

Table of Contents

Foreword	0
Part I Introduction	3
Part II Installation	3
1 Trial Version.....	3
2 Full Version.....	5
Part III How to Distribute It	6
Part IV Reference Guide	6
1 Properties.....	6
BackColor Property	6
BarColor Property	6
BarHeight Property	6
BearerBarType Property	7
BorderWidth Property	7
CheckDigit Property	7
CheckDigitToText Property	8
CodabarStartChar Property	8
CodabarEndChar Property	8
Data Property	9
NarrowBarWidth Property	9
Orientation Property	10
ShowText Property	10
Supplement Property	10
SupplementGap Property	11
SupplementType Property	11
SymbologyType Property	11
TextFont Property	11
UPCESystem Property	12
Wide2NarrowRatio Property	12
2 Methods.....	12
CodeOne Method	12
GetActualSize Method	13
MicroPDF417 Method	13
MicroQRCode Method	14
Render Method	14
SaveAsImage Method	15
SaveAsMemory Method	15
SetSize Method	16
3 Enumerations.....	17
BearerBarType Enumeration	17
CodaBarChar Enumeration	17
Orientation Enumeration	17
SupplementType Enumeration	17
SymbologyType Enumeration	18

UPCESystem Enumeration	20
Part V How to Use It in Reporting Services	20
1 Create a Report Project.....	20
2 Add Barcodes.....	24
Part VI License	29
Index	0

1 Introduction

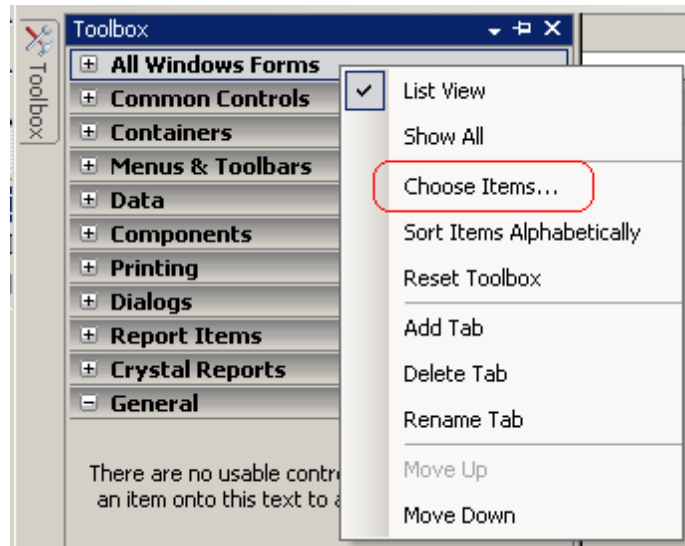
MW6 Barcode .NET Control supports over 90 symbologies including Code 39, Code 128, GS1-128, EAN 13, EAN 8, UPC-A, UPC-E, Royal Mail 4 State, USPS OneCode, Deutsche Post Identcode, Deutsche Post Leitcode, Japan Postal Code, Micro PDF417, Micro QRCode, CodaBlock-F, Code 16K and Code 49, please check out this page for complete list of featured barcodes.

