

Vector Markup Language (VML) Writer

The Vector Markup Language (VML) Format Writer module enables the Feature Manipulation Engine (FME) to be used in conjunction with the world wide web to translate vector data on-the-fly for display in web browsers. VML is an Extensible Markup Language (XML) based exchange, editing, and delivery format for vector graphics on the web. This section assumes familiarity with the VML format and the XML standard.

Overview

VML, which is written using the XML syntax, is a text-based markup language used for describing vector graphics that can be viewed and edited by a wide variety of tools. It provides for the description of lines, polygons, curves, images, and text objects. Positioning and layout of the vector graphics are accomplished by using the Cascading Style Sheets, Level 2 (CSS2) visual rendering model. For more information on VML, XML, CSS1, and CSS2 see the World Wide Web Consortium Web site at <http://www.w3.org>.

The two primary objects that VML describes are the shape and group elements. The shape element is used to define a visible vector graphic element whereas the group element is used to group together several shapes so that they may be transformed together as one unit. These are top-level elements that may define their own local coordinate system. The shape and group coordinate spaces define a CSS2 block level box.

In addition, VML defines several auxiliary top-level elements to help make the editing and representation of graphical information more compact and convenient. These auxiliary elements are the `shapetype` element and the predefined shape elements `line`, `polyline`, `curve`, `rect`, `roundrect`, `oval`, `arc` and `image`. The usage of some of these VML elements currently supported by FME are explained in subsequent sections.

VML Quick Facts

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

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

Writer Overview

The VML Format writer creates a single VML top-level group element that is used as the containing block for all feature data. This top-level group element defines the coordinate space for all of its sub-elements. The coordinate space for the top-level group element may be specified with the FME **COORDSIZE** and **COORDORIGIN** mapping file directives. These directives are described under the *Writer Directives*. All features are drawn relative to its containing block's —the top-level group element—top left corner. The specified coordinate space has its positive x and y axis in the lower right quadrant.

The VML writer creates a single HyperText Markup Language (HTML) file with the VML data contained in the <body> region of the HTML. At the time of this writing, the only browser supporting VML is Internet Explorer 5 (IE5) and, as a result, the header of the HTML file produced contains information specific to this browser. This information is needed in the header to let other browsers know that the embedded VML data is to be handed off to the browser's VML-specific processor.

Writer Directives

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

DATASET

Required/Optional: *Required*

The value for this directive is the name of the VML file to be created. You may want to add the `.html` extension to the file name since the file produced is an HTML file. If a file with this name already exists, then the file will be overwritten. A typical mapping file fragment specifying an output VML data set looks like:

```
VML_DATASET /tmp/outputFile
```

LEFT

Required/Optional: *Optional*

This directive specifies the left position on the web page for the top-level group element. The top-level group element is the container for all VML features that are drawn.

The syntax for VML `LEFT` is:

```
<WriterKeyword>_LEFT <value> (where <value> is in CSS length units)
```

The default value for this directive is 100pt.

Note: CSS length values are formed by an optional + or -, followed by a number, followed by a two-letter abbreviation that indicates the unit. There are two types of length units—relative and absolute. Relative length units give a length relative to another length property. The following relative units are available: `em`—ems, the height of the element's font; `ex`—x-height, the height of the letter 'x'; and `px`—pixels, relative to the canvas resolution. Absolute length units are highly dependent on the output medium. The following absolute units are available: `in`—inches; `cm`—centimetres; `mm`—millimetres; `pt`—points; and `pc`—picas. For more information on CSS units, please see the CSS1 or CSS2 specifications on the <http://www.w3.org> website.

TOP

Required/Optional: *Optional*

This directive specifies the top position on the web page for the top-level group element.

The syntax for VML `TOP` is:

```
<WriterKeyword>_TOP <top> (where <top> is in CSS length units)
```

The default value for this directive is 100pt.

