

2D Barcode Fonts

QR Code Barcode

<https://www.barcoderesource.com/qrcodebarcode.shtml>

Copyright (c) 2009-2018, 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

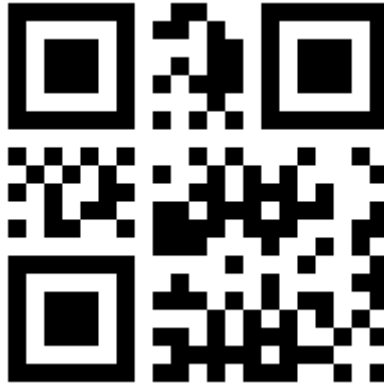
Table of Contents

| | | |
|-----------|---|-------------|
| 1. | QR Code Barcode | 1-2 |
| 1.1 | QR Code Barcode..... | 1-2 |
| 1.2 | ConnectCode QR Code Barcode Font | 1-2 |
| 1.2.1 | Data Compaction | 1-2 |
| 1.3 | Parameters of the QR Code Barcode | 1-3 |
| 1.3.1 | Error Correction..... | 1-3 |
| 1.3.2 | Mask..... | 1-3 |
| 1.4 | QR Code Barcode Fonts..... | 1-3 |
| 2. | Font Encoder | 2-4 |
| 3. | .NET SDK | 3-5 |
| 3.1 | .NET Framework 4.0 Notes | 3-6 |
| 4. | .NET Standard SDK..... | 4-7 |
| 5. | JavaScript SDK | 5-8 |
| 5.1 | QR Code Barcode with JavaScript and Barcode Web Fonts | 5-8 |
| 6. | Component Object Model Library | 6-9 |
| 6.1 | Tutorial on creating QR Code using COM | 6-10 |
| 7. | PowerBuilder | 7-14 |
| 7.1 | Tutorial on creating a QR Code in PowerBuilder | 7-14 |
| 8. | Crystal Reports UFL | 8-18 |
| 8.1 | Tutorial on creating QR Code with Crystal Reports UFL | 8-18 |

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 |

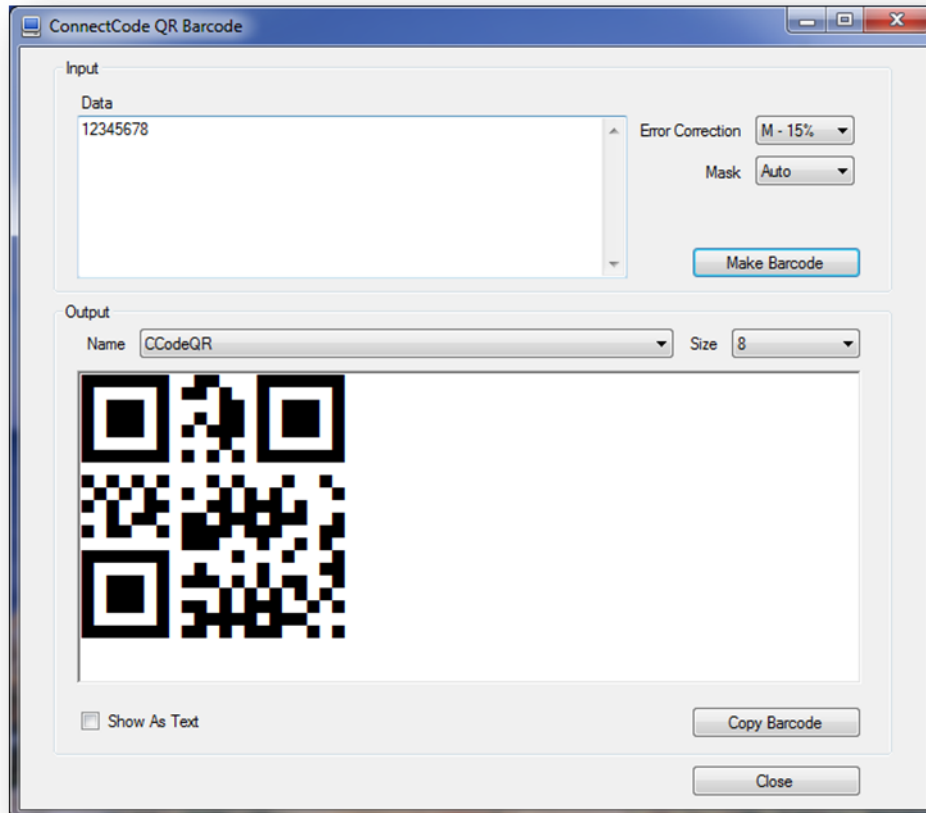
Note

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 8 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 2008, 2010, 2012, 2013, 2017 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

4. .NET Standard SDK

.NET Framework is a software framework that is developed by Microsoft to run primarily on Microsoft Windows. Over the years, the framework has been forked and enhanced to serve many different purposes. For example, the Universal Windows Platform uses a specific set of APIs from the .NET Framework to help programmers develop apps for the Windows Store. The .NET Core Framework also uses a subset of APIs from the .NET Framework to support the development of applications that can run on different operating systems such as Windows, Mac, and Linux.

.NET Standard is a specification of common APIs that are available on the different .NET frameworks. By creating a class library (DLL) that targets the .NET Standard, a developer can be assured that his library can be used or shared by projects developed on the various .NET frameworks.

