

2D Barcode Fonts

DataMatrix Barcode

<http://barcoderesource.com/datamatrixbarcode.shtml>

Copyright (c) 2009-2022, 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. | DataMatrix Barcode | 1-1 |
| 1.1 | DataMatrix Barcode | 1-1 |
| 1.2 | ConnectCode DataMatrix Barcode Font package | 1-1 |
| 1.2.1 | Data Encodation..... | 1-1 |
| 1.2.2 | Error Correction..... | 1-1 |
| 1.2.3 | Shape..... | 1-2 |
| 1.2.4 | Prefix | 1-4 |
| 1.2.5 | DataMatrix Barcode Font | 1-4 |
| 2. | Font Encoder | 2-5 |
| 3. | Crystal Reports UFL | 3-6 |
| 3.1 | Tutorial on creating DataMatrix with Crystal Reports UFL..... | 3-7 |
| 4. | Component Object Model Library | 4-13 |
| 4.1 | Tutorial on creating DataMatrix using COM..... | 4-14 |
| 5. | PowerBuilder | 5-18 |
| 5.1 | Tutorial on creating a DataMatrix in PowerBuilder | 5-18 |
| 6. | .NET Standard | 6-23 |
| 7. | .NET SDK | 7-24 |
| 7.1 | .NET Framework 4.0 Notes | 7-25 |

1. DataMatrix Barcode

1.1 DataMatrix Barcode

DataMatrix is a two-dimensional matrix symbology made up of square modules arranged in a square or rectangle shaped pattern. This barcode can pack large amount of data and has the capability of recovering the original data encoded even when it is partially damaged through an error correction technique. The DataMatrix barcode is composed of solid adjacent patterns known as the "finder patterns" in L-shape for locating and orienting the symbol.



1.2 ConnectCode DataMatrix Barcode Font package

This is a professional True Type (TTF) barcode font package for creating DataMatrix barcodes by using barcode fonts. The package includes a standalone Font encoder, a .NET Dynamic Link Library, Crystal Reports UFL, Component Object Model (COM) Library, PowerBuilder COM Library, and True Type fonts for creating a DataMatrix barcodes that strictly adheres to ISO/IEC 16022:2006 (ECC200).

1.2.1 Data Encodation

The DataMatrix Barcode can pack large amount of data using various encodation methods. Each encodation method is optimized for a specific type of data. For example, the C40 method is optimized for upper case alphanumeric characters.

The list below shows the different encodation methods specified in the DataMatrix specification.

- ASCII - ASCII 0 - 127
- C40 - Upper case alphanumeric
- Text - Lower case alphanumeric
- X12 - ANSI X12 EDI Data Set
- EDIFACT - AASCII 32 - 94
- Base 256 - Byte values 0 - 255

1.2.2 Error Correction

The DataMatrix barcode uses the Reed-Solomon error correction technique for data recovery. This allows a partially damaged barcode to be read and decoded. A partially damaged barcode could be the result of a partially torn printed barcode. The original data encoded in the barcode could be retrieved by encoding additional codewords in the barcode for redundancy and error correction.

1.2.3 Shape

The DataMatrix barcode supports 24 square symbols and 6 rectangular symbols. The sizes of the different symbols are shown below.

- Square

| Square | Row | Column |
|--------|-----|--------|
| | 10 | 10 |
| | 12 | 12 |
| | 14 | 14 |
| | 16 | 16 |
| | 18 | 18 |
| | 20 | 20 |
| | 22 | 22 |
| | 24 | 24 |
| | 26 | 26 |
| | 32 | 32 |
| | 36 | 36 |
| | 40 | 40 |
| | 44 | 44 |
| | 48 | 48 |
| | 52 | 52 |
| | 64 | 64 |
| | 72 | 72 |
| | 80 | 80 |
| | 88 | 88 |
| | 96 | 96 |
| | 104 | 104 |
| | 120 | 120 |
| | 132 | 132 |
| | 144 | 144 |

The diagrams below illustrate the 12 x 12, 32 x 32 and 64 x 64 DataMatrix symbols.



12x12



32 x 32



64 x 64

- Rectangle

| Rectangle | Row | Column |
|-----------|-----|--------|
| | 8 | 18 |
| | 8 | 32 |
| | 12 | 26 |
| | 12 | 36 |
| | 16 | 36 |
| | 16 | 48 |

The diagram below illustrates an **8 x 18** Rectangular DataMatrix barcode.



8 x 18

1.2.4 Prefix

The DataMatrix barcode supports the following prefixes in the symbol. A prefix is used to adhere to barcode industry standards such as one authorized by AIM International/GS1 or for abbreviating a header to the symbol.

