

# Table of Contents

Foreword	0
<b>Part I Introduction</b>	<b>2</b>
<b>Part II Installation</b>	<b>2</b>
1 Trial Version.....	2
2 Full Version.....	3
<b>Part III How to Distribute It</b>	<b>4</b>
<b>Part IV Reference Guide</b>	<b>4</b>
1 Properties.....	4
BackColor Property .....	4
BarColor Property .....	4
Data Property .....	4
HandleTilde Property .....	5
Mode Property .....	5
ModuleSize Property .....	5
Orientation Property .....	6
PreferredFormat Property .....	6
2 Methods.....	7
GetActualRC Method .....	7
GetActualSize Method .....	7
GetPatternData Method .....	8
Render Method .....	9
SaveAsImage Method .....	10
SaveAsMemory Method .....	10
SetSize Method .....	11
SetStructuredAppend Method .....	11
3 Enumerations.....	12
Mode Enumeration .....	12
Orientation Enumeration .....	12
PreferredFormat Enumeration .....	13
<b>Part V How to Use It in Reporting Services</b>	<b>13</b>
1 Create a Report Project.....	13
2 Add DataMatrix Barcodes.....	17
<b>Part VI License</b>	<b>23</b>
<b>Index</b>	<b>0</b>

# 1 Introduction

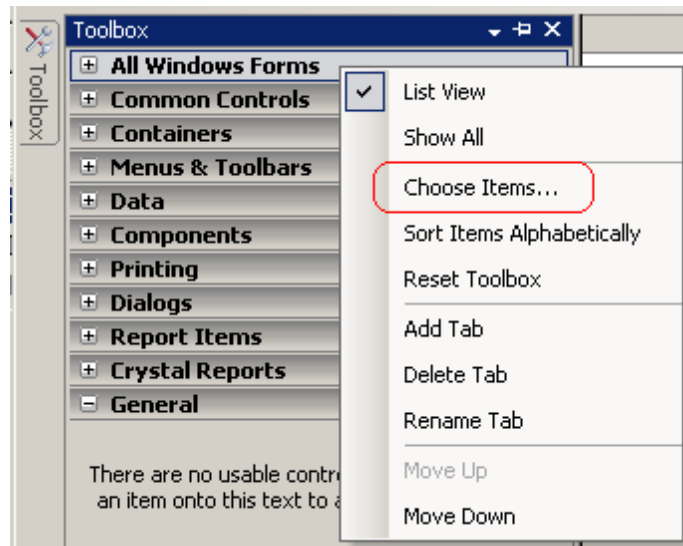
MW6 DataMatrix .NET control is a flexible and reliable .NET component and can create professional 2D DataMatrix images for your .NET application, you can save the DataMatrix as different image format files. It is easy to print the DataMatrix barcode using the PrintDocument Control provided by the .NET Framework.

DataMatrix is designed to pack a lot of information in a very small space, our DataMatrix .NET control supports the ECC-200 version, it is capable of encoding 1556 bytes, 2335 alphanumeric characters, or 3116 numeric digits.

