Use case: Generating an SVG DOM Document

by Jeremias Märki

Table of contents

1. Introduction

This page describes the generation of an SVG barcode in Java.

2. Preparing the barcode generator configuration

Barcode4J depends on the Apache Avalon Framework for configuration. So the first step is to prepare an `org.apache.avalon.framework.configuration.Configuration` object. There are several ways to obtain it:

- In an Avalon-based system you will probably already have `Configuration` objects around.
- You can build it from an XML file:

```
import org.apache.avalon.framework.configuration.Configuration;
import
org.apache.avalon.framework.configuration.DefaultConfigurationBuilder;

[..]

    DefaultConfigurationBuilder builder = new
DefaultConfigurationBuilder();
    File cfgFile = new File("C:\Temp\barcode.cfg");
    Configuration cfg = builder.buildFromFile(cfgFile);
```

- You can build it from other XML sources such as DOM and SAX events. For more information see the Javadocs from Apache Avalon Framework (look for DefaultConfigurationBuilder and SAXConfigurationHandler).
- You can build the Configuration object tree by hand using `org.apache.avalon.framework.configuration.DefaultConfiguration`.

3. Using the BarcodeUtil class to generate a DOM DocumentFragment

The easiest way to obtain the barcode in SVG as a DocumentFragment is to use `org.krysalis.barcode4j.BarcodeUtil`. See the following example:

```
import org.w3c.dom.DocumentFragment;

[..]
```

```
String myMessage = "012345679";
DocumentFragment frag = BarcodeUtil.getInstance()
        .generateBarcode(cfg, myMessage);
```

4. Serializing the barcode

Obviously, you can do everything you want with the SVG graphic in that DocumentFragment we got. One thing you might want to do is serializing it to an SVG file, for example.

```
TransformerFactory factory = TransformerFactory.newInstance();
Transformer trans = factory.newTransformer();
Source src = new DOMSource(frag);
Result res = new StreamResult(new File("C:\Temp\myBarcode.svg"));
trans.transform(src, res);
```