

JumpFillPattern

User Guide

Vers 0.2

by

Geoffrey G Roy
<http://www.cadpan.com.au>

Introduction

JumpFillPattern is a plugin for OpenJump that allows users to create and use their own fill patterns. There are two types of patterns:

- Vector-based patterns that are specified using the WKT (Well Known Text) specification. Details of WKT can be found at http://en.wikipedia.org/wiki/Well-known_text, or in a number of places in the Jump/JTS documentation.
- Images that form the basis of the pattern. Image patterns can be created from GIF, JPEG, PNG or SVG image files.

In both cases the pattern is based on a square or rectangular tile that is tessellated over the shape to be filled.

Installation

The JumpFillPattern.jar file is placed in the extensions directory (ext) of the OpenJump installation. The patterns are placed in a folder (inside the ext folder) named “FillPatterns”. All files that have an extension “.gif”, “.jpg” or “.jpeg”, “.png”, “.svg” or “.wkt” are assumed to be patterns that will be loaded when the plugin is initialized.

The JumpFillPattern plugin requires the Batik libraries to be installed and the Classpath for OpenJump adjusted to include them if necessary. These are available from Apache at: <http://xmlgraphics.apache.org/batik/download.cgi>

Operation

The plugin is loaded when OpenJump is launched. The new patterns will appear in the standard OpenJump Change Styles dialog when a fill pattern is selected for the current layer. The new styles are added to the bottom of the list in the combobox.

Specifying Patterns

Vector Patterns

Vector-based patterns are defined using WKT, and are included in text files with a “.wkt” file name extension. Each file contains just one pattern, and has the format:

```
<Line width>:<Extent size>:<WKT Specification>
```

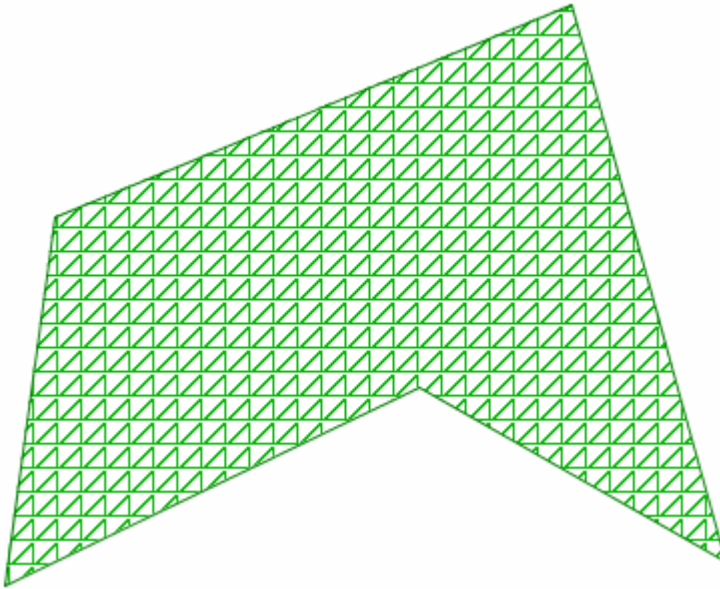
The Line width is the width of the lines used as an integer (1 is a thin line). Extent size is the size of a square tile in which the pattern is created (number of screen pixels).

The pattern is specified in WKT with the centre of the tile as the local origin. Any valid WKT specification can be used.

For example:

```
1:12:LINESTRING(5 5, 5 -5, -5 -5, 5 5)
```

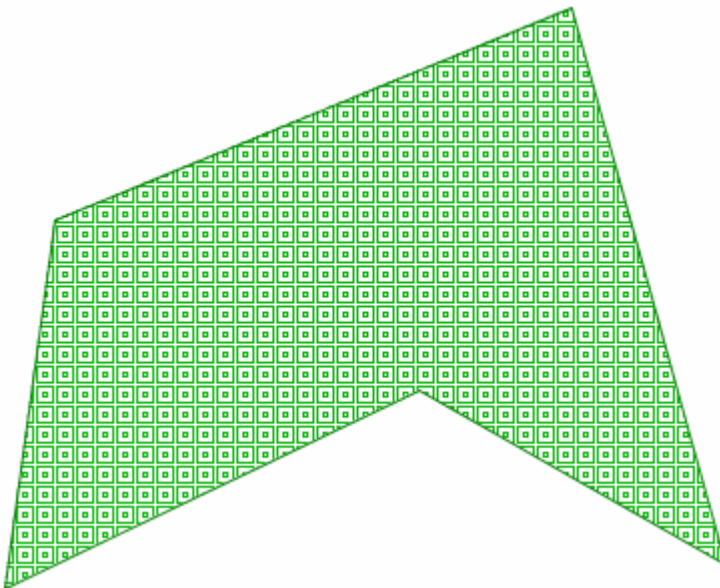
defines a triangle shape inside a 12x12 cell that looks like this:



This WKT specification:

```
1:10:POLYGON((4 4, 4 -4, -4 -4, -4 4, 4 4),(1 1, 1 -1, -1 -1, -1 1, 1 1))
```

defines two concentric squares of side 2 and 8 inside a 10x10 cell like this:

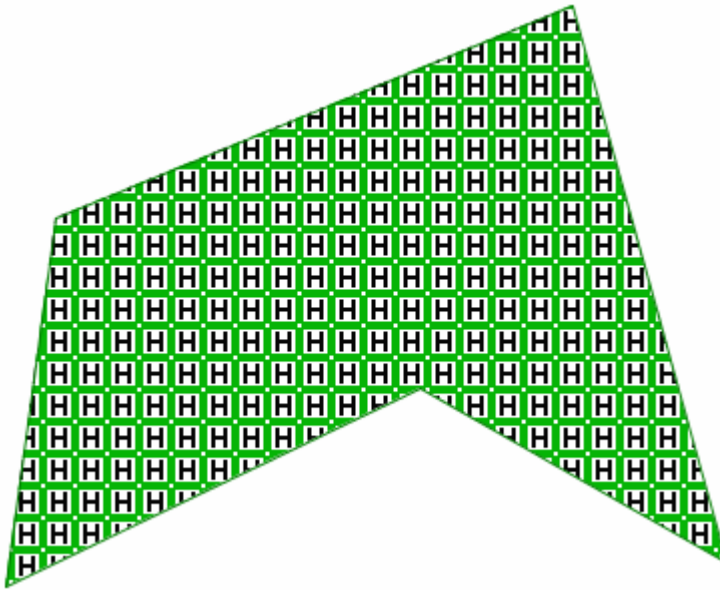


There is no real limit to the patterns that can be formed in this way. The text file can be flexibly laid out – over multiple lines as necessary; the two colons delineated the three elements.

Image Patterns

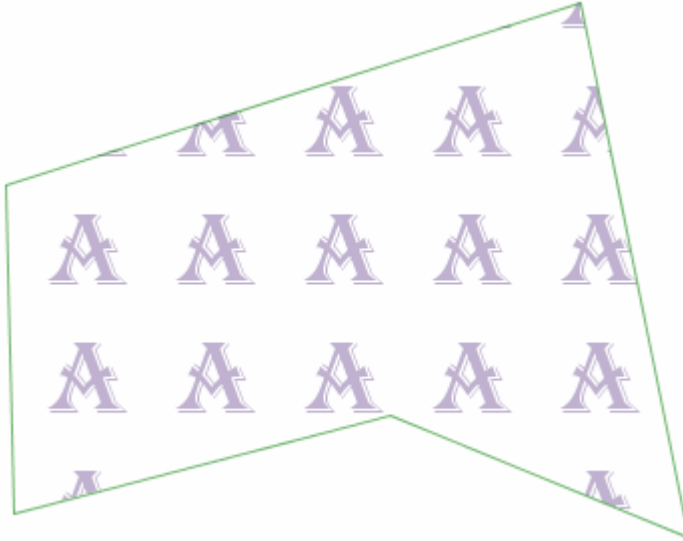
Images can also be used to create patterns. The raw image size defines the size of the tile (it can be square or rectangular) that will be tessellated over the shape to be filled. This applies to gif, jpg and png image files.

Here is an example of a 16x16 “Hospital” icon used as a pattern



Patterns can also be built from SVG images. In this case the SVG image is converted to a square bitmap for internal use. By default this will be 32x32 pixels. If another tile size is required then this is specified in the name of the SVG image file. For example a file named “name_x64.svg” will cause a tile of 64x64 to be created.

Here is a sample pattern created from an SVG image file:



Issues

Large Patterns

If a large pattern tile is specified and it has lots of white space in it, it may appear “blank” in the Fill Pattern combo in the Change Styles Dialog (there may not be enough space to show part of the pattern), though it should appear correctly in the Preview panel.

Missing Images

If an image file is used in a project, then it is removed from the FillPatterns folder, then the fill pattern will be shown with the text “No Pattern” in the map.

Printing

When using the Cadplan JumpPrinter plugin, there is a problem with the Quality renderer option – the printing fails with an error. The other two renderer options (“Accurate” and “External”) appear to work correctly. There is no solution to this at the moment.

Pattern Numbering

Numbering of User defined Patterns in the Fill Patterns Combo of the Change Styles Dialog shows the index number of the pattern as “0” in each case. This numbering appears to be hard coded in the OpenJump core and does not allow for user added patterns outside the core. It does not appear to have any impacts other than the new fill patterns are not numbers.