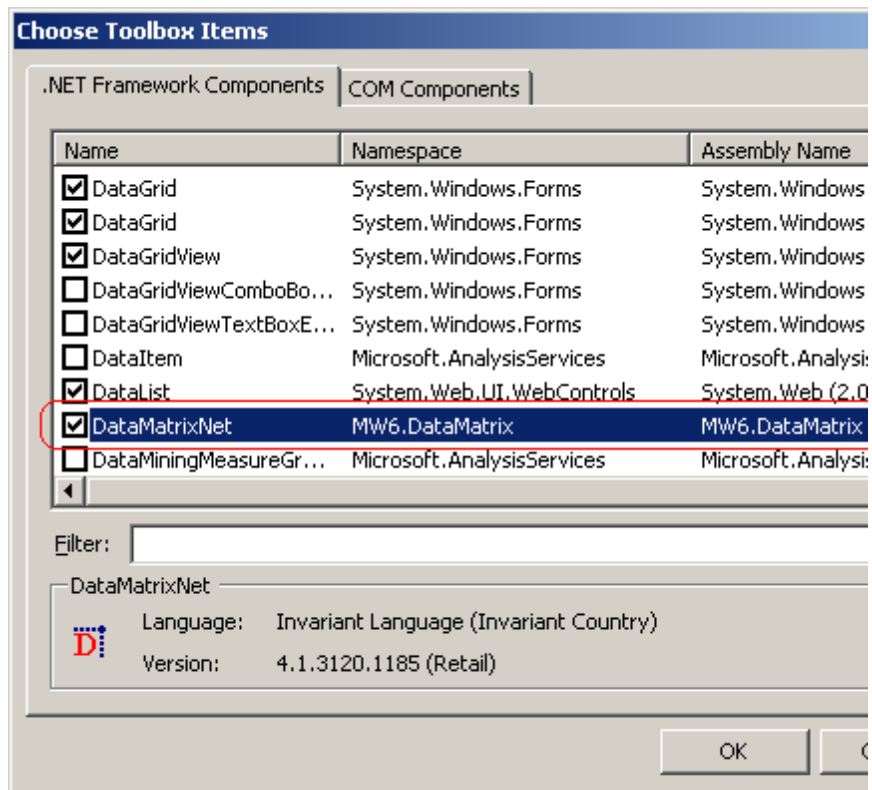
## 2 Installation

### 2.1 Trial Version

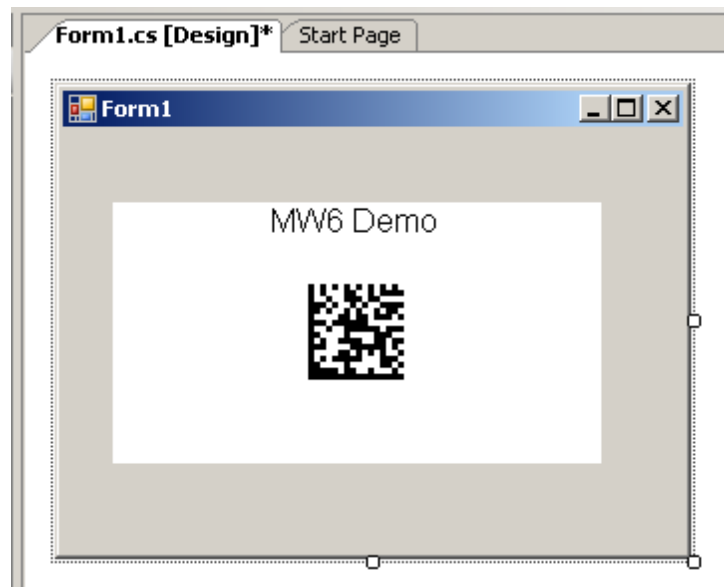
1. The trial version DataMatrix .NET control appends the "MW6 Demo" to the string encoded with the DataMatrix format.
2. After the installation for the trial version is finished, copy MW6.DataMatrix.dll in VB.NET or C# demo project sub folder of the destination folder (e.g. "C:\Program Files\MW6 .NET Components \DataMatrix") to your project folder.
3. Right click anywhere on the Toolbox to select "**Choose Items...**".



4. In the "**Choose Toolbox Items**" dialog, select "**.NET Framework Components**", click on the "**Browse**" and select MW6.DataMatrix.dll.
-



5. To use the control in your Windows form, just drag it from the Toolbox and drop it onto your form.



## 2.2 Full Version

1. Uninstall the trial version DataMatrix .NET control if applicable.
2. Complete the installation for the full version DataMatrix .NET control, copy MW6.DataMatrix.dll in VB.NET or C# demo project sub folder of the destination folder (e.g. "C:\Program Files\MW6 .NET Components\DataMatrix") to your project folder to replace the trial version DataMatrix .NET control .dll

file.

## 3 How to Distribute It

If you want to redistribute the DataMatrix .NET control as part of your application, simply put MW6.DataMatrix.dll into application local folder on the target machine.

## 4 Reference Guide

### 4.1 Properties

#### 4.1.1 BackColor Property

Gets or sets the background color of the DataMatrix barcode.

```
[Visual Basic .NET]
```

```
Public Property BackColor As Color
```

```
[C#]
```

```
public Color BackColor {get; set;}
```

#### Remarks

The default value is white color.

#### 4.1.2 BarColor Property

Gets or sets the color of the DataMatrix barcode.

```
[Visual Basic .NET]
```

```
Public Property BarColor As Color
```

```
[C#]
```

```
public Color BarColor {get; set;}
```

#### Remarks

The default value is black color.

#### 4.1.3 Data Property

Gets or sets the message to encode with DataMatrix .NET control.

```
[Visual Basic .NET]
```

```
Public Property Data As String
```

---

```
[C#]  
public string Data {get; set;}
```

#### Remarks

The default value is "12".

### 4.1.4 HandleTilde Property

Gets or sets a boolean flag indicating whether to process the tilde character "~" or not.

```
[Visual Basic .NET]  
Public Property HandleTilde As Boolean
```

```
[C#]  
public bool HandleTilde {get; set;}
```

#### Remarks

If this property is set to TRUE, non-printable characters can be passed to DataMatrix .NET control by using the tilde character, "~dNNN" represents the ASCII character encoded by the 3 digits NNN, for example, "~d010" represents the character LF (line feed).

"~1" is used to indicate FNC1. For example, "~10107612345678900~117100503" can be used to generate GS1 DataMatrix "(01)0107612345678900(17)100503", and "~110AC34563G3" can be used to generate GS1 DataMatrix "(10)AC34563G3".

"~5" is used to indicate Macro 5. For example, "~5ABCDEF[GS]123456" can be used to generate DataMatrix "[>][RS]05[GS]ABCDEF[GS]123456[RS][EOT]".

"~6" is used to indicate Macro 6. For example, "~6ABCDEF[GS]123456" can be used to generate DataMatrix "[>][RS]06[GS]ABCDEF[GS]123456[RS][EOT]".

[RS] is the record separator with ASCII value 30, [GS] is the group separator with ASCII value 29, and [EOT] is the end of transmission with ASCII value 4.

### 4.1.5 Mode Property

Gets or sets the encoding mode of the DataMatrix barcode.

```
[Visual Basic .NET]  
Public Property Mode As enumMode
```

```
[C#]  
public enumMode Mode {get; set;}
```

### 4.1.6 ModuleSize Property

Gets or sets the size (width/height) of the square-shaped module.

```
[Visual Basic .NET]
```

Public Property ModuleSize As float

[C#]

```
public float ModuleSize {get; set;}
```

### Remarks

The default value is 0.07, internally our DataMatrix .NET control converts the module size from centimeters to pixels based on the device resolution, round up or round down float pixel value to the nearest integer.

The centimeter to pixel conversion formula is :

$$size\_in\_pixels = size\_in\_centimeters * device\_resolution / 2.54$$

For example, if you render barcode on computer screen and the screen resolution is 96dpi.

(1) Set ModuleSize property to 0.04,  $size\_in\_pixels = 0.04 * 96 / 2.54 = 1.5118$ , round up 1.5118 to 2, so actual module size is 2 pixels.

(2) Set ModuleSize property to 0.06,  $size\_in\_pixels = 0.06 * 96 / 2.54 = 2.2677$ , round down 2.2677 to 2, so actual module size is 2 pixels.

(3) Set ModuleSize property to 0.07,  $size\_in\_pixels = 0.07 * 96 / 2.54 = 2.6456$ , round up 2.6456 to 3, so actual module size is 3 pixels.

Different ModuleSize property values might end up with same module size in pixels due to performing rounding operations.

## 4.1.7 Orientation Property

Gets or sets the orientation of the DataMatrix barcode.

[Visual Basic .NET]

Public Property Orientation As enumOrientation

[C#]

```
public enumOrientation Orientation {get; set;}
```

## 4.1.8 PreferredFormat Property

Gets or sets the format of the DataMatrix barcode.

[Visual Basic .NET]

Public Property PreferredFormat As enumPreferredFormat

[C#]

```
public enumPreferredFormat PreferredFormat {get; set;}
```

---

**Remarks**

If you set PreferredFormat to pfAuto (Auto format), our DataMatrix .NET control will automatically choose an appropriate format with enough data capacity to encode the string.

