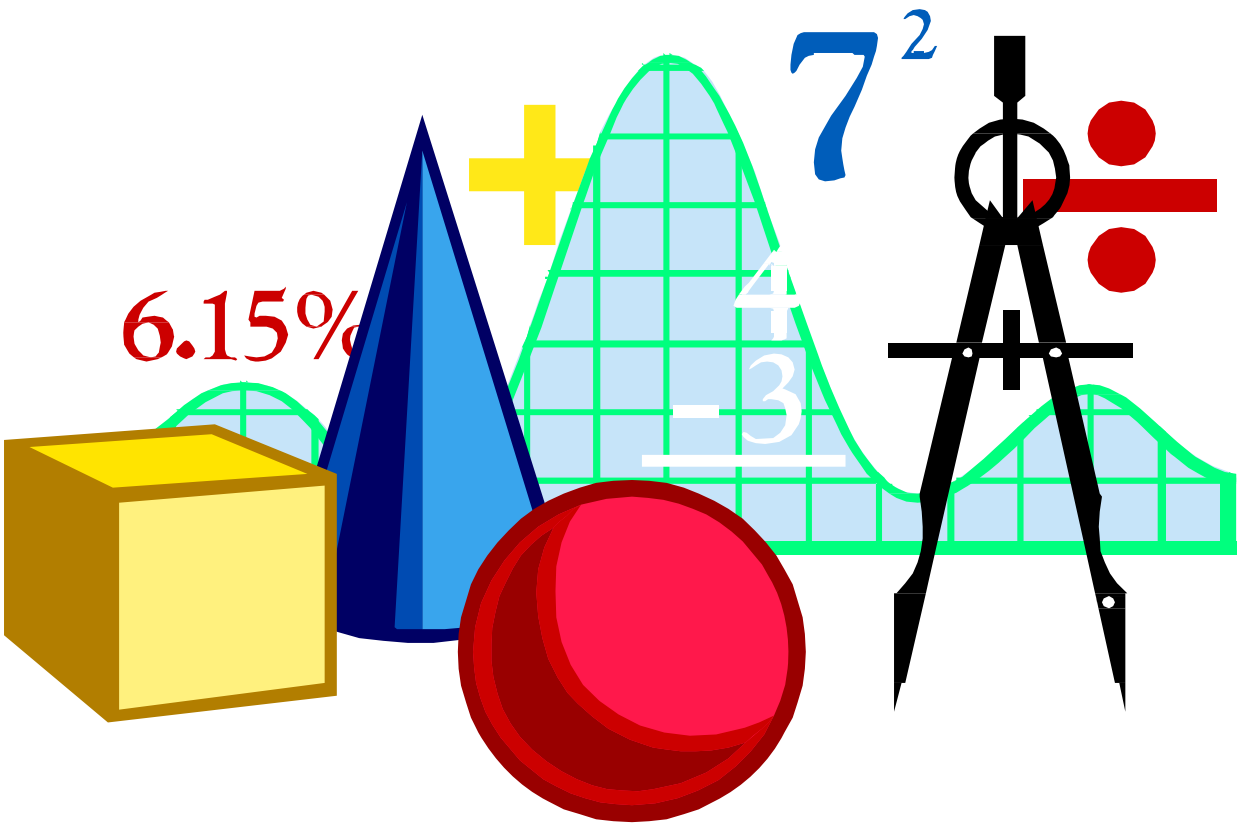
Ucamco
Former Barco ETS

The RS-274X Format

also known as

The Extended Gerber Format



December 2010
DMS Reference US-GXXX-TB-01-EN-D

Correspondence regarding this publication can be sent to:

Ucamco NV
Bijenstraat 19,
B-9051 Gent,
Belgium

For more information:

Our web site: http://www.ucamco.com

E-mail: info@ucamco.com

About Ucamco

Ucamco (former Barco ETS) is a market leader in PCB CAM software and laser photoplotting systems. We have more than 25 years continuous experience developing and supporting leading-edge front-end tooling solutions for the global PCB industry. We help fabricators world-wide raise yields, increase factory productivity, and cut enterprise risks and costs.

Today we have more than 1000 laser photoplotters and 5000 CAM systems installed around the world with local support in every major market. Our customers include the leading PCB fabricators across the global spectrum. Many of them have been with us for more than 20 years.

Key to this success has been our uncompromising pursuit of engineering excellence in all our products. For 25 years our product goals have been best-in-class performance, long-term reliability, and continuous development to keep each user at the cutting-edge of his chosen technology.

Preface

Today RS-274X (also known as Extended Gerber or X-Data) is the de facto standard for PCB image data transfer. Virtually every PCB design system outputs it and every PCB front-end engineering system inputs it. Implementations are thoroughly field-tested and debugged. Its widespread availability allows PCB professionals to exchange image drill and route securely and efficiently.

The RS-274X format is simple, compact and unequivocal. It is easy to interpret. It describes an image with very high precision. It is complete: one single file describes an image. It is portable and easy to debug by its use of printable 7-bit ASCII characters.

A well-constructed RS-274X file precisely defines the PCB image data and the functions of the different image elements. Unfortunately, poorly constructed or simply erroneous RS-274X files also circulate, sometimes leading to unjustified criticism of the format itself.

Errors may be due to a misunderstanding of the format. With more than 25 years experience in CAM software we at Ucamco know which areas are most often misunderstood. This revision of the RS-274X specification explains these areas more clearly.

Other files are not invalid but poorly constructed. Especially troublesome are painted or stroked pads and copper planes. Poorly constructed files take longer to process, require more manual work and increase the risk of errors. This revision of the RS-274X specification recommends constructions to make RS-274X files safer and more efficient, and hence fabrication more reliable, faster and cheaper.

A few words must be said about RS-274-D or Standard Gerber. This format was developed to drive NC machine tools and was used for Gerber vector plotters in the 1960s and 1970s. It is not an image description format. It is amazing that it is still used. It is like using teletype paper tape to transfer text documents. We call on industry experts and professional organizations to discourage the use of the obsolete RS-274-D format.

Although other data transfer formats have come into the market, they have not displaced RS-274X. The reason is simple. More than 90% of the problems in data transfer are due not to limitations in the RS-274X format but to poor practices and, worse, the use of RS-274-D. To quote a manufacturer: “If we would only receive proper RS-274X files, it would be a perfect world.” The new formats are more complex and less transparent to the user, and new implementations inevitably have bugs. Using the common poor practices in the newer and more complex formats makes matters worse, not better. Fabricators have not adopted the new formats. RS-274X remains the standard.

The emergence of RS-274X as a standard for image exchange is the result of effort by many individuals who developed outstanding software for RS-274X files. Without their dedication the widespread acceptance of RS-274X could not have been achieved. Ucamco thanks these dedicated individuals.

Karel Tavernier
Managing Director,
Ucamco

Ucamco
Former Barco ETS

Table of Contents

1 Introduction

1.1 Who should use this Guide?

In order to use this Guide, you should have an understanding of PCB fabrication or PCB design. This Guide is to be used by:

- PCB designers preparing data for conversion to RS-274X
- PCB fabricators creating or using RS-274X data files
- Developers of software applications using RS-274X files

1.2 About This Document

The following conventions are used in this document:

Note:	Provides essential extra information.
Tip:	Provides useful extra information.
Example:	Contains examples of file syntax, commands, settings, etc.
Warning:	Contains an important warning.

1.3 History of the Gerber Format

Gerber Format derives its name from the former Gerber Systems Corporation, a leading supplier of photoplotters.

Originally, Gerber used a subset of the Standard RS-274-D format as input format for its photoplotters. This subset became known as Standard Gerber. It is not an image description format, but a format to drive mechanical machine tools, which photo-plotters originally were, where apertures shapes where physical apertures in a so-called aperture wheel.

In subsequent years, Gerber extended the input format for its range of PCB devices and it actually became a family of formats. The Gerber formats developed into a capable image description format. In 1998 the formats were pulled together and standardized by the publication of the first version of this document. Since then, it has become the de-facto standard for PCB image data.

2 Syntax

2.1 Character Set

An RS-274X image file is expressed in printable 7-bit ASCII code.

2.2 Data Blocks

The file is composed of data blocks. Each data block ends with the mandatory end-of-block character asterisk *. Each data block may contain one or more parameters, codes or coordinates.

For example:

```
X0Y0D02*
X50000Y0DO1*
```

For readability, the following is recommended

- Begin independent data blocks on a new line.
- Do not split a data block over multiple lines.

2.3 Data Types

There are the following data types.

1 Coordinate Data
2 Function Codes
3 Parameters

2.4 Coordinate Data

Coordinate data to define points in the plane. (They were called addresses in the obsolete RS-274-D terminology).

Coordinate data can be:

- X, Y data to define the X, Y coordinates of a point
- I,J data to define an offset in the X, Y direction

The FS Parameter (Format Specification) specifies how the digits must be interpreted.

Right handed orthonormal coordinates are used.

Coordinates are modal. E.g. if an X is omitted, the last X value is used. If no X value was already used in the current layer, the default zero is used.

Offsets are not modal. If I or J is omitted, the default value zero is used.

Examples:

```
X200Y200*           Point  (+200,+200)
Y-300*              Point  (+200,-300)
I300J100*           Offset (+300,+100)
X200Y200I50J50*     Point (+200,+200) and Offset (+50,+50)
X+100I-50*          Point (+100,+200) and Offset (-50,0)
```

2.5 Function Codes

Function codes describe how coordinate data associated with them should be interpreted, e.g. draw a line or draw a circle. (Most, but not all, of these codes are inherited from the now obsolete RS-274-D format. They were called words or codes.)

Example:

```
G74*
```

Each code applies to coordinate data located in the same data block as the code and to all subsequent coordinate data until another code of the same type is encountered, or until a new layer is generated. This continuing action is referred to as *modal*.

For example, G02 specifies clockwise circular interpolation. All coordinate data following it will be interpreted as clockwise circular interpolation until another interpolation code is encountered, or until a new layer is generated.

Codes are described in a later chapter.

2.6 Parameters

Parameters define characteristics applying to an entire image or to a single layer. They are used to interpret the other data types. (Originally, they were called Mass Parameters.)

