

# 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         |
| Columns Property .....                            | 4         |
| Data Property .....                               | 5         |
| ErrorCorrectionLevel Property .....               | 5         |
| HandleTilde Property .....                        | 5         |
| Mode Property .....                               | 6         |
| NarrowBarWidth Property .....                     | 6         |
| Orientation Property .....                        | 6         |
| Rows Property .....                               | 7         |
| TruncateSymbol Property .....                     | 7         |
| Y2XRatio Property .....                           | 7         |
| 2 Methods.....                                    | 7         |
| GetActualSize Method .....                        | 7         |
| GetPatternData Method .....                       | 8         |
| Render Method .....                               | 9         |
| SaveAsImage Method .....                          | 10        |
| SaveAsMemory Method .....                         | 10        |
| SetSize Method .....                              | 11        |
| 3 Enumerations.....                               | 11        |
| ECLevel Enumeration .....                         | 11        |
| Mode Enumeration .....                            | 12        |
| Orientation Enumeration .....                     | 12        |
| <b>Part V How to Use It in Reporting Services</b> | <b>12</b> |
| 1 Create a Report Project.....                    | 12        |
| 2 Add PDF417 Barcodes.....                        | 16        |
| <b>Part VI License</b>                            | <b>22</b> |
| <b>Index</b>                                      | <b>0</b>  |

# 1 Introduction

MW6 PDF417 .NET control is a flexible and reliable .NET component and can create professional 2D PDF417 images for your .NET application, you can save the PDF417 as different image format files. It is easy to print the PDF417 barcode using the PrintDocument Control provided by the .NET Framework.

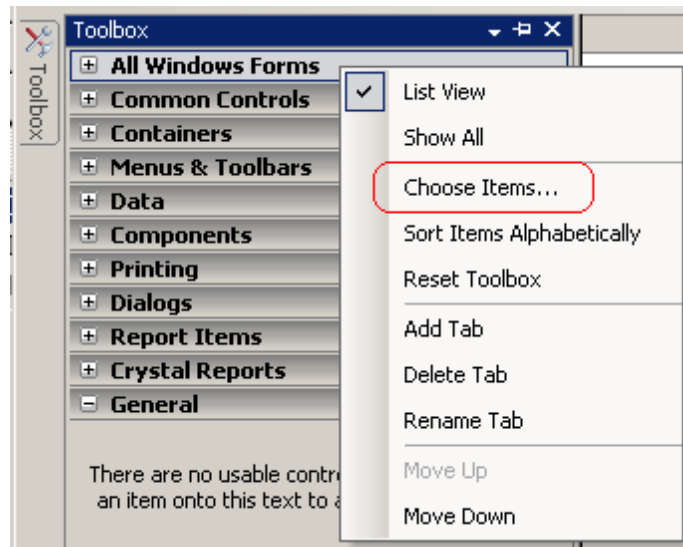
PDF417 is a multi-row, variable-length symbology offering high data capacity and error-correction capability, it is capable of encoding 1100 bytes, 1800 ASCII characters, or 2700 digits.

Every PDF417 symbol is composed of a stack of rows, from a minimum of 3 to a maximum of 90 rows, a PDF417 symbol character consists 17 modules arranged into 4 bars and 4 spaces.

