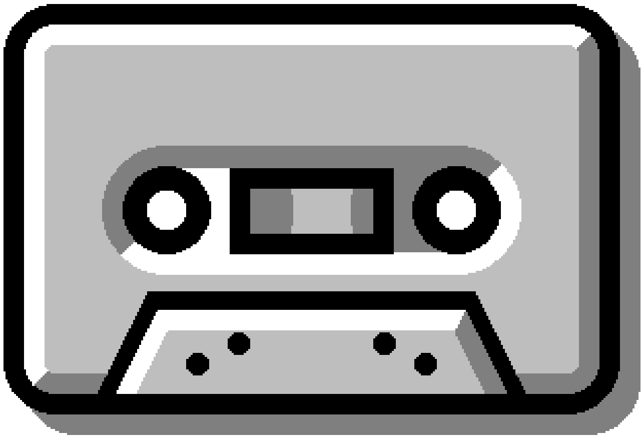

The Unofficial SmartCAM® Macro Command Dictionary

Compiled by Fred Lauzus

The Unofficial **SmartCAM**® Macro Command Dictionary
Last updated November 29, 2000

Preface

Introduction
This “unofficial” dictionary was created in part because the official documentation shipped with SmartCAM contains numerous instances of missing, incomplete or inaccurate descriptions of the supported functions and commands. Also, the information in the official documentation is scattered throughout various sections making it difficult rapidly locate specific items from different sections. Organizing the information in a dictionary format provides a quick reference guide for the writing, editing and debugging of macro files. It is being freely shared in the spirit of cooperation and exchange of information between the remaining dedicated SmartCAM users.

Purpose
This dictionary is intended to supplement but not replace the various SmartCAM manuals. It is to serve as an updated macro reference source for SmartCAM macro programmers and attempts to address incorrect and incomplete entries regarding macros in the SmartCAM Customization Guide.

The commands added or amended from the SmartCAM customization Guide include but are not limited to:

AUTO_ABORT[ ]
FILTER_USE[ ]
FIXUP_51[ ]
FPATH_SET[ ]
GRP_FILT_ADD[ ]
GRP_FILT_REMOVE[ ]
HOTSAVE[ ]
INTEGER:
INTPT[ ]
MIRROR[ ]
PTOP[ ]
ROTATE[ ]
SYSTEM_UNITS[ ]
STRING:
WP_CHANGE_NAME[ ]

Compatibility
The functions and commands listed in this dictionary are for use with SmartCAM v11.x applications but includes some obsolete yet functional commands from previous versions to aid in debugging existing macros created in those older versions. This dictionary does not include any macro commands used prior to SmartCAM v6.x applications or v3.x advanced applications.

Disclaimer
This publication is not a product of and the author has no affiliation with SDRC. Although every attempt is made to insure the accuracy of this document, the author assumes no liability for any costs incurred by the use of this information.

Licenses and Trademarks
SmartCAM and CAM Connection are registered trademarks of Structural Dynamics Research Corporation; SmartCAM Advanced Fabrication, SmartCAM Advanced 3-D Machining, SmartCAM Advanced Production Milling, SmartCAM Advanced Turning, SmartCAM Advanced Wire EDM, SmartCAM Customization Toolkit, SmartCAM FreeForm Machining, SmartCAM Production Fabrication, SmartCAM Production Milling, SmartCAM Production Turning, AutoCAD SmartCAM Connection, CAMCAD Standard, CNC Process Model, and Edit Plus are trademarks of Structural Dynamics Research Corporation. All other trademarks or registered trademarks belong to their respective holders.

Conventions

The conventions in this dictionary attempt to mimic conventional dictionary formats but is applied to the SmartCAM macro language.

NAME
(Long Name), *type*. Description. {Applications}

Note: optional

NAME[FN=$$, ST=##]

FN = String value
ST = Number values
    Default: 0

NAME	The "Name", in upper case bold, is the short, cryptic name of the macro function or command as it is used within a macro.
(Long Name)	The "Long Name", in mixed case in parentheses, is the longer common language name of the macro function or command, typically referring to a menu command that may record it.
type	The "type", in lower case italic, Identifies if the item being described is a *function,* a *command* or *script*. A function will returns a value while a command performs an action. Script is used in demo mode to display menus, control panels and dialog boxes in the user interface.
Description	The "description" is a brief description of what the function, command or script does. For a more in depth descriptions of functionality you may need to refer to the "official" SmartCAM users manuals.
{Applications}	The "Applications", within the braces { }, is a list of SmartCAM application that the function or command is valid for. If no applications are listed of a particular function or command it can be assumed that the function or command is valid for all applications.
Note:	Optional information may be provided in addition to the description for the function or command.
NAME[FN=$$, ST=##]	Syntax structure and parameter list. This illustrates how the function or command should appear along with its parameters within the macro.

FN = String value
ST = Number values
Default: 0
	Usually there is a short explanation of each of the parameters from the parameter list as well as valid options and default values.

Designates the parameter requires a numeric value in the form of a number, numeric variable or mathematical formula.

$$ Designates the parameter requires a string value in the form of quoted alphanumeric characters, string variable or a string expression.

- A -

4AXPATH[ ]
(4 Axis Path), *command.* Creates the 5-axis polyline tool path necessary for a 4-axis machining operation. {Awedm}

4AXPATH[P1=$$, P2=$$, S1=$$, S2=$$,
    DS=##, OF=##, CD=##, FA=##]

P1 = Primary range start (required)
    Default: None
P2 = Primary range end (required)
    Default: None
S1 = Secondary range start (required)
    Default: None
S2 = Secondary range end (required)
    Default: None
DS = Offset amount
    Default: Null
OF = Offset side
    0 = Left
    1 = Right
    2 = None
    Default: None
CD = Chordal deviation
    Default: From SMF
FA = Finish amount
    Default: 0

ABS( )
(Absolute), *function*. Returns the absolute value of a numeric expression.

ABS(numexp)

ACOS( )
(Arc Cosine), *function*. Returns the arc cosine value of an angle.

ACOS(numexp)
    numexp = angle in degrees

ADD_LAYER[ ]
(Add Layer), *command*, Adds a new layer to the process model.

ADD_LAYER[LY=##, CO=##, ST=##]

LY = Layer (required)
    New layer number
    Default: None
CO = Color (required)
    Valid color number
    Default: None
ST = Style
    0 = Solid
    1 = Dotted
    2 = Broken
    Default: 0

ANG( )
(Angle), *function.* Returns the angle value of selected line or swept value of selected arc element.

ANG(elnum)
    elnum = Element number or name

ARC[ ]
(Arc), *command*. Creates a full or partial arc.

ARC[XC=##, YC=##, XS=##, YS=##,
    XE=##, YE=##, XI=##, YI=##, SS=##,
    SC=##, TS=$$, AS=##, SE=##, TE=$$,
    AE=##, SI=##, TI=$$, DR=##, RA=##,
    LV=##]

XC = X center
    Default: Null
YC = Y center
    Default: Null
XS = X start
    Default: Null
YS = Y start
    Default: Null
XE = X end
    Default: Null
YE = Y end
    Default: Null
XI = X intermediate
    Default: Null
YI = Y intermediate
    Default: Null
SS = Start solution designator
    0 = +
    1 = -
    2 = N/A
    Default: 0
SC = Center solution designator
    0 = +
    1 = -
    2 = N/A
    Default: 0
TS = Start tangent
    Arc or line element number
    Default: Null
AS = Start angle
    Default: Null
SE = End solution designator
    0 = +
    1 = -
    2 = N/A
    Default: 0

TE = End tangent
    Arc or line element number
    Default: Null
AE = End angle
    Default: Null
SI = Intermediate solution designator
    0 = +
    1 = -
    2 = N/A
    Default: 0
TI = Intermediate tangent
    Arc or line element number
    Default: Null
DR = Direction
    0 = CW
    1 = CCW
    Default: 0
RA = Radius
    Default: Null
LV = Level (modal)

ARC_GRP[ ]
(Arc Group), *command*. Adds or removes full and/or partial arcs to the active group.

ARC_GRP[AR=##, FA=##, NR=##, XR=##]

AR= Add/Remove
    0 = Add
    1 = Remove
    Default: None
FA= Full arc
    0 = All arcs
    1 = Full arcs only
    2 = Partial arcs only
    Default: 0
NR= Minimum radius
    Default: All arcs less than XR
XR= Maximum radius
    Default: all arcs greater than NR

ARC_PROF[ ]
(Arc Profile), *command*. Creates an arc as part of a continuous profile.

ARC_PROF[XC=##, YC=##, XE=##,
    YE=##, AE=##, XI=##, YI=##, XJ=##,
    YJ=##, XK=##, YK=##, DR=##, RA=##,
    TI=##, SS=##, SC=##, SE=##, SP=##]

XC = X center
    Default: Null
YC = Y center
    Default: Null
XE = X end
    Default: Null
YE = Y end
    Default: Null
AE = End angle
    Default: Null
XI = First intermediate X
    Default: Null
YI = First intermediate Y
    Default: Null
XJ = Second intermediate X
    Default: Null
YJ = Second intermediate Y
    Default: Null
XK = Third intermediate X
    Default: Null
YK = Third intermediate Y
    Default: Null
DR = Direction
    0 = CW
    1 = CCW
    Default: 0
RA = Radius
    Default: Null
TI = Tangent/intersect to previous element
    0 = Intersect
    1 =Tangent
    Default: 1
SS = Solution start
    0 = +
    1 = -
    2 = N/A
    Default: 0
SC = Solution center
    0 = +
    1 = -
    2 = N/A
    Default: 0
SE = Solution end
    0 = +
    1 = -
    2 = N/A
    Default: 0
SP = Solution pending
    0 = +
    1 = -
    2 = N/A
    Default: 0

AREA_CLR[ ]
(Area Clear), *command*. Creates tool path for removing material from inside a closed profile. {Afab}

AREA_CLR[PB=$$, NB=$$, XS=##, YS=##,
    XI=##, YI=##, TY=##, NT=##, WC=##,
    FA=##, PA=##, OA=##]

PB = Element in roughing area boundary
    Default: None
NB = Element in the material closure
    boundary for notch
    Default: Null
XS = X start corner
    Default: None
YS = Y start corner
    Default: None
XI = X inside area
    Default: None
YI = Y inside area
    Default: None
TY = Area clear
    0 = Spiral
    1 = ZigZag
    Default: 1
NT = Notch
    0 = Closed boundary
    1 = Notch
    Default: 0
WC = Width of cut
    Default: 0
FA = Finish amount
    Default: 0
PA = Pass angle (for ZigZag)
    Default: 0
OA = Overlap amount
    Default: 0

ASIN( )
(Arcsine), *function*. Returns the arcsine of an angle.

ASIN(numexp)
    numexp = angle in degrees

ATAN( )
(Arctangent), *function*. Returns the arctangent of an angle.

ATAN(numexp)
    numexp = angle in degrees

AUTO_ABORT[ ]
(Auto Abort), *command*. Automatically aborts macro execution when encountering a critical error. No error message will be displayed and the application will close without saving the current model. The auto-abort mode will stay on until macro completes or until it is explicitly turned off by AUTO_ABORT [ABORT=0].

Note: Auto Abort is not a recordable macro command.

AUTO_ABORT[ABORT=##]

ABORT = Auto-abort mode switch
    0 = Off
    1 = On:

AUTO_ANSWER[ ]
(Auto Answer), *command*. Automatically responds to non-error messages that are displayed. The selected response will be applied to the message and the macro will continue without stopping. Critical errors will not be ignored and will still stop the macro. Auto Answer cannot be used to trigger or accept non-message dialog boxes (such as Save or Display Modes). The auto-answer mode will stay on until macro completes or until it is explicitly turned off by AUTO_ANSWER [AA=“OFF”].

Note: Auto Answer is not a recordable macro command.

AUTO_ANSWER[AA=$$]

AA = Answer (required)
    “OFF” = Notice will be displayed as it would normally (interactive with user and interrupt macro processing)
    “OK” = Answers with “Ok” or “Yes” where applicable.
    “YES” = Answers with “Ok” or “Yes” where applicable.
    “CANCEL” = Answers with “Cancel” when applicable; if not, then like “OK”
    “NO” = Answers with “No” when applicable; if not, then like “Cancel” when applicable; if not, then like “OK”

AUTO_BALANCE[ ]
(Auto Balance Roughing), *command*. Creates mirrored tool path for the second turret, enabling both turrets to cut the same tool path simultaneously. {Aturn}

AUTO_BALANCE [TL=##]

TL = Tool number for copy (required)
    Default: None

AUT_NEST[ ]
(Auto Nest), *command*. Creates multiple copies of elements in the active group into a defined rectangular area.

AUT_NEST[XS=##, YS=##, XE=##, YE=##,
    FD=##, NS=##, SP=##, RO=##, FL=##,
    CO=##, ST=##]

XS = X Area start corner
    Default: None
YS = Y Area start corner
    Default: None
XE = X 2nd corner
    Default: None
YE = Y 2nd corner
    Default: None
FD = Row direction
    0 = X axis
    1 = Y axis
NS = Nest pattern
    0 = Grid
    1 = Stagger
SP = Space
    Default: 0
RO = 180˚ rotation allowed
    0 = No
    1 = Yes
FL = Fill area
    0 = Limit copies to copy limit
    1 = Fill the area
CO = Copy limit
ST = Sort by tools
    0 = No
    1 = Yes

- B -

BASE[ ]
(Base View), *command*. Changes the size of the view in the window to display the full area defined in the Envelope dialog box.

BASE[]

BEGIN
(Begin Block), *script*. Identified the start of a block of user interface script. Part of a set that includes BEGIN, END.

Note: Begin Block only functions while running in the macro demo mode without using the default UI scripts. Each BEGIN must be followed by a corresponding END within the macro.

BEGIN

BLEND[ ]
(Blend), *command*. Inserts an arc with a given radius between two lines or arcs on the same work plane and trim them to the tangency points.

BLEND[E1=$$, E2=$$, S1=##, S2=##,
    IN=##, DR=##, RA=##]

E1 = First element (required)
    Default: None
E2 = Second element (required)
    Default: None
S1 = First blend side (required)
    0 = Mouse
    1 = Right
    2 = Left
    Default: None
S2 = Second blend side (required)
    0 = Mouse
    1 = Right
    2 = Left
    Default: None
IN = Blend number
    1 = First blend
    2 = Second blend
    Default: 1
DR = Arc direction
    0 = Small
    1 = Large
    Default: 0
RA = Radius (required)
    Default: None

BLEND_SURFS[ ]
(Blend Mesh), *command*. Creates a constant or variable radius polyline blend mesh between two ranges of profiles using the properties and sequence you specify with the Insert Property bar. {Amill, Ffm}

BLEND_SURFS[E1=$$, E2=$$, S1=##,
    T1=##,E3=$$, E4=$$, S2=##, T2=##,
    PD=##, SP=##, SV=##, DI=##, DV=##,
    RS=##, RE=##, CC=##, CM=##,
    VT=##]

E1 = First surface start element (required)
    Default: None
E2 = First surface end element (required)
    Default: None
S1 = First surface keep side/blend side
    0 = Pos
    1 = Neg
    Default: 0
T1 = Trim first surface
    0 = Off
    1 = On:
    Default: 1
E3 = Second surface start element
    (required)
    Default: None

E4 = Second surface start element
(required)
Default: None
S2 = Second surface keep side/blend side
0 = Pos
1 = Neg
Default: 0
T2 = Trim second surface
0 = Off
1 = On:
Default: 1
PD = Path direction
0 = Cross (arcs)
1 = Along (flow)
Default: 0
SP = Spacing (type)
0 = Distance
1 = Count
Default: 0
SV = Spacing value (required)
Default: None
DI = Division (type)
0 = Distance
1 = Count
Default: 0
DV = Division value (required)
Default: None
RS = Start radius (required)
Default: None
RE = End radius (required)
Default: None
CC = Contact curve
0 = Off
1 = On:
Default: 0
CM = Connect mesh
0 = Off
1 = On:
Default: 0
VT = Virtual mesh tolerance
Default: Global variable
Initial Default: 0.001

BOUND_ALL[ ]
(Multiple Boundaries), *command*. Creates several 2D or 3D profiles using the surface boundaries that define the faces of a solid. This function does not change the existing surface element.

BOUND_ALL[SE=##, TO=##]

SE = Element index of surface
Default: None
TO = Tolerance for curve approximation
Default: 0.001.

BOUND_ONE[ ]
(Single Boundary), *command*. Creates a 2D or 3D profile using the surface boundaries that define the face of a solid. This function does not change the existing surface element.

BOUND_ONE[SE=##, TO=##, LP=##,
VI=##, DI=##]

SE = Element index of surface
Default: None
TO = Tolerance for curve approximation
Default: 0.001.
LP = Index of boundary
Default: 1
VI = Index of start vertex
Default: 1
DI = Whether or not to reverse the output direction
0 = No
1 = Yes
Default: 0

BOX_GRP[ ]
(Box Group), *command*. Adds elements to the active group by defining a rectangular area in the current view.

BOX_GRP[X1=##, Y1=##, X2=##, Y2=##,
PC=##, AR=##]

X1 = First box corner X (required)
Screen coordinate value
Default: None
Y1 = First box corner Y (required)
Screen coordinate value
Default: None
X2 = Second box corner X (required)
Screen coordinate value
Default: None
Y2 = Second box corner Y (required)
Screen coordinate value
Default: None
PC = Partial/complete (required)
0 = Partial
1 = Complete
Default: None
AR = Add/remove (required, modal)
0 = Add
1 = Remove

- C -

CAVITY[ ]
(Cavity Roughing), *command*. Creates a pocketing style of tool path for each planar profile in the active group. {Amill, Ffm}

```
CAVITY[TY=##, EL=##, WC=##, AN=##,
    AR=##, XC=##, YC=##, XS=##, YS=##,
    BC=##, CC=##, IP=##, UA=##, CL=##,
    RS=##, CI=##, EW=##, OP=##, RM=##,
    RT=##, AC=##, RD=##, IT=##, LY=##,
    NN=$$, UT=##]
```

TY = Cavity method:
 0 = Spiral
 1 = Zigzag
 2 = Linear
 Default: 0
EL = Material boundary element (required)
WC = Width of cut (required)
AN = Pass angle
 Default: 0
AR = Ramp angle
 Default: 90.0 (vertical)
XC = Cut X (required)
YC = Cut Y (required)
XS = Start X (no default, not required)
YS = Start Y (no default, not required)
BC = Boundary clearance
 Default: 0.0
CC = Connect cavity profile
 0 = Off
 1 = On:
 Default: 0
IP = Final island pass
 0 = Off
 1 = On:
 Default: 0 (not used in v4.0)
UA = Uncut area geometry—use
 0 = Off
 1 = On:
 Default: 0
CL = Climb cut
 0 = Conventional
 1 = Climb
 Default: 1
RS = Ramp from start
 0 = Off, use ramp angle
 1 = On, ramp from start point
CI = Cut inside out
 0 = Off (cut outside in)
 1 = On
 Default: 1
EW = Equalize width passes
 0 = Off
 1 = On:
 Default: 0
OP = Overlap passes
 0 = Off
 1 = On:
 Default: 0
RM = Rough tolerance method
 0 = Off, minimum
 1 = On, Mid Tol
 Default: 0
RT = Rough tolerance amount (distance)
 Default: 0.00005
AC = Corner roll angle
 Default: 60.0
RD = Rapid to depth
 0 = Off
 1 = On:
 Default: 0
IT = Island top machining
 0 = Full pass
 1 = Clear top
 Default: 0
LY = Layer
 Default: 1
NN = AUM named group name
 Default: AUM
UT = Uncut material tolerance
 Default: 0.01

CEX( )
(Center X), *function*. Returns the X coordinate of the center point of selected arc, ellipse or helix element.

CEX(elnum)
 elnum = Element number or name

CEY( )
(Center Y), *function*. Returns the Y coordinate of the center point of selected arc, ellipse or helix element.

CEY(elnum)
 elnum = Element number or name

CEZ( )
(Center Z), *function*. Returns the Z coordinate of the center point of selected arc, ellipse or helix element.

CEZ(elnum)
 elnum = Element number or name

CHAIN[ ]
(Chain), *command*. Converts connected elements into a contiguous profile of sequential elements (the end point of one element is the start point of the next

element). You can also join individual lines or polylines into one polyline element.

CHAIN[EL=$$, PJ=##]

EL = Element in profile (required)
    Default: None
PJ = Polyline join
    0 = Chain
    1 = Poly join
    2 = Both
    Default: 0

CHAMF[ ]
(Chamfer), *command*. Inserts a line of a given size at an angle relative to two line, arc, elliptical, or polyline elements on the same work plane. The Chamfer modeling tool also trims or extends the two existing elements to the intersection point of the chamfer.

CHAMF[E1=$$, E2=$$, S1=##, S2=##,
    IN=##, AN=##, DS=##, D1=##, D2=##]

E1 = First element (required)
    Default: None
E2 = Second element (required)
    Default: None
S1 = First chamfer side
    0 = Left
    1 = Right
    Default: 1
S2 = Second chamfer side
    0 = Left
    1 = Right
    Default: 1
IN = Intersect number
    1 = First intersection
    2 = Second intersection
    3 = Third intersection
    4 = Fourth intersection
    Default: 1
AN = Angle
    Default: Null
DS = Length of chamfer (distance)
    Default: Null
D1 = Distance parallel
    Default: Null
D2 = Distance perpendicular
    Default: Null

CHECK_MODEL[ ]
(Check Model), *command*. Immediately executes the ACIS body checker. It checks the geometry and topology of solids, surfaces, and surface trimming curves. It processes all surfaces or solids that are in the active group. Warning and error messages are reported to the SmartCAM Diagnostics window. {Amill, Ffm}

CHECK_MODEL[MC=##, MA=##]

MC = Model Checking (level of checking)
    0 = None
    1 = Low
    2 = Medium
    3 = High
    Default: 0
MA = Action
    0 = None
    1 = Delete
    Default: 0

CL_NAME_ELMT[ ]
(Clear Name Element), *command*. Removes a name assigned to an element and removes the element's name from the element list.

CL_NAME_ELMT[EL=$$]

EL = Element name or number to clear
    (required)
    Default: None

CLOSE[ ]
(Close Item), *script*. Closes a dialog box being displayed in the user interface.

Note: Close Item only functions while running in the macro demo mode without using the default UI scripts.

CLOSE[$$]

$$ = Name of dialog box

CLR( )
(Clear), *function*. Returns the clearance value of an element.

CLR(elnum) Clearance value
    elnum = Element name or number

CNECT_1WAY[ ]
(Connect 1 Way), *command*. Connects unconnected mesh by creating connecting lines, inserting them into the database between each pair of elements from the start element to the end element. The new lines are on the same work plane
and step or layer as start element.

Note: Connect 1 Way is not a recordable macro command.

CNECT_1WAY[E1=##, E2=##]

E1 = Start element
E2 = End element

CNV_A2H[ ]
(Convert Arcs To Holes), *command*. Converts the arc elements in the active group to holes.

CNV_A2H[CL=##, CN=##, KA=##, LV=##,
    TD=##]

CL = Clear
    Default: None
CN = Clear On
    0 = off
    1 = on
    2 - no change from arc setting
    Default: 0
KA = Keep original arcs
    0 = off
    1= on
    Default: 0
LV = Level
    Level of created holes
    Default: None
TD = Tip Depth
    Tip Depth
    Default: None

CODE[ ]
(Code), *command*. Generates the NC code for all unmasked step-property elements in the database.

CODE[FN=$$, MS1=$$, MT1=$$, SS1=$$,
    ST1=$$, MS2=$$, MT2=$$, SS2=$$,
    ST2=$$, SH=##, DC=##, RP=##,
    ST=##, DT=##, SP=##]

FN = Code output file name
    Default: Current model file name
MS1 = filename of Master Smf for turret 1
    (was previously MF).
MT1 = filename of Master Tmp for turret 1
    (was previously TF).
SS1 = filename of Slave Smf for turret 1 (for
    Advanced Turning only).
ST1 = filename of Slave Tmp for turret 1 (for
    Advanced Turning only).
MS2 = filename of Master Smf for turret 2
    (for Advanced Turning only).
MT2 = filename of Master Tmp for turret 2
    (for Advanced Turning only).
SS2 = filename of Slave Smf for turret 2 (for
    Advanced Turning only).
ST2 = filename of Slave Tmp for turret 2 (for
    Advanced Turning only).
SH = Show_Path
    0 = Off
    1 = On: Default: 1
DC = Display code
    0 = Off
    1 = On: Default: 1
RP = Report
    0 = Off
    1 = On: Default: 0
ST = Show tool
    0 = Filled
    1 = Nibble, 2=Draw_End Default: 0
DT = Draw tool
    0 = 3-D
    1 = Flat Default: 0
SP = Speed
    0-9 Default: 8

CODFILE( )
(Code File), *function*. Returns the code file name of the last file coded in the current session.

Note: Code File is not a recordable macro function.

CODFILE()

COLOR CHG[ ]
(Color Change), *command*. Changes the color for existing elements assigned to a step tool, or to change the color and the line style (solid, broken, or dotted) for elements assigned to existing layers.