As of .NET Standard v2.0, the .Net Standard specification is implemented by the following frameworks:

- .NET Core
- .NET Framework
- Mono
- Xamarin.iOS
- Xamarin.Android
- Universal Windows Platform

.NET Standard QR Code Library

- Resource/NETStandard/netstandard2.0/NETStandardQRCode.dll

The QR Code Barcode package includes a .NET Standard compliant QR Code class library (also available as a nuget) that targets the .NET Standard 2.0 specification. This library can be used by projects developed on different .NET frameworks and when used together with ConnectCode Barcode Fonts, generates barcodes of the highest quality that is able to meet the strictest requirements of the auto-id industry.

Sample ASP.NET Core Sample

- Resource/NETStandard/ASPNETCoreQRCodeSample

5. JavaScript SDK

This is an elegant solution that enables you to generate high quality QR Code barcodes using JavaScript with a HTML5 Canvas or World Wide Web Consortium (W3C) compliant Web Fonts. The generated QR Code barcode is able to meet the strictest industry requirements required by the auto-id industry.

JavaScript QR Code Resource Folder

- Resource/Javascript

The snippet below illustrates the use of the HTML5 Canvas tag on a HTML page.

```
<canvas id="barcodeCanvas" width=300 height=300>12345678</canvas>
```

The following JavaScript code shows how to render a QR Code barcode on the HTML5 Canvas.

```
var elementBarcode = document.getElementById("barcodeCanvas");  
var barcode = new QRCode(elementBarcode.innerHTML, "L", 8);  
barcode.drawOnCanvas("barcodeCanvas");
```

Parameters

Input Data: elementBarcode.innerHTML (or any other input string)

Error Correction Level: "L" ("L", "M", "Q" or "H")

L - Allows recovery of up to 7% data loss
M - Allows recovery of up to 15% data loss
Q - Allows recovery of up to 25% data loss
H - Allows recovery of up to 30% data loss

Mask: 8 (0 to 7 or 8 for Auto)

The purpose of a mask pattern is to make the QR code easier for QR scanner to read.

5.1 QR Code Barcode with JavaScript and Barcode Web Fonts

The output generated by the JavaScript library can also be rendered as a QR Code Barcode through the use of a Web Open Font Format Font (WOFF). WOFF is an optimized font format recommended by World Wide Web Consortium (W3C) for use in web pages. It uses compression on Open Type or True Type fonts to achieve file size reduction so that it can be efficiently distributed over the web.

The QR Code Barcode WOFF fonts provided by ConnectCode have been tested vigorously to display and print on different desktop and mobile browsers. It is important to know that a font raster to the output device and is not limited to DPI (Dots per Inch) of the computer screen. This enables very high quality QR Code barcodes to be generated.

The font solution for generating barcodes is based on ConnectCode's True Type barcode font engine that has passed numerous independent audits and is widely adopted by many Fortune 500 companies. QR Code fonts in Embedded Open Type (EOT), and Open Type (OTF) format are also provided to ensure that the solution works on legacy browsers that have yet to fully support WOFF.

6. Component Object Model Library

This tutorial illustrates the use of a COM (Component Object Model) object library with a True Type Font (QR Code Barcode Font), provided in ConnectCode QR Code package, to create a ISO/IEC 18004:2015 standard-compliant QR Code in a .NET Windows Form application.

Prerequisites

- ConnectCode [QR Code](#) package is installed
- QRCodeCOMLibrary.dll in the Resource\QRCodeCOMLibrary subdirectory of ConnectCode QR Code package. The QR Code class library has been compiled with the "Register for COM interop" Visual Studio project property, exposing a COM-callable wrapper that enables COM interaction.
- Visual Studio 2015/2017
- Administrator Rights

6.1 Tutorial on creating QR Code using COM

1. Launch the Visual Studio Developer Command Prompt as Administrator from the Windows Start Menu.
2. In the Developer Command Prompt, use the "cd" command to go to the QR Code COM Library folder.

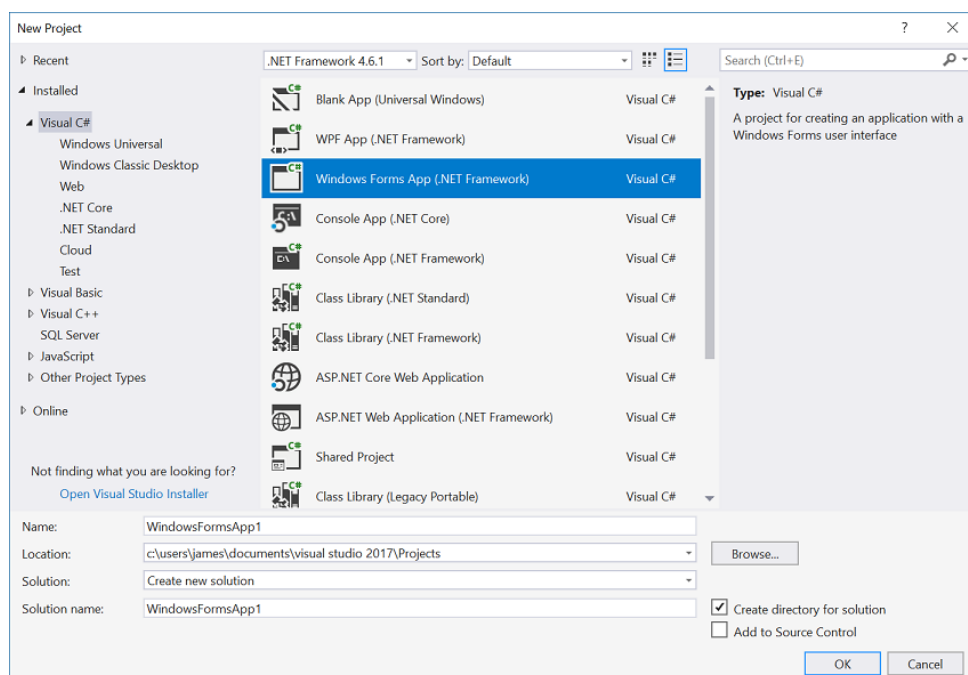
```
cd C:\Program Files (x86)\ConnectCodeQRCode\Resource\QRCodeCOMLibrary  
(or ConnectCodeQRCodeTrial if you are using the Trial package)
```

3. Enter the following command in the Developer Command Prompt to use Regasm.exe to register the QRCode COMLibrary assembly for use with COM.

