

QR Code generator library

Introduction

This project aims to be the best, clearest QR Code generator library in multiple languages. The primary goals are flexible options and absolute correctness. The secondary goals are compact implementation size and good documentation comments.

Features

Core features:

- Available in 6 programming languages, all with nearly equal functionality: Java, Typescript/JavaScript, Python, Rust, C++, C
- Significantly shorter code but more documentation comments compared to competing libraries
- Supports encoding all 40 versions (sizes) and all 4 error correction levels, as per the QR Code Model 2 Standard
- Output format: Raw modules/pixels of the QR symbol
- Detects finder-like penalty patterns more accurately than other implementations
- Encodes numeric and special-alphanumeric text in less space than general text
- Open-source code under the permissive MIT License

Manual parameters:

- User can specify minimum and maximum version numbers allowed, then the library will automatically choose smallest version in the range that fits the data
- User can specify mask pattern manually, otherwise library will automatically evaluate all 8 masks and select the optimal one
- User can specify absolute error correction level, or allow the library to boost it if it doesn't increase the version number
- User can create a list of data segments manually and add ECI segments

Optional advanced features (Java only):

- Encodes Japanese Unicode text in kanji mode to save a lot of space compared to UTF-8 bytes
- Computes optimal segment mode switching for text with mixed numeric/alphanumeric/general/kanji parts

More information about QR Code technology and this library's design can be found on the project home page.

Examples

The code below is in Java, but the other language ports are designed with essentially the same API naming and behavior.

```
import java.awt.image.BufferedImage;
import java.io.File;
import java.util.List;
import javax.imageio.ImageIO;
import io.qrcodegen.*;

// Simple operation
QrCode qr0 = QrCode.encodeText("Hello, world!", QrCode.Ecc.MEDIUM);
BufferedImage img = toImage(qr0, 4, 10); // See QrCodeGeneratorDemo
ImageIO.write(img, "png", new File("qr-code.png"));

// Manual operation
List<QrSegment> segs = QrSegment.makeSegments("3141592653589793238462643383");
QrCode qr1 = QrCode.encodeSegments(segs, QrCode.Ecc.HIGH, 5, 5, 2, false);
for (int y = 0; y < qr1.size; y++) {
    for (int x = 0; x < qr1.size; x++) {
        (... paint qr1.getModule(x, y) ...)
    }
}
```

License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
- The Software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and non-infringement. In no event shall the authors or copyright holders be liable for any claim, damages, or other liability, whether in an action of contract, tort, or otherwise, arising from, out of, or in connection with the Software or the use or other dealings in the Software.