

Table of Contents

Foreword	0
Part I Introduction	2
Part II Installation	2
1 Trial Version.....	2
2 Full Version.....	3
Part III How to Distribute It	4
Part IV Reference Guide	4
1 Properties.....	4
BackColor Property	4
BarColor Property	4
CorrectionLevel Property	4
Data Property	5
HandleTilde Property	5
Mode Property	5
ModuleSize Property	6
Orientation Property	6
PreferredFormat Property	6
2 Methods.....	7
GetActualRC Method	7
GetActualSize Method	8
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	12
Part V How to Use It in Reporting Services	13
1 Create a Report Project.....	13
2 Add Aztec Barcodes.....	17
Part VI License	23
Index	0

1 Introduction

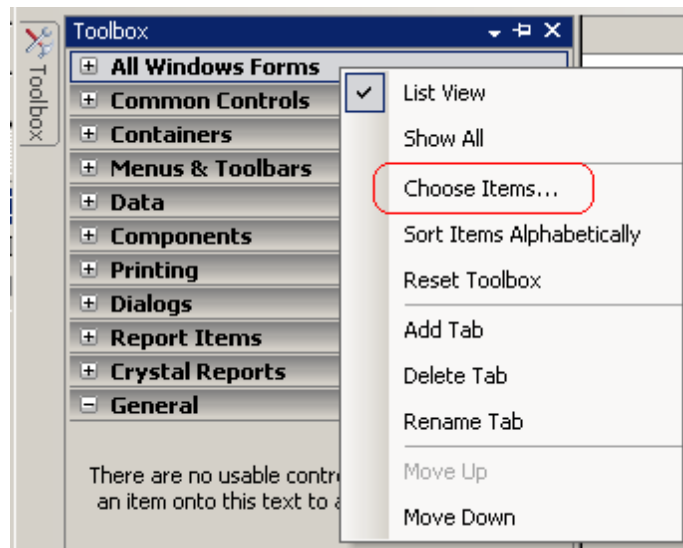
MW6 Aztec .NET control is a flexible and reliable .NET component and can create professional 2D Aztec images for your .NET application, you can save the Aztec as different image format files. It is easy to print the Aztec barcode using the PrintDocument Control provided by the .NET Framework.

Aztec is designed to pack a lot of information in a very small space, it is capable of encoding 1914 bytes, 3067 alphanumeric characters, or 3832 numeric digits.