COLOR_CHG[CO=##, TY=##, TL=##,
    DI=##, ST=##]

CO = Color
    1–15
    Default: 1
TY = Type
    0 = Step
    1 = Layer
    2 =Tool
    Default: 0
TL = Step/layer/tool (required)
    Default: None
DI = Tool diameter (not available in v4.0,
    ignored)
    Default: Null

ST = Style (for layer only)
0 = Solid
1 = Broken
2 = Dotted
Default: 0

CONNECT[ ]
(Connect Mesh), *function*. Connects individual mesh profiles in the active group into a continuous zigzag profile by inserting a line element between the end of one profile and the start of the adjacent profile. Can also disconnect grouped profiles, which results in elements traveling in the same direction and sequentially in the database. {Amill, Ffm}

CONNECT[DS=##, DM=##]

DS = Maximum connection distance
(required)
Default: None
DM = Disconnect mesh
0 = Off
1 = On:
Default: 0

CONTOUR[ ]
(Contour Machining), *command*. Creates a tool path that is a series of profile cuts at successive Z levels around grouped NURBS surfaces. {Ffm}

CONTOUR [SU=##, DS=##, DC=##,
EU=##, DE=##, US=##, XS=##, YS=##,
OA=##, TI=##, TO=##, SS=##, PT=##,
CL=##, CP=##, CF=##, LU=##, LN=$$,
FU=##, IU=##, FN=$$, FTS=$$, LC=##,
TY=##, AN=##, DI=##, RA=##, AS=##,
AE=##, AD=##, AV=##]

SU = Use Depth of Start Z/First Pass Level
0 = Off
1 = On:
Default: 0
DS = Depth of Start Z/First Pass Level
(required if SU= 1)
Default: 0
DC = Depth of Cut (required)
Default: 0
EU = Use Depth of Ending Z/Final Pass Level
0 = Off
1 = On:
Default: 0
DE = Depth of Ending Z/Final Pass Level
(required if EU = 1)
Default: 0
US = Use Profile Start Point
0 = Off
1 = On:
Default: 0
XS = Profile Start Point X value (required if US = 1)
Default: 0
YS = Profile Start Point Y value (required if US = 1)
Default: 0
OA = Offset (Finish) Amount (required)
Default: 0
TI = In Tolerance (required)
Default: 0.01
TO = Out Tolerance (required)
Default: 0.01
SS = Surface Side
0 = Off
1 = On:
Default: 0
PT = Path Type
0 = ZigZag
1 = One Way Plunge
Default: 1
CL = Climb Cut
0 = Conventional
1 = Climb
Default: 1
CP = Closed Profiles Only (required)
0 = Off/Open+Closed
1 = On/Closed Only)
Default: 1
CF = Clear/Feed
0 = Absolute
1 = Increment
Default: 0
LU = Log File, Use
0 = Off
1 = On:
Default: 0
LN = Log File, Name
Default: ffm.ini::LogFile.machine: Paths.def/machine.log (where Paths.def is the path from the Paths.def variable in the `ffm.ini` file)
FU = External File, Use
0 = Off
1 = On:
Default: 0
IU = Lead In/Lead Out, Use
0 = Off
1 = On:
Default: 0

FN = External File, Name
Default:
ffm.ini::ExternalFile.SurfaceMachine:
Paths.def/pmfile.ext (where
Paths.def/pmfile variable is in the
`ffm.ini` file; ext is `.sat`, `.sab`, or
`.igs`, depending on FT.)
FTS = External File, Type
SAT, SAB, IGSS
AS = Adjust First Pass (Start)
0 = No
1 = Yes
Default: 1
AE = Adjust Final Pass (End)
0 = No
1 = Yes
Default: 1
AD = Adjust Depth of Cut
0 = No
1 = Yes
Default: 1
AV = Adjustment Value
a valid constant or expression
Default: 0.01*(In Tol+Out Tol)/2 where
In Tol and Out Tol are input values from
the control panel Lead In/Lead Out
dialog variables:
LC = Location (required if IU = 1)
0 = In
1 = Out
2 = Both)
Default: 2
TY = Type (required if IU = 1)
0 = Line
1 = Arc
2 = Both
Default: 2
AN = Angle (required if IU = 1)
Default: 30
DI = Distance (Length of Line) (required if IU
= 1 and TY = 0 or 2)
Default: 2*TL(TL_DIA))
RA = Radius, for Arc (required if IU = 1 and
TY = 1 or 2)
Default: 1*TL(TL_DIA))

CONTOUR_RGH[ ]
(Contour Roughing), *command*. Creates tool path with roughing passes that follow the shape of an existing part profile. This command has been superceded by the TCONTOUR[] macro command. An attempt has been made to make this command forward compatible with newer command. {Pturn, Aturn}

Note: Contour Roughing is not a recordable macro command.

CONTOUR_RGH[ES=##, EE=##, XS=##,
YS=##, EM=##, ZF=##, XF=##, DP=##,
CL=##, NP=#]

ES = Finish profile start (required)
Default: None
EE = Finish profile end (required)
Default: None
XS = Z start (required)
Default: None
YS = X start (required)
Default: None
EM = Material element (required)
Default: None
ZF = Z finish amount (required)
Default: None
XF = X finish amount (required)
Default: None
DP = Pass depth (required)
Default: None
CL = Clear (required)
Default: None
NP = Number of passes
Default: Null

COONS[ ]
(Coons Mesh), *command*. Creates sculpted mesh profiles by using four connected boundary profiles that define the mesh area and cross-section profiles that define interior contour. {Amill, Ffm}

COONS[E1=$$, E2=$$, E3=$$, E4=$$,
BP=##, PD=##, SP=##, SV=##, ZT=##,
DI=##, DV=##, OF=##, OA=##, GC=##,
CM=##]

E1 = First profile start (required)
Default: None
E2 = First profile end (required)
Default: None
E3 = Second profile start (required)
Default: None
E4 = Second profile end (required)
Default: None
BP = By patch tangents
0 = Off
1 = On:
Default: 0
PD = Path direction
0 = Cross
1 = Along
2 = Planar
Default: 0

SP = Spacing
  0 = Distance
  1 = Count
  2 = Cusp Hgt
  Default: 0
SV = Spacing value (required)
  Default: None
ZT = Planar Z
  Default: 0.0
DI = Division
  0 = Distance
  1 = Count
  2 = Mid Tol
  3 = In Tol
  4 = Out Tol
  Default: 0
DV = Division value (required)
  Default: None
OF = Offset
  0 = Positive
  1 = Negative
  2 = 0
  Default: 2
OA = Finish amount
  Default: 0.0
GC = Gouge correction
  0 = Off
  1 = On:
  Default: 0
CM = Connect mesh
  0 = Off
  1 = On:
  Default: 0

COORDLST[ ]
(Coordinate List), *command*. Assigns a list of 2D or 3D coordinate to a variable to be used as a coordinate list variable.

Note: Coordinate List is not a recordable macro command.

COORDLST[VN=$$, DM=##]

VN = Variable name (required)
  Default: None
DM = Dimension, number of ordinates
  2 = 2-D (`.sh2`)
  3 = 3-D (`.sh3`)
  Default: 3

COS( )
(Cosine), *function*. Returns the cosine of an angle.

COS(numexp)
  numexp = angle in degrees

CPX( )
(Control Point X), *function*. Returns the X coordinate of a specified control point of selected polyline or spline element.

CPX(elnum,ptnum)
  elnum = Element number or name
  ptnum = Point number of polyline or spline

CPY( )
(Control Point Y), *function*. Returns the Y coordinate of a specified control point of selected polyline or spline element.

CPY(elnum,ptnum)
  elnum = Element number or name
  ptnum = Point number of polyline or spline

CPZ( )
(Control Point Z), *function*. Returns the Z coordinate of a specified control point of selected polyline or spline element.

CPZ(elnum,ptnum)
  elnum = Element number or name
  ptnum = Point number of polyline or spline

CRE_LINK[ ]
(Create Links), *command*. Creates lines to establish links between two defining profiles. {Awedm}

CRE_LINK[CC=##, P1=##, P2=##, S1=##,
  S2=##, XS=##, YS=##, ZS=##, XE=##,
  YE=##, ZE=##]

CC= Create Choice
  0 = Range
  1 = Element
  Default: None
P1 = Primary range start
  Default: None
P2 = Primary range End
  Default: None
S1 = Secondary range start
  Default: None
S2 = Secondary range end
  Default: None
XS = X start
  Default: None
YS = Y start
  Default: None

ZS = Z start
    Default: None
XE = X end
    Default: None
YE = Y end
    Default: None
ZE = Z end
    Default: None

CYLDIAM( )
(Cylinder Diameter), *function*. Returns the cylinder diameter of an acis cylinder. {Amill, Ffm}

CYLDIAM(elnum)
    elnum = Element number or name

- D -

DLGADD[ ]
(Dialog Add), *script*. Displays a dialog boy in the user interface.

Note: Dialog Add only functions while running in the macro demo mode without using the default UI scripts.

DLGADD[$$]

$$ = Name of dialog box

DRL_DEFINE[ ]
(Define Drill Sub), *command*. Assigns the active group of hole elements to a drilling subroutine, then places a related Drill Call element in the database. {Pmill, Amill, Ffm}

DRL_DEFINE[NN=##, CP=##, XE=##,
    YE=##, ZE=##, PK=##]

NN = Drill subroutine name
    Default: None
CP = Repeats
    Default: 0
XE = Handle point X
YE = Handle point Y
ZE = Handle point Z
PK = Peck
    0 = Off
    1 = On:

DEF_PLANE[ ]
(Define Plane), *command*. Creates a new work plane for the model or change the parameters for an existing non-reserved work plane.

DEF_PLANE[TY=##, FR=##, WP=$$,
    TP=$$, X1=##, Y1=##, Z1=##, X2=##,
    Y2=##, Z2=##, XI=##, YI=##, ZI=##,
    RO=##, AA=##, AB=##, AC=##, ZD=##,
    MP=##]

TY = Type (definition method)
    0 = 3 points
    1 = Line/angle
    2 = Rotation
    Default: 0
FR = From (which coordinate system)
    0 = Active plane
    1 = World
    Default: 1
WP = Work plane (required)
    Default: None
TP = Tool plane (required)
    Default: None
X1 = X of origin point (required)
    Default: None
Y1 = Y of origin point (required)
    Default: None
Z1 = Z of origin point (required)
    Default: None
X2 = X of plus-X point (required for 3-point and line/angle plane definitions)
    Default: None
Y2 = Y of plus-X point (required for 3-point and line/angle plane definitions)
    Default: None
Z2 = Z of plus-X point
    Default: Z1
XI = X of third point (required for 3-point plane definitions)
    Default: None
YI = Y of third point (required for 3-point plane definitions)
    Default: None
ZI = Z of third point (required for 3-point plane definitions)
    Default: None
RO = Rotation order (required)
    0 = ABC
    1 = BAC
    2 = ACB
    3 = BCA
    Default: None
AA = Angle A (required for line/angle and rotation)
    Default: None
AB = Angle B (required for rotation)
    Default: None
AC = Angle C
    Default: 0.0 for rotation

ZD = Z-axis direction
 0 = Pos
 1 = Neg
 Default: 0
MP = Match plane
 0 = Off
 1 = On:
 Default: 0

DELETE[ ]
(Delete Element), *command*. Removes a single element from the model.

DELETE[EL=$$]

EL = Element number or name (required)
 Default: None

DIM_RECT[ ]
(Linear Dimension), *command*. Creates a dimension for a 2-D linear distance.

```
DIM_RECT[XE=##, YE=##, XS=##, YS=##,
    XI=##, YI=##, LV=##, TX=$$, PX=$$,
    SX=$$, AL=##, DT=##, NF=##, EG=##,
    ED=##, DG=##, L1=##, L2=##, HT=##,
    LA=##, AR=##, FN=$$, AF=$$]
```

XE, YE = 2nd point to dimension
XS, YS = 1st point to dimension
XI, YI = Text location
LV = Level for dimension
TX = Text (optional). Will override dimension
 value specified
PX = Text prefix (optional)
SX = Text suffix (optional)
AL = Align text
 0 = Off
 1 = On:
 Default: 0
DT = Dimension type parallel
 0 = x axis
 1 = y axis
 2 = aligned with points
 Default: 2
NF = Numerical format
 Range 1-5
EG = Extension line gap
 Default: 0.0
ED = Extension line distance
 Default: 0.0
DG = Dimension line text gap
 Default: 0.0
L1 = 1st extension line create
 0 = Off
 1 = On:
 Default: 1
L2 = 2nd extension line create
 0 = Off
 1 = On:
 Default: 1
HT = Height of text
 Default: 1.0
LA = Length of arrow
 Default: 1.0
AR = Arrow (line terminators)
 0 = Off
 1 = On:
 Default: 1
FN = Text font file name
 Default: None
AF = Arrow file name
 Default: None

DIM_TEX[ ]
(Dimension Text), *command*. Creates a note on a model with or without a leader from the note's text that points to a particular area on the model.

```
DIM_TEX[XE=##, YE=##, LV=##, TX=$$,
    AL=##, AN=##, DG=##, HT=##, LA=##,
    AR=##, FN=$$, AF=$$, CL={}]
```

XE = X coordinate
YE = Y coordinate
LV = Z location of all text/leaders
 Default: None
TX = Text input
 Default: Blank
AL = Align text
 0 = Off
 1 = On:
 Default: 0
AN = Rotation angle for text
 Default: 0
DG = Dimension line text gap
 Default: 0.0
HT = Height of text
 Default: 1.0
LA = Length of arrow
 Default: 1.0
AR = Arrow (line terminators)
 0=Off, 1=On
 Default: 1
FN = Text font file name
 Default: None
AF = Arrow file name
 Default: None
CL = 2-D coordinate list {X1,Y1,....X9, Y9}
 location of text/leader
 Default: None

DIR( )
(Direction), *function*. Returns the direction of an arc, helix, or ellipse.

DIR(elnum)
 0 = CW
 1 = CCW
 -1 = Not an arc element

DISP_MODE[ ]
(Display Modes), *command*. Controls how SmartCAM displays your model in the graphic view.

```
DISP_MODE[TH=##, WA=##, WI=##,
    GR=##, DL=##, IP=##, RL=##, VF=##,
    SZ=##, GI=##, CS=##, DT=##, GM=##,
    AR=##, SN=##, GC=##, CR=##,
    CG=##, CB=##, GA=##]
```

TH = Thickness
 0 = Off
 1 = On:
 Default: N/C
WA = World XYZ axes
 0 = Off
 1 = On:
 Default: N/C
WI = Work plane indicator
 0 = Off
 1 = On:
 Default: N/C
GR = Grid
 0 = Off
 1 = On:
 Default: N/C
DL = Data list in world
 0 = No
 1 = Yes
 Default: N/C
IP = Input plane in world
 0 = No
 1 = Yes
 Default: N/C
RL = Ruler
 0 = Off
 1 = On:
 Default: N/C
VF = Vertical frequency
 0–99
 Default: N/C
SZ = Relative size for work plane indicator
 0–1
 Default: N/C
GI = Grid increment
 Default: N/C
CS = Curve segment count
 0–99
 Default: N/C
DT = Draw tools
 0 = No tools
 1 = Flat tools
 2 = Blips
 3 = No tools
 Default: N/C
GM = Group marking
 0 = Arrow
 1 = Dotted
 2 = Both
 Default: N/C
AR= Auto Redraw
 0 = Off
 1 = On:
SN = Surface normal display
 0 = Off
 1 = All
 2 = Group
 Default: 0
GC = Surface grid color
 0 = Match surface
 1 = Shade darker
 2 = Shade lighter
 3 = Fixed color RGB
 4 = Fixed default dark
 5 = Fixed default light
 Default: 3
CR = Surface grid red
 0–255
 Default: 82
CG = Surface grid green
 0–255
 Default: 139
CB = Surface grid blue
 0–255
 Default: 139
GA = Shade adjustment amount
 1–254
 Default: 80

DMS( )
(Degrees Minutes and Seconds), *function*. Returns a decimal angle from degrees, minutes and seconds.

DMS(ddd.mmss)
 ddd = degrees
 mm = minutes (must be 2 digits)
 ss = seconds (must be 2 digits)

DRAFT_SURF[ ]
(Draft Mesh), *command*. Creates mesh profiles by sweeping a generator profile

along a 3-D director profile's path. {Amill, Ffm}

DRAFT_SURF[E1=$$, E2=$$, E3=$$,
    E4=$$, PD=##, SP=##, OF=##, SV=##,
    ZT=##, 0A=##, CM=##, GC=##, VT=##]

E1 = Generator profile start (required)
    Default: None
E2 = Generator profile end (required)
    Default: None
E3 = Director profile start (required)
    Default: None
E4 = Director profile end (required)
    Default: None
PD = Path direction
    0 = Generator
    1 = Planar
    Default: 0
SP = Spacing
    0 = Distance
    1 = Count
    Default: 0
OF = Offset
    0 = Positive
    1 = Negative
    2 = 0
    Default: 2
SV = Spacing value (required)
    Default: None
ZT = Planar Z
    Default: 0.0
OA = Finish amount
    Default: 0.0
CM = Connect mesh
    0 = Off
    1 = On:
    Default: 0
GC = Gouge correction
    0 = Off
    1 = On:
    Default: 0
VT = Virtual mesh tolerance
    Default: Global variable
    Initial default: 0.001

DRL_CALL[ ]
(Drill Subroutine Call), *function*. Assigns a previously defined drilling subroutine to different locations on the model. {Pmill, Amill, Ffm}

DRL_CALL[NN=##, CP=##, XE=##, YE=##,
    ZE=##, PK=##]

NN = Name of subroutine
    Default: None
CP = Number of repeats
    Default: 0
XE = Insert point X coordinates
YE = Insert point Y coordinates
ZE = Insert point Z coordinates
PK = Peck
    0 = Off
    1 = On:

DUMP_GR[ ]
(Print Graphics), *command*. Prints the current graphic view to the configured printing device.

DUMP_GR[LM=##, TM=##, RM=##, ST=##,
    SF=##]

LM = Left margin
TM = Top margin
RM = Right margin
BM = Bottom margin
ST = Show tool
    0 = None
    1 = Nibble
    2 = Draw End - 2-D (.sh2)
    3 = Draw End - 3-D (.sh3)
    Default: 0
SF = Scale value
    Default: 1

DUMP_LST[ ]
(Print Element Data List), *command*. Prints a list of the element data for the entire file or the active group in the open process model.

DUMP_LST[GP=##, OM=##, FN=$$]

GP = What should be listed
    0 = Entire file
    1 = Active group
    Default: None
OM = Output method
    0 = Printer
    1 = File
    Default: None
FN = Filename (required for OM=1)
    Default: None

DYNAMIC_VIEW[ ]
(Dynamic View), *command*. Changes the orientation of the model view in the active window.

DYNAMIC_VIEW[XX=##, XY=##, XZ=##,
    YX=##, YY=##, YZ=##, OX=##, OY=##,
    OZ=##, LX=##, RX=##, TY=##, BY=##]

XX = Rotation plane vector
XY = Rotation plane vector
XZ = Rotation plane vector
YX = Rotation plane vector
YY = Rotation plane vector
YZ = Rotation plane vector
OX = X origin of view plane
OY = Y origin of view plane
OZ = Z origin of view plane
LX = Location/size of view window in view plane
RX = Location/size of view window in view plane
TY = Location/size of view window in view plane
BY = Location/size of view window in view plane

- E -

ELLIPSE[ ]
(Ellipse), *command*. Creates a full or partial ellipse oriented at any angle.

ELLIPSE[D1=##, D2=##, XC=##, YC=##, XS=##, YS=##, XE=##, YE=##, AS=##, AE=##, AN=##, DR=##, SS=##, SE=##, LV=##]

D1 = Primary distance (required)
    Default: None
D2 = Secondary distance (required)
    Default: None
XC = X center (required)
    Default: None
YC = Y center (required)
    Default: None
XS = X start
    Default: Null
YS = Y start
    Default: Null
XE = X end
    Default: Null
YE = Y end
    Default: Null
AS = Angle start
    Default: Null
AE = Angle end
    Default: Null
AN = Angle of inclination
    Default: 0.0
DR = Direction
    0 = CW
    1 = CCW
    Default: 0
SS = Solution start
    0 = +
    1 = -
    2 = N/A
    Default: 0
SE = Solution end
    0 = +
    1 = -
    2 = N/A
    Default: 0
LV = Level (modal)

ELMT_GRP[ ]
(Element Group), *command*. Adds or removes an element to or from the active group.

ELMT_GRP[EL=##, AR=##]

EL = Element (required)
    Default: None
AR = Add/Remove (modal)
    0 = Add
    1 = Remove

ELMT_SEQ[ ]
(Element Sequence), *command*. Sets the insertion point in the database relative to a specific element.

ELMT_SEQ[BA=##, EL=##, ME=##]

BA = Before/after (required)
    0 = Before
    1 = After
    Default: 1
EL = Element (required)
    Element number or element name
    Default: None
ME = Match element
    0 = Off
    1 = On:
    Default: 0

ELSE
(Logical Else), *command*. Is followed by macro commands to be preformed if a preceding conditional test proves false. Part of a command set that includes IF( ), ELSE, ENDIF.

Note: Logical Else is not a recordable macro command. Each ELSE command must be located between corresponding IF and ENDIF commands within the macro.

```
IF(expression)
     true
ELSE
     false
ENDIF
```

expression = A logic expression.
true = Macro commands to be performed if
     the logic expression proves true.
False = Macro commands to be performed if
     the logic expression proves false

END
(End Block), *script*. Identified the end of a block of user interface script. Part of a set that includes BEGIN, END.

Note: End Block only functions while running in the macro demo mode without using the default UI scripts. Each END must be preceded by a corresponding BEGIN within the macro.

```
END
```

ENDIF( )
(Logical End If), *command*. Identifies the end of an IF command set. Part of a command set that includes IF( ), ELSE, ENDIF.

Note: Logical End If is not a recordable macro command. Each ENDIF command must be preceded by a corresponding IF command within the macro.

```
IF(expression)
     true
ELSE
     false
ENDIF
```

expression = A logic expression.
true = Macro commands to be performed if
     the logic expression proves true.
False = Macro commands to be performed if
     the logic expression proves false.

ENDW
(Logical While Loop End), *command*. Identifies the end of a WHILE command set. Part of a command set that includes WHILE( ), ENDW.

Note: Logical While Loop End is not a recordable macro command. Each ENDW command must be preceded by a corresponding WHILE command within the macro.

```
WHILE(expression)
     true
ENDW
```

expression = A logic expression.
true = Macro commands to be performed if
     the logic expression proves true.

ENV_XE( )
(Envelope X end), *function*. Returns the X coordinate of the end of the view envelope.

```
ENV_XE()
```

ENV_YE( )
(Envelope Y end), *function*. Returns the Y coordinate of the end of the view envelope.

```
ENV_YE()
```

ENV_ZE( )
(Envelope Z end), *function*. Returns the Z coordinate of the end of the view envelope.

```
ENV_ZE()
```

ENV_XS( )
(Envelope X Start), *function*. Returns the X coordinate of the end of the view envelope.

```
ENV_XS()
```

ENV_YS( )
(Envelope Y Start), *function*. Returns the Y coordinate of the end of the view envelope.

```
ENV_YS()
```

ENV_ZS( )
(Envelope Z Start), *function*. Returns the Z coordinate of the end of the view envelope.

```
ENV_ZS()
```

ENVELOPE[ ]
(Envelope), *command*. Defines how much of the model and space around it to include in the base graphic view.

```
ENVELOPE[XS=##, YS=##, ZS=##, XE=##,
     YE=##, ZE=##]
```

XS = X start (required)
     Default: None

YS = Y start (required)
    Default: None
ZS = Z start (required)
    Default: None
XE = X end (required)
    Default: None
YE = Y end (required)
    Default: None
ZE = Z end (required)
    Default: None

ENX( )
(End Point X), *function*. Returns the X coordinate of the end point of selected element.

ENX(elnum)
    elnum = Element number or name

ENY( )
(End Point Y), *function*. Returns the X coordinate of the end point of selected element.

ENY(elnum)
    elnum = Element number or name

ENZ( )
(End Point Z), *function*. Returns the Z coordinate of the end point of selected element.

ENZ(elnum)
    elnum = Element number or name

EXECUTE[ ]
(Execute Action), *script*. Executes the macro command that preceded the block of script supporting it.

Note: Execute Action only functions while running in the macro demo mode without using the default UI scripts.

EXECUTE[]

EXPLODE[ ]
(Explode), *command*. Converts elements in the active group into primitive elements.

EXPLODE[EX=##, TO=##]

EX = Explode level
    0 = Lines
    1 = Arcs
    2 = Polylines
    Default: 2
TO = Tolerance
    Default: 0.001

EXPORT[ ]
(Export), *command*. Converts the elements in the existing model and produces an ACIS, DXF, DWG, IGES, or VDA-FS file. {Amill, Ffm}

Note: The Export can be recorded but will not function when executed.

EXPORT[FN=$$, FTS=$$, EX=##, LU=##,
    LN=$$, SF=$$, TF=$$]

FN = Filename (required)
FTS= Export file type
    SAT21
    SAB21
    SAT17
    SAT16
    SAT15
    DWG
    DXF
    IGS
    VDAFS
EX = Export Solids
    0 = On
    1 = Off
LU = Log file—use (optional)
    0 = No log file written
    1 = Write log file
    Default: 0
LN = Log file name (required)
SF = Setup File Name (required for CAM
    Connection file
    types DWG, DXF, IGES. Optional for
    other file types.)
    Default: from .ini file
TF = Temporary CAM Connection Input File
    Name
    (required for CAM Connection file types
    DWG, DXF, IGES. Optional for other file
    types.)
    Default: from .ini file

EXTEND[ ]
(Extend Mesh), *command*. Extends a selected range of mesh profiles beyond the mesh's defining boundaries. {Amill, Ffm}

EXTEND[E1=$$, E2=$$, D1=##, D2=##,
    DS=##, DE=##, CM=##]

E1 = Mesh start element (required)
    Default: None

E2 = Mesh end element (required)
Default: None
D1 = Along division start distance
Default: 0.0
D2 = Along end
Default: 0.0
DS = Across spacing start distance
Default: 0.0
DE = Across end
Default: 0.0
CM = Connect mesh
0 = Off
1 = On:
Default: 0

EXTEND_SURF[ ]
(Extend Surface), *command*. Extends an existing surface from a single edge boundary by creating a new NURBS surface that shares the edge boundary with the original surface element. {Amill, Ffm}

EXTEND_SURF [ES=$$, LN=##, EN=##, ET=##, XS=##, YS=##, ZS=##, XE=##, YE=##, ZE=##]

ES = Element number of surface to edit (required)
Default: None
LN = Loop number, relative to boundary ordering in ES database (required)
Default: None
EN = Edge number, relative to boundary ordering in ES database (required)
Default: None
ET = Extend type
0=G0, 1=G1
Default: 0
XS = Start/mid vector, X component
Default: 0.0
YS = Start/mid vector, Y component
Default: 0.0
ZS = Start/mid vector, Z component
Default: 0.0
XE = End vector, X component
Default: 0.0
YE = End vector, Y component
Default: 0.0
ZE = End vector, Z component
Default: 0.0

EXTRACT[ ]
(Extract), *command*. Explodes surfaces from solids or extract the constituent elements from a surface or group of surfaces. {Amill, Ffm}

EXTRACT[ET=##, CE=##, BE=##, TE=##, DE=##, AP-##, KO=##, HS=##, OL=##]

ET = Extract Type
0 = Extract surfaces from solids
1 = Extract elements from surfaces
Default: 1
CE = Construction geometry elements
0 = No extract
1 = Extract
Default: 0
BE = Base surface edges
0 = No extract
1 = Extract
TE = Trimming curve elements
0 = No extract
1 = Extract
Default: 0
DE = Display grid elements
0 = No extract
1 = Extract
Default: 0
AP = Assign properties
0 = Original
1 = Active properties
2 = Parent surface properties
Default: 0
KO = Keep original surface
0 = No keep/delete
1 = Keep
Default 1
HS = Hide surface
0 = No hide
1 = Hide
Default: 1
OL = Hide on layer (required)
1–99
Default: 99

- F -

F_CLOSE[ ]
(File Close), *command*. Closes an open ASCII text file.