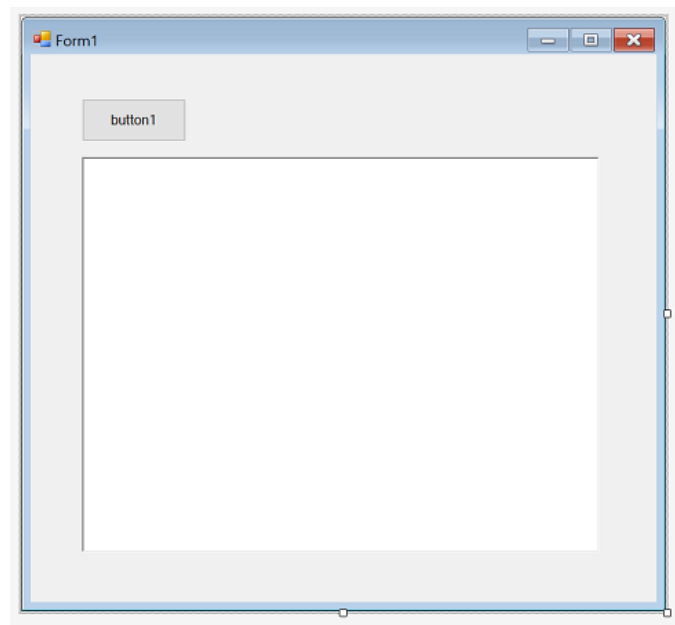
```
Regasm QRCodeCOMLibrary.dll /tlb:QRCodeCOMLibrary.tlb /codebase
```

Regasm.exe adds information about the class to the system registry so that COM clients can use the .NET Framework class transparently. The tlb option generates a type library defined within the assembly.

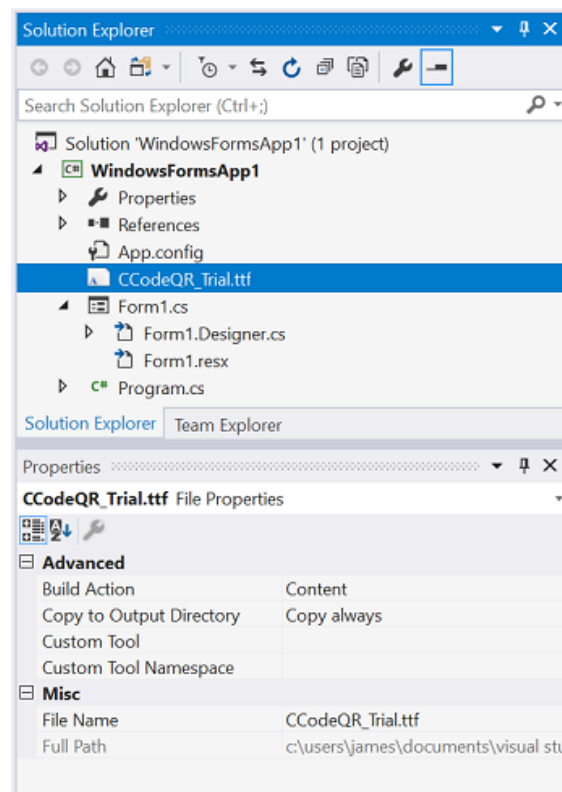
4. Launch Visual Studio. Create a new Windows Form project by clicking on "File->New Project", select a "Windows Forms App" and click on "Create" button.



5. Double click on "Form1.cs" in the "Solution Explorer" and add a "Button" and a "RichTextBox" from the Visual Studio Toolbox. You should see your Windows Form similar to the screenshot below.



6. Right click on the "WindowsFormsApp1" project in the "Solution Explorer" and select "Add->Existing Item". Navigate to the "C:\Program Files (x86)\ConnectCodeQRCode\" folder and select the "CCodeQR.ttf" font. If you are using the trial version, select the "CCodeQR_Trial.ttf" font instead.



7. In "Solution Explorer", select the "CCodeQR.ttf" font and change the "Build Action" to "Content" and "Copy to Output Directory" to "Copy Always" in the Properties pane. This will ensure that Visual Studio deploy the QR Code barcode font for use with the Windows Form application.

