Table of Contents

1 Introduction

MW6 Barcode Java Library supports over 100 different barcodes, including Code 39, Code 128, GS1-128, EAN 13, EAN 8, UPC-A, UPC-E, Royal Mail 4 State, USPS OneCode, Deutsche Post Identcode, Deutsche Post Leitcode, Japan Postal Code, PDF417, Micro PDF417, DataMatrix, MaxiCode, Aztec, QRCode, Micro QRCode, CodaBlock-F, Code 16K and Code 49.

Our Java library can be used in J2SE, J2EE and Java Reporting environments (e.g. Jasper Reports, iReport, Eclipse BIRT, and ORACLE Reports).

The library jar file is "mw6barcode.jar", make sure that you include this jar file in your project "Classpath" property.

2 Tutorials

2.1 Sample Java Application

2.1.1 MainApp.java

```
import java.awt.*;

public class MainApp extends Frame {

    public MainApp() {
        super("");
        setSize(300, 300);

        // create a barcode canvas object
        BarcodeCanvas bc = new BarcodeCanvas(1);
        add(bc);
    }

    public static void main(String[] args) {
        MainApp app = new MainApp();
        app.setVisible(true);
    }
}
```

2.1.2 BarcodeCanvas.java

```
import java.awt.*;
import java.io.*;

public class BarcodeCanvas extends Canvas {

    private com.mw6.barcode.Barcode myBC = new com.mw6.barcode.Barcode();
    private com.mw6.barcode.PDF417 myPDF417= new com.mw6.barcode.PDF417();
    private com.mw6.barcode.DataMatrix myDM = new
com.mw6.barcode.DataMatrix();
    private com.mw6.barcode.Aztec myAztec = new com.mw6.barcode.Aztec();
    private com.mw6.barcode.QRCode myQRCode = new com.mw6.barcode.QRCode();
    private com.mw6.barcode.MaxiCode myMC = new com.mw6.barcode.MaxiCode();
```

```
    private byte[] imageData = new byte[1000];
    private int[] sizeArray = new int[2];
    private int bCategory;

    public BarcodeCanvas(int barcodeCategory)
    {
        super();

        bCategory = barcodeCategory;

        try {
            if (barcodeCategory == 1)
            {
                // General Barcode
                myBC.setData("012349");
                myBC.setNarrowBarWidth(0.04);
                myBC.setBarHeight(2.0);
                myBC.setTextFont(new java.awt.Font("Helvetica",
java.awt.Font.PLAIN, 16));
                myBC.setRotation(com.mw6.barcode.Barcode.ORIENTATION_0);
                myBC.setResolution(96);
                myBC.setCheckDigit(false);
                myBC.setCheckDigitToText(true);
                myBC.setShowText(true);
                myBC.setBorderWidth(0.0);

                imageData = myBC.exportImageStream("png");
            }
            else if (barcodeCategory == 2)
            {
                // PDF417
                myPDF417.setNarrowBarWidth(0.08);
                myPDF417.setData("Your String");
                myPDF417.setRotation(com.mw6.barcode.PDF417.ORIENTATION_0);
                myPDF417.setTruncateSymbol(false);
                myPDF417.setColumns((short)3);
                myPDF417.setHandleTilde(true);
                myPDF417.getActualSize(sizeArray);

                imageData = myPDF417.exportImageStream("png");
            }
            else if (barcodeCategory == 3)
            {
                // DataMatrix
                myDM.setModuleSize(0.07);
                myDM.setData("Your String");
                myDM.setResolution(96);
                myDM.setMode(com.mw6.barcode.DataMatrix.MODE_ASCII);
                myDM.setPreferredFormat
(com.mw6.barcode.DataMatrix.FORMAT_48X48);
                myDM.setRotation(com.mw6.barcode.DataMatrix.ORIENTATION_0);
                myDM.setHandleTilde(true);
```

```
                myDM.getActualSize(sizeArray);

                imageData = myDM.exportImageStream("png");
            }
            else if (barcodeCategory == 4)
            {
                // Aztec
                myAztec.setPreferredFormat
(com.mw6.barcode.Aztec.FORMAT_67X67);
                myAztec.setModuleSize(0.12);
                myAztec.setData("Your String");
                myAztec.setResolution(96);
                myAztec.setRotation(com.mw6.barcode.Aztec.ORIENTATION_0);
                myAztec.setHandleTilde(true);
                myAztec.getActualSize(sizeArray);

                imageData = myAztec.exportImageStream("png");
            }
            else if (barcodeCategory == 5)
            {
                // QRCode
                myQRCode.setVersion(com.mw6.barcode.QRCode.VERSION_2);
                myQRCode.setModuleSize(0.10);
                myQRCode.setData("Your String");
                myQRCode.setResolution(96);
                myQRCode.setRotation(com.mw6.barcode.QRCode.ORIENTATION_0);
                myQRCode.getActualSize(sizeArray);

                imageData = myQRCode.exportImageStream("png");
            }
            else if (barcodeCategory == 6)
            {
                // MaxiCode
                myMC.setMode(com.mw6.barcode.MaxiCode.MODE_2);
                myMC.setData("[)
>~d03001~d0299615238~d029840~d029001~d029AIM, Inc~d029634 Alpha
Drive~d029Pittsburgh~d029PA~d030~d004");
                myMC.setHandleTilde(true);
                myMC.setResolution(96);
                myMC.setRotation(com.mw6.barcode.MaxiCode.ORIENTATION_0);
                myMC.setSize(400, 400);
                myMC.getActualSize(sizeArray);

                imageData = myMC.exportImageStream("png");
            }
        } catch (IOException e) {}

        // Set up canvas's size
        this.setSize(sizeArray[0], sizeArray[1]);
    }

    public void paint(Graphics g) {
```

```
        if (bCategory == 1)
            myBC.render(g, 0, 0);
        else if(bCategory == 2)
            myPDF417.render(g, 0, 0);
        else if(bCategory == 3)
            myDM.render(g, 0, 0);
        else if(bCategory == 4)
            myAztec.render(g, 0, 0);
        else if(bCategory == 5)
            myQRCode.render(g, 0, 0);
        else if(bCategory == 6)
            myMC.render(g, 0, 0);
    }

    public void saveImage(String fileName, String imgFormat) throws
IOException {

        if (bCategory == 1)
            myBC.saveAsImage(fileName, imgFormat);
        else if(bCategory == 2)
            myPDF417.saveAsImage(fileName, imgFormat);
        else if(bCategory == 3)
            myDM.saveAsImage(fileName, imgFormat);
        else if(bCategory == 4)
            myAztec.saveAsImage(fileName, imgFormat);
        else if(bCategory == 5)
            myQRCode.saveAsImage(fileName, imgFormat);
        else if(bCategory == 6)
            myMC.saveAsImage(fileName, imgFormat);
    }
}
```

