

有机码二维码生成查询说明

1. 有机码二维码生成规则

1.1. 生成规则

认证机构可以通过软件程序或工具批量制作有机码二维码。

生成规则为“[https://f.cnca.cn/f?p=](https://f.cnca.cn/f?p=10019887756456469)”+“有机码”



1.2. 最小尺寸规格

生成的二维码图片分辨率建议不小于 300PX*300PX。

印刷尺寸建议不小于 30*30PX（取决于商品大小、印厂）。

目前都是印厂在制订尺寸、二维码花纹样式。



1.3. 软件程序生成示例

1.3.1. 概述

程序可根据提供的有机码一次生成一张二维码，也可同时生成多张二维码图

片, 图片默认 PNG 格式, 也可通过调整程序生成其他图片格式。可根据需要在二维码图片中间插入 logo。

1.3.2. 调用参数

参数名称	类型	参数示例
orgaCode	字符串	"13417618112241138"
orgaCodeList	数组	["13417618112241138", "23657618112241526", "11097618112247739"]

1.3.3. 程序可调整参数

参数名称	说明	参数示例
QRCODE_SIZE	大小	300
QRCODE_WIDTH	分辨率宽度	300
QRCODE_HEIGHT	分辨率高度	300
filePath	生成二维码路径	D:\Temp\
urlPrefix	二维码网址前缀	https://f.cnca.cn/f/?p=
logoPath	logo 所在路径	D:\Temp\logo.png
picFormat	生成二维码格式	PNG

1.3.4. 程序示例

```
import java.awt.BasicStroke;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.Shape;
import java.awt.geom.RoundRectangle2D;
import java.awt.image.BufferedImage;
import java.io.File;
import java.util.Arrays;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.imageio.ImageIO;
import com.google.zxing.BarcodeFormat;
import com.google.zxing.EncodeHintType;
import com.google.zxing.MultiFormatWriter;
import com.google.zxing.common.BitMatrix;
import com.google.zxing.qrcode.decoder.ErrorCorrectionLevel;
```

```

public class QRCodeGenerator {

    private static final int QR_CODE_SIZE = 300;

    private static final int QR_CODE_WIDTH = 300;    // 图像宽度
    private static final int QR_CODE_HEIGHT = 300;    // 图像高度

    private static final String filePath = "D:\\Temp\\";
    private static final String urlPrefix = "https://f.cnca.cn/f/?p=";
    private static final String logoPath = "D:\\Temp\\logo.png";
    private static final String picFormat = "PNG";    // 图像类型

    /**
     * 生成二维码图片
     * @param orgaCode 有机码
     * @throws Exception
     */
    private static void createQRCodeImage(String orgaCode) throws Exception {
        String url = urlPrefix + orgaCode;
        Map<String,String> json = new HashMap<String,String>();
        json.put("zxing",url);
        Map<EncodeHintType, Object> hints = new HashMap<EncodeHintType, Object>();
        hints.put(EncodeHintType.CHARACTER_SET, "UTF-8");
        hints.put(EncodeHintType.ERROR_CORRECTION, ErrorCorrectionLevel.H);
        BitMatrix bitMatrix = new MultiFormatWriter().encode(url,BarcodeFormat.QR_CODE, QR_CODE_WIDTH, QR_CODE_HEIGHT,
        hints); //生成矩阵

        //插入logo
        BufferedImage image = new BufferedImage(QR_CODE_WIDTH,
QR_CODE_HEIGHT,BufferedImage.TYPE_INT_RGB);
        for (int x = 0; x < QR_CODE_WIDTH; x++) {
            for (int y = 0; y < QR_CODE_HEIGHT; y++) {
                image.setRGB(x, y, bitMatrix.get(x, y) ? 0xFF000000:0xFFFFFFFF);
            }
        }
        insertLogo(image); //不想插入图片可以注释掉此行

        String outPutPath = filePath + orgaCode + "." + picFormat;
        File outPutImage = new File(outPutPath);
        //如果图片不存在创建图片
        if(!outPutImage.exists())
            outPutImage.createNewFile();
    }
}

```

```

//输出图像
ImageIO.write(image, picFormat, outPutImage);

System.out.println(orgaCode+".PNG"+"已成功生成!");
}

/**
 * 插入logo图片
 * @param source
 * @throws Exception
 */
private static void insertLogo(BufferedImage source) throws Exception {
    BufferedImage logoImage = ImageIO.read(new File(logoPath));
    int width = logoImage.getWidth(null);
    int height = logoImage.getHeight(null);
    Image src =logoImage;
    Graphics2D graph = source.createGraphics();
    int x = (QR_CODE_SIZE - width) / 2;
    int y = (QR_CODE_SIZE - height) / 2;
    graph.drawImage(src, x, y, width, height, null);
    Shape shape = new RoundRectangle2D.Float(x, y, width, width, 6, 6);
    graph.setStroke(new BasicStroke(3f));
    graph.draw(shape);
    graph.dispose();
}

public static void main(String[] args) throws Exception {
    //单个二维码
    String orgaCode = "13417618112241138";
    QRCodeGenerator.createQRCodeImage(orgaCode);

    //多个二维码
    List<String> orgaCodeList = Arrays.asList("13417618112241138",
"23657618112241526", "11097618112247739");
    for(String code:orgaCodeList) {
        QRCodeGenerator.createQRCodeImage(code);
    }
}
}

```

1.3.5. 程序输出

1.3.5.1. 二维码图片（无 logo）



1.3.5.2. 二维码图片（有 logo）



注：图片仅为示意，根据实际情况选择图片

2. 扫码查询

使用认监委官方微信小程序（小程序名称：**有机码查询**）扫描二维码（无需手动输入有机码），系统展示符合查询条件的有机码及认证证书信息。

使用微信扫一扫，打开“有机码查询”小程序