Note: File Close is not a recordable macro command.

F_CLOSE[FN=$$]

FN = Filename (required)

F_EOF( )
(End Of File), *function*. Returns the end of file status of an open ASCII text file.

Note: End Of File is not a recordable macro function.

F_EOF(file)
    file = Filename (required)
Return values:
    0 = If not end of file
    1 = If end of file

F_ERROR( )
(File Error), *function*. Returns the error status of an open ASCII text file.

Note: File Error is not a recordable macro function.

F_ERROR()

Return values:
    0 = Successful operation (F_READ, F_OPEN, F_WRITE, F_CLOSE)
    1 = File already open (F_OPEN)
    2 = File not found (F_OPEN)
    3 = Path not found (F_OPEN)
    4 = File Is read-only (no write or append allowed) (F_OPEN)
    5 = File not open (F_READ)
    6 = File not open for read (F_READ)
    7 = End of file reached (F_READ)
    8 = File not open for write (F_WRITE)
    9 = Too many files already opened (F_OPEN)
    10 = Bad or missing mode; use A W R (F_WRITE)
    11 = Expected variable for formatter missing (F_WRITE, F_READ)
    12 = Bad character encountered in variable list (F_WRITE, F_READ)
    13 = Variable is not of format type (F_WRITE, F_READ)
    14 = Bad formatter (F_WRITE, F_READ)

F_ERRST( )
(File Error String), *function*. Returns the string for the error status returned by F_ERROR()

Note: File Error String is not a recordable macro function.

F_ERRSTR()

F_EXIST( )
(File Exist), *function*. Returns a value verifying the existence of a file.

Note: File Exist is not a recordable macro function.

F_EXIST(strexpr)
    strexpr = Filename (required)
Return values:
    0 = File does not exist
    1 = File does exist

F_LINE( )
(File Line), *function*. Returns the line number just read by the F_READ() command.

Note: File Line is not a recordable macro function.

F_LINE(strexpr)
    strexpr = Filename (required)

F_OPEN[ ]
(File Open), *function*. Opens an ASCII text file.

Note: File Open is not a recordable macro command.

F_OPEN[FN=$$, TY=$$]

FN = Filename (required)
TY = Open file "TY" file type
    Options: W = Write
    R = Read
    A = Append

F_READ[ ]
(File Read), *command*. Reads information from an open ASCII text file.

Note: File Read is not a recordable macro command.

F_READ[FN=$$, FMT="Format of string",
    VR="VARLIST"]

FN = Filename (required)
FMT = The format for each user variable
    specified by the VR option.
VR = The name of the user variables to be
    stored.
    Note: The variables must exist and be
    initialized or stringed.

F_WRITE[ ]
(File Write), *command*. Writes information to an open ASCII text file.

Note: File Write is not a recordable macro command.

F_WRITE[FN=$$, FMT="Format of string",
VR ="VARLIST"]

FN = Filename (required)
FMT = The format for each user variable specified by the VR option.
Format characters:
% = signals a variable follows
%% = prints a preceding % sign
~ = do not interpret the next character, output as a literal
~n = new line (carriage return) identifier
~t = insert a tab character
Numerical output formats:
D =decimal (suppress trailing / leading zeros)
T = trailing zero (no decimal, suppress leading zeros)
L =leading zero (no decimal, suppress trailing zeros)
F =filled (no decimal, fill unfilled fields with zeros)
P =padded (decimal, suppress leading zero)
E =filled (decimal, fill unfilled fields with zeros)
I =integer output
S =string (outputs all characters in the string)
VR =The name of the user variable
Note: The variables must exist and be initialized or stringed.

FAB_HOLE[ ]
(Fabrication Hole), *command*. Creates a hole-making operation by using the active step and related properties. {Afab}

FAB_HOLE[XE=##, YE=##, XS=##, YS=##,
AE=##, AN=##, DS=##, LV=##, RI=##,
SE=##]

XE = X end
Default: Null
YE = Y end
Default: Null
XS = X anchor
Default: Null
YS = Y anchor
Default: Null
AE = Tool angle
Default: Null
AN = Angle
Default: Null
DS = Distance
Default: Null
LV = Level
Default: Modal
RI = Radius input
0 = No
1 = Yes
SE = End solution designator
0 = +
1 = -
2 = N/A
Default: 0

FABRULE[ ]
(Fabrication Ruled Mesh), *command*. Creates ruled transitions between two defining profiles. {Afab}

FABRULE[E1=$$, E2=$$, E3=$$, E4=$$,
PD=##, SP=##, SV=##, ZT=##, DI=##,
DV=##, CM=##]

E1 = Start element of first profile (required)
E2 = End element of first profile (required)
E3 = Start element of second profile (required)
E4 = End element of second profile (required)
PD = Path direction
SP = Spacing type
0 = Distance
1 = Count
Default: 1
SV = Spacing value
ZT = Planar Z
DI = Division type
DV = Division value
CM = Build face binary
0 = Off
1 = On:
Default: 0

FACE[ ]
(Face Roughing), *command*. Creates tool path for removing material from a part face. {Pmill, Amill, Ffm}

FACE[TY=##, EL=##, FA=##, WC=##,
AN=##, XS=##, YS=##, BC=##, DS=##,
DC=##, DE=##, EP=##, IN=##, IP=##,
AR=##, UA=##, CC=##, RS=##, FL=##,
CI=##, EW=##, OP=##, RM=##, RT=##,
AC=##, RD=##, IT=##, LY=##, NN=$$,
UT=##]

TY = Face method
0 = Spiral

1 = Zigzag
2 = Linear
Default: 0
EL = Outside boundary element (required)
FA = Finish amount
Default: 0.0
WC = Width of cut (required)
AN = Pass Angle
Default: 0.0
XS = Start X (no default, not required)
YS = Start Y (no default, not required)
BC = Boundary clearance
Default: 0.0
DS = First pass level (depth, required)
DC = Depth of cut (required)
DE = Final level (depth, required)
EP = Equalized passes (modal)
0 = Off
1 = On:
IN = Group island—use
0 = Off
1 = On:
Default: 0
IP = Final (island) pass (modal)
0 = Off
1 = On:
AR = Ramp angle
Default: 90.0 (vertical)
UA = Uncut area geometry—use
0 = Off
1 = On:
Default: 0
CC = Climb cut
0 = Conventional
1 = Climb
Default: 1
RS = Ramp from start
0 = Off, use ramp angle
1 = On, ramp from start point
FL = Floor allowance
Default: 0.0
CI = Cut inside out
0 = Off (cut outside in)
1 = On
Default: 1
EW = Equalize width passes
0 = Off
1 = On:
Default: 0
OP = Overlap passes
0 = Off
1 = On:
Default: 0
RM = Rough tolerance method
0 = Off, minimum
1 = On, Mid Tol
Default: 0
RT = Rough tolerance amount (distance)
Default: 0.00005
AC = Corner roll angle
Default: 60.0
RD = Rapid to depth
0 = Off
1 = On:
Default: 0
IT = Island top machining
0 = Full pass
1 = Clear top
Default: 0
LY = Layer
Default: 1
NN = AUM named group name
Default: AUM
UT = Uncut material tolerance
Default: 0.01

FACES[ ]
(Build Faces), *command*. Creates closed faces from existing ruled elements. {Afab}

FACES[E1=$$, E2=$$, SC=##]

E1 = Start surface (required)
Default: Surface range
E2 = End surface (required)
Default: Surface range
SC = Scribe surface
0 = Off
1 = On:
Default: 0

FAUTOSEL[ ]
(Fabrication Automatic Tool Select), *command*. Searches the active group for closed profiles representing common fabrication shapes, automatically selects tools that fit the shapes, and creates tool path for removing material from inside the profiles using the selected tools. {Afab}

FAUTOSEL[RO=##, SQ=##, RE=##,
OB=##, CO=##, PT=##, LO=##, LS=##,
LN=##, RD=##, EA=##, CC=##, TF=##,
TT=##, AC=##]

RO = Round punches
0 = Off
1 = On
SQ = Square punches
0 = Off
1 = On
RE = Rectangular punches
0 = Off
1 = On

OB = Obround punches
    0 = Off
    1 = On
CO = Contouring tools
    0 = Off
    1 = On
PT = Punch tolerance
LO = Lead in/out option
    0 = None
    1 = In
    2 = Out
    3 = Both
LS = Lead in/out style
    0 = Line
    2 = Arc
    3 = Both
LN = Line length
RD = Arc radius
EA = Entry/exit angle
CC = Compensation codes
    0 = Off
    1 = On
TF = Use thinning tolerance
    0 = Off
    1 = On
TT = Thinning tolerance
AC = Corner roll angle

FILTER_USE[ ]
(Filter Use), *command*. Turns the use of the Group/Snap filter on or off.

Note: Filter Use on command is recorded when setting the group/snap filters but appears to have no effect on execution regardless of setting or if omitted entirely.

FILTER_USE[ON=##]

ON = Use of group/snap filters
    0 = Off
    1 = On
    Default: None

FIXUP_51[ ]
(File Fix Up), command. Fixes .PM4 files created between 4/27/94 and the release of version 4.2 of the advanced milling applications.

FLOWLINE[ ]
(Flow Line), *command*. Creates uniform tool path across single surfaces having a cylindrical or spherical shape or an isoparametric tool path in either the U direction or the V direction for a single untrimmed surface element. {Ffm}

FLOWLINE[EL=$$, PD=##, SP=##, SV=##,
    DI=##, DV=##, OA=##, OF=##]

EL = Element number of the surface
    Default: None
PD = Path direction
    0 = U direction
    1 = V direction
    Default: 0
SP = Spacing type
    0 = Distance
    1 = Count
    2 = Mid Tol
    Default: 0
SV = Spacing value
    Default: None
DI = Divisions type
    0 = Distance
    1 = Count
    2 = Mid Tol
    Default: 0
DV = Divisions value
    Default: None
OA = Offset (finish) amount value
    Default: 0
OF = Offset direction
    0 = Pos
    1 = Neg
    Default: 1

FOLD[ ]
(Fold), *command*. Folds geometry in the active group automatically calculating the required bend allowance by entering values for Bend Angle, Thickness, Inside Radius, and K Factor. {Afab}

FOLD[BE=##, FE=##, FL=##, AN=##,
    DI=##, ML=##, TH=##, IR=##, KF=##,
    HC=##, MT=##, AD=##, MO=##]

BE = Base element (required)
    Number of elements in base profile
FE = Face element (required)
    Number of elements in face profile
FL = Fold line element
    Number of line elements on fold line
BS = Bend side
    0 = Inside
    1 = Middle
    2 = Outside
    Default: 1
AN = Angle to bend to (required)
    Default: 0
DI = Direction
    0 = Neg

1 = Pos
Default: 1
ML = Mold lines
0 = Off
1 = 0
Default: 0
TH = Thickness
Default: 0
IR = Inside radius
Default: 0
KF = K factor
Default: 0
HC = Hem compensation
Default: 0
MT = Multiply
Default: 1
AD = Add
Default: 0
MO = Mode
0 = Fold
1 = Flatten
Default: 0

FORM_PATCH[ ]
(Form Patch), *command*. Create sculpted mesh profiles by using four connected boundary profiles that define the surface area. {Amill, Ffm}

FORM_PATCH[E1=$$, E2=$$, E3=$$,
E4=$$, PD=##, SP=##, SV=##, ZT=##,
DI=##, DV=##, OF=##, OA=##, CM=##,
GC=##]

E1 = First director start (required)
Default: None
E2 = First director end (required)
Default: None
E3 = Second director start (required)
Default: None
E4 = Second director end (required)
Default: None
PD = Path direction
0 = Cross
1 = Along
2 = Planar
Default: 0
SP = Spacing
0 = Distance
1 = Count
2 = Cusp Hgt
Default: 0
SV = Spacing value (required)
Default: None
ZT = Planar Z
Default: 0.0
DI = Division
0 = Distance
1 = Count
2 = Mid Tol
3 = In Tol
4 = Out Tol
Default: 0
DV = Division value (required)
Default: None
OF = Offset
0 = Positive
1 = Negative
2 = 0
Default: 2
OA = Finish offset amount
Default: 0
CM = Connect mesh
0 = Off
1 = On:
Default: 0
GC = Gouge correction
0 = Off
1 = On:
Default: 0

FPATH_SET[ ]
(File Path Set), *command*. Changes the default path for a specified file type in the [Paths] section of the application ini file.

Note: File Path Set is not a recordable macro command.

FPATH_SET[IN=$$, FP=$$]

IN = Type of path
ctg = Path for custom tool graphic files
def = Path for current Process Model file
export = path for internal export files (not Cam Connections)
fnt = Path for current font file
jof = Path for current job file
smf = Path for code generator files
import = path for internal import files (not Cam Connections)
FP = Path

FROM_MESH[ ]
(From Mesh), *command*. Creates a surface from an existing mesh. From Mesh uses the value in the Point Set Fitting field of the Surface Modes dialog box to control the tolerance for creating the surface. {Ffm}

FROM_MESH[ES=$$, EE=$$]

ES = Mesh starting element (required)

EE = Mesh ending element (required)

FULL[ ]
(Full), *command*. Redraws the graphic view to display all non-hidden elements of the model.

FULL[]

- G -

GEN_MESH[ ]
(Generate Mesh), *command*. Creates a polyline mesh from a surface. {Ffm}

GEN_MESH[EL=$$, PD=##, SP=##,
 SV=##, DI=##, DV=##]

EL = Element number of the surface
 Default: None
PD = Path direction arguments
 0 = U direction
 1 = V direction
 Default: 0
SP = Spacing type
 0 = Distance
 1 = Count
 2 = Mid Tol
 Default: 0
SV = Spacing value
 Default: None
DI = Division type
 0 = Distance
 1 = Count
 2 = Mid Tol
 Default: 0
DV = Division value
 Default: None

GET_GRP[ ]
(Get Group), *command*. Adds geometry previously assigned to a named group to the active group.

GET_GRP[GN=$$]

GN = Group name (required)
 Current group name
 Default: None

GET_NAME( )
(Get Name), *function*. Returns the name and extension of the filename.

Note: Get Name is not a recordable macro function.

GET_NAME(strexpr)
 strexpr = File pathname

GET_PATH( )
(Get Path), *function*. Returns path portion of filename.

Note: Get Path is not a recordable macro function.

GET_PATH(strexpr)
 strexpr = File pathname

GET_VIEW[ ]
(Get View), *command*. Displays any existing views from the Views list.

GET_VIEW[VN=$$]

VN = View name (required)
 Default: None

GOTO( )
(Go To), *command*. Redirects the execution of macro commands to another section of the macro. Part of a command set that includes GOTO( ), @label.

Note: Go To is not a recordable macro command. Each GOTO command must have a corresponding label within the macro. The GOTO command can only advance forward in the macro and cannot be made to loop backward.

GOTO(label)

@label

label = Name identifying a section of a
 macro (required)

GRAPHICS_OFF[ ]
(Graphics Off), *command*. Disables the redrawing of the graphics and list view to speed up long-processing macros. The graphics and list view will remain disabled even after the macro terminates until the GRAPHICS_ON[] command is issued.

Note: Graphics Off is not a recordable macro command.

GRAPHICS_OFF[]

GRAPHICS_ON[ ]
(Graphics On), *command*. Enables the redrawing of the graphics and list view after being disabled with the GRAPHICS_OFF[] command.

Note: Graphics On is not a recordable macro command.

```
GRAPHICS_ON[]
```

GROOVE[ ]
(Groove), *command*. Creates tool path to cut a groove on the inside diameter, outside diameter, or face of a part. {Pturn, Aturn}

```
GROOVE[FX=#, FY=#, DW=#, DD=#,
    DA=#, DM=#, P1=#, P2=#, P3=#, P4=#,
    R1=#, R2=#, R3=#, R4=#, TS=#]
```

FX = Anchor point X ordinate
    Default: Null
FY= Anchor point Y ordinate
    Default: Null
DW = Groove Width
    Default: Null
DD = Groove Depth
    Default: Null
DA = Finish Allowance
    Default: Null
DM = Groove max width of cut
    Default: Width of tool from Job Plan
P1 = T1 corner feature
    0 = Sharp
    1 = Radius
    2 = Chamfer
    Default: Null
P2 = T2 corner feature
    0 = Sharp
    1 = Radius
    2 = Chamfer
    Default: Null
P3 = B1 corner feature
    0 = Sharp
    1 = Radius
    2 = Chamfer
    Default: Radius is 1
P4 = B2 corner feature
    0 = Sharp
    1 = Radius
    2 = Chamfer
    Default: Radius is 1
R1 = T1 size value
    Default: Null
R2 = T2 size value
    Default: Null
R3 = B1 size value
    Default: Corner radius specified in Job Plan
R4 = B2 size value
    Default: Corner radius specified in Job Plan
TS = To size
    Options: 0 = Off
    1 = On:
    Default: 0

GROOVE_RGH[ ]
(Groove Rough), *command*. Creates tool path with straight-plunge roughing tool path to an ID, OD, or face profile. This command has been superceded by the TGROOVE[] macro command. An attempt has been made to make the former command forward compatible with newer command. {Pturn, Atrun}

Note: Groove Rough is not a recordable macro command.

```
GROOVE_RGH[ES=##, EE=##, XS=##,
    YS=##, EM=##, ZF=##, XF=##, DW=##,
    CL=##, CP=##, OP=##]
```

ES = Finish profile start (required)
    Default: None
EE = Finish profile end (required)
    Default: None
XS = Z start (required)
    Default: None
YS = X start (required)
    Default: None
EM = Material element (required)
    Default: None
ZF = Z finish amount (required)
    Default: None
XF = X finish amount (required)
    Default: None
DW = Pass width (required)
    Default: None
CL = Clear (required)
    Default: None
CP = Cleanup pass binary
    0 = Off
    1 = On
    Default: 0

GRP( )
(Grouped element Index), *function*. Returns the element number of an element in the active group or the total number of elements in the active group.

GRP(index)
index = 0, returns the number of elements in the active group or –1 If there is no group.
index > 0, returns the element number of the nth element in the active group or –1 If the number is out of range.

GRP_BLEND[ ]
(Group Blend), *command*. Adds or modifies blend elements (arcs) to corner elements in an active group of elements.

GRP_BLEND[RA=##, RO=##, RC=##]

RA = Inside radius (required)
Default: Null
RO = Outside radius
Default: Null
RC = Change radius
Default: Null

GRP_CHAIN[ ]
(Group Chain), *command*. Converts connected elements in the active group into contiguous profiles of sequential elements or polylines.

GRP_CHAIN[PJ=##]

PJ = Polyline join
0 = Chain
1 = PolyJoin
2 = Both
Default: 0

GRP_DELETE[ ]
(Group Delete), *command*. Deletes all elements in the active group.

GRP_DELETE[ ]

GRP_FILT_ADD[ ]
(Group Filter Add), *command*. Adds all elements that satisfy the filter criteria to the active group.

GRP_FILT_ADD[ ]

GRP_FILT_REMOVE[ ]
(Group Filter Remove), *command*. Removes all elements that satisfy the filter criteria from the active group.

GRP_FILT_REMOVE[ ]

GRP_POCK[ ]
(Group Pocket), *command*. Creates tool path for removing material from the inside of a group of closed profiles. {Pmill, Amill, Ffm, Aturn}

GRP_POCK[TY=##, FA=##, WC=##, AN=##, DC=##, EP=##, AR=##, UA=##, CC=##, RS=##, FL=##, CI=##, EW=##, OP=##, RM=##, RT=##, AC=##, RD=##, LY=##, NN=$$, UT=##]

TY = Pocketing method
0 = Spiral
1 = Zigzag
2 = Linear
Default: 0
FA = Finish amount
Default: 0.0
WC = Width of cut (required)
AN = Pass angle
Default: 0.0
DC = Depth of cut (required)
EP = Equalized passes (modal)
0 = Off
1 = On
AR = Ramp angle
Default: 90.0 (vertical)
UA = Uncut area geometry—use
0 = Off
1 = On:
Default: 0
CC = Climb cut
0 = Conventional
1 = Climb
Default: 1
RS = Ramp from start
0 = Off, use ramp angle
1 = On, ramp from start point
FL = Floor allowance
Default: 0.0
CI = Cut inside out
0 = Off (cut outside in)
1 = On
Default: 1
EW = Equalize width passes
0 = Off
1 = On
Default: 0
OP = Overlap passes
0 = Off
1 = On
Default: 0
RM = Rough tolerance method
0 = Off, minimum
1 = On, Mid Tol
Default: 0

RT = Rough tolerance amount (distance)
Default: 0.00005
AC = Corner roll angle
Default: 60.0
RD = Rapid to depth
0 = Off
1 = On
Default: 0
LY = Layer
Default: 1
NN = AUM named group name
Default: AUM
UT = Uncut material tolerance
Default: 0.01

GROUP_POLYARC_FIT[ ]
(Group Polyarc Fit), *command*. Creates a series of lines and arcs that approximate a smooth fit, within desired tolerances, over polylines in the active group.

GROUP_POLYARC_FIT[TO=##, SA=##,
LL=##, KP=##, MP=##]

TO = Fit tolerance (required)
Default: 0.0005
SA = Sharp angle (required)
Default: 34.0
LL = Long line (required)
Default: 17.0
KP = Keep original code (required)
0 = erase polyline from database and replace by polyarc profile
1 =keep original code
Default: 1
MP = Match properties code (required)
0 = Use active properties
1 = Use polyline
Default: 1

GRP_REVERSE[ ]
(Group Reverse), *command*. Reverses the direction and/or database order of profiles in the active group.

GRP_REVERSE[RP=##]

RP = Reverse profile
0 = Order and direction
1 = Direction only
2 = Order only
Default: 0

GRP_TRIM[ ]
(Group Trim), *command*. Trims or splits the active group of elements at the intersections with a selected profile or element.

GRP_TRIM[EL=$$, S1=##, TS=##, TY=##,
TO=##]

EL = Element in trimming profile (required)
Default: None
S1 = Keep side (required)
0 = Right
1 = Left
Default: None
TS = Trim/split
0 = Trim
1 = Split
Default: 1
TY = Type
0 = 2D
1 = 3D
Default: None
TO = Intersect tolerance
Default: None

GRP_WALL[ ]
(Group Wall), *command*. Creates parallel geometry that is offset from an existing, group of elements or profiles.

GRP_WALL[SD=##, DS=##, ME=##,
RP=##, LR=##, TA=##, TO=##, AN=##]

SD = Side (required)
0 = Left
1 = Right
Default: None
DS = Distance (required)
Default: None
ME = Match element’s properties
0 = No
1 = Yes
Default: 0
RP = Repeat
1–999
Default: 1
LR = Loop removal
0 = No
1 = Yes
Default: 0
TA = Tolerance application
0 = Off
1 = On:
Default: 0
TO = Tolerance amount
Default: 0.000005 [0.001]
AN = Corner roll angle
5–180
Default: 180

- H -

HELIX[ ]
(Helix), *command*. Creates helical and spiral curves that travel at a constant or changing radius along a defined axis.

HELIX[XS=##, YS=##, ZS=##, XE=##,
YE=##, ZE=##, XC=##, YC=##, AS=##,
AE=##, AT=##, RS=##, RE=##, DR=##,
DS=##, FT=##, PT=##]

XS = X start
Default: None
YS = Y start
Default: None
ZS = Z start
Default: None
XE = X end
Default: None
YE = Y end
Default: None
ZE = Z end
Default: None
XC = X center (required)
Default: None
YC = Y center (required)
Default: None
AS = Angle start
Default: None
AE = Angle end
Default: None
AT = Angle total
Default: None
RS = Radius start
Default: None
RE = Radius end
Default: None
DR = Direction
0 = CW, 1= CCW
Default: 0
DS = Distance (axis length)
Default: None
FT = Full turns
Default: None
PT = Pitch
Default: None

HILITE[ ]
(Highlight Item), *script*. Highlights an item in the user interface.

Note: Highlight Item only functions while running in the macro demo mode without using the default UI scripts.

HILITE[$$]

$$ = Name of main menu, pull down or
workbench item

HOLE[ ]
(Hole), *command*. Creates a hole element at the selected location. {Pmill, Amill, Ffm, Pturn, Aturn, Awedm}

HOLE[XE=##, YE=##, XS=##, YS=##,
AN=##, DS=##, LV=##, RI=##, SE=##]

XE = X end
Default: Null
YE = Y end
Default: Null
XS = X anchor
Default: Null
YS = Y anchor
Default: Null
AN = Angle
Default: Null
DS = Distance
Default: Null
LV = Level (modal)
RI = Radius input
0 = No
1 = Yes
SE = End solution designator
0 = +
1 = -
2 = N/A
Default: 0
Note: See PROPERTY to output other variables as needed.