2.2 Sample Java Servlet

```
<%
    response.setContentType("image/png");

    com.mw6.barcode.Barcode myBC = new com.mw6.barcode.Barcode();
    com.mw6.barcode.PDF417 myPDF417= new com.mw6.barcode.PDF417();
    com.mw6.barcode.DataMatrix myDM = new com.mw6.barcode.DataMatrix();
    com.mw6.barcode.Aztec myAztec = new com.mw6.barcode.Aztec();
    com.mw6.barcode.QRCode myQRCode = new com.mw6.barcode.QRCode();
    com.mw6.barcode.MaxiCode myMC = new com.mw6.barcode.MaxiCode();

    byte[] imageData = new byte[10000];
    int[] sizeArray = new int[2];
    int barcodeCategory = 1;

    // OutputStream
    javax.servlet.ServletOutputStream os = response.getOutputStream();

    if (barcodeCategory == 1)
```

```
        {
            // General barcode
            myBC.setData("012349");
            myBC.setNarrowBarWidth(0.04);
            myBC.setBarHeight(2.0);
            myBC.setTextFont(new java.awt.Font("Helvetica",
    java.awt.Font.PLAIN, 16));
            myBC.setRotation(com.mw6.barcode.Barcode.ORIENTATION_0);
            myBC.setResolution(96);
            myBC.setCheckDigit(false);
            myBC.setCheckDigitToText(true);
            myBC.setShowText(true);
            myBC.setBorderWidth(0.0);

            imageData = myBC.exportImageStream("png");
        }
        else if (barcodeCategory == 2)
        {
            // PDF417
            myPDF417.setNarrowBarWidth(0.08);
            myPDF417.setData("Your String");
            myPDF417.setRotation(com.mw6.barcode.PDF417.ORIENTATION_0);
            myPDF417.setTruncateSymbol(false);
            myPDF417.setColumns((short)3);
            myPDF417.setHandleTilde(true);

            imageData = myPDF417.exportImageStream("png");
        }
        else if (barcodeCategory == 3)
        {
            // DataMatrix
            myDM.setModuleSize(0.07);
            myDM.setData("Your String");
            myDM.setResolution(96);
            myDM.setMode(com.mw6.barcode.DataMatrix.MODE_ASCII);
            myDM.setPreferredFormat(com.mw6.barcode.DataMatrix.FORMAT_48X48);
            myDM.setRotation(com.mw6.barcode.DataMatrix.ORIENTATION_0);
            myDM.setHandleTilde(true);

            imageData = myDM.exportImageStream("png");
        }
        else if (barcodeCategory == 4)
        {
            // Aztec
            myAztec.setPreferredFormat(com.mw6.barcode.Aztec.FORMAT_67X67);
            myAztec.setModuleSize(0.12);
            myAztec.setData("Your String");
            myAztec.setResolution(96);
            myAztec.setRotation(com.mw6.barcode.Aztec.ORIENTATION_0);
            myAztec.setSize(500, 500);
            myAztec.setHandleTilde(true);

            imageData = myAztec.exportImageStream("png");
```

```
    }
    else if (barcodeCategory == 5)
    {
        // QRCode
        myQRCode.setVersion(com.mw6.barcode.QRCode.VERSION_2);
        myQRCode.setModuleSize(0.10);
        myQRCode.setData("Your String");
        myQRCode.setResolution(96);
        myQRCode.setRotation(com.mw6.barcode.QRCode.ORIENTATION_0);
        myQRCode.setSize(500, 500);

        imageData = myQRCode.exportImageStream("png");
    }
    else if (barcodeCategory == 6)
    {
        // Maxicode
        myMC.setMode(com.mw6.barcode.MaxiCode.MODE_2);
        myMC.setData("[)>~d03001~d0299615238~d029840~d029001~d029AIM,
Inc~d029634 Alpha Drive~d029Pittsburgh~d029PA~d030~d004");
        myMC.setHandleTilde(true);
        myMC.setResolution(96);
        myMC.setRotation(com.mw6.barcode.MaxiCode.ORIENTATION_0);
        myMC.setSize(400, 400);

        imageData = myMC.exportImageStream("png");
    }

    // Export barcode image
    os.write(imageData);
%>
```

2.3 iReport

2.3.1 ImageFactory Class

2.3.1.1 Methods

2.3.1.1.1 createAztec

Returns an Image object containing an Aztec barcode.

```
public static Image createAztec(String data, double moduleSize, int rotation, int resolution,
                                 int preferredFormat, boolean handleTilde, String imageFormat)
```

Parameters

data
Specifies a string which will be encoded by Aztec barcode.

moduleSize
Specifies the size (width/height) in centimeters of a square-shaped module.

rotation
Specifies the orientation of the barcode, a valid value should be 0, 1, 2, or 3. Please refer to Orientation Constants.

resolution
Specifies the resolution (DPI) of a device onto which a barcode will be rendered. The default value is 96 for computer screen.

preferredFormat
Specifies the preferred format for Aztec barcode, a valid value should be between 0 and 36. Please refer to Format Constants.

handleTilde
Specifies a boolean flag indicating whether or not it is required to process the tilde character "~". If it is set to TRUE, non-printable characters can be passed to a barcode using the tilde character, "**~dNNN**" represents an ASCII character encoded by the 3 digits NNN, for example, "~d010" represents the character LF (line feed).

imageFormat
Specifies barcode image format, it can be "bmp", "jpg", or "png".

2.3.1.1.2 createBarcode

Returns an Image object containing a barcode.

```
public static Image createBarcode(int barcodeType, String data, double narrowBarWidth, int rotation, int resolution,
                                   double barHeight, int fontSize, boolean showText, boolean addCheckDigit,
                                   boolean addCheckDigitToText, double borderWidth, String supplement,
                                   int supplementType, double supplementGap, String imageFormat)
```

Parameters

barcodeType
Specifies the barcode type, a valid value should be between 1 and 91. Please refer to Symbology Type Constants

data
Specifies a string which will be encoded by a barcode.

narrowBarWidth
Specifies the narrow bar width in centimeters.

moduleSize
Specifies the size (width/height) in centimeters of a square-shaped module.

rotation
Specifies the orientation of the barcode, a valid value should be 0, 1, 2, or 3. Please refer to Orientation Constants

resolution
Specifies the resolution (DPI) of a device onto which a barcode will be rendered. The default value is 96 for computer screen.

barHeight

Specifies the barcode height in centimeters. The default value is 1.5.

fontSize
Specifies the font size of the human readable text in the barcode.

showText
Sets a boolean flag indicating whether or not the human readable text should be displayed.

addCheckDigit
Sets a boolean flag indicating whether or not a check digit is required.

addCheckDigitToText
Sets a boolean flag indicating whether or not a check digit is added to the human readable text.

borderWidth
Specifies the border width in centimeters.

supplement
Specifies a supplement string for a UPC or EAN related barcode.

supplementType
Specifies the supplement type, a valid value should be 0, 1, or 2. Please refer to Supplement Type Constants

supplementGap
Specifies the distance, in centimeters, between the normal barcode and the supplement section.

imageFormat
Specifies barcode image format, it can be "bmp", "jpg", or "png".

2.3.1.1.3 createDataMatrix

Returns an Image object containing a DataMatrix barcode.

```
public static Image createDataMatrix(String data, double moduleSize, int rotation, int resolution,
                                      int mode, int preferredFormat, boolean handleTilde, String
imageFormat)
```

Parameters

data
Specifies a string which will be encoded by DataMatrix barcode.

moduleSize
Specifies the size (width/height) in centimeters of a square-shaped module.

rotation
Specifies the orientation of the barcode, a valid value should be 0, 1, 2, or 3. Please refer to Orientation Constants

resolution
Specifies the resolution (DPI) of a device onto which a barcode will be rendered. The default value is 96 for computer screen.

mode

Specifies the encoding mode for DataMatrix barcode, a valid value should be between 0 and 3. Please refer to Mode Constants.

preferredFormat
Specifies the preferred format for DataMatrix barcode, a valid value should be between 0 and 36. Please refer to Format Constants.

handleTilde
Specifies a boolean flag indicating whether or not it is required to process the tilde character "~". If it is set to TRUE, non-printable characters can be passed to a barcode using the tilde character, "**~dNNN**" represents an ASCII character encoded by the 3 digits NNN, for example, "~d010" represents the character LF (line feed).

imageFormat
Specifies barcode image format, it can be "bmp", "jpg", or "png".

2.3.1.1.4 createMaxiCode

Returns an Image object containing a MaxiCode barcode.

```
public static Image createMaxiCode(String data, int rotation, int resolution, String country, String serviceClass,
                                    String zipCode, int mode, boolean handleTilde, String imageFormat)
```

Parameters

data
Specifies a string which will be encoded by MaxiCode barcode.

rotation
Specifies the orientation of the barcode, a valid value should be 0, 1, 2, or 3. Please refer to Orientation Constants

resolution
Specifies the resolution (DPI) of a device onto which a barcode will be rendered. The default value is 96 for computer screen.

country
Specifies the country code of MaxiCode barcode.

serviceClass
Specifies the service class of MaxiCode barcode.

zipCode
Specifies the zip code of MaxiCode barcode.

mode
Specifies the encoding mode for Maxicode barcode, a valid value should be between 2 and 5. Please refer to Mode Constants.

handleTilde
Specifies a boolean flag indicating whether or not it is required to process the tilde character "~". If it is set to TRUE, non-printable characters can be passed to a barcode using the tilde character, "**~dNNN**" represents an ASCII character encoded by the 3 digits NNN, for example, "~d010" represents the

character LF (line feed).

imageFormat
Specifies barcode image format, it can be "bmp", "jpg", or "png".