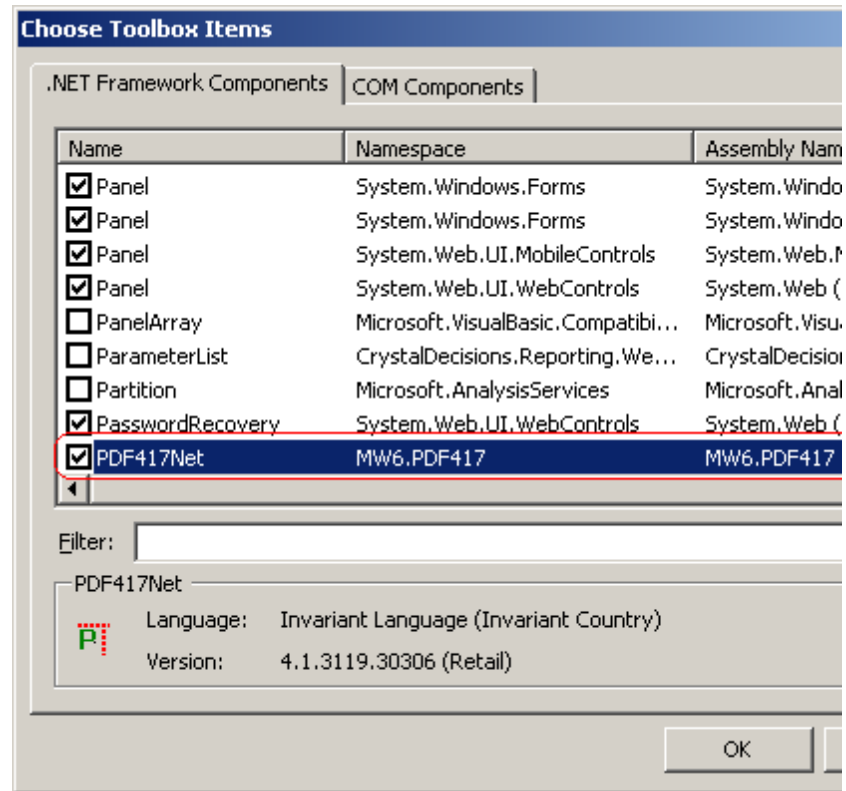
# 2 Installation

## 2.1 Trial Version

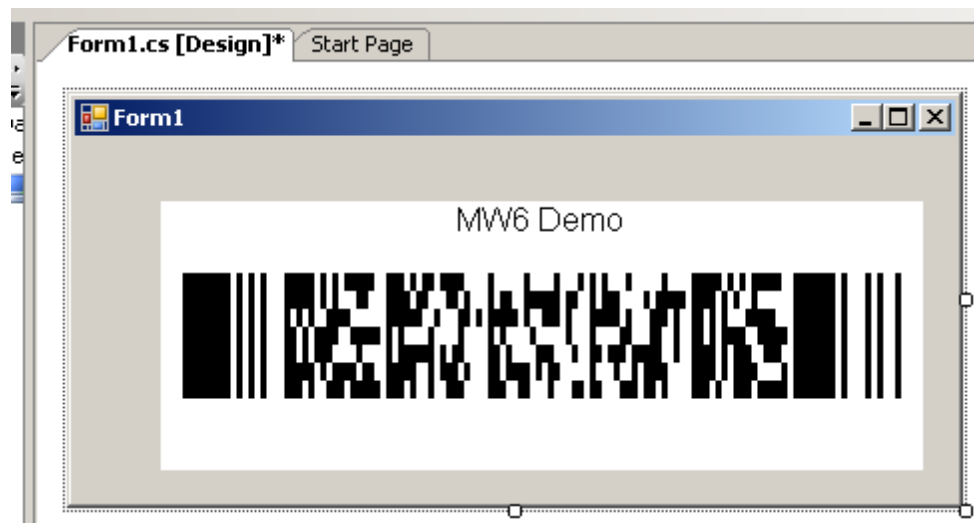
1. The trial version PDF417 .NET control appends the "MW6 Demo" to the string encoded with the PDF417 format.
2. After the installation for the trial version is finished, copy MW6.PDF417.dll in VB.NET or C# demo project sub folder of the destination folder (e.g. "C:\Program Files\MW6 .NET Components\PDF417") 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.PDF417.dll.
-



5. To use the PDF417 .NET 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 PDF417 .NET control if applicable.
2. Complete the installation for the full version PDF417 .NET control, copy MW6.PDF417.dll in VB.NET or C# demo project sub folder of the destination folder (e.g. "C:\Program Files\MW6 .NET

Components\PDF417") to your project folder to replace the trial version PDF417 .NET control .dll file.

## 3 How to Distribute It

If you want to redistribute the PDF417 .NET control as part of your application, simply put MW6.PDF417.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 PDF417 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 PDF417 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 Columns Property

Gets or sets the preferred number of the columns for the PDF417 barcode.