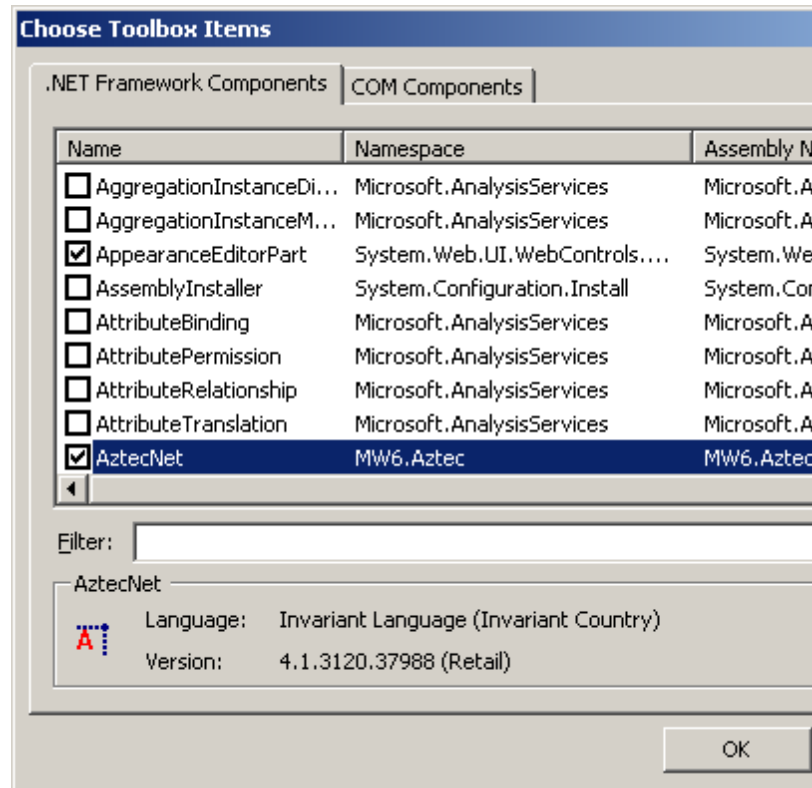
2 Installation

2.1 Trial Version

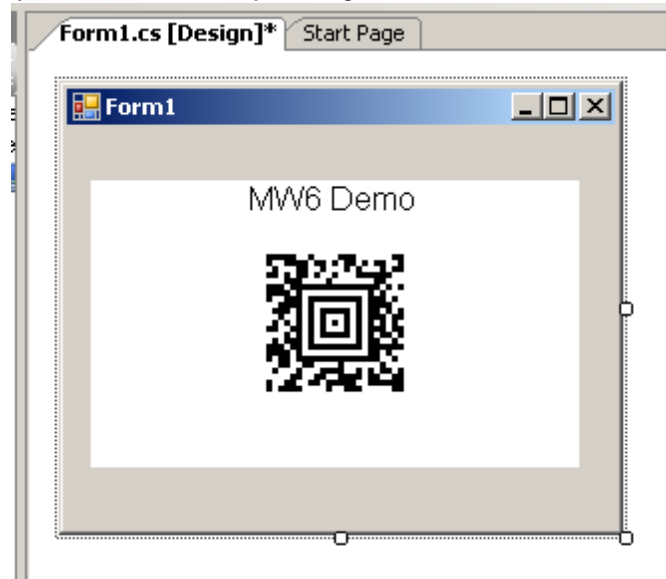
1. The trial version Aztec .NET control appends "MW6 Demo" to the string encoded with the Aztec format.
2. After the installation for the trial version is finished, copy MW6.Aztec.dll in VB.NET or C# demo project sub folder of the destination folder (e.g. "C:\Program Files\MW6 .NET Components\Aztec")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

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

3 How to Distribute It

If you want to redistribute the Aztec .NET control as part of your application, simply put MW6.Aztec.dll into the 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 Aztec 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 Aztec 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 CorrectionLevel Property

Gets or sets the percentage of errors which can be recovered.

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

```
Public Property CorrectionLevel As Integer
```

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

Remarks

The recommended value is 23 percent, the valid value should be between 5 percent and 95 percent.

4.1.4 Data Property

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

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

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

Remarks

The default value is "12".

4.1.5 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 Aztec .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).

4.1.6 Mode Property

Gets or sets the encoding mode of the Aztec barcode.

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

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

4.1.7 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 Aztec .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.

4.1.8 Orientation Property

Gets or sets the orientation of the Aztec barcode.

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

```
Public Property Orientation As enumOrientation
```

```
[C#]
```

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

4.1.9 PreferredFormat Property

Gets or sets the format of the Aztec barcode.

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

```
Public Property PreferredFormat As enumPreferredFormat
```

```
[C#]
```

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

Remarks

If you set PreferredFormat to pfAuto (Auto format), our Aztec .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 Aztec .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 Aztec 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 Aztec barcode.

ActualCols

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

Remarks

If you set PreferredFormat to pfAuto (Auto format), Aztec .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, Aztec .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 Aztec 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 Aztec 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 Aztec barcode (in pixels).

ActualHeight

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

4.2.3 GetPatternData Method

Gets the Aztec 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 Aztec 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 Aztec 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 Aztec barcode pattern matrix data and render the Aztec 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 Aztec barcode onto a device.

4.2.4 Render Method

Renders the Aztec 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 Aztec barcode.

4.2.5 SaveAsImage Method

Exports the Aztec 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 Aztec barcode image.

ImgFormat

Specifies the image format.

Remarks

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

See Also

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

4.2.6 SaveAsMemory Method

Exports the Aztec 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 Aztec barcode image.

ImgFormat

Specifies the image format.

Remarks

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

See Also

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

4.2.7 SetSize Method

Sets the size of the image which contains the Aztec 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 Aztec barcode, then use this method to set image size by adding surrounding white space around the Aztec 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 Aztec 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 26, which means that up to 26 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
mdAuto	Auto mode for mainly encoding ASCII characters (0-127)
mdBinary	Binary mode for 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	Capacity (in	Capacity (in	Capacity (in
------	-------------	--------------	--------------	--------------

