

Table of Contents

Foreword	0
Part I Introduction	3
Part II Installation	3
1 Trial Version.....	3
2 Full Version.....	4
Part III How to Distribute It	5
Part IV Reference Guide	5
1 Properties.....	5
BackColor Property	5
BarColor Property	5
Data Property	5
Level Property	6
Mask Property	6
ModuleSize Property	6
Orientation Property	7
Version Property	7
2 Methods.....	7
GetActualRC Method	7
GetActualSize Method	8
GetPatternData Method	9
Render Method	10
SaveAsImage Method	10
SaveAsMemory Method	11
SetSize Method	11
3 Enumerations.....	12
Level Enumeration	12
Mask Enumeration	12
Orientation Enumeration	13
Version Enumeration	13
Part V How to Use It in Reporting Services	14
1 Create a Report Project.....	14
2 Add QRCode Barcodes.....	17
Part VI Data Capacity Tables	23
1 Level L.....	23
2 Level M.....	23
3 Level Q.....	24
4 Level H.....	25

Part VII License	26
Index	0

1 Introduction

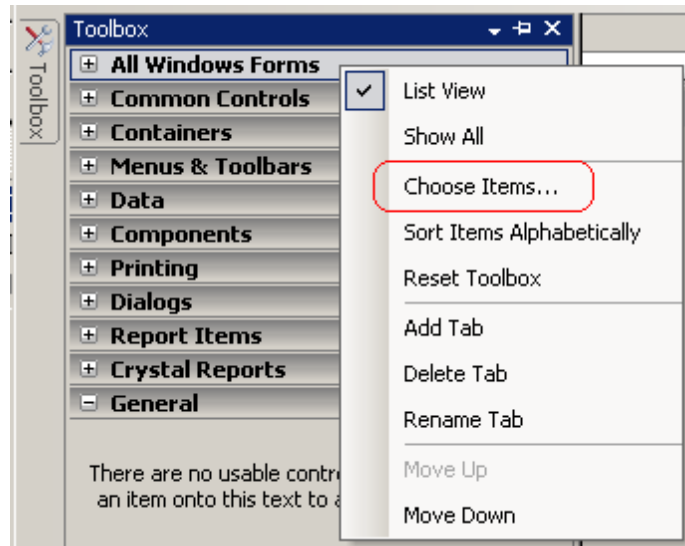
MW6 QRCode .NET control is a flexible and reliable .NET component and can create professional 2D QRCode images for your .NET application, you can save the QRCode as different image format files. It is easy to print the QRCode barcode using the PrintDocument Control provided by the .NET Framework.

QRCode is designed to pack a lot of information in a very small space, our QRCode .NET control supports Model 2, it is capable of encoding up to 2953 bytes, 4296 alphanumeric characters, or 7089 numeric digits.