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

```
Public Property Columns As Integer
```

---

```
[C#]  
public int Columns {get; set;}
```

**Remarks**

Typically this property value should be between 3 and 30.

#### 4.1.4 Data Property

Gets or sets the message to encode with PDF417 .NET control

```
[Visual Basic .NET]  
Public Property Data As String
```

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

**Remarks**

The default value is "1234".

#### 4.1.5 ErrorCorrectionLevel Property

Gets or sets the error correction level of the PDF417 barcode.

```
[Visual Basic .NET]  
Public Property ErrorCorrectionLevel As enumECLevel
```

```
[C#]  
public enumECLevel ErrorCorrectionLevel {get; set;}
```

#### 4.1.6 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 PDF417 .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).

### 4.1.7 Mode Property

Gets or sets the encoding mode of the PDF417 barcode.

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

```
Public Property Mode As enumMode
```

```
[C#]
```

```
public enumMode Mode {get; set;}
```

### 4.1.8 NarrowBarWidth Property

Gets or sets the narrow bar width in centimeters.

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

```
Public Property NarrowBarWidth As float
```

```
[C#]
```

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

#### Remarks

The default value is 0.07, internally our PDF417 .NET control converts narrow bar width 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 NarrowBarWidth property to 0.04,  $size\_in\_pixels = 0.04 * 96 / 2.54 = 1.5118$ , round up 1.5118 to 2, so actual narrow bar width is 2 pixels.

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

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

Different NarrowBarWidth property values might end up with same narrow bar width in pixels due to performing the rounding operations.

### 4.1.9 Orientation Property

Gets or sets the orientation of the PDF417 barcode.

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

```
Public Property Orientation As enumOrientation
```

```
[C#]
```

---

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

#### 4.1.10 Rows Property

Gets or sets the preferred number of the rows for the PDF417 barcode.

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

```
Public Property Rows As Integer
```

```
[C#]
```

```
public int Rows {get; set;}
```

##### Remarks

Typically this property value should be between 3 and 90.

#### 4.1.11 TruncateSymbol Property

Gets or sets a boolean flag indicating whether the right side of the PDF417 barcode should be removed or not.

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

```
Public Property TruncateSymbol As Boolean
```

```
[C#]
```

```
public bool TruncateSymbol {get; set;}
```

#### 4.1.12 Y2XRatio Property

Gets or sets the ratio of the height of the row to the width of the narrow element.

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

```
Public Property Y2XRatio As float
```

```
[C#]
```

```
public float Y2XRatio {get; set;}
```

##### Remarks

The default value is 3.0, typically this property value is between 3 and 6.

## 4.2 Methods

### 4.2.1 GetActualSize Method

Gets the actual size of the PDF417 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 PDF417 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 PDF417 barcode (in pixels).

#### *ActualHeight*

A pointer to the variable that receives the height of the PDF417 barcode (in pixels).

## 4.2.2 GetPatternData Method

Gets the PDF417 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 PDF417 barcode pattern matrix data row by row from the top left matrix corner, '1' indicates the narrow bar and '0' indicates the narrow space.

---



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 PDF417 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 PDF417 barcode pattern matrix data and render the PDF417 barcode onto any device such as the printer, only *Columns*, *Data*, *ErrorCorrectionLevel*, *HandleTilde*, *Mode*, *Rows* and *TruncateSymbol* properties affect the pattern matrix data output.

Based on the *Orientation* property value, rotate the pattern matrix accordingly before you render the PDF417 barcode onto a device.

If the *Orientation* property value is 0 (no rotation) or 2 (180-degree rotation), the ratio of the height to the width for the rectangles (the narrow bar and the narrow space) must be equal to the *Y2XRatio* property value, the width of the rectangles is specified by the *NarrowBarWidth* property.

If the *Orientation* property value is 1 (90-degree rotation) or 3 (270-degree rotation), the ratio of the width to the height for the rectangles (the narrow bar and the narrow space) must be equal to the *Y2XRatio* property value, the height of the rectangles is specified by the *NarrowBarWidth* property.

### **4.2.3 Render Method**

Renders the PDF417 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 PDF417 barcode.

#### 4.2.4 SaveAsImage Method

Exports the PDF417 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 PDF417 barcode image.