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

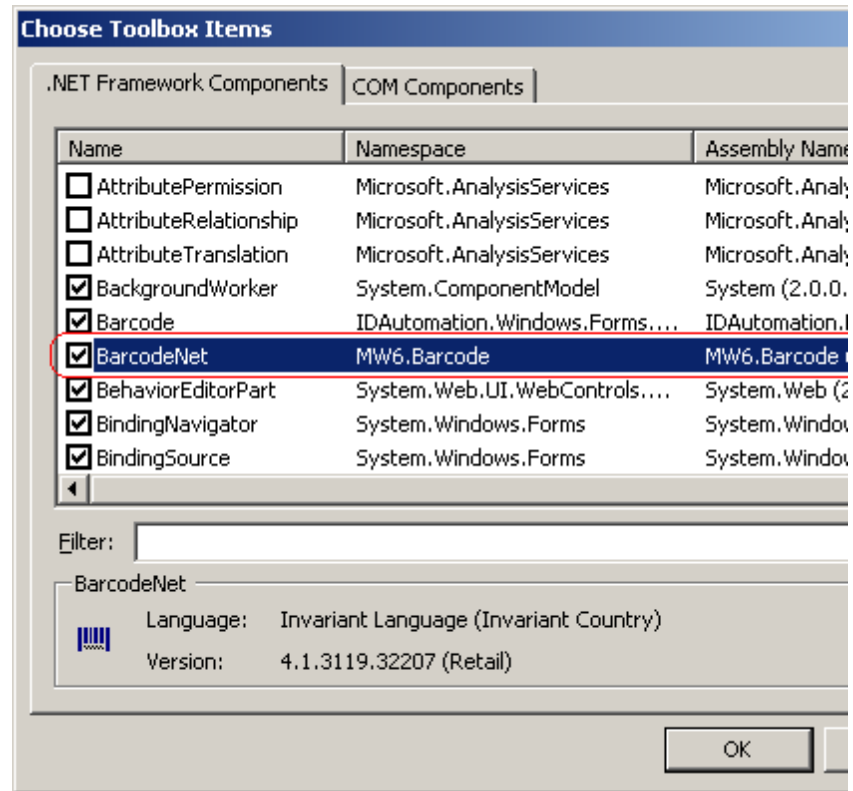
2 Installation

2.1 Trial Version

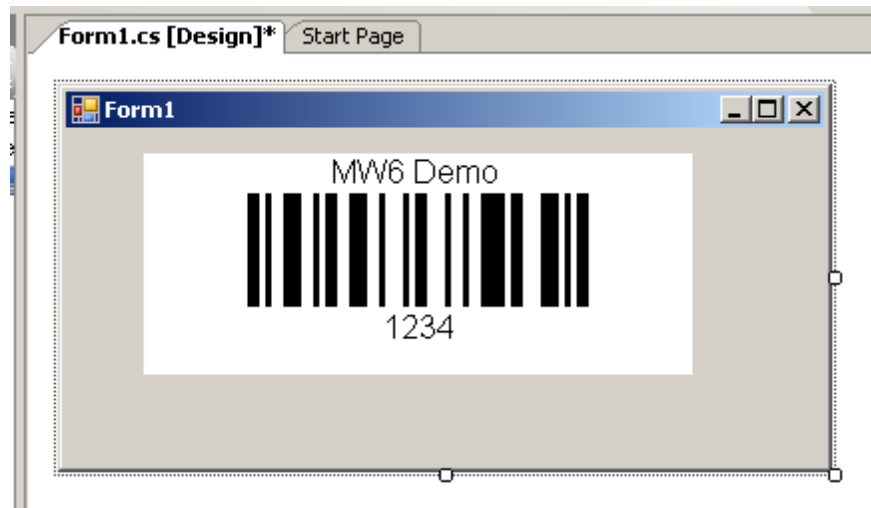
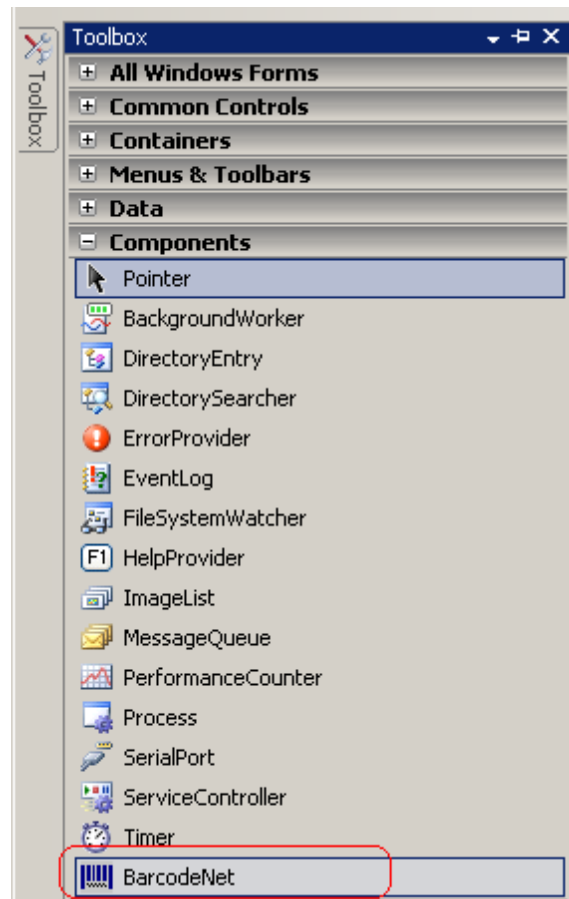
1. The trial version Barcode .NET control adds "MW6 Demo" at the top of the barcode.
2. After the installation for the trial version is finished, copy MW6.Barcode.dll in VB.NET or C#.NET demo project sub folder of the destination folder (e.g. "C:\Program Files\MW6 .NET Components \Barcode") to your project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



4. In the "**Choose Toolbox Items**" dialog, select the ".NET Framework Components", click the "**Browse**" and select MW6.Barcode.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 barcode .NET control if applicable.
2. Complete the installation for the full version barcode .NET control, copy MW6.Barcode.dll in VB.NET or C# demo project sub folder of the destination folder (e.g. "C:\Program Files\MW6 .NET Components

Barcode\2003") to your project folder to replace the trial version Barcode .NET control .dll file.

3 How to Distribute It

If you want to redistribute the Barcode .NET control as part of your application, simply put MW6.Barcode.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 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 barcode and the text.

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

```
Public Property BarColor As Color
```

```
[C#]
```

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

Remarks

The default value is black color.

4.1.3 BarHeight Property

Gets or sets the bar height in centimeters.

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

```
Public Property BarHeight As float
```

```
[C#]  
public float BarHeight {get; set;}
```

Remarks

The default value is 1.5, internally our barcode .NET control converts bar height 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 BarHeight property to 1.5, $size_in_pixels = 1.5 * 96 / 2.54 = 56.6929$, round up 56.6929 to 57, so actual bar height is 57 pixels.