HOLE_ALL[ ]
(All Holes From Solid), *command*. Creates one or multiple hole elements using all the cylinders contained in an ACIS solid body. {Amill, Ffm}

HOLE_ALL[SE=##, WS=##, MA=##,
SC=##, MD=##, HE=##, DI=##, TO=##]

SE = Surface element. Element index.
(required) Indicates the search area.
WS = Entire Solid
1 = Yes
0 = No
Default: 1
MA = Match Axis
0 = No
1 = Yes
Default: 1

SC = Search split cylinders
    0 = No
    1 = Yes
    Default: 0
MD = Match diameter
    0 = No
    1 = By example
    2 = By value
    Default: 0
HE = Cylindrical surface element. Element
    index. (required if MD = 1)
DI = diameter Positive floating point value
    Default = 1.0
TO = Match diameter tolerance
    Default: 0.001

HOLE_ONE[ ]
(Single Hole From Solid), *command*. Creates a hole elements using a cylinder contained in an ACIS solid body. {Amill, Ffm}

HOLE_ONE[SE=##, HI=##]

SE = Surface element Indicates the search
    area.
HI = Hole index. Index of the hole within the
    search area.

HOTSAVE[ ]
(Save), *command*. Saves the current
    process model and job operations setup
    file to disk with the designated file name
    and type.

Note: No message box will be displayed prompting to overwrite If the file already exists on disk.

HOTSAVE[FN=$$, FT=##]

FN = Filename
    Default: None
FT = File type
    Non-ACIS SmartCAM products
    0 = 2-D (.sh2)
    1 = 3-D (.sh3)
    2 = V6.x CAM Connect (.sh3)
    3 = Process Model (.pm4)
    Default: 3
    FreeForm Machining and Advanced
    Milling only
    0 = 2-D (.sh2)
    1 = 3-D (.sh3)
    2 = V6.x CAM Connect (.sh3)
    3 = Process Model SAB (.pm4)
    4 = No Surfaces (.pm4)
    5 = FreeForm V4.6 (.pm4)
    6 = Process Model SAT (.pm4)
    Default: 4

- I -

IF( )
(Logical If), *command*. Performs a conditional test for the branching of operations within a macro. Part of a command set that includes IF( ), ELSE, ENDIF.

Note: Logical If is not a recordable macro command. Each IF command must be followed by a corresponding ENDIF command within the macro. The ELSE command is optional.

```
IF(expression)
    true
ELSE
    false
ENDIF
```

expression = A logic expression that may
    contain the following operators

=	(equal)
<>	(not equal)
>	(greater than)
>=	(greater than or equal to)
<	(less than)
<=	(less than or equal to)
,AND	(logical and)
,OR	(logical or)

true = Macro commands to be performed if
    the logic expression proves true.
False = Macro commands to be performed if
    the logic expression proves false

IMPORT[ ]
(Import), *command*. Imports external geometric models from ACIS, ATS, DXF, DWG, IGES, and VDA-FS files into a SmartCAM process model file.

IMPORT[FN=$$, FTS=$$, EX=##, LU=##,
    LN=$$, SF=$$, TF=$$]

FN = From Filename (required)
FTS = Import file type
    SAT
    SAB
    ATS
    DWG

DXF
IGESW or IGES
IGSS
VDAFS

EX = Export Solids
0 = On
1 = Off
MR = Model resolution choice (required)
0 = Default
1 = IGES Global Value
2 = ACIS RESABS
3 = User constant value
MV = Model resolution user constant value (required)
EX = Explode solids (required)
0 = No explode
1 = Explode
Default: 0
LU = Log file—use (optional)
0 = No log file written
1 = Write log file
Default: 0
LN = Log file name (required)
SF = Setup File Name (required for CAM Connection wireframe file types DWG, DXF, IGES. Optional for other file types.)
Default: from .ini file
TF = Temporary CAM Connection Output File Name. (required for CAM Connection file types DWG, DXF, IGES Wireframe. Optional for other file types.)
Default: from .ini file

INCREMENT[ ]
(Increment), *command*. Sets the precision of coordinates and pointer movement in the graphic view, the pick distances for Snap, and the default behavior for the Snap Z-level setting.

```
INCREMENT[DS=##, AN=##, PL=##,
    TO=##, DZ=##, SZ=##]
```

DS = Linear increment (distance)
Default: N/C
AN = Angular increment
Default: N/C
PL = Pick limit
Default: N/C
TO = Pick tolerance
Default: N/C
DZ = Default Z
0 = Off
1 = On:
Default: 1
SZ = Snap Z
0 = Off
1 = On:
Default: 1

INT( )
(Integer), *function*. Returns the integer value of a numeric expression.

```
INT(numexp)
```

INTEGER:
(Declare Variable as Integer), *command*. Declares the variable that follows as a integer type.

Note: Declare Variable as Integer is not a recordable macro command. A variable must be declared as an integer prior to being assigned a value. Used primarily to manipulate variables shared by control panels and dialog boxes.

```
INTEGER:variable
```

INTERSECT[ ]
(Intersect), *command*. Trims the elements in the active group where a plane, cone, cylinder, sphere, or mesh intersects it. {Amill, Ffm}

```
INTERSECT[TY=$$, E1=$$, E2=$$, SD=##,
    OA=##, KP=##, VT=##]
```

TY = Type
0 = Plane
1 = Cone
2 = Sphere
3 = Mesh
Default: 0
E1 = Generator/start element (required)
Default: None
E2 = Director/end element (required)
Default: None
SD = Keep side (required)
0 = Positive
1 = Negative
Default: None
OA = Offset amount
Default: 0
KP = Keep pieces
0 = All
1 = First
2 = Last
Default: 0

VT = Virtual mesh tolerance
    Default: Global variable
    Initial default: 0.001

INTPT[ ]
(Intersection Points), *command*. Creates point elements at the intersections of the selected line or arc elements.

Note: Intersection Points is not a recordable macro command.

INTPT[E1=$$, E2=$$]

E1 = First element (required)
    Default: None
E2 = Second element (required)
    Default: None

- J -

JOBFILE( )
(Job File), *function*. Returns the job file path plus job file name.

Note: Job File is not a recordable macro function.

JOBFILE()

JOBINFO[ ]
(Print Job Information), *command*. Prints the job information or creates a report file that contains job information for the open process model.

JOBINFO[OM=##, FN=##, TS=##]

OM = Output method
    0 = Printer
    1 = File
FN = Filename (required for OM=1)
TS = Tool setup
    0 = No tool setup
    1 = Tool
    Default: 0

JOBPLN[ ]
(Job Plan), *command*. Attaches a different job file to the current process model.

JOBPLN[FN=$$, JT=##]

FN = Filename (required)
JT = Job type
    0 = Read `.jsf` and convert to `.jof`
    1 = Read `.jof`
    Note: JT is ignored in `.jsf` mode and defaults to 0 in `.jof` mode.

JOS( )
(Job Operation Setup), *function*. Returns the numeric value of a jos data tag.

JOS(tag)
    tag = Any valid numeric data tag for information in the planner.

JOS_CONTEXT_CLOSE[ ]
(Jos Context Close), *command*. Closes a JOS data context opened by a call to one of the three JOS context open calls: JOS_CONTEXT_OPEN, JOS_CONTEXT_DUP, or JOS_CONTEXT_NEW. Must be paired with one of these calls.

JOS_CONTEXT_CLOSE[]

JOS_CONTEXT_DUP[ ]
(Jos Context Duplicate), *command*. Opens a JOS context for editing by duplicating an item. Must be paired with a JOS_CONTEXT_CLOSE[] call.

JOS_CONTEXT_DUP[BASEITEM=##,
    USERNAME=$$]

BASEITEM = Basic Item for the Duplicate
    (required)
    1 = Step (Duplicate a Step)
    2 = Tool (Duplicate a tool assignment).
    Default: None
USERNAME = User name of the basic item
    to be duplicated. (required)
    The Step Number of the step to duplicate for steps or the Tool Number for the Tool to duplicate for Tools.
    Default: None.

JOS_ITEM_DUPLICATE[ ]
(Jos Item Duplicate), *command*. Creates a new item in context by duplicating an existing item. The current item of the specified type is replaced in context.

JOS_ITEM_DUPLICATE[BASEITEM=##,
    USERNAME=$$]

BASEITEM = Basic Item for the Duplicate
    (required)
    2 = Tool (duplicate a tool assignment).
    Default: None.

USERNAME = User name of the basic item
to be duplicated. (required)
The Tool Number for the Tool to
duplicate for Tools.
Default: None.

JOS_FIELD_EDIT[ ]
(Jos Field Edit), *command*. Changes a JOS value. This must be within paired calls to one of the three JOS context open macros; JOS_CONTEXT_OPEN, JOS_CONTEXT_DUP, or JOS_CONTEXT_NEW and the JOS context close call: JOS_CONTEXT_CLOSE. Will not be retained as a JOS value unless followed by a JOS_CHANGES_PROMOTE call within the same JOS context changes pairing.

JOS_FIELD_EDIT[FLDNAME=$$,
EXPR=$$]

FLDNAME = The JOS data tag of the field to
be changed. (required)
Possible values are given in the JOS
Expressions specification.
Default: None.
EXPR = The expression to set the value to.
(required)
Any valid expression given as a string.
Default: None.

JOS_ITEM_EDIT[ ]
(Jos Item Edit), *command*. Brings an item into a JOS context for editing replacing the currently active item in the structure.

JOS_ITEM_EDIT[BASEITEM=##,
USERNAME=$$]

BASEITEM = Basic Item for the Edit
(required)
2 = Tool (edit a tool assignment).
Default: None
USERNAME = User name of the basic item
to be edited. (required)
The Tool Number for the Tool to edit for
Tools.
Default: None.

JOS_CONTEXT_NEW[ ]
(Jos Context New), *command*. Opens a JOS context for editing by creating a new item of the specified type. Must be paired with a JOS_CONTEXT_CLOSE[] call.

JOS_CONTEXT_NEW[BASEITEM=##,
NEWSNCS=##, USERNAME=$$]

BASEITEM = Basic Item for the New
(required)
1 = Step (Create a Step)
2 = Tool (Create a tool assignment).
Default: None
NEWSNCS = The Section number of the
item to create. This provides defaults
and specifies the user interface for the
item. (required)
For Steps this will be an Op Section
number (generally in the 1030000 -
1039999 range) as defined in the Global
Specification
Shelf/Notebook/Chapter/Section. For
Tools this will be a Tool Assembly
Section number (generally in the range
2410000 - 2499999) as defined in the
Global Specification
Shelf/Notebook/Chapter/Section.
Default: None
USERNAME = User name of the basic item
after which the new item will be inserted.
If this is empty the item is added at the
end of the list.
The Step Number of the insert step for
steps or the Tool Number for the insert
Tool to for Tool s.
Default: Empty String.

JOS_ITEM_NEW[ ]
(Jos Item New), *command*. Creates a new item in context from the specified Section number defaults.

JOS_ITEM_NEW[BASEITEM=##,
NEWSNCS=##]

BASEITEM = Basic Item for the New
(required)
2 = Tool (create a new Tool
assignment).
8 = Operation (create a new Operation)
Default: None.
NEWSNCS = The Section number of the
item to create. This provides defaults
and specifies the user interface for the
item. (required)
For Operations this will be an Op
Section number (generally in the
1030000 - 1039999 range) as defined in
the Global Specification
Shelf/Notebook/Chapter/Section. For
Tools this will be a Tool Assembly
Section number (generally in the range

2410000 -2499999) as defined in the Global Specification Shelf/Notebook/Chapter/Section.
Default: None

JOS_CONTEXT_OPEN[ ]
(Jos Context Open), *command*. Opens a JOS context for editing. Must be paired with a JOS_CONTEXT_CLOSE[] call.

JOS_CONTEXT_OPEN[BASEITEM=##, USERNAME=$$]

BASEITEM = Basic Item for the Edit (required)
1 = Step (Open a Step changes context)
2 = Tool (open a tool assignment edit context)
6 = Job Info (open a job information edit context).
Default: None
USERNAME = User name of the basic item to be edited. (required)
The Step Number of the step to edit for steps, the Tool Number for the Tool to edit for Tools, or an empty string for Job Information editing.
Default: None

JOS_CHANGES_PROMOTE[ ]
(Jos Changes Promote), *command*. Commits all active changes to a JOS context to the current JOS data. This must be within paired calls to one of the three JOS context open macros: JOS_CONTEXT_OPEN, JOS_CONTEXT_DUP, or JOS_CONTEXT_NEW and the JOS context close call: JOS_CONTEXT_CLOSE.

JOS_CHANGES_PROMOTE[]

JOS_STR( )
(Job Operations Setup String), *function*. Returns the string value of a jos data tag.

JOS_STR(tag)
tag = Any valid string data value from the planner

JTOOL( )
(Job Tool), *function*. Returns tool number of a specified element.

JTOOL(elnum) Returns tool number of a specified element.

JTOOL_SEQ[ ]
(Job Tool Sequence), *command*. Sets the insert properties to insert new elements before the first use, or after the last use of the identified tool.
Note: This replaces TOOL SEQUENCE in applications using Job Operations Setup.

JTOOL_SEQ[BA=##, TL=##, ME=##]

BA = Before/after
0 = Before
1 = After
Default: 1
TL = Tool (required)
Valid tool number
Default: None
ME = Match element (modal)
0 = Off
1 = On

- K -

KILL_PLANE[ ]
(Kill Plane). *command*. Deletes un-reserved work planes that do not have elements associated with them.

KILL_PLANE[WP=$$]

WP = Plane name
? = Wild character
* = Wildcard (kill all unused planes)

- L -

LAST_VIEW[ ]
(Last View), *command*. Returns the graphics to the previously active view.

LAST_VIEW[]

LAY( )
(Layer), *function*. Returns layer assignment of selected element.

LAY(elnum)
-1 = when element is with step

LAYER_GRP[ ]
(Layer Group), *command*. Adds or removes elements to the active group according to their layer assignment.

LAYER_GRP[LY=##, AR=##]

LY = Layer (required)
  Default: None
AR = Add/remove (modal)
  0 = Add
  1 = Remove

LAYER_SEQ[ ]
(Layer Sequence), *command*. Sets the insert properties to insert new elements before the first use, or after the last use of the identified layer.

LAYER_SEQ[BA=##, LY=##, ME=##]

BA = Before/after
  0 = Before
  1 = After
  Default: 1
LY = Layer (required)
  Valid layer number
  Default: None
ME = Match element (modal)
  0 = Off
  1 = On

LEAD_IN[ ]
(Lead In/Out), *command*. Creates lines or arcs to produce gradual tool-feed movement into and out of a profile.

LEAD_IN[EL=$$, AN=##, DS=##, RA=##,
  D2=##, LC=##, TY=##, CS=##, RX=##,
  RY=##, RP=##, MO=##]

EL = Element (required)
  Default: None
AN = Angle
  Default: Null
DS = Distance (length of line)
  Default: Null
RA = Radius (for arc)
  Default: Null
D2 = Distance perpendicular
  Default: Null
LC = Location
  0 = In
  1 = Out
  2 = Both
  Default: 2
TY = Type
  0 = Line
  1 = Arc
  2 = Both
  Default: 0
CS = Change start
  0 = No
  1 = Yes
  Default: 0
RX = Reference Point X
  Default: None
RY = Reference Point Y
  Default: None
RP = Use reference point
  0 = No
  1= Yes
  Default: 0
MO = Line offset match
  0 = Off, no offset line
  1 = OFFSET on lines match profile
  offset
  Default: Off

LEAVE[ ]
(Exit), *command*. Closes the SmartCAM application.

LEAVE[]

Note: This does not test for file modification
  in macro
execution.

LEN( )
(Length), *function*. Returns the length value or hole depth of selected element.

LEN(elnum)
  elnum = Element number or name

LENE( )
(Element Path Length), *function*. Returns the path length of selected element.
Note: You cannot use this function on a helix. If you use this in a macro to measure a helix, SmartCAM will display an error message.

LENE(elnum)
  elnum = Element number or name

LENP( )
(Profile Path Length), *function*. Returns the path length of the profile of selected element.
Note: You cannot use this function on a helix. If you use this in a macro to measure a helix, SmartCAM will display an error message.

LENP(elnum)
  elnum = Element number or name

LINE[ ]
(Line), *command*. Creates a line element.

LINE[XS=##, YS=##, XE=##, YE=##, XI=##,
YI=##, AN=##, DS=##, TS=$$, TE=$$,
SS=##, SE=##, PK=##]

XS = X start
Default: Null
YS = Y start
Default: Null
XE = X end
Default: Null
YE = Y end
Default: Null
XI = X intermediate
Default: Null
YI = Y intermediate
Default: Null
AN = Angle
Default: Null
DS = Distance (length)
Default: Null
TS = Start tangent element
Arc element number
Default: Null
TE = End tangent element
Arc element number
Default: Null
SS = Start solution designator
0 = +
1 = -
2 = N/A
Default: 0
SE = End solution designator
0 = +
1 = -
2 = N/A
Default: 0
PK = Peck
0 = Off
1 = On:
ZE = Z end
ZI = Z intermediate
ZS = Z start

LINE_PROF[ ]
(Line Profile), *command*. Adds a line as part of a continuous profile.

LINE_PROF[XE=##, YE=##, XI=##, YI=##,
XJ=##, YJ=##, AN=##, DS=##, TI=##,
SS=##, SE=##, SP=##]

XE = X end
Default: Null
YE = Y end
Default: Null
XI = First intermediate X
Default: Null
YI = First intermediate Y
Default: Null
XJ = Second intermediate X
Default: Null
YJ = Second intermediate Y
Default: Null
AN = Angle
Default: Null
DS = Distance (length)
Default: Null
TI = Tangent/intersect to previous element
0 = Intersect
1 =Tangent
Default: 1
SS = Solution start
0 = +
1 = -
2 = N/A
Default: 0
SE = Solution end
0 = +
1 = -
2 = N/A
Default: 0
SP = Solution pending
0 = +
1 = -
2 = N/A
Default: 0

LINEAR_RGH[ ]
(Linear Roughing), *command*. Creates tool path with straight line roughing passes that are parallel to each other and equal distances apart. This macro command has been superceded by the TLINEAR[] macro command. An attempt has been made to make the former command forward compatible with newer command. {Pturn, Aturn}

Note: Linear Roughing is not a recordable macro command.

LINEAR_RGH[TY=##, ES=##, EE=##,
XS=##, YS=##, EM=##, ZF=##, XF=##,
DP=##, CL=##, AN=##, AP=##, CP=##,
OP=##]

TY = Type of operation
0 = Turning
1 = Facing
Default: 0

ES = Finish profile start (required)
Default: None
EE = Finish profile end (required)
Default: None
XS = Z start
Default: None
YS = X start
Default: None
EM = Material element
Default: None
ZF = Z finish amount (required)
Default: None
XF = X finish amount (required)
Default: None
DP = Pass depth (required)
Default: None
CL = Clear (required)
Default: None
AN = Pass angle (required)
Default: None
AP = Plunge angle (required)
Default: None
CP = Cleanup pass
0 = Off
1 = On
Default: 0
OP = Overlap passes binary
0 = Off
1 = On
Default: None

LNUM( )
(Layer Number), *function*. Returns the active layer number or -1 if a step number.

LNUM()

LOFTED[ ]
(Lofted Mesh), *function*. Creates sculpted mesh profiles that travel over a series of sequential cross-section profiles.

LOFTED[XS=$$, YS=$$, ZS=$$, PD=##,
SP=##, SV=##, ZT=##, DI=##, DV=##,
OF=##, OA=##, CM=##, GC=##]

XS = X start
Valid ordinate value
Default: 0.0
YS = Y start
Valid ordinate value
Default: 0.0
ZS = Z start
Valid ordinate value
Default: 0.0
PD = Path direction
0 = Cross
1 = Along
2 = Planar
Default: 0
SP = Spacing (type)
0 = Distance
1 = Count
2 = Cusp Hgt
Default: 0
SV = Spacing value (required)
Default: None
ZT = Planar Z, Z top
Default: 0.0
DI = Division (type)
0 = Distance
1 = Count
2 = Mid Tol
3 = In Tol
4 = Out Tol
Default: 0
DV = Division value (required)
Default: None
OF = Offset
0 = Positive
1 = Negative
2 = 0
Default: 2
OA = Finish offset amount
Default: 0.0
CM = Connect mesh
0 = Off
1 = On
Default: 0
GC = Gouge correction
0 = Off
1 = On
Default: 0

LVL( )
(Level), *function*. Returns level value of selected element.

LVL (elnum)

- M -

MAC_DEMO[ ]
(Macro Demo), *command*. Executes an existing macro including the display of menu picks and panel interaction.

MAC_DEMO[FN=$$, SP=##, DS=##]

FN = Filename (required)
Default: None

SP = Speed (required)
  0–9
  Default: None
DS = Default UI script (required)
  0 = Off
  1 = On
  Default: None

MAC_EXE[ ]
(Macro Execute), *command*. Executes an existing macro without displaying the menu picks or panel interaction.

MAC_EXE[FN = $$]

FN = Filename (required)
  Default: None

MASK[ ]
(Show/Mask), *command*. Will hide or show elements or to mask elements by Z level in the graphic view.

MASK[ST=##,TL=##, LY=##, WP=##,
  SH=##, ZT=##, ZB=##]

ST = STEP
  Valid step number, step range, or wildcard
  Default: Null
TL = Tool
  Valid tool number, tool range, or wildcard
  Default: Null
LY = Layer
  Valid layer number, layer range, or wildcard
  Default: Null
WP = Work plane
  Default: Null
SH = Show/hide
  0 = Hide
  1 = Show
  Default: 1
ZT = Mask Z top
ZB = Mask Z Bottom
  Note: MASK[ ] shows all Z levels.

MCLFILE( )
(Macro File), *function*. Returns the macro path plus current executing macro name.

MCLFILE()

MEAS_2D_EL( )
(Measure 2D to Element), *function*. Returns the 2D distance between a point and an element.

MEAS_2D_EL(x, y, elnum)

x = 1st Point X (required)
  Default: None
y = 1st Point Y (required)
  Default: None
elnum = Element name or number (required)
  Default: None

MEAS_2D_PT( )
(Measure 2D to Point), *function*. Returns the 2D distance between two points.

MEAS_2D_PT(x1, y1, x2, y2)

x1 = 1st Point X (required)
  Default: None
y1 = 1st Point Y (required)
  Default: None
x2 = 2nd Point X (required)
  Default: None
y2 = 2nd Point Y (required)
  Default: None

MEAS_3D_PT( )
(Measure 3D to Point), *function*. Returns the 3D distance between two points.

MEAS_3D_PT(x1, y1, z1, x2, y2, z2)

x1 = 1st Point X (required)
  Default: None
y1 = 1st Point Y (required)
  Default: None
z1 = 1st Point Z (required)
  Default: None
x2 = 2nd Point X (required)
  Default: None
y2 = 2nd Point Y (required)
  Default: None
z2 = 2nd Point Z (required)
  Default: None

MERGE[ ]
(Merge), *command*. Inserts geometry from a model file on disk into the current process model.

MERGE[FN=$$, FT=##, XS=##, YS=##,
  ZS=##]

FN = Filename (required)

FT = File type
 0 = 2-D (`.sh2`)
 1 = 3-D (`.sh3`)
 2 = PM4 (`.pm4`)
 Default: 1
XS = Base point (required)
YS = Base point (required)
ZS = Base point (required)

MERGE_PLANE[ ]
(Merge Plane), *command*. Combines elements associated with one or more selected work planes with the first parallel work plane found in the dialog box list.

MERGE_PLANE[WP=$$, MO=##]

WP = Plane name
 ? = Wild character
 * = Wildcard
MO = Merge opposite
 0 = Off
 1 = On
 Default: 0

MIRROR[ ]
(Mirror), *command*. Creates a reverse, or "mirror," image of the elements in the active group.

MIRROR[XS=##, YS=##, ZS=##, XE=##,
 YE=##, ZE=##, XI=##, YI=##, ZI=##,
 CP=##, TS=##, RP=##, TY=##, SP=##]

XS = 1st Point X (required)
 Default: None
YS = 1st Point Y (required)
 Default: None
ZS = 1st Point Z (required for 3D)
 Default: None
XE = 2nd Point X (required)
 Default: None
YE = 2nd Point Y (required)
 Default: None
ZE = 2nd Point Z (required for 3D)
 Default: None
XI = 3rd Point X (required for 3D)
 Default: None
YI = 3rd Point Y (required for 3D)
 Default: None
ZI = 3rd Point Z (required for 3D)
 Default: None
CP = Copy
 0 = No
 1 = Yes
 Default: 0
TS = Tool sort
 0 = Off
 1 = On
 Default: 1
RP = Reverse Order and Direction
 0 = No
 1 = Yes
TY = Type
 0 = 2D
 1 = 3D
SP = Suppress Planes
 0 = No
 1 = Yes

MO_SPLINE[ ]
(Modify Spline), *command*. Changes the geometric information for an existing spline element.

MO_SPLINE[EL=$$, IS=##, JS=##, KS=##,
 IE=##, JE=##, KE=##, DS=##, DE=##,
 CE=##, CL={ } or #CLVAR]

EL = Element number or name
IS = Start vector I
 Default: Null
JS = Start vector J
 Default: Null
KS = Start vector K
 Default: Null
IE = End vector I
 Default: Null
JE = End vector J
 Default: Null
KE = End vector K
 Default: Null
DS = Start vector distance (length)
 Default: 0.0
DE = End vector distance (length)
 Default: 0.0
CE = Close end
 0 = No
 1 = Yes
 Default: 0
CL = Coordinate list (required)
 A coordinate list or a variable declared as a coordinate list
 Default: None

MOD_ARC[ ]
(Modify Arc), *command*. Changes the geometric information for an existing selected arc element.

