

Adobe Geospatial PDF Writer

Note: Licensing options for this format begin with FME Professional Edition.

The PDF2D Writer enables FME to write Adobe® Portable Document Format (PDF) with vector drawings and geospatial information.

Overview

PDF is a document exchange format created by Adobe Systems.

The PDF2D writer will write features with 2D geometry as vector drawings or raster features as images on a page of a PDF document. The output PDF file can be viewed with Adobe Acrobat Reader or any other PDF viewer application.

Features will belong to a layer according to its feature type. Feature attribute can be queried using the analysis tools of the Adobe Acrobat Reader software. If features have a coordinate system defined, then geospatial coordinates of the cursor location can also be displayed.

Adobe Geospatial PDF Quick Facts

About Quick Facts Tables

Format Type Identifier	PDF2D
Reader/Writer	Writer
Licensing Level	Professional
Dependencies	None
Dataset Type	File
Feature Type	Layer
Typical File Extensions	PDF
Automated Translation Support	Yes
User-Defined Attributes	Yes
Coordinate System Support	Yes
Generic Color Support	Yes
Spatial Index	Never
Schema Required	Yes
Transaction Support	Never
Enhanced Geometry	Yes
Geometry Type Attribute	pdf_type
Encoding Support	Yes

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

Band Interpretations	Red8, Green8, Blue8, Alpha8, Gray8, UInt8, Red16, Green16, Blue16, Alpha16, Gray16, UInt16, Int16
Palette Key Interpretations	not applicable
Palette Value Interpretations	not applicable
Nodata Value	not applicable
Cell Origin (x, y)	0.5, 0.5
Rotation Support	No
GCP Support	No
World File Support	No
TAB File Support	No

Writer Overview

The writer outputs PDF version 1.7 files. The document will have one page and features will be drawn in a rectangular region of the page called the viewport. Measurements on the page use the unit of a typographical point. Also known as a PostScript point, it is defined as 1/72 of an inch on the output page.

If attribution is written, then each feature and feature type will be represented by a logical structure element. In Adobe Acrobat Reader, features can be visually picked using the Object Data tool.

Features with unsupported geometry types will not be drawn, but their attribution data will still be written.

Features will be grouped into layers according to their feature types. In Adobe Acrobat Reader, the visibility of layers can be toggled.

PDF files can be opened through a command or an URL that specifies what and how the contents are displayed.

For more details about this feature, see this external documentation:

http://www.adobe.com/devnet/acrobat/pdfs/pdf_open_parameters.pdf

Adobe Geospatial PDF Writer Parameters

Page Size

Page Size

Specifies the size of the output page of the PDF document. The default value for this parameter is Letter. Preset page sizes for common pager sizes can be selected or the page size can be specified in typographical points in the format <width> <height>.

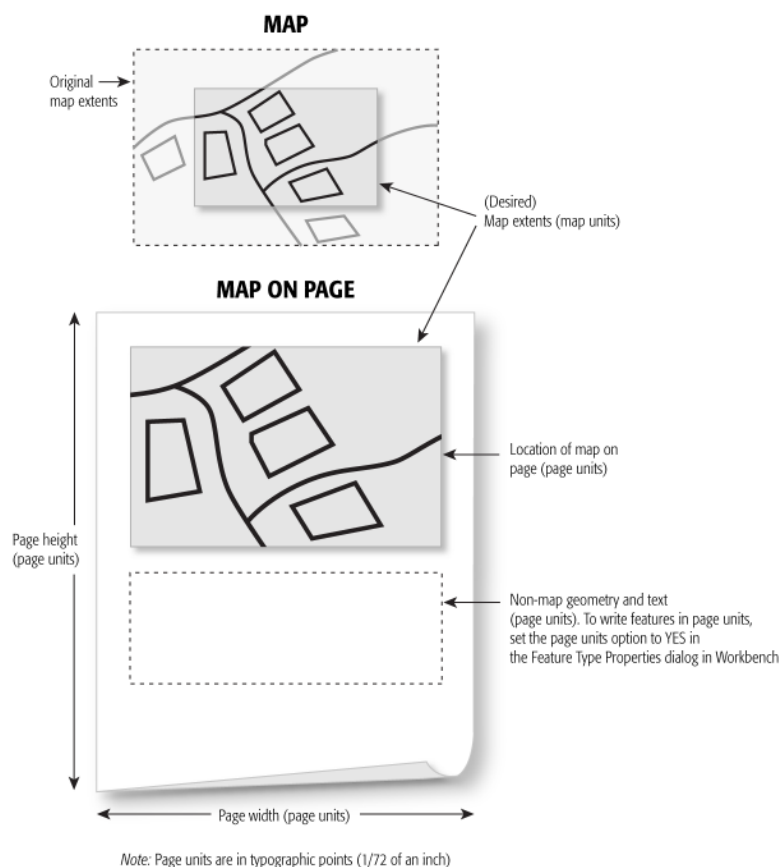
Location of Map on Page

This parameter determines where to place the map on the page, and how large the map should be on the page.