2.3.1.1.5 createPDF417

Returns an Image object containing a PDF417 barcode.

```
public static Image createPDF417(String data, double narrowBarWidth, int rotation, int resolution,
                                 boolean truncateSymbol, int mode, int ecl, boolean handleTilde,
                                 int rows, int columns, String imageFormat)
```

Parameters

data
Specifies a string which will be encoded by PDF417 barcode.

narrowBarWidth
Specifies the narrow bar width in centimeters.

rotation
Specifies the orientation of the barcode, a valid value should be 0, 1, 2, or 3. Please refer to Orientation Constants.

resolution
Specifies the resolution (DPI) of a device onto which a barcode will be rendered. The default value is 96 for computer screen.

mode
Specifies the encoding mode for PDF417 barcode, a valid value should be 0, 1 or 2. Please refer to Mode Constants.

ecl
Specifies the error correction level, a valid value should be between 0 and 8. Please refer to Error Correction Level Constants.

handleTilde
Specifies a boolean flag indicating whether or not it is required to process the tilde character "~". If it is set to TRUE, non-printable characters can be passed to a barcode using the tilde character, "**~dNNN**" represents an ASCII character encoded by the 3 digits NNN, for example, "~d010" represents the character LF (line feed).

rows
Specifies the preferred number of rows for PDF417 barcode. If it is set to 0, an optimized value will be chosen instead.

columns
Specifies the preferred number of columns for PDF417 barcode. If it is set to 0, an optimized value will be chosen instead.

imageFormat
Specifies barcode image format, it can be "bmp", "jpg", or "png".

2.3.1.1.6 createQRCode

Returns an Image object containing a QRCode barcode.

```
public static Image createQRCode(String data, double moduleSize, int rotation, int resolution,
                                  int version, int level, int mask, String imageFormat)
```

Parameters

data
Specifies a string which will be encoded by QRCode barcode.

moduleSize
Specifies the size (width/height) in centimeters of a square-shaped module.

rotation
Specifies the orientation of the barcode, a valid value should be 0, 1, 2, or 3. Please refer to Orientation Constants

resolution
Specifies the resolution (DPI) of a device onto which a barcode will be rendered. The default value is 96 for computer screen.

version
Specifies the version of QRCode barcode, a valid value should be between 0 and 40. Please refer to Version Constants.

level
Specifies the level of error correction allowing recovery, a valid value should be between 0 and 3. Please refer to Level Constants

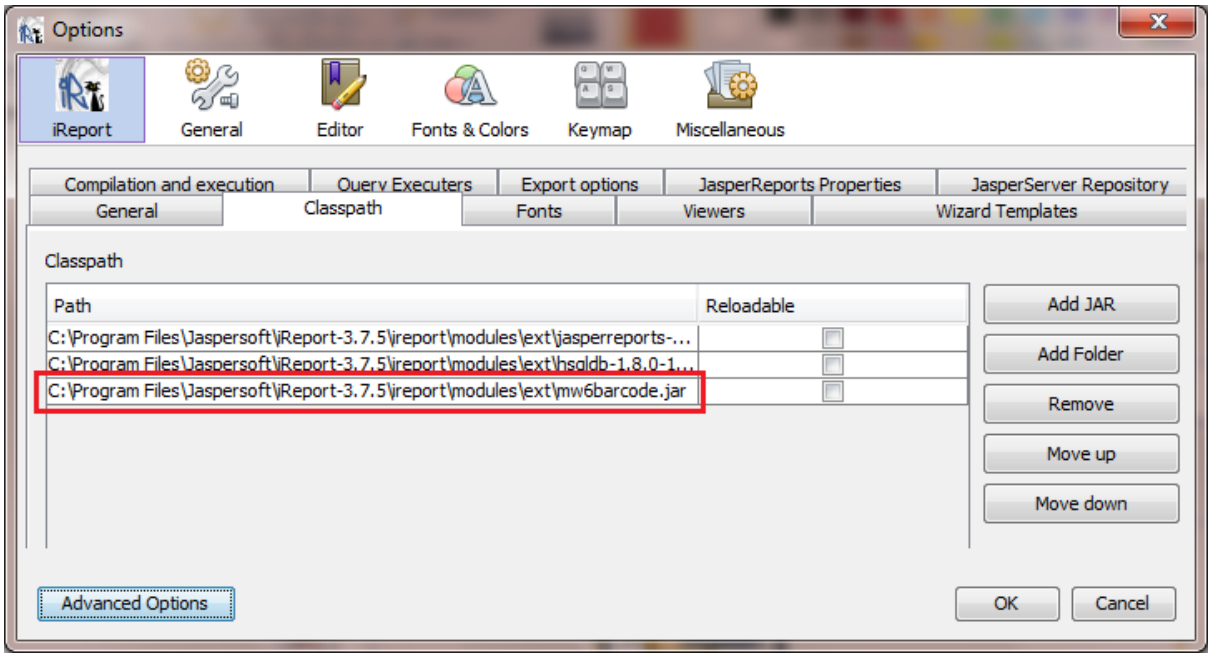
mask
Specifies the mask pattern for improving the readability, a valid value should be between 0 and 8. Please refer to Mask Constants

imageFormat
Specifies barcode image format, it can be "bmp", "jpg", or "png".