(2) Set BarHeight property to 1.52, $size_in_pixels = 1.52 * 96 / 2.54 = 57.4488$, round down 57.4488 to 57, so actual bar height is 57 pixels.

(3) Set BarHeight property to 1.54, $size_in_pixels = 1.54 * 96 / 2.54 = 58.2047$, round down 58.2047 to 58, so actual bar height is 58 pixels.

Different BarHeight property values might end up with same bar height in pixels due to performing rounding operations.

4.1.4 BearerBarType Property

Gets or sets the style of bearer bar for a few kinds of barcodes.

```
[Visual Basic .NET]  
Public Property Bearer Bar As enumBearerBarType
```

```
[C#]  
public enumBearerBarType BearerBar {get; set;}
```

4.1.5 BorderWidth Property

Gets or sets the border width in centimeters.

```
[Visual Basic .NET]  
Public Property BorderWidth As float
```

```
[C#]  
public float BorderWidth {get; set;}
```

4.1.6 CheckDigit Property

Gets or sets a boolean flag indicating whether the check digit is required or not.

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

```
Public Property CheckDigit As Boolean
```

```
[C#]
```

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

Remarks

The default value is FALSE, the check digit is mandatory for all symbologies except for Code 39, Industrial 2 of 5 and Codabar.

4.1.7 CheckDigitToText Property

Gets or sets a boolean flag indicating whether the check digit should be displayed in the human readable text or not.

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

```
Public Property CheckDigitToText As Boolean
```

```
[C#]
```

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

Remarks

The default value is FALSE.

4.1.8 CodabarStartChar Property

Gets or sets the start character of CodaBar.

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

```
Public Property CodabarStartChar As enumCodeBarChar
```

```
[C#]
```

```
public enumCodeBarChar CodabarStartChar {get; set;}
```

4.1.9 CodabarEndChar Property

Gets or sets the end character of CodaBar.

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

```
Public Property CodabarEndChar As enumCodeBarChar
```

```
[C#]
```

```
public enumCodeBarChar CodabarEndChar {get; set;}
```

4.1.10 Data Property

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

[Visual Basic .NET]

```
Public Property Data As String
```

[C#]

```
public string Data {get; set;}
```

Remarks

The default value is "1234".

If the SymbologyType property is set to UCC/EAN128 (GS1-128), you could use a "FNC1" string to indicate the end of a varied-length data field. For example, set the "Data" property to "(01)12345678901234(10)12345FNC1(11)080101", "FNC1" is used to indicate the end of the data field after the Application Identifier (AI) #10, since AI #10 allows the corresponding data field to have 1-20 alphanumeric characters.



4.1.11 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 barcode .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, $\text{size_in_pixels} = 0.06 * 96 / 2.54 = 2.2677$, round down 2.2677 to 2, so actual narrow bar width is 2 pixels.

(3) Set NarrowBarWidth property to 0.07, $\text{size_in_pixels} = 0.07 * 96 / 2.54 = 2.6456$, 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 rounding operations.

4.1.12 Orientation Property

Gets or sets the orientation of the barcode.

[Visual Basic .NET]

```
Public Property Orientation As enumOrientation
```

[C#]

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

4.1.13 ShowText Property

Gets or sets a boolean flag indicating whether the human readable text should be displayed or not.

[Visual Basic .NET]

```
Public Property ShowText As Boolean
```

[C#]

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

Remarks

The default value is TRUE.

4.1.14 Supplement Property

Gets or sets the supplement string to encode with UPC or EAN.

[Visual Basic .NET]

```
Public Property Supplement As String
```

[C#]

```
public string Supplement {get; set;}
```

Remarks

The default value is an empty string.

4.1.15 SupplementGap Property

Gets or sets the distance, in centimeters, between the normal barcode and the supplement section.

[Visual Basic .NET]

```
Public Property SupplementGap As float
```

[C#]

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

Remarks

The default value is 0.5, internally our barcode .NET control converts this property value from centimeters to pixels based on device resolution, so different SupplementGap property values might end up with same supplement gap in pixels due to performing rounding operations.

See Also

[BarHeight Property](#) | [NarrowBarWidth Property](#)

4.1.16 SupplementType Property

Gets or sets the supplement type.

[Visual Basic .NET]

```
Public Property SupplementType As enumSupplementType
```

[C#]

```
public enumSupplementType SupplementType {get; set;}
```

4.1.17 SymbologyType Property

Gets or sets the barcode type.

[Visual Basic .NET]

```
Public Property SymbologyType As enumSymbologyType
```

[C#]

```
public enumSymbologyType SymbologyType {get; set;}
```

4.1.18 TextFont Property

Gets or sets the font of human readable text in the barcode.

[Visual Basic .NET]

```
Public Property TextFont As Font
```

[C#]

```
public Font TextFont {get; set;}
```

4.1.19 UPCESystem Property

Gets or sets the encoding system of UPCE barcode.

