

2D Barcode Fonts

QR Code Barcode

[*http://barcoderesource.com/qrcodebarcode.shtml*](http://barcoderesource.com/qrcodebarcode.shtml)

Copyright (c) 2009-2013, ConnectCode

All Rights Reserved.

ConnectCode accepts no responsibility for any adverse affect that may result from undertaking our training.

Microsoft and Microsoft Excel are registered trademarks of Microsoft Corporation. All other product names are trademarks, registered trademarks, or service marks of their respective owners

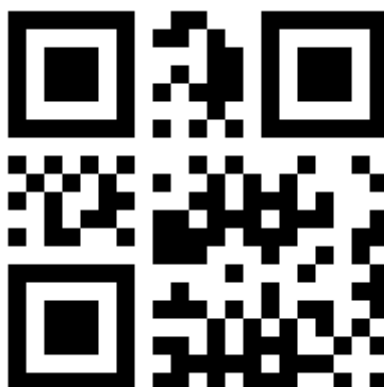
Table of Contents

1.	QR Code Barcode	iii
1.1	QR Code Barcode.....	iii
1.2	ConnectCode QR Code Barcode Font	iii
1.2.1	Data Compaction	iii
1.3	Parameters of the QR Code Barcode	iv
1.3.1	Error Correction.....	iv
1.3.2	Mask.....	iv
1.4	QR Code Barcode Fonts.....	iv
2.	Font Encoder	v
3.	.Net SDK	3-1
3.1	.Net Framework 4.0 Notes	3-2

1. QR Code Barcode

1.1 QR Code Barcode

The QR Code (Quick Response) barcode is a 2-dimensional barcode consisting of black square patterns on a white background. The barcode is capable of storing more information than a conventional barcode. It is developed by Denso-Wave in Japan and is one of the more popular 2-dimensional barcodes. Another reason for this barcode popularity is because it is adopted by many mobile or smartphone applications for linking physical world objects to a web URL (Uniform Resource Locator).



1.2 ConnectCode QR Code Barcode Font

This is a professional True Type (TTF) barcode font that is used to create a QR Code barcode by selecting a font in your favourite text editor. The package includes a standalone encoder, a .Net Dynamic Link Library (DLL), true type font for creating a QR Code barcode that strictly adheres to industry specifications.

1.2.1 Data Compaction

The QR Code is able to pack large amount of data using the various compaction methods. Each of the compaction method is optimized for a specific type of data. For example, the Numeric method is optimized for numbers. The ConnectCode encoder automatically scans through the data and detects the most optimized compaction method. On top of that, it also switches among the different compaction methods if one method is unable to fully encode the data.

- Numeric - Optimized for numbers.
- Alphanumeric - Optimized for numbers and alphabets. This compaction method is less optimized than Numeric.
- Binary - Optimized for any 8-bit binary data.
- Kanji - Optimized for Kanji data.

1.3 Parameters of the QR Code Barcode

The following sections detail the different configurable parameters of the QR Code barcode. If you are new to this barcode, it is recommended that you use the default or automatic settings mentioned below.

1.3.1 Error Correction

QR Code uses the Reed-Solomon error correction technique. This allows the barcode to be partially damaged without causing any loss of data. There are four different levels of error correction that can be chosen. The higher the level, the more resilient the barcode is to withstand damage. However the drawback is that more data codewords in the barcode are needed to store the error correction codewords instead of the actual data.

- Level L - 7% of codewords can be recovered
- Level M - 15% of codewords can be recovered
- Level Q - 25% of codewords can be recovered
- Level H - 30% of codewords can be recovered

1.3.2 Mask

QR Code's reliability can be improved by a method called masking. Masking regularizes the distribution of the black square patterns. Different types of masking patterns according to the specifications are supported by the ConnectCode QR Code. The default automatic masking option is recommended if you do not want to delve into the technical implementation of the barcode.

- Mask 0 – Mask Pattern 0
- Mask 1 – Mask Pattern 1
- Mask 2 – Mask Pattern 2
- Mask 3 – Mask Pattern 3
- Mask 4 – Mask Pattern 4
- Mask 5 – Mask Pattern 5
- Mask 6 – Mask Pattern 6
- Mask 7 – Mask Pattern 7
- Auto – Automatic Masking

1.4 QR Code Barcode Fonts

The following is the description of the QR Code barcode font used by the encoder or .Net DLL.

Font Name	Description	Recommended Sizes
CCodeQR (CCodeQR_Trial for the Trial version)	Standard QR Code Barcode Font.	Font Size 2..64