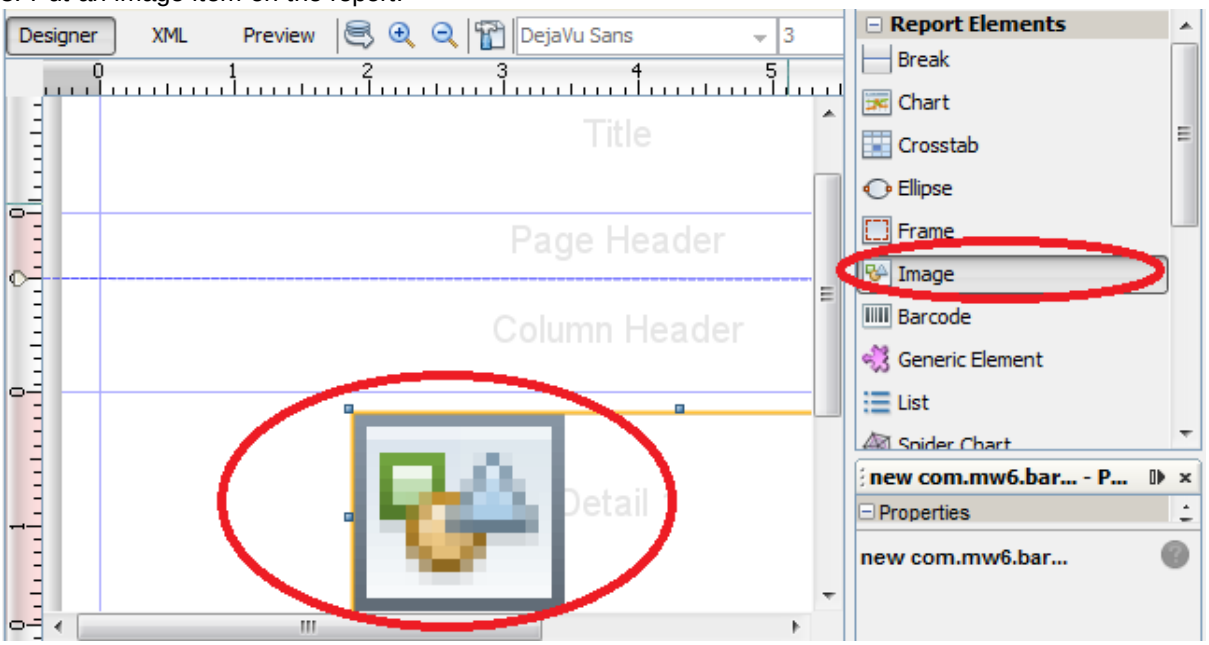
2.3.2 Instructions

1. Create a new report.

2. Add "mw6barcode.jar" to the report's **Classpath**.



3. Put an image item on the report.

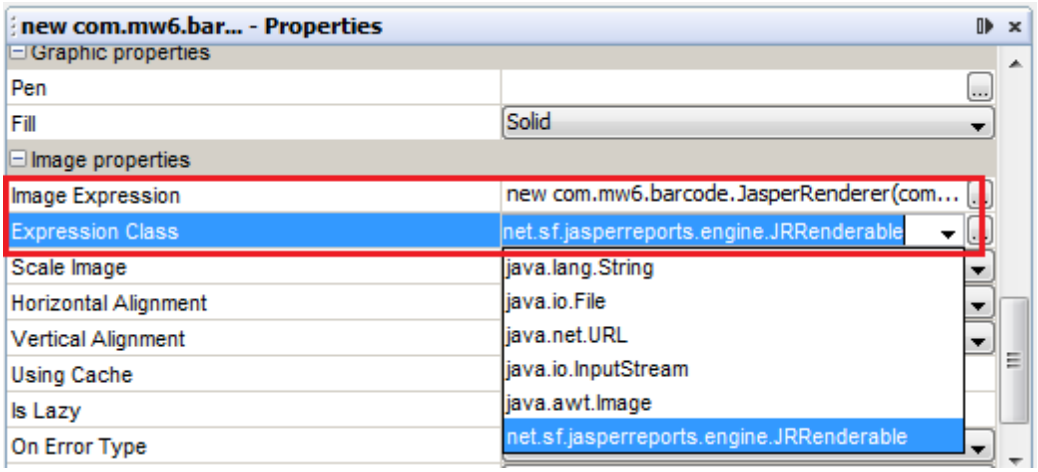


4. Click "**Cancel**" button for "**Select an image file**" window.

5. Edit the image item's properties by setting "**Expression Class**" to net.sf.jasperreports.engine.JRRenderable" and "**Image Expression**" to one of the following options:
   - `new com.mw6.barcode.JasperRenderer (com.mw6.barcode.ImageFactory.createBarcode(4, "1234", 0.07, 0, 96, 2.0, 12, true, true, true, 0, "", 0, 0, "png"))`
   - `new com.mw6.barcode.JasperRenderer`

```
(com.mw6.barcode.ImageFactory.createPDF417("1234", 0.07, 0, 96, false,
0, 0, true, 0, 0, "png"))
```

- `new com.mw6.barcode.JasperRenderer (com.mw6.barcode.ImageFactory.createDataMatrix("1234", 0.07, 0, 96, 0, 0, true, "png"))`
- `new com.mw6.barcode.JasperRenderer (com.mw6.barcode.ImageFactory.createAztec("1234", 0.07, 0, 96, 0, true, "png"))`
- `new com.mw6.barcode.JasperRenderer (com.mw6.barcode.ImageFactory.createQRCode("1234", 0.07, 0, 96, 0, 0, 0, "png"))`
- `new com.mw6.barcode.JasperRenderer (com.mw6.barcode.ImageFactory.createMaxiCode("1234", 0, 96, "", "", "", 2, true, "png"))`



6. Preview the report.

2.4 Eclipse BIRT

(1) Create a BLOB data type field in a database table.

(2) Use Java JDBC to upload a few barcode images to the BLOB data field of the table.

(3) Create a new BIRT report.

(4) Drop an "Image" element to the report.

(5) Set up the "Data Binding" properties of the "Image" element to connect to the BLOB field containing barcode image.

3 References

3.1 Common Parts

3.1.1 Methods

3.1.1.1 exportImageStream

Returns the barcode image data stream.

```
public byte[] exportImageStream(String formatName)
```

Parameters

formatName
Specifies the image format, it can be "bmp", "jpg", or "png".

3.1.1.2 getActualSize

Uses a two-element integer array to get the size (in pixels) of a barcode.

```
public void getActualSize(int[] sizeArray)
```

Parameters

sizeArray
Stores the barcode size, the first element stores the value of width and the second element stores the value of height.

3.1.1.3 getBackColor

Returns the background color.

```
public Color getBackColor()
```

3.1.1.4 getBarColor

Returns the color of barcode.

```
public Color getBarColor()
```

3.1.1.5 getData

Returns the string which will be encoded by a barcode.

```
public String getData()
```

3.1.1.6 getHandleTilde

Returns a boolean flag indicating whether or not it is required to process the tilde character "~".

```
public boolean getHandleTilde()
```

Remarks

This method is part of PDF417, DataMatrix, MaxiCode, and Aztec classes.

3.1.1.7 getModuleSize

Returns the size (width/height) in centimeters of a square-shaped module.

```
public double getModuleSize()
```

Remarks

This method is part of DataMatrix, Aztec and QRCode classes.

3.1.1.8 getNarrowBarWidth

Returns the narrow bar width in centimeters.

```
public double getNarrowBarWidth()
```

Remarks

This method is part of Barcode and PDF417 classes.

3.1.1.9 getResolution

Returns the resolution (DPI) of a device onto which a barcode will be rendered.

```
public int getResolution()
```

3.1.1.10 getRotation

Returns the orientation of a barcode.

```
public short getRotation()
```

See Also

Orientation Constants

3.1.1.11 render

Renders the barcode onto a device such as a computer monitor or printer.

```
public void render(Graphics targetG, int x, int y)
```

Parameters

targetG
Specifies a Graphics object to be used for rendering.

x
Stores the x coordinate (in pixels) of the top-left corner of a barcode.

y
Stores the y coordinate (in pixels) of the top-left corner of a barcode.

3.1.1.12 saveAsImage

Saves the barcode image to a file.

```
public void saveAsImage(String fileName, String formatName)
```

Parameters
fileName
Specifies a string that contains the name of file to which to save the barcode image.

formatName
Specifies the image format, it can be "bmp", "jpg", or "png".

3.1.1.13 setBackColor

Sets the background color.

```
public void setBackColor(Color backColor)
```

Parameters

backColor
Specifies the background color, the default value is white color.

3.1.1.14 setBarColor

Sets the color of barcode.

```
public void setBarColor(Color barColor)
```

Parameters

barColor
Specifies the color of barcode, the default value is black color.

3.1.1.15 setData

Specifies a string which will be encoded by a barcode.

```
public void setData(String data)
```

Parameters

data
Specifies a string which will be encoded by a barcode.

3.1.1.16 setHandleTilde

Specifies a boolean flag indicating whether or not it is required to process the tilde character "~".

```
public void setHandleTilde(boolean flag)
```

Parameters

flag
A boolean flag.

Remarks

If it is set to TRUE, non-printable characters can be passed to a barcode using the tilde character, "**~dNNN**" represents an ASCII character encoded by the 3 digits NNN, for example, "~d010" represents the character LF (line feed).

This method is part of PDF417, DataMatrix, MaxiCode, and Aztec classes.

3.1.1.17 setModuleSize

Specifies the size (width/height) in centimeters of a square-shaped module.

```
public void setModuleSize(double moduleSize)
```

Parameters

moduleSize
Specifies the size (width/height) in centimeters of a square-shaped module.

Remarks

This method is part of DataMatrix, Aztec and QRCode classes.

3.1.1.18 setNarrowBarWidth

Specifies the narrow bar width in centimeters.

```
public void setNarrowBarWidth(double narrowBarWidth)
```

Parameters

narrowBarWidth
Specifies the narrow bar width in centimeters.

Remarks

This method is part of Barcode and PDF417 classes.

3.1.1.19 setResolution

Sets the resolution (DPI) of a device onto which a barcode will be rendered.

```
public void setResolution(int resolution)
```

Parameters

resolution
Specifies the resolution (DPI) of a device onto which a barcode will be rendered. The default value is 96 for computer screen and a valid value can be 300, 600, 1200, or something else for printers.

3.1.1.20 setRotation

Sets the orientation of a barcode.

```
public void setRotation(short rotation)
```

Parameters

rotation
Specifies the orientation of a barcode, a valid value should be 0, 1, 2, or 3.

See Also

Orientation Constants

3.1.1.21 setSize

Sets the size of the image which contains a barcode.

```
public void setSize(int width, int height)
```

Parameters

width

Specifies the width, in pixels, of the image.

height
Specifies the height, in pixels, of the image.

3.1.2 Constants

3.1.2.1 Orientation

```
public static final short ORIENTATION_0 = 0;
public static final short ORIENTATION_90 = 1;
public static final short ORIENTATION_180 = 2;
public static final short ORIENTATION_270 = 3;
```

3.2 Aztec Class

3.2.1 Constructor

Constructs a new *Aztec* object.

```
public Aztec()
```

3.2.2 Methods

3.2.2.1 getPreferredFormat

Returns the preferred format for Aztec barcode.

```
public short getPreferredFormat()
```

See Also

Format Constants

3.2.2.2 setPreferredFormat

Specifies the preferred format for Aztec barcode.

```
public void setPreferredFormat(short format)
```

Parameters

format
Specifies the preferred format for Aztec barcode, a valid value should be between 0 and 36.

