

# Notes on the supported output formats

by Jeremias Märki

## Table of contents

|                                              |   |
|----------------------------------------------|---|
| 1 Introduction.....                          | 2 |
| 2 SVG.....                                   | 2 |
| 3 EPS (Encapsulated PostScript).....         | 2 |
| 4 Bitmap formats (PNG, TIFF, JPEG etc.)..... | 2 |
| 5 Java2D (AWT).....                          | 3 |

## 1 Introduction

This page gives information about the various output formats supported by **Barcode4J** and tells you how to implement additional output formats.

For information about accessing the various output formats from Java please refer to the [Java embedding page](#).

## 2 SVG

[SVG](#) was the first output format supported by this package. At this time, no verification has been performed to ensure that the output generated conforms to every SVG variant (especially SVG profiles basic and tiny). The code was tested with Apache Batik and Adobe SVG Viewer, only. If there are any problems with the output generated we'd appreciate to hear about it.

The SVG output is configurable to generate namespaced (with and without prefix) and non-namespaced SVG content. The generated document does not contain any schema or DTD declarations.

Depending on your needs on the Java side there is support for generating SVG as a W3C DOM or a JDOM. It is very easy to add support for other output methods such as XOM.

## 3 EPS (Encapsulated PostScript)

The EPS output format generates Level 1 PostScript and includes a `HiResBoundingBox` for exact positioning in the target document. The font used in the barcode is expected to be present in the target PostScript interpreter.

### Note:

To minimise file size and for performance reasons, the generated EPS files do not contain a bitmap preview. So if you import the EPS file into an application (such as Visio) you may not see the barcode.

## 4 Bitmap formats (PNG, TIFF, JPEG etc.)

Bitmap formats are supported through the Java2D output support. What bitmap formats are supported depends on your environment:

- JDK 1.3.x only provides support for JPEG output by using the Sun JPEG encoder included in the JDK.
- JDK 1.4 or later provides the Image I/O API which contains support for JPEG, PNG by default. Additional formats (such as TIFF or BMP) are supported if [JAI \(Java Advanced Imaging\)](#) is present in the classpath (Download the "Image I/O Tools"). Support for these formats is enabled automatically.
- Additional output formats can be supported by adding ImageIO-compatible codecs or by registering additional implementations of the `BitmapEncoder` interface provided by **Barcode4J**. For example, a `BitmapEncoder` implementation could be coded to interface with the older JIMI imaging library available from Sun.

By default, bitmaps are generated as grayscale images (8 bit). You can choose to enable or disable anti-aliasing for grayscale images. You can also choose to generate monochrome images (1 bit). Additionally, you can specify the resolution for the bitmap.

**Note:**

Although JPEG output is supported, it is NOT recommended as it is a lossy compression format and contrast will suffer and thus degrading the readability of the generated barcode.

## 5 Java2D (AWT)

The Java2D implementation can be used to embed **Barcode4J** in a Java application (such as a report engine, for example). It is also used to paint barcodes on bitmap images.