- None
- FNC1 – GS1 FNC1 character
- 05 Macro – Macro codeword 236
- 06 Macro – Macro codeword 237
- Reader Programming – Indicates the symbol encodes a message used to program the reader system

1.2.5 DataMatrix Barcode Font

The following is the description of the DataMatrix Barcode Font used by the Encoder or .Net DLL.

| Font Name | Description | Recommended Sizes |
|---|-----------------------------------|-------------------|
| CCodeDataMatrix (CCodeDataMatrix_Trial for the Trial version) | Standard DataMatrix Barcode Font. | Font Size 2..64 |

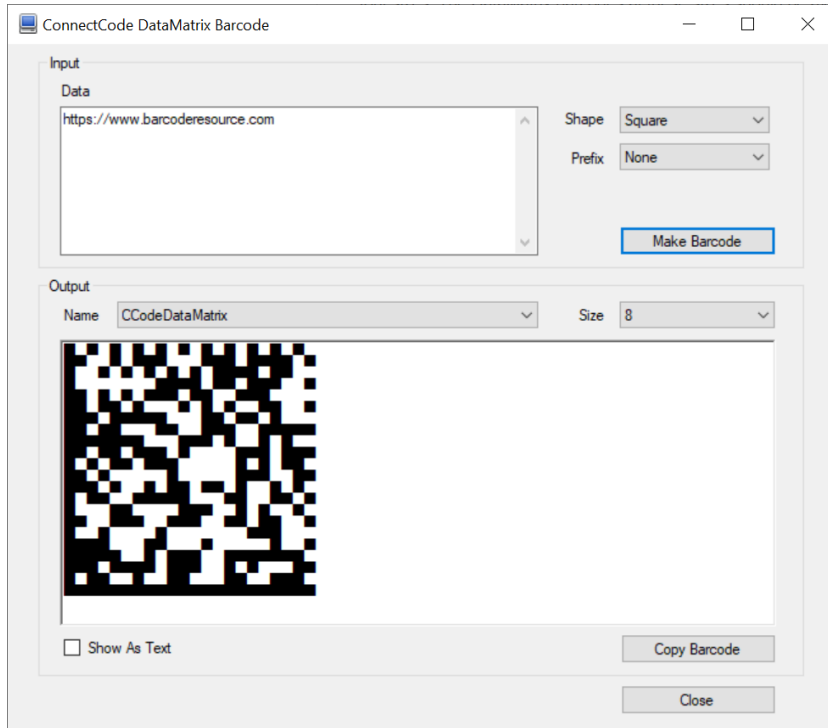
Note

1. You may see spaces (though rare) between multiple Rows when you use the DataMatrix Code barcode fonts in certain font sizes. The DataMatrix barcodes at these sizes should be avoided. The spaces can be easily removed by increasing or decreasing the font size by 1 point.

2. Font Encoder

ConnectCode's DataMatrix barcode font package 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 on your brochure or product items. The Encoder supports all parameters as described in the sections above.

The Shape and Prefix parameters can be easily specified through the user interfaces shown below.



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

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 to other applications easily.

3. Crystal Reports UFL

This tutorial illustrates the use of a UFL (User Function Library for Crystal Reports) with a True Type Font (DataMatrix Barcode Font), provided in ConnectCode DataMatrix package, to create an ISO/IEC 16022:2006 compliant DataMatrix 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

Crystal Reports 2016

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

Crystal Reports 2020

- Crystal Reports 2020
- ConnectCode DataMatrix package is installed
- CRUFL_DataMatrixBarcode.dll in the Resource\CrystalReportsUFL\x64 subdirectory of ConnectCode DataMatrix package.
- x64 Native Tools Command Prompt
- Administrator Rights

Note – x86/x64 Native Tools Command Prompt is available in the Desktop Development with C++ Workload of Visual Studio.

3.1 Tutorial on creating DataMatrix with Crystal Reports UFL

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

Crystal Reports 2016

In the "Resource\CrystalReportsUFL" subfolder, locate the "CRUFL_DataMatrixBarcode.dll" and "crw32.exe.config" files, and copy the DLL to the Crystal Reports Library folder. The Crystal Reports Library folder is in a folder like 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>
```

Crystal Reports 2020

In the "Resource\CrystalReportsUFL\x64" subfolder, find the "CRUFL_DataMatrixBarcode.dll" and copy it to the Crystal Reports Library folder. The Crystal Reports Library folder is in a folder like the one below.

