to a center plane or axis. Straightness is applied to a line or a cylinder.

TOL/TRNOUT

Function Specifies a total runout tolerance and assigns to it a label.
Default None.
Input T(label)=TOL/TRNOUT,tolzon,DAT(x) var2 var2
Output TA(label)=TOL/TRNOUT,tolzon,var1,DAT(x) var2 var2
Where
var1 INTOL or: OUTOL
var2 ,DAT(x) or: does not exist
label is an alphanumeric label assigned to the tolerance, and is up to 10 characters in length.
TRNOUT signifies total runout.
tolzon is the width of the tolerance zone defined by two parallel lines within which all points of the feature must lie, or the distance between parallel planes within which the center plane of the feature must lie, or the diameter of a cylindrical tolerance zone within which the axis of the feature must lie.
DAT(x) is the datum to be used as an axis and in the form of (x-x) for constructed

compound datums.

INTOL signifies the actual is within tolerance.

OUTOL signifies the actual is out of tolerance.

Note: Total runout provides composite control of an entire feature. The tolerance is applied simultaneously to all circular and profile measuring positions as the part is rotated 360 degrees. Where applied to surfaces constructed around a datum axis, total runout is used to control cumulative variations of circularity, straightness, coaxiality, angularity, taper, and profile of a surface. Where applied to surfaces constructed at right angles to a datum axis, total runout controls cumulative variations of perpendicularity (to detect wobble) and flatness (to detect concavity or convexity).

TOOLDF

Function Defines a tool used on a manufacturing device.

Default None.

Input TL(label) = TOOLDF / MD(label), 'text'

Output None, (activated with the REPORT statement)

Where

TL(label) is the name assigned to this definition.

MD(label) is the name of the manufacturing device that this tool definition is associated with.

'text' is an alphanumeric field, enclosed with apostrophes that identifies the manufacturing tool.

Note: This statement is associated with the CUTCOM and MFGDEF statements for applications to adjust the manufacturing process based on inspection results. The TOOLDF statement is passed on to the output file when executed.

TRANS

Function Translates a part coordinate system along an axis, and assigns to it a label.

Default None.

Input D(label) = TRANS / var1, var2,var3 ,var4

Output D(label) = TRANS / var1,var2,var3,var4

Where

var1 XORIG or: YORIG or: ZORIG

var2 value or: F(label1) or: FA(label1) or: DAT(x) or: PRBRAD, actual probe radius or: -PRBRAD, actual probe radius

var3 ,var1, var2 or: does not exist

var4 ,CR(label) var4 var4 .

D(label) is the name assigned to the new part coordinate system.

XORIG signifies that the coordinate system origin is to be translated on the X axis, if value is given. Signifies the X coordinate of the origin is to be translated to the X coordinate of the following feature, if F(label1), FA(label1), or DAT(x) is given.

YORIG signifies that the coordinate system origin is to be translated on the Y axis, if value is given. Signifies the Y coordinate of the origin is to be translated to the Y coordinate of the following feature, if F(label1), FA(label1), or DAT(x) is given.

ZORIG signifies that the coordinate system origin is to be translated on the Z axis, if value is given. Signifies the Z coordinate of the origin is to be translated to the Z coordinate of the following feature, if F(label1), FA(label1), or DAT(x) is given.

value is the distance the coordinate system origin is to be translated. A positive value is a translation in the positive direction along the axis and a negative value is a translation in the negative direction along the axis.

F(label1) is the nominal feature used to establish the origin.

FA(label1) is the measured feature used to establish the origin.

PRBRAD signifies that the preceding origin component will be translated one half the probe diameter in the positive direction. The probe diameter is established by the DME during calibration. **-**

PRBRAD signifies that the preceding origin component will be translated one half the probe diameter in the negative direction. The probe diameter is established by the DME during calibration.

actual probe radius signifies the actual probe radius.