See Also

Format Constants

3.2.3 Constants

3.2.3.1 Format

```
public static final short FORMAT_AUTO = 0;
public static final short FORMAT_15X15_COMPACT = 1;
public static final short FORMAT_19X19 = 2;
public static final short FORMAT_19X19_COMPACT = 3;
public static final short FORMAT_23X23 = 4;
public static final short FORMAT_23X23_COMPACT = 5;
public static final short FORMAT_27X27 = 6;
public static final short FORMAT_27X27_COMPACT = 7;
public static final short FORMAT_31X31 = 8;
public static final short FORMAT_37X37 = 9;
public static final short FORMAT_41X41 = 10;
public static final short FORMAT_45X45 = 11;
public static final short FORMAT_49X49 = 12;
public static final short FORMAT_53X53 = 13;
public static final short FORMAT_57X57 = 14;
public static final short FORMAT_61X61 = 15;
public static final short FORMAT_67X67 = 16;
public static final short FORMAT_71X71 = 17;
public static final short FORMAT_75X75 = 18;
public static final short FORMAT_79X79 = 19;
public static final short FORMAT_83X83 = 20;
public static final short FORMAT_87X87 = 21;
public static final short FORMAT_91X91 = 22;
public static final short FORMAT_95X95 = 23;
public static final short FORMAT_101X101 = 24;
public static final short FORMAT_105X105 = 25;
public static final short FORMAT_109X109 = 26;
public static final short FORMAT_113X113 = 27;
public static final short FORMAT_117X117 = 28;
public static final short FORMAT_121X121 = 29;
public static final short FORMAT_125X125 = 30;
public static final short FORMAT_131X131 = 31;
public static final short FORMAT_135X135 = 32;
public static final short FORMAT_139X139 = 33;
public static final short FORMAT_143X143 = 34;
public static final short FORMAT_147X147 = 35;
public static final short FORMAT_151X151 = 36;
```

3.3 Barcode Class

3.3.1 Constructor

Constructs a new *Barcode* object.

```
public Barcode()
```

3.3.2 Methods

3.3.2.1 getBarHeight

Returns the barcode height in centimeters.

```
public double getBarHeight()
```

3.3.2.2 getBearerBar

Gets the bearer bar type.

```
public short getBearerBar()
```

See Also

Bearer Bar Constants

3.3.2.3 getBorderWidth

Returns the border width in centimeters.

```
public double getBorderWidth()
```

3.3.2.4 getCheckDigit

Gets a boolean flag indicating whether or not a check digit is required.

```
public boolean getCheckDigit()
```

3.3.2.5 getCheckDigitToText

Gets a boolean flag indicating whether or not a check digit is added to the human readable text.

```
public boolean getCheckDigitToText()
```

3.3.2.6 getShowText

Gets a boolean flag indicating whether or not the human readable text should be displayed.

```
public boolean getShowText()
```

3.3.2.7 getSupplement

Returns the supplement string for a UPC or EAN related barcode.

```
public String getSupplement()
```

3.3.2.8 getSupplementGap

Returns the distance, in centimeters, between the normal barcode and the supplement section.

```
public double getSupplementGap()
```

3.3.2.9 getSupplementType

Returns the supplement type.

```
public short getSupplementType()
```

See Also

Supplement Type Constants

3.3.2.10 getSymbologyType

Returns the barcode type.

```
public short getSymbologyType()
```

See Also

Symbology Type Constants

3.3.2.11 getTextFont

Returns the font of the human readable text in a barcode.

```
public Font getTextFont()
```

3.3.2.12 getWide2NarrowRatio

Returns the ratio of the wide bar to the narrow bar.

```
public double getWide2NarrowRatio()
```

3.3.2.13 setCodaBar

Specifies the start character and end character for CodaBar barcode.

```
public void setCodaBar(char startChar, char stopChar)
```

Parameters

startChar
Specifies the start character, a valid value should be 0, 1, 2 or 3.

stopChar
Specifies the end character, a valid value should be 0, 1, 2 or 3.

See Also

Codabar Character Constants

3.3.2.14 setCodeOne

Specifies the version of 2D Code One barcode.

```
public void setCodeOne(short COVersion)
```

Parameters

COVersion
Specifies the version of Code One, a valid value should be one of the following values:

Value	Size
1	16 X 18
2	22 X 22
3	28 X 32
4	40 X 42
5	52 X 54
6	70 X 76
7	104 X 98
8	148 X 134
9	8 X varied width
10	16 X varied width

3.3.2.15 setMicroPDF417

Specifies the number of columns for 2D Micro PDF417 barcode.

```
public void setMicroPDF417(short Columns)
```

Parameters

Columns
Specifies the number of columns, a valid value should be one of the following values:

Value	Description
1	1 column
2	2 columns
3	3 columns
4	4 columns

3.3.2.16 setMicroQRCode

Specifies the version and error correction level for 2D Micro QRCode barcode.

```
public void setMicroQRCode(short MQVersion, short MQLevel)
```

Parameters

MQVersion
Specifies the version of Micro QRCode, a valid value should be one of the following values:

Value	Description
1	Version M1 with the size 11 X 11
2	Version M2 with the size 13 X 13
3	Version M3 with the size 15 X 15
4	Version M4 with the size 17 X 17

MQLevel
Specifies the error correction level of Micro QRCode, a valid value should be one of the following values:

Value	Description
1	L (applicable to version M2, M3 and M4)
2	M (applicable to version M2, M3 and M4)
3	Q (applicable to version M4 only)

3.3.2.17 setBarHeight

Specifies the barcode height in centimeters.

```
public void setBarHeight(double barHeight)
```

Parameters

barHeight
Specifies the barcode height in centimeters. The default value is 1.5.

3.3.2.18 setBearerBar

Sets the bearer bar type.

```
public void setBearerBar(short bearerBar)
```

Parameters

bearerBar
Specifies the bearer bar type, a valid value should be 0, 1, or 2.

See Also

Bearer Bar Constants

3.3.2.19 setBorderWidth

Specifies the border width in centimeters.

```
public void setBorderWidth(double borderWidth)
```

Parameters

borderWidth
Specifies the border width in centimeters.

3.3.2.20 setCheckDigit

Sets a boolean flag indicating whether or not a check digit is required.

```
public void setCheckDigit(boolean flag)
```

Parameters

flag
Specifies whether or not a check digit is required.

3.3.2.21 setCheckDigitToText

Sets a boolean flag indicating whether or not a check digit is added to the human readable text.

```
public void setCheckDigitToText(boolean flag)
```

Parameters

flag
Specifies whether or not a check digit is added to the human readable text.

3.3.2.22 setShowText

Sets a boolean flag indicating whether or not the human readable text should be displayed.

```
public void setShowText(boolean flag)
```

Parameters

flag
Specifies whether or not the human readable text should be displayed.

3.3.2.23 setSupplement

Sets a supplement string for a UPC or EAN related barcode.

```
public void setSupplement(String supplement)
```

Parameters

supplement
Specifies a supplement string, the length of this string can be either 2 or 5.

3.3.2.24 setSupplementGap

Sets the distance, in centimeters, between the normal barcode and the supplement section.

```
public void setSupplementGap(double supplementGap)
```

Parameters

supplementGap
Specifies the distance, in centimeters, between the normal barcode and the supplement section.

3.3.2.25 setSupplementType

Specifies the supplement type.

```
public void setSupplementType(short supplementType)
```

Parameters

supplementType
Specifies the supplement type, a valid value should be 0, 1, or 2.

See Also

Supplement Type Constants

3.3.2.26 setSymbologyType

Specifies the barcode type.

```
public void setSymbologyType(short symbologyType)
```

Parameters

symbologyType
Specifies the barcode type, a valid value should be between 1 and 91.

See Also

Symbology Type Constants

3.3.2.27 setTextFont

Specifies the font of the human readable text in a barcode.

```
public void setTextFont(Font textFont)
```

Parameters

textFont
Specifies the font of the human readable text in a barcode.

3.3.2.28 setUPCESystem

Specifies the encoding system of UPC-E barcode.

```
public void setUPCESystem(short system)
```

Parameters

system
Specifies the encoding system, a valid value should be 0 or 1.