Note: The values for the `LEFT` and `TOP` directive are written out to be the values for the CSS “left” and “top” style attributes of the top-level group element, respectively. You can modify these two CSS style attributes, with any text editor, to reposition the drawing on the web page. Note that the CSS “left” and “top” style attributes are ignored by web browsers if the CSS *position* style for a shape is not set to *absolute*. By default, the VML writer sets the top-level group CSS *position* style to *absolute*. If this is not suitable, you may use a text editor to change the *position* style attribute to be either *static* or *relative*.

WIDTH

Required/Optional: *Optional*

This directive sets the width, in CSS units, of the containing block for the top-level group element.

The syntax for VML `WIDTH` is:

```
<WriterKeyword>_WIDTH <width> (where <width> is in CSS length units)
```

The default value for this directive is 512pt.

HEIGHT

Required/Optional: *Optional*

This directive sets the height, in CSS units, of the containing block for the top-level group element.

The syntax for VML `HEIGHT` is:

```
<WriterKeyword>_HEIGHT <height> (where <height> is in CSS length units)
```

The default value for this directive is 512pt.

Note: The values for the `WIDTH` and `HEIGHT` directives are written out to be the values for the CSS *width* and *height* style attributes of the top-level group element, respectively. These CSS style attributes for the group may be modified in the translated output file to change the size of the vector drawing on the web page. Changing these does not affect the local coordinate space set by the top-level group element.

COORDSIZE

Required/Optional: *Optional*

This directive defines the number of units along the width and height of the containing block for the top-level group element.

The syntax for VML `COORDSIZE` is:

<WriterKeyword>_COORDSIZE <width> <height>

Note: The value for this directive becomes the value for the VML `coordsize` attribute in the top-level group element. This value should not be modified after the VML output file has been written. Modifying this in the VML output file produces a malformed vector drawing because the position of all elements within the group were calculated based on the original values given by the `COORDSIZE` and `COORDORIGIN` directives.

COORDORIGIN

Required/Optional: *Optional*

This directive defines the coordinate at the top left corner of the containing block for the top-level group element. The positive y-axis is downwards.

The syntax for VML `COORDORIGIN` is:

<WriterKeyword>_COORDORIGIN <left> <top>

Note: The value for this directive becomes the value for the VML `coordorigin` attribute in the top-level group element. The y-axis is inverted; positive is downwards.

SPATIAL_EXTENT

Required/Optional: *Optional*

This directive fixes the spatial extent that the VML output covers, in ground units. The specified spatial extent must be greater than or equal to the minimum bounding rectangle of the feature data. This directive when used in conjunction with the `COORDSIZE` and `COORDORIGIN` directives is useful for maintaining the same VML coordinate space for different output VML files that were translated at different times. Assuming that the `SPATIAL_EXTENT`, `COORDSIZE` and `COORDORIGIN` were kept invariant for all translations, the contents of the output VML files may then be combined into one file by copying and pasting the shapes from the different groups into a single group. If this directive is not specified, then the spatial extent will be set equal to the minimum bounding rectangle of the feature data.

The syntax for VML `SPATIAL_EXTENT` is:

<WriterKeyword>_SPATIAL_EXTENT <min-x> <min-y> <max-x> <max-y>

KEEP_ASPECT_RATIO

Required/Optional: *Optional*

This directive directs the VML writer to maintain the original aspect ratio—determined by spatial extent in ground units—of the input feature data.

The syntax for VML `KEEP_ASPECT_RATIO` is:

<WriterKeyword>_KEEP_ASPECT_RATIO (YES|NO)

The default value for this directive is YES.

PRETTY_PRINT

Required/Optional: *Optional*

This directive gives the option for the VML writer to print the output file in a more attractive format.

The syntax for VML PRETTY_PRINT is:

```
<WriterKeyword>_PRETTY_PRINT (YES|NO)
```

The default value for this directive is NO.

Tip: Enabling this option produces a considerably larger VML output file due to extra blank spaces.

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.

Special attributes direct the VML writer as it writes the features into the VML file. The most important of these is the `vml_type` attribute which controls the interpretation of the feature. Acceptable values for `vml_type` are `vml_text`, `vml_polyline`, `vml_polygon`, and `vml_point`.