The format for the parameter is four integers separated by spaces describing the lower left corner and the upper right corner of the viewport/rectangle, specified in typographical points. The lower left corner of the page has coordinate (0,0) and the top right corner has coordinate (<width>, <height>) where these two values are the page size specified by the parameter Page Size.

If the aspect ratios of the page viewport and the world viewport differ, then the lesser scaling factor will be chosen: data inside the world viewport will not be clipped and data outside the world viewport might become visible.

If a value for the directive is not specified, then the page viewport rectangle will be a centered rectangle with a width and length that is 90% of the page width and length. The page viewport coordinates must be between (0,0) and (page width,page height).



Values: <lowerLeftX> <lowerLeftY> <upperRightX> <upperRightY>

Map Extents

Specifies a rectangular region of space, in world coordinates, that will be mapped to the page viewport. Geometry outside the world viewport will be clipped when drawn on the page. The format for this option is four floating point numbers separated by spaces describing the lower left corner and the upper right corner of the rectangle. If a value is not specified, then the world viewport rectangle will be the bounding box of the entire dataset.

Values: <lowerLeftX> <lowerLeftY> <upperRightX> <upperRightY>

Map Parameters

Default Fill Opacity

Specifies the opacity value of the fill color of area geometries. The boundaries of area geometries are not affected by this setting. A value of 0 corresponds to complete transparency and a value of 1 is complete opaqueness.

Default Point Radius

The default radius in pixels for point geometry.

Default Line Width

The default width in pixels for line geometry and boundaries of area geometry.

Navigation Panel to Display

Determines the panel that is visible immediately after opening the output PDF file in Adobe Acrobat software. If None is specified, then no panel will be initially displayed. If Layers is specified, then the Layer panel will be visible after opening the file. If Pages is specified, then the Page Thumbnails panel will be visible.

Randomize Feature Type Color

Specifies whether features without the fme_color attribute set will be assigned a random color based on its feature type. If this box is not checked, then features without their fme_color attributes set will be assigned the color black.

Text Parameters**Text in Rich Text Format**

Specifies whether the text string of text features is in the rich text format. If the box is not checked, then the text string is written as-is to the page. If the box is checked, then the text string will be processed for style directives.

TrueType Font Directories

Specifies the directories that the writer will search in to find the TrueType fonts used in the workspace. The workspace directory of the translation is always searched.

Attribution**Write Attributes**

Specifies whether attribution data will be written. Not writing attribution data will decrease the file size of the output file and may improve viewing performance.

Compression**Compress Streams**

Determines whether streams in the output file will be compressed.

Compatibility**PDF 1.4 Compatible**

This parameter specifies whether or not to write files compatible with PDF 1.4 viewers.

By default, this parameter is not selected. This means that features introduced in PDF 1.5 and later will be used, including object stream compression and the JPEG2000 raster image format.

If this parameter is selected, the output file will not contain compressed object streams, cross-reference information is stored in a cross-reference table and xref trailer, and rasters are encoded in the JPEG format.

Note: Note that the *Compress Object Streams* parameter is deprecated, and equivalent to the opposite of this directive.

Multi-page Support

The default behavior of the PDF writer is to write to a single frame on a single page.

A frame is a rectangle on a page where features are drawn. A feature belongs to one or more frames, and a frame belongs to a single page. There are two sets of format attributes for specifying frame and page properties.

Frame specification format attributes:

Attribute Name	Contents
pdf_frame_name	<p>If this attribute is set, it specifies the name of the frame to which the feature belongs.</p> <p>The other frame specification attributes will apply to this frame. See subsection "Named Frames" below for more information. If this attribute is unset, then an unnamed frame will be created. See Unnamed Frames below for more information.</p>
pdf_frame_order	<p>This attribute specifies the order in which the frames are drawn.</p> <p>Valid values are integers. Frames with higher frame order numbers are drawn above frames with lower order numbers. If this attribute is unset on all features belonging to a given frame, then it will assume the default value of '0'.</p>
pdf_frame_rectangle	<p>This attribute specifies the position of the frame on the page.</p> <p>The units are in page units, and the format is identical to the PAGE_VIEWPORT writer directive, including support for negative and percentage values. If this attribute is unset on all features belonging to a given frame, then it will assume the value of the PAGE_VIEWPORT directive.</p>
pdf_world_rectangle	<p>This attribute specifies the extents of the map that will be written to the frame.</p> <p>The units are in world units, and the format is identical to the WORLD_VIEWPORT writer directive. If this attribute is unset on all features belonging to a given frame, then it will assume the value of the WORLD_VIEWPORT directive.</p>
pdf_page_number	<p>This attribute specifies the page number to which the frame belongs.</p> <p>This attribute is also used to specify the page specifications for that page number. For unnamed frames, if this attribute is unset, then the attribute is set to the value '1'. If this attribute is unset on</p>

Attribute Name	Contents
	all features belonging to a given named frame, then it will assume the value of '1'.