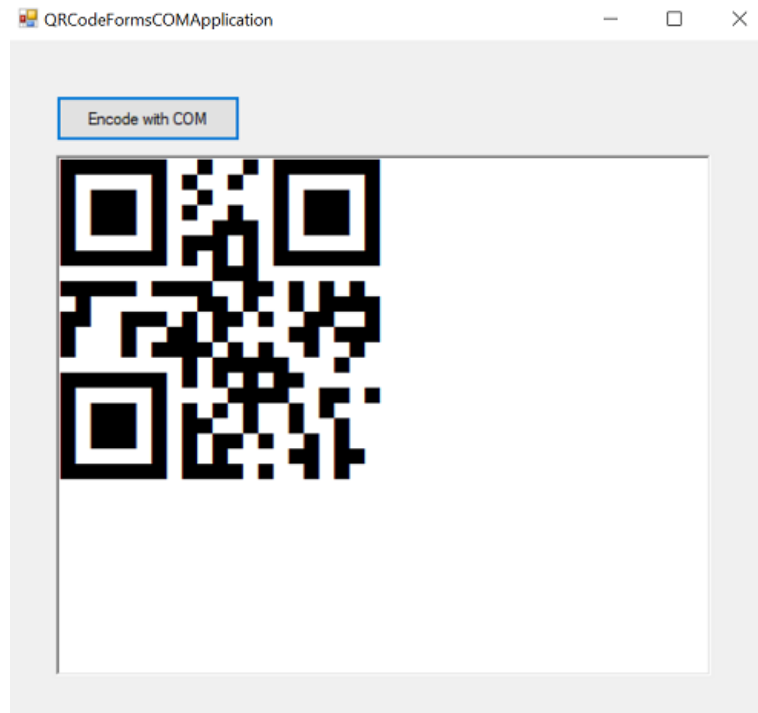
8. Double click on "Form1.cs" in the "Solution Explorer". In the designer, double click on the Button. This will generate the button1_Click function in the editor. Enter the C# programming codes as shown below:

```
using System.Drawing.Text;
.
.
.
.
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        Type comObjectType =
            Type.GetTypeFromProgID("Net.ConnectCode.QRCodeCOMLibrary");
        dynamic theComObject = Activator.CreateInstance(comObjectType, false);

        //or
        //Guid myGuid = new Guid("5E206D5A-D9C2-45AA-BBFD-2E9B36AD437D");
        //Type comObjectType = Type.GetTypeFromCLSID(myGuid);
        //dynamic theComObject = Activator.CreateInstance(comObjectType, false);

        string inputData = "12345678";
        string ecl = "L"; //L, M, Q or H
        int mask = 8; //0 to 7 or 8 for Auto
        string result1 = theComObject.Encode_QRCode(inputData,ecl,mask);
        string result = result1;
        richTextBox1.Text = result;
        System.Diagnostics.Debug.WriteLine(result1);
        PrivateFontCollection pfc = new PrivateFontCollection();
        pfc.AddFontFile("CCodeQR.ttf"); //pfc.AddFontFile("CCodeQR_Trial.ttf");
        richTextBox1.Font = new Font(pfc.Families[0], 8, FontStyle.Regular);
    }
    catch (Exception ex)
    {
        System.Diagnostics.Debug.WriteLine(ex);
    }
}
```

The C# function above creates a QR Code COM object and then uses it to generate a QR Code barcode with the input data "12345678". The result is placed in "richTextBox1" and the final output is displayed with the CCodeQR True Type font. When you run the application, click on the "Encode with COM" button, you should see the QR Code barcode as shown below.



The "QRCodeCOMApplication" folder in "C:\Program Files (x86)\ConnectCodeQRCode\Resource" contains the full source code of the above application.

7. PowerBuilder

This tutorial illustrates the use of a COM (Component Object Model) library and a barcode font available in ConnectCode QR Code package for creating QR Code barcode in PowerBuilder. The generated QR Code complies with the ISO/IEC 18004:2015 standards and is able to meet the strictest requirements of the Auto-ID industry.

Prerequisites

- PowerBuilder v12 (or APPEON PowerBuilder 2017. In 2016, SAP and Appeon has entered into an agreement whereby Appeon would be responsible for developing and marketing PowerBuilder.)
- ConnectCode QR Code package is installed

7.1 Tutorial on creating a QR Code in PowerBuilder

1. Launch a Windows Command Prompt as Administrator. In the Command Prompt, enter the following command to go to the QRCodeCOMLibrary folder.

```
cd C:\Program Files(x86)\ConnectCodeQRCode\Resource\QRCodeCOMLibrary
```

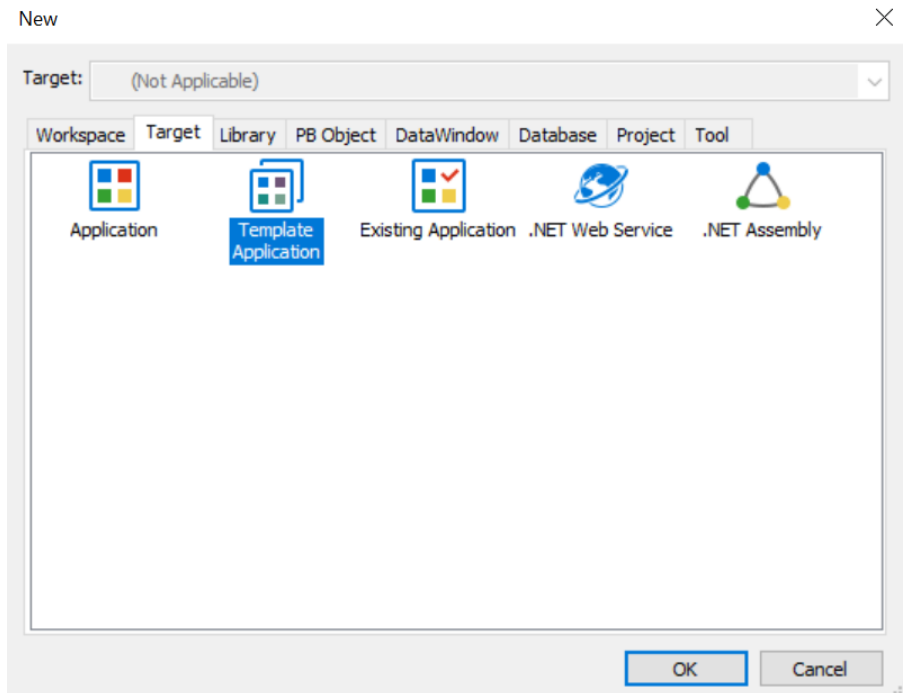
2. Next, use the Assembly Registration Tool (Regasm.exe) to register the QRCodeCOMLibrary.dll assembly with COM.

```
Regasm QRCodeCOMLibrary.dll /tlb:QRCodeCOMLibrary.tlb /codebase
```

If Regasm is not available, you can verify if the following folder exists and add the folder into your PATH. The folder may be different depending on your version of .NET.