Parameters operating on the entire image are usually placed at the beginning of the file. Parameters generating a new layer are place at the appropriate place in the file

Parameters consist of two alphabetic characters followed by one or more optional modifiers.

Parameters are delimited by the parameter delimiter %. Each parameters is contained in a data block that itself must be ended by an *. The parameter delimiter must immediately follow the end-of-block without intervening spaces.

For example:

```
%FSLAX23Y23*%
```

Parameters may be entered single or grouped between delimiters, up to a maximum of 4096 characters between delimiters..

For example:

```
%SFA1.0B1.0*ASAXBY*%
```

Line breaks are permitted between parameters to improve readability.

For example:

```
%SFA1.0B1.0*
```

ASAXBY*%

For readability and simplicity it is recommended to have one parameter per line.

Use an explicit decimal point with all numerical values associated with a parameter. If the decimal point is omitted, an integer value is assumed.

The syntax for parameters is:

%Parameter code<required modifiers>[optional modifiers]*%

Syntax	Comments
Parameter code	2-character code (AD, AM, FS, etc…)
<required modifiers>	Must be entered to complete the definition
[optional modifiers]	Entry depends on the required modifiers

3 Image Generation

3.1 Information Layers

An information layer is a component of the image consisting of one or more consecutive data blocks. The data blocks of an information layer create a layer image. A polarity (dark or clear) may be assigned to a layer.

Note: An information layer in an RS-274X must not be confused with a layer in a PCB.

Information layer images are superposed to the final image in the order they appear in the file. Dark layer image marks the image, clear layer image clears (unmarks, rubs, erases) the image in *all* the lower layers. In other words, after superposing a clear layer, the complete image in that layer is cleared and will be white in the final image, whatever was there before. Subsequent dark layers may again mark the cleared area. The order of exposing the layers is important in the presence of clear layers.

In addition, an individual layer may be “knocked out “of the surrounding graphic image, and may be repeated and/or rotated individually.

3.2 Generating the Information Layer Image

The RS-274X file defines a stream of graphic operators and operands. Most operators generate a *graphics object*, such as a flash or draw. This results in a stream of graphic objects which are added to the current layer. These objects mark the image area, regardless of the pre-existing objects at that location.

The graphic operator is reset at the start of the file and at the start of each new layer. Within a layer, the graphic operator is modified with Function Codes.

The operands are supplied by the stream of Coordinate Data.

The action of the graphic operator depends not only on the operands, but also on the implicitly defined *current point*. The current point is set to the origin or (0,0) at the start of the file and at the start of each new layer. The graphic operator typically creates graphic object that starts at the current point, and ends at a point defined by the coordinate data; the current point is then set to this endpoint; this end point then becomes the begin point of the next object.

4 Parameters

The following parameters exist:

1. **Aperture parameters** describe the shape of lines and components
2. **Directive parameters** control overall file processing
3. **Image parameters** supply information about an entire image
4. **Layer-specific parameters** describe processing of one or more data layers

Parameter		Function	Comments	Default
Required	**Optional**			
	AS	Axis select	Single use recommended. When used more than once, enter these parameters at the beginning of a layer. These codes do not generate a new layer.	A=X, B=Y
FS		Format statement		
	MI	Mirror Image		No mirror
	MO	Mode (inch or millimeter units)		Inch
	OF	Offset		A=0, B=0
	SF	Scale Factor		A=1.0, B=1.0
	IJ	Image Justify	Use only once at the beginning of the file.	No justification
	IN	Image Name		
	IO	Image Offset		A=0, B=0
	IP	Image Polarity		Positive
	IR	Image Rotation		0
	PF	Imagesetter Film		
	AD	Aperture Description	May be used singly or may be layer-specific. Enter these parameters at the beginning of the file or layer.	
	AM	Aperture Macro		
	LN	Layer Name		
	LP	Layer Polarity		Positive
	KO	Knockout		Off
	SR	Step and Repeat		A=1, B=1
	RO	Rotate		No rotation

Parameter Overview

Parameters are usually placed at the beginning of the file in the order shown above. Layer-specific parameters are embedded within the file at the appropriate location.

4.1 AD - Aperture Definition

The AD parameter is used to describe the apertures (D-codes) used in the RS-274X file. All apertures used must be described in terms of shape and size. The AD parameter must precede use of the associated aperture D-code. A definition remains in effect until redefined.

Two kinds of apertures may be used: *standard apertures* and *special apertures*.

In general, aperture parameters apply to an entire file. An exception is an embedded AD parameter, which will generate a new layer if it redefines a D-code previously used in the image data.

4.1.1 Standard Apertures

The AD parameter identifies standard apertures by D-code number and describes them in terms of shape and size. Apertures may be *solid* or *open* (with centered hole).

4.1.2 Special Apertures

The AD parameter is also used to assign a D-code to a file containing an aperture description created with the AM (Aperture Macro) parameter. See the AM parameter description for further information on aperture macros.

4.1.3 Syntax Rules

- Begin and end each parameter block with the parameter delimiter %.
- Within the AD parameter block, separate each modifier by an X.
- Dimensions must be positive.
- The range of D-code is from 10 to 999.
- Rotation is expressed in integer degrees; positive is counterclockwise, negative is clockwise.

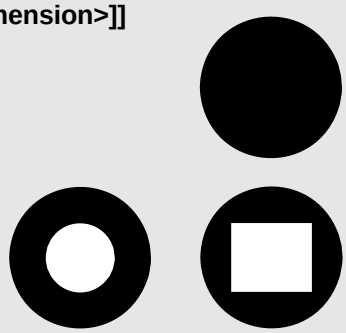
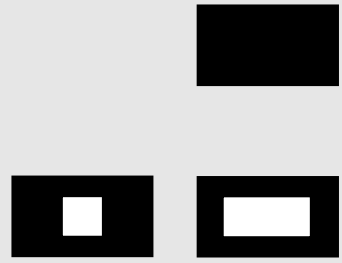
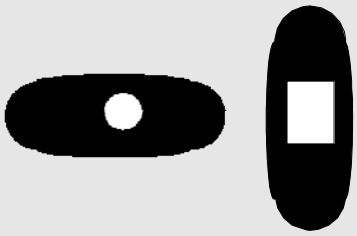
4.1.4 Data Block Format

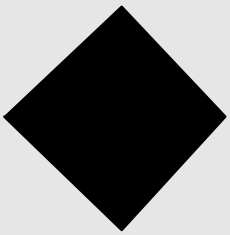
The syntax for the AD parameter is:

%ADD<D-code number><aperture type>,<modifier>[X<modifer>]*%

Syntax	Comments
ADD	**AD** for Aperture Description and **D** for D-code
<D-code number>	The D-code number being defined (10 - 999)
<aperture type>, <modifier>[X<modifier>]	The Aperture Type has two possible formats: A standard aperture (C, R, O or P) with modifiers separated by X (all dimensions are positive) or an aperture macro name previously defined by the AM parameter. Notes: Be sure to use the units (inches or millimeters) specified by the MO parameter for all modifiers Holes in apertures are transparent

4.1.5 Aperture Definition with Standard Apertures

Syntax	Comments
C,<outside diameter>[X<X-axis hole dimension >[X<Y-axis hole dimension>]] 	**Circle**. To define a solid aperture, enter only the diameter. To define a hole, enter one dimension for a round hole, two for a rectangular hole. The hole must fit within the aperture. Both aperture and hole will be centered. For example, **%ADD10C,.05X0.025*%** D-code 10 is a .05 circle with a .025 round hole in the center.
R,<X-axis dimension>X<Y-axis dimension>[X<X-axis hole dimension>X<Y-axis hole dimension>] 	**Rectangle or square**. May be solid or open. If the X axis dimension equals the Y dimension, the aperture will be square. To define a solid aperture, enter only the X and Y dimensions; omit the hole dimensions. To define a hole, enter one dimension for a round hole, two for a rectangle. The hole must fit within the aperture. Both rectangle and hole will be centered. For example, **%ADD22R,0.020X0.040*%** D-code 22 is a .02 x .04 solid rectangle.
O,<X-axis dimension>X<Y-axis dimension>[X<X-axis hole dimension>[X<Y-axis hole dimension>]] 	**Obround (oval).** May be solid or open. If the X dimension is larger than Y, the shape will be horizontal. If the X dimension is smaller than Y, the shape will be vertical. To define a solid aperture, enter only the X and Y dimensions; omit the hole dimensions. To define a hole, enter one hole dimension for a round hole, two for a rectangular or square hole. If open, the hole must fit within the aperture. For example, **%ADD22O,0.020X0.04X0.005X0.010*%** D-code 22 is a vertical obround .02 wide x .04 high with a .05 x .01 rectangular hole.
P,<outside dimension>X<number of sides>[X<degrees of rotation>[X<X-axis hole dimension>X<Y-axis hole dimension>]]	**Regular polygon**. May be solid or open. To define a solid aperture, enter only the outside dimension and number of sides (3 to 12). Without rotation, one vertex is on the X-axis through the center point. May be rotated. If open, the hole must fit within the outside dimension. Holes do not rotate.

	For example, **%ADD17Diamond,.030X4X0.0*%** D-code 17 is a polygon within an outside dimension of .03, 4 sides, with no center hole. Note: To enter hole dimension modifiers you need to enter a rotation angle. Enter rotation angle 0 if you do not want to rotate.

4.1.6 Examples

Syntax	Comments
%ADD10C,.025*%	Define D-code 10: 25 mil round
%ADD22R,.050X.050X.027*%	Define D-code 22: 50 mil square with 27 mil round hole
%ADD57O,.030X.040X.015*%	Define D-code 57: obround 30 x 40 mil with 15 mil round hole
%ADD30P,.016X6*%	Define D-code 30: polygon (hexagon), 16 mil outside dimension with 6 sides
%ADD15CIRC*%	Define D-code 15: a special aperture described by aperture macro CIRC defined previously by an aperture macro

4.2 AM - Aperture Macro

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

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 aperture shapes.
- 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 an arithmetic function of other macro variables.