*ImgFormat*

Specifies the image format.

##### Remarks

Before you call this method, use `GetActualSize()` method to obtain the actual size of PDF417 barcode and use `SetSize()` method to set image size by adding surrounding white space around the PDF417 barcode.

##### See Also

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

#### 4.2.5 SaveAsMemory Method

Exports the PDF417 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 PDF417 barcode image.

*ImgFormat*

---

Specifies the image format.

### Remarks

Before you call this method, use `GetActualSize()` method to obtain the actual size of the PDF417 barcode and use `SetSize()` method to set image size by adding surrounding white space around the PDF417 barcode.

### See Also

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

## 4.2.6 SetSize Method

Sets the size of the image which contains the PDF417 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 PDF417 barcode, then use this method to set image size by adding surrounding white space around the PDF417 barcode.

### See Also

[GetActualSize\(\) Method](#)

## 4.3 Enumerations

### 4.3.1 ECLevel Enumeration

An enumeration type for all possible error correction level values.

### Members

| Name | Comment |
|------|---------|
|------|---------|

|       |                          |
|-------|--------------------------|
| ecl_0 | Error Correction Level 0 |
| ecl_1 | Error Correction Level 1 |
| ecl_2 | Error Correction Level 2 |
| ecl_3 | Error Correction Level 3 |
| ecl_4 | Error Correction Level 4 |
| ecl_5 | Error Correction Level 5 |
| ecl_6 | Error Correction Level 6 |
| ecl_7 | Error Correction Level 7 |
| ecl_8 | Error Correction Level 8 |

### 4.3.2 Mode Enumeration

An enumeration type for all possible encoding mode values.

#### Members

| Name     | Comment     |
|----------|-------------|
| mdBinary | Binary mode |
| mdText   | Text mode   |
| mdAuto   | Auto mode   |

### 4.3.3 Orientation Enumeration

An enumeration type for all possible orientation values.

#### Members

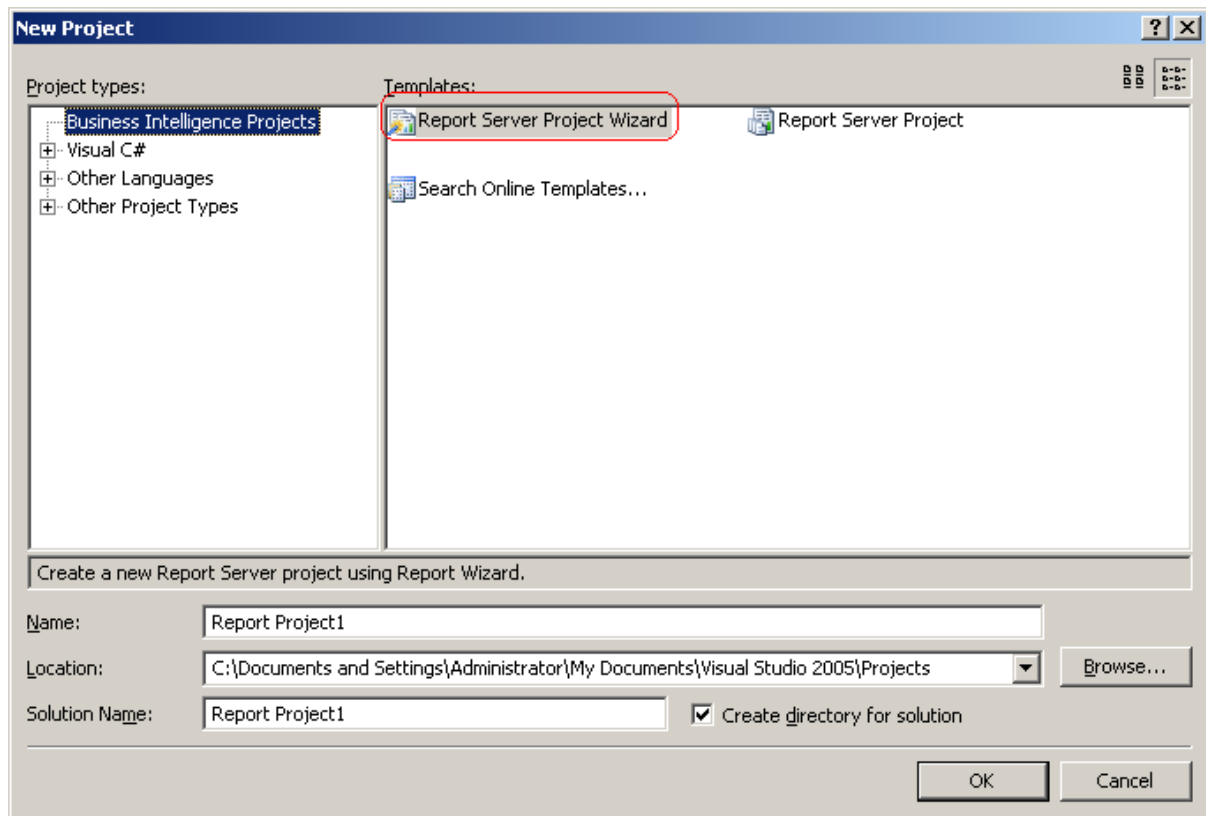
| Name  | Comment     |
|-------|-------------|
| or0   | 0 Degree    |
| or90  | 90 Degrees  |
| or180 | 180 Degrees |
| or270 | 270 Degrees |

