

VertexSymbol Plugin

For

OpenJUMP

Version 0.15
© 2007 cadplan.com.au

Introduction

VertexSymbols for OpenJUMP is a plugin to display a wide range of symbols on the vertices associated with points, lines and polygons. By default OpenJUMP provides a single vertex symbol (a square shape) that can be enabled or disabled.

This plugin offers three types of vertex symbols:

- Parameterized vector symbols that are defined by a vector drawing that can be scaled for size, rotated for orientation, filled or unfilled, and dotted if required. These symbols are hard coded.
- Images that are created by the user from gif, jpeg , png image formatted files and from SVG (Scaled Vector Graphic) files. These symbols are user defined.
- Vector –based symbols defined by a WKT (Well Known Text) specification. These symbols are user defined.

Vertex symbols can also be rotated on each feature in a layer according to a value of a selected attribute when the attribute value specified an angle in the range 0 to 360°.

In each case a vertex symbol can have an associated text label. These text labels are *not* the same as the built-in label option. The vertex text labels can be:

- The value of any feature attribute (of type Integer, Double, String, Date or Geometry). It is also possible to select the feature ID (\$FID) and the actual coordinate of the vertex (\$POINT).
- They can have their own font, style and colour settings.
- They can be offset from the vertex position, either automatically according to the vertex symbol size, or to a set fixed amount.
- String type attributes can be formatted as HTML.
- Vertex labels can be enabled/disabled selectively for each feature in a layer.

Normally, a symbol is applied to a layer, so that all features in the layer will have the same symbol. From vers 0.15 it is possible to specify different symbols for selected features. This means that the layer as a default symbol, and a feature can have a special symbol.

Installation

The VertexSymbols.jar file is placed in the plugins directory. From Version 0.3, The Apache Batik libraries must also be installed. These are available from Apache at:

<http://xmlgraphics.apache.org/batik/download.cgi>

All the jar files contained in the Batik release (i.e. inside the lib folder) should be placed in a “Batik” folder inside the OpenJUMP lib folder. ***Each of the jar files (about 18 in all) must be added to the Classpath for OpenJUMP (perhaps all are not needed, but most are). It is necessary to edit the OpenJump.bat file to update the Classpath.***

Some other plugins also use the Batik libraries, there should be no incompatibilities.

If any of the required Batik libraries are missing, an error message will show when the VertexSymbols plugin is being loaded.

If image type symbols are to be used then they are placed in a folder named “VertexSymbols” inside the plugins folder. All files having “gif”, “jpeg”, “jpg”, “png” or “svg” extensions in this folder are treated as vertex symbol images and are loaded when the plugin is loaded.

There are in fact two plugins in the jar file:

- VertexSymbols: that allows vertex symbols to be specified.
- VertexNote: that allows the text labels to be placed selectively on chosen features with the selected layer. It also provides a place where the attribute value of a chosen feature can be edited (only applies for attributed that can/should be edited).

Once OpenJUMP is launched, the plugin is accessed from “Vertex Symbols” option on the “Plugins” menu.

Operation

From the Vertex Symbols Dialog, the user can select a required vertex symbol. Upon accepting this choice the chosen vertex symbol is applied to all currently selected layers. Each layer has a single vertex style. The vertex style is applied to all points, lines and polygons in the layer.

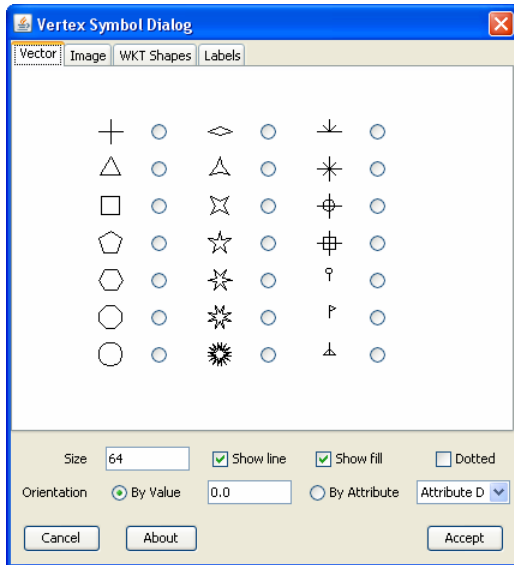
Labels can be added to each vertex, set by selecting an attributed from the selected layer. Labels can also be displayed selectively on a feature by feature basis.