Page specification format attributes:

Attribute Name	Contents
pdf_page_number	<p>This attribute specifies the page number of the page to which the page specifications will apply.</p> <p>For unnamed frames, if this attribute is unset, then the attribute is set to the value '1', and any page specification attributes that are set on this feature will apply to the first page. If this attribute is unset on all features belonging to a given named frame, then it will assume the value of '1'. For a given feature that does not set this attribute, any page specification attributes that are set on that feature are ignored.</p>
pdf_page_size	<p>This attribute specifies the size of the page.</p> <p>If this attribute is unset on all features belonging to a given page specification, then the page will assume the page size specified by the last prior page that has a page size specification. If no such page exists (e.g. on the first page) then the page size is set to the PAGE_SIZE writer directive.</p>

Unnamed Frames

If the `pdf_frame_name` attribute is not set, then the feature belongs to an unnamed frame. Multiple unknown frames may be created during a translation. Features are assigned to an unnamed frame by first assigning default values to unset frame specification attributes. Then features with identical frame specification attributes are assigned to the same unnamed frame.

Named Frames

A feature can belong to a named frame by setting the `pdf_frame_name` attribute. There are two rules for determining the frame specification of a named frame. Given any two features belonging to the same frame, if the same frame specification attribute is set on both features, then their values must be identical. If a specific frame specification attribute is unset on all features belonging to a frame at the end of the translation, then the frame will assume the default value for that attribute.

By setting the `pdf_no_output` format attribute to a non-empty value, frame and page specifications can be specified using dummy features without writing these features onto any page.

Page Specifications

Page specifications are handled similarly to named frames, except that pages are referenced by the `pdf_page_number` attribute. Given any two features belonging to the same page specification, if the same page specification attribute is set on both features, then their values must be identical. If a specific page specification attribute is unset on all features belonging to a page specification at the end of the translation, then the page will assume the default value for that attribute.

Feature Representation

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

FME raster features represent raster data and use several concepts that are unlike those used in the handling of vector data. See [About FME Rasters](#).

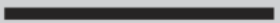



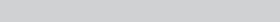
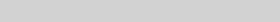
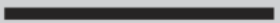



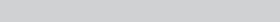
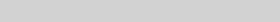
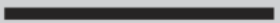



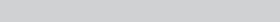
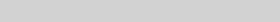
PDF files can be written with non-square pixel dimensions.

PDF supports rasters with an arbitrary number of bands, provided all bands are the same data type and no band has a palette. PDF also supports rasters with a single band that has a palette.

Attribute Name	Contents
pdf_type	<p>The value specifies PDF geometric type of this entity.</p> <p>Range:</p> <ul style="list-style-type: none"> pdf_area pdf_collection pdf_line pdf_point pdf_text <p>Default: No default</p>
pdf_name	<p>If attribution data is written, then the value of this attribute determines the name of the structure element associated with the feature. If this attribute is not set, then the structure element will be numbered sequentially. The names need not be unique.</p> <p>Range:</p> <ul style="list-style-type: none"> <string> <p>Default: <auto-generated integer></p>

Attribute Name	Contents
pdf_line_width	<p>The value specifies the line width in typographical points of line geometries and boundaries of area geometries. Points within half the line width perpendicular distance from the line path will be painted.</p> <p>Range: <float></p> <p>Default: 1.0</p>
pdf_line_cap_style	<p>The value specifies the cap style for the ends of lines.</p> <p>Range: <0,1,2></p> <p>0 - Butt cap: Lines are squared off at the end and do not project past the end of the line.</p> <p>1 - Round cap: Semicircles with diameter equal to the line width cap the ends of lines.</p> <p>2 - Projecting square cap: Lines project past the end by a distance equal to half the line width and are squared off.</p> <p>Default: 0</p>
pdf_line_join_style	<p>The value specifies the shape of corners between segments of paths.</p> <p>Range: <0,1,2></p> <p>0 - Miter join: Outer edges of segments are extended until they meet.</p> <p>1 - Round join: Arcs with diameter equal to the line width are drawn around corners.</p> <p>2 - Bevel join: Two adjacent segments are finished with butt caps, and the notch beyond the ends is filled with a triangle.</p> <p>Default: 0</p>
pdf_line_miter_limit	<p>For miter joins, the miter limit imposes a maximum on the ratio of the miter length to the line width. For example, a miter limit of 1.414 will bevel the ends of two segments meeting at an angle less than 90 degrees (the far corner will be at a distance $\sqrt{1^2+1^2}=\sqrt{2}$ from the line).</p>