MOD_ARC[EL=$$, XC=##, YC=##, XS=##,
 YS=##, XE=##, YE=##, XI=##, YI=##,
 SS=##, SC=##, TS=$$, AS=##, SE=##,

TE=$$, AE=##, SI=##, TI=$$, DR=##,
RA=##, LV=##]

EL = Element number or name
XC = X center
Default: Null
YC = Y center
Default: Null
XS = X start
Default: Null
YS = Y start
Default: Null
XE = X end
Default: Null
YE = Y end
Default: Null
XI = X intermediate
Default: Null
YI = Y intermediate
Default: Null
SS = Start solution designator
0 = +
1 = -
2 = N/A
Default: 0
SC = Center solution designator
0 = +
1 = -
2 = N/A
Default: 0
TS = Start tangent
Arc or line element number
Default: Null
AS = Start angle
Default: Null
SE = End solution designator
0 = +
1 = -
2 = N/A
Default: 0
TE = End tangent
Arc or line element number
Default: Null
AE = End angle
Default: Null
SI = Intermediate solution designator
0 = +
1 = -
2 = N/A
Default: 0
TI = Intermediate tangent
Arc or line element number
Default: Null
DR = Direction
0 = CW
1 = CCW
Default: 0
RA = Radius
Default: Null
LV = Level (modal)

MOD_DRLS[ ]

(Modify Drill Subroutine), *command*. Changes the geometric information for an existing selected drill call element. {Pmill, Amill, Ffm}

MOD_DRLS[EL=$$, NN=##, CP=##,
XE=##, YE=##, ZE=##, PK=##]

EL = Element number or name
NN = Name of subroutine
Default: None
CP = Number of repeats
Default: 0
XE = Insert point X coordinates
YE = Insert point Y coordinates
ZE = Insert point Z coordinates
PK = Peck
0 = Off
1 = On:

MOD_ELLIPSE[ ]

(Modify Ellipse), *command*. Changes the geometric information for an existing selected ellipse element.

MOD_ELLIPSE[EL=$$, D1=##, D2=##,
XC=##, YC=##, XS=##, YS=##, XE=##,
YE=##, AS=##, AE=##, AN=##, DR=##,
SS=##, SE=##, LV=##]

EL = Element number or name
D1 = Primary distance (required)
Default: None
D2 = Secondary distance (required)
Default: None
XC = X center (required)
Default: None
YC = Y center (required)
Default: None
XS = X start
Default: Null
YS = Y start
Default: Null
XE = X end
Default: Null
YE = Y end
Default: Null
AS = Angle start
Default: Null
AE = Angle end
Default: Null

AN = Angle of inclination
  Default: 0.0
DR = Direction
  0 = CW
  1 = CCW
  Default: 0
SS = Solution start
  0 = +
  1 = -
  2 = N/A
  Default: 0
SE = Solution end
  0 = +
  1 = -
  2 = N/A
  Default: 0
LV = Level (modal)

MOD_FAB_HOLE[ ]
(Modify Fab Hole), command. Changes the geometric information for an existing selected fab hole element.

MOD_FAB_HOLE[EL=$$, XE=##, YE=##,
  XS=##, YS=##, AE=##, AN=##, DS=##,
  LV=##, RI=##, SE=##]

EL = Element number or name
XE = X end
  Default: Null
YE = Y end
  Default: Null
XS = X anchor
  Default: Null
YS = Y anchor
  Default: Null
AE = Tool angle
  Default: Null
AN = Angle
  Default: Null
DS = Distance
  Default: Null
LV = Level
  Default: Modal
RI = Radius input
  0 = No
  1 = Yes
SE = End solution designator
  0 = +
  1 = -
  2 = N/A
  Default: 0

MOD_GROOVE[ ]
(Modify Groove), *command*. Changes the geometric information for an existing selected groove element.

MOD_GROOVE[EL=$$, FX=#, FY=#,
  DW=#, DD=#, DA=#, DM=#, P1=#,
  P2=#, P3=#, P4=#, R1=#, R2=#, R3=#,
  R4=#, TS=#]

EL = Element number or name
FX = Anchor point X ordinate
  Default: Null
FY= Anchor point Y ordinate
  Default: Null
DW = Groove Width
  Default: Null
DD = Groove Depth
  Default: Null
DA = Finish Allowance
  Default: Null
DM = Groove max width of cut
  Default: Width of tool from Job Plan
P1 = T1 corner feature
  0 = Sharp
  1 = Radius
  2 = Chamfer
  Default: Null
P2 = T2 corner feature
  0 = Sharp
  1 = Radius
  2 = Chamfer
  Default: Null
P3 = B1 corner feature
  0 = Sharp
  1 = Radius
  2 = Chamfer
  Default: Radius is 1
P4 = B2 corner feature
  0 = Sharp
  1 = Radius
  2 = Chamfer
  Default: Radius is 1
R1 = T1 size value
  Default: Null
R2 = T2 size value
  Default: Null
R3 = B1 size value
  Default: Corner radius specified in Job Plan
R4 = B2 size value
  Default: Corner radius specified in Job Plan
TS = To size
  Options: 0 = Off
  1 = On:
  Default: 0

MOD_HELIX[ ]
(Modify Helix), *command*. Changes the geometric information for an existing selected helix element.

MOD_HELIX[EL=$$, XS=##, YS=##,
    ZS=##, XE=##, YE=##, ZE=##, XC=##,
    YC=##, AS=##, AE=##, AT=##, RS=##,
    RE=##, DR=##, DS=##, FT=##, PT=##]

EL = Element number or name
XS = X start
    Default: None
YS = Y start
    Default: None
ZS = Z start
    Default: None
XE = X end
    Default: None
YE = Y end
    Default: None
ZE = Z end
    Default: None
XC = X center (required)
    Default: None
YC = Y center (required)
    Default: None
AS = Angle start
    Default: None
AE = Angle end
    Default: None
AT = Angle total
    Default: None
RS = Radius start
    Default: None
RE = Radius end
    Default: None
DR = Direction
    0 = CW, 1= CCW
    Default: 0
DS = Distance (axis length)
    Default: None
FT = Full turns
    Default: None
PT = Pitch
    Default: None

MOD_HOLE[ ]
(Modify Hole), *command*. Changes the geometric information for an existing selected hole element. {Pmill, Amill, Ffm, Pturn, Aturn, Awedm}

MOD_HOLE[EL=$$, XE=##, YE=##,
    XS=##, YS=##, AN=##, DS=##, LV=##,
    RI=##, SE=##]

EL = Element number or name
XE = X end
    Default: Null
YE = Y end
    Default: Null
XS = X anchor
    Default: Null
YS = Y anchor
    Default: Null
AN = Angle
    Default: Null
DS = Distance
    Default: Null
LV = Level (modal)
RI = Radius input
    0 = No
    1 = Yes
SE = End solution designator
    0 = +
    1 = -
    2 = N/A
    Default: 0
    Note: See PROPERTY CHANGE to alter other variables as needed.

MOD_LINE[ ]
(Modify Line), *command*. Changes the geometric information for an existing selected line element.

MOD_LINE[EL=$$, XS=##, YS=##, XE=##,
    YE=##, XI=##, YI=##, AN=##, DS=##,
    TS=$$, TE=$$, SS=##, SE=##, PK=##]

EL = Element number or name
XS = X start
    Default: Null
YS = Y start
    Default: Null
XE = X end
    Default: Null
YE = Y end
    Default: Null
XI = X intermediate
    Default: Null
YI = Y intermediate
    Default: Null
AN = Angle
    Default: Null
DS = Distance (length)
    Default: Null
TS = Start tangent element
    Arc element number
    Default: Null

TE = End tangent element
Arc element number
Default: Null
SS = Start solution designator
0 = +
1 = -
2 = N/A
Default: 0
SE = End solution designator
0 = +
1 = -
2 = N/A
Default: 0
PK = Peck
0 = Off
1 = On:
ZE = Z end
ZI = Z intermediate
ZS = Z start

MOD_POINT[ ]
(Modify Point), *command*. Changes the geometric information for an existing selected point element.

MOD_POINT[EL=$$, XE=##, YE=##,
XS=##, YS=##, AN=##, DS=##, LV=##,
CO=##, SE=##]

EL = Element number or name
XE = X end
Default: Null
YE = Y end
Default: Null
XS = X anchor
Default: Null
YS = Y anchor
Default: Null
AN = Angle
Default: Null
DS = Distance
Default: Null
LV = Level (modal)
CO = Clear Only
0 = Off
1 = On
Default: 0
SE = End solution designator
0 = +
1 = -
2 = N/A
Default: 0

MOD_POLYLINE[ ]
(Modify Polyline), *command*. Changes the geometric information for an existing selected polyline element.

MOD_POLYLINE[EL=$$, CE=##, CL={} or
#CLVAR]
#clvar can be used instead of CL{}.

EL = Element number or name
CE = Close ends
0 = No
1 = Yes
Default: 0
CL = Coordinate list (required)
A coordinate list or a variable declared as a coordinate list
Default: None
Note: See COORDINATE LIST.

MOD_PO5X[ ]
(Modify % Axis Polyline), *command*. Changes the geometric information for an existing selected 5 axis polyline element. {Awedm}

MOD_PO5X[EL=$$, CE=##, CL={x, y, z, u,
v, w, ... x, y, z, u, v, w} or #CLVAR]
#clvar can be used instead of CL{}. The default on macro record is DM = 4

EL = Element number or name
CE = Close ends
0 = No
1 = Yes
Default: 0
CL = Coordinate list (required)
A coordinated list or variable declared as coordinate list.

MOD_SUBROUTINE[ ]
(Modify Subroutine), *command*. Changes the geometric information for an existing selected sub call element. {Pmill, Amill, Ffm, Afab, Awedm}

MOD_SUBROUTINE[EL=$$, NN=##,
CP=##, UE=##, XE=##, YE=##, ZE=##,
AS=##, RR=##]

EL = Element number or name
NN = Name of subroutine (required)
CP = Number of repeats to create
Default: 0
UE = Use properties of grouped elements
0 = No
1 = Yes
XE = Insert point X coordinates
YE = Insert point Y coordinates
ZE = Insert point Z coordinates

AS = Start angle
Default: 0
RR = Rotate repeats
0 = No
1 = Yes
Default: 0

MOD_THREAD[ ]

(Modify Thread), *command*. Changes the geometric information for an existing selected thread element. {Pturn, Aturn}

MOD_THREAD[EL=$$, ND=##, PT=##, YS=##, XS=##, YE=##, XE=##, CX=##, AN=##, AL=##, DS=##, CL=##, TL=##, FN=$$]

EL = Element number or name
ND = Nominal diameter
Default: Null
PT = Pitch
Default: Null
YS = X root start
Default: Null
XS = Z root start
Default: Null
YE = X root end
Default: Null
XE = Z root end
Default: Null
CX = Crest X at start
Default: Null
AN = Taper angle
Default: Null
AL = Axial length
Default: Null
DS = First pass depth
Default: Null
CL = Clear
Default: Null
TL = Thread lead in
Default: Null
FN = Thread table filename
Default: Null
Note: The values required to define a Thread are Root Start and End, Crest, Clearance, Pitch, and depth of first pass. Any of these values can be calculated from other inputs as defined in interactive operations.
Note: If a thread table (FN) is not specified, the macro looks for the default thread table if a nominal diameter (ND) variable is in the option list. Both the ND and FN variables need to be removed from the command list to avoid using a thread table and ensure that the hard-coded values in the command are used as is.
Note: If a thread lookup table is specified, it writes to the fields not specified in the Macro; changing values of entered data if need be.

MOD_TXT[ ]

(Modify Text), *command*. Changes the geometric information for an existing selected text element.

MOD_TXT[EL=$$, FN=$$, TX=$$, XC=##, YC=##, XS=##, YS=##, AN=##, RA=##, HT=##, TY=##, LV=##]

EL = Element number or name
FN = Filename of font file (modal)
TX = Text string (required)
Default: None
XC = X center
Default: Null
YC = Y center
Default: Null
XS = X start
Default: Null
YS = Y start
Default: Null
AN = Angle start
Default: Null
RA = Radius
Default: Null
HT = Height (required)
Default: None
TY = Type (text orientation)
0 = Linear
1 = CW
2 = CCW
Default: 2
LV = Level
Default: Null

MOD_UEVENT[ ]

(Modify User Event), *command*. Changes the geometric information for an existing selected user command element.

MOD_UEVENT[EL=$$, XS=##, YS=##, LV=##, TX=$$]

EL = Element number or name
XS = Start point X coordinate (required)
Default: None
YS = Start point Y coordinate (required)
Default: None
LV = Level
Default: None

TX = Text string (required)
Default: None

MOD_LINK[ ]

(Modify Links), *command*. Changes the start and end points of links between elements that control the wire inclination between the primary and secondary profiles at various points along the profiles. {Awedm}

MOD_LINK[EL=$$, XS=##, YS=##, ZS=##,
XE=##, YE=##, ZE=##]

EL = Element number or name of link to modify (required)
Default: None
XS = X coordinate for link start point
Default: None
YS = Y coordinate for link start point
Default: None
ZS = Z coordinate for link start point
Default: None
XE = X coordinate for link end point (required)
Default: None
YE = Y coordinate for link end point (required)
Default: None
ZE = Z coordinate for link end point (required)
Default: None

MOVE[ ]

(Move), *command*. Changes the location of the active group of elements or, with the Copy option, creates one or more copies to new locations.

MOVE[CP=##, NC=##, TS=##, XS=##,
YS=##, XE=##, YE=##]

CP = Copy mode (overrides NC if false)
0 = Off
1 = On
Default: 0
NC = Num. of copies (requires CP=1 to work)
0–999
Default: 1
TS = Tool sort (modal)
0 = Off
1 = On
XS = X start
Default: 0.0
YS = Y start
Default: 0.0
XE = X end (required)
Default: None
YE = Y end (required)
Default: None

MOVE_REF[ ]

(Move Reference), *command*. Changes the location of the active group of elements or, with the Copy option, move one or more copies while changing from the current work plane to a destination plane.

MOVE_REF[DP=$$, CP=##]

DP = Destination plane (required)
Default: None
CP = Copy mode (overrides NC if false)
0 = Off
1 = On
Default: 0

- N -

NAME_ELMT[ ]

(Name Element), *command*. Assign a name to an element and place the element's name in the element list.

NAME_ELMT[EL=$$, EN=$$]

EL = Element number or name to be named (required)
Default: None
EN = Name of element (required)
Default: None

NAME_GRP[ ]

(Name Group), *command*. Assigns a name to the current active group to retrieve for future use.

NAME_GRP[GN=$$, RM=##]

GN = Group name (required)
Valid name
Default: None
RM = Remove
0 = Off
1 = On
Default: 0

NAME_VIEW[ ]

(Name View), *command*. Assigns a name to the current view orientation.

NAME_VIEW[VN=$$, RM=##]

VN = View name
Default: None
RM = Remove
0 = No
1 = Yes
Default: 0

NEST_MOV[ ]
(Nest Move), *command*. Moves or copies elements in the active group on a two-dimensional plane. {Afab}

NEST_MOV[XS=##, YS=##, XE=##,
YE=##, CP=##, ST=##]

XS = X From Point (required)
Default: None
YS = Y From Point (required)
Default: None
XE = X To Point (required)
Default: None
YE = Y To Point (required)
Default: None
CP = Copy
0 = Move
1 = Copy
ST = Sort by tools
0 = No
1 = Yes

NEST_PAR[ ]
(Nest Parallel), *command*. Moves, rotates, or copies elements in the active group parallel to a reference element on a two-dimensional plane. {Afab}

NEST_PAR[RE=$$, PE=$$, XS=##, YS=##,
CP=##, ST=##, SP=##]

RE = Reference element
Default: None
PE = Parallel element
Default: None
XS = X From Point (required)
Default: None
YS = Y From Point (required)
Default: None
CP = Copy
0 = Move
1 = Copy
ST = Sort by tools
0 = No
1 = Yes
Default: 0
SP = Space

NEST_ROT[ ]
(Nest Rotate), *command*. Rotates or copies elements in the active group around a point on a two-dimensional plane. {Afab}

NEST_ROT[XS=##, YS=##, RI=##, CP=##,
ST=##]

XS = X From Point (required)
Default: None
YS = Y From Point (required)
Default: None
RI = Rotation increment
CP = Copy
0 = Move
1 = Copy
ST = Sort by tools
0 = No
1 = Yes

NEW[ ]
(New, Load Job File), *command*. Creates a new CNC Process Model using the current default unit of measurement or units specified by optionally loading a job file.

Note: New, Load job File is not a recordable macro command.

NEW[FN=$$, JT=##]

FN = Filename (optional)
JT = Job type
0 = Read `.jsf` and convert to `.jof`
1 = Read `.jof`
Note: JT is ignored in `.jsf` mode and defaults to 0 in `.jof` mode.

NEW_GRP[ ]
(New Group), *command*. Removes all elements, from an active group.

NEW_GRP[]

NEW_INCH[ ]
(New Inch), *command*. Creates a new CNC Process Model using inch units of measurement.

NEW_INCH[]

NEW_METRIC[ ]
(New Metric), *command*. Creates a new CNC Process Model using metric units of measurement.

NEW_METRIC[]

NOCORE[ ]
(No Core), *command*. Creates a continuous tool path that removes all the material within a boundary without any scrap pieces. {Awedm}

NOCORE[EL=##, XC=##, YC=##, HR=##,
FA=##, FC=##, WC=##, LW=##]

EL = Outside boundary element (required)
Default: None
XC = X center of start hole (required)
Default: None
YC = Y center of start hole (required)
Default: None
HR = Start hole diameter (required)
Default: None
FA = Finish amount
Default: 0.0
FC = Finish pass count
Default: 1
WC = Width of cut
Default: None
LW = Last width of cut
Default: None

NORMDRAW[ ]
(Normals Draw), *command*. Creates temporary arrows near the center of each surface displaying the surface normal vector direction. {Amill, Ffm}

Note: Normals Draw is not a recordable macro command.

NORMDRAW[GI=##]

GI = Group Index. Indicates whether to draw normals only on surface elements in the active group.
Value 0 = Off. Does not use the active group. All visible (show/masked on) surface elements have their normals drawn.
Value 1 = On. Uses the active group. All surface elements in the active group have their normals drawn.
Default 0

NORMFLIP[ ]
(Normals Flip), *command*. Reverses the surface normal vector direction on the ACIS face for each surface element in the active group. {Amill, Ffm}

Note: Normals Flip is not a recordable macro command.

NORMFLIP[MB=##]

Reverses the surface normal vector direction on the ACIS face for each surface element in the active group. The normal is at the surface center (u=0.5, v=0.5).

MB = Manifold Bodies. Indicates whether to exclude faces that are in manifold solid bodies (or any non-single-face body) in the group of faces to have their normals flipped.
Value 0 = Off. Includes manifold solid body faces in those to be reversed to keep the multi-face bodies pointing the same way.
Value 1 = On. Excludes manifold solid body faces from those to be reversed. Use Surface Edit—Extract and Extract Surfaces From Solids to change individual faces.

- O -

OFFSET[ ]
(Offset Mesh), *command*. Creates a mesh offset from a range of mesh profiles selected using the assigned tool's geometry and the mesh normal orientation. {Amill, Ffm}

OFFSET[E1=$$, E2=$$, OC=##, OF=##,
OA=##, KO=##, GC=##, CM=##,
VT=##]

E1 = Surface start element (required)
Default: None
E2 = Surface end element (required)
Default: None
OC = Offset calculation
0 = Full
1 = Partial
2 = De-offset
Default: 0
OF = Offset direction
0 = Positive
1 = Negative
Default: 0
OA = Offset amount (required)
Default: None
KO = Keep original
0 = Off

1 = On
Default: 0
GC = Gouge correction
0 = Off
1 = On
Default: 0
CM = Connect mesh
0 = Off
1 = On
Default: 0
VT = Virtual mesh tolerance
Default: Global variable
Initial default: 0.001

OFS( )
(Offset), *function*. Returns offset setting of selected element.

OFS (elnum)
-1 = Left
1 = Right
0 = None

ON_LAYER[ ]
(On Layer), *command*. Sets the current insert properties for new elements as layer (reference) elements.

ON_LAYER[LV=##, WP=$$, LY=##, PT=##]

LV = Level (modal)
WP = Work plane (modal)
LY = On layer (modal)
PT = Profile top (modal)
Note: To set profile top off, set PT=?.

OPEN[ ]
(Open Item), *script*. Opens a control panel or dialog box for display in the user interface.

Note: Open Item only functions while running in the macro demo mode without using the default UI scripts.

OPEN[$$]

$$ = Name of control panel or dialog box

OPN_PROF[ ]
(Open Profile), *command*. Creates tool path for a pocket that opens to the side of the part. {Pmill, Amill, Ffm, Aturn}

OPN_PROF[E1=##, E2=##, E3=##, UA=##,
CC=##, CO=##, AP=##, WC=##,
FA=##, ED=##, XC=##, YC=##, DS=##,
DC=##, DE=##, FL=##, AN=##, EP=##,
IN=##, IP=##, AR=##, CI=##, EW=##,
OP=##, RM=##, RT=##, AC=##,
RD=##, IT=##, LY=##, NN=$$, UT=##]

E1 = Part boundary start element (required)
E2 = Part boundary end element (required)
E3 = Material boundary element (required)
UA = Uncut area geometry—use
0 = Off
1 = On
Default: 0
CC = Climb cut
0 = Conventional
1 = Climb
Default: 1
CO = Connect passes
0 = Off
1 = On
Default: 0
AP = Alternate passes
0 = Off
1 = On
Default: 0
WC = Width of cut (required)
FA = Finish amount
Default: 0.0
ED = Extension distance
Default: 0.0
XC = X cut point (required)
YC = Y cut point (required)
DS = First pass level (depth; required)
DC = Depth of cut (required)
DE = Final level (depth; required)
FL = Floor allowance
Default: 0.0
AN = Pass angle
Default: 0.0
EP = Equalized passes (modal)
0 = Off
1 = On
IN = Group island—use
0 = Off
1 = On
Default: 0
IP = Final (island) pass (modal)
0 = Off
1 = On
AR = Ramp angle
Default: 90.0 (vertical)
CI = Cut inside out
0 = Off (cut outside in)
1 = On
Default: 1
EW = Equalize width passes
0 = Off
1 = On
Default: 0

OP = Overlap passes
    0 = Off
    1 = On
    Default: 0
RM = Rough tolerance method
    0 = Off, minimum
    1 = On, Mid Tol
    Default: 0
RT = Rough tolerance amount (distance)
    Default: 0.00005
AC = Corner roll angle
    Default: 60.0
RD = Rapid to depth
    0 = Off
    1 = On
    Default: 0
IT = Island top machining
    0 = Full pass
    1 = Clear top
    Default: 0
LY = Layer
    Default: 1 (not global)
NN = AUM named group name
    Default: AUM
UT = Uncut material tolerance
    Default: 0.01

OPNLIST( )
(Step List Index), *function*. Returns values from the job operations setup step list.

OPNLIST(index)
    index = 0, returns the number of steps in the step list or –1 If no steps exist.
    index = Valid step index, returns the step number of that step.
    index = Greater than the total number of steps or less than 0, returns a -1.

OPTIMIZE[ ]
(Optimize), *command*. Resequences tool path elements in the active group to reduce machine travel and tool changes. {Pmill, Amill, Ffm, Afab, Awedm}

OPTIMIZE[XS=##, YS=##, OM=##, XY=##,
    SP=##, SN=##, ST=##, MT=##]

XS = X Start point (required)
    Default: None
YS = Y Start point (required)
    Default: None
OM = Optimization method
    0 = Closest
    1 = Zigzag
    2 = Improve
    Default: 0
XY = X or Y direction
    0 = X direction
    1 = Y direction
    Default: 0
SP = Spacing
    Default: 0
SN = Sensitivity
    Default: 0
ST = Sort by tools
    0 = Off
    1 = On
    Default: 0
MT = Maximum time (for OM=2)
    Default: Runs till optimum found

OPTYPE( )
(Operation Type), *function*. Returns the operation SNCS number for a step number.

OPTYPE(stpnum)

ORDSTP[ ]
(Order Step), *command*. Sorts or renumbers the Step list.

Note: Obsolete. Maps to Plan_Step_Order
    when TY=1.

ORDSTP[TY=##, ST=##, IN=##]

TY = Type
    0 = By model
    1 = By step number
    2 = Renumber required
ST = Start number (required if TY = 2)
IN = Increment (required if TY = 2)

- P -

PAN[ ]
(Pan), *command*. Shifts the location of the viewing window along the distance between two points without changing its size.

PAN[X1=##, Y1=##, X2=##, Y2=##]

X1 = From X (required)
    Screen coordinate value
    Default: None
Y1 = From Y (required)
    Screen coordinate value
    Default: None
X2 = To X (required)
    Screen coordinate value
    Default: None

Y2 = To Y (required)
    Screen coordinate value
    Default: None

PART_CURVES[ ]
(Parting Line Curves), *command*. Generates polyline curves around the silhouette of the surfaces in the active group, as viewed down the active work plane Z axis from +Z to -Z. {Amill, Ffm}

Note: Parting Line Curves is not a recordable macro command.

PART_CURVES[PT=##]

PT = Approximation tolerance of how far the
    curve can deviate from the surface.
    (Optional)
    Default: 0.001

PAUSE[ ]
(Pause), *command*. Pauses the macro while a message box is displayed.

Note: Pause is not a recordable macro command.

PAUSE[TX=$$, LR=##, LC=##, SR=##,
    SC=##, PT=##]

TX = " Body of text"
LR = Location row of upper left corner of
    dialog box
LC = Location column of upper left corner of
    dialog box
SR = Size of dialog box in rows
SC = Size of dialog box in columns
PT = Pause time options
    0 = Single OK button
    1 = Two buttons
    2 or greater = Delay time (in seconds)
    for display
    Use the PAUSE macro command to
    display a timed message.

PCHG_HP[ ]
(Property Change Holes/Points), *command*. Changes the properties of hole or point elements in the active group.

PCHG_HP[TY=##, TL=##, LV=##, CL=##,
    AN=##, CN=##, PK=##, TD=##]

Note: This replaces PCHG_JHP in
    applications not using Job Operations
    Setup.

TY = Type
    0 = N/C
    1 = Hole
    2 = Point
    Default: 0
TL = Tool
    Default: N/C
LV = Level
    Default: N/C
CL = Clear
    Default: N/C
AN = Tool Angle
    Default: None
CN = Clear on
    0 = Off
    1 = On
    2 = N/C
    Default: 2
PK = Peck
    0 = Off
    1 = On
    2 = N/C
    Default: 2
TD = Tip Depth
    (Tip Depth is computed from the Spot
    Diameter or Full Depth)
    Default: N/C

PCHG_JHP[ ]
(Property Change Jos Hole point), *command*. Changes the properties of hole or point elements in the active group.
Note: This replaces PCHG_HP in applications using Job Operations Setup.