If you set PreferredFormat to other values and the data capacity of the selected format is not big enough to encode the string, our DataMatrix .NET control will also automatically choose an appropriate format with bigger data capacity to encode the string.

**See Also**

GetActualRC() Method

## 4.2 Methods

### 4.2.1 GetActualRC Method

Gets the actual numbers of rows and columns for the DataMatrix barcode.

[Visual Basic .NET]

```
Public Sub GetActualRC(ByRef ActualRows As Integer, ByRef ActualCols As Integer)
```

[C#]

```
public void GetActualRC(ref int ActualRows, ref int ActualCols);
```

**Parameters**

*ActualRows*

A pointer to the variable that receives the final number of rows for the DataMatrix barcode.

*ActualCols*

A pointer to the variable that receives the final number of columns for the DataMatrix barcode.

**Remarks**

If you set PreferredFormat to pfAuto (Auto format), DataMatrix .NET control will automatically choose an appropriate format with enough data capacity to encode the string, use this method to retrieve the information about the final numbers of rows and columns.

If you set PreferredFormat to other values and the data capacity of the selected format is not big enough to encode the string, DataMatrix .NET control will also automatically choose an appropriate format with bigger data capacity to encode the string, so the final numbers of rows and columns might not be equal to the numbers of rows and columns specified by the PreferredFormat property.

### 4.2.2 GetActualSize Method

Gets the actual size of the DataMatrix barcode which is rendered onto either computer screen or other devices such as printers.

```
[Visual Basic .NET]
Public Sub GetActualSize(ByVal ScreensTarget As Boolean, _
                        ByVal TargetG As Graphics, _
                        ByRef ActualWidth As Integer, _
                        ByRef ActualHeight As Integer)
```

```
[C#]
public void GetActualSize(bool ScreensTarget,
                        Graphics TargetG,
                        ref int ActualWidth,
                        ref int ActualHeight);
```

### Parameters

#### *ScreensTarget*

Indicates whether the DataMatrix barcode is rendered onto computer screen or not.

#### *TargetG*

Graphics object to be used for rendering, if the parameter *ScreensTarget* is set to TRUE, set this parameter to NULL.

#### *ActualWidth*

A pointer to the variable that receives the width of the DataMatrix barcode (in pixels).

#### *ActualHeight*

A pointer to the variable that receives the height of the DataMatrix barcode (in pixels).

## 4.2.3 GetPatternData Method

Gets the DataMatrix barcode pattern matrix data.

```
[Visual Basic .NET]
Public Function GetPatternData(ByRef Buffer() As Char, _
                              ByRef Size As Long, _
                              ByRef Rows As Integer, _
                              ByRef Columns As Integer) As Boolean
```

```
[C#]
public bool GetPatternData(ref char[] Buffer,
                          ref long Size,
                          ref int Rows,
                          ref int Columns);
```

### Parameters

#### *Buffer*

Pointer to a buffer that receives the character stream ('1's and '0's) storing the DataMatrix barcode pattern matrix data row by row from the top left matrix corner, '1' indicates the black module and '0' indicates the white module.

---



If the function fails and the variable pointed to by *Size* returns the required buffer size, in characters.

#### *Size*

[in/out] On input, specifies the size, in characters, of the *Buffer*. On output, receives the size, in characters, of the DataMatrix barcode pattern matrix ('1's and '0's).

#### *Rows*

A pointer to the variable that receives the number of the rows for the pattern matrix.

#### *Columns*

A pointer to the variable that receives the number of the columns for the pattern matrix..

### **Return Value**

If the function succeeds, the return value is a nonzero value, otherwise the return value is zero.

### **Remarks**

You can use this method to obtain the DataMatrix barcode pattern matrix data and render the DataMatrix barcode onto any device such as the printer, only *Data*, *HandleTilde*, *Mode*, and *PreferredFormat* properties affect the pattern matrix data output.

Based on the *Orientation* property value, rotate the pattern matrix accordingly before you render the DataMatrix barcode onto a device.

## **4.2.4 Render Method**

Renders the DataMatrix barcode onto the device such as a computer monitor or a printer.

[Visual Basic .NET]

```
Public Sub Render(ByVal renderG As Graphics, ByVal p As Point)
```

[C#]

```
public void Render(Graphics renderG, Point p);
```

### **Parameters**

#### *renderG*

Graphics object to be used for rendering.

#### *p*

Stores the coordinates (in pixels) of the top-left corner of the DataMatrix barcode.

### 4.2.5 SaveAsImage Method

Exports the DataMatrix barcode image to a file.

[Visual Basic .NET]

```
Public Sub SaveAsImage(ByVal FileName As String, ByVal ImgFormat As ImageFormat)
```

[C#]

```
public void SaveAsImage(string FileName, ImageFormat ImgFormat);
```

#### Parameters

*FileName*

A string that contains the name of the file to which to save DataMatrix barcode image.

*ImgFormat*

Specifies the image format.

#### Remarks

Before you call this method, use `GetActualSize()` method to obtain the actual size of DataMatrix barcode and use `SetSize()` method to set image size by adding surrounding white space around the DataMatrix barcode.

#### See Also

[GetActualSize\(\) Method](#) | [SetSize\(\) Method](#)

### 4.2.6 SaveAsMemory Method

Exports the DataMatrix barcode image byte stream to the memory.

[Visual Basic .NET]

```
Public Sub SaveAsMemory(ByVal MS As MemoryStream, ByVal ImgFormat As ImageFormat)
```

[C#]

```
public void SaveAsMemory(MemoryStream MS, ImageFormat ImgFormat);
```

#### Parameters

*MS*

Specifies the memory stream that holds the byte stream of the DataMatrix barcode image.

*ImgFormat*

Specifies the image format.

#### Remarks

---

Before you call this method, use `GetActualSize()` method to obtain the actual size of the DataMatrix barcode and use `SetSize()` method to set the image size by adding the surrounding white space around the DataMatrix barcode.

#### See Also

[GetActualSize\(\) Method](#) | [SetSize\(\) Method](#)

### 4.2.7 SetSize Method

Sets the size of the image which contains the DataMatrix barcode.

