

# Encapsulated PostScript (EPS) Writer

## FORMAT NOTES:

This format is not supported by FME Base Edition.

The Encapsulated PostScript® (EPS) Writer module allows the Feature Manipulation Engine (FME) to write Encapsulated PostScript export files.

EPS is typically used for high-quality plots in desktop publishing software.

## EPS Quick Facts

Format Type Identifier	EPS
Reader/Writer	Writer
Licensing Level	Professional
Dependencies	None
Dataset Type	File
Feature Type	Not used
Typical File Extensions	.eps
Automated Translation Support	Yes
User-Defined Attributes	No
Coordinate System Support	No
Generic Color Support	Yes
Spatial Index	Not applicable
Schema Required	Yes
Transaction Support	No
Geometry Type	eps_type

Geometry Support			
Geometry	Supported?	Geometry	Supported?
aggregate	no	point	yes
circles	yes	polygon	yes
circular arc	yes	raster	no
donut polygon	yes	solid	no
elliptical arc	yes	surface	no
ellipses	yes	text	yes
line	yes	z values	no
none	no		

## Overview

EPS is a two-dimensional (2D) system with no provision for storing user-defined attributes for the geometric data.

All EPS information is contained within one page, beginning with a version header as well as a bounding box definition. EPS is based upon the PostScript format which provides methods for graphical drawing, simple programming control structures and the ability to create user-defined variables and functions.

All EPS data is contained in a single file with an `.eps` extension.

Filename	Extension	Contents
<code>.</code>	<code>eps</code>	All vector geometric data.

The EPS writer supports export of lines, polygons, arcs, ellipses (ellipse/circle), and text geometric data.

Some geometric entities may have display properties such as pen and brush width, type, pattern, and color. Color may be specified in red/green/blue (RGB) as well as cyan/magenta/yellow/black (CMYK).

## Writer Overview

The EPS writer creates and writes feature data to an EPS file specified by the `DATASET` directive. The writer searches the mapping file for the `<WriterKeyword>_DATASET` directive in the mapping file. This directive is required to be in the mapping file. An old EPS file in the directory with the same file name is overwritten with the new feature data. A typical mapping file fragment specifying the output EPS file looks like:

```
EPS_DATASET /usr/data/eps/myfile.eps
```

## Writer Directives

The suffixes shown are prefixed by the current `<ReaderKeyword>` in a mapping file. By default, the `<ReaderKeyword>` for the EPS writer is `EPS`.

### DATASET

**Required/Optional:** *Required*

The EPS writer processes the `DATASET` directive as described in *Writer Overview*. In addition, the writer scans the mapping file for `<WriterKeyword>_RESOLUTION _X` and `<WriterKeyword>_RESOLUTION _Y` directives. Both of these are optional directives and they define the bounding box of the EPS output file. The bounding box extends from the lower left corner of the page (defined as 0,0) and extends out to the values entered. By default, the X value is set to 612 and the Y value is set to 792. These values map onto an 8.5- by 11-inch piece of paper.

**Workbench Parameter:** [`<WorkbenchParameter>`](#)

### FORCE\_CMYK

**Required/Optional:** *Optional*

This directive specifies whether or not to force all output colors to be in CMYK format and defined as such in the EPS file. By setting the value following this keyword to `YES`, then all color usage output to the EPS file is done in CMYK. By default, this value is `NO`, meaning that a mix of RGB and CMYK color schemes may be in the output EPS file. However, despite forcing CMYK color output, some EPS viewers may not support the `setcmykcolor` call in their library. In these cases, the actual output of colors is done using a function we define in PostScript which interfaces exactly like the `setcmykcolor` call but uses `setrgbcolor` underneath. This will depend on the EPS viewer you are using.

**Value:** `YES` | `NO`

**Default Value:** `NO`

**Workbench Parameter:** [<WorkbenchParameter>](#)

### **LINE\_JOIN\_TYPE**

**Required/Optional:** *Optional*

This directive specifies the default corner types to be drawn onto paths. The values specify the default shape to be put at corners of paths painted: 0 specifies a sharp corner, 1 specifies a rounded corner, and 2 specifies a butt-end corner.