PCHG_JHP[TY=##, ST=##, LV=##, CL=##,
    CN=##, PK=##, TD=##]

TY = Type
    0 = N/C
    1 = Hole
    2 = Point
    Default: 0
ST = Step
    Default: N/C
LV = Level
    Default: N/C
CL = Clear
    Default: N/C
CN = Clear on
    0 = Off
    1 = On
    2 = N/C
    Default: 2
PK = Peck
    0 = Off
    1 = On

2 = N/C
Default: 2
TD = Tip Depth
(Tip Depth is computed from the Spot Diameter or Full Depth)
Default: N/C

PCHG_LY[ ]
(Property Change Layer), *command*. Changes one or more layer properties for an active group of elements, including layer, level and profile top.

PCHG_LY[LY=##]

LY = Layer
Default: N/C

PCHG_SURFACES[ ]
(Property Change Surfaces), *command*. Changes the settings that control the visual display of surfaces in the active group.

PCHG_SURFACES[CG=##, SU=##, SV=##, ST=##, RN=##, MS=##]

CG = Change grid, whether to change the surface grid settings (by using SU, SV, and ST)
0 = No
1 = Yes
Default: 1 (This allows compatibility with macros from previous revisions)
SU = Surface grid lines, count in U direction
Integer greater than 0
Default: 3
SV = Surface grid lines, count in V direction
Integer greater than 0
Default: 3
ST = Surface grid lines, curve tolerance
Floating point number greater than 0
Default: 0.01
RN = Reverse normals (required)
0 = No
1 = Yes
Default = 0
MS = Exclude multi-surface bodies
0=No
1 = Yes
Default: 0

PCHG_TP[ ]
(Property Change Tool Path), *command*. Changes one or more tool path properties for an active group of elements, including step, tool offset, level, clear and profile top.

PCHG_TP[TL=##, LV=##, PN=##, PT=##, OF=##, CN=##, CL=##]

Note: This replaces PCHG_ST in applications not using Job Operations Setup.
TL = Tool
Default: N/C
LV = Level
Default: N/C
PN = Profile top on
0 = Off
1 = On
2 = N/C
Default: 2
PT = Profile top value
Default: N/C
OF = Offset
0 = Left
1 = Right
2 = None
3 = N/C
Default: 3
CN = Clear on
0 = Off
1 = On
2 = N/C
Default: 2
CL = Clear
Default: N/C

PCHG_ST[ ]
(Property Change Step), *command*. Changes one or more tool path properties for an active group of elements, including step, tool offset, level, clear and profile top.

PCHG_ST[ST=##, LV=##, PN=##, PT=##, OF=##, CN=##, CL=##]

Note: This replaces PCHG_TP in applications using Job Operations Setup.
ST = Step
Default: N/C
LV = Level
Default: N/C
PN = Profile top on
0 = Off
1 = On
2 = N/C
Default: 2
PT = Profile top value
Default: N/C
OF = Offset
0 = Left
1 = Right

2 = None
3 = N/C
Default: 3
CN = Clear on
0 = Off
1 = On
2 = N/C
Default: 2
CL = Clear
Default: N/C

PCHG_WP[ ]
(Property Work Plane), *command*. Assigns elements in the active group to a different work plane while maintaining their current world positions.

PCHG_WP[WP=$$]

WP = Work plane (required)
Default: None

PECK( )
(Peck), *function*. Returns peck switch setting of selected element.

PECK(elnum) Returns peck switch setting of selected element.
0 = Off
1 = On

PLANAR[ ]
(Planar Cuts), *command*. Creates planar profiles parallel to the active work plane at the Z levels you specify on a selected range of existing mesh profiles.

PLANAR[E1=$$, E2=$$, ZT=##, SV=##,
BR=##, DV=##, OF=##, OA=##, VT=##]

E1 = Surface start element (required)
Default: None
E2 = Surface end element (required)
Default: None
ZT = Planar Z (required)
Default: None
SV = Spacing value (required)
Default: None
BR = Background refine
0 = Distance
1 = Count
2 = Out Tol
3 = N/C
Default: 3
DV = Background value
Default: 0.1
OF = Offset
0 = Positive
1 = Negative
2 = 0
Default: 2
OA = Finish amount
Default: 0.0
VT = Virtual mesh tolerance
Default: Global variable
Initial default: 0.001

PLANE_DATA[ ]
(Plane Data), *command*. Use Plane Data to provide information about a work plane's orientation and location in space. You can use Plane Data to inquire about work planes in unfamiliar files or to redefine work planes easily.

PLANE_DATA[WP=$$, RO=##]

WP = Work plane (required)
Default: None
RO = Rotation order (required)
0 = ABC
1 = BAC
2 = ACB
3 = BCA
Default: None

PLAN_ALL_REMOVE[ ]
(Planner Remove All), *command*. Removes all items in the specified list that are un-referenced by the process model or other lists.

PLAN_ALL_REMOVE[LIST=##]

LIST = Which list is to be cleaned up. (required)
1 = Step List (remove Steps)
2 = Tool List (remove tool assignments)
Default: None

PLAN_ITEM_MOVE[ ]
(Planner Move), *command*. Moves an item from one position in the list to another.

PLAN_ITEM_MOVE[WHICHLST=##,
WHICH=$$, WHERE=##, DEST=##]

WHICHLST = Which list the move occurs in (required)
1 = Step List (move a Step)
2 = Tool List (move a tool assignment)
Default: None

WHICH = The username of the item to move. (required)
    The Step Number for Steps or the Tool Number for Tools
    Default: None
WHERE = Where to put the item. (required)
    0 = After the item specified in the DEST variable
    1 = At the beginning of the list
    2 = At the end of the list
    Default: None
DEST = The item after which the selected item will move if WHERE = 0 (Must be specified if WHERE = 0)
    The Step Number for Steps or the Tool Number for Tools.
    Default: 0

PLAN_ITEM_REMOVE[ ]
(Planner Remove Item), *command*. Removes the specified item from the JOS data if it can be.

PLAN_ITEM_REMOVE[LIST=##,
    USER=$$]

LIST = Which list the item is to be removed from (required)
    1 = Step List (remove a Step)
    2 = Tool List (remove a tool assignment)
    Default: None
USER = Which item is to be removed (required)
    The Step Number for Steps or the Tool Number for Tools
    Default: None

PLAN_STEP_ORDER[ ]
(Planner Sort By), *command*. Sorts or renumbers the Step list.

PLAN_STEP_ORDER[TY=##, ST=##,
    IN=##]

TY = The type of action to perform (required)
    0 = Reorder Steps by their first appearance in the process model
    1 = Reorder Steps by their step number
    2 = Renumber the Step List
    Default: None.
ST = The number for the first step if steps are being renumbered
    1 to 9999
    Default: 1
IN = The increment for each successive step
    1 to 9999
    Default: 1

PLAN_TOOL_ORDER[ ]
(Planner Tool Order), *command*. Sorts the Tool list.

PLAN_TOOL_ORDER[TY=##]

TY = The type of sort to perform (required)
    0 = Reorder Tools by their first appearance in the step list
    1 = Reorder Tools by their first appearance in the process model
    2 = Reorder Tools by their Tool Number
    Default: None.

PLN( )
(Work Plane Name), *function*. Returns work plane index number of selected element.

PLN(elnum)
    elnum = Element number or name
Results
    0 = XY plane
    1 = YZ plane
    2 = XZ plane

PLNIX( )
(Plane IX Vector), *function*. Returns X components of plane’s I orientation vectors.

PLNIX(index)
    index = Plane index number

PLNIY( )
(Plane IY Vector), *function*. Returns Y components of plane’s I orientation vectors.

PLNIY(index)
    index = Plane index number

PLNIZ( )
(Plane IZ Vector), *function*. Returns Z components of plane’s I orientation vectors.

PLNIZ(index)
    index = Plane index number

PLNJX( )
(Plane JX Vector), *function*. Returns X components of plane’s J orientation vectors.

PLNJX(index)
    index = Plane index number

PLNJY( )
(Plane JY Vector), *function*. Returns Y components of plane's J orientation vectors.

PLNJY(index)
index = Plane index number

PLNJZ( )
(Plane JZ Vector), *function*. Returns Z components of plane's J orientation vectors.

PLNJZ(index)
index = Plane index number

PLNKX( )
(Plane KX Vector), *function*. Returns X components of plane's K orientation vectors.

PLNKX(index)
index = Plane index number

PLNKY( )
(Plane KY Vector), *function*. Returns Y components of plane's K orientation vectors.

PLNKY(index)
index = Plane index number

PLNKZ( )
(Plane KZ Vector), *function*. Returns Z components of plane's K orientation vectors.

PLNKZ(index)
index = Plane index number

PLNTX( )
(Plane TX Vector), *function*. Returns X components of plane's T orientation vectors.

PLNTX(index)
index = Plane index number

PLNTY( )
(Plane TY Vector), *function*. Returns Y components of plane's T orientation vectors.

PLNTY(index)
index = Plane index number

PLNTZ( )
(Plane TZ Vector), *function*. Returns Z components of plane's T orientation vectors.

PLNTZ(index)
index = Plane index number

PLNNUM( )
(Plane Index Number), *function*. Returns the plane index number from the named plane.

PLNNUM(plnname) Returns the plane index number from the named plane.

POCKET[ ]
(Pocket), *command*. Creates tool path for removing material from inside a closed profile. {Pmill, Amill, Ffm, Aturn}

POCKET[TY=##, EL=##, FA=##, WC=##,
AN=##, XS=##, YS=##, DS=##, DC=##,
DE=##, EP=##, IN=##, IP=##, AR=##,
UA=##, CC=##, RS=##, FL=##, CI=##,
EW=##, OP=##, RM=##, RT=##,
AC=##, RD=##, IT=##, LY=##, NN=$$,
UT=##]

TY = Pocketing method
0 = Spiral
1 = Zigzag
2 = Linear
Default: 0
EL = Outside boundary element (required)
FA = Finish amount
Default: 0.0
WC = Width of cut (required)
AN = Pass angle
Default: 0.0
XS = Start X (no default, not required)
YS = Start Y (no default, not required)
DS = First pass level (depth; required)
DC = Depth of cut (required)
DE = Final level (depth; required)
EP = Equalized passes (modal)
0 = Off
1 = On
IN = Group island—use
0 = Off
1 = On
Default: 0
IP = Final (island) pass (modal)
0 = Off
1 = On
AR = Ramp angle
Default: 90.0 (vertical)
UA = Uncut area geometry—use
0 = Off
1 = On
Default: 0
CC = Climb cut
0 = Conventional
1 = Climb
Default: 1

RS = Ramp from start
    0 = Off, use ramp angle
    1 = On, ramp from start point
FL = Floor allowance
    Default: 0.0
CI = Cut inside out
    0 = Off (cut outside in)
    1 = On
    Default: 1
EW = Equalize width passes
    0 = Off
    1 = On
    Default: 0
OP = Overlap passes
    0 = Off
    1 = On
    Default: 0
RM = Rough tolerance method
    0 = Off, minimum
    1 = On, Mid Tol
    Default: 0
RT = Rough tolerance amount (distance)
    Default: 0.00005
AC = Corner roll angle
    Default: 60.0
RD = Rapid to depth
    0 = Off
    1 = On
    Default: 0
IT = Island top machining
    0 = Full pass
    1 = Clear top
    Default: 0
LY = Layer
    Default: 1
NN = AUM named group name
    Default: AUM
UT = Uncut material tolerance
    Default: 0.01

POINT[ ]
(Point or Point/Rapid), *command*. Creates a point element.

```
POINT[XE=##, YE=##, XS=##, YS=##,
    AN=##, DS=##, LV=##, CO=##, SE=##]
```

XE = X end
    Default: Null
YE = Y end
    Default: Null
XS = X anchor
    Default: Null
YS = Y anchor
    Default: Null
AN = Angle
    Default: Null
DS = Distance
    Default: Null
LV = Level (modal)
CO = Clear Only
    0 = Off
    1 = On
    Default: 0
SE = End solution designator
    0 = +
    1 = -
    2 = N/A
    Default: 0

POINTPOLY[ ]
(Polyline From Points), *command*. Creates a polyline element from a sequence of points in the active group with vertices corresponding to each point in the current group in database order.

Note: Polyline From Points is not a recordable macro command.

```
POINTPOLY[]
```

POLY5X[ ]
(5 Axis Polyline), *command*. Creates a\5-axis polyline tool path necessary for a 4-axis machining operation. {Awedm}

```
POLY5X[CE=##, CL={x, y, z, u, v, w, ... x, y,
    z, u, v, w} or #CLVAR]
```

#clvar can be used instead of CL{}. The
    default on macro record is DM = 4
CE = Close ends
    0 = No
    1 = Yes
    Default: 0
CL = Coordinate list (required)
    A coordinated list or variable declared
    as coordinate list.

POLYLINE[ ]
(Polyline), *command*. Creates a single element made up of a sequence of line segments.

```
POLYLINE[CE=##, CL={} or #CLVAR]
    #clvar can be used instead of CL{}.
```

CE = Close ends
    0 = No
    1 = Yes
    Default: 0
CL = Coordinate list (required)
    A coordinate list or a variable declared

as a coordinate list
Default: None
Note: See COORDINATE LIST.

POLYPOINT[ ]
(Polyline To Points), *command*. Creates point elements at all the vertices of each polyline in the active group.

Note: Polyline To Points is not a recordable macro command.

POLYPOINT[]

POLYTHIN[ ]
(Polyline Thinning), *command*. Refines the polylines in the active group using the current Virtual Mesh Tolerance setting.

Note: Polyline Thinning is not a recordable macro command.

POLYTHIN[]

PROF_GRP[ ]
(Profile Group), *command*. Adds or removes all elements of a selected profile to the active group.

PROF_GRP[EL=##, AR=##]

EL = Element in profile (required)
A valid element number
Default: None
AR = Add/remove (modal)
0 = Add
1 = Remove

PROF_SEQ[ ]
(Profile Sequence), *command*. Sets the insert properties to insert new elements before the first element, or after the last element, of an existing profile in the model.

PROF_SEQ[BA=##, EL=##, ME=##]

BA = Before/after
0 = Before
1 = After
Default: 1
EL = Element (required)
Element number or element name
Default: None
ME = Match element (modal)
0 = Off
1 = On
Default: 0

PROF_SLICE[ ]
(Profile Splice), *command*. Creates a profile on the intersection of a plane with an ACIS solid body. {Amill, Ffm}

PROF_SLICE[SE=##, ZL=##, TO=##]

SE = Element index of surface
Default: None
ZL = Elevation of slice plane
Default: 0.0
TO = Tolerance
Default: 0.001

PROF_START[ ]
(Profile Start), *command*. Re-sequences a single valid profile so that the element you select is the first element.

PROF_START[EL=$$]

EL = Element in profile (required)
Default: None

PROF_TRIM[ ]
(Profile Trim), *command*. Trim to intersect, trim or extend, and sequentially order groups of planar elements that are on the same level and work plane.

PROF_TRIM[XS=##, YS=##, TO=##]

XS = X start, for indication point (required)
Default: None
YS = Y start, for indication point (required)
Default: None
TO = Tolerance
Default: 0.1

PROJECT[ ]
(Project, Milling), *command*. Projects a polyline copy of the active group of elements onto a plane, cone, cylinder, or sphere. {Amill, Ffm}

PROJECT[TY=$$, E1=$$, E2=$$, OF=##,
OA=##, DE=##, VT=##]

TY = Type
0 = Plane
1 = Cone
2 = Sphere
3 = Mesh
Default: 0
E1 = Start element (required)
Default: None

E2 = End element (required)
Default: None
OF = Offset side
0 = Positive
1 = Negative
2 = 0
Default: 2
OA = Offset amount
Default: 0
DE = Explode distance
Default: 0.01
VT = Virtual mesh tolerance
Default: Global variable
Initial default: 0.001

PROJECT[ ]
(Project, Wire Edm), *command*. Projects an image of the active group of planar elements onto a plane. {Awedm}

PROJECT[PT=$$, E1=$$, E2=$$, ED=##,
OS=##, OA=##]

PT = Project to
Options: 0 = Plane (No other option)
Default: 0
E1 = First element (required)
Default: None
E2 = Second element (required)
Default: None
ED = Explode distance
Default: 0.01
OS = Offset side
Options: 0 = Pos
1 = Neg
2 = Zero
Default: 2
OA = Offset amount
Default: 0

PROJECTED[ ]
(Projected), *command*. Creates tool path that follows any line or polyline path you create in SmartCAM across multiple surfaces. {Ffm}

PROJECTED[OA=##, TI=##, TO=##,
CF=##, BG=##, LU=##, LN=$$, FU=##,
FN=$$, FTS=$$]

OA = Offset (finish) amount (required)
Default: 0.0
TI = In tolerance (required)
Default: 0.01
TO = Out tolerance (required)
Default: 0.01
CF = Clear/feed
0 = Absolute
1 = Incremental
2 = Straight
Default: 0
BG = Bridge gaps
0 = Straight
1 = Clear/feed
Default: 0
LU = Log file—use
0 = Off
1 = On
Default: 0
LN = Log file name
Default: `surfmach.log`
FU = External file—use
0 = Off
1 = On
Default: 0
FN = External file name
Default: `pmfile. Ext` where `.ext` is `.sat`, `.sab`, or `.igs`, depending on the file type
FTS= External file import type
SAT
SAB
IGES

PROMPT[ ]
(Prompt), *command*. Pauses the macro displaying A message box that requests input for a user variable.

Note: Prompt is not a recordable macro command.

PROMPT[TX=$$, VN=$$, LR=##, LC=##,
SR=##, SC=##, DV=##]

TX = “ Body of text”
VN = “ Variable name”
LR = Location row of upper left corner of dialog box
LC = Location column of upper left corner of dialog box
SR = Size of dialog box in rows
SC = Size of dialog box in columns
DV = Display variable options
0 = Off
1 = On (variable name displayed in text)

PROPERTY[ ]
(Property), *command*. Sets insert properties for the current layer or step and the tip depth of holes.

PROPERTY[LV=##, WP=$$, PT=##,
CL=##, OF=##, TD=##]

LV = Level (modal)
WP = Work plane (modal)
PT = Pro Top (modal)
To set Prof Top off, set PT=?
CL = Clear (modal)
To set Clear off, set CL=?
OF = Offset (modal)
0 = Left
1 = Right
2 = None
TD = Tip depth
Default: Null

PROFDIVIDE[ ]
(Profile Divide), *command*. Converts a profile into a polyline that has a specified number of control points.

Note: Profile Divide is not a recordable macro command.

PROFDIVIDE[E1=#, E2=#, CN=#]

E1 = Element number for start of profile
E2 = Element number for end of profile
CN = Count number of points in resultant polyline
Default: 2

PTCNT( )
(Point Count), *function*. Returns the number of control points of selected spline or polyline element.

PTCNT(elnum)
elnum = Element number

PTOP( )
(Profile Top), *function*. Returns the profile top value of selected element.

PTOP(elnum)
elnum = Element number

- Q -

- R -

RAD( )
(Radius), *function*. Returns the radius value of selected arc element.

RAD(elnum)

RANGE_GRP[ ]
(Range Group), *command*. Adds or removes a selected range of sequential elements from the database to or from the active group.

RANGE_GRP[E1=##, E2=##, AR=##]

E1 = Element start (required)
Default: None
E2 = Element end (required)
Default: None
AR = Add/remove (modal)
0 = Add
1 = Remove

READ[ ]
(Open), *command*. Opens an existing process model file.

READ[FN=$$, FT=##]

FN = Filename (required)
Default: None
FT = File type
0 = 2-D (`.sh2`)
1 = 3-D (`.sh3`)
2 = `.pm4`
Default: 1

READOUT[ ]
(Readout), *command*. Pauses the macro execution, displays a message in the read-out line and waits for a key to be pressed. Used primarily for demos.

Note: Readout is not a recordable macro command.

READOUT[TX=$$]

TX=pause message (optional)
Note: If no text is specified, the phrase "Macro paused. Press a key to continue..." is displayed on the read-out line.

REDRAW[ ]
(Redraw), *command*. Refreshes the graphic view and redisplays all the visible elements in the active viewing window.

REDRAW[]

REFINE[ ]
(Refine Mesh), *command*. Changes the controlling point count, spacing, divisions, and cut direction on a range of mesh profiles with the option to keep the original range of profiles. {Amill, Ffm}

REFINE[E1=$$, E2=$$, SP=##, DI=##,
SV=##, DV=##, KO=##, CD=##,
CM=##]

E1 = Mesh start element (required)
Default: None
E2 = Mesh end element (required)
Default: None
SP = Spacing
0 = Distance
1 = Count
2 = Cusp Hgt
3 = N/C
Default: 3
DI = Division
0 = Distance
1 = Count
2 = Mid Tol
3 = N/C
Default: 3
SV = Spacing value (required)
Default: None
DV = Division value (required)
Default: None
KO = Keep original
0 = Off
1 = On
Default: 0
CD = Change cut direction
0 = Off
1 = On
Default: 0
CM = Connection mesh
0 = Off
1 = On
Default: 0

REM_LINK[ ]
(Remove Link), *command*. Deletes an individual link or a range of links from between two profiles. {Awedm}

REM_LINK[P1=##, P2=##, S1=##, S2=##,
EL=##]

P1 = Primary range start
Default: None
P2 = Primary range end
Default: None
S1 = Secondary range start
Default: None
S2 = Secondary range end
Default: None
EL = Element number or name
Default: None

REMOVE_SYNCH[ ]
(Delete Synch), *command*. Deletes a pair of matching Wait commands, or all Start, End, and Wait synch commands associated with the active group. {Aturn}

REMOVE_SYNCH[EL=##]

EL = Wait command element (deletes pair)

RENDER[ ]
(Render), *command*. Opens the Render Image window and creates a single shaded image of a group of surface elements. The current view angle is used to display the image. {Amill. Ffm}

RENDER[RT=##, RF=##, GN=$$, ST=##,
NT=##, GL=##, LS=##, L1=##, X1=##,
Y1=##, Z1=##, I1=##, L2=##, X2=##,
Y2=##, Z2=##, I2=##, AM=##, MP=##,
WLX=##, WLY=##, WSX=##, WSY=##]

RT = Rendering type
0 = Flat
1 = Smooth
RF = Rendering faceting
0 = Coarse
1 = Medium
2 = Fine
3 = Very Fine
GN = Group name for the surface elements
to be rendered.
ST = Surface deviation tolerance (required)
Defaults in Inches:
Coarse = 0.1
Medium = 0.01
Fine = 0.005
Very Fine = 0.0005
Note: Defaults are converted to
millimeters for metric job file.
NT = Normal tolerance (required)
Between 0 and 180 degrees
Defaults:Coarse = 30
Medium = 10
Fine = 5
Very Fine = 2
GL = Number of Grid Lines (required)
Defaults in Inches:
Coarse = 8

Medium = 8
Fine = 16
Very Fine = 32

LS = Light Sources
0 = Vector light
1 = Point light
2 = Both vector and point light
Default: 0

L1 = Light 1 on/off
0 = off
1 = on
Default: 1

X1 = View space location of tail of vector, X coordinate (required if Vector light is on)
Default: None

Y1 = View space location of tail of vector, Y coordinate (required if Vector light is on)
Default: None

Z1 = View space location of tail of vector, Z coordinate (required if Vector light is on)
Default: None

I1 = Intensity of light 1 vector light
Between 0.0 and 1.0
Default: 1.0

L2 = Light 2 on/off
0 = off
1 = on
Default: 0

X2 = Model space location of point light, X coordinate (required if Point light is on)
Default: None

Y2 = Model space location of point light, Y coordinate (required if Point light is on)
Default: None

Z2 = Model space location of point light, Z coordinate (required if Point light is on)
Default: None

I2 = Intensity of light 2 point light
Between 0.0 and 1.0
Default: 1.0

AM = Ambient light level
Between 0.0 and 1.0
Default: 0.0

MP = Mark point light location
0 = off
1 = on
Default: 0

WLX = Window location, X coordinate, upper left corner of the Render Image window
units = screen pixels from the upper left corner of the screen with +x to the right and +y down.

WLY = Window location, Y coordinate, upper left corner of the Render Image window
units = screen pixels from the upper left corner of the screen with +x to the right and +y down.

WSX = Window size, width of the Render Image window
units = screen pixels. The number of units depends on the size of your screen.

WSY = Window size, height of the Render Image window.
units = screen pixels. The number of units depends on the size of your screen.

For WLX, WLY, WSX, and WSY, the value's number of units depends on the size of your screen. For example, if your screen resolution is 1024 by 768, a value of 512 would equal half the width and a value of 375 would equal about half the height. A sample setting might be 50, 600, 350, 250.

RENDER_LOAD[ ]
(Render Load), *command*. Reloads an image stored using the RENDER_SAVE command. {Amill, Ffm}

Note: Render Load is not a recordable macro command.

RENDER_LOAD[FN=$$]

FN = Image filename (optional)
Note: If a file name is not given, the file name defaults to the same name as the active process model file with an `.rnd` extension, and retrieves the file from the current model directory.

RENDER_SAVE[ ]
(Render Save), *command*. Stores the render image in a file. The image is stored in a native SmartCAM image file format. The default file name extension is `.rnd`. The image can be loaded at a later time using the RENDER_LOAD macro. {Amill, Ffm}

Note: Render Save is not a recordable macro command.

RENDER_SAVE[FN=$$]

FN = Image filename (optional)
Note: If a file name is not given, the file name defaults to the same name as the active process model file with an `.rnd`

extension, and stores the file in the current model directory.

RENDER_CLOSE[ ]
(Render Close), *command*. Closes the Render Image window. {Amill, Ffm}

RENDER_CLOSE[]

REPORT_JOB[ ]
(Report Job), *command*. Prints job information or create a report file that contains job information for the open process model.

REPORT_JOB[FF=$$, OF=$$, OD=##]

FF = Report Format File (required)
The full path to the report formatter directives file to use to generate the report
Default: None
OF = Output File (required)
The full path of the output file
Default: None
OD = Output Device (required)
0 = Printer
1 = File
Default: None

REPORT_STEP[ ]
(Report Step), *command*. Prints information about the steps used in the active job operations setup or to create a report file that contains step information.