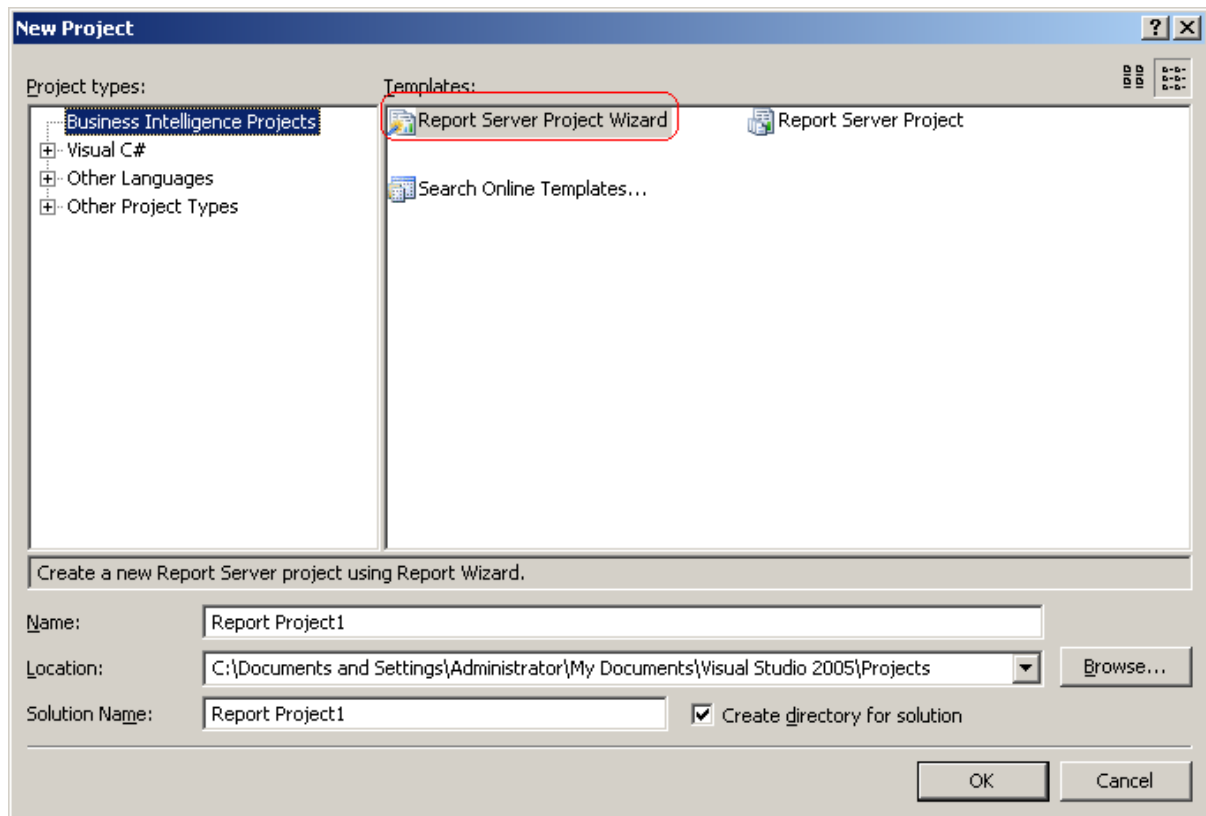
		digits)	alphanumeric characters)	bytes)
pfAuto	Auto format			
pf15X15_Compact	15 X 15 compact format	13	12	6
pf19X19	19 X 19	18	15	8
pf19X19_Compact	19 X 19 compact format	40	33	19
pf23X23	23 X 23	49	40	24
pf23X23_Compact	23 X 23 compact format	70	57	33
pf27X27	27 X 27	84	68	40
pf27X27_Compact	27 X 27 compact format	110	89	53
pf31X31	31 X 31	128	104	62
pf37X37	37 X 37	178	144	87
pf41X41	41 X 41	232	187	114
pf45X45	45 X 45	294	236	145
pf49X49	49 X 49	362	291	179
pf53X53	53 X 53	433	348	214
pf57X57	57 X 57	516	414	256
pf61X61	61 X 61	601	482	298
pf67X67	67 X 67	691	554	343
pf71X71	71 X 71	793	636	394
pf75X75	75 X 75	896	718	446
pf79X79	79 X 79	1008	808	502
pf83X83	83 X 83	1123	900	559
pf87X87	87 X 87	1246	998	621
pf91X91	91 X 91	1378	1104	687
pf95X95	95 X 95	1511	1210	753
pf101X101	101 X 101	1653	1324	824
pf105X105	105 X 105	1801	1442	898
pf109X109	109 X 109	1956	1566	976
pf113X113	113 X 113	2116	1694	1056
pf117X117	117 X 117	2281	1826	1138
pf121X121	121 X 121	2452	1963	1224
pf125X125	125 X 125	2632	2107	1314
pf131X131	131 X 131	2818	2256	1407
pf135X135	135 X 135	3007	2407	1501
pf139X139	139 X 139	3205	2565	1600
pf143X143	143 X 143	3409	2728	1702
pf147X147	147 X 147	3616	2894	1806
pf151X151	151 X 151	3832	3067	1914