## 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...

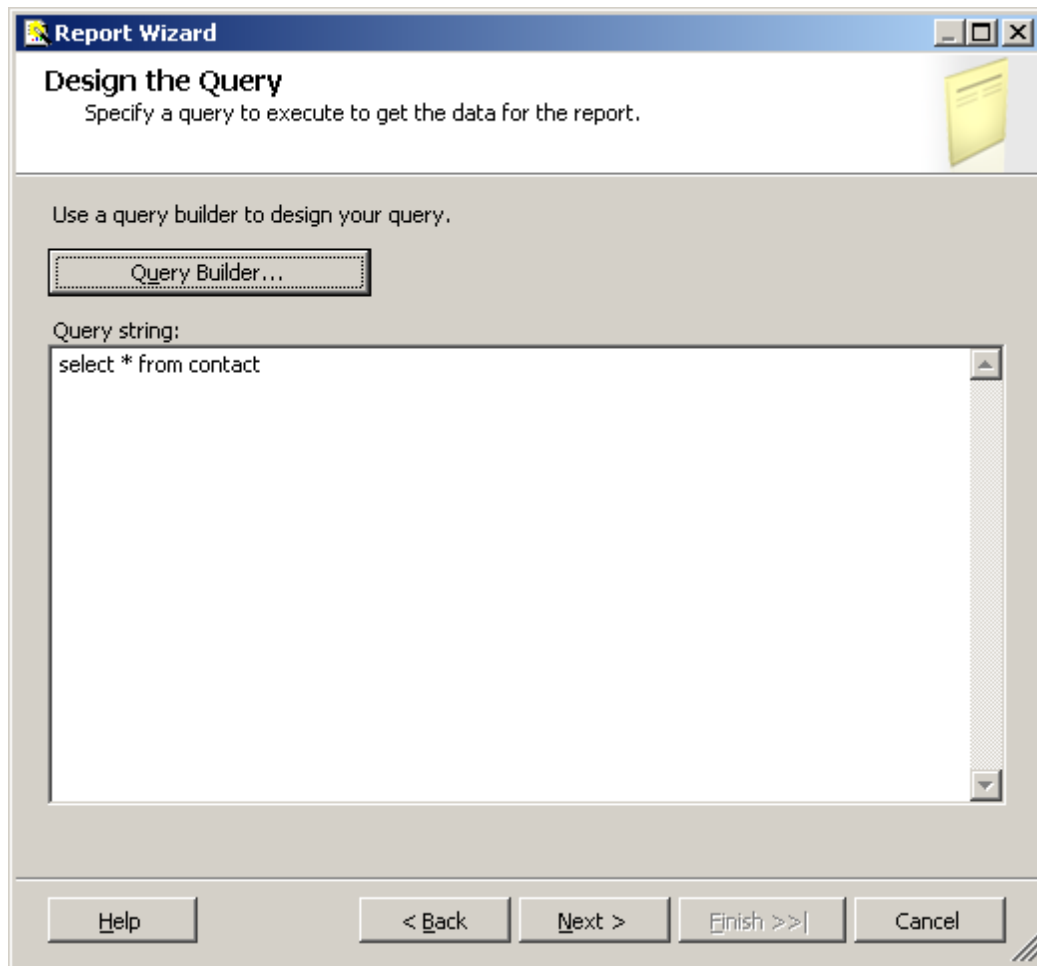
Logical name:

Advanced...

