

# The SVG Serializer (2.1 legacy document)

## Table of contents

1 Comments.....5

## Table of contents

1 SVG Serializer.....	3
1.1 Introduction.....	3
1.2 Usage.....	3
1.2.1 Basic Example.....	3
1.2.2 Advanced Example.....	4

**Warning:**

This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

## 1. SVG Serializer

### 1.1. Introduction

The SVG Serializer is an advanced serializer which accepts valid Scalable Vector Graphic documents (currently to the 2000-08-02 Candidate Recommendation specification) and renders it to an image which is served just like any other document in Cocoon.

Why would you want to do this? Well, charts can be produced from the same data which generates tables, graphical images with text labels all following a standard theme can be generated or normal pages can be beautified.

For examples of this serializer, see the Cocoon welcome page in the distribution ([cocoon root]/welcome).

So how does this serializer work?

1. Parse and validate SVG document
2. Call Batik's Transcoder to encode this image as an image file, and return it to the user.

The SVG xml document must have a root element `<svg>`, using xml namespace it is recommended to use the prefix *em*, and the namespace uri *http://www.w3.org/2000/svg*.

### 1.2. Usage

The best way to explain how this serializer works is to show some examples.

#### 1.2.1. Basic Example

This is a basic example of the serializer.

```
<map:serializers> <map:serializer> <map:serializer name="svg2jpeg"
mime-type="image/jpeg" src="org.apache.cocoon.serialization.SVGSerializer"> <parameter
name="transcoder" value="org.apache.batik.transcoder.image.JPEGTranscoder"/>
</map:serializer> </map:serializers> ... <map:pipeline> <map:match pattern="sample.jpeg">
<map:generate type="file" src="sample.svg"/> <map:serialize type="svg2jpeg"/>
</map:match> </map:pipeline>
```

When the resource `sample.jpeg` is requested, a SAX event stream is generated from the file `sample.svg`, which is serialized using the `svg2jpeg` serializer. This serializer is configured to use a specific transcoder. The MIME type is specified so that Cocoon can tell the client which type the document is. It can be seen that in general the use of this serializer is identical to that of the other serializers.

The parameter *transcoder* selects explicitly a batik transcoder. You may want to rely on the default *mime-type* to transcoder association, omitting the transcoder parameter

mime-type	transcoder-class
image/jpeg	org.apache.batik.transcoder.image.JPEGTranscoder
image/jpg	org.apache.batik.transcoder.image.JPEGTranscoder
image/png	org.apache.batik.transcoder.image.PNGTranscoder

image/tiff	org.apache.batik.transcoder.image.TIFFTranscoder
------------	--

### 1.2.2. Advanced Example

This is a more advanced sample of using the SVG Serializer.

```
<map:serializers> <map:serializer> <map:serializer name="svg2jpeg"
mime-type="image/jpeg" src="org.apache.cocoon.serialization.SVGSerializer"> <parameter
name="transcoder" value="org.apache.batik.transcoder.image.JPEGTranscoder"/>
<parameter name="background_color" type="color" value="#00FF00"/> </map:serializer>
<map:serializers> ... <map:pipeline> <map:match pattern="sample.jpeg"> <map:generate
type="file" src="sample.svg"/> <map:serialize type="svg2jpeg"/> </map:match>
</map:pipeline>
```

In this example another parameter is given to the serializer, `background_color`. This parameter is passed to the transcoder. The type argument specifies the type of data to convert the value to. In this example the string `"#00FF00"` is converted to a `Color` object, which is passed to the transcoder as the background colour to use.

For a list of the parameters available for each transcoder, refer to the Batik API docs.

rossb@apache.org: Create a document summarising the transcoder hints

The following table summarizes most useful general SVG ImageTranscoder parameters, all of these parameters are mapped to Batik's SVG ImageTranscoder hints.

width	float	Specifies the width of the rasterized image explicitly. If no height is specified the aspect ratio is kept.
height	float	Specifies the width of the rasterized image explicitly. If no width is specified the aspect ration is kept.
background_color	color	Defines the background color to use for opaque image formats, or the background color that may be used for image formats that support alpha channel. A color value of format RRGGBB, or #RRGGBB sets the background color of the rasterized image, by default the background color is white.
language	string	to set the default language to use (may be used by a <switch> SVG element for example), by default language is set to en.
user_stylesheet_ur	string	to fix the URI of a user stylesheet
pixel_to_mm	float	to specify the pixel to millimeter conversion factor, by default its value is 0.264583 (96dpi).

For this to work reliably with any transcoder, some magic must be done.

1. First, the parameter name is transformed to upper-case and then "KEY\_" is prepended. This is to match the internal naming scheme of the hints in the Batik Transcoder interfaces.
2. This name is then looked up via Reflection to ensure it is a valid parameter on the specified transcoder.
3. Then the value is converted to the type specified in the type attribute (currently supported types are string, float, integer, boolean and color) and passed to the transcoder.

## **1. Comments**

add your comments