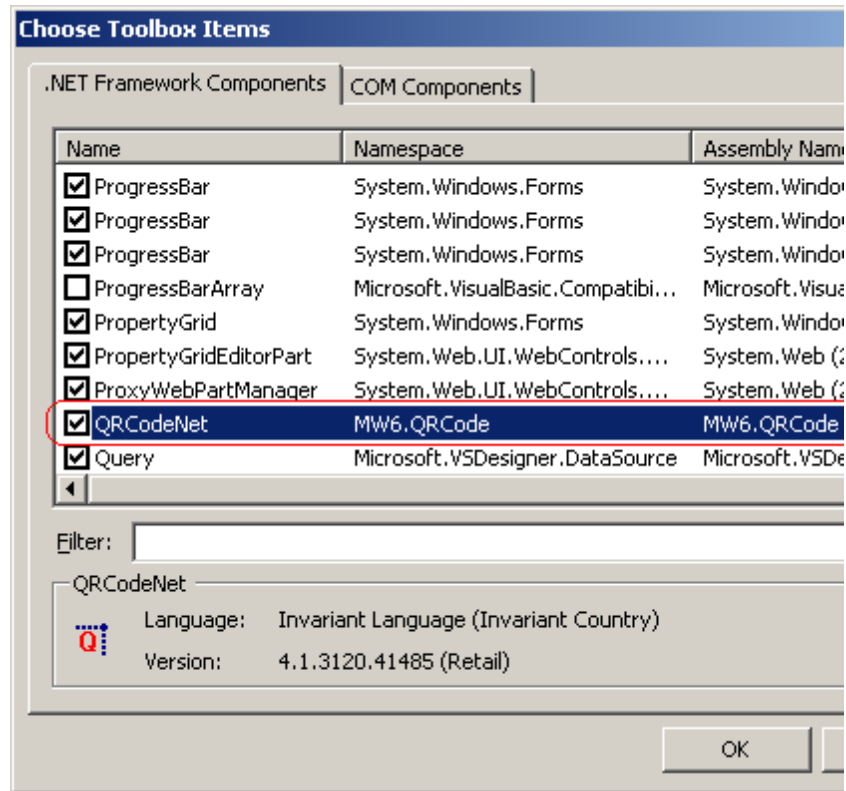
2 Installation

2.1 Trial Version

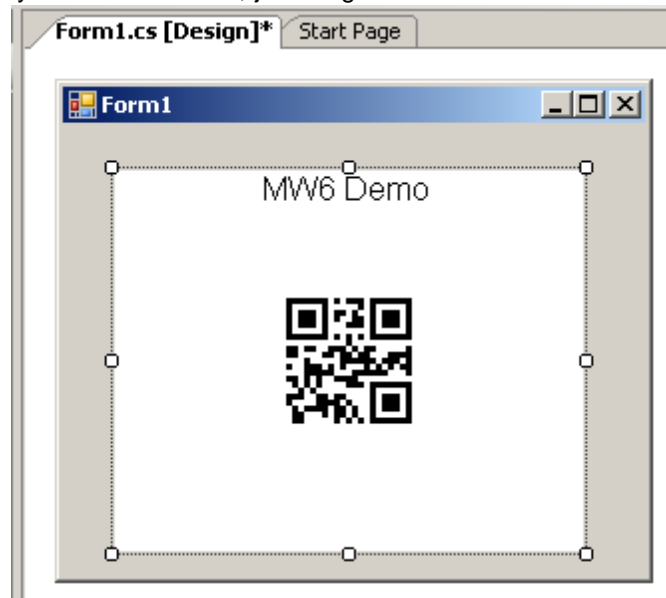
1. The trial version QRCode .NET control appends "MW6 Demo" to the string encoded with the QRCode format.
2. After the installation for the trial version is finished, copy MW6.QRCode.dll in VB.NET or C# demo project sub folder in the destination folder (e.g. "C:\Program Files\MW6 .NET Components\QRCode")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

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

3 How to Distribute It

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

Gets or sets the level of error correction allowing recovery.

```
[Visual Basic .NET]  
Public Property Level As enumLevel
```

```
[C#]  
public enumLevel Level {get; set;}
```

4.1.5 Mask Property

Gets or sets the mask pattern for improving the readability.

```
[Visual Basic .NET]  
Public Property Mask As enumMask
```

```
[C#]  
public enumMask Mask {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 QRCode .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 QRCode barcode.

[Visual Basic .NET]

```
Public Property Orientation As enumOrientation
```

[C#]

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

4.1.8 Version Property

Gets or sets the version of the QRCode barcode.

[Visual Basic .NET]

```
Public Property Version As enumVersion
```

[C#]