```
C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win64_x64
```

2. For Crystal Reports 2016, launch the "x86 Native Tools Command Prompt for VS 2017/2019" (the 32-bit prompt is required for loading a UFL to Crystal Reports 2016) as Administrator from the Windows Start Menu.

For Crystal Reports 2020, launch the "x64 Native Tools Command Prompt for VS 2017/2019" (the 64-bit prompt is required for loading a UFL to Crystal Reports 2020) as Administrator from the Windows Start Menu.

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

Crystal Reports 2016

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

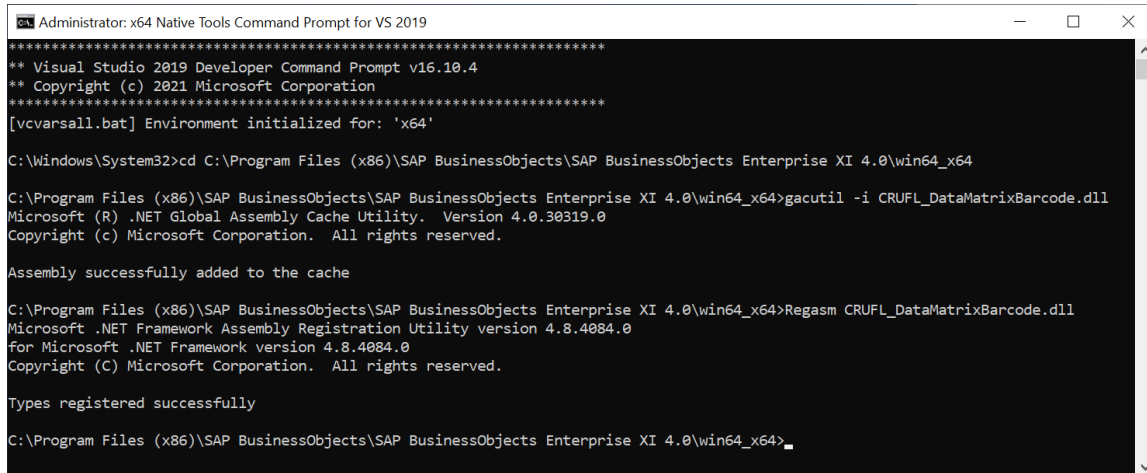
Crystal Reports 2020

```
cd C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win64_x64
```

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

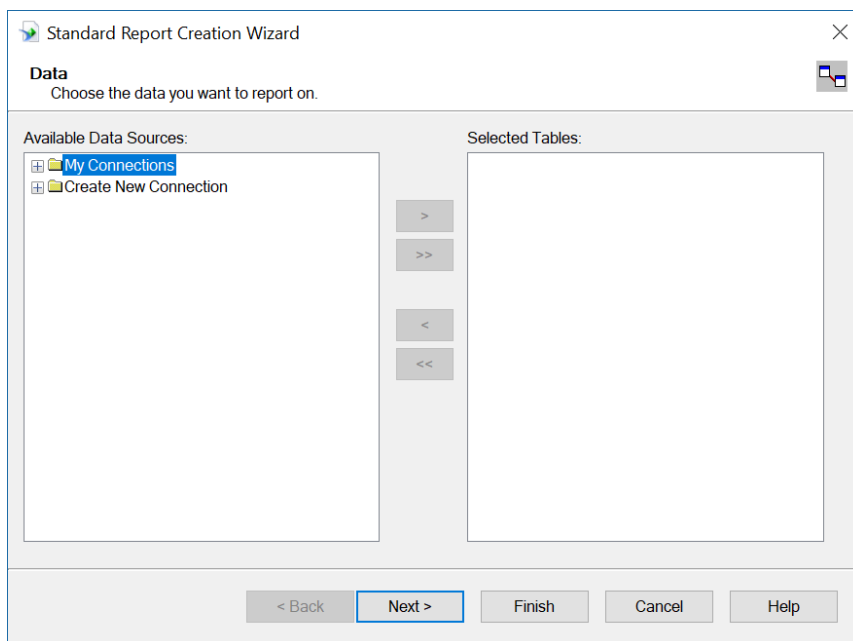
```
gacutil -i CRUFL_DataMatrixBarcode.dll  
Regasm CRUFL_DataMatrixBarcode.dll
```

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

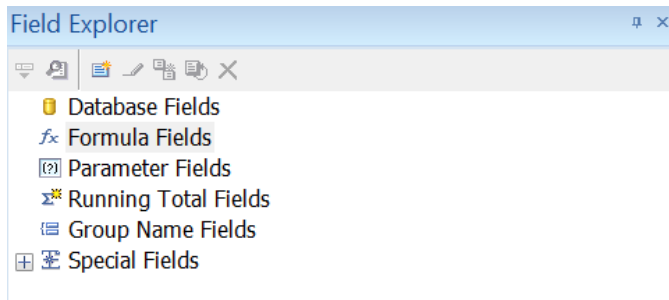