[Visual Basic .NET]

```
Public Sub SetSize(ByVal Width As Integer, ByVal Height As Integer)
```

[C#]

```
public void SetSize(int Width, int Height);
```

#### Parameters

*Width*

The width, in pixels, of the image.

*Height*

The height, in pixels, of the image.

#### Remarks

First call `GetActualSize()` method to obtain the actual size of the DataMatrix barcode, then use this method to set image size by adding surrounding white space around the DataMatrix barcode.

#### See Also

[GetActualSize\(\) Method](#)

### 4.2.8 SetStructuredAppend Method

Specifies which symbol this is in a sequence and the total number of symbols in the sequence.

[Visual Basic .NET]

```
Public Sub SetStructuredAppend(ByVal AllowSA As Boolean, _  
                               ByVal SymbolID As Integer, _  
                               ByVal SymbolCount As Integer)
```

[C#]

```
public void SetStructuredAppend(bool AllowSA,  
                                int SymbolID,  
                                int SymbolCount);
```

**Parameters***AllowSA*

Indicates whether the structured append is allowed in the current DataMatrix barcode, if this is FALSE, the parameters *SymbolID* and *SymbolCount* are irrelevant.

*SymbolID*

Specifies which symbol this is in a sequence, the parameter must be between 1 and *SymbolCount*.

*SymbolCount*

Specifies the total number of symbols in the sequence, the maximum value is 16, which means that up to 16 symbols can be linked together using the structured append protocol.

**Remarks**

Don't call this method if you don't need the structured append feature.

## 4.3 Enumerations

### 4.3.1 Mode Enumeration

An enumeration type for all possible encoding mode values.

**Members**

Name	Comment
mdAscii	ASCII mode for mainly encoding ASCII characters (0-127)
mdC40	C40 mode for mainly encoding numeric and upper case characters
mdText	Text mode for mainly encoding numeric and lower case characters
mdBase256	Base256 mode for mainly encoding bytes of data

### 4.3.2 Orientation Enumeration

An enumeration type for all possible orientation values.

**Members**

Name	Comment
or0	0 Degree
or90	90 Degrees
or180	180 Degrees
or270	270 Degrees

### 4.3.3 PreferredFormat Enumeration

An enumeration type for all possible preferred format values.

#### Members

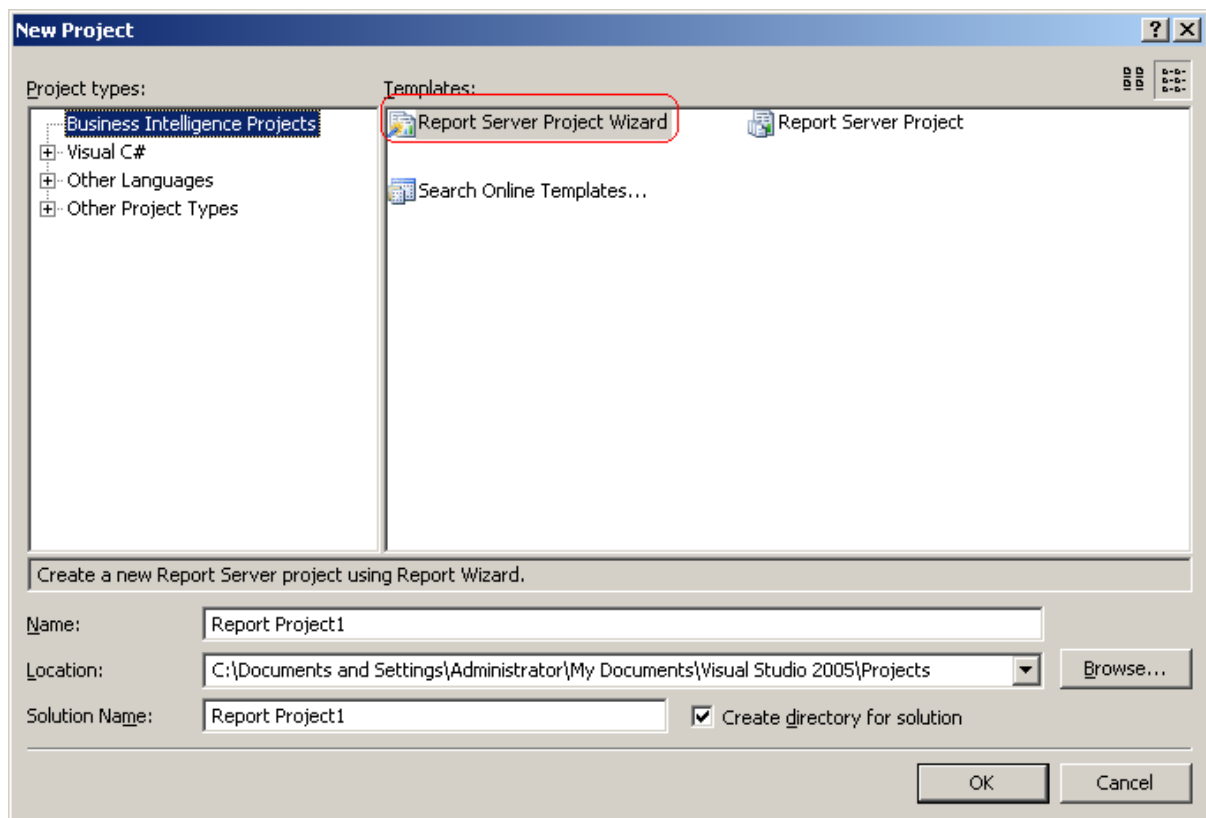
Name	Description	Data Capacity		
		Numeric	Alphanumeric	Byte
pfAuto	Auto format			
pf10X10	10 X 10 format	6	3	1
pf12X12	12 X 12 format	10	6	3
pf14X14	14 X 14 format	16	10	6
pf16X16	16 X 16 format	24	16	10
pf18X18	18 X 18 format	36	25	16
pf20X20	20 X 20 format	44	31	20
pf22X22	22 X 22 format	60	43	28
pf24X24	24 X 24 format	72	52	34
pf26X26	26 X 26 format	88	64	42
pf32X32	32 X 32 format	124	91	60
pf36X36	36 X 36 format	172	127	84
pf40X40	40 X 40 format	228	169	112
pf44X44	44 X 44 format	288	214	142
pf48X48	48 X 48 format	348	259	172
pf52X52	52 X 52 format	408	304	202
pf64X64	64 X 64 format	560	418	278
pf72X72	72 X 72 format	736	550	366
pf80X80	80 X 80 format	912	682	454
pf88X88	88 X 88 format	1152	862	574
pf96X96	96 X 96 format	1392	1042	694
pf104X104	104 X 104 format	1632	1222	814
pf120X120	120 X 120 format	2100	1573	1048
pf132X132	132 X 132 format	2608	1954	1302
pf140X140	144 X 144 format	3116	2335	1556
pf8X18	8 X 18 format	10	6	3
pf8X32	8 X 32 format	20	13	8
pf12X26	12 X 26 format	32	22	14
pf12X36	12 X 36 format	44	31	20
pf16X36	16 X 36 format	64	46	30
pf16X48	16 X 48 format	98	72	47

