

Adobe Illustrator (IEPS) Writer

The Adobe Illustrator Encapsulated PostScript® (IEPS) Writer module allows FME to write Encapsulated PostScript export files specifically formatted to work with Adobe Illustrator. Illustrator IEPS is a different flavour of EPS and makes use of some of the functionality of Adobe Illustrator. The most significant additions are the use of layers and object attributes. In this format, many of the PostScript keywords have been shortened into special Adobe Illustrator single letter functions. The implication is that EPS files produced by this writer cannot be used outside of Adobe Illustrator. The standard EPS writer should be used if the EPS is to be used in other applications.

IEPS is most often used for high-quality plots in desktop publishing software.

Note: This writer may write files that are quite large since it does create an output coordinate for every source coordinate. If you find your .eps files getting too large, it is recommended that you first generalize your source data to make it less dense using the FME's @Generalize function (or the FME Workbench Generalizer transformer).

IEPS Quick Facts

Format Type Identifier	IEPS
Reader/Writer	Writer
Licensing Level	Base
Dependencies	None
Dataset Type	File
Feature Type	Layer name
Typical File Extensions	.eps
Automated Translation Support	Yes
User-Defined Attributes	Yes
Coordinate System Support	No
Generic Color Support	Yes
Spatial Index	Not applicable
Schema Required	Yes
Transaction Support	No
Geometry Type Attribute	ieps_type

Geometry Support			
Geometry	Supported?	Geometry	Supported?
aggregate	no	point	yes
circles	no	polygon	yes
circular arc	no	raster	no
donut polygon	yes	solid	no
elliptical arc	no	surface	no
ellipses	no	text	no

Geometry Support			
Geometry	Supported?	Geometry	Supported?
line	yes	z values	no
none	no		no

Overview

IEPS is a two-dimensional (2D) format with the ability to store user-defined attributes for the geometric data.

All IEPS information is contained in a single page beginning with a version header as well as a bounding box definition. IEPS 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 IEPS data is contained in a single file with an `.ieps` extension.

File Name Extension	Contents
<code>.ieps</code>	All vector geometric data.

The IEPS writer supports export of points, lines, polygons, and text geometric data.

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

Writer Overview

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

```
IEPS_DATASET /usr/data/ieps/myfile.ieps
```

Writer Directives

The directives processed by the IEPS writer are listed below. The suffixes shown are prefixed by the current `<WriterKeyword>` in a mapping file. By default, the `<WriterKeyword>` for the IEPS writer is `IEPS`.

DATASET

Required/Optional: *Required*

The IEPS writer processes the `DATASET` keyword as described in *Writer Overview*. Additional keywords can be used to set default parameters that are applied to all applicable features in the file. However, the values set by the keywords can be overwritten if the feature itself has a value defined for that parameter. For example, although the `LINE_WIDTH` keyword may be used to specify a default width of 5 for all lines in the file,

if an `ieps_polyline` feature has its `ieps_line_width` set to a value of 2, then the line width of 2 will be used over the default value of 5.

DEF

Required/Optional: *Required*

This is a required keyword that defines the layers within the file. DEF lines also list the attributes that will be saved as object tags on features of that layer, and may also include the attribute `IEPS_LAYER_COLOR`. This should be followed by an RGB combination ranging in intensities from 0 to 255, separated by commas. This defines the layer color seen in Adobe Illustrator.

Attribute	Contents	Required/Optional
<code>IEPS_LAYER_COLOR</code>	This is an attribute that can be used on a DEF line. It defines the layer color seen in Illustrator. Range: 0..255, 0..255, 0..255 Default: No Default	Optional

RESOLUTION _X and RESOLUTION _Y

Required/Optional: *Optional*

These directives define the bounding box of the IEPS 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 x 11-inch piece of paper.

Range: *Integer > 0*

Default:

- *RESOLUTION_X: 612*
- *RESOLUTION_Y: 792*

MAINTAIN_ASPECT

Required/Optional: *Optional*

This directive is followed by a value of YES or NO. By default, the value is set to YES. 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 defined destination bounding box.

Range: *YES / NO*

Default: *YES*

LINE_WIDTH

Required/Optional: *Optional*

This directive is followed by the value in pixels of the line width you wish to use by default. The default value is set to 0, which is the thinnest printable line width.

Range: *float* ≥ 0

Default: *0.0 (1 pixel wide: the thinnest line that can be rendered at device resolution)*

TEXT_WIDTH

Required/Optional: *Optional*

This directive has an attribute just like `LINE_WIDTH` except that this width is applied to text features. The default value is set to 0, which is the thinnest printable line width.