```
Administrator: x64 Native Tools Command Prompt for VS 2019  
*****  
** Visual Studio 2019 Developer Command Prompt v16.10.4  
** Copyright (c) 2021 Microsoft Corporation  
*****  
[vcvarsall.bat] Environment initialized for: 'x64'  
C:\Windows\System32>cd C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win64_x64  
C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win64_x64>gacutil -i CRUFL_DataMatrixBarcode.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\win64_x64>Regasm CRUFL_DataMatrixBarcode.dll  
Microsoft .NET Framework Assembly Registration Utility version 4.8.4084.0  
for Microsoft .NET Framework version 4.8.4084.0  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Types registered successfully  
C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win64_x64>_
```

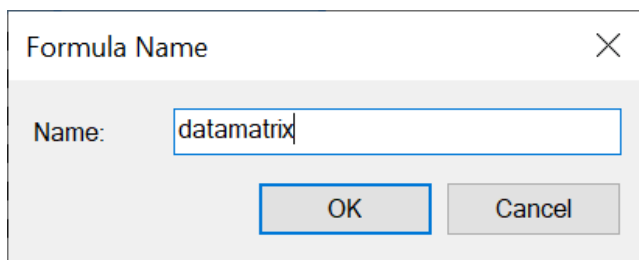
5. Launch Crystal Report and create a new "Standard Report". Click on the Finish button when prompted to select the data that you want to report on.



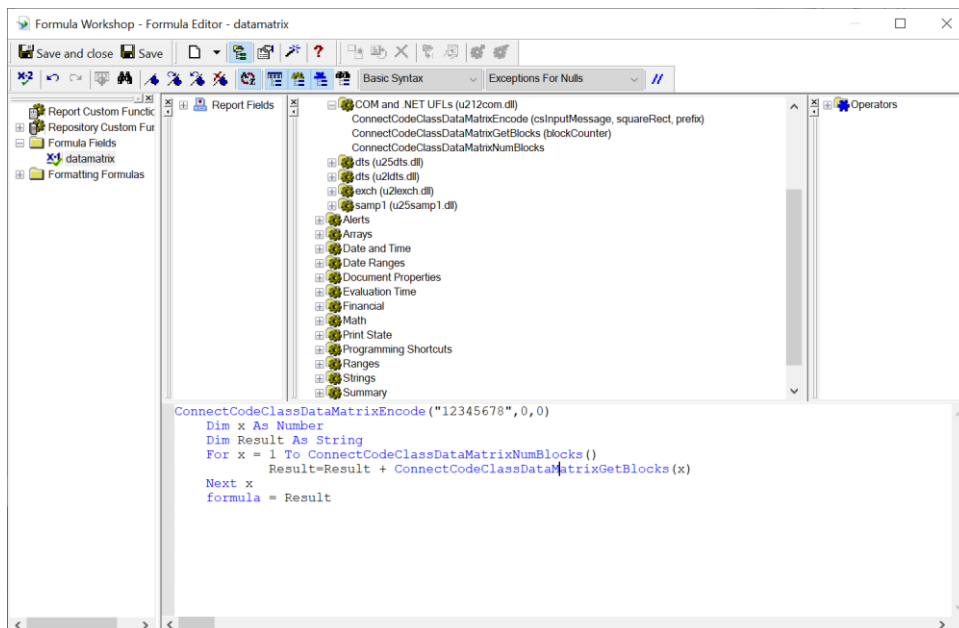
6. When the report is created, right click on "Formula Fields" in the "Field Explorer" and select "New" to create a new formula.



7. Name the formula as "datamatrix".



8. In the "Formula Workshop", expand "Functions->Additional Functions->COM and .NET UFLs (u212com.dll)". Check that you see the "ConnectCodeClassDataMatrixEncode" 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:

```
ConnectCodeClassDataMatrixEncode("12345678",0,0)
Dim x As Number
Dim Result As String
For x = 1 To ConnectCodeClassDataMatrixNumBlocks()
    Result=Result + ConnectCodeClassDataMatrixGetBlocks(x)
Next x
formula = Result
```

The formula uses "12345678" as the input data and a second parameter "0" to specify a Square DataMatrix barcode. To generate a Rectangle DataMatrix barcode, set the second parameter as "1". The final parameter "0" specifies a DataMatrix barcode without a Prefix.

First Parameter: Input Data

- ASCII - ANSI X3.4 and ISO8859-1

Second Parameter: Shape

- 0 - Square
- 1 - Rectangle

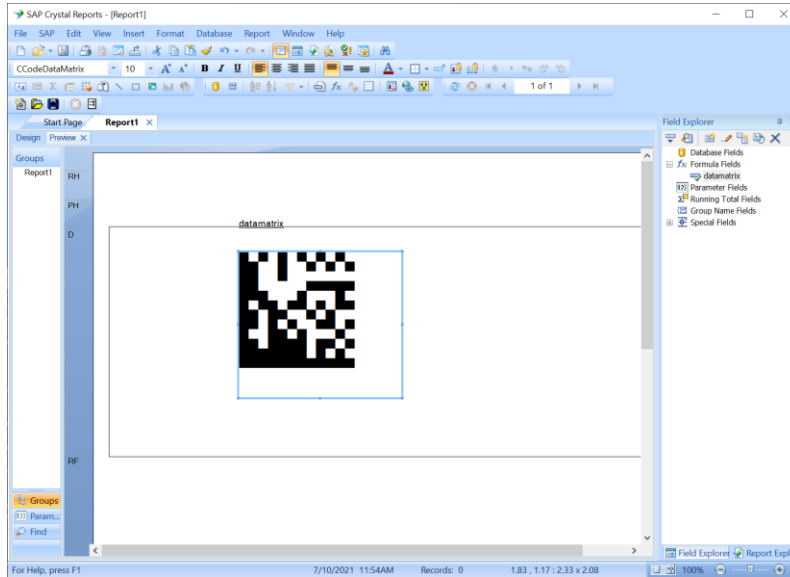
Third Parameter: Prefix

- 0 - None
- 1 - GS1 FNC1 character
- 2 - 05 Macro
- 3 - 06 Macro
- 4 - Reader Programming

After the DataMatrix 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 "datamatrix" formula onto the report. On the Design tab, select the object created and change the font to "CCodeDataMatrix" (or "CCodeDataMatrix_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 DataMatrix barcode.



4. Component Object Model Library

This tutorial illustrates the use of a COM (Component Object Model) object library with a True Type Font (DataMatrix Barcode Font), provided in ConnectCode DataMatrix package, to create an ISO/IEC 16022:2006 standard-compliant DataMatrix in a .NET Windows Form application.

Prerequisites

- ConnectCode [DataMatrix](#) package is installed
- DataMatrixCOMLibrary.dll in the Resource\DataMatrixCOMLibrary subdirectory of ConnectCode DataMatrix package. The DataMatrix 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 2015/2017/2019
- Administrator Rights

4.1 Tutorial on creating DataMatrix 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 DataMatrix COM Library folder.

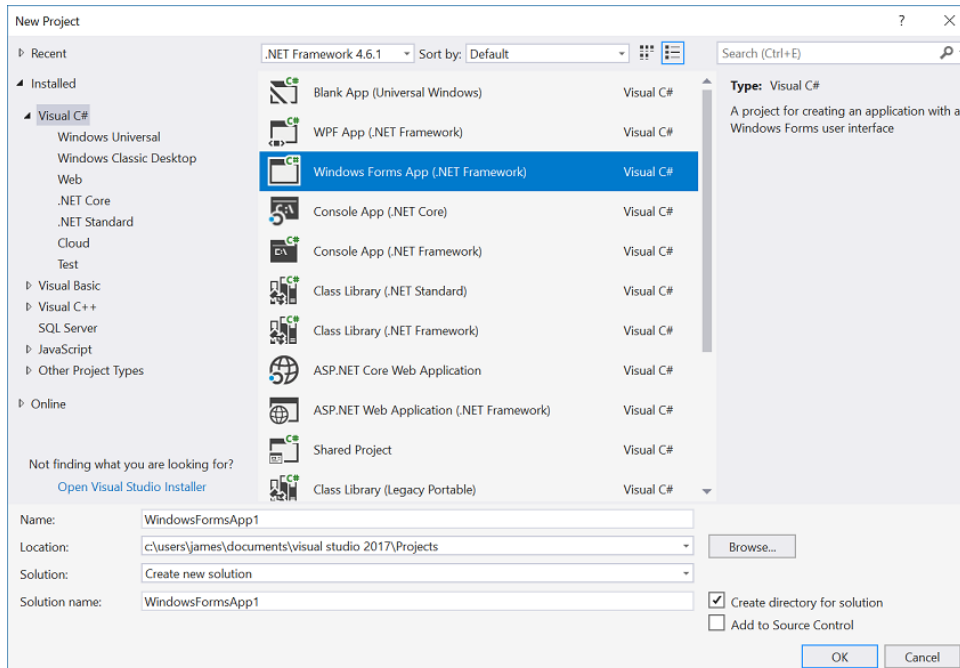
```
cd C:\Program Files (x86)\ConnectCodeDataMatrix\Resource\DataMatrixCOMLibrary  
(or ConnectCodeDataMatrixTrial if you are using the Trial package)
```

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

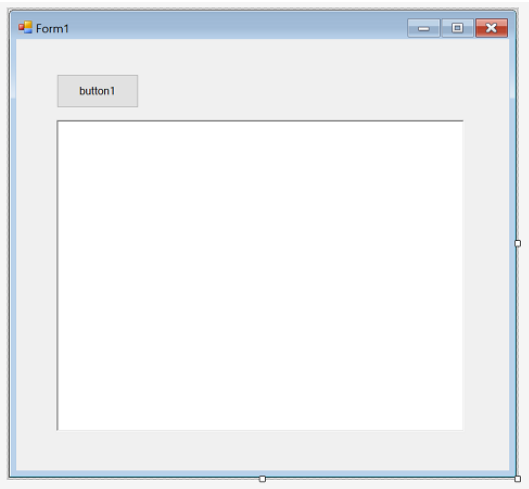
```
Regasm DataMatrixCOMLibrary.dll /tlb:DataMatrixCOMLibrary.tlb /codebase
```

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.

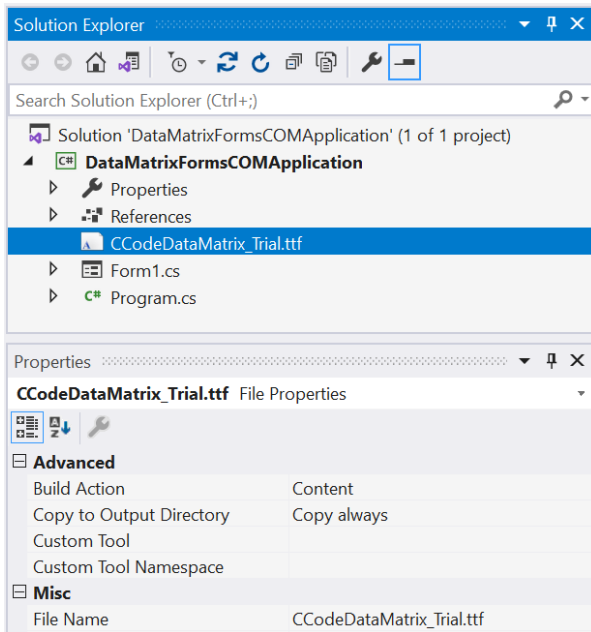
4. Launch Visual Studio. Create a new Windows Form project by clicking on "File->New Project", select a "Windows Forms App" and click on "Create" button.



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 like the screenshot below.



6. Right click on the "WindowsFormApp1" project in the "Solution Explorer" and select "Add->Existing Item". Navigate to the "C:\Program Files (x86)\ConnectCodeDataMatrix\" folder and select the "CCodeDataMatrix.ttf" font. If you are using the trial version, select the "CCodeDataMatrix_Trial.ttf" font instead.



7. In "Solution Explorer", select the "CCodeDataMatrix.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 DataMatrix 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:

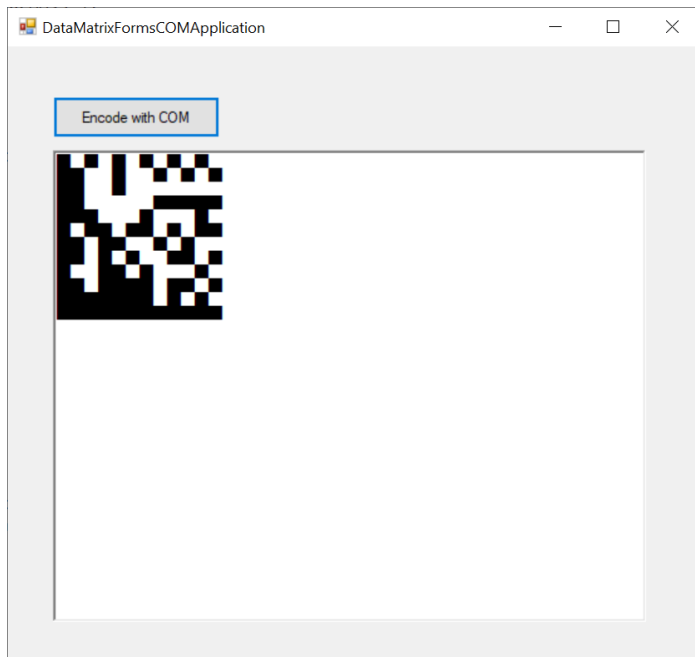
```
using System.Drawing.Text;
.
.
.
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        PrivateFontCollection pfc = new PrivateFontCollection();
        pfc.AddFontFile("CCodeDataMatrix.ttf");
        //pfc.AddFontFile("CCodeDataMatrix_Trial.ttf");
        richTextBox1.Font = new Font(pfc.Families[0], 8, FontStyle.Regular);