## 5 How to Use It in Reporting Services

### 5.1 Create a Report Project

Follow the instructions to create a report project in the Reporting Services environment:

1. Select **File | New | Project** from the Visual Studio .NET IDE menu, in the **New Project** dialog, highlight the "**Business Intelligence Projects**", then choose the "**Report Server Project Wizard**", enter the name of the report in the "**Name**" box, click the "**OK**" button.



2. In the "**Select the Data Source**" dialog, click the "**Edit**" button to open the "**Connection Properties**" dialog, enter the SQL server instance name in the "**Server name**" box, choose a database from the drop-down list, click the "**OK**" button.

**Connection Properties** [?] [X]

Data source:  
Microsoft SQL Server (SqlClient) [Change...]

Server name:  
PC-V40G4L4T6\SQLEXPRESS\_NEW [Refresh]

Log on to the server

- Use Windows Authentication
- Use SQL Server Authentication

User name: [ ]

Password: [ ]

Save my password

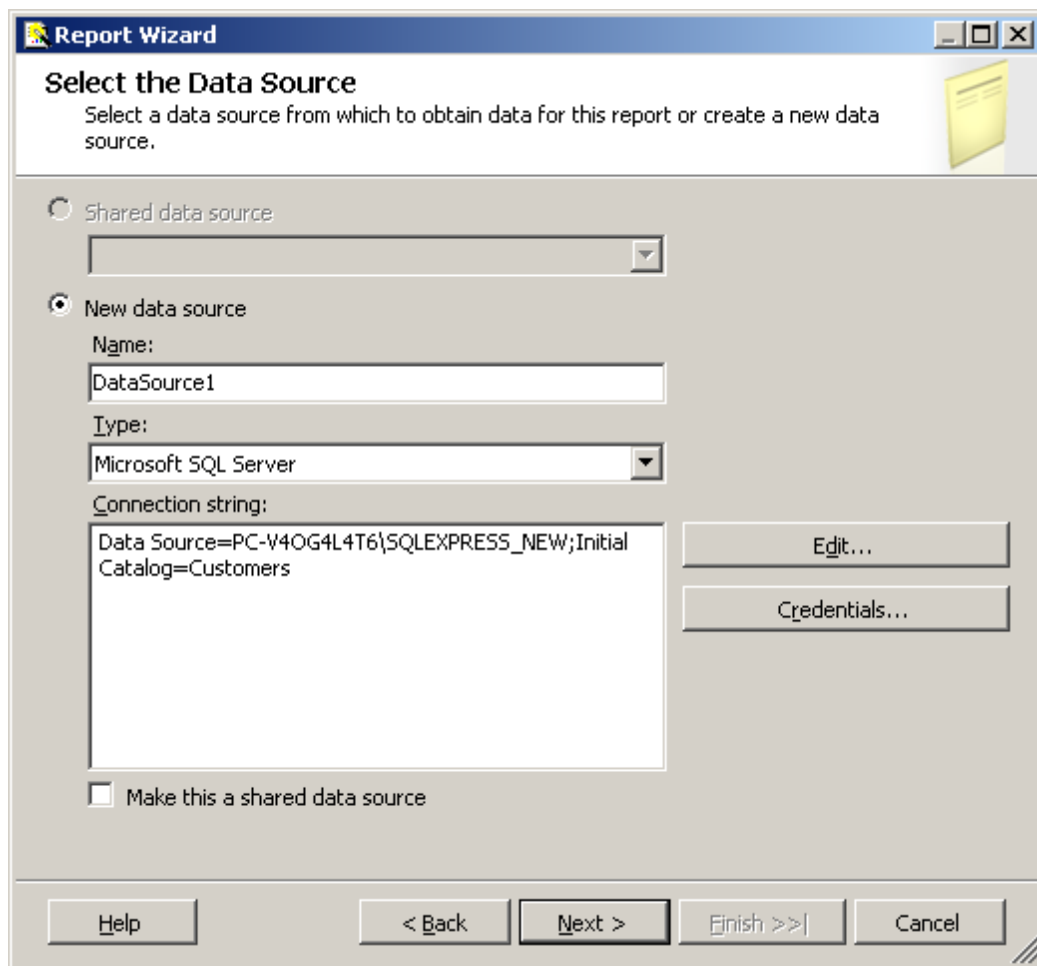
Connect to a database

- Select or enter a database name:  
Customers [ ]
- Attach a database file:  
[ ] [Browse...]