Attribute Name	Contents
	<p>Range: <0.0...></p> <p>Default: 0.0</p>
pdf_line_dash_pattern{}	<p>The values in this list attribute specify the dash pattern for line geometries and the boundaries of area geometries. This attribute works together with <code>pdf_line_dash_pattern_phase</code> to establish a simple dashed line style. Elements of the list specify the alternating lengths of dashes and gaps. The pattern starts with a dash.</p> <p>Range: For each element in the list: <1,2,...></p> <p>Default: Empty list</p>
pdf_line_dash_pattern_phase	<p>The value specifies the starting phase of the dash pattern. This attribute works together with <code>pdf_line_dash_pattern</code> to establish a simple dashed line style. The following is an example dash pattern specification:</p> <pre>pdf_line_dash_pattern{0} = 2 pdf_line_dash_pattern{1} = 3 pdf_line_dash_pattern_phase = 1</pre> <p>A dash of length 1 will be drawn, then gaps of length 3 and dashes of length 2 will cyclically follow thereafter.</p> <p>Before beginning to stroke a path, the dash array is cycled, adding the lengths of dashes and gaps. When the accumulated length equals the value specified by the dash phase, stroking of the path begins, and the dash array is used cyclically from that point onward.</p> <p>The table below shows examples of line dash patterns. As shown, an empty dash array and zero phase can be used to restore the dash pattern to a solid line:</p>

Attribute Name	Contents																					
	<table><tr><th>Dash Array and Phase</th><th>Appearance Divided by Unit</th><th>Description</th></tr><tr><td>[] 0</td><td></td><td>No dash; solid, unbroken lines</td></tr><tr><td>[3] 0</td><td></td><td>3 on, 3 off, ...</td></tr><tr><td>[2] 1</td><td></td><td>1 on, 2 off, 2 on, 2 off, ...</td></tr><tr><td>[2 1] 0</td><td></td><td>2 on, 1 off, 2 on, 1 off, ...</td></tr><tr><td>[3 5] 6</td><td></td><td>2 off, 3 on, 5 off, 3 on, 5 off, ...</td></tr><tr><td>[2 3] 11</td><td></td><td>1 on, 3 off, 2 on, 3 off, 2 on, ...</td></tr></table> <p>Range: <0,1,2,...></p> <p>Default: 0</p>	Dash Array and Phase	Appearance Divided by Unit	Description	[] 0		No dash; solid, unbroken lines	[3] 0		3 on, 3 off, ...	[2] 1		1 on, 2 off, 2 on, 2 off, ...	[2 1] 0		2 on, 1 off, 2 on, 1 off, ...	[3 5] 6		2 off, 3 on, 5 off, 3 on, 5 off, ...	[2 3] 11		1 on, 3 off, 2 on, 3 off, 2 on, ...
Dash Array and Phase	Appearance Divided by Unit	Description																				
[] 0		No dash; solid, unbroken lines																				
[3] 0		3 on, 3 off, ...																				
[2] 1		1 on, 2 off, 2 on, 2 off, ...																				
[2 1] 0		2 on, 1 off, 2 on, 1 off, ...																				
[3 5] 6		2 off, 3 on, 5 off, 3 on, 5 off, ...																				
[2 3] 11		1 on, 3 off, 2 on, 3 off, 2 on, ...																				
pdf_url	<p>If this attribute is set, then the feature will become an interactive annotation. When a user clicks on the feature in a PDF viewer application that supports URI actions, the value will be treated as a URI and it will be resolved. In the common case that the value is a URL, Adobe Acrobat Reader will open a web browser to resolve the address specified.</p> <p>Note: See the "Annotations" section under Feature Representation for behavioral notes.</p>																					
pdf_tooltip	<p>If this attribute is set, then the feature will become an interactive annotation. The value specifies the tooltip string that will be displayed when an user hovers over the feature with the mouse cursor in the PDF viewer application.</p> <p>Note: See the "Annotations" section under Feature Representation for behavioral notes.</p>																					
pdf_fill_opacity	<p>The value specifies the opacity of the fill color of the feature. A value of 1.0 is fully opaque, and 0.0 is completely transparent. If this value is not set, then the opacity of the feature is determined by the pdf_default_opacity feature type parameter. If the feature type parameter is not set either, then the writer directive DEFAULT_OPACITY determines the opacity.</p>																					
pdf_pen_opacity	<p>The value specifies the opacity of the stroking color of the feature. A value of 1.0 is fully opaque, and 0.0 is completely transparent. If this value is not set, then the stroking opacity is set to fully opaque.</p>																					