Test Connection OK Cancel

The screenshot shows the 'Report Wizard' dialog box with the title 'Select the Data Source'. The instruction reads: 'Select a data source from which to obtain data for this report or create a new data source.' There are two radio buttons: 'Shared data source' (unselected) and 'New data source' (selected). Under 'New data source', there is a 'Name:' text box containing 'DataSource1', a 'Type:' dropdown menu set to 'Microsoft SQL Server', and a 'Connection string:' text box containing 'Data Source=PC-V4OG4L4T6\SQLEXPRESS\_NEW;Initial Catalog=Customers'. To the right of the connection string box are two buttons: 'Edit...' and 'Credentials...'. At the bottom, there is a 'Make this a shared data source' checkbox (unchecked) and a row of navigation buttons: 'Help', '< Back', 'Next >', 'Finish >>', and '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



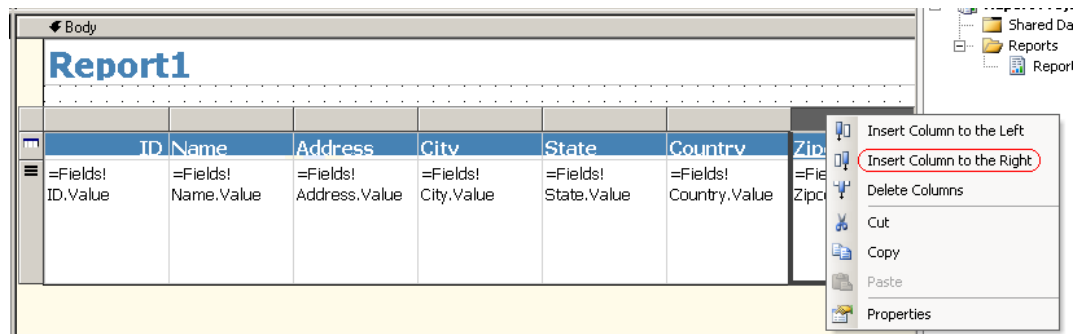
4. In the following dialogs, choose the appropriate options for the type of the report, the way of how to group the data in the table and the table style, then click the **"Finish"** button.

## 5.2 Add PDF417 Barcodes

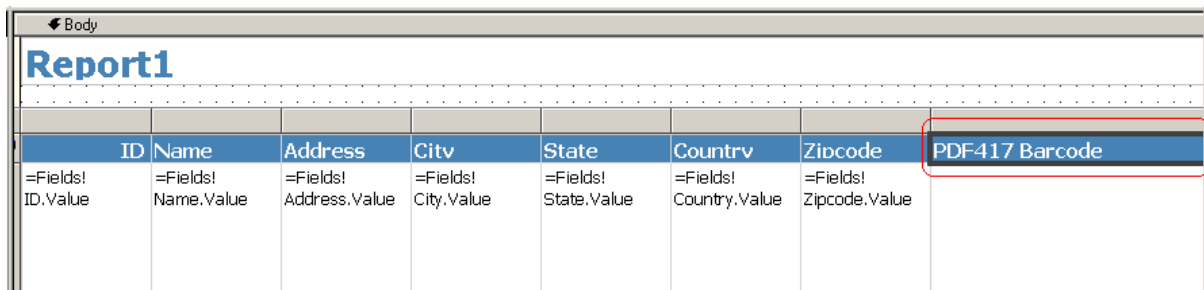
Follow the instructions to add the PDF417 barcodes to the report.


1. In order to use the PDF417 .NET control in the Reporting Services, please copy MW6.PDF417.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"**.