REPORT_STEP[SS=##, SB=##, IJ=##, FF=$$, OF=$$, OD=##]

SS = Selection Set, the Steps that will be reported (required)
0 = All Steps
1 = Used in process model (The Steps that are used in the Process Model)
Default: None
SB = Sort By, how to sort the selected steps (required)
0 = By current order (show steps as they are in the current process plan)
1 = By step number (show steps sorted by Step Number)
2 = By process model order (show Steps in the order that they are first referenced in the current process model)
Default: None
IJ = Include Job, whether to include the job information (required)
0 = Do not include job information
1 = Include job information in the report
Default: None
FF = Report Format File (required)
The full path to the report formatter directives file to use to generate the report
Default: None
OF = Output File (required)
The full path of the output file
Default: None
OD = Output Device (required)
0 = Printer
1 = File
Default: None

REPORT_TOOL[ ]
(Report Tool), *command*. Prints information about the tools in the active job operations setup or to create a report file that contains tooling information.

REPORT_TOOL[SS=##, SB=##, IJ=##, FF=$$, OF=$$, OD=##]

SS = Selection Set, the Tools that will be reported (required)
0 = All Tools
1 = Used by Steps (The tools that are referenced by any Step in the process plan)
2 = Used in Process Model (The Tools that are referenced by Steps used in the Process Model)
Default: None
SB = Sort By, how to sort the selected tools (required)
0 = By current order (show tools as they are in the current process plan)
1 = By tool number (show tools sorted by Tool Number or Turret/Tool combination for Turning)
2 = By step list order (show tools in the order as they first are referenced by Steps in the current Step List)
3 = By process model order (show tools in the order that they are first referenced in the current process model)
Default: None
IJ = Include Job, whether to include the job information (required)
0 = Do not include job information
1 = Include job information in the report
Default: None

FF = Report Format File (required)
The full path to the report formatter directives file to use to generate the report
Default: None
OF = Output File (required)
The full path of the output file
Default: None
OD = Output Device (required)
0 = Printer
1 = File
Default: None

RES_PLANE[ ]
(Reserve Plane), *command*. Prevents a work plane from being modified when you use Transform modeling tools or functions such as Kill Plane, Merge Plane, or Name Plane or when you merge a file.

```
RES_PLANE[WP=$$]
```

WP = Work plane (required)
Default: None

REV_ORDER[ ]
(Reverse Order), *command*. Reverses the direction or database order of a selected profile.

```
REV_ORDER[EL=$$, RP=##]
```

EL = Element in profile (required)
Default: None
RP = Reverse profile
0 = Order and direction
1 = Direction only
2 = Order only
Default: 0

ROTATE[ ]
(Rotate), *command*. Rotates or copies an active group of elements about a pivot axis.

```
ROTATE[XS=##, YS=##, ZS=##, XE=##,
    YE=##, ZE=##, AN=##, CP=##, NC=##,
    TS=##]
```

XS = Pivot Axis Point X coordinate (required)
Default: None
YS = Pivot Axis Point Y coordinate (required)
Default: None
ZS = Pivot Axis Point Z coordinate (required for 3D)
Default: None
XE = Axis End Point X coordinate (required for 3D)
Default: None
YE = Axis End Point Y coordinate (required for 3D)
Default: None
ZE = Axis End Point Z coordinate (required for 3D)
Default: None
AN = Angle (required)
Default: None
CP = Copy
0 = No
1 =Yes
Default: 0
NC = Number of copies
Default: 1
TS = Tool sort
0 = Off
1 = On
Default: 1
TY = Type
0 = 2D
1 =3D
SP = Suppress Planes
0 = No
1 =Yes

ROUGH_SURFACES[ ]
(Roughing, Surfaces), *command*. Creates a series of roughing cuts limited in Z depth using selected surface elements as the input for the roughing process. {Ffm}

```
ROUGH_SURFACES[AN=##, SS=##,
    SO=##, OA=##, TI=##, TO=##, PT=##,
    CF=##, BG=##, ZS=##, ZD=##, UE=##,
    RU=##, RD=##, EU=##, ED=##,
    PH=##, SU=##, SD=##, UU=##, LU=##,
    LN=$$, CU=##, CE=$$, CX=##, CY=##,
    FU=##, FN=$$, FTS=$$]
```

AN = Cut angle (required)
Default: 0.0
SS = Start side
0 = Left
1 = Right
Default: 0
SO = Stepover distance (required)
Default: 0.1
OA = Offset (finish) amount (required)
Default: 0.0
TI = In tolerance (required)
Default: 0.01
TO = Out tolerance (required)
Default: 0.01

PT = Path type
  0 = Zigzag
  1 = One way no plunge
  2 = One way plunge
  Default: 0
CF = Clear/feed
  0 = Absolute
  1 = Incremental
  2 = Straight
  Default: 0
BG = Bridge Gaps
  0 = Straight
  1 = Clear/feed
  Default: 0
ZS = Start Z level (required)
  Default: 0.0
ZD = Decrement by Z (required)
  Default: 0.5
UE = Extras—use
  0 = Off
  1 = On
  Default: 0
RU = Roll edge—use
  0 = Off
  1 = On
  Default: 0
RD = Roll edge distance
  Default: 0.125
EU = Extend—use
  0 = None
  1 = Start
  2 = End
  3 = Both
  Default: 0
ED = Extend distance
  Default: 0.25
PH = Cut phase
  0 = As calculated
  1 = Odd
  2 = Even
  Default: 0
SU = Super sample—use
  0 = Off
  1 = On:
  Default: 0
SD = Super sample distance
  Default: 0.1
UU = Uphill motion only
  0 = Off
  1 = On:
  Default: 0
LU = Log file—use
  0 = Off
  1 = On:
  Default: 0
LN = Log file name
  Default: `surfmach.log`
CU = Containment—use
  0 = Off
  1 = On:
  Default: 0
CE = Containment boundary profile element number
  Default: None
CX = Containment cut area and start indication, X
CY = Containment cut area and start indication, Y
FU = External file—use
  0 = Off
  1 = On:
  Default: 0
FN = External file name
  Default: Taken from `ffm.ini` file
FTS= External file import type
  SAT
  SAB
  IGSS

ROUGH_TOOLPATH[ ]
(Roughing, Toolpath), *command*. Creates a series of roughing cuts limited in Z depth using selected tool path elements as the input for the roughing process. {Ffm}

ROUGH_TOOLPATH[ZS=##, ZD=##]

ZS= Start Z level (required)
  Default: 0.0
ZD = Decrement by Z (required)
  Default: 0.5

RULED[ ]
(Ruled Mesh), *command*, Creates mesh profiles between two defining profiles. {Amill, Ffm}

RULED[E1=$$, E2=$$, E3=$$, E4=$$,
  PD=##, SP=##, SV=##, ZT=##, DI=##,
  DV=##, CM=##, GC=##, OF=##,
  OA=##]

E1 = First profile start (required)
  Default: None
E2 = First profile end (required)
  Default: None
E3 = Second profile start (required)
  Default: None
E4 = Second profile end (required)
  Default: None
PD = Path direction
  0 = Cross
  1 = Along

2 = Planar
Default: 0
SP = Spacing
0 = Distance
1 = Count
2 = Cusp Hgt
Default: 0
SV = Spacing value (required)
Default: None
ZT = Planar Z
Default: 0.0
DI = Division
0 = Distance
1 = Count
2 = Mid Tol
3 = In Tol
4 = Out Tol
Default: 0
DV = Division value (required)
Default: None
CM = Connection mesh
0 = Off
1 = On:
Default: 0
GC = Gouge correction
0 = Off
1 = On:
Default: 0
OF = Offset
0 = Positive
1 = Negative
2 = 0
Default: 2
OA = Finish amount
Default: 0.0

- S -

SAVE[ ]
(Save As), *command*. Saves the current process model and job operations setup file to disk with the existing file names. If the files already exist on disk, a message box is displayed prompting to whether to overwrite existing files.

SAVE[WF=##, FN=$$, JF=$$, FT=##,
GS=##, WP=$$]

WF = Which file
0 = Process model file only
1 = Job file only
2 = Both
Default: 0
FN = Filename (required for WF = 0 or 2)
Default: None
JF = Job file name (required for WF = 1 or 2; ignored when
running in `.jsf` mode.
Default: Active job file name when WF = 0
FT = File type
Non-ACIS SmartCAM products
0 = 2-D (.sh2)
1 = 3-D (.sh3)
2 = V6.x CAM Connect (.sh3)
3 = Process Model (.pm4)
Default: 3
FreeForm Machining and Advanced Milling only
0 = 2-D (.sh2)
1 = 3-D (.sh3)
2 = V6.x CAM Connect (.sh3)
3 = Process Model SAB (.pm4)
4 = No Surfaces (.pm4)
5 = FreeForm V4.6 (.pm4)
6 = Process Model SAT (.pm4)
Default: 4
GS = Group save
0 = Entire file
1 = Active group
Default: 0
WP = Work plane (modal; for FT = 0 only)

SCALE[ ]
(Scale), *command*. Increases or reduces the physical size of each element within a selected group of elements proportionally or distorted along the X, Y, or Z axes.

SCALE[XF=##, YF=##, ZF=##, XS=##,
YS=##, ZS=##]

XF = X factor
Default: 1.0
YF = Y factor
Default: 1.0
ZF = Z factor
Default: 1.0
XS = X start (required)
Default: None
YS = Y start (required)
Default: None
ZS = Z start (required)
Default: None

SELECT[ ]
(Select Item), *script*. Selects an item in the user interface.

Note: Select Item only functions while running in the macro demo mode without using the default UI scripts.

SELECT[$$]

$$ = Name of main menu, pull down or
    workbench item

SELECT_ALL[ ]
(Select All), *command*. Add all non-masked elements into the active group.

SELECT_ALL[]

SEQ_MOVE[ ]
(Sequence Move), *command*. Changes the position of the active group into a different sequence location within the database.

SEQ_MOVE[SC=##]

SC = Sort criteria
    0 = Use existing order in group
    1 = Sort by group selection order
    Default: 0

SEQBA( )
(Sequence Before Or After), *function*, Returns the insert mode.

SEQBA() Returns the insert mode.
    0 = Before
    1 = After

SEQEL( )
(Sequence Element), *function*. Returns the current selected insert element number.

SEQEL()

SGN( )
(Sign), *function*. Returns the sign (+/-) of a numeric expression.

SGN(numexp)
    numexp = angle in degrees

SHELL[ ]
(Shell), *command*. Executes an external script, DOS or UNIX command or executable program to augment the SmartCAM macro system software.

Note: Shell is not a recordable macro command.

SHELL[CMD_LN=$$, WA=##,DP=##]

CMD_LN=DOS or UNIX Command

WA = Wait (required)
    0 = Continues operation
    1 = Waits for completion of operation
DP = Display option (required)
    0 = Runs in window
    1 = Runs in background
    Note: The DP variable does not work on
    Sun Sparc workstations.

SHOW_PATH[ ]
(Show Path), *command*. Graphically simulates the sequence and tool path of the model.

SHOW_PATH[E1=$$, E2=$$, ST=##,
    DT=##, SP=##]

E1 = Range start, first element
    Element number
    Default: 1
E2 = Range end, second element
    Element number
    Default: Last element in database list
ST = Show tool
    0 = Animate
    1 = Nibble
    2 = Draw end
    3 = Filled
    Default: 0
DT = Draw tool
    0 = 3-D (`.sh3`)
    1 = Flat
    Default: 0
SP = Speed
    0–9
    Default: 8

SHPFILE( )
(Shape File), *function*. Returns the process model file path plus process model file name.

Note: Shape File is not a recordable macro function.

SHPFILE()

SIN( )
(Sine), *function*. Returns the sine of an angle.

SIN(numexp)
    numexp = angle in degrees

SMFFILE( )
(Smf File), *function*. Returns the machine file path plus .smf file name.

Note: Smf File is not a recordable macro function.

SMFFILE()

SNUM( )

(Step Number), *function*. Returns the step number of the active step or -1 if layer.

SNUM()

SPLINE[ ]

(Spline), *command*. Creates a tangent curve through control points and starting and ending tangent vectors.

SPLINE[IS=##, JS=##, KS=##, IE=##,
JE=##, KE=##, DS=##, DE=##, CE=##,
CL={ } or #CLVAR]

IS = Start vector I
Default: Null
JS = Start vector J
Default: Null
KS = Start vector K
Default: Null
IE = End vector I
Default: Null
JE = End vector J
Default: Null
KE = End vector K
Default: Null
DS = Start vector distance (length)
Default: 0.0
DE = End vector distance (length)
Default: 0.0
CE = Close end
0 = No
1 = Yes
Default: 0
CL = Coordinate list (required)
A coordinate list or a variable declared as a coordinate list
Default: None

SPLIT[ ]

(Split), *command*. Divides an arc, line, ellipse, or polyline at any point along its length. You can extend or trim the elements at the split point to create a break or an overlap.

SPLIT[EL=$$, XS=##, YS=##, LC=##,
DS=##, GW=##, FR=##, OL=##]

EL = Element to split (required)
Default: None
XS = X start for near point
Default: Null
YS = Y start for near point
Default: Null
LC = Split location
0 = Nearest
1 = Division
2 = Distance
Default: 1
DS = Distance
Default: Null
GW = Gap width
Default: 0.0
FR = From
0 = Start
1 = End
Default: 0
LN = Length
Default: 0.5

SPUN[ ]

(Spun Mesh), *command*. Creates mesh profiles by sweeping a profile about an axis. {Amill, Ffm}

SPUN[E1=$$, E2=$$, E3=$$, PD=##,
OF=##, SP=##, AN=##, OA=##, SV=##,
KO=##, CM=##, ZT=##, GC=##, VT=##]

E1 = Generator profile start (required)
Default: None
E2 = Generator profile end (required)
Default: None
E3 = Axis element number (required)
Default: None
PD = Path direction
0 = Generator
1 = Radial
2 = Planar
3 = Both
Default: 0
OF = Offset
0 = Positive
1 = Negative
2 = 0
Default: 2
SP = Spacing
0 = Distance
1 = Count
2 = Cusp Hgt
3 = Junction
Default: 0
AN = Revolve angle (required)
Default: None

OA = Finish amount
    Default: 0.0
SV = Spacing value (required)
    Default: None
KO = Keep original entity type
    0 = Off
    1 = On:
    Default: 0
CM = Connection mesh
    0 = Off
    1 = On:
    Default: 0
ZT = Planar Z
    Default: 0.0
GC = Gouge correction
    0 = Off
    1 = On:
    Default: 0
VT = Virtual mesh tolerance
    Default: Global variable
    Initial default: 0.001

SQR( )
(Square Root), *function*. Returns the square root of a numeric expression.

SQR(numexp)

SRF_COONS[ ]
(Coons Surface), *command*. Creates a sculpted surface by using four connected boundary profiles that define the surface and cross-section profiles that define interior contour. {Ffm}

SRF_COONS[E1=$$, E2=$$, E3=$$,
    E4=$$, BP=##]

E1 = First profile start (required)
    Default: None
E2 = First profile end (required)
    Default: None
E3 = Second profile start (required)
    Default: None
E4 = Second profile end (required)
    Default: None
BP = By patch-tangents
    0 = Off
    1 = On:
    Default: 0

SRF_DRAFT[ ]
(Draft Surface), *command*. Creates a surface by sweeping a 2-D generator profile along a 2-D or 3-D director curve. The generator rotates as it sweeps so that its projection remains perpendicular to the 2-D projection of the director curve into the active work plane. {Ffm}

SRF_DRAFT[E1=$$, E2=$$, E3=$$, E4=$$]

E1 = Generator profile start (required)
    Default: None
E2 = Generator profile end (required)
    Default: None
E3 = Director profile start (required)
    Default: None
E4 = Director profile end (required)
    Default: None

SRF_LOFTED[ ]
(Lofted Surface), *command*. Creates a surface that is defined by being blended through a series of curves in an active group. {Ffm}

SRF_LOFTED[]

SRF_PUNCH[ ]
(Punch Surface), *command*. Removes a piece of an existing surface, based on the intersection of that surface with a volume defined by a punch profile and punch distance. {Ffm}

SRF_PUNCH[OS=##, PP=##, PD=##]

OS = Element number of surfaces to edit
PP = Profile element (required)
    Default: None
PD = Distance number, default
    Default: 1000

SRF_RULED[ ]
(Ruled Surface), *command*. Creates a surface that is defined by straight lines drawn between two profiles. {Ffm}

SRF_RULED[E1=$$, E2=$$, E3=$$,
    E4=$$]

E1 = Start element of first profile (required)
E2 = End element of first profile (required)
E3 = Start element of second profile
    (required)
E4 = End element of second profile
    (required)

SRF_SPLIT[ ]
(Split Surface), *command*. Divides a surface element or group, or to create polyline curves at the intersection of two surfaces or groups. {Ffm}

SRF_SPLIT[ET=##, EGN=##, ES=##,
OT=##, OGN=##, OS=##, CI=##]

ET = Choose element or group for surfaces to edit.
0 = Group
1 = Element
EGN = Group name for surfaces to edit
ES = Element number of surface to edit.
OT = Choose element or group for intersecting surfaces
0 = Group
1 = Element
OGN = Group name for intersecting surfaces
OS = Element number of intersecting surface
CI = Create intersection curves
0 = Off
1 = On:

SRF_SPUN[ ]
(Spun Surface), *command*. Creates a surface by rotating a 2-D generator curve around an axis vector line. {Ffm}

SRF_SPUN[E1=$$, E2=$$, E3=$$, AN=##]

E1 = Generator profile start element (required)
Default: None
E2 = Generator profile end element (required)
Default: None
E3 = Axis element (required)
Default: None
AN = Revolve angle (required)
Default: None

SRF_TRANSLATED[ ]
(Translated Surface), *command*. Creates a surface by sweeping a 2-D generator profile along a 2-D or 3-D director curve. The generator profile maintains a perpendicular orientation to the tangent vector for each segment as it moves along the director curve. {Ffm}

SRF_TRANSLATED[E1=$$, E2=$$, E3=$$,
E4=$$]

E1 = Generator profile start element (required)
Default: None
E2 = Generator profile end element (required)
Default: None
E3 = Director profile start element (required)
Default: None
E4 = Director profile end element (required)
Default: None

START_PROF[ ]
(Start Prof), *command*. Identifies the start point of the first element in a new profile.

START_PROF[XE=##, YE=##, XS=##,
YS=##, AN=##, DS=##]

XE = X end
Default: Null
YE = Y end
Default: Null
XS = X anchor
Default: Null
YS = Y anchor
Default: Null
AN = Angle
Default: Null
DS = Distance
Default: Null

STEP( )
(Step), *function*. Returns the step number of selected element.
-1 = when element is on layer

STEP(elnum)

STEP_GRP[ ]
(Step Group), *command*. Adds or removes the specified step to or from the active group.

STEP_GRP[ST=##, AR=##]

ST = Step (required)
Step number
Default: None
AR = Add/remove (modal)
0 = Add
1 = Remove

STEP_SEQ[ ]
(Step Sequence), *command*. Inserts new elements before the first use, or after the last use, of an existing step.

STEP_SEQ[BA=##, ST=##, ME=##]

Note: This replaces TOOL SEQUENCE in
applications using Job Operations
Setup.

BA = Before/after
  0 = Before
  1 = After
  Default: 1
ME = Match element (modal)
  0 = Off
  1 = On:
ST = Step (required)
  Valid tool number
  Default: None

STEP_SORT[ ]

(Step Sort), *command*. Resequences a group of elements according to the steps associated with them.

STEP_SORT[ST=##, SD=##]

ST = Step (required)
  Step number, or * for all steps
  Default: None
SD = Sort domain
  0 = Entire file
  1 = Active group
  Default: 1

STEP_SORT_ALL[ ]

(Step Sort All), *command*. Resequences all the step elements in the process model according to the steps associated with them.

STEP_SORT_ALL[SD=##]

SD = Sort domain
  0 = Entire file
  1 = Active group
  Default: 1

STRAIGHT[ ]

(Straight), *command*. Creates tool path that follows a straight line across multiple surfaces. {Ffm}

STRAIGHT[AN=##, SS=##, SO=##, OA=##,
  TI=##, TO=##, PT=##, CF=##, BG=##,
  UE=##, RU=##, RD=##, EU=##,
  ED=##, PH=##, SU=##, SD=##, UU=##,
  LU=##, LN=$$, CU=##, CE=$$, CX=##,
  CY=##, FU=##, FN=$$, FTS=$$]

AN = Cut angle (required)
  Default: 0.0
SS = Start side
  0 = Left
  1 = Right
  Default: 0
SO = Stepover distance (required)
  Default: 0.1
OA = Offset (finish) amount (required)
  Default: 0.0
TI = In tolerance (required)
  Default: 0.01
TO = Out tolerance (required)
  Default: 0.01
PT = Path type
  0 = Zigzag
  1 = One way no plunge
  2 = One way plunge
  Default: 0
CF = Clear/feed
  0 = Absolute
  1 = Increment
  2 = Straight
  Default: 0
BG = Bridge gaps
  0 = Straight
  1 = Clear/feed
  Default: 0
UE = Extras—use
  0 = Off
  1 = On:
  Default: 0
RU = Roll edge—use
  0 = Off
  1 = On:
  Default: 0
RD = Roll edge distance
  Default: 0.125
EU = Extend—use
  0 = None
  1 = Start
  2 = End
  3 = Both
  Default: 0
ED = Extend distance
  Default: 0.25
PH = Cut phase
  0 = As calculated
  1 = Odd
  2 = Even
  Default: 0
SU = Super sample—use
  0 = Off
  1 = On:
  Default: 0
SD = Super sample distance
  Default: 0.1
UU = Uphill motion only
  0 = Off
  1 = On:
  Default: 0

LU = Log file—use
0 = Off
1 = On:
Default: 0
LN = Log file name
Default: `surfmach.log`
CU = Containment—use
0 = Off
1 = On:
Default: 0
CE = Containment boundary profile element number
Default: None
CX = Containment cut area and start indication, X
CY = Containment cut area and start indication, Y
FU = External file—use
0 = Off
1 = On:
Default: 0
FN = External file name
Default: Taken from `ffm.ini` file
FTS= External file import type
SAT
SAB
IGES

STREQUAL( )
(String Equal), *function*. Compares 2 string expressions.

Note: String Equal is not a recordable macro function.

STREQUAL(strexpr1, strexpr2)
0 = strings are not identical.
1 = strings are identical.

STRING:
(Declare Variable as String), *command*. Declares the variable that follows as a string type.

Note: Declare Variable as String is not a recordable macro command. A variable must be declared as a string before being assigned a text value.

STRING:variable

STRIP_EXT( )
(Strip Extension), *function*. Returns a truncated string without file extension.

Note: Strip Extension is not a recordable macro function.

STRIP_EXT(strexpr)

STRLEN( )
(String Length), *function*. Returns number of characters in a string.

Note: String Length is not a recordable macro function. Some eastern double-byte characters count as 2.

STRLEN (strexpr)

STRSUB( )
(String Sub), *function*. Returns a substring of a string beginning at start and continuing for a specified number of characters.

Note: String Sub is not a recordable macro function. Double-byte users must be aware that they can accidentally corrupt 2-byte characters with this function.

STRSUB (strexpr, start, len)

STRTMP( )
(String Template), *function*. Returns a string containing the combined contents of a variable along with some literal text.

Note: String Template is not a recordable macro function.

STRTMP ("literal %VAR")
literal = Any combination of alphanumeric characters. If the character is used as part of the STRTMP syntax (% \ “) it must be preceded by the backslash (\ ) character in order to be output.
%VAR = Any string or numeric variable. The %syntax allows the variable value to be displayed in quotation marks, but only inside STRTMP.

STX( )
(Starting Point X), *function*. Returns the X coordinate of the start point of selected element.

STX(elnum)

STY( )
(Starting Point Y), *function*. Returns the Y coordinate of the start point of selected element.

STY(elnum)

STZ( )
(Starting Point Z), *function*. Returns the Z coordinate of the start point of selected element.