        Type comObjectType =
            Type.GetTypeFromProgID("Net.ConnectCode.DataMatrixCOMLibrary");
        dynamic theComObject = Activator.CreateInstance(comObjectType, false);

        //or
        //Guid myGuid = new Guid("B21B180F-8A9C-49D3-9583-0FBFDB62AA93 ");
        //Type comObjectType = Type.GetTypeFromCLSID(myGuid);
        //dynamic theComObject = Activator.CreateInstance(comObjectType, false);

        string inputData = "12345678";
        int squareRect = 0; //0 - square, 1 - rectangle
        int prefix = 0; //0-None,1-FNC1,2-05 Macro,3-06 Macro,4-Reader Programming
        string result = theComObject.Encode_DataMatrix(inputData,squareRect,prefix);
        richTextBox1.Text = result;
        System.Diagnostics.Debug.WriteLine(result);
    }
    catch (Exception ex)
    {
        System.Diagnostics.Debug.WriteLine(ex);
    }
}
}
```

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



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

5. PowerBuilder

This tutorial illustrates the use of a COM (Component Object Model) library and a barcode font available in ConnectCode DataMatrix package for creating DataMatrix barcode in PowerBuilder. The generated DataMatrix complies with the ISO/IEC 16022:2006 standards and can meet the strictest requirements of the Auto-ID industry.

Prerequisites

- PowerBuilder v12 (or APPEON PowerBuilder 2017-2021. In 2016, SAP and Appeon has entered into an agreement whereby Appeon would be responsible for developing and marketing PowerBuilder.)
- ConnectCode DataMatrix package is installed

5.1 Tutorial on creating a DataMatrix in PowerBuilder

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