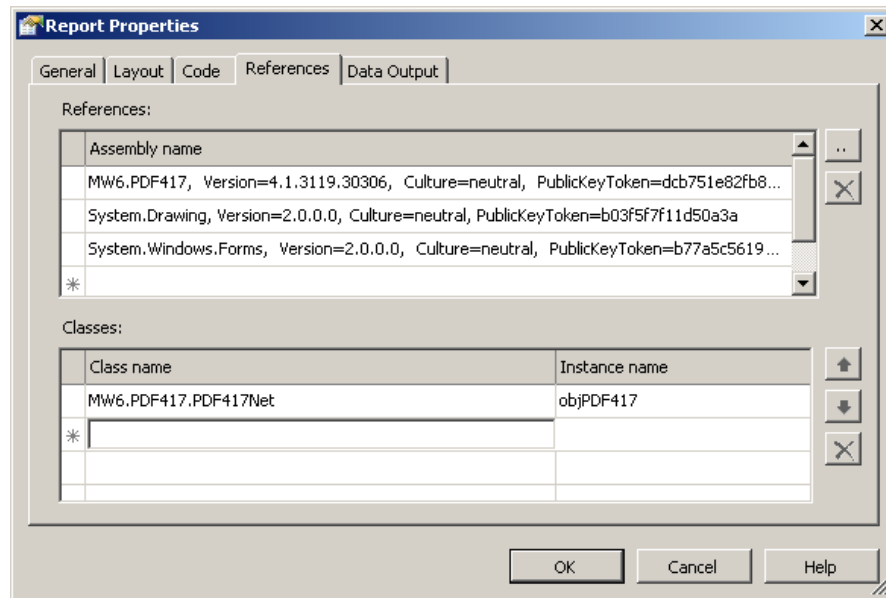


3. Change the column title to the "PDF417 Barcode".



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.PDF417.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.PDF417.PDF417Net" in the "Class name" box, enter "objPDF417" in the "Instance name" box to create an assembly object to use in the code to retrieve the PDF417 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 PDF417 barcode image byte stream, modify the code a bit to meet your application requirements.

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

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

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

    objPDF417.Data = DataStr

    ' Narrow Bar Width
    objPDF417.NarrowBarWidth = 0.05

    ' Y2X Ratio
    objPDF417.Y2XRatio = 3.0

    ' 0 for Binary Mode, 1 for Text Mode, 2 for Auto mode
    objPDF417.Mode = 1

    ' Error Correction Level (0 - 8)
    objPDF417.ErrorCorrectionLevel = 2

    ' PDF417 Rows (3 - 90)
    objPDF417.Rows = 7

    ' PDF417 Columns (3 - 30)
    objPDF417.Columns = 3

    ' Truncate Symbol or not?
    objPDF417.TruncateSymbol = False

    ' Handle Tilde or not?
    objPDF417.HandleTilde = True

    ' Orientation
    objPDF417.Orientation = 0

    objPDF417.GetActualSize(True, Nothing, ActualWidth, ActualHeight)

    ExtraWidth = 20
    ExtraHeight = 60

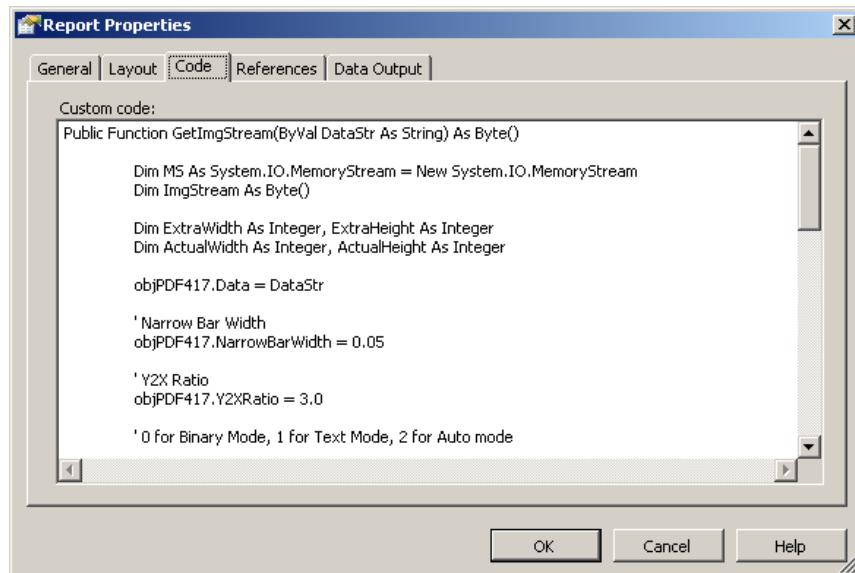
    ' Image size = barcode size + extra space
    objPDF417.SetSize(ActualWidth + ExtraWidth, ActualHeight + ExtraHeight)