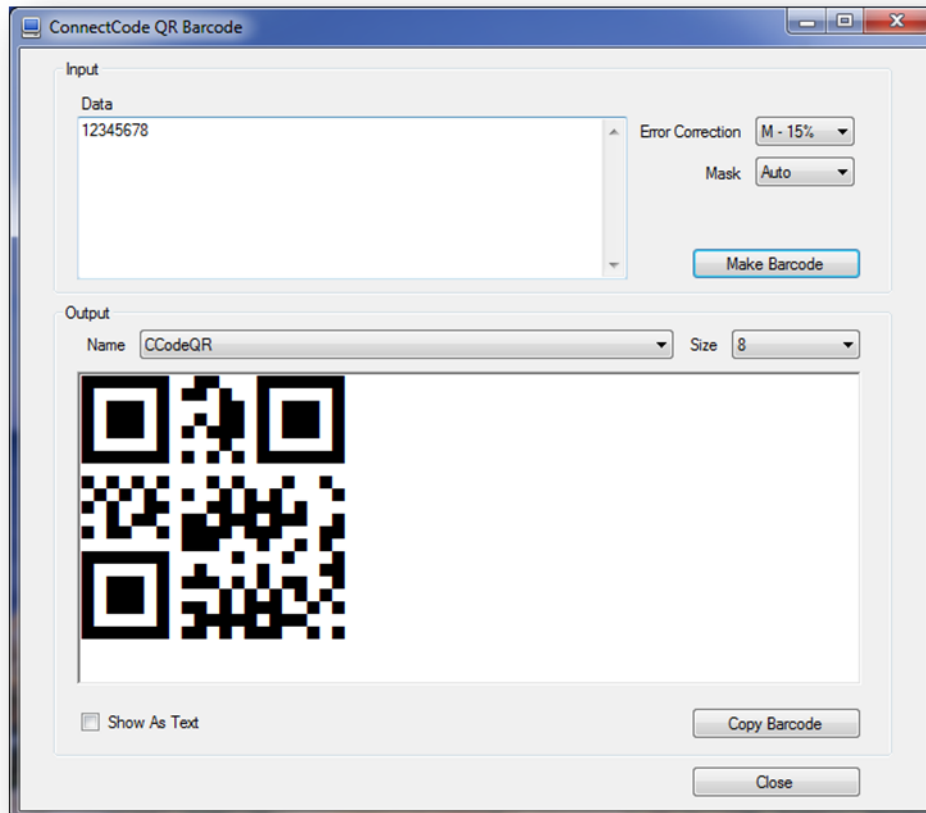
Note

1. You may see spaces between multiple Rows when you use the QR Code barcode fonts in certain font sizes. The spaces can be easily removed by increasing or decreasing the font size by 1 point.

2. Font Encoder

The QR Code Barcode Font package in ConnectCode comes bundled with a Font Encoder that allows you to encode the barcode quickly and easily. This is useful if you like to encode a single barcode to be pasted into your brochure, on packaging or product items. The Encoder supports all parameters as described in the sections above.

The Error Level and Mask are parameters that are supported (see previous section for detailed description)..



The Font Name and Font Size in the "Output" section can be changed after the QR Code barcode is created. This allows the height and the size of the barcode to be changed after the barcode is created.

The "Show As Text" option allows you to see the text output of the barcode in a normal text font. The "Copy Barcode" button allows the barcode to be copied and pasted onto other applications easily.

3. .Net SDK

A .Net Barcode SDK is also bundled in the ConnectCode QR Code Barcode Font package. This SDK can be bundled in your applications if you purchase the necessary distribution licenses.

Library Name

QRCode.dll

Namespace

ConnectCode.BarcodeFonts2D

Class Name

QRCode

Requirements

.Net 2.0 and onwards

Constructors and Functions

QR(String data, String errorcorrectionlevel, int mask);

This is the constructor for the QR barcode. It is used for initializing the QR barcode.

data : The data input string to be encoded as a barcode.
errorcorrectionlevel : The Error Correction Level "L", "M", "Q" or "H".
mask : Mask Pattern. Any number from 0..7 or -1 for Auto.

String Encode();

This function encodes the barcode based on the parameters specified in the constructor. The result will be returned as a string.

int LengthExceeded();

The Encode() function may return an empty output string either due to invalid inputs or the length of the data exceeded the length specified by the QR Code specifications. A call to this function after the Encode() function allows you to determine whether the data length has exceeded.

Sample Usage (C#)

```
Using ConnectCode.BarcodeFonts2D;
.
.
.
QR barcode = new QR("12345678","M",0);
String result = barcode.Encode();
Font font = new Font("CCodeQR", 8);
richTextBox1.Text = outputstr; //private System.Windows.Forms.RichTextBox richTextBox1;
richTextBox1.SelectAll();
richTextBox1.SelectionFont = font;
```

Sample Visual Studio Project

- Name - ConnectCode Encoder
- Solution Name - ConnectCode.sln
- Language - C#
- Requirements - .Net 2.0 and onwards, Visual Studio 2005, 2008, 2010, 2012 and onwards.

3.1 .Net Framework 4.0 Notes

.Net Framework 4.0 includes and uses CLR 4.0. It does not automatically use its version of the common language runtime to run applications that are built with earlier versions of .NET Framework. This is unlike .Net 2.0-3.5 where the framework uses CLR 2.0 to run applications. Basically, there is no version 3 of the CLR.

Hence, ConnectCode 2D Barcode SDK provides two sets of .Net DLLs for different versions of the .Net Framework as shown below:

For .Net 2.0 to 3.5 please use the DLLs and samples in

- /Resource subdirectory
- /.Net Samples subdirectory

For .Net 4.0 please use the DLLs and samples in

- /Net4 subdirectory
- /Net4/.Net Samples subdirectory