```
cd C:\Program Files(x86)\ConnectCodeDataMatrix\Resource\DataMatrixCOMLibrary
```

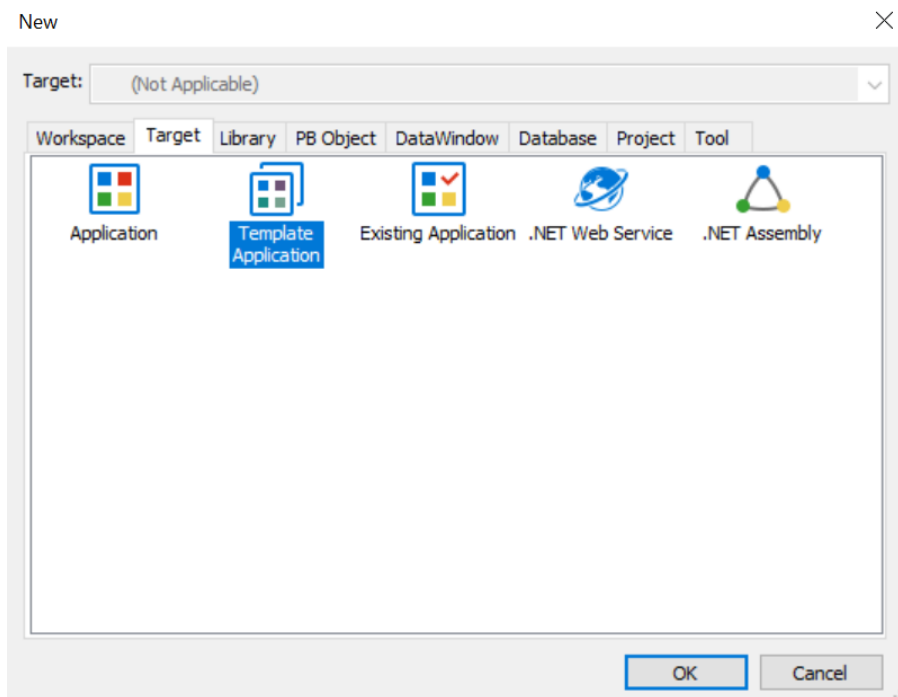
2. Next, use the Assembly Registration Tool (Regasm.exe) to register the DataMatrixCOMLibrary.dll assembly with COM.

```
Regasm DataMatrixCOMLibrary.dll /tlb:DataMatrixCOMLibrary.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.

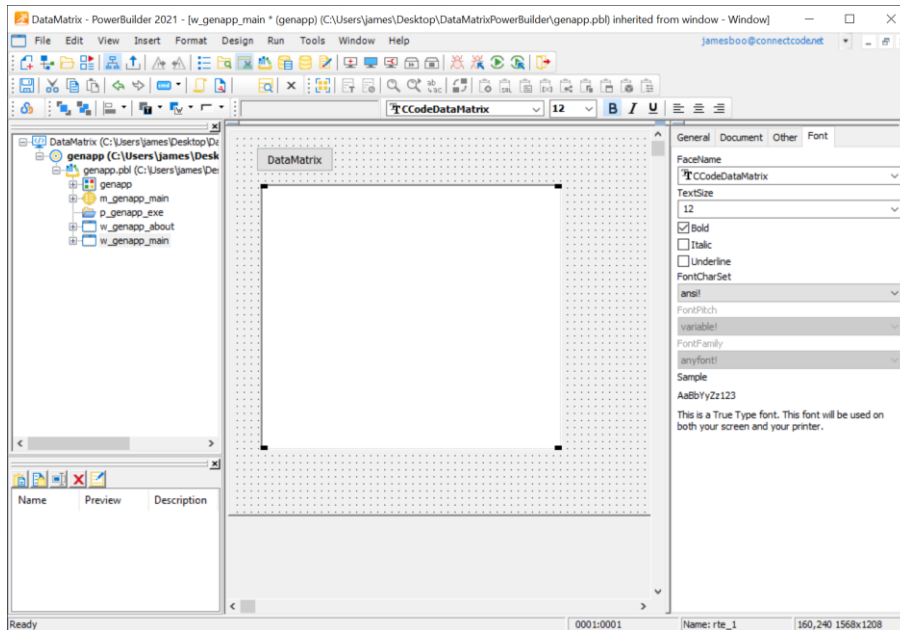
```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\
```

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 "DataMatrix". Next, insert a TextEdit control and layout the components as shown in the screenshot below.