Annotations

pdf_type: any

Features with the pdf_url or the pdf_tooltip attribute set become annotation objects. There are several behavioral differences between annotation objects and non-annotation objects:

- Annotation objects will always appear above non-annotation objects, regardless of layer ordering.
- The interactive area of an annotation object is the rectangular bound of the feature instead of its precise outline.
- Annotation objects are no longer selectable through the Object Data tool or the Model Tree interface.
- Even when the annotation object's layer is hidden, the annotation will still provide tooltips and be interactive. The annotation object's parent layer does not affect the visibility of the annotation; only the object's layer itself will affect its visibility.

Points

pdf_type: pdf_point

A PDF point feature is drawn as a point with a radius of 1 typographical point.

The following attribute is applicable to point features:

pdf_point_width	The value specifies the point width in typographical points of point geometries. Range: <float> Default: 1.0
-----------------	---

Lines

pdf_type: pdf_line

A PDF line feature is drawn as a stroked line.

Area

pdf_type: pdf_area

A PDF area feature is written as a filled area with a stroked boundary. The fill opacity is controlled by the **DEFAULT_OPACITY** directive.

Collection

pdf_type: pdf_collection

Each component of a PDF collection feature is drawn according to their geometry type.

Text

pdf_type: pdf_text

A PDF text feature is drawn as a text annotation according to its `fme_text_string`, `fme_text_size`, and `fme_rotation` attributes.

The encoding of the text string is determined as follows: if the font is one of the PDF Core 14 fonts, then the string is decoded using Windows ANSI code page 1252. If the font is a TrueType font, then the string is decoded using the Macintosh Roman code page. If the TrueType font has a Microsoft Symbol character map table, then the font is treated as a symbolic font, and the text string can specify characters in the FF00-FFFF range of the character map by encoding only the low-byte of the code point. Desired characters in symbolic fonts can either be specified using XML numeric character references (NCR) in rich text format (see below) or if the code point coincides with ASCII characters, the ASCII characters themselves.

The following attributes are applicable to text features:

Attribute Name	Contents
<code>pdf_text_font</code>	The value specifies the default font family of the text representation. If left blank, Helvetica will be used. Default: Helvetica
<code>pdf_text_underline</code>	If the value is 'Y', the text will be underlined. Default: N
<code>pdf_text_strikethrough</code>	If the value is 'Y', the text will have a strikethrough. Default: N
<code>pdf_text_bold</code>	If the value is 'Y', the text will have a bold style. Default: N
<code>pdf_text_italic</code>	If the value is 'Y', the text will have an italic or oblique style. Default: N

The text string can be specified in a rich text format. The format is a subset of XHTML. For more information on XHTML, visit <http://www.w3.org/TR/xhtml1/>. The following are the supported XML elements:

- `<body>...</body>`, `...`, `<p>...</p>` - Can be used to specify a style for its enclosed text through its "style" attribute.
- `...` - Bolds the enclosed text.
- `<i>...</i>` - Italicizes the enclosed text.

- `<u>...</u>` - Underlines the enclosed text.
- `...` - Adds a strikethrough to the enclosed text.
- `
` - Adds a line break.

The "style" XML attribute has the following format:

`"property:value;...;property:value"`

The following properties are supported:

- font-family - Specifies the font family of the text.
- font-size - Specifies the point size of the font.
- color - Specifies the color of the text. The color can be specified through the format "#RRGGBB" where each color component is specified as a hexadecimal value, or through the 16 HTML color names (<http://www.w3.org/TR/REC-html40/types.html#h-6.5>).
- text-decoration - Valid values are "underline" and "line-through".

The following is an example rich text `fme_text_string` value:

`<body>Hello
world!</body>`

In the PDF document, the text "Hello" will use the styling specified through the format attributes. The text "World!" appears on the next line and will have a font size of 30 but will inherit all other style attributes.

Rasters

pdf_type: pdf_raster

A PDF raster feature is drawn as a grid of pixels comprising an image. Rasters written to PDF are converted to a JPEG2000 byte string and stored as a blob. Thus the nature of raster support is defined by the JPEG2000 format.