Logical name:  
[ ]

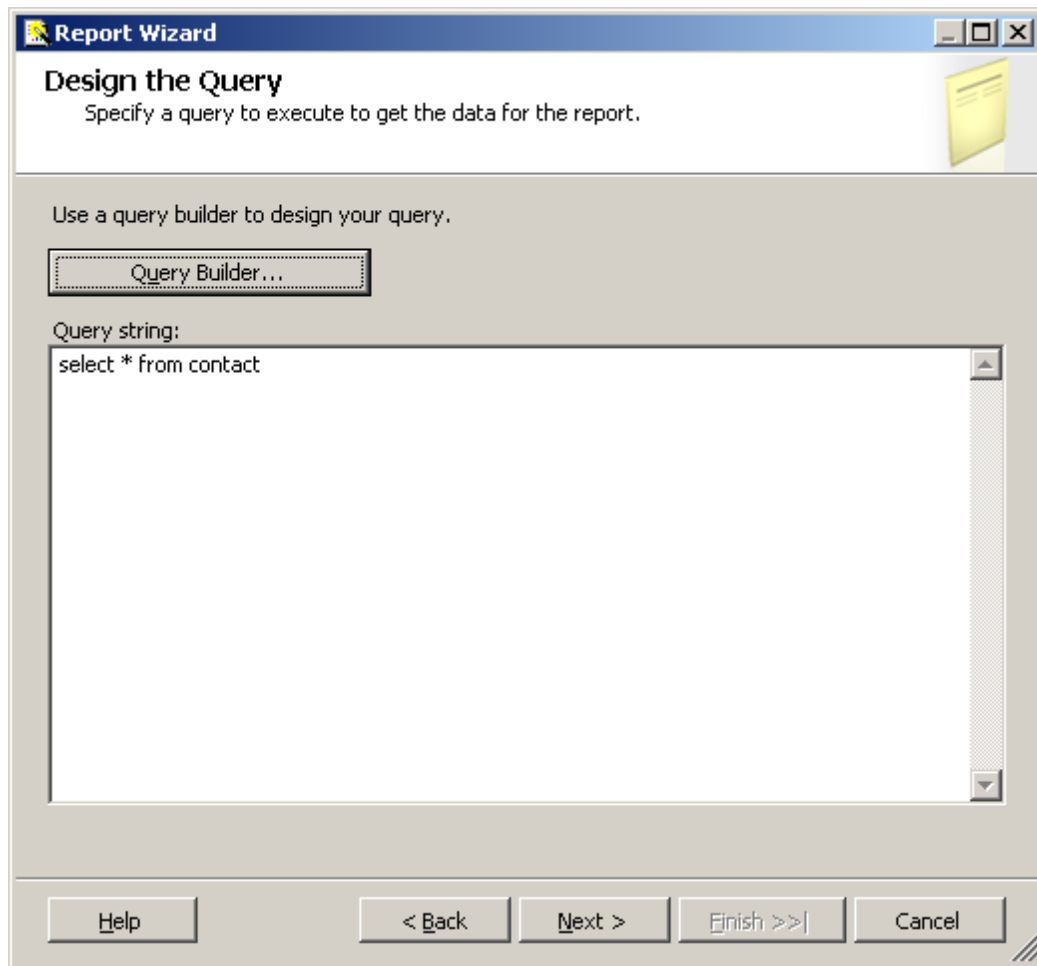
[Advanced...]

[Test Connection] [OK] [Cancel]



3. Click the "**Next**" button to open the "**Design the Query**" dialog, enter the query string to extract the data, click the "**Next**" button



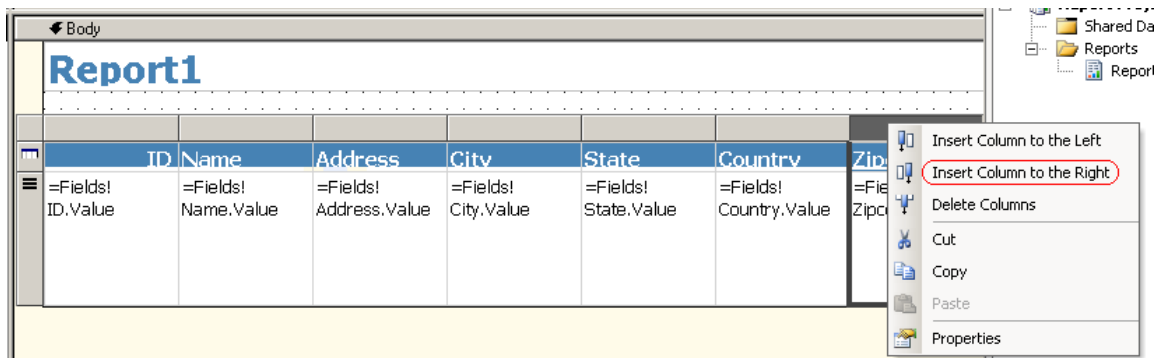


4. In the following dialogs, choose the appropriate options for the type of the report, the way of how to group the data in the table and the table style, then click the **"Finish"** button.

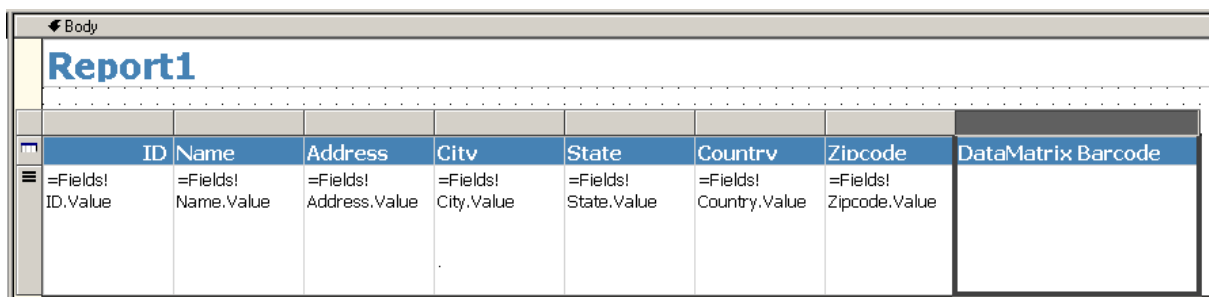
## 5.2 Add DataMatrix Barcodes


Follow the instructions to add the DataMatrix barcodes to the report.

1. In order to use the DataMatrix .NET control in the Reporting Services, please copy MW6.DataMatrix.dll to "C:\Program Files\Microsoft Visual Studio X\Common7\IDE\PrivateAssemblies" for 32 bit OS or "C:\Program Files (x86)\Microsoft Visual Studio X\Common7\IDE\PrivateAssemblies" for 64 bit OS, the X value is associated with Visual Studio .NET version, it might be 8 for .NET 2005, 9.0 for .NET 2008, 10.0 for .NET 2010, 11.0 for .NET 2012.
2. Right click the last column in the table, Select the **"Insert Column to the Right"**.