```
public enumVersion Version {get; set;}
```

Remarks

If you set `Version` to `vrAuto` (Auto version), our QRCode .NET control will automatically choose an appropriate version with enough data capacity to encode the string.

If you set `Version` to other values and the data capacity of the selected version is not big enough to encode the string, our QRCode .NET control will also automatically choose an appropriate version 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 QRCode 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 QRCode barcode.

ActualCols

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

Remarks

If you set *Version* to *vrAuto* (Auto version), QRCode .NET control will automatically choose an appropriate version 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 *Version* to other values and the data capacity of the selected version is not big enough to encode the string, QRCode .NET control will also automatically choose an appropriate version 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 *Version* property.

4.2.2 GetActualSize Method

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

ActualHeight

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

4.2.3 GetPatternData Method

Gets the QRCode 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 QRCode 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 QRCode 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 QRCode barcode pattern matrix data and render the QRCode barcode onto any device such as the printer, only *Data*, *Level*, *Mask* and *Version* properties affect the pattern matrix data output.

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

4.2.4 Render Method

Renders the QRCode 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 QRCode barcode.

4.2.5 SaveAsImage Method

Exports the QRCode 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 QRCode barcode image.

ImgFormat

Specifies the image format.

Remarks

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

See Also

GetActualSize() Method | SetSize() Method

4.2.6 SaveAsMemory Method

Exports the QRCode 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 QRCode barcode image.

ImgFormat

Specifies the image format.

Remarks

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

See Also

GetActualSize() Method | SetSize() Method

4.2.7 SetSize Method

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

See Also

`GetActualSize()` Method

4.3 Enumerations

4.3.1 Level Enumeration

An enumeration type for all possible level values.

Members

Name	Comment