4.2.1 Contents

An aperture macro contains the following elements:

- Aperture macro name
- One or more of the seven aperture primitives. (See table below)
- Primitive modifiers specifying exposure, position, dimensions, etc.
- Variable primitive modifiers supplied by the AD parameters
- Optional embedded comment block
- Arithmetic operators

4.2.2 Syntax Rules

- Like other parameters, enclose each parameter block with the delimiter %.
- Within the AM parameter block, separate each primitive and modifier group by the end-of-block character *.
- Separate modifiers by commas within each primitive group.
- Modifiers may be number, such as 0, 1, 2, or 9.05, or they may be variable modifiers supplied by the AD parameter using the special aperture.
- 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 depends on the primitive. See table below.
- 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.

- The following arithmetic operators can be used with variable modifiers:

Operator	Function
+	Add
-	Subtract
/	Divide
x	Multiply
=	Equate
n	Number, with or without decimal point

The standard arithmetic precedence rules apply.

- Exposure can be *on* or *off*. Exposure on creates a solid part of the aperture. Exposure off creates a hole in it. (Note: A hole is transparent. One sees the objects below it. This is *not* the same as clear exposure in Layer Polarity, where all object below are cleared or erased.) Exposure is set with the exposure modifier:

  0 = off

  1 = on

  2 = toggle exposure mode (0 if exposure mode was not yet set)

- Rotation angle is expressed in integer degrees; positive for counterclockwise rotation, negative for clockwise rotation.

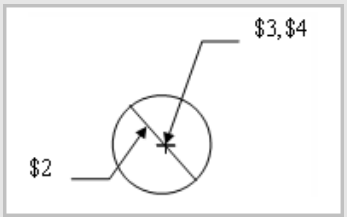
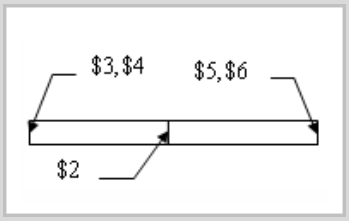
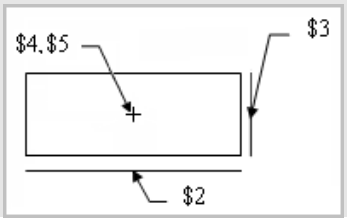
4.2.3 Data Block Format

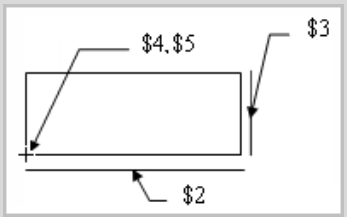
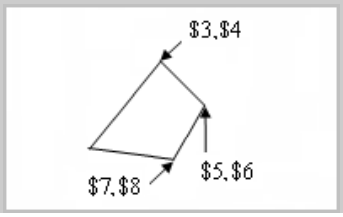
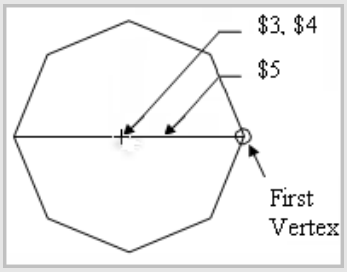
The syntax for the AM parameter is:

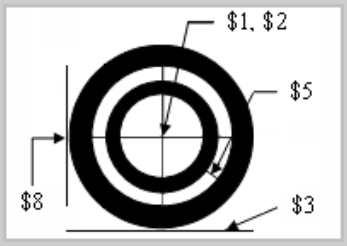
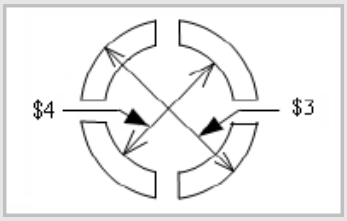
%AM<aperture macro name>*<primitive number>,<modifier$1>,<modifier$2>,[<...>]*[<primitive number>[<modifiers>]]*...*%

Syntax	Comments
AM	**AM** for Aperture Macro
<aperture macro name>	The name of the Aperture macro
<primitive number>, <modifier$1>, <modifier$2>, [<...>]	The primitive number with modifiers. The primitive number specifies the shape (e.g. polygon) and the modifier specify parameters (.e.g. diameter). The modifiers depend on the primitive. Use either a value (e.g. 0.050 for a diameter) or a variable placeholder (for example, $1 for exposure).

4.2.4 Primitives

Prim` Num	Graphical Description	Var Mod	Description
1	**Circle** 	**$1**	Exposure on/off/toggle
		$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/toggle
		$2	Line width
		$3	X start point
		$4	Y start point
		$5	X end point
		$6	Y end point
		$7	Rotation angle around the center point of the line.
21	**Line (center):** a centered rectangle defined by width, height, and center point. The end points are rectangular. 	**$1**	Exposure on/off/toggle
		$2	Rectangle width
		$3	Rectangle Height
		$4	X center point
		$5	Y center point
		$6	Rotation angle around the center point of the line.

22	**Line (lower left):** a rectangle defined by width, height, and the lower left point. The end points are rectangular. 	**$1**	Exposure on/off/toggle
		$2	Width
		$3	Height
		$4	X lower left point
		$5	Y lower left point
		$6	Rotation angle around the center point of the line.
4	**Outline**: the area enclosed by the outline defined by the start point and n additional points. (Note: Do not use very large values for n as applications may have severe implementation limits. Older versions of this document suggested an implementation limit of 50.) The X and Y coordinates are *not* modal. If the last point is not equal to the start point, the outline is closed. Self-intersecting outlines are not allowed. 	**$1**	Exposure on/off/toggle
		$2	The number of additional points
		$3	X start point
		$4	Y start point
		$5	X point #1
		$6	Y point #1
		$...	Subsequent points
		$last	Rotation angle around the start point of the outline.
5	**Polygon**: an n-sided regular polygon (n 3 to 10 inclusive), a center point, diameter, and rotation. 	**$1**	Exposure on/off/toggle
		$2	Number of sides
		$3	X center point
		$4	Y center point
		$5	Diameter
		$6	Rotation angle around the center point of the polygon. Without rotation, one vertex is on the X-axis through the center.

6	**Moiré:** a cross hair centered on n concentric circles defined by the center point, outside diameter, line thickness, and gap between circles. Exposure is always on. 	**$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 angle around the center point
7	**Thermal**: Ring (annulus) interrupted by four gaps. See image. Exposure is always on. 	**$1**	X center point
		$2	Y center point
		$3	Outside diameter
		$4	Inside diameter
		$5	Gap thickness
		$6	Rotation angle around the center point of the thermal. Without rotation, the gaps are on the axes through the center.

4.2.5 Examples

All units are inch

Example 1

Defines an aperture macro named DONUTFIX consisting of two concentric circles with fixed diameter sizes.

%AMDONUTFIX*1,1,0.100,0,0*1,0,0.080,0,0*%

Syntax	Comments
AMDONUTFIX	Define an aperture macro named DONUTFIX
1,1,0.100,0,0	Circle (1), exposure on (1), diameter (0.100), X center (0), Y center (0)
1,0,0.080,0,0	Circle (1), exposure off (0), diameter (0.080), X center (0), Y center (0)

The AD parameter using this aperture macro will look like the following:

%ADD33DONUTFIX*%

Example 2

Defines an aperture macro named DONUTVAR consisting of two concentric circles with variable diameter sizes:

%AMDONUTVAR*1,1,$1,$2,$3*1,0,$4,$2,$3*%

Syntax	Comments
%AMDONUTVAR*	Define an aperture macro named DONUTVAR
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), X center and Y center ($2 and $3, same as first circle)

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

%ADD34DONUTVAR,0.100X0X0X0.080*%

Example 3

Defines an aperture macro named DONUTCAL consisting of two concentric circles with diameter of the second circle defined as a function of the diameter of the first:

%AMDONUTCAL*1,1,$1,$2,$3*$4=$1x0.75*1,0,$4,$2,$3*%

Syntax	Comments
AMDONUTCAL	Define an aperture macro named DONUTCAL
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
$4=$1x0.75	Define variable $4 to be used to calculate the diameter of the inner circle. The diameter of this circle is 0.75 times the diameter of the outer circle.
1,0,$4,$2,$3	Circle (1), exposure off (0), diameter ($4), and center point X, Y ($2, $3, same as the first circle).

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

%ADD35DONUTCAL,0.020X0X0*%

This defines a donut with outer circle 0.02 and inner circle 0.015 inch

4.3 Directive parameters

AS	Axis Select
FS	Format Statement
MI	Mirror Image
MO	Mode of units
OF	Offset
SF	Scale Factor

Directive parameters control overall file processing.

It is recommended to place directive parameters at the beginning of the file. Each directive parameter controls processing until another one is encountered It is recommended to use each directive parameter only once in a file.

Directive parameters do not generate a new layer.

4.3.1 AS – Axis Select

The AS parameter is used to assign any two data axes to the output device's A and B axes.

4.3.1.1 Data Block Format

The syntax for the AS parameter is:

AS A<X or Y>B<X or Y>*

Syntax	Comments
AS	**AS** for Axis Select
AX or AY	Assign output device axis A to data axis X or Y Default is AX
BX or BY	Assign output device axis B to date axis X or Y Default is BY

4.3.1.2 Example

Syntax	Comments
%ASAYBX*%	Assign output device axis A to data axis Y and output device axis B to data axis X

4.3.2 FS – Format Specification

The FS parameter is used to define the format of the input coordinate data and to define the allowable N, G, D, and M-code lengths. It should be the first RS-274X parameter in the file. It is recommended that only one be used in the file. It is usually the first parameter.

The FS parameter allows you to specify the following format characteristics:

- Number of integer and decimal places in coordinate data (coordinate format)
- Zero omission (leading or trailing zeroes omitted)
- Absolute or incremental coordinate notation
- N-code length
- G-code length
- D-code length
- M-code length
-  Note: Explicit decimal points are not allowed in coordinates.

4.3.2.1 Coordinate Format

Coordinate format specifies how many integer and how many decimal places to expect in the coordinate data. For example, the 2.3 format specifies 2 integer and 3 decimal places. A maximum of six integer and six decimal places may be specified (nnnnnn.nnnnnn). Different formats may be defined for the X and Y axes. Signs are allowed. The “+” sign may be omitted.

4.3.2.2 Zero Omission

Zero omission compresses data by omitting *either* leading or trailing zeroes from the coordinate values.