Range: *float* ≥ 0

Default: *0.0 (1 pixel wide: the thinnest line that can be rendered at device resolution)*

TEXT_FONT

Required/Optional: *Optional*

This directive specifies the default font applied to all text features. The font must be a PostScript name. The fonts supported depend on the destination of the IEPS file. Some typical fonts are NewBaskerville, Times, Helvetica and Courier. The default is NewBaskerville since it is the most commonly installed with Adobe Illustrator.

Range: *String*

Default: *NewBaskerVille*

TEXT_STYLE

Required/Optional: *Optional*

This directive specifies the default style to be applied to the text font all text features. This attribute must be matched to the current font since it is the combination of text font and text style that is recognized by Adobe Illustrator. Some typical font and style combinations are NewBaskerville-(None, Bold), Times-(None, Roman, Italic, Bold, BoldItalic), Helvetica-(None, Oblique, Bold, BoldOblique), and Courier-(None, Oblique, Bold, BoldItalic). Note that the keyword `NONE` can be used to specify that no style should be applied to the font.

Range: *String*

Default: *Bold*

LINE_JOIN_TYPE

Required/Optional: *Optional*

This directive is followed by the values 0, 1, or 2. These 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.

Range: *0, 1, 2*

Default: 0

LINE_CAP_TYPE

Required/Optional: *Optional*

This directive is followed by the values 0, 1, or 2. These values specify the default cap that will be used on line segments. 0 specifies butt-end caps, 1 specifies rounded-end caps and 2 specifies square-end caps.

Range: 0, 1, 2

Default: 0

FORCE_CMYK

Required/Optional: *Optional*

By setting the value following this keyword to YES, then all color usage output to the IEPS file is in CMYK. By default, this value is NO, meaning that a mix of RGB and CMYK color schemes may be in the output IEPS file. However, despite forcing CMYK color output, some IEPS 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 IEPS viewer you are using.

Range: YES / NO

Default: NO

LOCK_FEATURES

Required/Optional: *Optional*

If set to YES, by default all features will be locked and cannot be selected or edited in Adobe Illustrator. Note: Even if LOCK_FEATURES is set to YES, individual features can be unlocked if its `eps_lock_feature` is set to 0 (meaning NOT locked). Hence, an individual `eps_lock_feature` value overrides this LOCK_FEATURES default value.

Range: YES / NO

Default: **NORENDER_TYPE**

Required/Optional: *Optional*

This directive determines how the text is output. This value will be used as the default render type for all text in the file but it will be overridden if the text feature has its own user-defined render type value.

This directive is followed by the values 0, 1, or 2. These values specify the default rendering that will be applied to text features: 0 = fill, 1 = stroke, 2 = stroke and fill. The default value is 2.

Range: 0, 1, 2

Default: 2

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.

IEPS 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 a feature written to IEPS is used to specify its layer in the output Adobe Illustrator file.

All IEPS features contain a `ieps_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
<code>ieps_type</code>	The IEPS geometric type of this entity. Range: <code>ieps_polyline </code> <code>ieps_area </code> <code>ieps_text ieps_point</code> Default: No default
<code>ieps_cmyk_color</code>	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 <code>ieps_color</code> and <code>fme_color</code> attributes. Range: String. (0..1, 0..1, 0..1, 0..1) Default: String (0,0,0, 1)
<code>ieps_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>ieps_fill_color</code> and <code>fme_fill_color</code> attributes. Range: String. (0..1, 0..1, 0..1, 0..1) Default: String (0,0,0,1)
<code>ieps_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. Note that if this attribute is not found, then <code>fme_color</code> will be used. Range: String. (0..1, 0..1, 0..1) Default: String (0,0,0)
<code>ieps_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> . Range: String. (0..1, 0..1, 0..1) Default: None

Attribute Name	Contents
ieps_url	Allows you to attach a URL to a feature. The URL should be formatted as <i>http://www.safe.com</i> . Range: String Default: No Default
ieps_dash_on	The number of pixels to be used as the <i>on</i> part of the dashed line used to draw the feature. If <i>ieps_pen_linewidth</i> is specified, then this value is multiplied by the size of the pen to determine the number of pixels. If both <i>ieps_dash_on</i> and <i>ieps_dash_off</i> are 0, then a solid line is used. Range: Integer > 0 Default: 0
ieps_dash_off	The number of pixels to be used as the <i>off</i> part of the dashed line used to draw the feature. If <i>ieps_pen_linewidth</i> is specified, then this value is multiplied by the size of the pen to determine the number of pixels. If both <i>ieps_dash_on</i> and <i>ieps_dash_off</i> are 0, then a solid line is used. Range: Integer > 0 Default: 0
ieps_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 Range: 0, 1, 2 Default: 0 Optional: Yes
ieps_line_cap_type	Specify the type of caps on line ends. 0 = butt end caps, 1 = rounded end caps, 2 = square end caps Range: 0, 1, 2 Default: 0 Optional: Yes
ieps_locked_flag	This determines whether or not the feature can be selected for editing when the document is opened in Adobe Illustrator. If set to 0, the feature can be selected for editing. If set to 1, the feature is locked and cannot be selected. Range: 0, 1 Default: 0 Optional: Yes

Areas

ieps_type: ieps_area

IEPS 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 IEPS polygon settings.