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

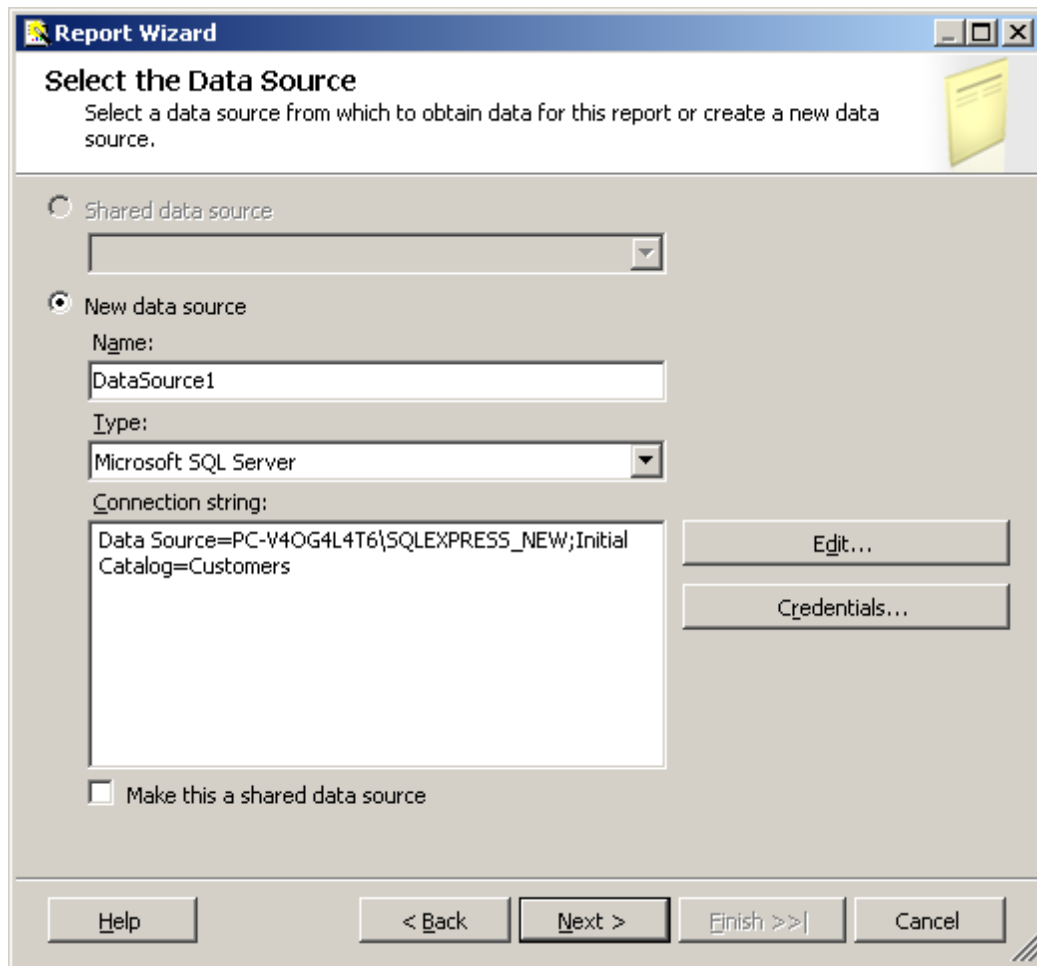
Connect to a database

Select or enter a database name:
Customers []

