

# FOP Design: SVG

Version 627324

## Table of contents

- 1 Introduction..... 2
- 2 Text Drawing.....2
- 3 PDF Links..... 2
- 4 Images.....2
- 5 PDF Transcoder.....3
- 6 Other Outputs..... 3

## 1 Introduction

---

SVG is rendered through Batik.

The XML from the XSL:FO document is converted into an SVG DOM with batik. This DOM is then set as the Document on the Foreign Object area in the Area Tree.

This DOM is then available to be rendered by the renderer.

SVG is rendered in the renderers via an XMLHandler in the FOUserAgent. This XML handler is used to render the SVG. The SVG is rendered by using batik. Batik converts the SVG DOM into an internal structure that can be drawn into a Graphics2D. So for PDF we use a PDFGraphics2D to draw into.

This creates the necessary PDF information to create the SVG image in the PDF document.

Most of the work is done in the PDFGraphics2D class. There are also a few bridges that are plugged into batik to provide different behaviour for some SVG elements.

## 2 Text Drawing

---

Normally batik converts text into a set of curved shapes.

This is handled as any other shapes when rendering to the output. This is not always desirable as the shapes have very fine curves. This can cause the output to look a bit bad in PDF and PS (it can be drawn properly but is not by default). These curves also require much more data than the original text.

To handle this there is a PDFTextElementBridge that is set when using the bridge in batik. If the text is simple enough for the text to be drawn in the PDF as with all other text then this sets the TextPainter to use the PDFTextPainter. This inserts the text directly into the PDF using the drawString method on the PDFGraphics2D.

Text is considered simple if the font is available, the font size is useable and there are no tspan or other complications. This can make the resulting PDF significantly smaller.

## 3 PDF Links

---

To support links in PDF another batik element bridge is used. The PDFElementBridge creates a PDFANode which inserts a link into the PDF document via the PDFGraphics2D.

Since links are positioned on the page without any transforms then we need to transform the coordinates of the link area so that they match the current position of the a element area. This transform may also need to account for the svg being positioned on the page.

## 4 Images

---

Images are normally drawn into the PDFGraphics2D. This then creates a bitmap of the image data that can be inserted into the PDF document.

As PDF can support jpeg images then another element bridge is used so that the jpeg can be directly inserted into the PDF.

## 5 PDF Transcoder

---

Batik provides a mechanism to convert SVG into various formats. Through FOP we can convert an SVG document into a single paged PDF document. The page contains the SVG drawn as best as possible on the page. There is a `PDFDocumentGraphics2D` that creates a standalone PDF document with a single page. This is then drawn into by batik in the same way as with the `PDFGraphics2D`.

## 6 Other Outputs

---

When rendering to AWT the SVG is simply drawn onto the awt canvas using batik.

The PS Renderer uses a similar technique as the PDF Renderer.

The SVG Renderer simply embeds the SVG inside an `svg` element.