

## Embedding SVG



Some examples of how to embed an SVG image inside your FO documents

## Introduction

This document shows some examples of how an SVG document can be embedded directly in an fo document and how FOP will render it.

There are a couple of ways to specify the XML for embedding. There are a number of factors which specify exactly how the SVG image will be rendered, such as size, clipping etc.

Note: if you want to know about SVG itself you should consult the appropriate SVG resource.

The [Batik Project](#) is a good resource for information about SVG and samples etc.

## XML Syntax

Here we have some examples of how the XML can be specified in the fo document.

This first example specifies the name space on the instream-foreign-object using the xmlns:svg attribute. Then all the svg elements start with the svg: namespace prefix. It can be specified on any element above where the svg:svg tag appears (eg. fo:root).

```
<fo:instream-foreign-object xmlns:svg="http://www.w3.org/2000/svg"> <svg:svg width="20"
height="20"> <svg:g style="fill:red; stroke:#000000"> <svg:rect x="0" y="0" width="15"
height="15"/> <svg:rect x="5" y="5" width="15" height="15"/> </svg:g> </svg:svg> </
fo:instream-foreign-object>
```



This example specifies the name space on the svg:svg element using the xmlns:svg attribute. Then all the svg elements start with the svg: namespace prefix. It is able to determine the namespace for the current element.

```
<fo:instream-foreign-object> <svg:svg xmlns:svg="http://www.w3.org/2000/svg" width="20"
height="20"> <svg:g style="fill:red; stroke:#000000"> <svg:rect x="0" y="0" width="15"
height="15"/> <svg:rect x="5" y="5" width="15" height="15"/> </svg:g> </svg:svg> </
fo:instream-foreign-object>
```



This example specifies the namespace for the current element, rather than for the svg: prefix.

This means that all elements under this one are assumed to have the same namespace.

```
<fo:instream-foreign-object> <svg xmlns="http://www.w3.org/2000/svg" width="20" height="20">
<g style="fill:red; stroke:#000000"> <rect x="0" y="0" width="15" height="15"/> <rect x="5"
y="5" width="15" height="15"/> </g> </svg> </fo:instream-foreign-object>
```



## Sizing

The size of the instream-foreign-object is obtained in a number of ways.

Specify the size on the instream-foreign-object element.

```
<fo:instream-foreign-object content-width="20pt" content-height="20pt"> <svg:svg> <svg:g
style="fill:red; stroke:#000000"> <svg:rect x="0" y="0" width="15" height="15"/> <svg:rect
x="5" y="5" width="15" height="15"/> </svg:g> </svg:svg> </fo:instream-foreign-object>
```



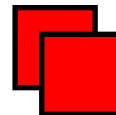
Specify the size on the svg document element.

```
<fo:instream-foreign-object> <svg:svg width="20pt" height="20pt"> <svg:g style="fill:red;
stroke:#000000"> <svg:rect x="0" y="0" width="15" height="15"/> <svg:rect x="5" y="5"
width="15" height="15"/> </svg:g> </svg:svg> </fo:instream-foreign-object>
```



Specify the size on the svg document element and for the content size on the instream-foreign-object.

```
<fo:instream-foreign-object content-width="40pt" content-height="40pt"> <svg:svg width="20pt"
height="20pt"> <svg:g style="fill:red; stroke:#000000"> <svg:rect x="0" y="0" width="15"
height="15"/> <svg:rect x="5" y="5" width="15" height="15"/> </svg:g> </svg:svg> </
fo:instream-foreign-object>
```



## Clipping

## Interesting Functionality