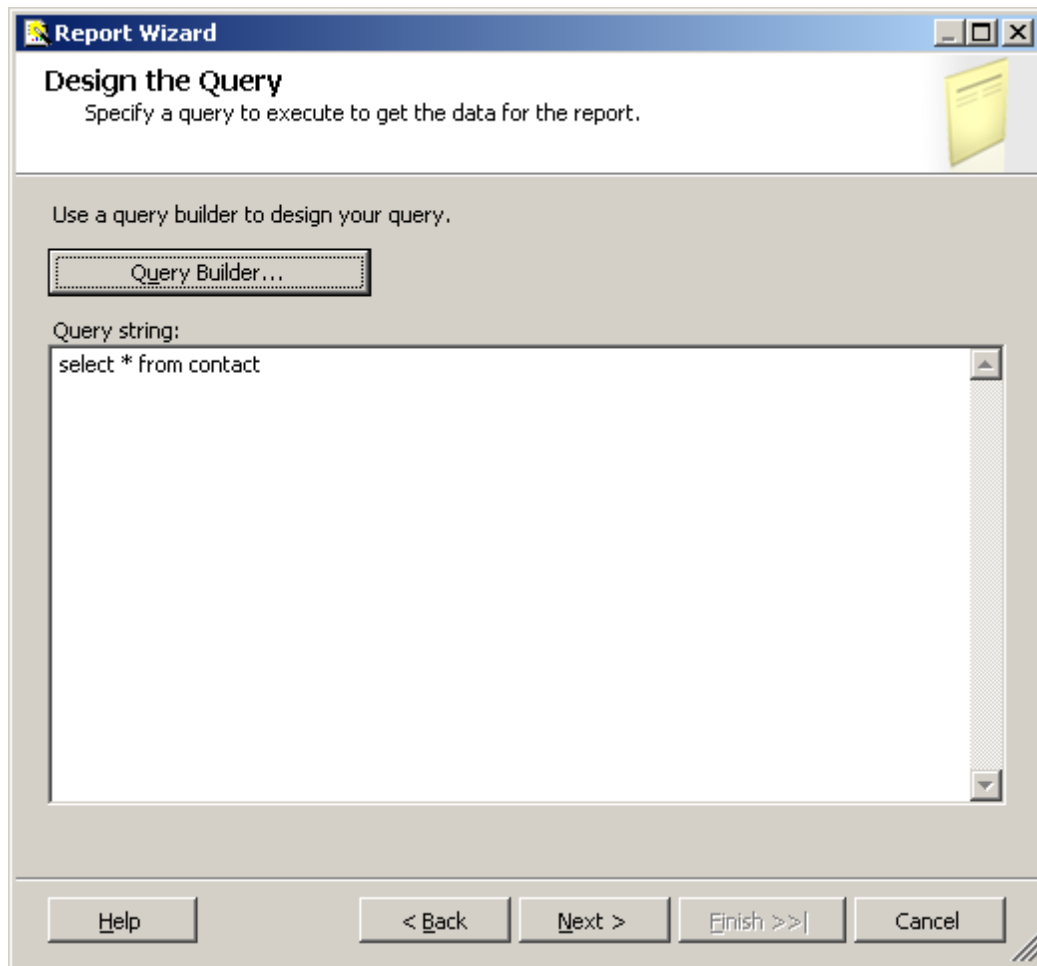
Attach a database file:
[] [Browse...]
Logical name: []

[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

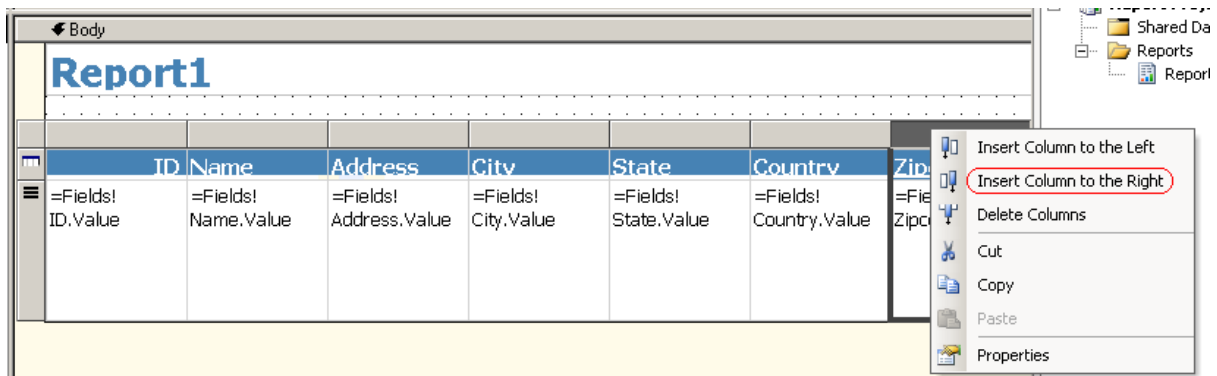


4. In the following dialogs, choose the appropriate options for the type of the report, the way of how to group the data in the table and the table style, then click the "**Finish**" button.