The following attribute is applicable to raster features:

Attribute Name	Contents
pdf_raster_compression_level	<p>It sets the quality of the compression.</p> <p>The range of the value is from 0 (best quality) to 100 (worst quality). The default value is 75. Setting the value to 0 enables lossless compression.</p> <p>This is a writer attribute.</p>

Format Mapping File Directives

Note: FME translations were originally based entirely on Mapping Files. Mapping files still exist under the surface but the interface has been almost entirely replaced by Workbench's graphical interface. Information on mapping files is included in this manual for technical reference purposes.

Mapping Files are ASCII text files that contain a series of rules that specify the FME readers, writers, and transformations (in Workbench, these are represented by transformers).

A mapping file (.fme) is a series of commands for FME to perform. Mapping files use functions and factories to transform the data. They also contain the definition and parameters for the readers and writers. A mapping file can be run through the FME Quick Translator. Before FME Workbench was designed and developed (about 2001), this was the only way to configure a translation process.

You can create a mapping file either by manually programming it or by using FME Workbench. In Workbench, there is still an Export as .fme tool on the toolbar. The Workbench file format itself (.fmw file) is partially a mapping file with an XML header. When FME runs a workspace it is converted into a mapping file.

Since mapping files are written in a plain ASCII format, so you can use any text editor to edit them. To see what a mapping file looks like: select one or more transformers in Workbench, copy them, and then paste them in a text editor. The mapping file equivalent of those transformers will be pasted.

Directives and Reader/Writer Keywords

Directives are processed by the reader or writer. Directives are prefixed by the current **<ReaderKeyword>** or **<WriterKeyword>** in a mapping file. By default, the keywords for formats are the format shortname (viewable in the Formats Gallery, or in the Format Quick Facts tables).

PDF2D Mapping File Directives

DATASET

Required/Optional: *Required*

The value for this directive is the path to the output file. If the output file does not exist, then the writer will create a new file. If the output file exists, then the writer will overwrite it. If other applications have the output file opened, then the writer will be unable to continue and the translation will fail.

Workbench Parameter: *Destination PDF File*

DEF

Required/Optional: *Required*

The PDF2D writer uses **PDF2D_DEF** lines to define feature types. A typical mapping file fragment specifying a feature type looks like:

```
PDF2D_DEF <featureName> \
    [pdf_layer_order <layerOrder>]? \
    [pdf_in_page_coordinates <pageCoordinates>]? \
    [pdf_default_opacity <opacity>]? \
    [pdf_layer_visibility <visibility>]? \
    [<attributeName> <attributeType>]*
```

The configuration parameters present on the definition line are described in the following table:

Parameter	Contents
featureName	This declares the name of the feature type.
attributeName	This declares the name of an attribute. The maximum length of attribute names is 200 characters.
attributeType	This declares the type of the attribute. The only valid attribute type is string.
layerOrder	This declares the layer order of the feature type. Valid values are all integers. Feature types with lower layer orders will be drawn first. Therefore, features in feature types with higher layer orders will appear on top of features in feature types with lower layer orders. If a value is not specified, then the feature type will have an effective layer order value of '0'. If two features have identical layer order values, then the two will be ordered arbitrarily.
pageCoordinates	The value specifies whether the coordinates of geometries will be interpreted in page coordinates. If this attribute is set to YES, then the coordinates of the geometry are treated as page coordinate values, and the feature can be drawn anywhere on the page. The default value is NO.
opacity	This determines the opacity level of features of this feature type when their pdf_opacity feature attribute is unset. If this parameter is set, the value overrides the writer parameter DEFAULT_OPACITY. A value of 1.0 is fully opaque, and 0.0 is completely transparent.
visibility	If the value is VISIBLE, then the layer will be visible by default after opening the output file in Adobe Acrobat Reader. If set to HIDDEN, then the layer will not be initially visible. The visibility of layers can be toggled in Adobe Acrobat Reader after opening the file.

PAGE_SIZE

This directive specifies the size of the output page of the PDF document. The default page size is Letter.

Preset page sizes for common paper sizes can be selected, or the page size can be specified in typographical points in the format <width> <height>.