User-defined attributes are ignored by the VML writer. Limited user attribution can be output in the VML file by using the `vml_title` attribute or the `vml_url` attribute allowing a feature to have a URL link which may point to some external data source.

Some VML feature attributes have data type VML boolean, VML number, or VML color. The value for these attributes are copied directly from the FME mapping file to the VML output file, and therefore must conform to the VML specification. The following table lists the VML data type specification for these attributes.

Data Type	Description
VML boolean	An attribute which can take true and false values. The following directives are defined for VML. Value for true: true yes on t 1 Value for false: false no off f 0
VML number	Numeric data used for values that are integer or fractional numbers and for values that specify lengths. Lengths and numbers follow the lexical form defined for CSS with a suffix indicating a scale factor.

Data Type	Description
VML color	<p>The full set of values are taken from HTML, CSS1, and VML specification.</p> <p>Example: HTML defines the following 16 colours. black silver gray white maroon red purple fuchsia green lime olive yellow navy blue teal aqua</p> <p>Example: Using the CSS1 rgb form "rgb(red,green,blue)" where red, green, and blue are values in the range of 0..255.</p>

The following table lists the attributes that are common to all VML features:

Attribute Name	Contents
vml_color	<p>The color of the brush used to stroke the feature.</p> <p>Range: VML color Default: black</p>
vml_weight	<p>The width of the brush used to stroke the feature.</p> <p>Range: VML number Default: 0.75pt</p>
vml_title	<p>The title of the feature that may be displayed by the VML viewer.</p> <p>Range: string Default: None</p>
vml_url	<p>The URL to jump to if this feature is clicked on.</p> <p>Range: string Default: None</p>
vml_target	<p>The target frame in an URL.</p> <p>Range: string Default: None</p>
vml_z_index	<p>The <i>z-index</i> of the feature in the output VML file. Positive numbers are in front of the screen. Negative numbers are behind the screen. Features having a higher <i>z-index</i> obscure features with lower <i>z-index</i>.</p> <p>Range: integer Default: 0 (for vml_polygon features) 10 (for vml_polyline features) 11 (for vml_point features) 12 (for vml_text features)</p>
vml_fill_color	<p>The color of the brush used to fill the feature. This attribute is not applicable for vml_polyline features.</p> <p>Range: VML color Default: black for points and text No Default for polygons</p>

Attribute Name	Contents
vml_fill_attr{#}	<p>This list attribute allows the feature to be filled with customized effects. If a feature has this list attribute the VML element that represents the feature in the output file will contain a VML fill sub-element. The contents of the vml_fill_attr{#} list attribute must be of the following form: <attribute-name>=<attribute-value></p> <p>Where <attribute-name> is a name of an attribute for the VML fill sub-element, and <attribute-value> is one of the possible values for that attribute name. Please refer to the VML specification for all the possible attributes that the VML fill sub-element may contain. The VML specification may be found at http://www.w3.org.</p> <p>For example, to specify that a feature is filled with a gradient and blue color add the following vml_fill_attr{#} list attribute with the following values to the feature: vml_fill_attr{0} "type=gradient" vml_fill_attr{1} "color=blue"</p> <p>NOTE: If the above string values contain spaces then they must be enclosed between double quotes.</p> <p>Also note that the index for the vml_fill_attr{0} must start from 0. The order in which the attributes are listed in the list attribute is of no importance.</p> <p>Range: <attribute-name>=<attribute-value> as described above</p> <p>Default: No Default</p>
vml_stroke_attr{#}	<p>This list attribute allows the feature to be render with a customized outline. The values for this list attribute is similar to the vml_fill_attr{#} described above with the values for the <attribute-name> and <attribute-value> taken from the VML stroke sub-element instead.</p> <p>Example, to specify that a feature should be render with a dotted blue line add the following vml_stroke_attr{#} list attribute with the following values to the feature: vml_stroke_attr{0} "color=blue" vml_stroke_attr{1} "dashstyle=dot"</p> <p>Range: <attribute-name>=<attribute-value></p> <p>Default: No Default</p>
vml_shadow_attr{#}	<p>This list attribute allows the feature to be render with a shadow effect. The values for this list attribute is similar to the vml_fill_attr{#} described above with the values for the <attribute-name> and <attribute-value> taken from the VML shadow sub-element instead.</p> <p>Example, to specify that a feature should be render with a dotted blue line add the following list attribute with the following values to the feature: vml_shadow_attr{0} "on=true" vml_shadow_attr{1} "type=perspective"</p> <p>Range: <attribute-name>=<attribute-value></p> <p>Default: No Default</p>