With *leading zero omission* some or all leading zeroes may be omitted but all trailing zeroes must be retained. To interpret the coordinate string, it is first padded with zeroes in front until its length fits the coordinate format. For example, with 2.3 coordinate format, “015” is padded to “00015” and therefore represents 0.015.

With *trailing zero omission* some or all trailing zeroes may be omitted but all leading zeroes must be retained. To interpret the coordinate string, it is first padded with zeroes at the back until its length fits the coordinate format. For example, with 2.3 coordinate format, “15”is padded to “15000” therefore represents 15.000.

With *no zeroes omitted* all zeroes are present, the format length fits and there is no padding. It is conventional to indicate leading zero omission.

4.3.2.3 Absolute or Incremental Notation

Coordinate values may be expressed as either absolute distances from the origin or as incremental distances from the preceding coordinate position.

4.3.2.4 Function code Lengths

The FS parameter can be used to specify length limits for the following standard RS-274-D-codes: The codes are described further in this manual.

4.3.2.5 *Data Block Format*

The syntax for the FS parameter is:

%FS<L or T><A or I>[Nn][Gn]<Xn><Yn>[Dn][Mn]

Syntax	Comments
FS	**FS** for Format Specification
<L or T>	Use L to omit leading zeroes. Use T to omit trailing zeroes.
<A or I>	Use A for absolute coordinate values. Use I for incremental coordinate values.
[Nn],[Gn],[Dn] or [Mn]	Enter the code and an integer length limit, for example, N2 to specify two-digit sequence codes.
<Xn> and <Yn>	Enter X or Y and the number of integer and decimal places in the coordinate data for each axis, for example, X23 for X-axis data with two integer and three decimal places (99.999). 6.6 is the maximum. The X and Y axes may have different values.

4.3.2.6 *Example*

Syntax	Comments
%FSLAX25Y25*%	Coordinate data will have leading zeros omitted and be expressed as absolute positions with two integer and five decimal places in both axes.

4.3.3 MI – Mirror Image

The MI parameter is used to turn mirror imaging either on or off. When on, all A- and/or B-axis data following the parameter will be mirrored (that is, inverted or multiplied by -1) until another MI command is used. Notice that mirroring A-axis data flips the image about the B axis. Mirroring B-axis data flips the image about the A axis.

Notes: MI does not mirror special apertures.
Use the AS parameter to correlate data axes with output device axes

4.3.3.1 Data Block Format

The syntax for the **MI** parameter is:

%MI[A<0 or 1>][B<0 or 1>]*%

Syntax	Comments
MI	**MI** for Mirror image
A<0 or 1>	Use A0 to disable mirroring Use A1 to invert the A-axis. The image will be flipped over the B-axis.
B<0 or 1>	Use B0 to disable mirroring Use B1 to invert the B-axis. The image will be flipped over the A-axis.

4.3.3.2 Examples

Syntax	Comments
%MIA0B0*%	No mirroring of A or B axis (default)
%MIA0B1*%	No mirroring of A-axis data Invert B-axis data, flipping the image over the A-axis.

4.3.4 MO – Mode

The MO parameter specifies that dimension data should be interpreted as inches or millimeters. If the MO parameter is missing inches are assumed.

Note: Use the FS parameter to specify the integer and decimal place format.

4.3.4.1 Data Block Format

The syntax for the **MO** parameter is:

%MO<IN or MM>*%

Syntax	Comments
MO	**M** for Mode
<IN or MM>	Use IN to specify units in inches Use MM to specify units in millimeters

4.3.4.2 Examples

Syntax	Comments
%MOIN*%	Dimensions in inches (default)
%MOMM*%	Dimensions in millimeters

4.3.5 OF - Offset

The OF parameter is used to offset the final image up to plus or minus 99999.99999 units from the imaging device 0,0 point. The data may be offset along the imaging device A or B axis, or both. Values used with the OF parameter are expressed in units specified by the MO parameter, are always absolute, and are used primarily with absolute coordinate data. Incremental coordinate data may be offset simply by moving the imaging device to the desired offset position before starting the image. The FS parameter specifies whether the data is absolute or incremental.

Note: If an embedded FS parameter changes the format from absolute to incremental, the OF parameter value is saved and reinstated. The next FS parameter returns the format to absolute.

4.3.5.1 Data Block Format

The syntax for the **OF** parameter is:

OF[A<soffset value>][Bsoffset value>]

Syntax	Comments
OF	**OF** for Offset
A<sn>	The offset along the output device A axis s = optional sign (+ or -) n = offset value
B<sn>	The offset along the output device B axis s = optional sign (+ or -) n = offset value

4.3.5.2 Examples

Syntax	Comments
%OFA0B0*%	No offsets (default)
%IOA1.0B-1.5*%	Offset of 1 unit along the A axis Offset of -1.5 units along the B axis
%IOB5.0*%	Offset of 0 units along the A axis Offset of 5 units along the B axis

4.3.6 SF – Scale Factor

The SF parameter is used to specify a scale factor from 0.0001 to 999.99999 for the output device A- and/or B-axis data. The factor may be different for each axis. All data following the parameter will be multiplied by the factor until another SF parameter is encountered.

Note: Use the AS parameter to correlate data axes with output device axes

4.3.6.1 Data Block Format

The syntax for the **SF** parameter is:

%SF[A<factor>][B<factor>]*%

Syntax	Comments
SF	**SF** for Scale Factor
A<factor>	The A-axis data scale factor
B<factor>	The B-axis data scale factor

4.3.6.2 Example

Syntax	Comments
%SFA1B1*%	Scale factor 1 (default)
%SFA.5B3*%	Multiply A-axis data by 0.5 Multiply B-axis data by 3

4.4 Image parameters

IJ Image Justify

IN Image Name

IO Image Offset

IP Image Polarity

IR Image Rotation

PF Plot Film

Image parameters affect the entire image. If an image parameter occurs more than once, the last one encountered is valid for the whole file. It is recommended to use the Image Parameters only once and place it at the beginning of the file.

4.4.1 IN – Image Name

The IN parameter is used to assign a name of up to 77 alphanumeric characters to the entire image of the RS-274X file. Information layers may also be named; see the LN parameter.

4.4.1.1 Data Block Format

The syntax for the **IN** parameter is:

%IN<character string>*%

Syntax	Comments
IN	**IN** for Image Name
<character string>	up to 77 alphanumeric characters.

4.4.1.2 Examples

Syntax	Comments
%INSOLDERMASK*%	Image name “Soldermask”
%INPANEL_1*%	Image name “Panel_1”

4.4.2 IJ – Image Justify

The IJ parameter is used to override the absolute data coordinates for final placement of the image on the output device. The image may be centered or may be placed at an absolute position relative to the lower left of the platen. When more than one IJ parameter appears in the data, the final entry encountered is the one used.

Note: When centered, the pixel coordinates for the platen reside in the first quadrant (+X and +Y). X and Y are positive numbers, greater than zero and less than the platen size.

4.4.2.1 Data Block Format

The syntax for the **IJ** parameter is:

%IJ[A<parameter >B<parameter>][<offset>]*%

Syntax	Comments
IJ	**IJ** for Image Justify
A	The imagesetter A axis justification
<L or C>	Left or Center
B	The imagesetter B axis justification
<L or C>	Left or Center
<offset>	The starting position offset relative to 0,0

4.4.2.2 Examples

Syntax	Comments
%IJ*%	Left justify in X and lower justify in Y
%IJAC*%	Center justify in X and lower justify in Y
%IJACB,100*%	Center justify in X, 1 units in Y
%IJALB,10*% or %IJB.100*%	Left justify in X, offset, 1 units in Y
%IJA1B1 *%	Offset image 1 unit in X and Y

4.4.3 IO – Image Offset

The IO parameter is used to offset an image from the 0,0 point. The offset is expressed as an increment along the output device A and B axis. The offset may be different for each axis and may be entered for a single axis.

Notes: Use the MO parameter to set the unit.
Use the AS parameter to correlate data axes with output device axes.

4.4.3.1 Data Block Format

The syntax for the **IO** parameter is:

%IOA<sn>B<sn>*%

Syntax	Comments
IO	**IO** for Image Offset
A<sn>	The offset along the output device A axis s = optional sign (+ or -) n = offset value
B<sn>	The offset along the output device B axis s = optional sign (+ or -) n = offset value

4.4.3.2 Examples

Syntax	Comments
%IOA0B0*%	No offsets (default)
%IOA1.0B-1.5*%	Offset of 1 unit along the A axis Offset of -1.5 units along the B axis
%IOB5.0*%	Offset of 0 units along the A axis Offset of 5 units along the B axis

4.4.4 IP – Image Polarity

The IP parameter is used to specify the positive or negative polarity of the entire file image. This *image polarity* differs from *layer polarity*, which is specified by the LP parameter and which applies only to one or more data layers of the entire image. With positive polarity the image is exposed or displayed as is. With negative polarity dark areas are displayed as clear, and clear areas displayed as dark,


Note: Some CAD applications are adding a positive background behind the objects for visualization.

4.4.4.1 Data Block Format

The syntax for the **IP** parameter is:

%IP<POS or NEG>*%

Syntax	Comments
IP	**IP** for Image Polarity
<POS>	Output the image with positive polarity. (Default)
<NEG>	Output the image with negative polarity

4.4.4.2 Examples

Syntax	Comments
%IPNEG*%	Output the image with negative polarity  Note: IPNEG is not swapping the specified polarity

4.4.5 IR – Image Rotation

The IR parameter is used to rotate the entire image counterclockwise in 90° increments around the 0,0 coordinate. All apertures follow the rotation.

If there is no IR parameter, a rotation of 0° is assumed.

4.4.5.1 Data Block Format

The syntax for the **IR** parameter is:

%IR<n>*% with n 0, 90, 180 or 270

Syntax	Comments
IR	**IR** for Image Rotation
<0>	Rotate the image over 0°; no rotation
<90>	Rotate the image over 90°
<180>	Rotate the image over 180°
<270>	Rotate the image over 270°

4.4.5.2 Examples

Syntax	Comments
%IR0*%	No rotation (default)
%IR90*%	Rotate the image over 90° counterclockwise
%IR270*%	Rotate the image over 270° counterclockwise  Note: This is equal to a rotation of the image over 90° clockwise

4.4.6 PF – Image Film

The PF parameter indicates to the operator the type of film (or other media) to be used to image the data file

4.4.6.1 Data Block Format

The syntax for the **PF** parameter is:

%PF<name>*%

Syntax	Comments
PF	**PF** for Image Film
<name>	Up to 20 alphanumeric characters.The asterisk character cannot be used inside the name.