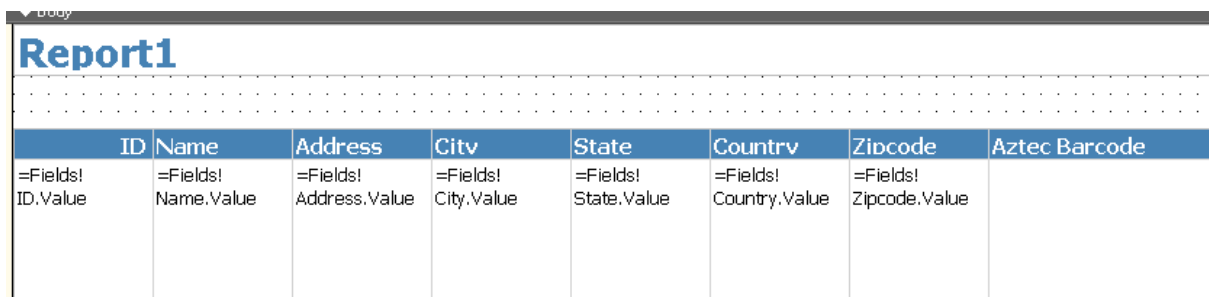
5.2 Add Aztec Barcodes


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

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

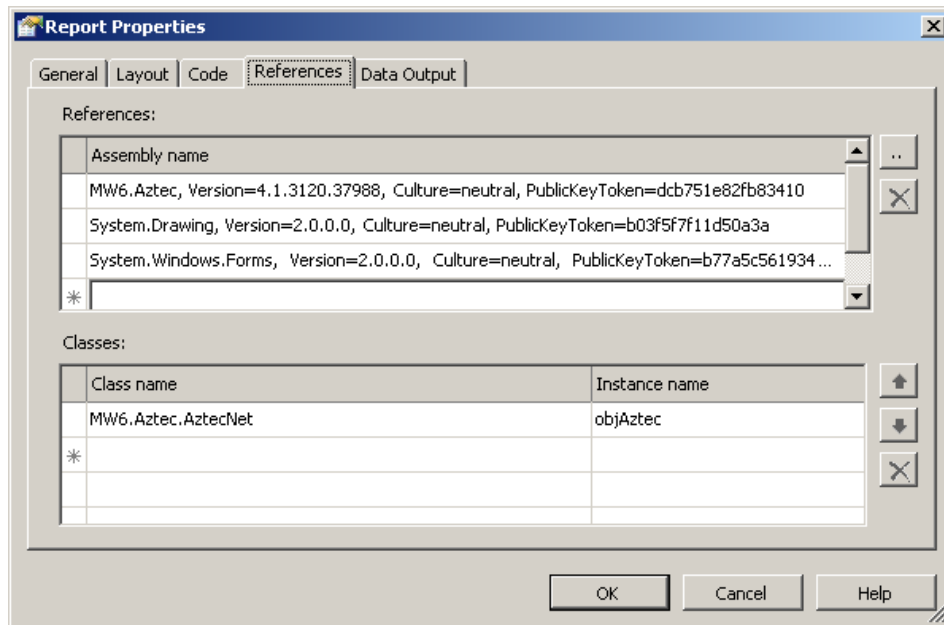


3. Change the column title to the "Aztec Barcode".



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.Aztec.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.Aztec.AztecNet" in the "Class name" box, enter "objAztec" in the "Instance name" box to create an assembly object to use in the code to retrieve the Aztec 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 Aztec 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
    objAztec.Data = DataStr

