Instructions for the Apache FOP extension

by Jeremias Märki

Table of contents

1 Introduction

This page describes how to generate barcodes within an XSL-FO document with that is processed with Apache FOP.

2 Setting up the barcode extension with Apache FOP 0.93 or later

To setup the barcode extension, do the following:

- Add `barcode4j.jar` and `barcode4j-fop-ext.jar` to the classpath.
- Alternatively, you can use the combined JAR: `barcode4j-fop-ext-complete.jar` which combines both **Barcode4J** and the FOP extension.

3 Setting up the barcode extension with Apache FOP 0.20.5

To setup the barcode extension, do the following:

- Add `barcode4j.jar` and `barcode4j-fop-ext-0.20.5.jar` to the classpath.
- Alternatively, you can use the combined JAR: `barcode4j-fop-ext-0.20.5-complete.jar` which combines both **Barcode4J** and the FOP extension.
- **Important:** When you use FOP 0.20.5 together with **Barcode4J**, make sure you use the version of Avalon Framework that comes with **Barcode4J** (or the latest release of Avalon Framework). The version from FOP 0.20.5 is slightly out-of-date.

4 Using the barcode extension for Apache FOP

This package contains an extension for Apache FOP for generating barcodes.

The barcode extension is tied to the follwing namespace: **`http://barcode4j.krysalis.org/ns`**

To create a barcode the Barcode XML can be used within fo:instream-foreign-object elements.

Here's an example (XSL-FO snippet):

```
<fo:block>
  <fo:instream-foreign-object>
    <barcode:barcode
          xmlns:barcode="http://barcode4j.krysalis.org/ns"
          message="my message" orientation="90">
      <barcode:code128>
        <barcode:height>8mm</barcode:height>
      </barcode:code128>
    </barcode:barcode>
  </fo:instream-foreign-object>
</fo:block>
```

The attribute "orientation" specifies the orientation of the barcode in degrees. The attribute is optional and if set must be one of the following values: 0, 90, -90, 180, -180, 270, -270.

Some characters like RS (record separator) and GS (group separator) are sometimes required for certain applications but these characters are not valid XML characters and cannot be encoded. You can escape them by using Java's Unicode escaping syntax. To escape RS, you can write "\u001E". GS would be encoded as "\u001D" and so forth. If for some reason, you need to encode a string containing "\u" in

your message, you will have to escape that using "\\u", i.e. the double backslash will be converted to a single backslash and the subsequent "u" is treated as a normal message character. It is an error to specify an incomplete escape sequence, i.e. "00\u012xsomething" will result in an error.

5 Including FOP's page-number in Barcode Output

FOP's page-number can be incorporated into barcode output. To incorporate the page-number into the barcode, insert `#page-number#` into your barcode message. For example, to insert a page number between `12345-` and `-6789` do the following:

```
<barcode:barcode
    xmlns:barcode="http://barcode4j.krysalis.org/ns"
    message="12345-#page-number#-6789">
  <code39/>
</barcode:barcode>
```

You can also use #page-number:{number-format}# where "{number-format}" is the number format pattern supported by Java's java.text.DecimalFormat. This allows you to format page "`3`" as "`0003`" if you need a constant number of characters in the barcode message. For this example, you would use "#page-number:0000#" in the message.

> **Note:**
> This feature is exclusively available in the FOP extension. It does not work in the XSLT extensions for Xalan-J or SAXON because only FOP has the information about the current page number. The XSLT stage runs before any layout is processed.

6 Using FOP from the command-line

If you use FOP from the command-line (by calling fop.bat on Windows) it may not be enough to simply add the JAR files mentioned above to FOP's lib directory. On Unix all JAR files are automatically added to the classpath through the fop.sh script, so you only have to make sure that you don't have two different avalon-framework.jar files the the lib directory. On Windows, you will have to edit fop.bat and manually edit the entries there.

7 Some technical background

If you wonder how the FOP extension is found by FOP just by adding it to the classpath and by using the right namespace, open the extension JAR in a ZIP Viewer and look at the file in the directory META-INF\services\. FOP will look for this file at startup and will recognize the ElementMapping class.

The normal FOP extension described above internally creates the barcode in SVG format. The generated SVG image is then converted by Batik to the target format (PDF, PostScript or whatever).

The latest FOP versions even allow Barcode4J to generate the barcode in the native output format.For example, when PostScript output is selected in FOP, Barcode4J creates the barcode as an EPS graphic which is generally much faster than taking a detour via SVG.