Attribute Name	Contents
ieps_line_width	Defines the line width used to draw the polyline. By default, the line is drawn one pixel wide. Range: Float ≥ 0 Default: 0.0 (the thinnest line that can be rendered at device resolution, i.e. 1 pixel wide)

Polylines

ieps_type: ieps_polyline

IEPS 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 IEPS polyline settings.

Attribute Name	Contents
ieps_line_width	Defines the line width used to draw the polyline. By default, the line is drawn one pixel wide. Range: Float ≥ 0 Default: 0.0 (the thinnest line that can be rendered at device resolution, i.e. 1 pixel wide)

Text

ieps_type: ieps_text

IEPS text is used for text annotation in IEPS. 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 IEPS text:

Attribute Name	Contents
ieps_size	The size of the text specified in ground units Range: float > 0 Default: 0

Attribute Name	Contents
ieps_illustrator_size	The size of the point text specified in points. If this is set, it will override the <code>ieps_size</code> value. Range: float > 0 Default: 12pt
ieps_rotation	The text rotation is given in degrees and measured counterclockwise up from the horizontal. Range: -360..360 Default: 0
ieps_font	The PostScript name of the font. The fonts supported depend on the destination of the IEPS file. Some typical fonts are Times, Helvetica and Courier. Range: String Default: NewBaskerville
ieps_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 IEPS recognizes. Some typical fonts and styles are Times-(None, Roman, Italic, Bold, BoldItalic), Helvetica-(None, Oblique, Bold, BoldOblique) and Courier-(None, Oblique, Bold, BoldOblique). Note the keyword 'NONE' can be specified to indicate no style on the font. Range: String Default: Bold
ieps_text_string	The text to be displayed. Range: String Default: No default
ieps_text_width	Defines the line width used to stroke the text. By default, the stroked line is drawn one pixel wide. Range: Float >= 0 Default: 0.0 (the thinnest line that can be rendered at device resolution, i.e. 1 pixel wide)
ieps_render_type	This determines how the text is output. 0 = filled, 1 = stroked, 2 = stroked and filled Range: 0,1,2 Default: 2

Point

ieps_type: ieps_point

IEPS point is used for point annotation in IEPS. Points will be represented as text. By default, a symbol will be represented by a period.

Attribute Name	Contents
ieps_size	The size of the point text specified in ground units Range: float > 0 Default: 0

Attribute Name	Contents
ieps_illustrator_size	<p>The size of the point text specified in points. If this is set, it will override the <code>ieps_size</code> value.</p> <p>Range: float > 0 Default: 12pt</p>
ieps_rotation	<p>The text rotation is given in degrees and measured counterclockwise up from the horizontal.</p> <p>Range: -360..360 Default: 0</p>
ieps_font	<p>The PostScript name of the font. The fonts supported depend on the destination of the IEPS file. Some typical fonts are Times, Helvetica and Courier.</p> <p>Range: String Default: NewBaskerville</p>
ieps_style	<p>The style of the font. This attribute must be matched with the current font since it's the combination of font and style that IEPS recognizes. Some typical fonts and styles are Times-(None, Roman, Italic, Bold, BoldItalic), Helvetica-(None,Oblique, Bold, BoldOblique) and Courier-(None,Oblique, Bold, BoldOblique). Note the keyword 'NONE' can be specified to indicate no style on the font.</p> <p>Range: String Default: Bold</p>
ieps_symbol_string	<p>The text to be displayed.</p> <p>Range: String Default: "."</p>
ieps_symbol_width	<p>Defines the line width used to stroke the text. By default, the stroked line is drawn one pixel wide.</p> <p>Range: Float >= 0 Default: 0.0 (the thinnest line that can be rendered at device resolution, i.e. 1 pixel wide)</p>
ieps_render_type	<p>This determines how the text is output.</p> <p>0 = filled, 1 = stroked, 2 = stroked and filled</p> <p>Range: 0,1,2 Default: 2</p>