

# Embedding in a Java application (using the JavaBean API)

by Jeremias Märki

## Table of contents

1 Introduction.....	2
2 Basic steps.....	2
3 Creating a barcode bean.....	2
4 Configuring the bean.....	2
5 Creating a CanvasProvider and generating the barcode.....	2
6 A complete example.....	3

## 1 Introduction

Normally, you will configure the barcode generators using XML as described on the [Barcode XML page](#). Some people prefer using JavaBeans instead of XML. Here's how to use that API.

### Note:

If you work with the Bean API you don't need avalon-framework.jar in your classpath.

## 2 Basic steps

The steps necessary to create barcodes using JavaBeans is similar to the steps using XML:

1. Create a barcode bean
2. Set the desired values to configure the barcode generator
3. Create a CanvasProvider (depending on the output format)
4. Finally generate the barcode

## 3 Creating a barcode bean

As the first step you have to instantiate a barcode bean.

```
Code39Bean bean = new Code39Bean();
```

Here's a list of available bean classes:

- org.krysalis.barcode4j.impl.codabar.CodabarBean
- org.krysalis.barcode4j.impl.code128.Code128Bean
- org.krysalis.barcode4j.impl.code39.Code39Bean
- org.krysalis.barcode4j.impl.int2of5.Interleaved2Of5Bean
- org.krysalis.barcode4j.impl.postnet.POSTNETBean
- org.krysalis.barcode4j.impl.upcean.EAN13Bean
- org.krysalis.barcode4j.impl.upcean.EAN8Bean
- org.krysalis.barcode4j.impl.upcean.UPCABean
- org.krysalis.barcode4j.impl.upcean.UPCEBean

## 4 Configuring the bean

Each bean has specific getter and setter methods to control various aspects of the individual implementations. Please refer to the [JavaDocs](#) for the available properties.

An example:

```
bean.setChecksumMode(ChecksumMode.CP_CHECK);
bean.setWideFactor(3);
```

## 5 Creating a CanvasProvider and generating the barcode

From here it's the same steps as with XML configuration. Please refer to the [primary embedding page](#).

## 6 A complete example

```
//Create the barcode bean
Code39Bean bean = new Code39Bean();

final int dpi = 150;

//Configure the barcode generator
bean.setModuleWidth(UnitConv.in2mm(1.0f / dpi)); //makes the narrow bar
                                                    //width exactly one pixel
bean.setWideFactor(3);
bean.doQuietZone(false);

//Open output file
File outputFile = new File("out.png");
OutputStream out = new FileOutputStream(outputFile);
try {
    //Set up the canvas provider for monochrome PNG output
    BitmapCanvasProvider canvas = new BitmapCanvasProvider(
        out, "image/x-png", dpi, BufferedImage.TYPE_BYTE_BINARY, false, 0);

    //Generate the barcode
    bean.generateBarcode(canvas, "123456");

    //Signal end of generation
    canvas.finish();
} finally {
    out.close();
}
```