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

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

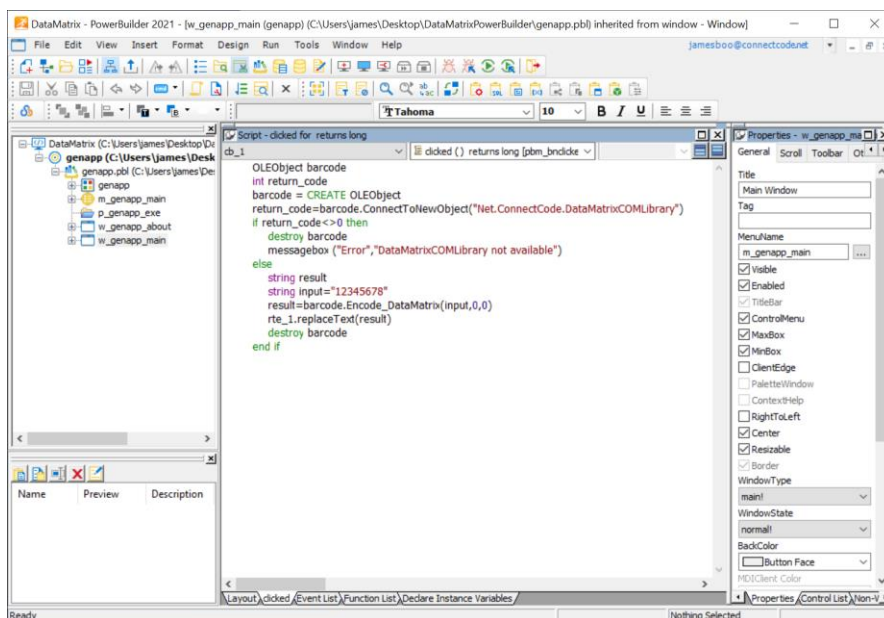
```
OLEObject barcode
int return_code
barcode = CREATE OLEObject
return_code=barcode.ConnectToNewObject("Net.ConnectCode.DataMatrixCOMLibrary")
if return_code<>0 then
    destroy barcode
    messagebox ("Error","DataMatrixCOMLibrary not available")
else
    string input="12345678"
    result=barcode.Encode_DataMatrix(input,0,0)
    //2nd parameter: 0 - square, 1 - rectangle
    //3rd parameter: 0-None,1-FNC1,2-05 Macro,3-06 Macro,4-Reader Programming
    rte_1.replaceText(result)
    destroy barcode
end if
```

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

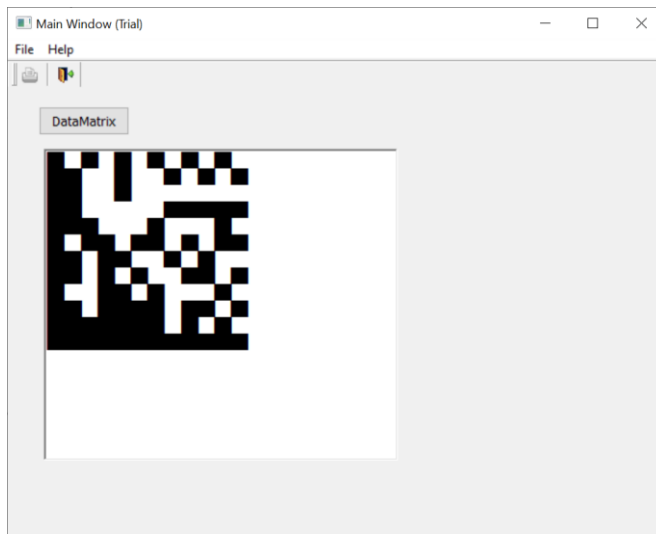
Second Parameter: 0 - Square DataMatrix, 1 - Rectangle DataMatrix

Third Parameter: Prefix

- 0 - None
- 1 - FNC1 (GS1 FNC1 character)
- 2 - 05 Macro (Macro codeword 236)
- 3 - 06 Macro (Macro codeword 237)
- 4 - Reader Programming (Indicates the symbol encodes a message used to program the reader)



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



6. .NET Standard

.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
- .NET Mono
- Xamarin.iOS
- Xamarin.Android
- Universal Windows Platform
- Windows App SDK
- .NET MAUI

.NET Standard DataMatrix Library

- [Resource/NETStandard/netstandard2.0/NETStandardDataMatrix.dll](#)

The DataMatrix Barcode package includes a .NET Standard compliant DataMatrix class library 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 can meet the strictest requirements of the auto-id industry.

7. .NET SDK

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

Library Name

DataMatrix.dll

Namespace

ConnectCode.BarcodeFonts2D

Class Name

DataMatrix

Requirements

.NET 2.0 and onwards

Constructors and Functions

DataMatrix(String data, int shape, int prefix);

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

data: The data input string to be encoded as a barcode.

shape: 0 for Square and 1 for Rectangle

prefix: DataMatrix prefix. 0 for None, 1 for GS1 FNC1, 2 for 05 Macro, 3 for 06 Macro and 4 for Reader Programming.

String Encode();

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

Sample Usage (C#)

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

Sample Visual Studio Project

1. Name - ConnectCode Encoder
2. Solution Name - ConnectCode.sln
3. Language - C#
4. Requirements - .NET 2.0 and onwards

7.1 .NET Framework 4.0 Notes

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