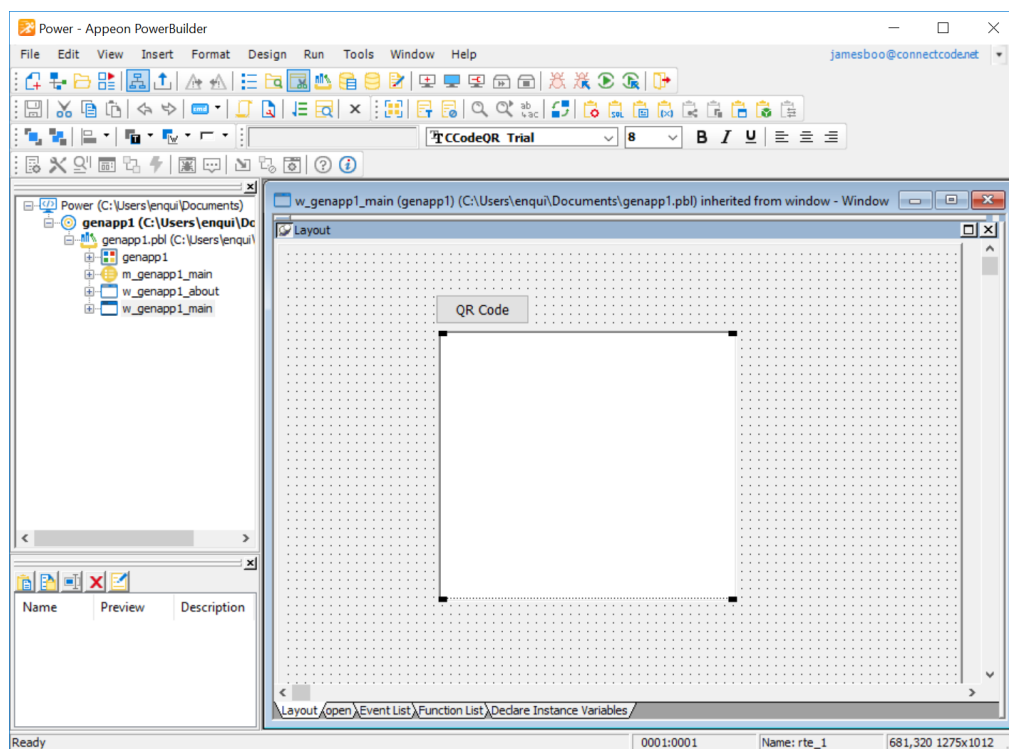
```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\
```

3. Launch PowerBuilder and create a new Template Application in the Target tab.



4. You can select "SDI application" as the "Application Type" and "None" in Connectivity Options" to create a sample application.

5. Click on w_genapp_main when the application is created. From the Menu select "Insert->Control->CommandButton" and rename the button to "QR Code". Next, insert a TextEdit control and layout the components as shown in the screenshot below.



Select the TextEdit control, change the Font to **CCodeQR** (or CCodeQR_Trial) and the Font Size to 8 to fit the barcode nicely on the TextEdit control. The output generated by the QRCodeCOMLibrary will be placed in the TextEdit control and displayed as a barcode with the CCodeQR True Type font.

6. Next, double click on the button and add the following script:

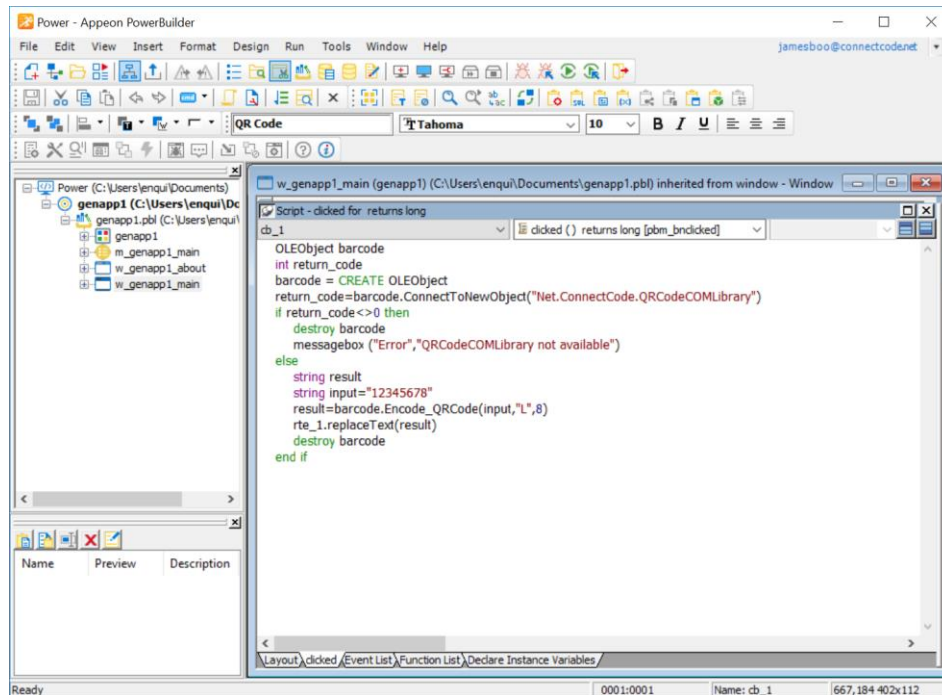
```
OLEObject barcode
int return_code
barcode = CREATE OLEObject
return_code=barcode.ConnectToNewObject("Net.ConnectCode.QRCodeCOMLibrary")
if return_code<>0 then
    destroy barcode
    messagebox ("Error","QRCodeCOMLibrary not available")
else
    string result
    string input="12345678"
    result=barcode.Encode_QRCode(input,"L",8)
    rte_1.replaceText(result)
    destroy barcode
end if
```