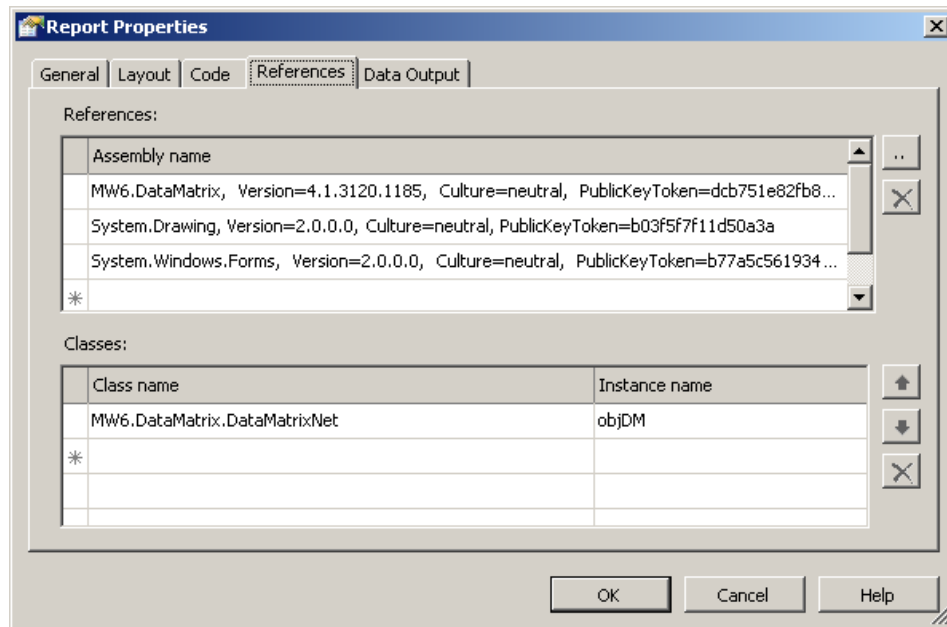


3. Change the column title to the "DataMatrix Barcode".



4. On the "Report" menu, click the "Report Properties", click the "References" tab, click the two-dot button  to open the "Add Reference" dialog.

- A. Click the "Browse" tab, navigate to the location of the assembly MW6.DataMatrix.dll, select the file and click the "Add" button.
- B. Click the ".NET" tab, highlight the assemblies "System.Drawing" and "System.Windows.Form", click the "Add" button.
- C. Enter "MW6.DataMatrix.DataMatrixNet" in the "Class name" box, enter "objDataMatrix" in the "Instance name" box to create an assembly object to use in the code to retrieve the DataMatrix barcode image byte stream.