Attribute Name	Contents
vml_imagedata_attr{#}	<p>This list attribute allows the feature to have a picture render on top of it. The values for this list attribute is similar to the vml_fill_attr{#} described above with the values for the <attribute-name> and <attribute-value> taken from the VML imagedata sub-element instead.</p> <p>Example, to specify that a feature should be rendered with the sample 's.jpg' image on top add the vml_imagedata_attr{#} list attribute with the following values to the feature: vml_imagedata_attr{0} "src=c:\temp\s.jpg"</p> <p>Range: <attribute-name>=<attribute-value> Default: No Default</p>

Points

vml_type: vml_point

Point features must have exactly one coordinate. The VML writer uses the predefined VML oval element to generate round point features. Point features have their vml_fill_color set to black by default.

VML point features have the following additional attribute:

Attribute Name	Contents
vml_point_size	<p>The size of the point in ground units.</p> <p>Range: real>0 Default: x, where $x=0.006*\text{deltaY}$. deltaY is the y-coordinate range of the spatial extent of the input data in ground units.</p>

Polylines

vml_type: vml_polyline

Polyline features must have at least two coordinates. The VML writer writes out a vml_polyline feature in the VML output file as a predefined VML polyline element. The writer also writes out the z-index attribute to 10 so that, by default, the polylines only obscure the polygons.

Polygon

vml_type: vml_polygon

Polygon features must have at least four coordinates, with the last coordinate equal to the first coordinate. The vml_polygon features may not contain holes. The VML writer writes out a vml_polygon feature in the output file as a predefined VML polyline element. By default, the VML writer does not set the z-index attribute. When it's not set, the web browser interprets the polygon—the VML filled polyline element—to the have a z-index of 0. Polygons produced by the VML writer, therefore, by default do not obscure other objects with the exception of overlapping polygons with the same z-index value. Objects drawn later with equal z-indexes obscure earlier ones.

Text

vml_type: vml_text

Text features must have exactly one coordinate. The `vml_text_string` attribute must also be present in the `vml_text` feature. Text is drawn in the output file by placing a VML `textpath` sub-element inside of a VML `shape` element. By default, `vml_text` features have their `vml_fill_color` attribute set to `black`.

VML text features have the following additional attributes:

Attribute Name	Contents
<code>vml_text_string</code>	The text string may contain blanks and there is no limit on its length. This attribute must be present for all <code>vml_text</code> features. Range: string Default: None
<code>vml_text_size</code>	The size of the text in ground units. Range: real>0 Default: x , where $x=0.008*\text{deltaY}$. <code>deltaY</code> is the y-coordinate range of the spatial extent of the input data in ground units.
<code>vml_text_justification</code>	The justification of the text. Range: left center right Default: left
<code>vml_rotation</code>	The rotation of the text, as measured in degrees counter-clockwise from the horizontal. Range: -360.0...360.0 Default: 0
<code>vml_font_family</code>	The CSS1 font family name. Range: CSS1 font family name. Default: Times New Roman
<code>vml_font_style</code>	The style of the font. Range: normal italic oblique Default: normal
<code>vml_font_weight</code>	The weight of the font. Range: normal bold bolder lighter 100 200 300 400 500 600 700 800 900 Default: normal
<code>vml_rotate_letters</code>	Rotate the letters of the text by 90 degrees. Range: VML boolean Default: f
<code>vml_same_letter_heights</code>	Stretches lowercase letters to the height of uppercase letters. Range: VML boolean Default: f