See Also

UPC System Constants

3.3.2.29 setWide2NarrowRatio

Specifies the ratio of the wide bar to the narrow bar.

```
public void setWide2NarrowRatio(double wide2NarrowRatio)
```

Parameters

wide2NarrowRatio
Specifies the ratio of the wide bar to the narrow bar.

3.3.3 Constants

3.3.3.1 Bearer Bar

```
public static final short BEARER_BAR_NONE = 0;
public static final short BEARER_BAR_HORIZONTAL = 1;
public static final short BEARER_BAR_BOX = 2;
```

3.3.3.2 Codabar Character

```
public static final short CODABAR_CHAR_A = 0;
public static final short CODABAR_CHAR_B = 1;
public static final short CODABAR_CHAR_C = 2;
public static final short CODABAR_CHAR_D = 3;
```

3.3.3.3 Supplement

```
public static final short SUPPLEMENT_NONE = 0;
public static final short SUPPLEMENT_2 = 1;
public static final short SUPPLEMENT_5 = 2;
```

3.3.3.4 Symbology

```
public static final short SYMBOLOGY_CHANNEL_CODE = 1;
public static final short SYMBOLOGY_CODABAR = 2;
public static final short SYMBOLOGY_CODE_11 = 3;
public static final short SYMBOLOGY_CODE_128 = 4;
public static final short SYMBOLOGY_CODE_128_SET_A = 5;
public static final short SYMBOLOGY_CODE_128_SET_B = 6;
public static final short SYMBOLOGY_CODE_128_SET_C = 7;
public static final short SYMBOLOGY_CODE_32_OR_ITALIAN_PHARMACODE = 8;
public static final short SYMBOLOGY_CODE_39 = 9;
public static final short SYMBOLOGY_CODE_39_EXTENDED = 10;
public static final short SYMBOLOGY_CODE_93 = 11;
public static final short SYMBOLOGY_DATA_LOGIC_2_5 = 12;
public static final short SYMBOLOGY_EAN128_UCC_GS1_128 = 13;
```

```
        public static final short SYMBOLOGY_EAN_13 = 14;
        public static final short SYMBOLOGY_EAN_8 = 15;
        public static final short SYMBOLOGY_EAN_VELOCITY = 16;
        public static final short SYMBOLOGY_FLATTERMARKEN = 17;
        public static final short SYMBOLOGY_GS1_DATABAR_14 = 18;
        public static final short SYMBOLOGY_GS1_DATABAR_EXPANDED = 19;
        public static final short SYMBOLOGY_GS1_DATABAR_EXPANDED_STACKED = 20;
        public static final short SYMBOLOGY_GS1_DATABAR_LIMITED = 21;
        public static final short SYMBOLOGY_GS1_DATABAR_STACKED = 22;
        public static final short SYMBOLOGY_GS1_DATABAR_STACKED_OMNIDIRECTIONAL = 23;
        public static final short SYMBOLOGY_GS1_DATABAR_TRUNCATED = 24;
        public static final short SYMBOLOGY_GS1_DATABAR_14_COMPOSITE = 25;
        public static final short SYMBOLOGY_GS1_DATABAR_EXPANDED_COMPOSITE = 26;
        public static final short SYMBOLOGY_GS1_DATABAR_EXPANDED_STACKED_COMPOSITE
= 27;
        public static final short SYMBOLOGY_GS1_DATABAR_LIMITED_COMPOSITE = 28;
        public static final short SYMBOLOGY_GS1_DATABAR_STACKED_COMPOSITE = 29;
        public static final short
SYMBOLOGY_GS1_DATABAR_STACKED_OMNIDIRECTIONAL_COMPOSITE = 30;
        public static final short SYMBOLOGY_HIBC_CODE_128_FOR_LIC_OR_PAS = 31;
        public static final short SYMBOLOGY_HIBC_CODE_39_FOR_LIC_OR_PAS = 32;
        public static final short SYMBOLOGY_HIBC_CODABLOCK_F_FOR_LIC_OR_PAS = 33;
        public static final short SYMBOLOGY_HIBC_MICRO_PDF417_FOR_LIC_OR_PAS = 34;
        public static final short SYMBOLOGY_IATA_2_OF_5_BARCODE = 35;
        public static final short SYMBOLOGY_INDUSTRIAL_2_OF_5_BARCODE = 36;
        public static final short SYMBOLOGY_INTERLEAVED_2_OF_5_BARCODE = 37;
        public static final short
SYMBOLOGY_ISBN_OR_INTERNATIONAL_STANDARD_BOOK_NUMBER = 38;
        public static final short
SYMBOLOGY_ISMN_OR_INTERNATIONAL_STANDARD_MUSIC_NUMBER = 39;
        public static final short
SYMBOLOGY_ISSN_OR_INTERNATIONAL_STANDARD_SERIAL_NUMBER = 40;
        public static final short SYMBOLOGY_ITF_14_OR_UPC_SHIPPING_CONTAINER_SYMBOL =
41;
        public static final short SYMBOLOGY_JAN_13 = 42;
        public static final short SYMBOLOGY_JAN_8 = 43;
        public static final short SYMBOLOGY_LOGMARS = 44;
        public static final short SYMBOLOGY_MATRIX_2_OF_5_BARCODE = 45;
        public static final short SYMBOLOGY_MSI_PLESSEY = 46;
        public static final short SYMBOLOGY_NUMLY_NUMBER_OR_ESN = 47;
        public static final short SYMBOLOGY_OPTICAL_PRODUCT_CODE = 48;
        public static final short SYMBOLOGY_PHARMACODE_ONE_TRACK = 49;
        public static final short SYMBOLOGY_PHARMACODE_TWO_TRACK = 50;
        public static final short SYMBOLOGY_PHARMA_ZENTRAL_NUMMER = 51;
        public static final short SYMBOLOGY_SCC_14_OR_SHIPPING_CONTAINER_CODE = 52;
        public static final short
SYMBOLOGY_SSCC_18_OR_UPC_128_SHIPPING_CONTAINER_CODE = 53;
        public static final short SYMBOLOGY_TELEPEN_ALPHA = 54;
        public static final short SYMBOLOGY_TELEPEN_NUMERIC = 55;
        public static final short SYMBOLOGY_UK_PLESSEY = 56;
        public static final short SYMBOLOGY_UPC_A = 57;
        public static final short SYMBOLOGY_UPC_E = 58;
        public static final short SYMBOLOGY_VICS_BOL_OR_VICS_BILL_OF_LADING = 59;
```

```
public static final short SYMBOLOGY_VICS_SCAC_PRO = 60;
public static final short SYMBOLOGY_AUSTRALIA_POSTAL_STANDARD_CUSTOMER = 61;
public static final short SYMBOLOGY_AUSTRALIA_POSTAL_REDIRECTION = 62;
public static final short SYMBOLOGY_AUSTRALIA_POSTAL_REPLY_PAID = 63;
public static final short SYMBOLOGY_AUSTRALIA_POSTAL_ROUTING = 64;
public static final short SYMBOLOGY_CHINA_POSTAL_CODE = 65;
public static final short SYMBOLOGY_DANISH_POSTAL_CODE = 66;
public static final short SYMBOLOGY_DEUTSCHE_POST_IDENTCODE = 67;
public static final short SYMBOLOGY_DEUTSCHE_POST_LEITCODE = 68;
public static final short SYMBOLOGY_FRANCE_POSTAL_CODE_39 = 69;
public static final short SYMBOLOGY_ITALY_POSTAL_CODE_2_5 = 70;
public static final short SYMBOLOGY_ITALY_POSTAL_CODE_39 = 71;
public static final short SYMBOLOGY_JAPAN_POSTAL_CODE = 72;
public static final short SYMBOLOGY_KIX_OR_NETHERLANDS_POSTAL_CODE = 73;
public static final short SYMBOLOGY_KOREAN_POSTAL_CODE = 74;
public static final short SYMBOLOGY_ROYAL_MAIL_4_STATE = 75;
public static final short SYMBOLOGY_SINGAPORE_POSTAL_CODE = 76;
public static final short SYMBOLOGY_SWISS_PARCEL_POST_BARCODE = 77;
public static final short SYMBOLOGY_USPS_DAFT_CODE = 78;
public static final short SYMBOLOGY_USPS_FACING_IDENTIFICATION_MARK = 79;
public static final short SYMBOLOGY_USPS_HORIZONTAL_BARS = 80;
public static final short SYMBOLOGY_USPS_ONECODE_OR_USPS_INTELLIGENT_MAIL =
81;
public static final short SYMBOLOGY_USPS_PLANET = 82;
public static final short SYMBOLOGY_USPS_POSTNET = 83;
public static final short SYMBOLOGY_USPS_SACK_LABEL = 84;
public static final short SYMBOLOGY_USPS_TRAY_LABEL = 85;
public static final short SYMBOLOGY_CODABLOCK_F = 86;
public static final short SYMBOLOGY_CODE_16K = 87;
public static final short SYMBOLOGY_CODE_49 = 88;
public static final short SYMBOLOGY_CODE_ONE = 89;
public static final short SYMBOLOGY_MICRO_PDF417 = 90;
public static final short SYMBOLOGY_MICRO_QRCODE = 91;
```