**Value:** `0`, `1`, `2`

**Default Value:** `0`

**Workbench Parameter:** [<WorkbenchParameter>](#)

### **LINE\_WIDTH**

**Required/Optional:** *Optional*

This directive specifies the default line width used to draw lines. This is measured in EPS units.

**Value:** *float*  $\geq 0$

**Default Value:** `0.0` (the thinnest line that can be rendered at device resolution, i.e. 1 pixel wide)

**Workbench Parameter:** [<WorkbenchParameter>](#)

### **MAINTAIN\_ASPECT**

**Required/Optional:** *Optional*

This directive specifies whether or not the source map dimensions will be kept or stretched to fit to the output bounding box.. A `YES` indicates that the original map aspect will be maintained to fit within the destination-defined bounding box. This means that the entire destination bounding box defined may not be used. Alternatively, the value `NO` causes the original map to be stretched onto the destination bounding box defined.

**Value:** `YES` | `NO`

**Default Value:** `YES`

[Workbench Parameter: <WorkbenchParameter>](#)

## MAP\_BUFFER

**Required/Optional:** *Optional*

This directive specifies the percentage of buffer room between the border of the output EPS map within the specified bounding box. It should be followed by a percentage value in decimals (for example, 0.20 is 20%). This value is used to buffer the border of the outputted EPS map within the specified bounding box. This prevents the border of the output map from being precisely on the bounding box border. The default value is 0.05 (5%), which places a 2.5% buffer between each map border and the bounding box border. This creates a total 5% buffer in the x and y axes.

**Value:** *0...1*

**Default Value:** *0.05*

[Workbench Parameter: <WorkbenchParameter>](#)

## RESOLUTION\_X

This directive specifies the maximum EPS units (1 unit = 1/72 inch) for the x dimension of the output map.

**Value:** *Integer > 0*

**Default Value:** *612*

[Workbench Parameter: <WorkbenchParameter>](#)

## RESOLUTION\_Y

This directive specifies the maximum EPS units (1 unit = 1/72 inch) for the y dimension of the output map.

**Value:** *Integer > 0*

**Default Value:** *792*

[Workbench Parameter: <WorkbenchParameter>](#)

# Feature Representation

In addition to the generic FME feature attributes that FME Workbench adds to all features (see *About Feature Attributes* on page 7), this format adds the format-specific attributes described in this section.

EPS features consist of geometry but no user-defined attributes, although there are special attributes to hold the type of the geometric entity and its display parameters. The feature type of features written to EPS is ignored.

All EPS features contain a `eps_type` attribute, which identifies the geometric type. Each element type also has a color associated with it. Depending on the geometric type, the

feature contains additional attributes specific to the geometric type. These are described in subsequent sections.

Attribute Name	Contents
eps_type	The EPS geometric type of this entity. <b>Range:</b> eps_polyline  eps_area  eps_text  eps_ellipse  eps_arc <b>Default:</b> No default
eps_cmyk_color	This is a string that represents the color intensities of the element. It is formatted as cyan (C), magenta (M), yellow (Y) and black (K). This color attribute has highest priority. If present, it will be used in preference over eps_color and fme_color attributes. <b>Range:</b> String. (0..1, 0..1, 0..1, 0..1) <b>Default:</b> String (0,0,0, 1)
eps_color	This is a string that represents the color intensities of the element. It is formatted as red, green, blue intensities which range between 0..1 Note that if this attribute is not found, then fme_color will be used. <b>Range:</b> String. (0..1, 0..1, 0..1) <b>Default:</b> String (0,0,0)

## Arc

**eps\_type:** eps\_arc

The arc definition here handles arcs, including those with different primary and secondary axis values. EPS arc features are linear features used to specify elliptical arcs. As such, the feature definition for eps\_arc is similar to the ellipse definition with two additional angles to control the portion of the ellipse boundary drawn. EPS arcs also support rotation.

**Tip:** The function @Arc() can be used to convert an arc to a linestring. This is useful for storing Arcs in systems not supporting them directly.

In addition to the attributes below, arcs also make use of the pen attributes as defined for eps\_area since arcs can also have fills.

