AM
Aperture Macro

The AM parameter is used to define named apertures (sometimes called *special apertures*) in *aperture macro* format consisting of building blocks called *primitives*. The named aperture macros may be used in AD parameter descriptions just like the standard apertures (that is, circle, rectangle, obround, polygon, and thermal). Every non-standard aperture must be described before the D-code associated with it occurs in the file.

Special apertures offer two advantages over standard apertures:

- They allow multiple shapes called primitives to be combined in a single aperture, which permits creation of unusual or complicated apertures.
- They need not be centered.
- Aperture macro modifiers may be variable. Variable modifiers are supplied by the AD parameter that references the aperture macro.
- An aperture macro variable may be a numerical function of another macro variable (+, -, etc.).

Aperture macro contents

An aperture macro contains the following elements:

- aperture macro name
- one or more of the seven aperture primitives, each identified by a primitive number (see Table 3 below for a description of the primitives)
- primitive modifiers that describe the primitive in terms of exposure, position, dimensions, etc.
- variable primitive modifiers to be supplied by the AD parameter
- optional embedded comment blocks
- numerical operators

AM parameter syntax rules

- Like other mass parameters, begin and end each parameter block with a parameter delimiter (typically %).
- Within the AM parameter block, separate each primitive and modifier group by an end-of-block character (typically *).
- Within each primitive group, separate modifiers by commas.
- Modifiers may be absolute values, such as 0, 1, 2, or 9.05, or they may be variable modifiers to be supplied by the AD parameter when it refers to the aperture macro.
- Identify variable modifiers to be supplied by the AD parameter as *$n* where *n* indicates the order in which the modifier is expected in the AD parameter. $1 would be the first variable modifier expected in the AD parameter, $2 the

second, and so on, numbering sequentially from left to right. If an absolute value is entered instead of a variable, the variables shift right. For example, if an absolute value is entered for the first variable, the next variable becomes $1 even though it is the second modifier of the primitive.

- The interpretation of each modifier differs for each primitive. See Table 3 on the next page for a full explanation of aperture macro primitives and modifiers.
- Do not begin a variable primitive modifier with a minus sign (for example, -$1). To indicate negative, precede the variable with 0 (for example, **0-$1**).
- Start optional comment strings with a leading 0 (for example, ***0 THIS IS A COMMENT*).**
- Position and dimensions are expressed in the units specified by the MO parameter. Decimal points are permitted.
- Use only the following numerical operators with variable modifiers:

Operator	Function
+	add
-	subtract
/	divide
x	multiply
=	equate
n	numerical factor

- Make sure the aperture macro file name matches the aperture macro name and that it has a .mac extension.

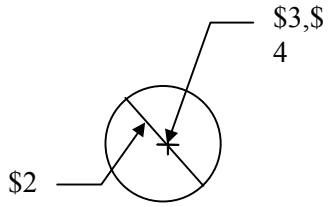
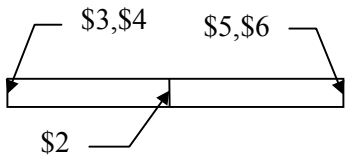
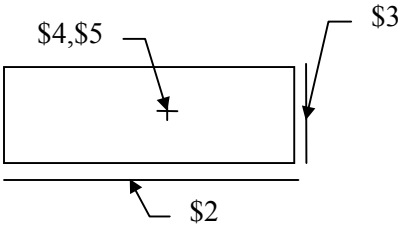
Data Block Format