CR(label) signifies a previously defined carriage when more than one exists. Note that this is optional for systems limited to one active part coordinate system. This label provides the required association between part coordinate systems and carriages when programming parallel or dual systems.

Note: There are some implied limitations when translating to features. For example, it is not possible to translate the origin along an axis to a line that is parallel to the axis. Additionally, a limit of one translation along each axis is allowed for each TRANS statement. A SAVE statement must be issued prior to the TRANS statement if the current part coordinate system is to be used again with the RECALL statement. The new part coordinate system is activated when the TRANS statement is executed. TRANS is passed through to the output file at the time it is executed.

UNITS

Function Specifies the units that will be active throughout the program.

Default None.

Input UNITS / var1, var2 var3

Output UNITS / var1, var2 var3

Where

var1 MM or: CM or: M or: INCH or: FEET

var2 ANGDEC or: ANGDMS or: ANGRAD

var3 ,TEMPF or: ,TEMPC or: does not exist

MM signifies distance in millimeters.

CM signifies distance in centimeters.

M signifies distance in meters.

INCH signifies distance in inches.

FEET signifies distance in feet.

ANGDEC signifies angles in decimal form.

ANGRAD signifies angles in radian form.

TEMPF signifies temperature in degrees Fahrenheit.

TEMPC signifies temperature in degrees Centigrade.

Note: The UNITS statement issued only once in a program. Unless otherwise specified in the syntax of a statement, the UNITS statement determines the units being used for a program. The UNITS statement is passed on to the output file when executed.

UPDATE

Function Update a feature's characteristic based on reference feature or feature actual or actual vector or just reverse the features vector.

Default None.

Input UPDATE/FA(label),var1

Output None.

Where

var1 F(label) or: FA(label) or: REVERS or: i,j,k

F(label) Nominal feature whose vector is used.

FA(label) Actual feature whose vector is used.

REVERS Specifies that the feature's vector should be reversed.

i,j,k Actual vector direction.

Note:

VALUE

Function Sets a variable equal to a measurement result.

Default None

Input var1=VALUE/var2,var3

Output None

Where

var1 var2 var3 are the names of the previously declared variables to which the values are assigned. FA(label) or: TA(label)if var2 is: FA(label) then SIZE if var2 is: TA(label) then INTOL or: OUTOL or: ACT or: DEV or: AMT

VALUE signifies that a measurement value is to be assigned to a variable.

FA(label) is the feature whose measurement result will be used.

TA(label) is the tolerance whose evaluation result will be used.

SIZE signifies that the actual size of the feature (diameter, radius, or angle) will be assigned to the variable.

ACT signifies that the actual (nominal plus material condition bonus) size of the tolerance zone (tolzon) will be assigned to the variable.

DEV signifies that the deviation (actual-nominal) will be assigned to the variable.

AMT signifies that the amount out of tolerance (actual-(nominal + tolerance)) will be assigned to the variable.

INTOL signifies that the value will be set to 1 if tolerance is evaluated to be within specification. Otherwise it is set to 0.

OUTOL signifies that the value will be set to 1 if the tolerance is evaluated to be out of specification. Otherwise it is set to 0.

Note: Variables must be declared prior to this statement with the DECL command. Value will evaluate the tolerance statement(s) in the last EVAL and/or OUTPUT command. For example: EVAL/F(label),T(label) VAR = VALUE/TA(label),ACT

WKPLAN

Function Used to explicitly declare or change a working plane.

Default None.

Input WKPLAN/var1

Output None.

Where

var1 XYPLAN or: YZPLAN or: ZXPLAN

XYPLAN signifies the XY plane of the current part coordinate system is the working plane.

YZPLAN signifies the YZ plane of the part current coordinate system is the working plane.

ZXPLAN signifies the ZX plane of the current part coordinate system is the working plane.

Note: This plane will be active for the purpose of determining polar coordinates, offset angles, measuring circles, the true position plane, etc. It is in effect until a new working plane is named.