Note that the specification of the vertex symbols is contained in the OpenJUMP project file. A project file created using this plugin will not open if the plugin is removed or not available when OpenJump is launched.

Vector Symbols

The Vertex Symbols Dialog has three tabbed panes. The first is the Vector panel that contains a selection of hard coded vector generated symbols as shown in the following figure. Currently there are three styles:

- Polygon based shapes
- Star based shapes
- Other shapes



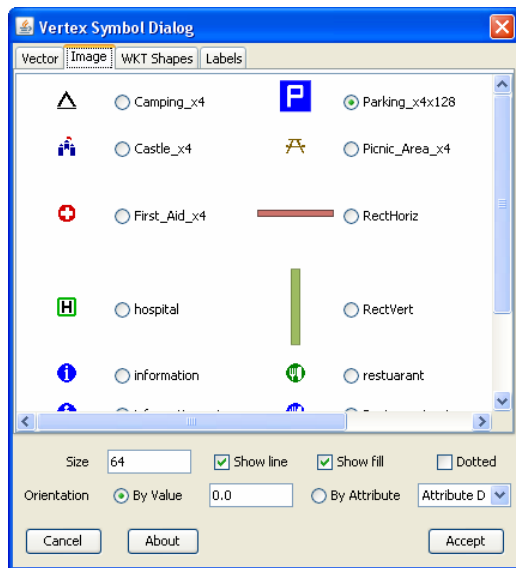
The required symbol is chosen by the adjacent radio button. There are five parameters that fully define each symbol:

1. Size: the width/height of the symbol in screen pixels. The size is “nominal” only. The value is chosen so that the symbol looks right on the screen.
2. Orientation: there are two options:
 - By Value: where the orientation of the symbol in degrees is set in the text field. (e.g. 45 will rotate all symbols in the selected layer 45° in a clockwise direction). Note that the direction of rotation is opposite to that used for label rotation in OpenJUMP.
 - By Attribute: where the orientation is set according to the value of an attribute in the selected layer. The attribute is selected from the list in the following combobox. For the currently selected layer all available (numeric valued) attributes are listed. It is assumed that the value of the selected attribute is an angular measure the range 0 to 360, and this represents a clockwise angular rotation from North. If text labels are enabled, the label is not rotated, only the symbol is rotated.
3. Show line: will display the edge line as specified in the layer.
4. Show fill: will fill the symbol with the layer fill color.
5. Dotted: will place a small dot at the centre (i.e. the reference co-ordinate) of the symbol.

The Show Line, Show Fill and Dotted options only applies to Vector symbols.

Image Symbols

Selecting the Image tab will display the image symbol options as shown in the following picture (examples only). The images are presented in a scrolling panel to allow for the user to add as many images as required.



The images to be loaded are user defined (e.g. created using a third party tool) and are placed in the VertexImages folder of the plugin folder. All images found in this folder will be loaded when the plugin is loaded.

Images can be any of: gif, jpeg, png or svg file formats, with appropriate file name extensions (“.gif”, “.jpg” or “.jpeg”, “.png”, “.svg”).

The name of the image that appears in the dialog matches the corresponding file name with the extension part removed.

SVG images are defined by a vector specification and are not normally subject to resolution problems. The implementation in this plugin, however, converts each SVG image into a bitmap for internal use.

Once loaded there is no distinction made between bit-mapped and SVG image types.

As a spin-off, it is possible to use image vertices to place larger images in a layer. There is no real limit on the size of an image and all you need is a point in a layer to attach an image to it and thus in the layer. Moving the point around will move the image to a suitable location. Just imagine the images you could add to your project!