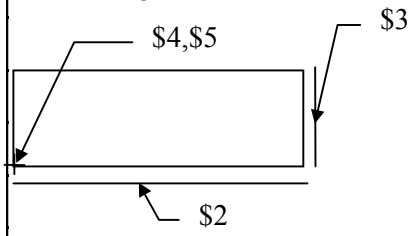
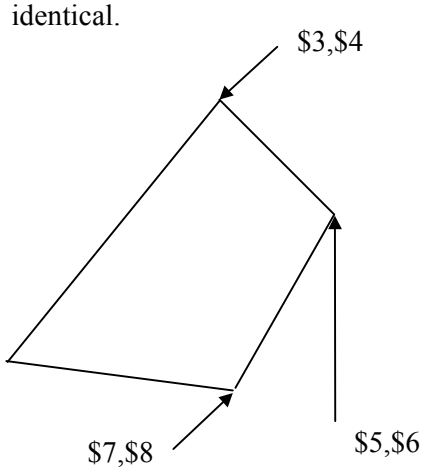
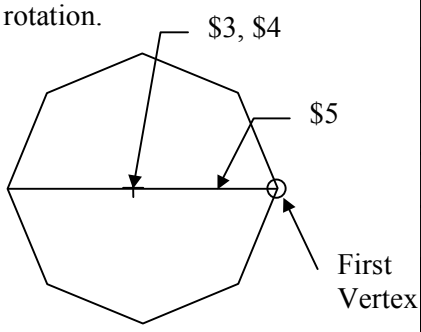
%AM<aperture macro name>*<primitive number>,<modifier$1>,<modifier$2>,[<...>]*[<primitive number>[<modifiers>]]*...*%

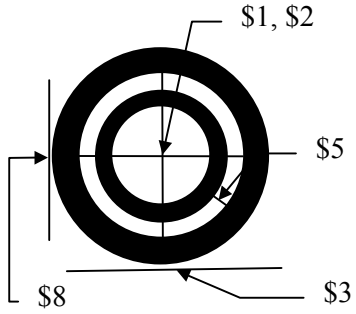
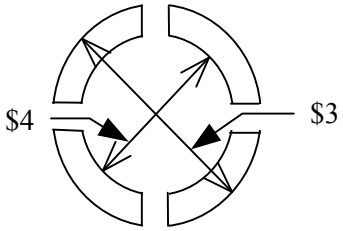
where:

AM	the AM parameter
<aperture macro name>*	the name to be used in the AD parameter
<primitive number>,<modifier$1>,<modifier$2>,<modifier$3>,...*	the primitive number with modifiers. The primitive number identifies the geometry (outline, polygon, etc.). The modifiers differ with the various primitive numbers. Use either actual values (for example, 0.050 for a width) or a variable placeholder (for example, $1 for exposure on/off).

Table 3 Aperture macro primitives

Primitive number	Description	Variable Modifiers	Description
1	**Circle** 	$1	Exposure on/off 0 = OFF (laser off, no exposure) 1 = ON (laser on, image exposed) 2 = reverse current exposure state
		$2	Diameter
		$3	X center position
		$4	Y center position
2 or **20**	**Line (vector)**: a line defined by width, and beginning and end points. The line ends are rectangular. 	$1	Exposure on/off 0 = OFF (laser off, no exposure) 1 = ON (laser on, image exposed) 2 = reverse current exposure state
		$2	Line width
		$3	X start point
		$4	Y start point
		$5	X end point
		$6	Y end point
		$7	Rotation in degrees (+ = counterclockwise, - = clockwise)
21	**Line (center)**: a centered rectangle defined by width, height, and center point. The end points are rectangular. 	$1	Exposure on/off 0 = OFF (laser off, no exposure) 1 = ON (laser on, image exposed) 2 = reverse current exposure state
		$2	Rectangle width
		$3	Rectangle height
		$4	X center point
		$5	Y center point
		$6	Rotation in degrees (+ = counterclockwise, - = clockwise)

