

SSRS Reporting With QR Codes

Table of Contents

SSRS Reporting With QR Codes.....	1
Legal Notice.....	2
Overall Description.....	4
User Needs.....	4
Assumptions and Dependencies.....	4
System Features and Requirements.....	4
External Interface Requirements.....	4
INTRODUCTION.....	5
General Overview.....	5
Purpose.....	5
Intended Audience.....	5
Intended Use.....	5
Scope.....	5
References.....	6
Installation.....	7
Usage.....	8

Overall Description

QR code printing is not a part of SSRS. External, third-party plugins must be used for this functionality. Typically, QR code printing on SSRS reports is to produce a link, to email a subscriber, to pull a copy of a report. The functionality required in this particular project is to print a QR code to access a link to record the use of a credit.

User Needs

Users must have a QR code reader. This may be on their smartphone. The smartphone must be internet enabled to follow the link to process the credit.

Assumptions and Dependencies

For this feature to work in SSRS, an assembly—DLL—must be uploaded into the 'bin' directory of SQL Services Report Manager.

System Features and Requirements

External Interface Requirements

QRCoder.dll from the QRCoder package must be installed.

The level of .Net must be matched with the appropriate dll.

References

<https://github.com/codebude/QRCoder> -- QRCoder is a simple library, written in C#.NET, which enables you to create QR codes. It hasn't any dependencies to other libraries and is available as .NET Framework and .NET Core PCL version on NuGet.

This open source software was used for this project.

<https://www.youtube.com/watch?v=BW99WK6SQKI> -- This YouTube video fully explains all the necessary steps to install and use QRCoder with an SSRS report.

Installation

1. Determine the version of .Net you are using.

From a PowerShell Prompt type:

```
reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NET Framework Setup\NDP"
```

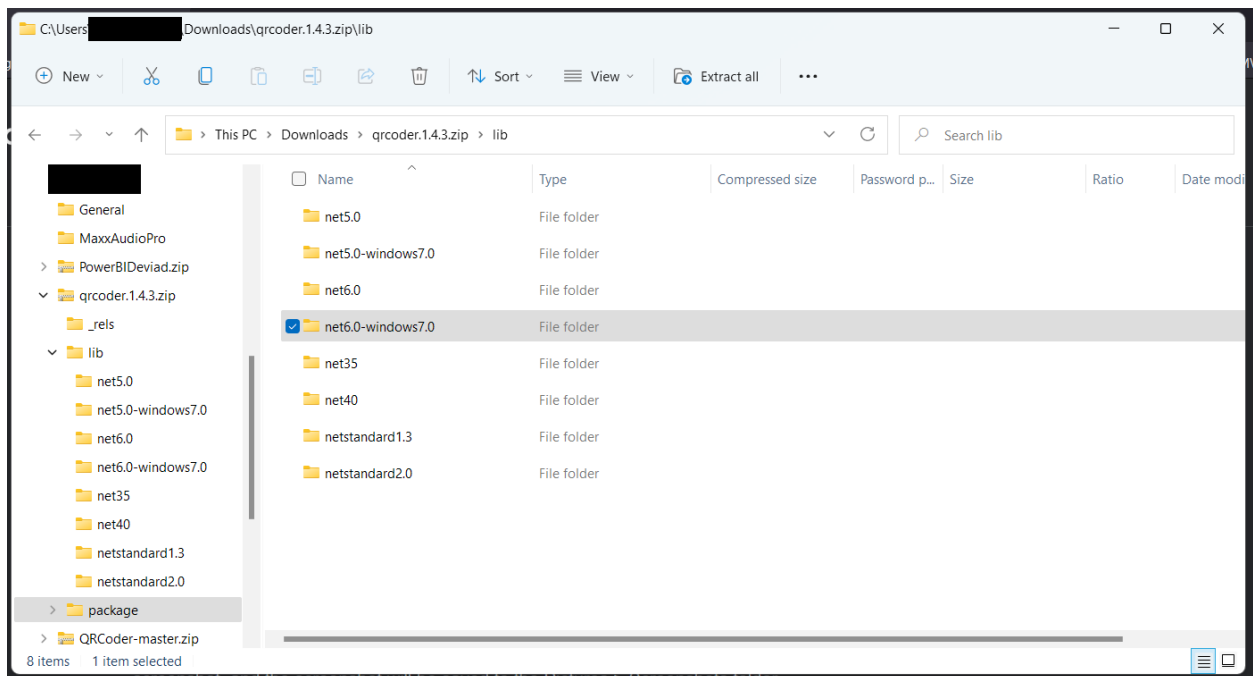
2. Download the QRCode package either from the github repository or; use nuget.org to get a copy of the distro. Rename the nuget package to a ".zip" file.

<https://github.com/codebude/QRCode>

<https://www.nuget.org/packages/QRCode/>

3. From within the lib directory of the distro, locate the directory for the appropriate version of QRCode.dll

C:\Users\...\Downloads\qrcode.1.4.3.zip\lib



4. Copy the dll to the "bin" directory in SSRS or Visual Studio, depending on your needs.

For Visual Studio, the path is:

C:\Program Files\Microsoft Visual Studio\2022\Community\Common7\IDE\PrivateAssemblies

Usage

1. In SSRS, create a report.

In Report Properties, add a reference for the QRCode.dll

(Locate this with the path that you used to install it in SSRS or Visual Studio.)

2. In report properties, add the following code:

```
Public Function QRCodeByteArray(ByVal inputString As String) as Byte()
```

```
    Return
```

```
    QRCode.BitmapByteQRCodeHelper.GetQRCode(inputString,QRCode.QRCodeGenerator.ECCLevel.Q,20)
```

```
End Function
```

3. At this point, save the report to make sure the setup is fine. If there is an error, the report will not save.

4. Add an image field.

Image Source: Database

MIME Type: image/png

Expression:

```
=Code.QRCodeByteArray("https://www.raptns.com/linkToWhatever")
```

NOTE: This code passes a string to QRCode. This string is converted into a QR Code. This must be a string surrounded by quotes.

Since the method is expecting a string as a parameter, a database field by itself will not work. The database field must be embedded in a string. Ex. "This is a message about "+fieldName

For the purposes of this project, the message is used with POLink embedded in the text.

Ex. "This is..." + polink + "this is the end of the message."