    ' Module Size
    objAztec.ModuleSize = 0.07

    ' Error Correction Level
    objAztec.CorrectionLevel = 23

    ' Mode
    objAztec.Mode = 0

    ' Preferred Format
    objAztec.PreferredFormat = 0

    ' Orientation
    objAztec.Orientation = 0

    ' Handle Tilde Character?
    objAztec.HandleTilde = True

    ExtraWidth = 60
    ExtraHeight = 80

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

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

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

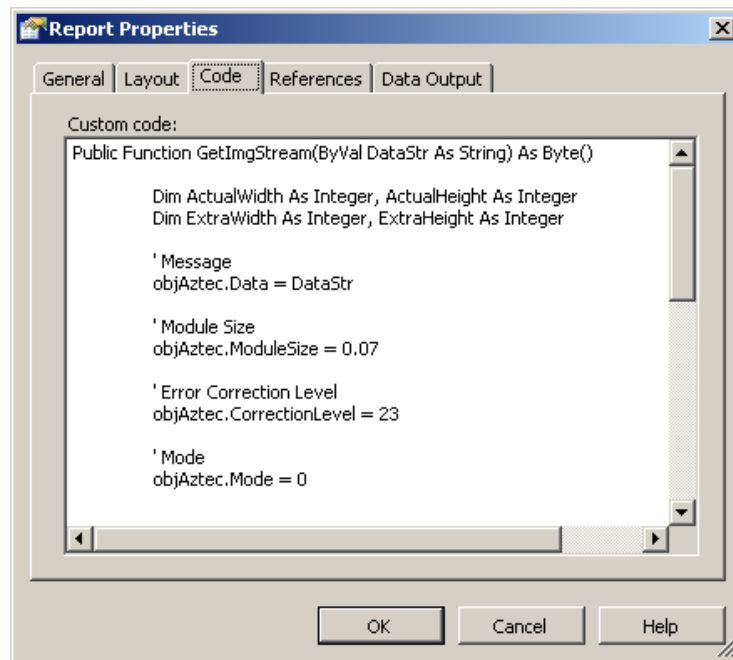
    objAztec.SaveAsMemory(MS, System.Drawing.Imaging.ImageFormat.Jpeg)

    ImgStream = MS.ToArray

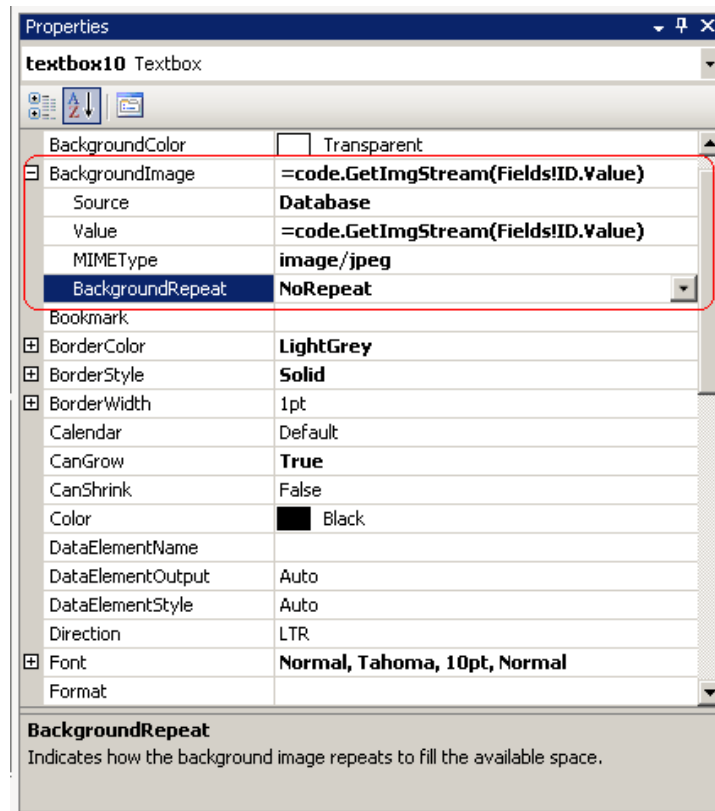
    MS.Close()

    Return ImgStream

End Function
```





6. Change the "**BackgroundImage**" property of the text box in the "**Aztec Barcode**" column to display the Aztec 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 Aztec 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 Aztec 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 Aztec barcode result.

The screenshot shows a report preview window titled 'Report1'. The report contains a table with the following data:

ID	Name	Address	City	State	Country	Zipcode	Aztec Barcode
10	John Smith	123 Cool Street	One City	One State	USA	98765	MW6 Demo 
11	Tedd Hill	456 Warm Street	Some City	Some State	USA	12345	MW6 Demo 

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.Aztec.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.