4.4.6.2 Examples

Syntax	Comments
%PFFILM_28X48*%	Image film name is “FILM_28X48”

4.5 Layer-specific parameters

KO Knockout

LN Layer Name

LP Layer Polarity

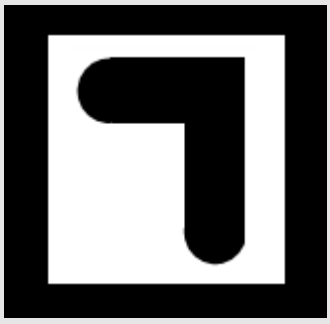
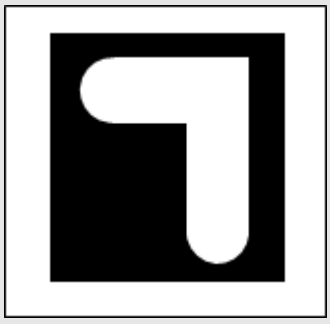
SR Step and Repeat

Layer-specific parameters specify the processing of information layers. (Information layers are not to be confused with board layers). They may be used more than once in a file. Layer-specific parameters *always* generate a new layer.

Note: Generating a new layer resets the graphic operator and the current point.

4.5.1 KO – Knock Out

The KO parameter is used to specify that a rectangular region of the image will have the opposite polarity of the image or layer in which it occurs (either clear or dark), making it a "knockout" from the surrounding region.

	
Positive image polarity Dark layer polarity Clear knockout	Negative image polarity Clear layer polarity Dark knockout

KO is typically used to:

- Set the initial background polarity of the final image
- Knock out a region around a component
- Disable knockout on the current information layer

A knockout can be defined in one of two ways:

- Lower left point and height and width
- Border width around a component

When a knockout is defined around a component, the knockout is applied to all data following the KO parameter until the knockout is disabled. To disable a previously defined knockout, enter KO with no modifiers.

4.5.1.1 *Data Block Format*

The syntax for the **KO** parameter is:

%KO[C or D][X<coordinate>Y<coordinate>I<width>J<height>]

Syntax	Comments
KO	**KO** for Image Rotation
[C or D]	Use C for clear polarity of the background Use D for dark polarity of the background To create a knockout defined by the data extents, do not enter modifiers. To disable a previously enabled knockout, enter neither C nor D.
X<coordinate>	Lower left X coordinate
Y<coordinate>	Lower left Y coordinate
I<width>	Width of the knockout area
J<height>	Height of the knockout area

4.5.1.2 *Examples*

Syntax	Comments
%KODX0I0I20J26*%	Create a dark knockout that extends from 0,0 to 20,26.
%KOCK.050*%	Create a clear knockout defined by the extents of the data following the KO parameter plus a border of 0.50 units.
%KOD*%	Create a dark knockout defined by the extents of the data following the KO parameter.
%KO*%	Disable a previously enabled knockout.

4.5.2 LN – Layer Name

The LN parameter is used to assign a name of up to 77 alphanumeric characters to the information layer that follows the parameter in the RS-274X file.

 Note: Use the IN parameter to name the entire image file.

4.5.2.1 Data Block Format

The syntax for the **LN** parameter is:

%LN<character string>*%

Syntax	Comments
LN	**LN** for Layer Name
<character string>	Up to 77 alphanumeric characters.

4.5.2.2 Examples

Syntax	Comments
%LNGREEN*%	Layer name “Internal VCC”

4.5.3 LP – Layer Polarity

The LP parameter is used to specify the dark (positive) or clear (negative) polarity of the information layer or layers following it. The layer polarity applies to all data following the LP parameter until another LP parameter is encountered. (Layer polarity is not image polarity, which is specified by the IP parameter, and which applies to the entire image.)

4.5.3.1 Data Block Format

The syntax for the **LP** parameter is:

%LP<C or D>*%

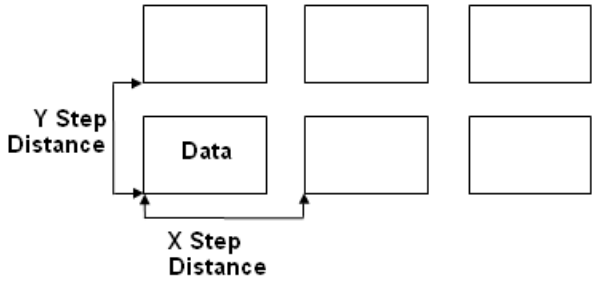
Syntax	Comments
LP	**LP** for Layer polarity
<C or D>	Use C for clear polarity Use D for dark polarity

4.5.3.2 Examples

Syntax	Comments
%LPD*%	Use dark polarity (default)
%LPC *%	Use clear polarity for all succeeding data

4.5.4 SR – Step and Repeat

The SR parameter is used to duplicate the data following the parameter a specific number of times (repeats) at a regular intervals (steps). The number of times the image is repeated and the space between repeats may be specified independently for X and Y data. When entered without parameters, it is used to disable a previous SR parameter.



4.5.4.1 Data Block Format

The syntax for the **SR** parameter is:

%SR[X<repeats>][Y<repeats>][I<X-axis step>][J<Y-axis step>]

Syntax	Comments
SR	**SR** for Step and Repeat
X<repeats>	The number of times the data is repeated along the X-axis
Y<repeats>	The number of times the data is repeated along the Y-axis
I<X-axis step>	The distance between the X-axis repeats This parameter is only valid if X<repeats> value is > 1
J<Y-axis step>	The distance between the Y-axis repeats This parameter is only valid if Y<repeats> value is > 1

4.5.4.2 Example

Syntax	Comments
%SRX1Y1I0J0*%	Step and Repeat with 1 data block, not repeated (default)
%SRX2Y3I2.0J3.0*%	Repeat the image 2 times along the X axis and 3 times along the Y axis. X-axis repeats will be spaced 2.0 units apart. Y-axis repeats will be spaced 3.0 units apart.
%SRX4I5.0J2*%	Repeat the image 4 times along the X axis with 5.0 units from one step to the next. The J modifier will be ignored because no Y repeats were specified.
%SR*%	Disable a previous SR parameter.

5 Function Codes

Function codes specify how coordinate data should be interpreted. Function codes apply to Coordinate Data in the same block as well as to subsequent Coordinate Cata. They do not affect Coordinate Data preceding the block in which they occur.

Functions Codes and Coordinate Data normally follow the Parameters.

Codes are grouped as follows. Assignment over groups is partly historical, inherited from the obsolete RS-274-D format:

1 **N-codes** assign a sequence number to data blocks. Sequence numbers may range from 0 to 99999. N-codes are optional. They have *no* effect on the image. They are specified mainly for historic compatibility.

2 **G-codes** (general functions) specify how to interpolate and move to the coordinate locations following the code until changed or until a new layer is generated (modal).

3 **D-codes** (drawing functions) select and control tools, specify line type, etc.

4 **M-codes** (miscellaneous functions) perform the program end.

The next section lists the D-codes, G-codes and M-codes. Subsequent sections explain the more intricate codes in detail.

5.1 Overview

Code	Function	Comments
D01	Exposure and Draw mode on.	Draws a line using the current aperture. You cannot draw using a special aperture s. The current point is moved to the end point.
D02	Exposure off.	Generates no image. The current point is moved. This is the default state
D03	Set Flash mode.	Flash the current aperture. The current point is moved to the flash point.
D10-D999	Select an aperture defined by an AD parameter	

D-codes

Code	Function	Comments
G01	Linear interpolation	A modified of the draw operator. Default
G02	Clockwise circular interpolation	A modified of the draw operator.
G03	Counter clockwise circular interpolation	
G04	Ignore data block	Used for comments
G36	Turn on Outline Fill	See below.

G37	Turn off Outline Fill	
G54	Select aperture	Optionally precedes an aperture D-code. This code is historic, and its use is not required..
G70	Specify inches	See also MO parameter.
G71	Specify millimeters	
G74	Cancel multi quadrant mode.	A modified of the circular interpolation. Single quadrant is the default.
G75	Enable multi quadrant mode	
G90	Specify absolute format	See also FS parameter.
G91	Specify incremental format	

G-codes

Code	Function	Comments
M00	Program stop	No effect on the image. Historic code.
M01	Optional stop	No effect on the image. Historic code.
M02	End of program	Ends image generation. Every file must end in a M02

M-codes

5.2 Linear Interpolation (G01)

Linear interpolation generates a straight line from the current point to the X, Y coordinate specified by the data block. The current point is set to the X, Y coordinate.

5.2.1 Data Block Format

The syntax for the **Linear Interpolation code**s is:

Gn Xsn Ysn Dn

Syntax	Comments
Gn	Linear interpolation mode
Xsn	s = optional sign (+ or -) n = X coordinate of line endpoint
Ysn	s = optional sign (+ or -) n = Y coordinate of line endpoint
Dn	Exposure n = 01 => On n = 02 => Off

5.3 Circular Interpolation (G02, G03, G74, G75)

Circular interpolation generates a circular arc from the current point to the X, Y coordinate in the data block. The current point is then set to the X, Y coordinate.

There are two orientation modes:

- clockwise mode, specified by G02
- counter clockwise mode, specified by G03

The orientation is defined around the center of the arc, moving from begin to end.

There are two quadrant modes:

- single quadrant mode
- multi quadrant mode

In single quadrant mode the arc is not allowed to extend over more than 90°. The following relation must hold: 0° =< |arc angle| =<90°. If the start point of the arc is equal to the end point, the arc has length zero, i.e. it covers 0°. A data block is required for each quadrant. A minimum of four data blocks is required for a full circle.

In multi quadrant mode the arc is allowed to extend over more than 90°. To avoid ambiguity between 0° and 360° arcs the following relation must hold: 0° < |arc angle|. If the start point of the arc is equal to the end point, the arc is a full circle of 360°.