Attribute Name	Contents
eps_primary_axis	The length of the semi-major axis in ground units. (x-axis) <b>Range:</b> Any real number > 0 <b>Default:</b> No default
eps_secondary_axis	The length of the semi-minor axis in ground units. (y-axis) <b>Range:</b> Any real number > 0 <b>Default:</b> No default

Attribute Name	Contents
eps_start_angle	Refer to the @Arc (function) in the <i>FME Functions and Factories manual</i> for a detailed definition of start_angle. <b>Range:</b> 0.0..360.0 <b>Default:</b> 0
eps_sweep_angle	Refer to the @Arc (function) in the <i>FME Functions and Factories manual</i> for a detailed definition of sweep_angle. <b>Range:</b> 0.0..360.0 <b>Default:</b> No default
eps_rotation	The rotation of the major axis. The rotation is measured in degrees counter clockwise up from horizontal. <b>Range:</b> 360.0..360.0 <b>Default:</b> 0

## Areas

**eps\_type:** eps\_area

EPS polygon features specify area (polygonal) features. The areas that make up a single feature may or may not be disjoint, and may contain polygons that have holes. Each area has a pen style associated with it to control the color, line weight, line type, and brush pattern used when it's drawn. If the area contains holes then when the fill pattern is applied, the holes enclosed by the area will **not** be filled. If no pen style is defined for a polygon entity, the previous style is used.

The following table lists the special FME attribute names used to control the EPS polygon settings.

Attribute Name	Contents
eps_line_width	Defines the line width used to draw the polyline. By default, the line is drawn one pixel wide. <b>Range:</b> Float >= 0 <b>Default:</b> 0.0
eps_dash_on	The number of pixels to be used as the on part of the dashed line used to draw the feature. If eps_line_width is specified, then this value is multiplied by the size of the pen to determine the number of pixels. If both eps_dash_on and eps_dash_off are 0, then a solid line is used. <b>Range:</b> Integer > 0 <b>Default:</b> 0
eps_dash_off	The number of pixels to be used as the off part of the dashed line used to draw the feature. If eps_line_width is specified, then this value is multiplied by the size of the pen to determine the number of pixels. If both eps_dash_on and eps_dash_off are 0, then a solid line is used. <b>Range:</b> Integer > 0 <b>Default:</b> 0

Attribute Name	Contents
<code>eps_line_join_type</code>	Specify the type of corner that should be drawn onto this path. 0 = sharp corners, 1 = rounded corners, 2 = butt-end corners <b>Range:</b> 0, 1, 2 <b>Default:</b> 0 <b>Optional:</b> Yes
<code>eps_cmyk_fill_color</code>	This is a string that represents the fill color intensities of the element. It is formatted as cyan (C), magenta (M), yellow (Y) and black (K), This color attribute has highest priority. If present, it will be used in preference over <code>eps_fill_color</code> and <code>fme_fill_color</code> attributes. <b>Range:</b> String. (0..1, 0..1, 0..1, 0..1) <b>Default:</b> String (0,0,0,1)
<code>eps_fill_color</code>	This is a string that represents the color intensities of the element. It is formatted as red, green, blue intensities which range between 0..1. If this attribute is not found, then the writer will refer to <code>fme_fill_color</code> . <b>Range:</b> String. (0..1, 0..1, 0..1) <b>Default:</b> None

## Ellipse

**eps\_type:** `eps_ellipse`

The `eps_ellipse` features are point features, and have only a single coordinate. This point serves as the centre of the ellipse. Additional attributes specify the primary axis (X) and secondary axis (Y) of the ellipse. EPS ellipses also support rotation.

Secondary  
Axis

---

**Tip:** The primary ellipse axis is not necessarily the longest axis, but rather the one on the x axis.

---

From the EPS ellipse, we also can arrive at circles (since they are just ellipses with both primary and secondary axes being equal).

In addition to the attributes below, ellipses also make use of the brush and pen attributes as defined by `eps_area`.