Value	Barcode Description	Allow Bearer Bars?	Allow Supplement 2 or 5?	Sample Barcode String
1D Barcodes				
1	Channel Code			
2	Codabar			
3	Code 11			
4	Code 128			1234ABCD+/
5	Code 128 (Set A)			
6	Code 128 (Set B)			
7	Code 128 (Set C)			
8	Code 32 or Italian Pharmacode			
9	Code 39			1234ABCD
10	Code 39 Extended			
11	Code 93			

12	Data Logic 2/5	Yes		
13	EAN128/UCC (GS1-128)			(21)95FNC1(11)090101
14	EAN 13		Yes	123456789012
15	EAN 8		Yes	1234567
16	EAN Velocity		Yes	
17	Flattermarken			
18	GS1 Databar-14			1234567890123
19	GS1 DataBar Expanded			
20	GS1 DataBar Expanded Stacked			
21	GS1 Databar Limited			
22	GS1 Databar Stacked			
23	GS1 DataBar Stacked Omnidirectional			
24	GS1 Databar Truncated			
25	GS1 Databar-14 Composite			
26	GS1 DataBar Expanded Composite			(01)1234567890123
27	GS1 DataBar Expanded Stacked Composite			
28	GS1 Databar Limited Composite			
29	GS1 Databar Stacked Composite			
30	GS1 DataBar Stacked Omnidirectional Composite			
31	HIBC Code 128 for LIC or PAS			+H123ABC01234567890D
32	HIBC Code 39 for LIC or PAS			+/EAH783B
33	HIBC CodaBlock-F for LIC or PAS			+/EAH783/Z34H159$
34	HIBC Micro PDF417 for LIC or PAS			
35	IATA 2 of 5 Barcode	Yes		
36	Industrial 2 of 5 Barcode	Yes		
37	Interleaved 2 of 5 Barcode	Yes		
38	ISBN or International Standard Book Number		Yes	3161484100
39	ISMN or International Standard Music Number		Yes	M-2306-7118-7
40	ISSN or International Standard Serial Number		Yes	0264-3596
41	ITF-14 or UPC Shipping Container Symbol	Yes		
42	JAN 13		Yes	
43	JAN 8		Yes	
44	Logmars			
45	Matrix 2 of 5 Barcode	Yes		
46	MSI/Plessey			
47	Numly Number or ESN			1234567890123456789
48	Optical Product Code		Yes	123456789
49	Pharmacode One-Track			
50	Pharmacode Two-Track			
51	Pharma-Zentral-Nummer			123456

52	SCC-14 or Shipping Container Code			
53	SSCC-18 or UPC-128 Shipping Container Code			
54	Telepen Alpha			
55	Telepen Numeric			
56	UK Plessey			
57	UPC-A		Yes	1234567890
58	UPC-E		Yes	1234567
59	VICS BOL or VICS Bill of Lading			
60	VICS SCAC PRO			
Postal Code Barcodes				
61	Australia Postal Standard Customer			
62	Australia Postal Redirection			
63	Australia Postal Reply Paid			
64	Australia Postal Routing			
65	China Postal Code			
66	Danish Postal Code			CC12345678
67	Deutsche Post Identcode			12345678901
68	Deutsche Post Leitcode			1234567890123
69	France Postal Code 39			RA12345678
70	Italy Postal Code 2/5			
71	Italy Postal Code 39			
72	Japan Postal Code			1234567AZ
73	KIX or Netherlands Postal Code			A12345678
74	Korean Postal Code			123456
75	Royal Mail 4 State			
76	Singapore Postal Code			
77	Swiss Parcel Post Barcode			
78	USPS DAFT Code			DAFTTFAD
79	USPS Facing Identification Mark			A
80	USPS Horizontal Bars			
81	USPS OneCode or USPS Intelligent Mail			12345678901234567890 +50309
82	USPS PLANET			
83	USPS POSTNET			
84	USPS Sack Label			50309123
85	USPS Tray Label			5030912345
2D Barcodes				
86	Codablock-F			
87	Code 16K			
88	Code 49			
89	Code One			
90	Micro PDF417			
91	Micro QRCode			

3.3.3.5 UPC System

```
public static final short UPC_SYSTEM_0 = 0;
public static final short UPC_SYSTEM_1 = 1;
```

3.4 DataMatrix Class

3.4.1 Constructor

Constructs a new *DataMatrix* object.

```
public DataMatrix()
```

3.4.2 Methods

3.4.2.1 getMode

Returns the encoding mode for DataMatrix barcode.

```
public short getMode()
```

See Also

Mode Constants

3.4.2.2 getPreferredFormat

Returns the preferred format for DataMatrix barcode.

```
public short getPreferredFormat()
```

See Also

Format Constants

3.4.2.3 setMode

Specifies the encoding mode for DataMatrix barcode.

```
public void setMode(short mode)
```

Parameters

mode
Specifies the encoding mode for DataMatrix barcode, a valid value should be between 0 and 3.

See Also

Mode Constants

3.4.2.4 setPreferredFormat

Specifies the preferred format for DataMatrix barcode.

```
public void setPreferredFormat(short format)
```

Parameters

format
Specifies the preferred format for DataMatrix barcode, a valid value should be between 0 and 30.

See Also

Format Constants

3.4.3 Constants

3.4.3.1 Format

```
public static final short FORMAT_FORMAT_AUTO = 0;
public static final short FORMAT_10X10 = 1;
public static final short FORMAT_12X12 = 2;
public static final short FORMAT_14X14 = 3;
public static final short FORMAT_16X16 = 4;
public static final short FORMAT_18X18 = 5;
public static final short FORMAT_20X20 = 6;
public static final short FORMAT_22X22 = 7;
public static final short FORMAT_24X24 = 8;
public static final short FORMAT_26X26 = 9;
public static final short FORMAT_32X32 = 10;
public static final short FORMAT_36X36 = 11;
public static final short FORMAT_40X40 = 12;
public static final short FORMAT_44X44 = 13;
public static final short FORMAT_48X48 = 14;
public static final short FORMAT_52X52 = 15;
public static final short FORMAT_64X64 = 16;
public static final short FORMAT_72X72 = 17;
public static final short FORMAT_80X80 = 18;
public static final short FORMAT_88X88 = 19;
public static final short FORMAT_96X96 = 20;
public static final short FORMAT_104X104 = 21;
public static final short FORMAT_120X120 = 22;
public static final short FORMAT_132X132 = 23;
public static final short FORMAT_144X144 = 24;
public static final short FORMAT_8X18 = 25;
public static final short FORMAT_8X32 = 26;
public static final short FORMAT_12X26 = 27;
public static final short FORMAT_12X36 = 28;
public static final short FORMAT_16X36 = 29;
public static final short FORMAT_16X48 = 30;
```

3.4.3.2 Mode

```
public static final short MODE_ASCII = 0;
public static final short MODE_C40 = 1;
public static final short MODE_TEXT = 2;
public static final short MODE_BASE256 = 3;
```

3.5 MaxiCode Class

3.5.1 Constructor

Constructs a new *MaxiCode* object.

```
public MaxiCode()
```

3.5.2 Methods

3.5.2.1 getCountry

Returns the country code of MaxiCode barcode.

```
public String getCountry()
```

3.5.2.2 getMode

Returns the encoding mode for MaxiCode barcode.

```
public short getMode()
```

See Also

Mode Constants

3.5.2.3 getServiceClass

Returns the service class of MaxiCode barcode.

```
public String getServiceClass()
```

3.5.2.4 getZipCode

Returns the zip code of MaxiCode barcode.

```
public String getZipCode()
```

3.5.2.5 setCountry

Specifies the country code of MaxiCode barcode.

```
public void setCountry(String country)
```

Parameters

country
Specifies the country code of MaxiCode barcode.