The default mode is single quadrant. The G74 and G75 command allow switching between the two modes. A data block containing G75 enables multi quadrant mode. Every block following it will be interpreted as multi quadrant, until cancelled by a G74. A data block containing G74 code turns off multi quadrant mode, reverting to single quadrant mode.

5.3.1 Single Quadrant Mode

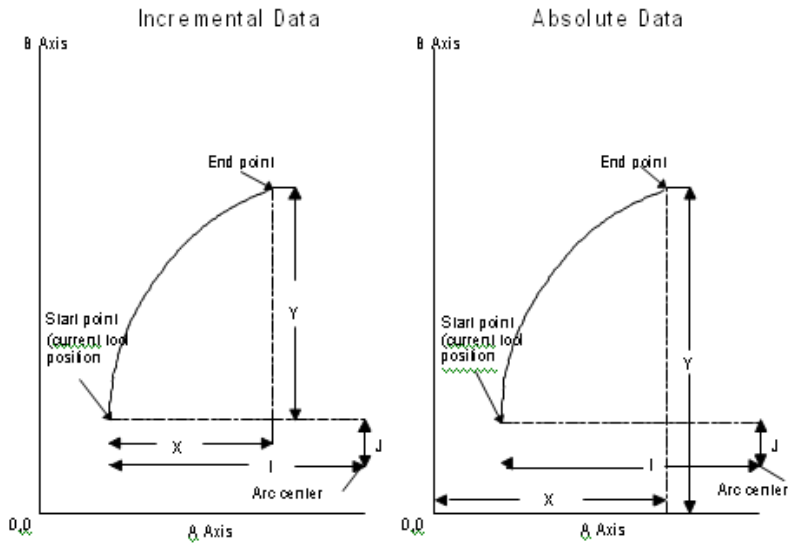
5.3.1.1 Data Block Format

The syntax in **Single Quadrant Circular Mode** is:

Gn Xsn Ysn In Jn Dn

Syntax	Comments
Gn	Specifies direction of arc n = 02 => clockwise n = 03 => counter clockwise
Xsn	s = optional sign (+ or -) n = X coordinate of arc endpoint
Ysn	s = optional sign (+ or -) n = Y coordinate of arc endpoint
In	n = The offset between the arc start point and the center parallel to the X axis. The value is always positive. A sign is not allowed. The direction to the center is determined implicitly.
Jn	n = The offset between the arc start point and the center parallel to the Y axis. The value is always positive. A sign is not allowed. The direction to the center is determined implicitly.
Dn	Exposure n = 01 => On n = 02 => Off

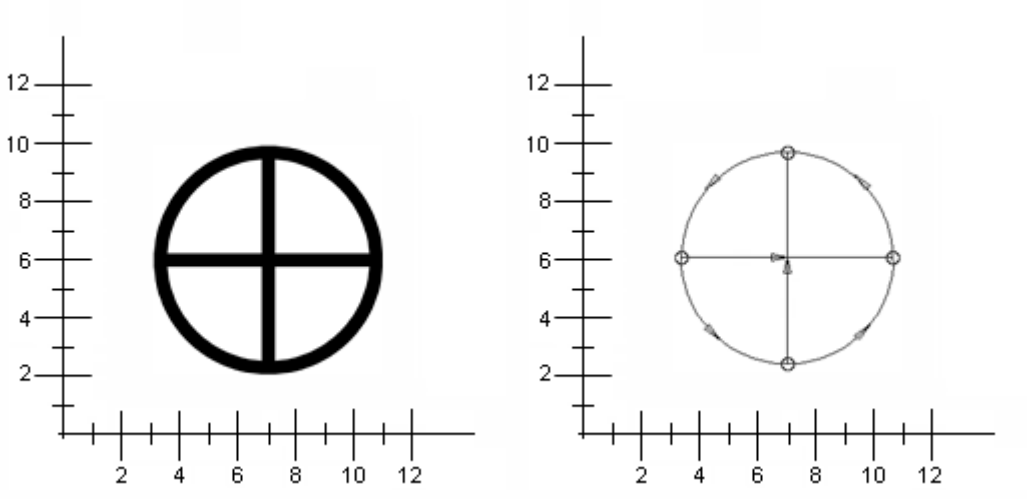
5.3.1.2 Image



5.3.1.3 Example

Syntax	Comments
G74* G54D10* G01X1100Y600D02* G03X700Y1000I400J0D01* X300Y600I0J400* X700Y200I400J0* X1100Y600I0J400* G01X300D02* X1100D01* X700Y200D02* Y1000D01*	Single quadrant mode Use aperture D10 Start from 11,6 Quarter arc (radius 4) to 7,10 Quarter arc (radius 4) to 3,6 Quarter arc (radius 4) to 7,2 Quarter arc (radius 4) to 11,6 Start from 3,6 Line to 11,6 Start from 7,2 Line to 7,10

Resulting Image



5.3.2 Multi Quadrant Mode

The I and J offsets are signed. If no sign is present, the offset is positive.

5.3.2.1 Data Block Format

The syntax in **Multi Quadrant Mode** is:

Gn Xsn Ysn Isn Jsn Dn

Syntax	Comments
Gn	Specifies direction of arc (see also table 5) n = 02 => clockwise n = 03 => counter clockwise
Xsn	s = optional sign (+ or -) n = X coordinate of arc endpoint

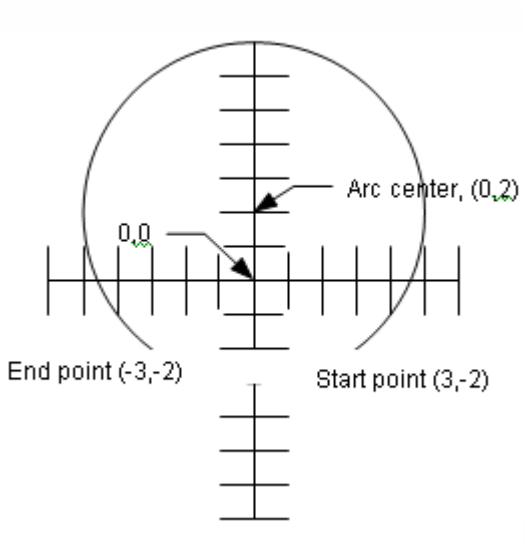
Ysn	s = optional sign (+ or -) n = Y coordinate of arc endpoint
Isn	s = optional sign (+ or -) n = Defines the incremental distance between the arc start point and the center measured parallel to the X axis.
Jsn	s = optional sign (+ or -) n = Defines the incremental distance between the arc start point and the center measured parallel to the Y axis.
Dn	Exposure (see table 4) n = 01 => On n = 02 => Off

5.3.3 Example

Syntax	Comments

`G75*`
`G01X300Y-200D02*`
`G03X-300Y-200I-300J400D01`
`G01*`
	`Multi quadrant mode`
	`Start from 3,-2`
	`Arc ccw (rel. center -3,4) to -3,-2`
	`Back to linear interpolation mode`

Resulting Image



5.4 Outline Fill (G36, G37)

G36 and G37 created filled areas by simply defining its closed outline.

G36 turns on outline fill, G37 turn it off. There are no variables or apertures. Following a G36 and before G37, all lines drawn with D01 are considered edges of the outline. D02 closes and fills the outline.

Self-intersecting outlines are not allowed because their interpretation is not obvious. Outline edges can coincide, allowing cut-ins to create holes in solid areas. Care must be taken that rounding errors, especially with arcs, does not turn a proper outline into a self-intersecting one, with unpredictable results; construct outlines defensively.

Outline area fill commands in the LPC section are reverse (not transparent).

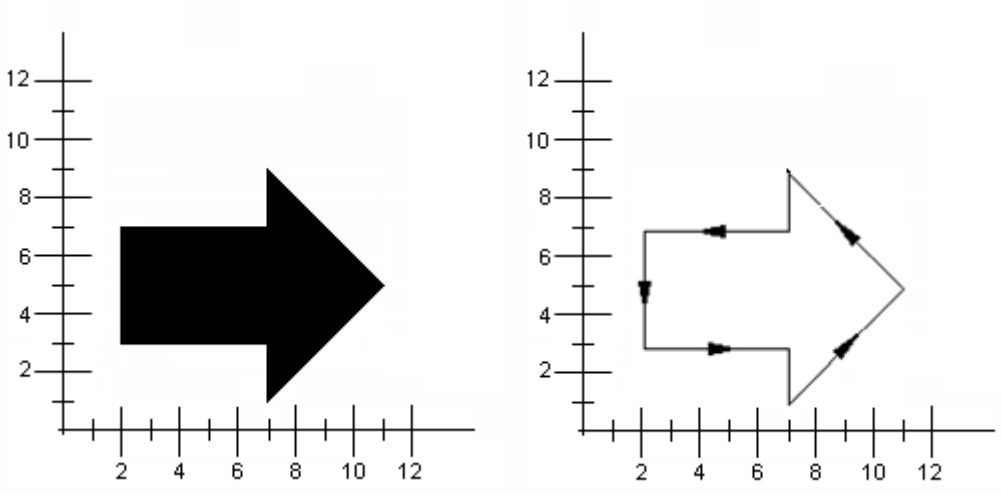
Outline area fill is much more efficient than the obsolete method of stroke fill.

(In previous versions of this document “outline fill” was also called “polygon fill”.)