[Visual Basic .NET]

```
Public Property UPCESystem As enumUPCESystem
```

[C#]

```
public enumUPCESystem UPCESystem {get; set;}
```

4.1.20 Wide2NarrowRatio Property

Gets or sets the ratio of the wide bar to the narrow bar.

[Visual Basic .NET]

```
Public Property Wide2NarrowRatio As float
```

[C#]

```
public float Wide2NarrowRatio {get; set;}
```

Remarks

The default value is 2.0, typically this property value is between 2 and 3.

4.2 Methods

4.2.1 CodeOne Method

Specifies the version of 2D Code One barcode.

[Visual Basic .NET]

```
Public Sub CodeOne(ByVal COVersion As Integer)
```

[C#]

```
public void CodeOne(short COVersion);
```

Parameters

COVersion

Specifies the version of Code One, this parameter can be one of the following values:

Value	Size
1	16 X 18
2	22 X 22

3	28 X 32
4	40 X 42
5	52 X 54
6	70 X 76
7	104 X 98
8	148 X 134
9	8 X varied width
10	16 X varied width

4.2.2 GetActualSize Method

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

ActualHeight

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

4.2.3 MicroPDF417 Method

Specifies the number of columns for 2D Micro PDF417 barcode.

[Visual Basic .NET]

```
Public Sub MicroPDF417(ByVal Columns As Integer)
```

[C#]

```
public void MicroPDF417(short Columns);
```

Parameters*Columns*

Specifies the number of columns, this parameter can be one of the following values:

Value	Description
1	1 column
2	2 columns
3	3 columns
4	4 columns

4.2.4 MicroQRCode Method

Specifies the version and error correction level for 2D Micro QRCode barcode.

[Visual Basic .NET]

```
Public Sub MicroQRCode(ByVal MQVersion As Integer, ByVal MQLevel As Integer)
```

[C#]

```
public void MicroQRcode(short MQVersion, short MQLevel);
```

Parameters*MQVersion*

Specifies the version of Micro QRCode, this parameter can be one of the following values:

Value	Description
1	Version M1 with the size 11 X 11
2	Version M2 with the size 13 X 13
3	Version M3 with the size 15 X 15
4	Version M4 with the size 17 X 17

MQLevel

Specifies the error correction level of Micro QRCode, this parameter can be one of the following values:

Value	Description
1	L (applicable to version M2, M3 and M4)
2	M (applicable to version M2, M3 and M4)
3	Q (applicable to version M4 only)

4.2.5 Render Method

Renders the 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 barcode.

4.2.6 SaveAsImage Method

Exports the 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 the barcode image.

ImgFormat

Specifies the image format.

Remarks

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

See Also

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

4.2.7 SaveAsMemory Method

Exports the 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 barcode image.

ImgFormat

Specifies the image format.

Remarks

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

See Also

`GetActualSize()` Method | `SetSize()` Method

4.2.8 SetSize Method

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

See Also

`GetActualSize()` Method

4.3 Enumerations

4.3.1 BearerBarType Enumeration

An enumeration type for all possible bearer bar options.

Members

Name	Comment
bbNone	No Bearer Bar
bbHoriBars	Horizontal Bars
bbBox	Bearer Bar Box around Barcode

4.3.2 CodaBarChar Enumeration

An enumeration type for all possible CodeBar start/end character values.

Members

Name	Comment
cbc_A	Start/End Character 'A'
cbc_B	Start/End Character 'B'
cbc_C	Start/End Character 'C'
cbc_D	Start/End Character 'D'

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

4.3.4 SupplementType Enumeration

An enumeration type for all possible supplement type values.

Members

Name	Comment
spNone	No Supplement
spSupplement2	Supplement 2
spSupplement5	Supplement 5

4.3.5 SymbologyType Enumeration

An enumeration type for all possible symbology type values.

Value	Barcode Description	Allow Bearer Bars?	Allow Supplement 2 or 5?	Sample Barcode String
1D Barcodes				
1	Channel Code			
2	Codabar			
3	Code 11			
4	Code 128			1234ABCD+ /
5	Code 128 (Set A)			
6	Code 128 (Set B)			
7	Code 128 (Set C)			
8	Code 32 or Italian Pharmacode			
9	Code 39			1234ABCD
10	Code 39 Extended			
11	Code 93			
12	Data Logic 2/5	Yes		
13	EAN128/UCC (GS1-128)			(21)95FNC1(11)090101
14	EAN 13		Yes	123456789012
15	EAN 8		Yes	1234567
16	EAN Velocity		Yes	
17	Flattermarken			
18	GS1 Databar-14			1234567890123
19	GS1 DataBar Expanded			
20	GS1 DataBar Expanded Stacked			
21	GS1 Databar Limited			
22	GS1 Databar Stacked			
23	GS1 DataBar Stacked Omnidirectional			
24	GS1 Databar Truncated			
25	GS1 Databar-14 Composite			
26	GS1 DataBar Expanded Composite			(01)1234567890123
27	GS1 DataBar Expanded Stacked Composite			
28	GS1 Databar Limited Composite			
29	GS1 Databar Stacked Composite			
30	GS1 DataBar Stacked Omnidirectional Composite			
31	HIBC Code 128 for LIC or PAS			+H123ABC0123456789 0D
32	HIBC Code 39 for LIC or PAS			+ /EAH783B
33	HIBC CodaBlock-F for LIC or PAS			+ /EAH783/Z34H159\$
34	HIBC Micro PDF417 for LIC or PAS			
35	IATA 2 of 5 Barcode	Yes		
36	Industrial 2 of 5 Barcode	Yes		