In the source code above, an OLE object is created with the QRCodeCOMLibrary. The "Encode_QRCode" method is used to generate the barcode.

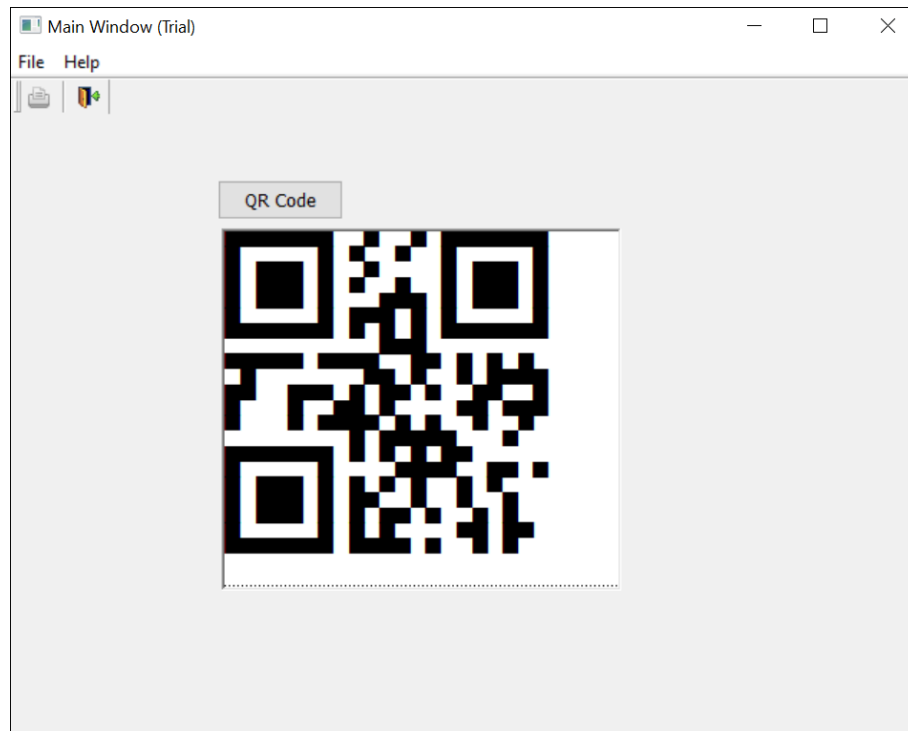
Error Correction Level: "L" ("L", "M", "Q" or "H")

- L - Allows recovery of up to 7% data loss
- M - Allows recovery of up to 15% data loss
- Q - Allows recovery of up to 25% data loss
- H - Allows recovery of up to 30% data loss

Mask: 8 (0 to 7 or 8 for Auto)



7. Run the application by going to the menu and select "Run->Select and Run". Click on the "QR Code" button and see that you get the QR Code barcode output in the TextEdit control. If you get an error message saying that "QRCodeCOMLibrary is not available", check that you have carried step 2 successfully.



8. Crystal Reports UFL

This tutorial illustrates the use of a UFL (User Function Library for Crystal Reports) with a True Type Font (QR Code Barcode Font), provided in ConnectCode QR Code package, to create a ISO/IEC 18004:2015 standard-compliant QR Code barcode in Crystal Reports. A User Function Library is a dynamic link library that enables Crystal Reports to add customized functions to Formula Workshop.

Prerequisites

- ConnectCode QR Code package is installed
- CRUFL_QRCodeBarcode.dll in the Resource\CrystalReportsUFL subdirectory of ConnectCode QR Code package.
- Crystal Reports 2016 (the UFL works on earlier versions as well)
- x86 Native Tools Command Prompt
- Administrator Rights

8.1 Tutorial on creating QR Code with Crystal Reports UFL

1. Launch Windows Explorer and go to the ConnectCode QR Code package folder. The QR Code package is installed in "C:\Program Files (x86)\ConnectCodeQRCode" folder by default.

In the "Resource\CrystalReportsUFL" subfolder, locate the "CRUFL_QRCodeBarcode.dll" and "crw32.exe.config" files, and copy the DLL to the Crystal Reports Library folder. The Crystal Reports Library folder is located in a folder similar to the one below.

```
C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win32_x86
```

The "crw32.exe.config" file contains the following to enable mixed-mode assembly.

```
<?xml version="1.0"?>
<configuration>
  <startup useLegacyV2RuntimeActivationPolicy="true" >
    <supportedRuntime version="v4.0" />
  </startup>
</configuration>
```

2. Launch the "x86 Native Tools Command Prompt for VS 2017" (the 32-bit prompt is required for loading a UFL to Crystal Reports) as Administrator from the Windows Start Menu.