5.4.1 Example 1: Simple outline

Syntax	Comments
G36* X200Y300D02* X700D01* Y100D01* X1100Y500D01* X700Y900D01* Y700D01* X200D01* Y300D01* G37*	Outline fill mode Start at 2,3 Line to 7,3 Line to 7,1 Line to 11,5 Line to 7,9 Line to 7,7 Line to 2,7 Line to 7,1 End of outline fill mode

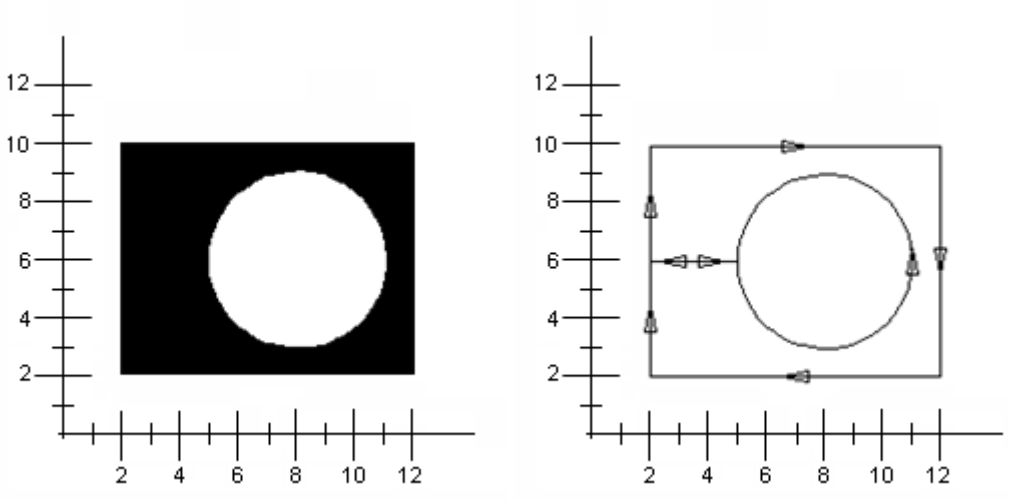
Resulting Image



5.4.2 Example 2: Cut In

Syntax	Comments
G36* X200Y1000D02* X1200D01* Y200* X200* Y600* X500* G75* G3X500Y600I0J300D01* G74* G1X200D01* Y1000* G37*	Outline fill mode Start at 2,10 Line to 12,10 Line to 12,2 Line to 2,2 Line to 2,6 Line to 5,6 Multi quadrant mode Full arc ccw (radius = 300) Single quadrant mode Line to 2,6 Line to 2,10 End of outline fill mode

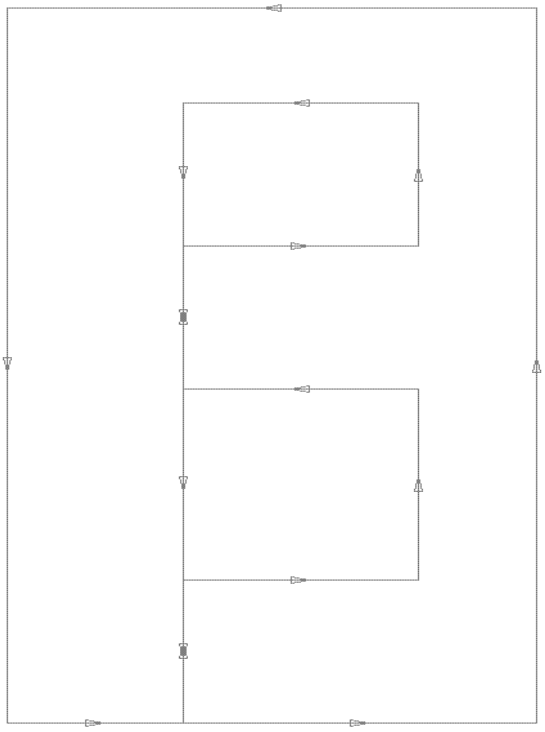
Resulting Image



5.4.3 Example 2: Cut In - bis

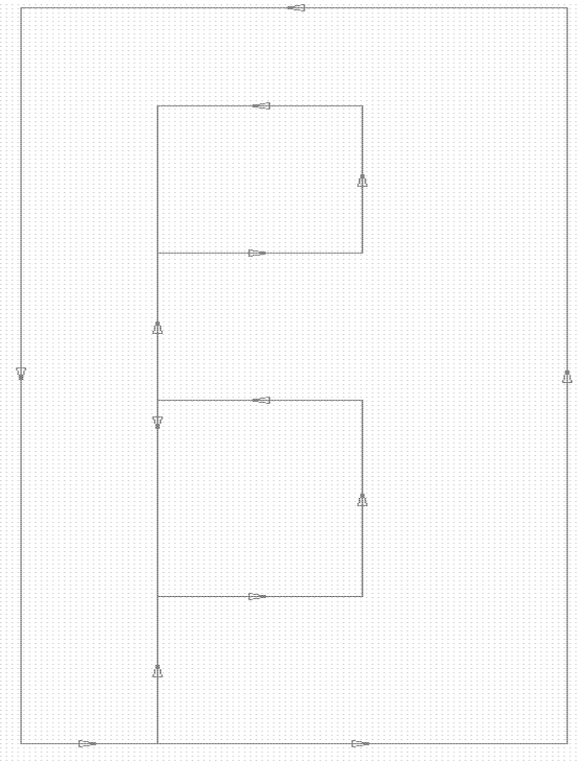
Cut-ins are susceptible to rounding problems. When the vertices move due to rounding, the contour may become self-intersecting, and therefore illegal. With a little bit of care this problem can be avoided. The construction below is defensive. Draws that are on top of one another have the *same* end vertices. When the vertices move under rounding, the draws will remain exactly on top of one another, and no self-intersections are created.

```
G36*
X1220000Y2570000D02*
X1250000D01*
Y2600000*
X1290000*
Y2640000*
X1250000*
Y2670000*
X1290000*
Y2700000*
X1250000*
Y2670000*
Y2640000*
Y2600000*
Y2570000*
X1310000*
Y2720000*
X1220000*
Y2570000*
G37*
```



The example below creates the same image, but with a less robust construction. The number of draws has been reduced by eliminating vertices between collinear draws. When the vertices move slightly due to rounding, the draws that were on top of one another may become intersecting, with unpredictable results. When a RS-274X file moves from system to system, numerical rounding must be expected. Therefore this construction is bad practice.

```
G36*
X1110000Y2570000D02*
Y2600000D01*
X1140000*
Y2640000*
X1110000*
Y2670000*
X1140000*
Y2700000*
X1110000*
Y2570000*
X1170000*
Y2720000*
X1090000*
Y2570000*
X1110000*
G37*
```



6 RS-274X Best Practices and Common Errors

Some RS-274X files produce the desired image but are needlessly cumbersome to work with or error-prone.
Some common poor practices:

Poor Practice	Problems	Best Practice
Low numerical precision.	Poor registration of objects between PCB layers. Rounding when writing the file can results in outlines to self-intersect, invalid arcs, zero-arcs, with unexpected result downstream.. Software processing the file will unavoidably add further numerical rounding, aggravating the problem.	**Always use high numerical precision.** Do not sacrifice precision to save a few bytes. Do not throw away the precision when writing the file
G74 and G75 not specified	Some Gerber readers use the wrong default. If you specify explicitly there can be no misunderstanding.	Always specify G74 or G75
G75 or multi quadrant mode.	The begin- and end-point of an arc will move somewhat due to rounding. For a very small are they can happen to move on top of another. Under G75 mode the small arc suddenly becomes a full circle. Under G74 it remains small, actually zero size.	Use G74 single quadrant mode unless you are very careful with rounding on small arcs.
Long file with lots of identical X and Y coordinate values.	File needlessly long	Modal usage of coordinates within the same Layer-specific parameter.
Painted or stroked pads.	Painted pads produce the correct image but are very awkward and time consuming for CAM software, e.g. for DRC checks, electrical test and so on. Stroking was needed for vector photoplotters in the 1960's and1970's, but these devices are as outdated as the mechanical typewriter Using them today is like sending your file on paper tape.	**Never use stroked pads**. Define pads, including SMD pads, with the AD and AM parameters.
Painted or stroked areas	Painted areas produce the correct image, but the files are needlessly large and the data is very confusing for CAM software. Stroking was needed for vector photoplotters in the 1960's and1970's, but these devices are as outdated as the mechanical typewriter.	**Never use stroked areas.** Use the Outline Fill commands;

Some RS-274X files do not produce the desired image due to incorrect interpretation of the specifications.
Some common errors:

Symptom	Cause	Best Practice
Polygons are smaller than expected	The inside dimension of a Regular Polygon are used (e.g. for an octagon: distance between opposite sides) instead of the outside dimension	
If exposure = “off”, previously defined underlying objects are displayed through holes in standard apertures.	Wrong aperture definition order	Objects that use macro definitions should be used before all other objects that are going to be used at the same location.
Outline Fill defined in a Layer Polarity clear section hides a previously defined object on that location	Outline Fill commands are using the current active Layer Polarity (LP)	Use Outline Fill commands with negative polarity before all other objects used at the same location.
A Mirror command is used to mirror a Macro definition but the result is not as expected.	The Mirror command is not applied to aperture definitions.	Don’t use the Mirror command. Apply the transfor-mation directly in the aperture definition and object coordinates.
A Rotation command is used to rotate a primitive but the result is not as expected.	The Rotation command rotates the primitive around the center point of the macro definition.	Don’t use the Rotation command on primitives for which the center point is not in the center of the macro definition. Apply the transformation directly in the primitive definition
Negative image only visible if a positive background image is used.	Gerber files specified with negative Image Polarity (IP) are not influencing the specified Layer Polarity (LP) value.	Specify polarity by using Layer Polarity (LP) command instead of Image Polarity (IP).

7 Glossary

ABSOLUTE POSITION: Position expressed in Cartesian coordinates from the *origin* 0,0.

APERTURE: A shape that is used for drawing lines or flashing. (The name is historic; one used to expose shapes on film by shining light through an *aperture* in an aperture wheel.)

APERTURE MACRO: A parameter describing the geometry of a special aperture and assigns it to a D-code.

APERTURE PARAMETER: A parameter (AD or AM) that assigns an aperture description to a D-code.

CIRCULAR INTERPOLATION: Drawing a circular arc.

COMPOSITE IMAGE: The entire image, including all information layers.

DIRECTIVE PARAMETER: A parameter that controls overall file processing.

IMAGE PARAMETER: A parameter that supplies information about an entire image.

KNOCKOUT: A rectangular region about an information layer whose polarity is the opposite of the layer polarity.

LAYER: A named information component of Gerber data that may be treated as a unit, for example, rotated or repeated; has no relationship to a physical PCB layer.

LAYER-SPECIFIC PARAMETER: A parameter that applies to a single information layer (for example KO, LN, LP, and SR).

LINEAR INTERPOLATION: Drawing straight lines.