WKT Defined Symbols

WKT vertex symbols are defined using the WKT specification. They can be formed from:

- **LINestring** or **MULTILINestring**: i.e. composed of one or more sets of connected line segments. These symbols are line drawings only.
- **POLYGON** or **MULTIPOLYGON**; i.e. composed of one or more polygons, possibly with holes in the polygons. These symbols can be drawn filled or unfilled.
- **GEOMETRYCOLLECTION**: i.e. composed of **LINestring** and **POLYGON** shapes.

Each symbol definition is contained in a single text file with a “.wkt” extension. The format is:

<Line width>:<Symbol extent>:<WKT specification>

Where:

- Line width is the width of the line (integer).
- Symbol extent is the size of the square shape containing the symbol (integer)
- WKT specification is a standard WKT text string.

These three elements are colon delimited and can be extended over multiple lines as required.

Some WKT examples:

```
1:10:MULTILINESTRING((-5 -3, 5 -3), (-5 0, 5 0), (0 0, 0 3))
```

Defines the *sample4* symbol in the figure below. It is three line segments in a 10x10 cell.

```
1:10:POLYGON((5 5, 5 -5, -5 -5, -5 5, 5 5),(1 1, 1 -1, -1 -1, -1 1, 1 1))
```

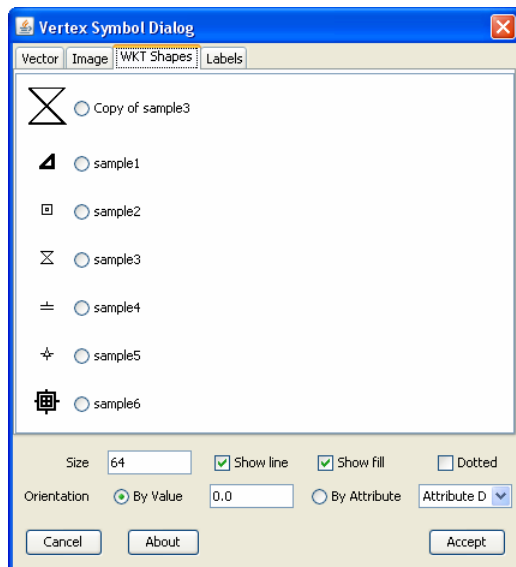
Defines the *sample2* symbol in the figure below. It is a square with a square hole at the centre in a 10x10 cell.

```
2:20:GEOMETRYCOLLECTION(MULTILINESTRING((-10 0, 10 0),(0 10, 0 -10)),  
POLYGON((-7 7, 7 7, 7 -7, -7 -7, -7 7),(-4 4, 4 4, 4 -4, -4 -4, -4 4)))
```

Defines the *sample6* symbol in the figure below as a cross shape together with a square with a square hole inside a 20x20 cell size using a line width of 2.

The origin for the WKT coordinates is at the centre of the vertex cell extent.

If there is an error in the WKT specification then a warning message will be displayed when the plugin is first loaded.



Label Option

Each vertex in a layer can have an associated text label. This is in addition to the label option provided in the core. Labels can be multi-line, where each line is delimited by a 'l' character in the attribute value (only for String type attributes).

As shown in the Vertex Note Plugin section (below), for STRING type attributes, the text value can also be formatted as an HTML specification. This allows a wide range of formatting to be embedded in the label. The default font (face, type, size) is specified in the VertexSymbols dialog for the current layer. In this case the attribute must begin with “<HTML>” and end with “</HTML>”.

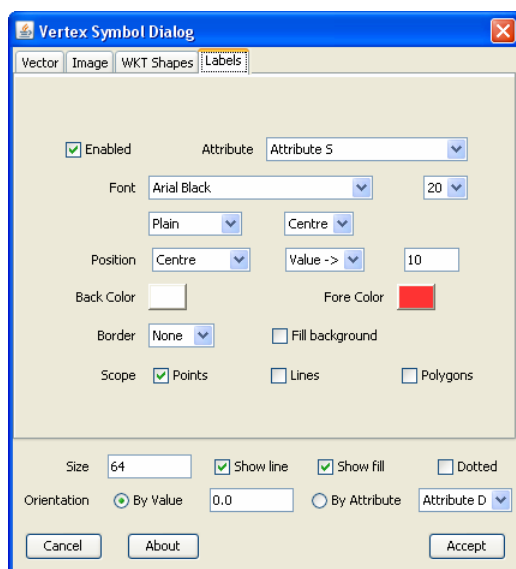
Labels can be made from attributes with types Integer, Double, String and Date. In the case of Date the format is like: “2007/07/17” if the date has zero hours, minutes and seconds, else it is: “2007/07/17 16:23:43” (over two lines).