3.5.2.6 setMode

Specifies the encoding mode for MaxiCode barcode.

```
public void setMode(short mode)
```

Parameters

mode
Specifies the encoding mode for Maxicode barcode, a valid value should be between 2 and 5.

See Also

Mode Constants

3.5.2.7 setServiceClass

Specifies the service class of MaxiCode barcode.

```
public void setServiceClass(String serviceClass)
```

Parameters

serviceClass
Specifies the service class of MaxiCode barcode.

3.5.2.8 setZipCode

Specifies the zip code of MaxiCode barcode.

```
public void setZipCode(String zipCode)
```

Parameters

zipCode
Specifies the zip code of MaxiCode barcode.

3.5.3 Constants

3.5.3.1 Mode

```
public static final short MODE_2 = 2;
public static final short MODE_3 = 3;
public static final short MODE_4 = 4;
public static final short MODE_5 = 5;
```

3.6 PDF417 Class

Enter topic text here.

3.6.1 Constructor

Constructs a new *PDF417* object.

```
public PDF417()
```

3.6.2 Methods

3.6.2.1 getColumns

Returns the preferred number of columns for PDF417 barcode.

```
public short getColumns()
```

3.6.2.2 getErrorCorrectionLevel

Returns the error correction level for PDF417 barcode.

```
public short getErrorCorrectionLevel()
```

See Also

Error Correction Level Constants

3.6.2.3 getMode

Returns the encoding mode for PDF417 barcode.

```
public short getMode()
```

See Also

Mode Constants

3.6.2.4 getNarrowBarWidth

Returns the narrow bar width in centimeters.

```
public double getNarrowBarWidth()
```

3.6.2.5 getRows

Returns the preferred number of rows for PDF417 barcode.

```
public short getRows()
```

3.6.2.6 getTruncateSymbol

Returns a boolean flag indicating whether or not the right side of PDF417 barcode should be removed.

```
public boolean getTruncateSymbol()
```

3.6.2.7 getY2XRatio

Returns the ratio of the height of row to the width of narrow element.

```
public double getY2XRatio(double y2XRatio)
```

3.6.2.8 setColumns

Specifies the preferred number of columns for PDF417 barcode.

```
public void setColumns(short columns)
```

Parameters

columns
Specifies the preferred number of columns for PDF417 barcode.

Remarks

Typically the number of columns should be between 3 and 30.

3.6.2.9 setErrorCorrectionLevel

Specifies the error correction level for PDF417 barcode.

```
public void setErrorCorrectionLevel(short ecl)
```

Parameters

ecl
Specifies the error correction level, a valid value should be between 0 and 8.

See Also

Error Correction Level Constants

3.6.2.10 setMode

Specifies the encoding mode for PDF417 barcode.

```
public void setMode(short mode)
```

Parameters

mode
Specifies the encoding mode for PDF417 barcode, a valid value should be 0, 1 or 2.

See Also

Mode Constants

3.6.2.11 setNarrowBarWidth

Specifies the narrow bar width in centimeters.

```
public void setNarrowBarWidth(double narrowBarWidth)
```

Parameters

narrowBarWidth
Specifies the narrow bar width in centimeters.

3.6.2.12 setRows

Specifies the preferred number of rows for PDF417 barcode.

```
public void setRows(short rows)
```

Parameters

rows
Specifies the preferred number of rows for PDF417 barcode.

Remarks

Typically the number of rows should be between 3 and 90.

3.6.2.13 setTruncateSymbol

Sets a boolean flag indicating whether or not the right side of PDF417 barcode should be removed.

```
public void setTruncateSymbol(boolean flag)
```

Parameters

flag
A boolean flag.

3.6.2.14 setY2XRatio

Specifies the ratio of the height of row to the width of narrow element.

```
public void setY2XRatio(double y2XRatio)
```

Parameters

y2XRatio
Specifies the ratio of the height of row to the width of narrow element. A valid value should be between 2

and 6.

3.6.3 Constants

3.6.3.1 Mode

```
public static final short MODE_BINARY = 0;
public static final short MODE_TEXT = 1;
public static final short MODE_AUTO = 2;
```

3.6.3.2 Error Correction Level

```
public static final short ECLEVEL_0 = 0;
public static final short ECLEVEL_1 = 1;
public static final short ECLEVEL_2 = 2;
public static final short ECLEVEL_3 = 3;
public static final short ECLEVEL_4 = 4;
public static final short ECLEVEL_5 = 5;
public static final short ECLEVEL_6 = 6;
public static final short ECLEVEL_7 = 7;
public static final short ECLEVEL_8 = 8;
```

3.7 QRCode Class

3.7.1 Constructor

Constructs a new *QRCode* object.

```
public QRCode()
```

3.7.2 Methods

3.7.2.1 getLevel

Returns the level of error correction allowing recovery.

```
public short getLevel()
```

See Also

Level Constants

3.7.2.2 getMask

Returns the mask pattern for improving the readability.

```
public short getMask()
```

See Also

Mask Constants

3.7.2.3 getVersion

Returns the version of QRCode barcode.

```
public short getVersion()
```

See Also

Version Constants

3.7.2.4 setLevel

Specifies the level of error correction allowing recovery.

```
public void setLevel(short level)
```

Parameters

level
Specifies the level of error correction allowing recovery, a valid value should be between 0 and 3.

See Also

Level Constants

3.7.2.5 setMask

Specifies the mask pattern for improving the readability.

```
public void setMask(short mask)
```

Parameters

mask
Specifies the mask pattern for improving the readability, a valid value should be between 0 and 8.

See Also

Mask Constants

3.7.2.6 setVersion

Specifies the version of QRCode barcode.

```
public void setVersion(short version)
```

Parameters

version
Specifies the version of QRCode barcode, a valid value should be between 0 and 40.

See Also

Version Constants

3.7.3 Constants

3.7.3.1 Level

```
public static final short LEVEL_L = 0;
public static final short LEVEL_M = 1;
public static final short LEVEL_Q = 2;
public static final short LEVEL_H = 3;
```

3.7.3.2 Mask

```
public static final short MASK_AUTO = 0;
public static final short MASK_0 = 1;
public static final short MASK_1 = 2;
public static final short MASK_2 = 3;
public static final short MASK_3 = 4;
public static final short MASK_4 = 5;
public static final short MASK_5 = 6;
public static final short MASK_6 = 7;
public static final short MASK_7 = 8;
```

3.7.3.3 Version

```
public static final short VERSION_AUTO = 0;
public static final short VERSION_1 = 1;
public static final short VERSION_2 = 2;
public static final short VERSION_3 = 3;
public static final short VERSION_4 = 4;
public static final short VERSION_5 = 5;
public static final short VERSION_6 = 6;
public static final short VERSION_7 = 7;
public static final short VERSION_8 = 8;
public static final short VERSION_9 = 9;
public static final short VERSION_10 = 10;
public static final short VERSION_11 = 11;
public static final short VERSION_12 = 12;
public static final short VERSION_13 = 13;
public static final short VERSION_14 = 14;
public static final short VERSION_15 = 15;
public static final short VERSION_16 = 16;
public static final short VERSION_17 = 17;
public static final short VERSION_18 = 18;
public static final short VERSION_19 = 19;
public static final short VERSION_20 = 20;
public static final short VERSION_21 = 21;
public static final short VERSION_22 = 22;
public static final short VERSION_23 = 23;
public static final short VERSION_24 = 24;
public static final short VERSION_25 = 25;
public static final short VERSION_26 = 26;
public static final short VERSION_27 = 27;
```

```
public static final short VERSION_28 = 28;
public static final short VERSION_29 = 29;
public static final short VERSION_30 = 30;
public static final short VERSION_31 = 31;
public static final short VERSION_32 = 32;
public static final short VERSION_33 = 33;
public static final short VERSION_34 = 34;
public static final short VERSION_35 = 35;
public static final short VERSION_36 = 36;
public static final short VERSION_37 = 37;
public static final short VERSION_38 = 38;
public static final short VERSION_39 = 39;
public static final short VERSION_40 = 40;
```