22	**Line (lower left)**: a rectangle defined by width, height, and the lower left point. The end points are rectangular.  	$1	Exposure on/off 0 = OFF (laser off, no exposure) 1 = ON (laser on, image exposed) 2 = reverse current exposure state
		$2	Width
		$3	Height
		$4	X lower left point
		$5	Y lower left point
		$6	Rotation in degrees (+ = counterclockwise, - = clockwise)
3	End of file	none	Must be used to end .des files.
4	**Outline**: an open or closed shape defined by a start point, *n* additional points (up to 50), and the X,Y coordinates that define them. For a closed shape, the first and last points must be identical.  	$1	Exposure on/off 0 = OFF (laser off, no exposure) 1 = ON (laser on, image exposed) 2 = reverse current exposure state
		$2	*n*, the number of points in the outline
		$3	X start point
		$4	Y start point
		$5	X point #1
		$6	Y point #1
		$7	X point #2
		$8, etc.	Y point #2. Continue as needed.
		$9 or the last number used	Rotation in degrees (+ = counterclockwise, - = clockwise)
5	**Polygon**: a closed, symmetrical, centered shape defined by n vertices (3 to 10 inclusive), a center point, diameter, and rotation.  	$1	Exposure on/off 0 = OFF (laser off, no exposure) 1 = ON (laser on, image exposed) 2 = reverse current exposure state
		$2	number of vertices (integer)
		$3	X center point
		$4	Y center point
		$5	Diameter
		$6	Rotation in degrees (+ = counterclockwise, - = clockwise)

6	**Moiré**: a cross hair centered on *n* concentric circles defined by the center point, outside diameter, line thickness, and gap between circles.  	$1	X center point
		$2	Y center point
		$3	Outside diameter
		$4	Circle line thickness
		$5	Gap between circles
		$6	number of circles
		$7	Cross hair thickness
		$8	Cross hair length
		$9	Rotation in degrees (+ = counterclockwise, - = clockwise)
7	**Thermal**: a cross hair centered on a circle defined by outside and inside diameter.  	$1	X center point
		$2	Y center point
		$3	Outside diameter
		$4	Inside diameter
		$5	Cross hair thickness
		$6	Rotation in degrees (+ = counterclockwise, - = clockwise)

Example 1

%AMDONUT*1,1,$1,$2,$3*1,0,$4,$2,$3*% Define an aperture macro named DONUT consisting of two concentric circles:

1,1,$1,$2,$3 Circle (1), exposure on (1), diameter ($1), X center ($2), Y center ($3) all to be supplied by AD parameter

1,0,$4,$2,$3 Circle (1), exposure off (0), diameter ($4, different from first circle), X center and Y center ($2 and $3, same as first circle)

The AD parameter using this macro might look like the following:

%ADD32DONUT,0.100X0X0X0.080*% Define D-code 32 to be aperture macro **DONUT**. The diameter of the first circle will be 0.100. The center of both circles will be at 0,0. The diameter of the second circle will be 0.080.

$1 = 0.100
$2 = 0
$3 = 0
$4 = 0.080

Example 2

%AMDONUT*1,1,$1,$2,$3*$1=$2+0.030*1,0,$1-$4,$2,$3*%

Define an aperture macro named **DONUT** consisting of two concentric circles with diameter of the second circle defined as a function of the diameter and center point of the first:

1,1,$1,$2,$3 Circle (1), exposure on (1), diameter ($1), and center point X,Y ($2, $3) to be defined in the AD parameter

$1=$2+0.030 Define a variable to be used to calculate the diameter of second circle to be a function of the diameter and center point X coordinate of the first.

1,0,$1-$4,$2,$3 Circle (1), exposure off (0), diameter ($1-$4), and center point X,Y ($2, $3, same as the first circle).

The ADD parameter using this aperture macro might look like:

%ADD33DONUT,0.020X0X0X0.014*% Define D-code 33 to be aperture macro **DONUT**. The diameter of the first circle 0.020. The center of both circles will be 0,0. The diameter of the second circle will be ((0 + 0.030) - 0.014).

Example 3

%AMDONUT*1,1,0.100X0X0*1,0,0.080X0X0*%

Define an aperture macro named **DONUT** consisting of two concentric circles, using primitive modifiers.

%ADD32DONUT*% The resulting AD command only needs to reference the aperture macro name.