5. On the same "**Report Properties**" dialog, click the "**Code**" tab, copy and paste the following code into this tab, this function is used to retrieve the DataMatrix barcode image byte stream, modify the code a bit to meet your application requirements.

```
Public Function GetImgStream(ByVal DataStr As String) As Byte()

    Dim ActualWidth As Integer, ActualHeight As Integer
    Dim ExtraWidth As Integer, ExtraHeight As Integer

    ' Message
    objDM.Data = DataStr

    ' Module Size
    objDM.ModuleSize = 0.07

    ' Mode
    objDM.Mode = 0

    ' Preferred Format
    objDM.PreferredFormat = 0

    ' Orientation
    objDM.Orientation = 0

    ' Handle Tilde Character?
    objDM.HandleTilde = True

    ExtraWidth = 60
    ExtraHeight = 70

    objDM.GetActualSize(True, Nothing, ActualWidth, ActualHeight)

    objDM.SetSize(ActualWidth + ExtraWidth, ActualHeight + ExtraHeight)

    Dim MS As System.IO.MemoryStream = New System.IO.MemoryStream
    Dim ImgStream As Byte()

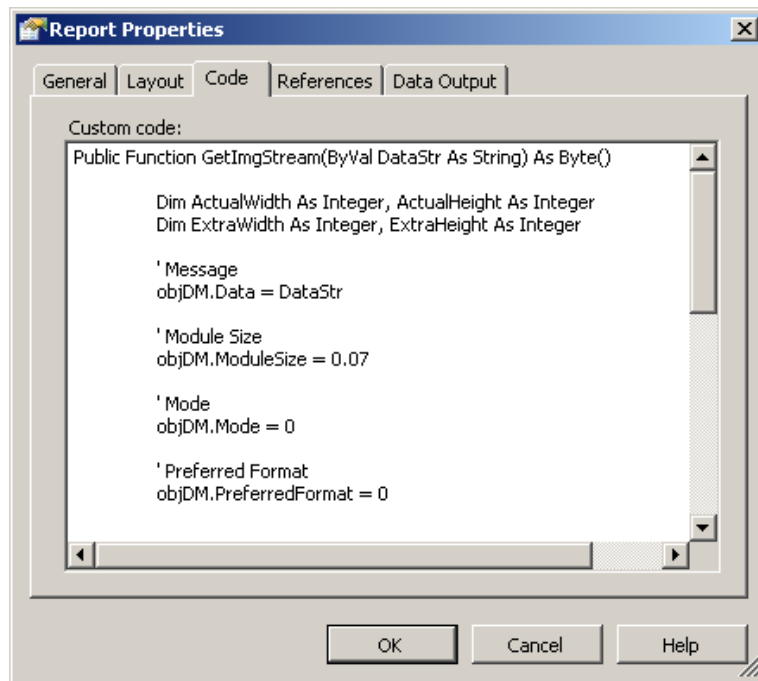
    objDM.SaveAsMemory(MS, System.Drawing.Imaging.ImageFormat.Jpeg)

    ImgStream = MS.ToArray

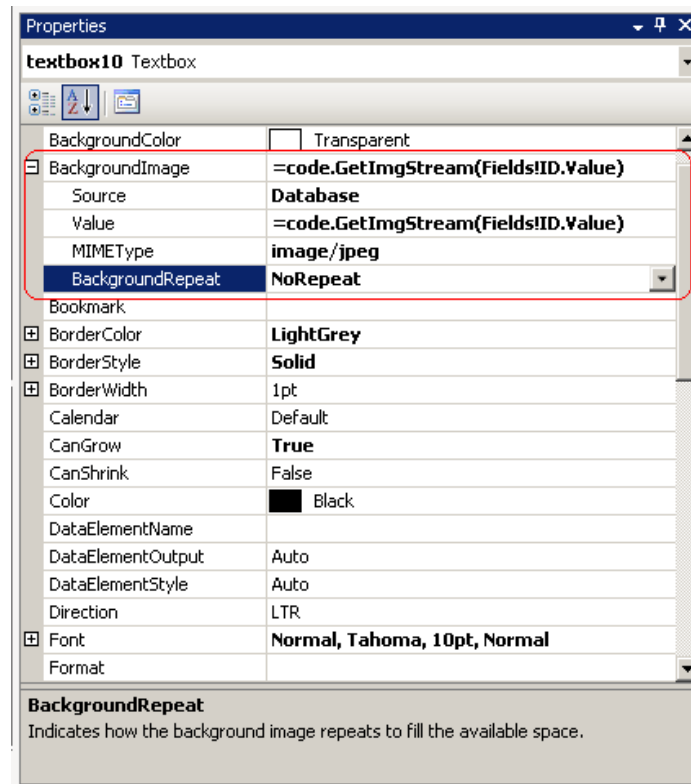
    MS.Close()

    Return ImgStream

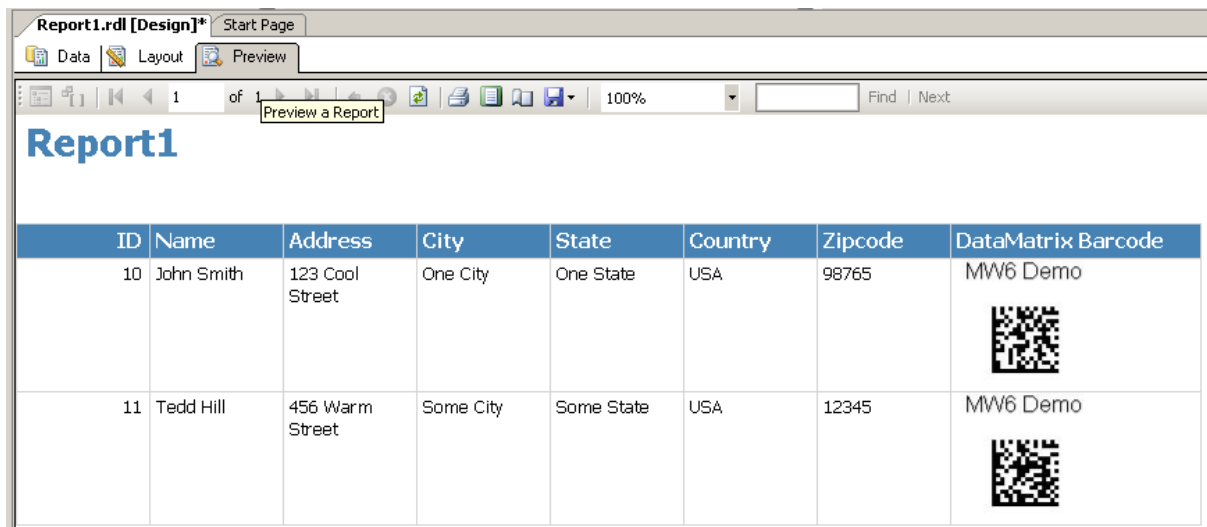
End Function
```



6. Change the "**BackgroundImage**" property of the text box in the "**DataMatrix Barcode**" column to display the DataMatrix barcode image encoding ID value.
- Source:** Select the Database from the drop-down list, since the data will be pulled from a database field.
  - Value:** Enter the string "=code.GetImgStream(Fields!ID.Value)", it will ask the report to use the GetImgStream() function to retrieve the DataMatrix barcode image byte stream for the ID field value of the database.
  - MIMEType:** Select the image/jpeg from the drop-down list, since the DataMatrix barcode image is in jpeg format.
  - BackgroundRepeat:** Select the NoRepeat from the drop-down list, so only one image will be placed in the text box.



7. Click the "Preview" tab to check the DataMatrix barcode result.



8. For the reporting services deployment, check out Microsoft Knowledge Base article 842419 with the title "How to grant permissions to a custom assembly that is referenced in a report in Reporting Services", it is required to update the security settings of the .NET Framework to allow the MW6.DataMatrix.dll assembly to run properly.

---

## 6 License

### License agreement

This License Agreement ("LA") is the legal agreement between you and MW6 Technologies, Inc. ("MW6") for the font, and any electronic documentation ("Package"). By using, copying or installing the Package, you agree to be bound by the terms of this LA. If you don't agree to the terms in this LA, immediately remove unused Package.

#### 1. License

\* The Single User License allows the use of the software on **ONE** computer by **ONE** person in your organization.

\* The Site License allows the use of the software at exactly 1 physical site by up to 10,000 users in your organization.

\* The Single Developer License allows 1 developer in your organization the royalty-free distribution (up to 10,000 users) of the software to the third parties.

\* The 2 Developer License allows 2 developers in your organization the royalty-free distribution (up to 10,000 users) of the software to the third parties.

\* The 3 Developer License allows 3 developers in your organization the royalty-free distribution (up to 10,000 users) of the software to the third parties.

\* The 4 Developer License allows 4 developers in your organization the royalty-free distribution (up to 10,000 users) of the software to the third parties.

\* The 5 Developer License allows 5 developers in your organization the royalty-free distribution (up to 10,000 users) of the software to the third parties.

\* The Unlimited Developer License allows unlimited number of developers in your organization the royalty-free distribution (unlimited number of users) of the software to the third parties.

#### 2. User Disclaimer

The software is provided "as is" without warrant of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or noninfringement. MW6 assumes no liability for damages, direct or consequential, which may result from the use of the software. Further, MW6 assumes no liability for losses caused by misuse or abuse of the software. This responsibility rests solely with the end user.

#### 3. Copyright

The software and any electronic documentation are the proprietary products of MW6 and are protected by copyright and other intellectual property laws.

---