It is also possible to select the Geometry attribute, though this option may not be useful for anything other than points. The geometry is displayed in WKT format.

The “\$FID” attribute can also be chosen. It is the feature ID that is unique to each feature and allocated when the feature is first created. The feature ID is the same for all vertices belonging to the same feature.

A “\$POINT” attribute is also available to display the actual coordinates of the vertex in the form “(x,y)”. This option allows the coordinates of each vertex in the line or polygon to be displayed as a label.

The Labels panel provides the various setting options.



The fields are:

- Enabled:** Check to enable labels.
- Attribute:** The attribute from which the value to be displayed is selected (only of type Integer, Double, String, Date and Geometry) as well as \$FID and \$POINT.
- Font:** The required font: name, size, style and justification for multi-line labels.
- Position:** The position of the label relative to the vertex point; can be Centre, North, North-East, East, South-East, South, South-West, West and North-West. The Offset can be “None” – the label is positioned at the vertex; “Auto” – adjusts for vertex size; or a “Value ->” a set value.

The Auto offset option uses the size of the vertex symbol to determine the amount of offset, depending on the chosen position for the label. If the symbol has lots of white space included in its square size*size shape, then the manual offset value may give a better position. The vertex symbol is always assumed to be a square shape centred at the vertex position.

Back Color: The background color for the text block (if fill enabled). Click button to launch color chooser.

Fore Color: The foreground color for the text and border (if enabled). Click button to launch color chooser.

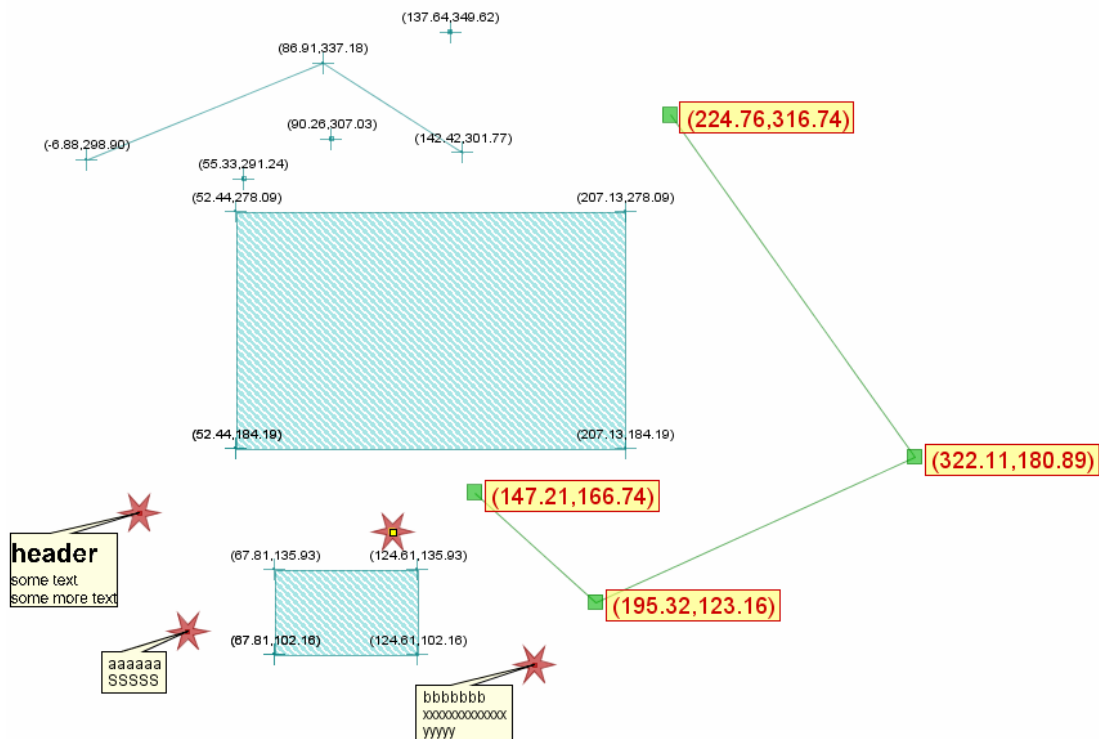
Border: The type of border:

- None – no border
- Boxed – a rectangular border
- Callout – a rectangular border with an pointer to the vertex.

Fill Background: Fill the text block with a background color.

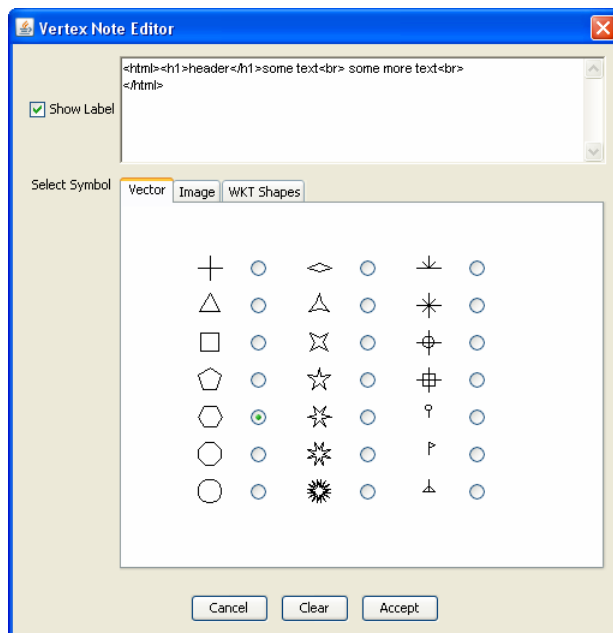
Scope: Apply the labels to Points, Lines and/or Polygons in the layer (check as required). This allows the selection of which type of features in a layer will have labels attached. Labels are only attached to vertices, i.e. points, and the ends of line segments for lines and polygons.

The following image shows some example vertex symbols and their labels where specified and enabled.



Vertex Note Plugin

When a single feature is selected, then the Vertex Note option from the Plugins menu will launch the Vertex Note Editor:



The steps required are:

1. Make the selected layer editable.
2. Select the required feature using the selection tool from the Editor Toolbox.
3. Select the Vertex Note option from the Plugins menu, or the Vertex Note Icon from the menu bar.

This dialog has three fields:

- The Show Label checkbox: if checked then the label will be shown on this feature, otherwise the label is not displayed. This allows labels to be selectively displayed on individual features in the currently selected layer.
- The text field shows the value of the attribute (as selected in the VertexSymbols plugin dialog). The user can edit/replace the text in this field. Care is required if editing other than String type attributed (currently no type mismatches are checked). When accepted, the value of the attribute is updated for the selected feature.
- A tabbed pane from which a special symbol can be selected for the currently selected feature. This will override the default symbol for the current layer.

When this dialog is launched for a selected feature for the first time, two new attributes, named “ShowLabel” and “SymbolName” are added to the schema for the current layer. The “ShowLabel” attribute is of type INTEGER and will have a value of 0 or 1 depending on whether the label for this feature will be displayed (value = 1). The “SymbolName” attribute is of type STRING and will contain the name of the special symbol to be used for this feature. These attributes are saved with the schema for the layer when the layer is saved.

The symbols are the same group that can be selected in the Vertex Symbols Dialog. The properties, size and orientation of the symbols is always that for the layer.

Symbol Scaling

The sizing of vertex symbols (both vector and image based) requires some care. Symbols with a size 16x16 (screen pixels) generally appear at a reasonable size on the screen. High quality printing, however, requires higher resolution image type symbols, perhaps 4 times higher.

Vector symbols are created dynamically as required using the hard-coded vector specification. The size of these symbols is specified in the Dialog, and they will print at the resolution of the printer.

