

NAME

qrencode - Encode input data in a QR Code and save as a PNG or EPS image.

SYNOPSIS

qrencode [-o FILENAME] [OPTION]... [STRING]

DESCRIPTION

Libqrencode is a library for encoding data in a QR Code symbol, a kind of 2D symbology that can be scanned by handy terminals such as a mobile phone with CCD. The capacity of QR Code is up to 7000 digits or 4000 characters, and has high robustness.

Qrencode is a utility software using libqrencode to encode string data in a QR Code and save as a PNG or EPS image.

OPTIONS

-h, --help

display help message.

-o FILENAME, --output=FILENAME

write image to FILENAME. If '-' is specified, the result will be output to standard output. If -S is given, structured symbols are written to FILENAME-01.png, FILENAME-02.png, ... (suffix is removed from FILENAME, if specified)

-r FILENAME, --read-from=FILENAME

read input data from FILENAME.

-s NUMBER, --size=NUMBER

specify the size of dot (pixel). (default=3)

-l {LMQH}, --level={LMQH}

specify error correction level from L (lowest) to H (highest). (default=L)

-v NUMBER, --symversion=NUMBER

specify the minimum version of the symbol. See SYMBOL VERSIONS for more information. (default=auto)

-m NUMBER, --margin=NUMBER

specify the width of margin. (default=4)

-d NUMBER, --dpi=NUMBER

specify the DPI of the generated PNG. (default=72)

-t

{PNG,PNG32,EPS,SVG,XPM,ANSI,ANSI256,ASCII,ASCIIi,UTF8,UTF8i,ANSIUTF8,ANSIUTF8i,ANSI256UTF8,ANSI256UTF8i}

--type={PNG,PNG32,EPS,SVG,XPM,ANSI,ANSI256,ASCII,ASCIIi,UTF8,UTF8i,ANSIUTF8,ANSIUTF8i,ANSI256UTF8,ANSI256UTF8i}

specify the type of the generated image. (default=PNG)

-S, --structured

make structured symbols. Version number must be specified with '-v'.

-k, --kanji

assume that the input text contains kanji (shift-jis).

-c, --casesensitive

encode lower-case alphabet characters in 8-bit mode. (default)

-i, --ignorecase

ignore case distinctions and use only upper-case characters.

-8, --8bit

encode entire data in 8-bit mode. -k, -c and -i will be ignored.

-M, --micro

encode in a Micro QR Code. See MICRO QR CODE for more information.

--rle

enable run-length encoding for SVG.

--svg-path

use single path to draw modules for SVG.

--inline

only useful for SVG output, generates an SVG without the XML tag.

--foreground=RRGGBB[AA]**--background=RRGGBB[AA]**

specify foreground/background color in hexadecimal notation. 6-digit (RGB) or 8-digit (RGBA) form are supported. Color output support available only in PNG, EPS and SVG.

--strict-version

disable automatic version number adjustment. If the input data is too large for the specified version, the program exits with the code of 1.

-V, --version

display the version number and copyrights of the qrencode.

--verbose

display verbose information to stderr.

[STRING]

input data. If it is not specified, data will be taken from standard input.

SYMBOL VERSIONS

The symbol versions of QR Code range from Version 1 to Version 40. Each version has a different module configuration or number of modules, ranging from Version 1 (21 x 21 modules) up to Version 40 (177 x 177 modules). Each higher version number comprises 4 additional modules per side by default. See <http://www.qrcode.com/en/about/version.html> for a detailed version list.

MICRO QR CODE

With Micro QR Code, You can embed data in a smaller area than with QR Code, but the data capacity is strongly limited. The symbol versions range from Version 1 to 4.

EXAMPLES

qrencode -l L -v 1 -o output.png 'Hello, world!'

encode into a symbol version 1, level L.

qrencode -iSv 1 --output=output.png

read standard input and encode it into a structured-appended symbols in case-insensitive mode.

qrencode -S -v 40 -l L -r bigfile.txt -o output.png

read input data from bigfile.txt and encode into a symbol version 40, level L.

AUTHOR

Written by Kentaro Fukuchi.

RESOURCES

Main Web Site: <https://fukuchi.org/works/qrencode/>

Source code repository: <https://github.com/fukuchi/libqrencode/>

COPYRIGHT

Copyright (C) 2006-2018 Kentaro Fukuchi.