37	Interleaved 2 of 5 Barcode	Yes		
38	ISBN or International Standard Book Number		Yes	3161484100
39	ISMN or International Standard Music Number		Yes	M-2306-7118-7
40	ISSN or International Standard Serial Number		Yes	0264-3596
41	ITF-14 or UPC Shipping Container Symbol	Yes		
42	JAN 13		Yes	
43	JAN 8		Yes	
44	Logmars			
45	Matrix 2 of 5 Barcode	Yes		
46	MSI/Plessey			
47	Numly Number or ESN			1234567890123456789
48	Optical Product Code		Yes	123456789
49	Pharmacode One-Track			
50	Pharmacode Two-Track			
51	Pharma-Zentral-Nummer			123456
52	SCC-14 or Shipping Container Code			
53	SSCC-18 or UPC-128 Shipping Container Code			
54	Telepen Alpha			
55	Telepen Numeric			
56	UK Plessey			
57	UPC-A		Yes	1234567890
58	UPC-E		Yes	1234567
59	VICS BOL or VICS Bill of Lading			
60	VICS SCAC PRO			
Postal Code Barcodes				
61	Australia Postal Standard Customer			
62	Australia Postal Redirection			
63	Australia Postal Reply Paid			
64	Australia Postal Routing			
65	China Postal Code			
66	Danish Postal Code			CC12345678
67	Deutsche Post Identcode			12345678901
68	Deutsche Post Leitcode			1234567890123
69	France Postal Code 39			RA12345678
70	Italy Postal Code 2/5			
71	Italy Postal Code 39			
72	Japan Postal Code			1234567AZ
73	KIX or Netherlands Postal Code			A12345678
74	Korean Postal Code			123456
75	Royal Mail 4 State			
76	Singapore Postal Code			

77	Swiss Parcel Post Barcode			
78	USPS DAFT Code			DAFTTFAD
79	USPS Facing Identification Mark			A
80	USPS Horizontal Bars			
81	USPS OneCode or USPS Intelligent Mail			12345678901234567890 +50309
82	USPS PLANET			
83	USPS POSTNET			
84	USPS Sack Label			50309123
85	USPS Tray Label			5030912345
2D Barcodes				
86	Codablock-F			
87	Code 16K			
88	Code 49			
89	Code One			
90	Micro PDF417			
91	Micro QRCode			

4.3.6 UPCESystem Enumeration

An enumeration type for all possible UPCE system values.

Members

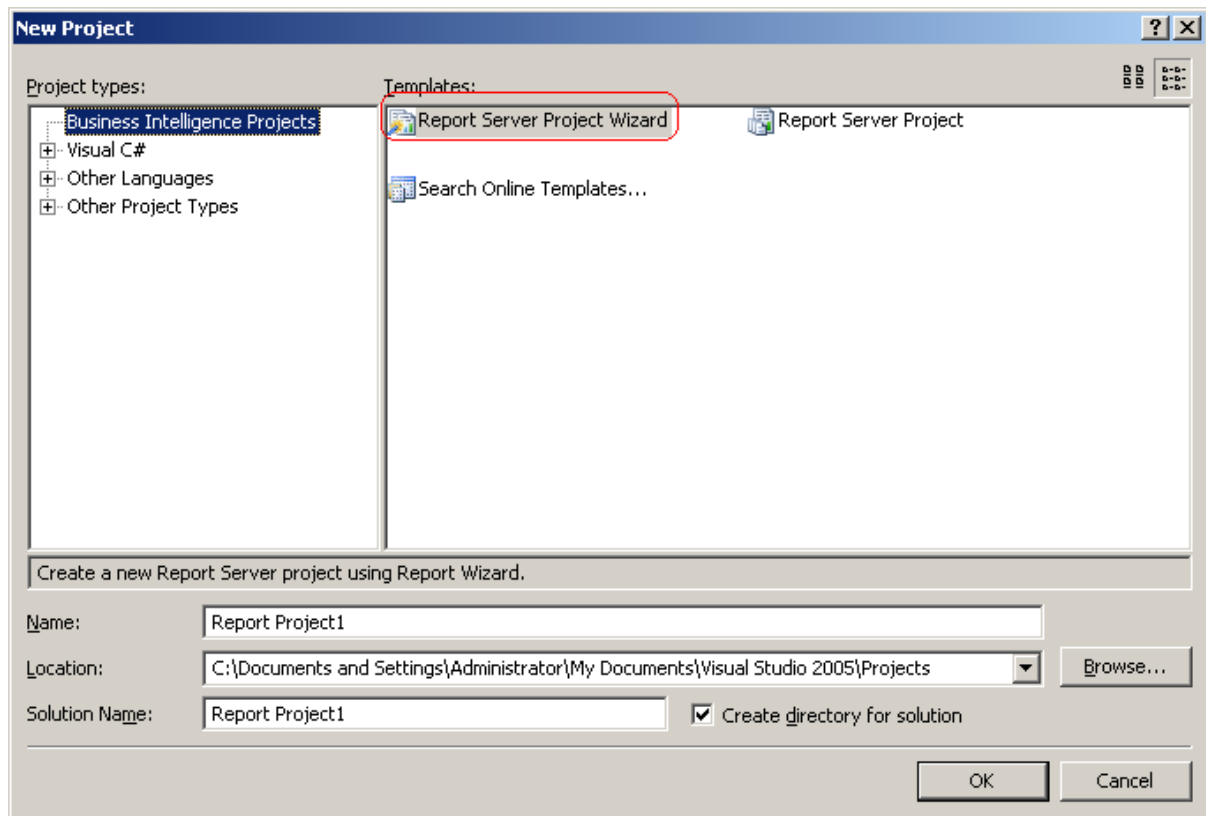
Name	Comment
ustSystem0	System 0
ustSystem1	System 1