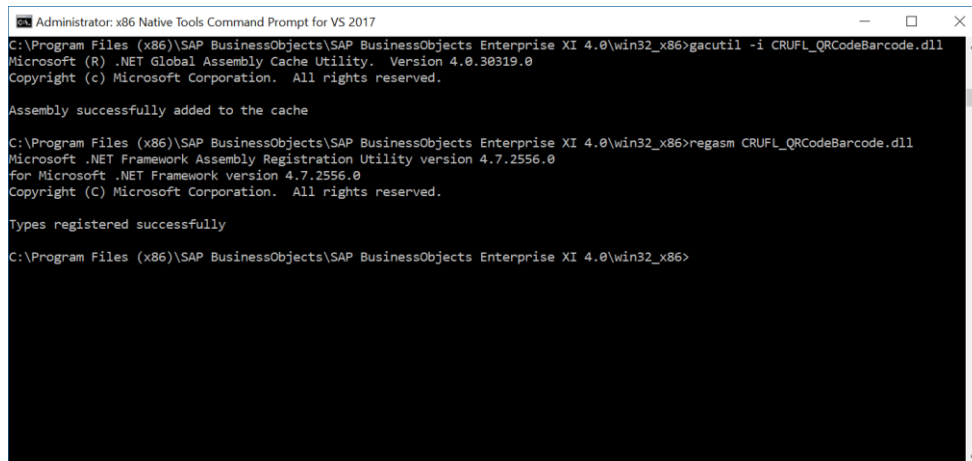
3. In the Command Prompt, use the "cd" command to go to the Crystal Reports Library folder.

```
cd C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win32_x86
```

4. Enter the following command in Command Prompt to load the UFL.

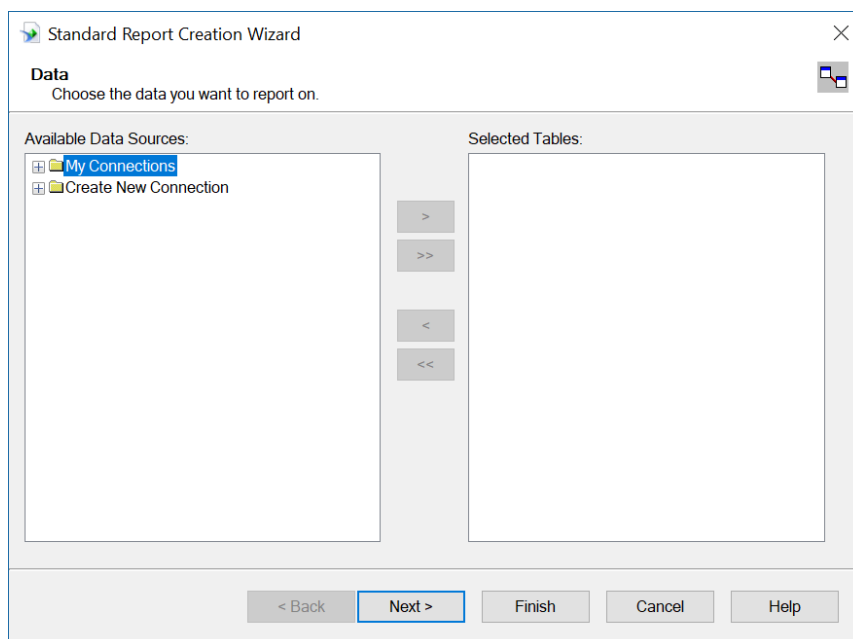
```
gacutil -i CRUFL_QRCodeBarcode.dll  
Regasm CRUFL_QRCodeBarcode.dll
```

Ensure that the UFL is loaded successfully as shown in the screenshot below.

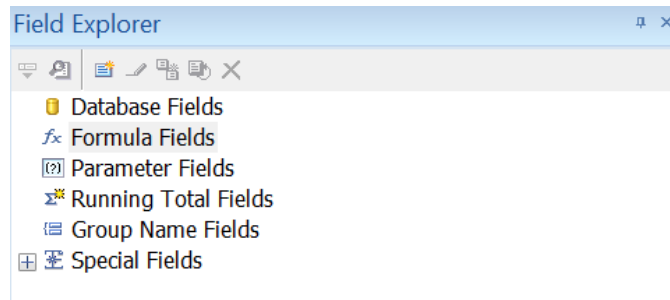


```
Administrator: x86 Native Tools Command Prompt for VS 2017  
C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win32_x86>gacutil -i CRUFL_QRCodeBarcode.dll  
Microsoft (R) .NET Global Assembly Cache Utility. Version 4.0.30319.0  
Copyright (c) Microsoft Corporation. All rights reserved.  
  
Assembly successfully added to the cache  
  
C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win32_x86>regasm CRUFL_QRCodeBarcode.dll  
Microsoft .NET Framework Assembly Registration Utility version 4.7.2556.0  
for Microsoft .NET Framework version 4.7.2556.0  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Types registered successfully  
  
C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win32_x86>
```

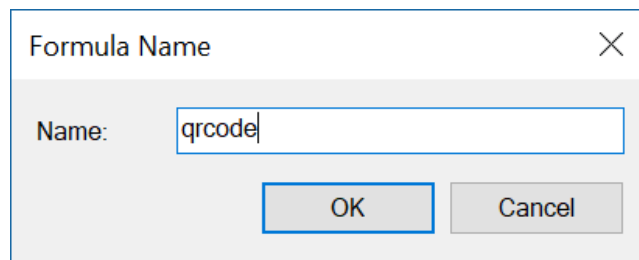
5. Launch Crystal Report and create a new "Standard Report". Click on the Finish button when prompted to select the data that you want to report on.