Required/Optional

Optional

Values

A3 | A4 | A5 | B5 | Ledger | Legal | Legal-half | Letter (Default) | Letter-half | <0 ...>
<0 ...>

* Workbench Parameter

Page size

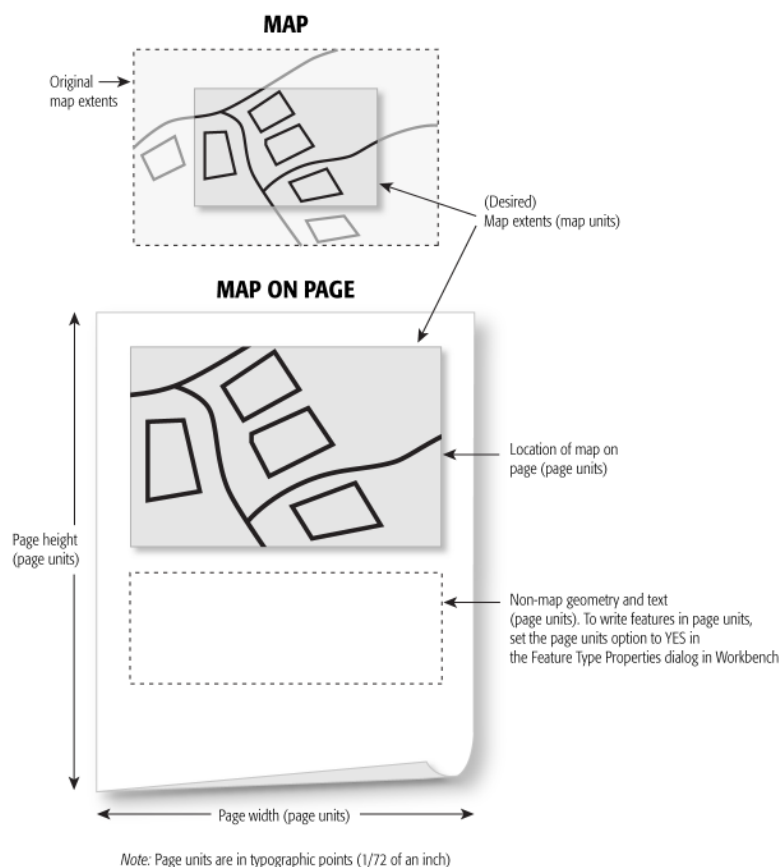
PAGE_VIEWPORT (Location of Map on Page)

This directive determines where to place the map on the page, and how large the map should be on the page.

The format for this directive is four integers separated by spaces describing the lower left corner and the upper right corner of the viewport/rectangle, specified in typographical points. The lower left corner of the page contains coordinate (0,0) and the top right corner contains coordinate (<width>,<height>), where these two values are the page size specified by PAGE_SIZE.

If the aspect ratios of the page viewport and the world viewport (WORLD_VIEWPORT) differ, then the lesser scaling factor will be chosen: data inside the world viewport will not be clipped and data outside the world viewport might become visible.

If a value for the directive is not specified, then the page viewport rectangle will be a centered rectangle with a width and length that is 90% of the page width and length. The page viewport coordinates must be between (0,0) and (page width,page height).



Required/Optional

Optional

Values

<minimum x> <minimum y> <maximum x> <maximum y>

The values can also be specified as a percentage of the page width and page height. The values must be an integer ending with a percentage sign. The values can also be negative values, and they are interpreted as being relative to the top and right edges instead of the left and bottom edges. For example, for a page size of 1000 by 1000 points, the rectangle "50 50 -50 -50" is identical to the rectangle "50 50 950 950" for this page size.

* Workbench Parameter

Page viewport dimensions

WORLD_VIEWPORT (Map Extents)

This directive specifies the extents of the map to write within the page viewport, by defining the lower left and upper right corners of the page viewport in map units.

Geometry outside these extents will be clipped when drawn on the page. The format for the directive is four floating point numbers separated by spaces describing the lower left corner and the upper right corner of the rectangle.

If a value for the directive is not specified, then the world viewport rectangle will be the bounding box of the entire dataset.

Required/Optional

Optional

Values

<minimum x> <minimum y> <maximum x> <maximum y>

*** Workbench Parameter**

World viewport dimensions

DEFAULT_OPACITY

This directive specifies the opacity value of the fill color of area geometries. The boundaries of area geometries are not affected by this setting.

Required/Optional

Optional

Values

<0.0...1.0>

A value of 0 corresponds to complete transparency and a value of 1 is complete opaqueness.

Default Value: 0.4

*** Workbench Parameter**

Default fill opacity value

DEFAULT_POINT_SIZE

This directive specifies the default radius in typographical points for point geometry.

Required/Optional

Optional

Values

<0.0...>

Default Value: 1.0

*** Workbench Parameter**

Default point size value

DEFAULT_LINE_WIDTH

This directive specifies the default width in typographical points for line geometry and boundaries of area geometry.

Required/Optional

Optional

Values

<0.0...>

Default Value: 1.0

*** Workbench Parameter**

Default line width value

PANEL_VISIBILITY

This directive determines the panel that is visible immediately after opening the output PDF file in Adobe Acrobat software.