STZ(elnum)

SUB_CALL[ ]
(Subroutine Call), *command*. Creates a sub call element that calls a pre-defined subroutine and want to place additional copies of it in the model. {Pmill, Amill, Ffm, Afab, Awedm}

SUB_CALL[NN=##, CP=##, UE=##, XE=##, YE=##, ZE=##, AS=##, RR=##]

NN = Name of subroutine (required)
CP = Number of repeats to create
Default: 0
UE = Use properties of grouped elements
0 = No
1 = Yes
XE = Insert point X coordinates
YE = Insert point Y coordinates
ZE = Insert point Z coordinates
AS = Start angle
Default: 0
RR = Rotate repeats
0 = No
1 = Yes
Default: 0

SUB_DEFINE[ ]
(Define Subroutine), *command*. Assigns the active group of elements to a subroutine and places a related Sub Call element in the database. {Pmill, Amill, Ffm, Afab, Awedm}

SUB_DEFINE[NN=##, CP=##, UE=##,
 XE=##, YE=##, ZE=##, AS=##, RR=##]

NN = Name of subroutine (required)
CP = Number of repeats to create
 Default: 0
UE = Use properties of grouped elements
 0 = No
 1 = Yes
 Default: 0
XE = Handle point X coordinate
YE = Handle point Y coordinate
ZE = Handle point Z coordinate
AS = Start angle
RR = Rotate repeats
 0 = No
 1 = Yes
 Default: 0

SUB_REMOVE[ ]
(Remove Sub), *command*. Deletes a specified subroutine that has no sub call elements in the database. {Pmill, Amill, Ffm, Afab, Awedm}

SUB_REMOVE[NN=##]

NN = Name of subroutine (required)

SURFMODES[ ]
(Surface Modes), *command*. Use Surface Modes to control settings that SmartCAM uses for creating mesh tool path and surfaces. {Amill, Ffm}

SURFMODES[VT=##, BT=##, ZM=##,
 CN=##, NC=##, DS=##, DF=##,
 GU=##, GV=##, GT=##, RA=##,
 RZ=##, RF=##, SF=##, MC=##, MA=##,
 RD=##, IM=##]

VT = Virtual mesh tolerance (required)
 Global Value
 Default: .001
BT = Boundary tangent angle (required; C1 continuity)
 Global Value
 Default: 2
ZM = Z maxima tolerance (required)
 Global Value
 Default: .01
CN = Coincident normals angle (required)
 Global Value
 Default: 5.0
NC = Non-uniform correction
 0 = Off
 1 = On:
 Default: 0
DS = Initial surface draw
 0 = Off
 1 = On:
 Default: 0
DF = Final surface draw
 0 = Off
 1 = On:
 Default: 1
GU = Display grid lines, count in U direction (FFM)
 Global value
 Default: 1
GV = Display grid lines, count in V direction (FFM)

Global value
Default: 1
GT = Display grid tolerance (FFM)
Global value
Default: 0.01
RA = Resolution, Absolute (RESABS) (FFM)
(required if RD is off)
Range 1.0E-8 to 1.0E-2
Default: 1.0E-6
RZ = Resolution, Zero (RESNOR) (FFM)
(required if RD is off)
Range 1.0E-15 to RA*0.1
Default: 1.0E-10
RF = Resolution, Fitting (RESFIT) (FFM)
(required if RD is off)
Range RA*100 to RA*10000
Default: 1.0E-3
SF = Point Set Fitting Tolerance (FFM)
Global value
Default: 0.001
MC = Model Checking (level of checking)
0 = none
1 = Low
2 = Medium
3 = High
Default: 0
MA = Action (FFM)
0 = None
1 = Delete
Default: 0
RD = Resolution, use system defaults
0 = Off - Use RA, RZ and RF values
1 = On - Do not use RA and RF values; use system defaults instead
IM = Incoming Model Resolution (FFM)
0 = Off - Do not use incoming model header values for RESABS and RESFIT tolerances. Use the RA and RF values instead.
1 = On - Use incoming model header values for RESABS and RESFIT tolerances. Do not use the RA and RF values. Incoming models are those that are being imported or merged into the active process model.
Default: 0

SYNCHRONIZE[ ]
(Synchronize), *command*. Creates Start and End user command elements and resequences two turning operations from different turrets for simultaneous turning operations. {Aturn}

SYNCHRONIZE[PR=##]

PR = Primary turret # at start of synched range (required)
0 = Turret #1
1 = Turret #2
Default: None

SYSTEM_UNITS[ ]
(System Units), *command*. Sets the system units for new jobs and controls the display of the New Job Units dialog box when a new file is started.

SYSTEM_UNITS[UNITSYS=##,
ASKME=##]

UNITSYS = Unit of measurement
0 = Inch
1 = Metric
ASKME = Ask for unit for new file.
0 = No
1 = Yes

- T -

TAN( )
(Tangent), *function*. Returns the tangent of an angle.

TAN(numexp)
numexp = angle in degrees

TAPER_CHG[ ]
(Taper Change), *command*. Creates an @TAPER command at the beginning of a tapered profile to be cut containing the desired wire angle value for the profile that follows. {Awedm}

TAPER_CHG[QA=##, EL=##]

QA = Q angle (required)
Default: None
EL = Element (required)
Default: None

TCONTOUR[ ]
(Turn Contour), *command*. Creates centerline tool path with roughing passes that follow the shape of an existing part profile. {Pturn, Aturn}

TCONTOUR[PS=##, PE=##, MM=##,
MS=##, ME=##, ZS=##, XS=##, DP=##,
NP=##, LA=##, TA=##, XF=##, ZF=##,
CD=##, LD=##, BP=##, TP=##, LY=##]

PS = Part profile starting element
PE = Part profile ending element
MM = Material closure method
    0 = Material profile
    1 = Material point
    2 = Part profile
MS = Material profile starting element
ME = Material profile ending element
ZS = Z ordinate of implicit material profile
XS = X ordinate of implicit material profile
DP = Pass depth
NP = Number of passes, when MM=2
LA = Effective leading edge angle
TA = Effective trailing edge angle
XF = X-axis finish amount
ZF = Z-axis finish amount
CD = Clearance distance
LD = Lead distance
BP = Bidirectional passes
    0 = Off
    1 = On
TP = Create theoretical part profile (0) off, (1) on
LY = Theoretical part profile layer 1..99

TDESC( )
(Tool Description), *function*. Returns the tool description (string) for step number.

Note: Tool Description is not a recordable macro function.

TDESC(stpnum)

TEXT[ ]
(Text), *command*. Creates line and arc elements depicting alphanumeric characters in the model.

```
TEXT[FN=$$, TX=$$, XC=##, YC=##,
    XS=##, YS=##, AN=##, RA=##, HT=##,
    TY=##, LV=##]
```

FN = Filename of font file (modal)
TX = Text string (required)
    Default: None
XC = X center
    Default: Null
YC = Y center
    Default: Null
XS = X start
    Default: Null
YS = Y start
    Default: Null
AN = Angle start
    Default: Null
RA = Radius
    Default: Null
HT = Height (required)
    Default: None
TY = Type (text orientation)
    0 = Linear
    1 = CW
    2 = CCW
    Default: 2
LV = Level
    Default: Null

TGROOVE[ ]
(Turn Groove), *command*. Creates a groove element to cut a groove with straight sides and square, chamfer, or radius corners, on the inside diameter, outside diameter, or face of a part. {Pturn, Aturn}

```
TGROOVE [PS=##, PE=##, MM=##,
    MS=##, ME=##, ZS=##, XS=##, DP=##,
    PM=##, RM=##, PD=##, LA=##, TA=##,
    XF=##, ZF=##, CD=##, LD=##, BP=##,
    TP=##, LY=## ]
```

PS = id of part profile starting element
PE = id of part profile ending element
MM = material closure method
    0 = material profile
    1 = material point
    2 = part profile
MS = id of material profile starting element
ME = id of material profile ending element
ZS = Z ordinate of implicit material profile
XS = X ordinate of implicit material profile
DP = step over distance
PM = pecking method
    0 = default
    1 = depth first
    2 = width first
RM = retract method
    0 = full retract
    1 = partial retract
PD = pecking depth
LA = effective leading edge angle
TA = effective trailing edge angle
XF = X-axis finish amount
ZF = Z-axis finish amount
DP = pass width
CD = clearance distance
LD = lead distance
BP = bi-directional passes
    0 = off
    1 = on
TP = create theoretical part profile
    0 = off
    1 = on

LY = theoretical part profile layer 1..99

THREAD[ ]
(Thread), *command*. Creates a thread element for inside- or outside-diameter threads, straight or tapered threads, or scroll threads with a face-grooving tool. {Pturn, Aturn}

THREAD[ND=##, PT=##, YS=##, XS=##,
YE=##, XE=##, CX=##, AN=##, AL=##,
DS=##, CL=##, TL=##, FN=$$]

ND = Nominal diameter
Default: Null
PT = Pitch
Default: Null
YS = X root start
Default: Null
XS = Z root start
Default: Null
YE = X root end
Default: Null
XE = Z root end
Default: Null
CX = Crest X at start
Default: Null
AN = Taper angle
Default: Null
AL = Axial length
Default: Null
DS = First pass depth
Default: Null
CL = Clear
Default: Null
TL = Thread lead in
Default: Null
FN = Thread table filename
Default: Null
Note: The values required to define a Thread are Root Start and End, Crest, Clearance, Pitch, and depth of first pass. Any of these values can be calculated from other inputs as defined in interactive operations.
Note: If a thread table (FN) is not specified, the macro looks for the default thread table if a nominal diameter (ND) variable is in the option list. Both the ND and FN variables need to be removed from the command list to avoid using a thread table and ensure that the hard-coded values in the command are used as is.
Note: If a thread lookup table is specified, it writes to the fields not specified in the Macro; changing values of entered data if need be.

TIME_OFF[ ]
(Time Off), *command*. Outputs the total time since the TIME_ON command.

Note: Time Off is not a recordable macro command.

TIME_OFF[]

TIME_ON[ ]
(Time On), *command*. TIME_ON resets the “clock” and begins measuring the elapsed time from that point and outputs the information to a file.

Note: Time Off is not a recordable macro command.

TIME_ON[FN=$$, TX=$$, AP=##, LV=##]

FN = “ output_filename”
Default: stdout, saved to user run directory
TX = “ Comment” (annotation, 40 characters maximum)
Default: NULL string
AP = Append
0 = New file
1 = Append file
Default: 0
LV =What information to show
0 = Only time_on/time_off
1 = Every macro execution/termination
2 = Every macro command
Default: 0

TL_PLANE[ ]
(Tool Plane), *command*. Assigns a tool plane to an existing work plane. {Pmill, Amill, Ffm, Aturn}

TL_PLANE[WP=$$, TP=$$]

WP = Work plane (required)
Default: None
TP = Tool plane (required)
Default: None

TLCMT( )
(Tool Comment), *function*. Returns the tool comment (string) for step number.

Note: Tool Comment is not a recordable macro function.

TLCMT(stpnum)

TLDIA( )
(Tool Diameter), *function*. Returns the diameter of the tool for step number.

TLDIA(stpnum)

TLEN( )
(Tool Length), *function*. Returns the length of the tool for step number.

TLEN(stpnum)

TLINEAR[ ]
(Turn Linear), *command*. Creates centerline tool path with straight line roughing passes that are parallel to each other and equal distances apart. {Pturn, Aturn}

```
TLINEAR[PS=##, PE=##, MM=##, MS=##,
    ME=##, ZS=##, XS=##, DP=##, AN=##,
    FC=##, LA=##, TA=##, XF=##, ZF=##,
    CD=##, LD=##, BD=##, BP=##, OP=##,
    TP=##, LY=##]
```

PS = id of part profile starting element
PE = id of part profile ending element
MM = material closure method
  0 = material profile
  1 = material point
  2 = part profile
MS = id of material profile starting element
ME = id of material profile ending element
ZS = Z ordinate of implicit material profile
XS = X ordinate of implicit material profile
DP = pass depth
AN = pass angle
FC = facing flag
  0 = off
  1 = on
LA = effective leading edge angle
TA = effective trailing edge angle
XF = X-axis finish amount
ZF = Z-axis finish amount
CD = clearance distance
LD = lead distance
BD = back-off distance
BP = bi-directional passes
  0 = off
  1 = on
OP = overlap passes
  0 = off
  1 = on
TP = create theoretical part profile
  0 = off
  1 = on
LY = theoretical part profile layer 1..99

TLTYPE( )
(Tool Type), *function*. Returns the type number of tool for step number.

TLTYPE(stpnum)

TMPFILE( )
(Tmp File), *function*. Returns the template file path plus .tmp file name.

TMPFILE()

TNLIST( )
(Tool List Index), *function*. Replaced by OPNLIST(index). Returns values from the job plan tool list based on the following settings:

TNLIST(index)
  index = 0 returns the number of tools in the tool list and returns a –1 if no tools exist.
  index = a valid index in the tool list, returns the tool number of that list entry.
  index = greater than the total number of entries in the tool list or less than 0, TLNLIST returns a -1.

TOOL( )
(Tool), *function*. Replaced by JTOOL( ). Returns the tool number of selected element.

TOOL(elnum)
  -1 = when element is on layer

TOOL_GRP[ ]
(Tool Group), *command*. Adds or removes elements associated to the specified tool to or from the active group.

TOOL_GRP[TL=##, AR=##]

TL = Tool (required)
  Default: None
AR = Add/remove (modal)
  0 = Add
  1 = Remove

TOOL_SEQ[ ]
(Tool Sequence), *command*. Inserts new elements before the first use, or after the last use of the specified existing tool.

TOOL_SEQ[BA=##, TL=##, ME=##]

Note: This replaces STEP SEQUENCE and
    JOB TOOL
in applications not using Job Operations
    Setup.
BA = Before/after
    0 = Before
    1 = After
    Default: 1
TL = Tool (required)
    Valid tool number
    Default: None
ME = Match element (modal)
    0 = Off
    1 = On:

TOOL_SORT[ ]
(Tool Sort), *command*. Resequences a group of elements in the process model according to the tools associated with them.

TOOL_SORT[TL=##, SD=##]

TL = Tool (required)
    Tool number, or * for all tools
    Default: None
SD = Sort domain
    0 = Entire file
    1 = Active group
    Default: 1

TOOL_SORT_ALL[ ]
(Tool Sort All), *command*. Resequences all the step elements in the process model according to the tools associated with them.

TOOL_SORT_ALL[SD=##]

SD = Sort domain
    0 = Entire file
    1 = Active group
    Default: 1

TOTEL( )
(Total Elements), *function*. Returns the number of elements in the database.

TOTEL()

TPLN( )
(Tool Plane Name), *function*. Returns the tool plane name assignment (string) of selected element.

Note: Tool Plane Name is not a recordable macro function.

TPLN(elnum)

TPROFILE[ ]
(Turning Profile), *command*. Creates centerline tool path using single or multiple passes that machine an open or closed profile with an automatic offset from the selected part profile. {Pturn, Aturn}

TPROFILE [PS=##, PE=##, CC=##, LA=##,
    TA=##, XF=##, ZF=##, CD=##, LD=##,
    BD=##, BP=##, TP=##, LY=##]

PS = id of part profile starting element
PE = id of part profile ending element
CC = compensation codes
    0 = off
    1 + on
LA = effective leading edge angle
TA = effective trailing edge angle
XF = X-axis finish amount
ZF = Z-axis finish amount
CD = clearance distance
LD = lead distance
BD = back-off distance
BP = bi-directional passes
    0 = off
    1 = on
TP = create theoretical part profile
    0 = off
    1 = on
LY = theoretical part profile layer 1..99

TRANSLATED[ ]
(Translated Mesh), *command*. Creates mesh profiles by sweeping a generator profile along the path of a director profile. {Amill, Ffm}

TRANSLATED[E1=$$, E2=$$, E3=$$,
    E4=$$, PD=##, SP=##, OF=##, SV=##,
    ZT=##, OA=##, KO=##, CM=##,
    GC=##, VT=##]

E1 = Generator profile start (required)
    Default: None
E2 = Generator profile end (required)
    Default: None

E3 = Director profile start (required)
    Default: None
E4 = Director profile end (required)
    Default: None
PD = Path direction
    0 = Generator
    1 = Director
    2 = Planar
    3 = Both
    Default: 0
SP = Spacing
    0 = Distance
    1 = Count
    2 = Cusp Hgt
    3 = Junction
    Default: 0
OF = Offset
    0 = Positive
    1 = Negative
    2 = 0
    Default: 2
SV = Spacing value (required)
    Default: None
ZT = Planar Z
    Default: 0.0
OA = Finish amount
    Default: 0.0
KO = Keep entity type
    0 = Off
    1 = On:
    Default: 0
CM = Connection mesh
    0 = Off
    1 = On:
    Default: 0
GC = Gouge correction
    0 = Off
    1 = On:
    Default: 0
VT = Virtual mesh tolerance
    Default: Global variable
    Initial default: 0.001

TRIM_EXT[ ]
(Trim Extend), *command*. Trims and/or extends disconnected elements on the same work plane to their intersection point.

TRIM_EXT[E1=$$, E2=$$, S1=##, S2=##,
    IN=##]

E1 = First element (required)
    Default: None
E2 = Second element (required)
    Default: None
S1 = First keep side
    0 = Left
    1 = Right
    2 = Mouse
    3 = N/A
    Default: 3
S2 = Second keep side (required)
    0 = Left
    1 = Right
    Default: None
IN = Intersection number
    1 = First intersection
    2 = Second intersection
    3 = Third intersection
    4 = Fourth intersection
    Default: 1

TRIM_SURFS[ ]
(Trim Mesh), *command*. Trims two ranges of either tool path or layer geometry at their intersection and/or creates a profile along the points of intersection. {Amill, Ffm}

TRIM_SURFS[E1=$$, E2=$$, S1=##,
    T1=##, E3=$$, E4=$$, S2=##, T2=##,
    CC=##, VT=##]

E1 = First surface start (required)
    Default: None
E2 = First surface end (required)
    Default: None
S1 = Keep side surface 1
    0 = Pos
    1 = Neg
    Default: 0
T1 = Trim surface 1
    0 = Off
    1 = On:
    Default: 1
E3 = Second surface start (required)
    Default: None
E4 = Second surface end (required)
    Default: None
S2 = Keep side surface 2
    0 = Pos
    1 = Neg
    Default: 0
T2 = Trim surface 2
    0 = Off
    1 = On:
    Default: 1
CC = Create intersecting curve
    0 = Off
    1 = On:
    Default: 0
VT = Virtual mesh tolerance
    Default: Global variable
    Initial Default: 0.001

TRIM_SURFS[ ]
(Ruled Intersection), *command*. Determines the intersection between two transition pieces and creates a profile and polylines at the point of intersection on both faces. {Afab}

TRIM_SURFS[E1=$$, E2=$$, E3=$$,
E4=$$, S1=##, T1=##, S2=##, T2=##,
CC=##]

E1 = First surface start (required)
Default: Surface range
E2 = First surface end (required)
Default: Surface range
E3 = Second surface start (required)
Default: None
E4 = Second surface end (required)
Default: None
S1 = Keep side of 1st surface
0 = Pos
(always positive)
T1 = Trim first surface
0 = Off
1 = On:
Default: 1
(means build faces for first surface)
S2 = Keep side of 2nd surface
0 = Pos
(always positive)
T2 = Trim second surface
0 = Off
1 = On:
Default: 1
(means build faces for second surface)
CC = Create intersection curve
0 = Off
1 = On:
Default: 0

TYP( )
(Type), *function*. Returns the element type number of selected element.

TYP(elnum)
1= Point
2= Line
3 = Arc
4 = USR
5 = Thread
7 = Ellipse
9 = Helix
10 = Polyline
12 = Cubic spline
13 = Groove
14 = Sub call
18 = Poly 5x

- U -

UEVENT[ ]
(User Event), *command*. Creates a user command element to output a literal CNC machine code, assign value to a template variable or call a template section at a specific sequence point in the model..

UEVENT[XS=##, YS=##, LV=##, TX=$$]

XS, YS = Start point (required)
Default: None
LV = Level
Default: None
TX = Text string (required)
Default: None

UNDO[ ]
(Undo), *command*. Applicable only to commands that support Undo.

UNDO[ ]

UNFOLD[ ]
(Unfold), *command*. Simultaneously unfolds multiple flange geometry on a 3-D model automatically evaluating the thickness of the material and other factors to ensure accurate part size. {Afab}

UNFOLD[BE=##, BS=##, ML=##, TH=##,
IR=#, KF=##, HC=##, MT=##, AD=##]

BE = Base element (required)
Number of element in base profile
BS = Bend side
0 = Bottom
1 = Middle
2 = Top
Default: 1
ML = Mold lines
0 = No
1 = Yes
Default: 0
TH = Thickness
Default: 0
IR = Inside radius
Default: 0
KF = K factor
Default: 0
HC = Hem compensation
Default: 0
MT = Multiply
Default: 1

AD = Add
Default: 0

UNRES_PLANE[ ]
(Unreserve Plane), *command*. Removes the reserved status form a non-system reserved work plane.

UNRES_PLANE[WP=$$]

WP = Work plane
? = Wild character
*= Wildcard (unreserve all)

UNWRAP[ ]
(Unwrap), *command*. Restores wrapped geometry to its planar state. {Amill, Ffm, Aturn}

UNWRAP[WP=$$]

WP = Unwrap plane name (required)
Default: None

UPDATE[ ]
(Update Fields), *script*. Updates the data displayed in the fields for the macro executed.

Note: Update Fields only functions while running in the macro demo mode without using the default UI scripts.

UPDATE[]

- V -

VAR_REMOVE[ ]
(Variable Remove), *command*. Removes the specified user-defined macro variable from memory.

Note: Variable Remove is not a recordable macro command.

VAR_REMOVE[VN=$$]

VN = Variable Name

VAR_REMOVE_ALL[ ]
(Variables Remove All), *command*. Removes all user-defined macro variables from memory.

Note: Variable Remove All is not a recordable macro command.

VAR_REMOVE_ALL[]

VIEW_ANGLE[ ]
(View Angle), *command*. Changes the view orientation of your model for the most advantageous view.

VIEW_ANGLE[VA=##, VP=##, A1=##,
A2=##, RS=##, WP=##]

VA = Vertical axis
0 = X
1 = Y
2 = Z
Default: 0
VP = View plane
0 = World
1 = Work plane
Default: 0
A1 = Angle 1 (required)
0-2 pi (measured in radians)
Default: None
A2 = Angle 2 (required)
0-2 pi (measured in radians)
Default: None
RS = Redraw size
0 = Envelope
1 = Full
2 = Current
Default: 0
WP = Work plane
0 = XY plane
1 = YZ plane
2 = XZ plane
3 = ZX plane
Default: 0

VIEWSURF[ ]
(View Surf), *command*. Displays a simulation of the surface to be created by the wire with a correctly offset tool path by building a temporary ruled surface on the graphic display between the primary and secondary profiles using the links in the active group. {Awedm}

VIEWSURF[P1=##, P2=##, S1=##, S2=##]

P1 = Primary range start
Default: None
P2 = Primary range end
Default: None
S1 = Secondary range start
Default: None
S2 = Secondary range end
Default: None

VIS( )

(Visible), *function*. Returns the visibility status of selected element.

VIS(elnum)
0 = Hidden
1 = Shown

- W -

WAIT[ ]

(Wait), *command*. Create 2 user command elements to control the timing of the two synchronized turrets. {Aturn}

WAIT[B1=#, E1=##, B2=#, E2=##]

B1 = Before/after primary element (required)
0 = Before
1 = After
Default: Keep
E1 = Primary turret wait element (required)
Default: None
B2 = Before/after secondary element (required)
0 = Before
1 = After
Default: Keep
E2 = Secondary turret wait element (required)
Default: None

WALL[ ]

(Wall Offset), *command*. Creates parallel geometry that is offset from an existing element or a profile.

WALL[EL=$$, SD=##, DS=##, ME=##,
RP=##, LR=##, TA=##, TO=##, AN=##]

EL = Element (required)
Default: None
SD = Side (required)
0 = Left
1 = Right
Default: None
DS = Distance (required)
Default: None
ME = Match element’s properties
0 = No
1 = Yes
Default: 0
RP = Repeat
1–999
Default: 1
LR = Loop removal
0 = No
1 = Yes
Default: 0
TA = Tolerance application
0 = Off
1 = On:
Default: 0
TO = Tolerance amount
Default: 0.000005 [0.001]
AN = Corner roll angle
5–180
Default: 180

WHILE( )

(Logical While Loop), *command*. Performs a conditional test and while proven true will loop through a series of commands within a macro. Part of a command set that includes WHILE( ), ENDW.

Note: Logical While Loop is not a recordable macro command. Each WHILE command must be followed by a corresponding ENDW command within the macro.

WHILE(expression)
true
ENDW

expression = A logic expression that may contain the following operators

=	(equal)
<>	(not equal)
>	(greater than)
>=	(greater than or equal to)
<	(less than)
<=	(less than or equal to)
,AND	(logical and)
,OR	(logical or)

true = Macro commands to be performed if the logic expression proves true.

WINDOW[ ]

(Window), *command*. Magnifies part of the graphic view according to the size of the viewing window specified by two opposing corners.

WINDOW[X1=##, Y1=##, X2=##, Y2=##]

X1 = First box corner X (required)
Screen coordinate value
Default: None
Y1 = First box corner Y (required)
Screen coordinate value
Default: None

X2 = Second box corner X (required)
    Screen coordinate value
    Default: None
Y2 = Second box corner Y (required)
    Screen coordinate value
    Default: None

WITH_STEP[ ]
(With Step), *command*. Assigns tool and operation data to each new element, which then becomes part of the tool path.

WITH_STEP[LV=##, WP=$$, ST=##,
    OF=##, CL=##, PT=##]

Note: This replaces WITH TOOL in
    applications using
Job Operations Setup.
LV = Level (modal)
WP = Work plane (modal)
ST = Step (modal)
OF = Offset (modal)
    0 = Left
    1 = Right
    2 = None
CL = Clear (modal)
PT = Profile top (modal)
    Note: To set Clear off, set CL=?. To set Prof Top off, set PT =?.

WITH_TOOL[ ]
(With Tool), *command*. Assigns tool and operation data to each new element, which then becomes part of the tool path.

WITH_TOOL[LV=##, WP=$$, TL=##,
    OF=##, CL=##, PT=##]

Note: This replaces WITH STEP in
    applications not using Job Operations Setup.
LV = Level (modal)
WP = Work plane (modal)
TL = Tool (modal)
OF = Offset (modal)
    0 = Left
    1 = Right
    2 = None
CL = Clear (modal)
PT = Prof_Top (modal)
    Note: To set Clear off, set CL=?. To set Prof Top off, set PT =?.

WKPLN( )
(Work Plane Name), *function*. Returns the work plane name assignment (string) of selected element.

WKPLN(elnum)

WP_CHANGE_NAME[ ]
(Work Plane Change Name), *command*. Use Change Name to change the name of an existing work plane.

WP_CHANGE_NAME[WP=$$, NN=$$]

WP = Work plane (required)
    Default: None
NN = New name (required)
    Default: None

WRAP[ ]
(Wrap), *command*. Changes the active group of elements so that they wrap around the X and/or Y axis and assigns a new work plane for rotary contouring. {Amill, Ffm, Aturn}

WRAP[XS=##, YS=##, LV=##, XE=##,
    YE=##, RA=##]

XS = Seam axis start X (required)
    Default: None
YS = Seam axis start Y (required)
    Default: None
LV = Level
    Default: Current level
XE = Seam axis end X (required)
    Default: None
YE = Seam axis end Y (required)
    Default: None
RA = Radius (required)
    Default: None

- X -

- Y -

- Z -

ZOOM[ ]
(Zoom), *command*. Changes the relative size of geometry displayed in the graphic view by a magnification factor oriented at the zoom center specified.

ZOOM[X1=##, Y1=##, MF=##]

X1 = X center (required)
Screen coordinate value
Default: None
Y1 = Y center (required)
Screen coordinate value
Default: None
MF = Magnification factor
0–999
Default: 1.0

Z_SORT[ ]
(Sort By Z Level), *command*. Sorts the elements in the given range by increasing the Z elevation of their start points in the insert work plane coordinate system.

Z_SORT[E1=##, E2=##]

E1 = Start Element
E2 = End element

Preface

- A -

- B -

- C -

- D -

- E -

- F -