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

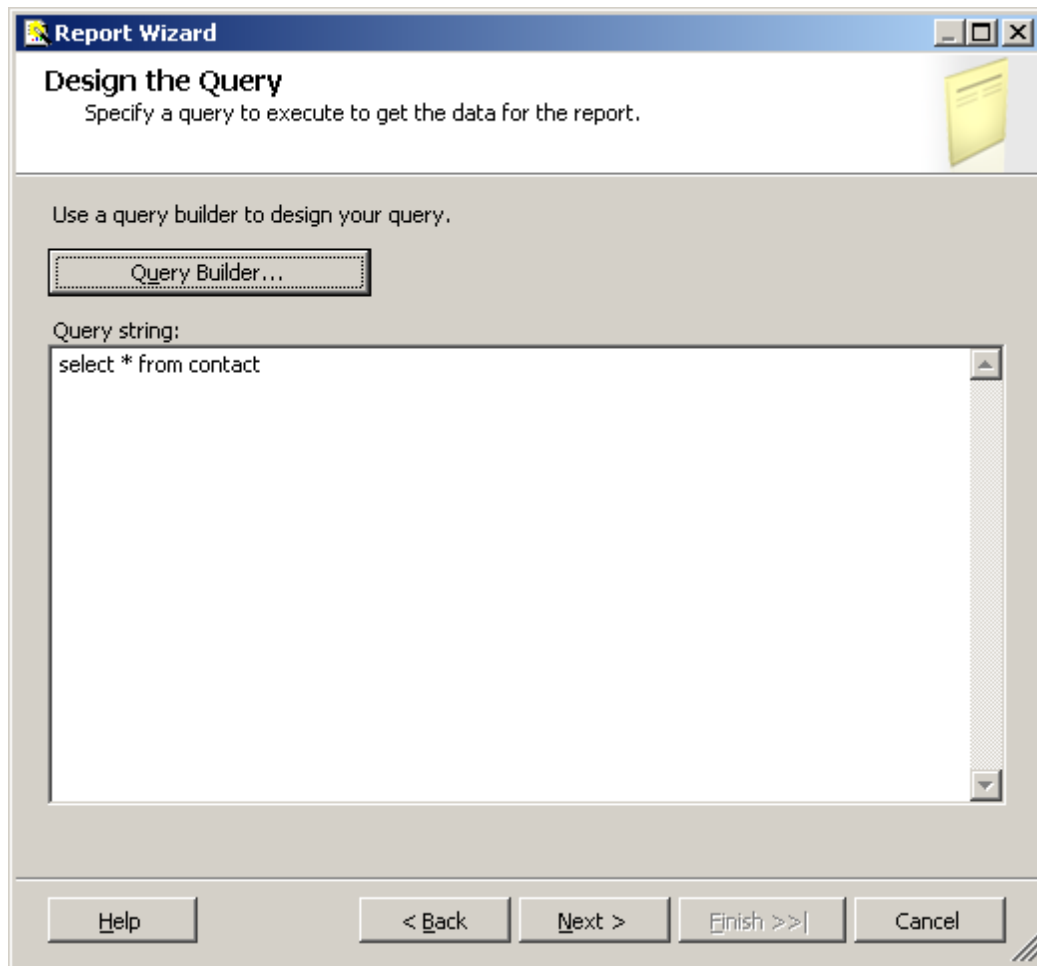
Select or enter a database name:
Customers []
 Attach a database file:
[] [Browse...]
Logical name: []