Required/Optional

Optional

Values

None (default): No panel will be initially displayed

Layers: Layer panel will be visible after opening the file

Pages: Page Thumbnails panel will be visible

*** Workbench Parameter**

Navigation Panel to Display

RANDOMIZE_FEATURE_TYPE_COLOR

This directive specifies whether features without the `fme_color` attribute set will be assigned a random color based on its feature type.

Required/Optional

Optional

Values

YES (default)

NO (features without the `fme_color` attribute set will be assigned the color black)

* Workbench Parameter

Randomize Feature Type Color

RICH_TEXT

Required/Optional: *Optional*

This directive specifies whether the text string of text features is in the rich text format. If this directive is set to NO, then the text string is written as-is to the page. If this directive is set to YES, then the text string will be processed for style directives. For more details, see the "Text" section under Feature Representation.

Values: *YES|NO*

Default Value: *NO*

Workbench Parameter: *Text in rich text format*

FONT_DIRECTORIES

Required/Optional: *Optional*

This directive specifies the directories that the writer will search in to find the TrueType fonts used in the workspace. The workspace directory of the translation is always searched.

Values: *<multiple directories>*

Default Value:

Workbench Parameter: *TrueType font directories*

WRITE_ATTRIBUTES

Required/Optional: *Optional*

This directive specifies whether attribution data will be written. Not writing attribution data will decrease the file size of the output file and may improve viewing performance.

Values: *YES|NO*

Default Value: *YES*

Workbench Parameter: *Write attributes*

COMPRESS_STREAMS

Required/Optional: *Optional*

This directive specifies whether streams in PDF files will be compressed.

Values: *YES|NO*

Default Value: *YES*

Workbench Parameter: *Compress streams*

COMPRESS_OBJECT_STREAMS

This directive specifies the object stream compression, and the output format of the PDF cross-reference information and trailer.

If the directive is set to YES, the output file will contain compressed object streams, and cross-reference information will be stored in cross-reference streams.

If the directive is set to NO, the output file will not contain compressed object streams, and cross-reference information will be stored in a cross-reference table and trailer. Compressed object streams and cross-reference streams are supported in PDF1.5 and higher.

Required/Optional

Optional

Values

YES (default) | NO

*** Workbench Parameter**

Compress Object Streams

PDF_14_COMPATIBILITY

Required/Optional: Optional

This directive specifies whether or not to write files compatible with PDF 1.4 viewers. The default value of NO means that features introduced in PDF 1.5 and later will be used, including object stream compression and the JPEG2000 raster image format. A value of YES means that the output file will not contain compressed object streams, cross-reference information is stored in a cross-reference table and xref trailer, and rasters are encoded in the JPEG format. Note that the COMPRESS_OBJECT_STREAMS directive is deprecated, and equivalent to the opposite of this directive.

Values: YES|NO

Default Value: NO

Workbench Parameter: PDF 1.4 Compatibility

FRAME_SPECS

Required/Optional: *Optional*

This directive can be used to specify frame and page properties (see the Multi-page Support section). If the output will have a static number of frames and pages (i.e., the number and properties of frames and pages is not dependant on the data) then this directive provides a convenient and simple method of specifying frame and page properties. The format for this directive is a whitespace-delimited sequence of keywords and values:

```

Specification: [PageSpec|FrameSpec]*
PageSpec: page <page number> PageSpecAttributes*
PageSpecAttributes:
page_size <page size>
FrameSpec: frame <frame name> FrameSpecAttributes*
FrameSpecAttributes:
    page_number <page number>
| frame_rectangle <frame rectangle>
| frame_order <frame order>
| world_rectangle <world rectangle>

```

The formats of the <page number>, <page size>, <frame rectangle>, <frame order>, and <world rectangle> values are identical to the formats for the pdf_page_number, pdf_page_size, pdf_frame_rectangle, and pdf_world_rectangle attributes respectively. The order that these attributes are specified is insignificant. Properties that are not specified will use their default values as described in the Multi-page Support section.

The following is an example value for this directive:

```

frame f1 frame_rectangle 10% 10% -10% -10% page_number 2
frame f2 frame_order 0
frame f3 frame_order 1
page 1 page_size 500 500

```

The result is as follows: The output document has two pages. Both pages have size 500x500 points (the unspecified second page inherits its page size from the first page). Frame "f1" will have a 10% margin on the second page. Frame "f3" will be drawn above frame "f2" on the first page, and the two frames will have the frame rectangle specified by the PAGE_VIEWPORT writer directive. All three frames use the world rectangle specified by the WORLD_VIEWPORT writer directive. Features can be written to one of these three frames by setting the pdf_frame_name attribute to "f1", "f2", or "f3".

* Workbench Parameter

Frame and Page Specification Lines