    ' Save the barcode image in the memory using the Jpeg format
    objPDF417.SaveAsMemory(MS, System.Drawing.Imaging.ImageFormat.Jpeg)

    ImgStream = MS.ToArray

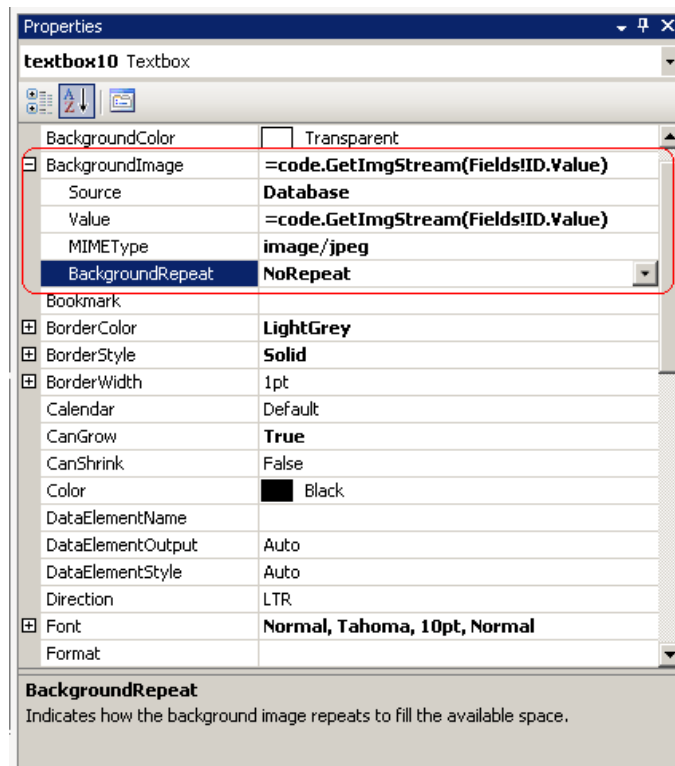
    MS.Close()
```

```
Return ImgStream  
End Function
```



6. Change the "**BackgroundImage**" property of the text box in the "**PDF417 Barcode**" column to display the PDF417 barcode image encoding ID value.

- A. **Source:** Select the Database from the drop-down list, since the data will be pulled from a database field.
- B. **Value:** Enter the string "=code.GetImgStream(Fields!ID.Value)", it will ask the report to use the GetImgStream() function to retrieve the PDF417 barcode image byte stream for the ID field value of the database.
- C. **MIMETYPE:** Select the image/jpeg from the drop-down list, since the PDF417 barcode image is in jpeg format.
- D. **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 PDF417 barcode result.

The screenshot shows the 'Report1.rdl [Design]' window in 'Preview' mode. The report displays a table with the following data:

| ID | Name       | Address         | City      | State      | Country | Zipcode | PDF417 Barcode |
|----|------------|-----------------|-----------|------------|---------|---------|----------------|
| 10 | John Smith | 123 Cool Street | One City  | One State  | USA     | 98765   | MW6 Demo<br>   |
| 11 | Tedd Hill  | 456 Warm Street | Some City | Some State | USA     | 12345   | MW6 Demo<br>   |

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.PDF417.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, **each individual developer requires a separate Single Developer License as long as he or she needs access to MW6's product(s) and document(s).**

\* 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.

---