[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:' field containing 'DataSource1', a 'Type:' dropdown menu set to 'Microsoft SQL Server', and a 'Connection string:' text area containing 'Data Source=PC-V4OG4L4T6\SQLEXPRESS_NEW;Initial Catalog=Customers'. To the right of the connection string are 'Edit...' and 'Credentials...' buttons. 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

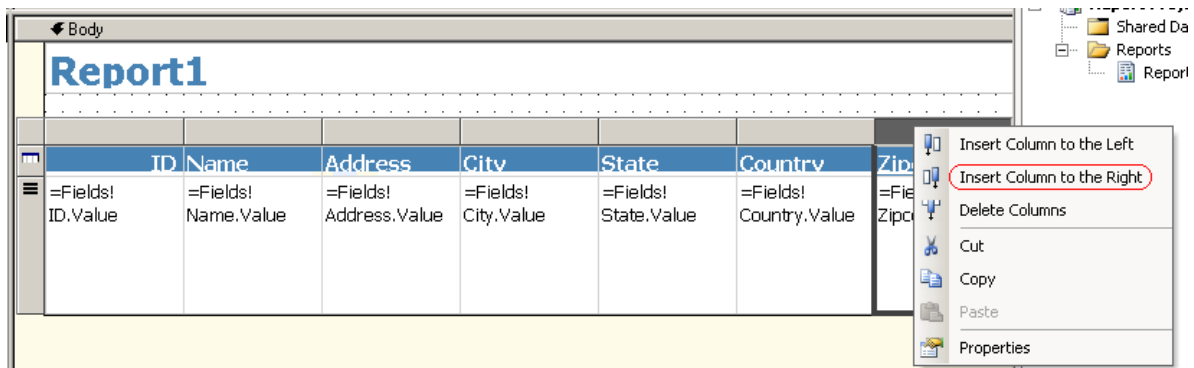


4. In the following dialogs, choose the appropriate options for the type of the report, the way of how to group the data in the table and the table style, then click the "**Finish**" button.

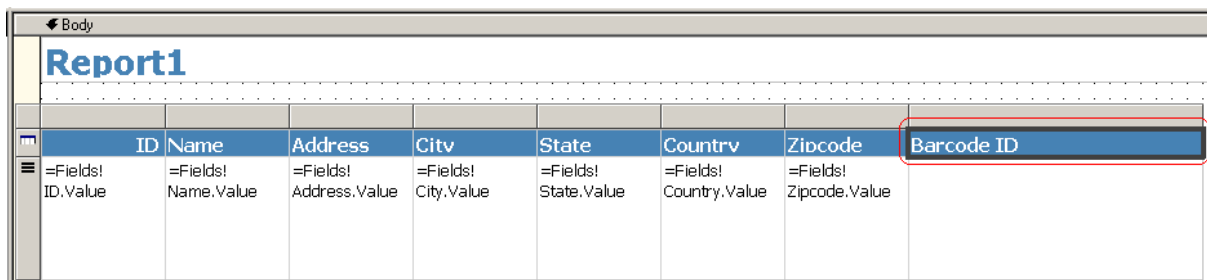
5.2 Add Barcodes


Follow the instructions to add the barcodes to the report.

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

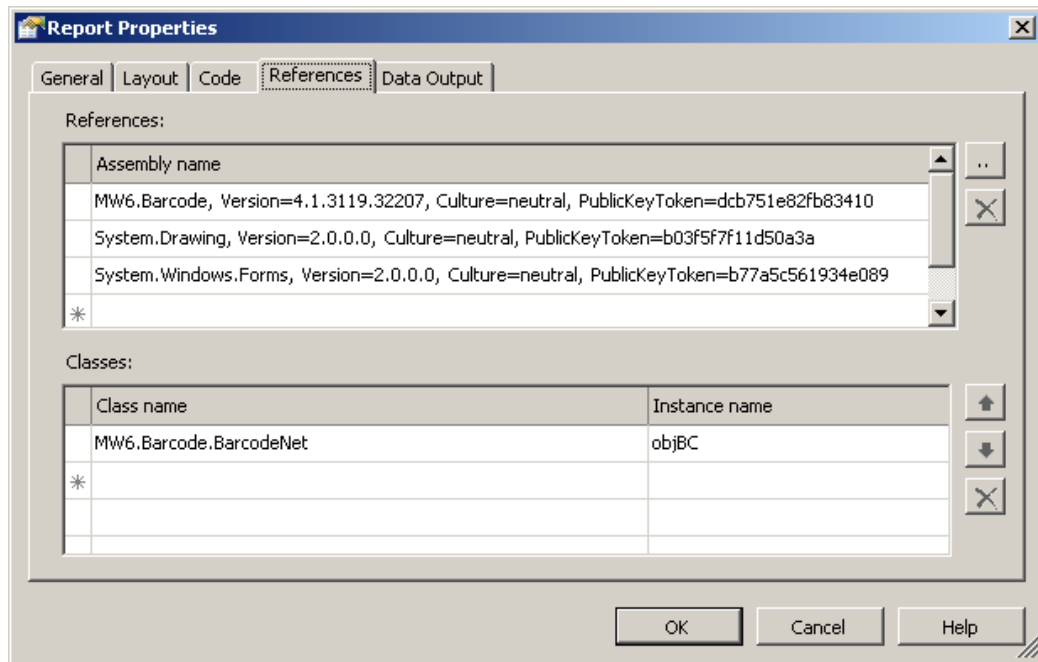


3. Change the column title to the "**Barcode ID**".



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.Barcode.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.Barcode.BarcodeNet" in the "**Class name**" box, enter "objBC" in the "**Instance name**" box to create an assembly object to use in the code to retrieve the 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 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