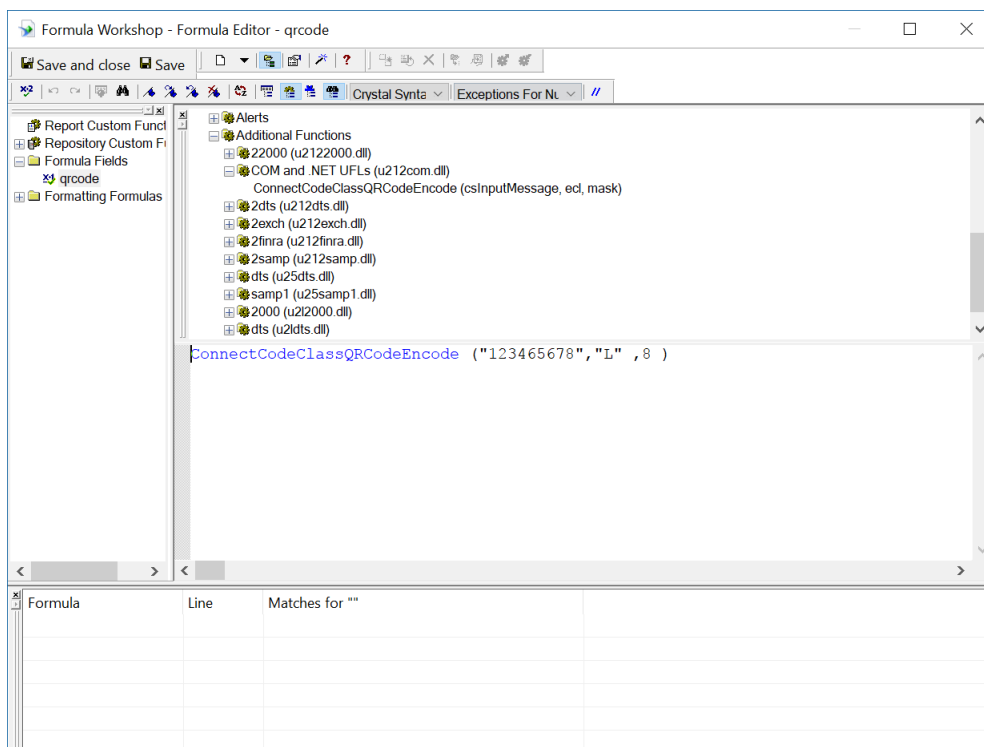
6. When the report is created, right click on "Formula Fields" in the "Field Explorer" and select "New" to create a new formula.



7. Name the formula as "qrcode".



8. In the "Formula Workshop", expand "Functions->Additional Functions->COM and .NET UFLs (u212com.dll)". Check that you see the "ConnectCodeClassQRCodeEncode" formula. If you do not see this formula, please ensure that you have run steps 2-4 successfully.



9. Double click on the formula, change "Crystal Syntax" to "Basic Syntax" and enter the following VBA programming codes below:

```
ConnectCodeClassQRCodeEncode("12345678","L",8)
Dim x As Number
Dim Result As String
For x = 1 To ConnectCodeClassQRCodeNumBlocks()
    Result=Result + ConnectCodeClassQRCodeGetBlocks(x)
Next x
formula = Result
```

The formula uses "12345678" as the input data, "L" as the error correction level and 8 as the Mask.

Error Correction Level: "L" ("L", "M", "Q" or "H")

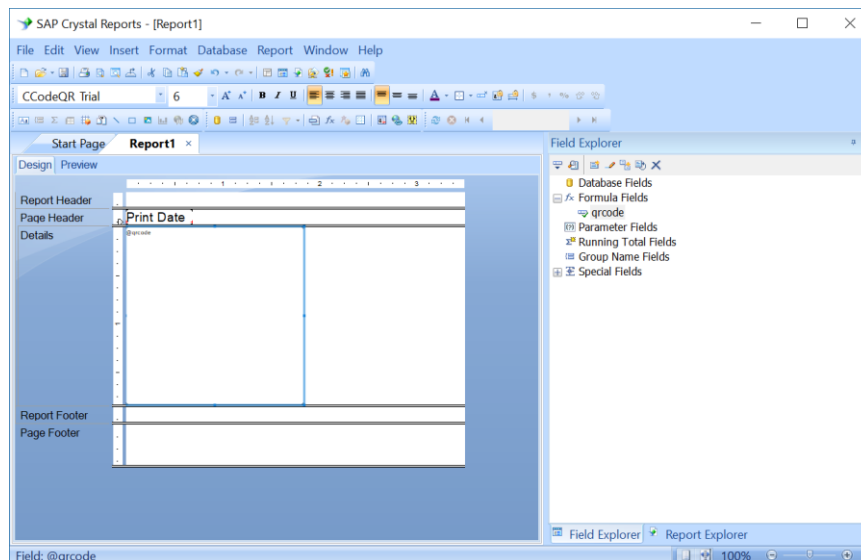
L - Allows recovery of up to 7% data loss
M - Allows recovery of up to 15% data loss
Q - Allows recovery of up to 25% data loss
H - Allows recovery of up to 30% data loss

Mask: 8 (0 to 7 or 8 for Auto)

The purpose of a mask pattern is to make the QR code easier for QR scanner to read.

After the QR Code is encoded, a loop is required to return the output in blocks. The reason is because Crystal Reports enforces a 255 characters length limit on the output returned by a UFL formula.

10. When ready, click on the "Save and close" button. In the designer, drag the "qrcode" formula onto the report.



On the Design tab, select the object created and change the font to "CCodeQR" (or "CCodeQR_Trial"). Change the Font Size to 6 to fit the barcode nicely on the report.

11. Click on "View->Print Preview" to preview the report with the QR Code barcode.