PARAMETERS: Commands that specify how the data should be processed.

MULTI QUADRANT MODE: The arc is allowed extend over more than 90°. If the start point of the arc is equal to the end point, the arc is a full circle of 360°.

NEGATIVE: An artwork in which the intended conductive pattern is transparent to light and the areas to be free from conductive material are opaque.

NUMERICAL PRECISION: The number of integer and decimal places used to express a number.

POLARITY: Describes whether the circuitry will be imaged as dark on a clear background (positive) or clear on a dark background (negative). Polarity may be applied to an entire image or to a single layer.

OUTLINE FILL: A feature to create filled (solid) areas.

INCREMENTAL POSITION: Position expressed as a distance in X and Y from the current position.

SINGLE QUADRANT MODE: The arc cannot extend over more than 90°. If the start point of the arc is equal to the end point, the arc has length zero, i.e. covers 0°.

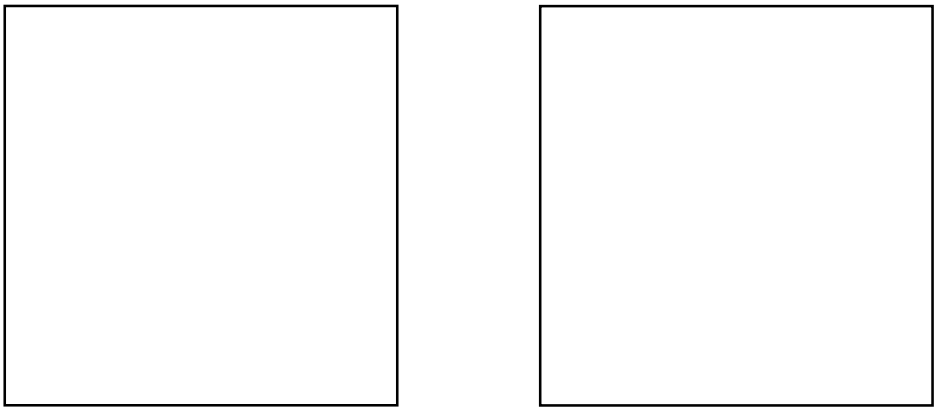
STEP AND REPEAT: A method by which successive exposures of a single image are made to produce a multiple image production master.

8 Sample Files

The examples on these pages illustrate the use of both Parameters and Function codes.

8.1.1 Example 1

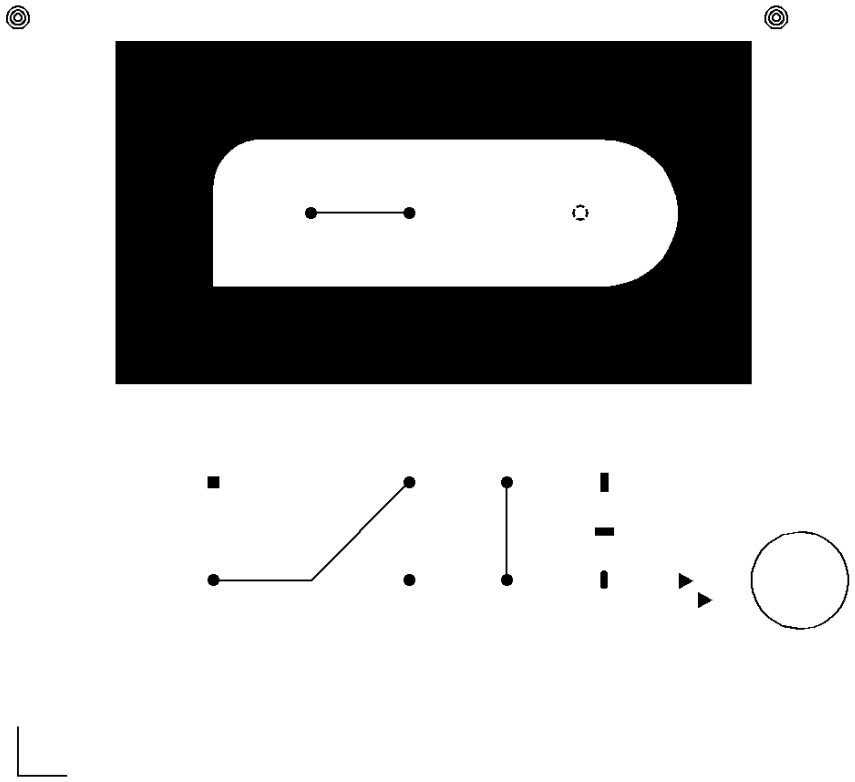
Example 1 illustrates a single layer image with 2 square boxes.



G04 EXAMPLE 1: 2 BOXES*	
%FSLAX23Y23*%	Format statement - leading zeroes omitted, absolute coordinates, X2.3, Y2.3.
%MOIN*%	Set units to inches.
%OFA0B0*%	No offset
%SFA1.0B1.0*%	Scale factor is A1, B1
%ADD10C,0.010*%	Define aperture with D-code 10 as a 10 mil circle
%LNBOXES*%	Name layer “BOXES”.
G54D10*	
X0Y0D02*	Start from 0,0
X5000Y0D01*	Line to 5,0
Y5000*	Line to 5,5
X0*	Line to 0,5
Y0*	Line to 0,0
X6000D02*	Start from 6,0
X11000D01*	Line to 11,0
Y5000*	Line to 11,5
X6000*	Line to 6,5
Y0*	Line to 6,0
M02*	End of program

8.1.2 Example 2

Example 2 illustrates RS-274X image showing various shapes.



%ASAXBY*	Axis Select, A=X, B=Y
FSLAX23Y23*	**Format Statement, Leading zeros omitted, absolute data, 2 integer digits and 3 fractional digits**
MIA0B0*	**Mirror about the specified axis; 0=no, 1=yes**
MOIN*	**Mode inches**
OFA0B0*	**Offset 0**
SFA1.0B1.0*%	**Scale Factor**
%IJALBL*	**Image Justify**
INXTEST*	**Image Name**
IOA0B0*	**Image Offset**
IPPOS*	**Image Polarity**
IR0*%	**Image Rotation**
G04 Define Apertures*	**Comment**
%AMTARGET125*	**Aperture Macro**
6,0,0,0.125,.01,0.01,3,0.003,0.150,0*%	**Moire description**
%AMTHERMAL80*	**Aperture Macro**

7,0,0,0.080,0.055,0.0125,45*%	Thermal description
%ADD10C,0.01*	Aperture description, D10 is a circular aperture with 0.01” diameter
ADD11C,0.06*	Aperture Description, D11 is a circular aperture with 0.06” diameter
ADD12R,0.06X0.06*	Aperture Description, D12 is a rectangular aperture, 0.06” X 0.06”
ADD13R,0.04X0.100*	Aperture Description, D13 is a rectangular aperture, 0.04” X 0.100”
ADD14R,0.100X0.04*	Aperture Description, D14 is a rectangular aperture, 0.100” X 0.04”
ADD15O,0.04X0.100*	Aperture Description, D15 is a obround aperture, 0.04” X 0.100”
ADD16P,0.100X3*	Aperture Description, D16 is a 3 sided polygon 0.100” overall size
ADD17P,0.100X3*	Aperture Description, D17 is a 3 sided polygon 0.100” overall size
ADD18TARGET125*	Aperture Description, D18 is a special aperture called “TARGET125”
ADD19THERMAL80*%	Aperture Description, D19 is a special aperture called "THERMAL80”
%LNXTEST1*	Layer Name XTEST1
LPD*	Layer Polarity Dark
SRX1Y1I0J0*%	Step and Repeat set to 1 X 1 (Not Required)
G54D10*	Aperture select
G01X0Y250D02*	Linear move with light off
X0Y0D01*	Linear move with light on
X250Y0D01*	Linear move with light on
X1000Y1000D02*	Linear move with light off
X1500D01*	Linear move with light on
X2000Y1500*	Notice since D01 is modal it does not need to be repeated
X2500D02*	Notice since the X & Y commands are modal, Y is not repeated
Y1000D01*	X is not repeated and uses its previous value of 2.500”
D02*	Light off no move
G54D11*	New aperture selected
G55X1000Y1000D03*	G55 prepares for flash It is not necessary. D03 is the flash command.
X2000D03*	Y value does not change
X2500D03*	This method reduces the size of the file
Y1500D03*	Here, X does not change from previous value

X2000D03*	Flash
G54D12*	New aperture select
X1000Y1500D03*	Move to (1.0, 1.5) and flash
G54D13*	New aperture select
X3000Y1500D03*	Move and flash
G54D14*	New aperture select
Y1250D03*	Move and flash
G54D15*	New aperture select
Y1000D03*	Move and flash
G54D10*	New aperture select
G01X3750Y1000D02*	Linear move, light off. Start point of the following arc command
G75*	Sets multi quadrant mode
G03X3750Y1000I250J0D01*	Move from start point above to end point drawing a complete circle
G54D16*	New aperture select
G55X3400Y1000D03*	Flash
G54D17*	New aperture select
G55X3500Y900D03*	Flash
G54D10*	New aperture select
G36*	Start Outline fill
G01X500Y2000D02*	
Y3750D01*	
X3750*	
Y2000*	
X500*	
X500Y2000D02*	
G37*	End Outline fill
G54D18*	New aperture select
G55X0Y3875D03*	Flash
X3875Y3875D03*	Flash
%LNXTEST2* LPC*%	Layer Name
G36*	Start Outline fill
G01X1000Y2500D02*	
Y3000D01*	
G74*	Single Quadrant mode
G02X1250Y3250I250J0D01*	Clockwise arc move with radius .25”
G01X3000*	Complete 90° arc
G75*	Sets multi quadrant mode
G02X3000Y2500I0J-375D01*	Clockwise arc move with radius .375”

G01X1000*	Linear move light on
X1000Y2500D02*	Linear move light off
G37*	End Outline fill
%LNXTEST3*	Layer Name
LPD*%	Layer Polarity Dark
G54D10*	New aperture select
X1500Y2875D02*	
X2000D01*	
D02*	
G54D11*	
X1500Y2875D03*	
X2000D03*	
G54D19*	New aperture select
X2875Y2875D03*	
M02*	End of Program