    objBC.Data = DataStr
    objBC.CheckDigit = True
    objBC.CheckDigitToText = True
    objBC.ShowText = True
    objBC.Size = New System.Drawing.Size(408, 104)
    objBC.SupplementGap = 0.5
    objBC.SymbologyType = MW6.Barcode.enumSymbologyType.Code_128
    objBC.TextFont = New System.Drawing.Font("Arial", 12.0!)
    objBC.UPCESystem = MW6.Barcode.enumUPCESystem.ustSystem0
    objBC.Wide2NarrowRatio = 2.0

    ' Get the actual barcode width and height (in pixels)
    ' Render the barcode on the computer screen
    objBC.GetActualSize(True, Nothing, ActualWidth, ActualHeight)

    ExtraWidth = 20
    ExtraHeight = 60

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

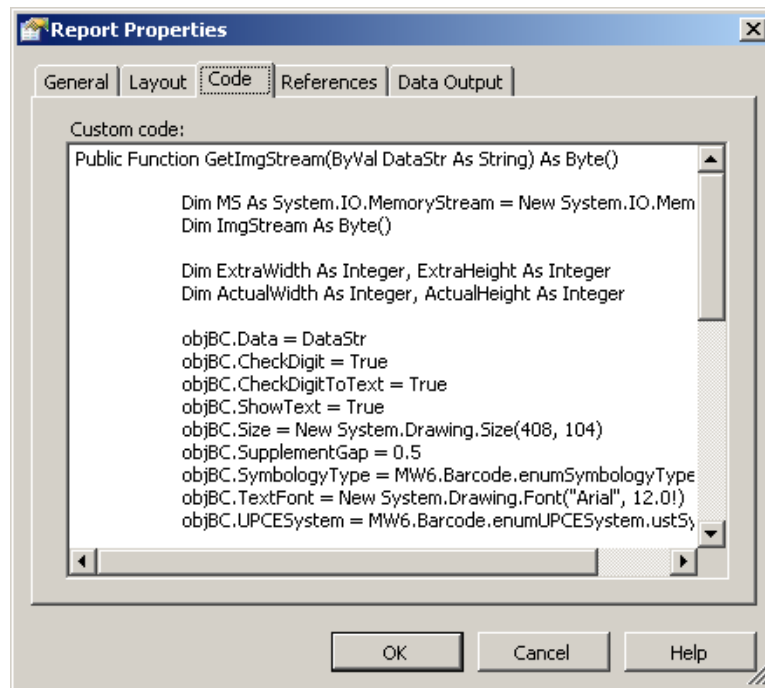
    ' Save the barcode image in the memory using the Jpeg format
    objBC.SaveAsMemory(MS, System.Drawing.Imaging.ImageFormat.Jpeg)

    ImgStream = MS.ToArray

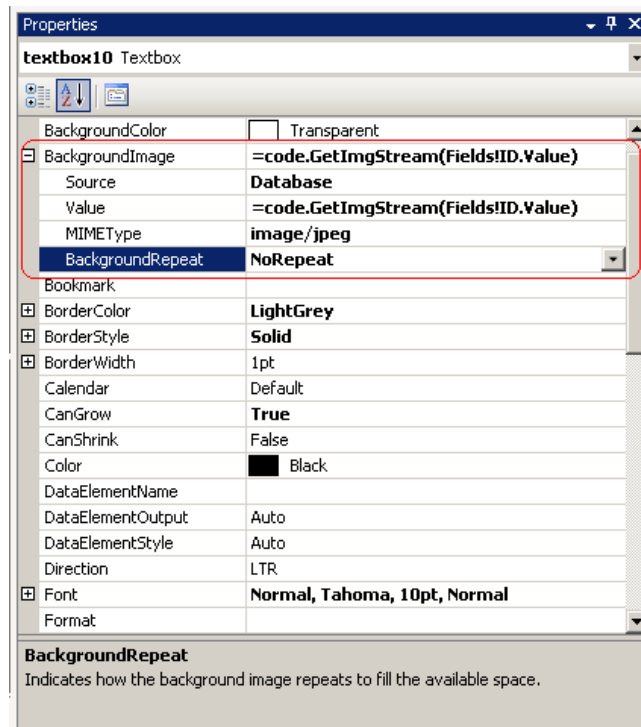
    MS.Close()

    Return ImgStream

End Function
```





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

Report1

ID	Name	Address	City	State	Country	Zipcode	Barcode ID
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.Barcode.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.