WKT symbols have a natural size (their extent) that is used to display the symbol in the Dialog. The symbol is scaled to the required size when displayed on a map layer.

Image symbols require special attention. Images are normally shown at full size, but can be scaled down in size to improve their quality for printing. Images of about 16x16 pixels will give quite good screen resolution, and smaller ones could be used where needed.

For high quality printing, higher quality images are often required. They should be at least 64x64 pixels for quality printing (to print at the correct relative physical size).

By default all SVG images are scaled to a 64x64 image for internal use. At this size, the resulting image is generally good enough for high quality printing of vertex symbols.

Image scaling provides some help in resolving these issues:

- Bit-mapped images (gif, jpg and png) have a native size. These can be scaled down in size for screen display while allowing their higher resolution to be available for printing.
- SVG images are defined by a vector specification. These are converted to an internal bit-mapped image, and the size of this image can be user controlled, and then further reduced in size for screen display as required.

Primary image scaling is specified by the user when preparing the images. The scaling specification is included in the name of the image file. This is used for displaying the image in the Vertex Symbol Dialog.

For example an image named “image_x4.gif” will be linearly scaled by a factor of 4 (times smaller) for display in the Vertex Symbols Dialog. The scaling factor is defined by adding “_x4” to the file name just before the extension part; i.e. an underscore character followed by an ‘x’ followed by the required scaling factor as one or more digits.

The actual size of the image when placed on a vertex in a map layer will be that specified by the Size field in the Vertex Symbols Dialog.

In the case of SVG images, the SVG specification is converted into a bit-map for internal use. The default size of this bit-map is 64x64. If a higher quality image is required it is possible add an image size code to the file name.

For example an SVG named “image_x4x256.svg” will be converted to an image of 256x256 pixels, then scaled for display in the Vertex Symbols Dialog by a factor of 4 (time smaller). The 256x256 size is retained, and is available for printing.

Interaction With OpenJUMP Change Styles Dialog

By default OpenJUMP offers square shaped vertex symbols. Once a new Vertex Symbol type is added using this plugin, the new symbol will appear in the Change Styles Dialog. The display of vertex symbols can be enabled/disabled from this dialog, and their size (for vector images only) can also be changed (though this dialog limits the size to 20 pixels).

Note that if you have a vertex size greater than 20, and you open the Change Vertex Styles Dialog, the size will be set back to 20. You will need to use the Vertex Symbols Dialog to reset the vertex size. This bug requires a fix in the Jump core code.

Once a new vertex symbol is added to a layer, then it is not possible to revert to the built-in vertex symbol, but an equivalent square shaped symbol is available from the plugin dialog.

Some Issues

The specification of a vertex symbol used in a layer is retained in the project file when the project is saved. It is therefore assumed that when the project is re-loaded the actual images used in the project are available (i.e. they are to be found in the VertexImages folder). If an image is not found (it has been removed, or just does not exist) then the missing images will be displayed with the message “No Image” wherever the image is required.

Projects that use this plugin can have SVG files saved as normal for image capture. The Cadplan JumpPrinter plugin has been updated to accommodate the need for high quality printing for these new vertex symbol options. While the VertexSymbols plugin can be installed by itself, the JumpPrinter plugin (vers 1.30 and later) requires the VertexSymbols plugin to be concurrently installed.

The facility to load SVG images as vertex symbols is currently under development and may change in the future.

This version of the plugin is not compatible with previous versions as changes have been made to the vertex symbol style classes. An error (xml2java) will be generated if an older project file is opened with this new plugin version installed. The work-arounds are:

1. Rebuild the project file again from scratch using the same set of layers, then reset all the layer properties.
2. Manually edit the project file to replace all referenced to PolygonVertexStyle, StartVertexStyle, AnyShapeVertexStyle, ImageVertexStyle and WKTVertexStyle with a SquareVertexStyle (with some default settings). This will allow the project file to be loaded, then reset the vertex symbols.

The HTML text format option can include any valid HTML tag that is supported by the Java JEditorPane, but does not recognize hyperlinks. This is because the HTML page is converted to an image for display in the map layer.