Attribute Name	Contents
<code>eps_primary_axis</code>	The length of the semi-major axis in ground units. (x-axis) <b>Range:</b> Any real number > 0 <b>Default:</b> No default
<code>eps_secondary_axis</code>	The length of the semi-minor axis in ground units. (y-axis) <b>Range:</b> Any real number > 0 <b>Default:</b> No default
<code>eps_rotation</code>	The rotation of the major axis. The rotation is measured in degrees counterclockwise up from horizontal. <b>Range:</b> -360.0..360.0 <b>Default:</b> 0

## Polylines

**eps\_type:** `eps_polyline`

EPS polyline features specify linear features defined by a sequence of x and y coordinates. Polylines encapsulate the concept of a line since a line is just a sequence of two points. Each polyline has a pen style associated with it that specifies the color, line weight, and line type used when the line is drawn. If no pen type is defined for a polyline entity, if line attributes aren't found, then default parameters are used.

The table below lists the special FME attribute names used to control the EPS polyline settings.

Attribute Name	Contents
<code>eps_line_width</code>	Defines the line width used to draw the polyline. By default, the line is drawn one pixel wide. <b>Range:</b> Float >= 0 <b>Default:</b> 0.0
<code>eps_dash_on</code>	The number of pixels to be used as the <code>on</code> part of the dashed line used to draw the feature. If <code>eps_line_width</code> is specified, then this value is multiplied by the size of the pen to determine the number of pixels. If both <code>eps_dash_on</code> and <code>eps_dash_off</code> are 0, then a solid line is used. <b>Range:</b> Integer > 0 <b>Default:</b> 0
<code>eps_dash_off</code>	The number of pixels to be used as the <code>off</code> part of the dashed line used to draw the feature. If <code>eps_line_width</code> is specified, then this value is multiplied by the size of the pen to determine the number of pixels. If both <code>eps_dash_on</code> and <code>eps_dash_off</code> are 0, then a solid line is used. <b>Range:</b> Integer > 0 <b>Default:</b> 0

Attribute Name	Contents
eps_line_join_type	Specify the type of corner that should be drawn onto this path. 0 = sharp corners, 1 = rounded corners, 2 = butt-end corners <b>Range:</b> 0, 1, 2 <b>Default:</b> 0 <b>Optional:</b> Yes

## Text

**eps\_type:** eps\_text

EPS text is used for text annotation in EPS. The coordinates specify the lower left coordinates of the text when it is placed. In addition, the size and angle in which the text is output can be specified.

The table below lists the special FME attribute names used to control the EPS text:

Attribute Name	Contents
eps_size	The size of the text specified in ground units <b>Range:</b> float > 0 <b>Default:</b> 0
eps_rotation	The text rotation is given in degrees and measured counterclockwise up from the horizontal. <b>Range:</b> -360..360 <b>Default:</b> 0
eps_font	The PostScript name of the font. The fonts supported depend on the destination of the EPS file. Some typical fonts are Times, Helvetica and Courier. <b>Range:</b> String <b>Default:</b> Times
eps_style	The style of the font. This attribute must be matched with the current font since it's the combination of font and style that EPS recognizes. Some typical fonts and styles are Times-(Roman, Italic, Bold, BoldItalic), Helvetica-(Oblique, Bold, BoldOblique) and Courier-(Oblique, Bold, BoldOblique) <b>Range:</b> String <b>Default:</b> Roman
eps_text_string	The text to be displayed. <b>Range:</b> String <b>Default:</b> No default

## Point

**eps\_type:** eps\_point

EPS point is used for points in EPS.

The table below lists the special FME attribute names used to control the EPS point:

<b>Attribute Name</b>	<b>Contents</b>
eps_size	The size of the point specified in ground units <b>Range:</b> float > 0 <b>Default:</b> 0
eps_rotation	The point rotation is given in degrees and measured counterclockwise up from the horizontal. <b>Range:</b> -360..360 <b>Default:</b> 0
eps_font	The PostScript name of the font. The fonts supported depend on the destination of the EPS file. Some typical fonts are Times, Helvetica and Courier. <b>Range:</b> String <b>Default:</b> Times
eps_style	The style of the font. This attribute must be matched with the current font since it's the combination of font and style that EPS recognizes. Some typical fonts and styles are Times-(Roman, Italic, Bold, BoldItalic), Helvetica-(Oblique, Bold, BoldOblique) and Courier-(Oblique, Bold, BoldOblique) <b>Range:</b> String <b>Default:</b> Roman
eps_symbol_string	The text to be displayed. <b>Range:</b> String <b>Default:</b> "."