lvL	Level L
lvM	Level M
lvQ	Level Q
lvH	Level H

4.3.2 Mask Enumeration

An enumeration type for all possible mask values.

Members

Name	Comment
mkAuto	Auto Mask
mk0	Mask 0
mk1	Mask 1
mk2	Mask 2
mk3	Mask 3
mk4	Mask 4
mk5	Mask 5
mk6	Mask 6
mk7	Mask 7

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 Version Enumeration

An enumeration type for all possible version values.

Members

Name	Comment
vrAuto	Auto
vr1	21 X 21
vr2	25 X 25
vr3	29 X 29
vr4	33 X 33
vr5	37 X 37
vr6	41 X 41
vr7	45 X 45
vr8	49 X 49
vr9	53 X 53
vr10	57 X 57
vr11	61 X 61
vr12	65 X 65
vr13	69 X 69
vr14	73 X 73
vr15	77 X 77
vr16	81 X 81
vr17	85 X 85
vr18	89 X 89
vr19	93 X 93
vr20	97 X 97
vr21	101 X 101
vr22	105 X 105
vr23	109 X 109
vr24	113 X 113
vr25	117 X 117
vr26	121 X 121
vr27	125 X 125
vr28	129 X 129
vr29	133 X 133
vr30	137 X 137

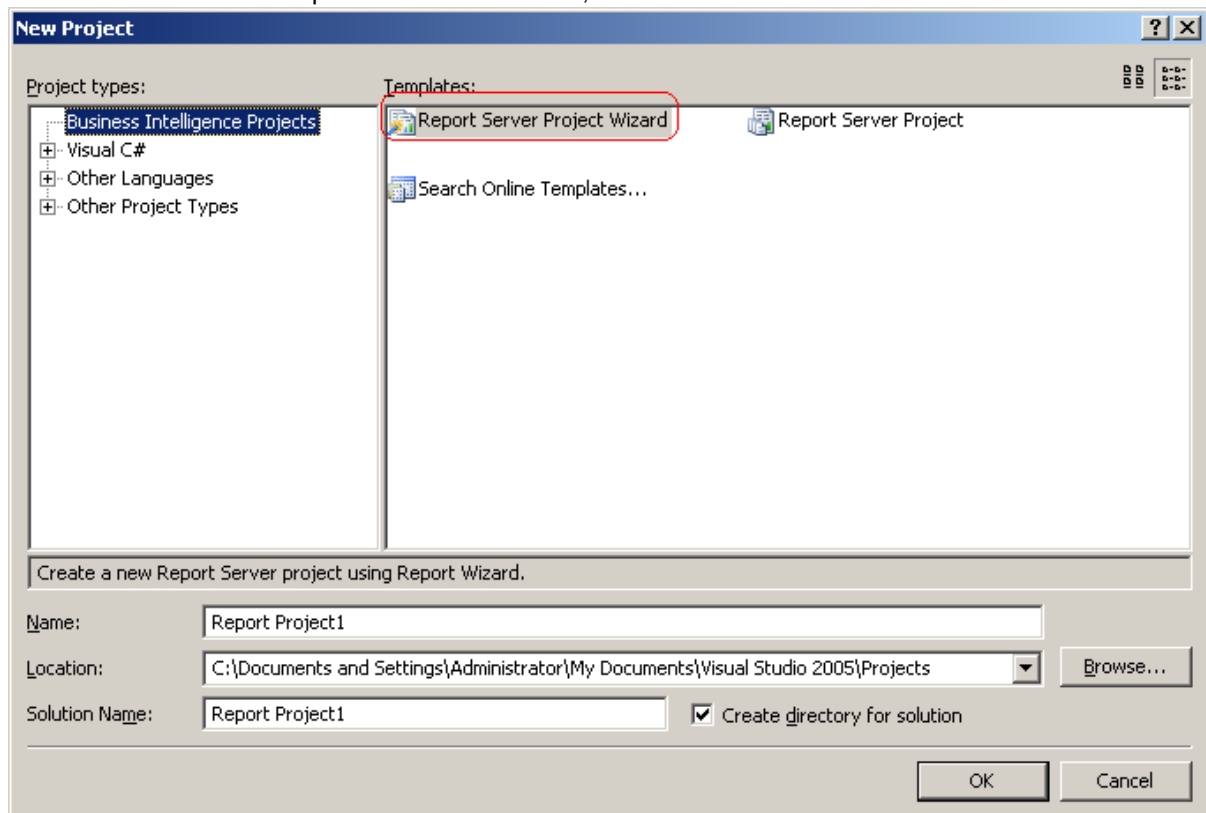
vr31	141 X 141
vr32	145 X 145
vr33	149 X 149
vr34	153 X 153
vr35	157 X 157
vr36	161 X 161
vr37	165 X 165
vr38	169 X 169
vr39	173 X 173
vr40	177 X 177

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

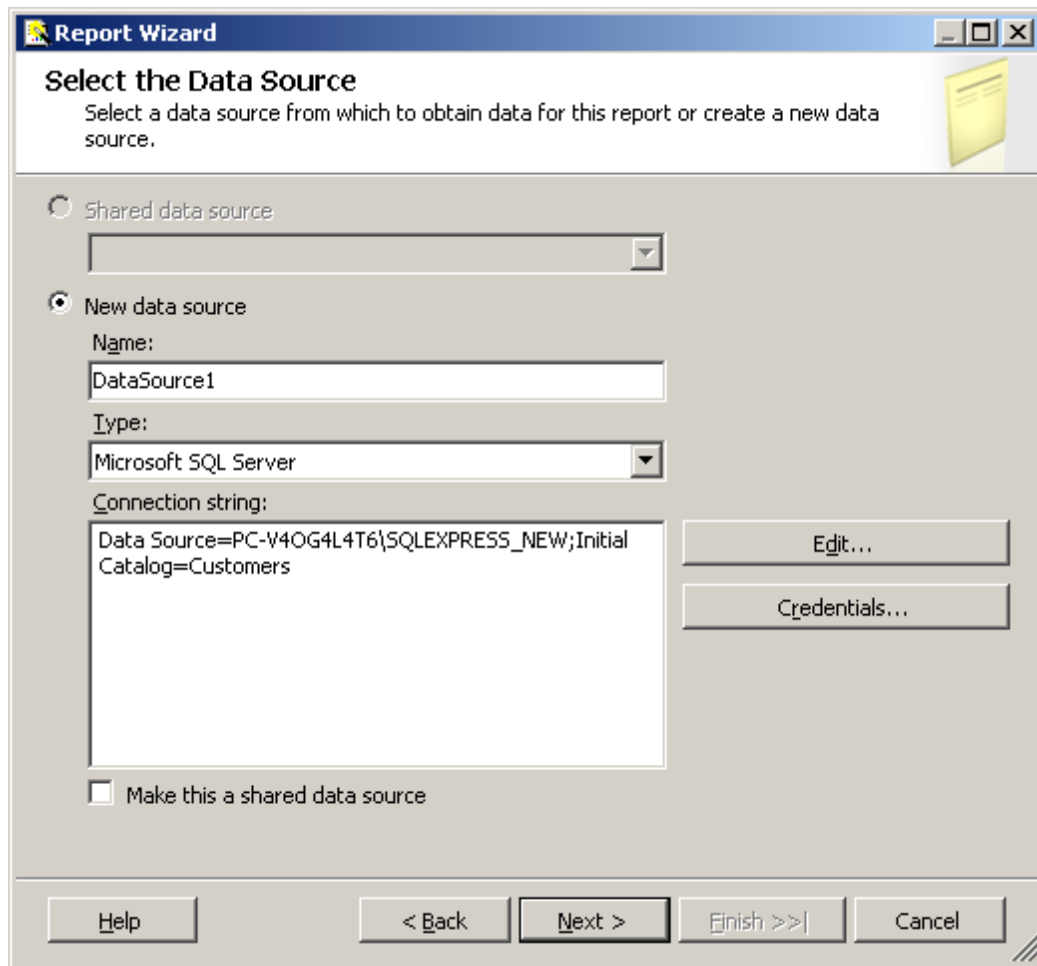
Connect to a database

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

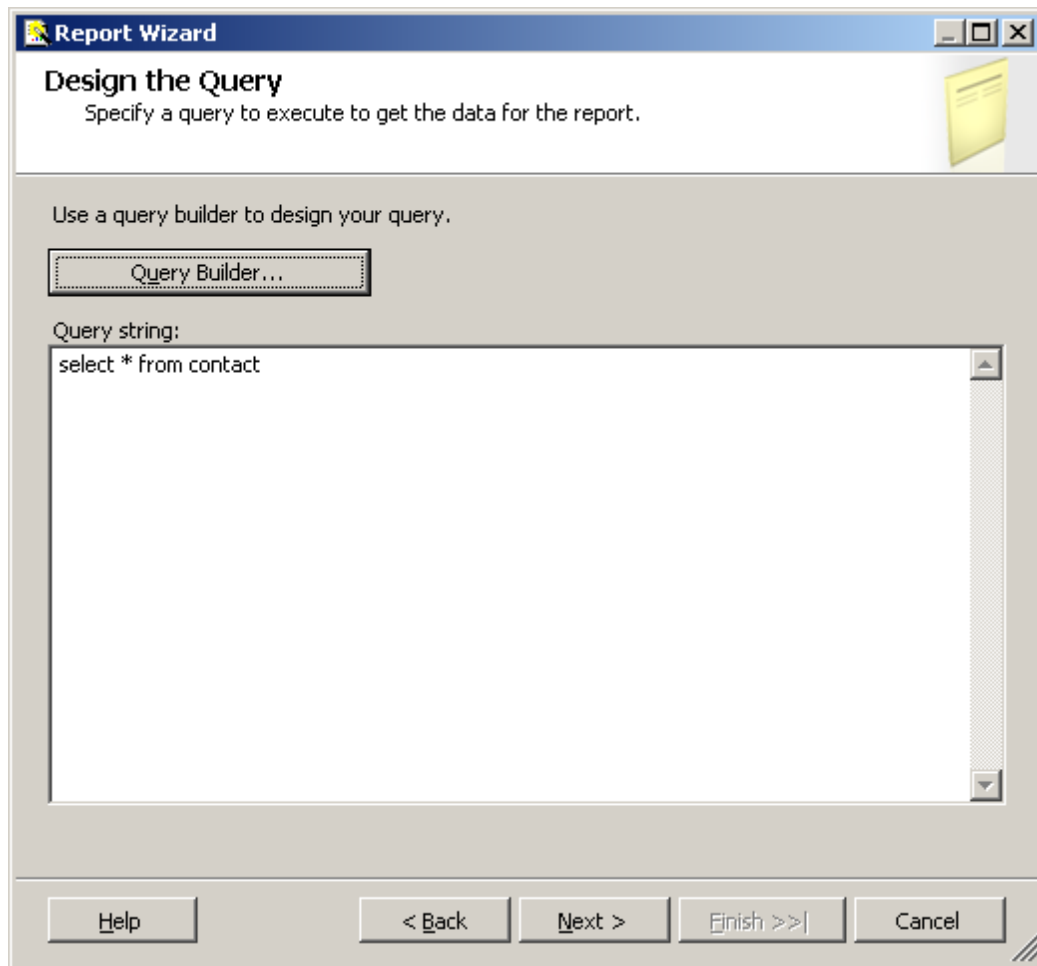
Logical name: []

[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

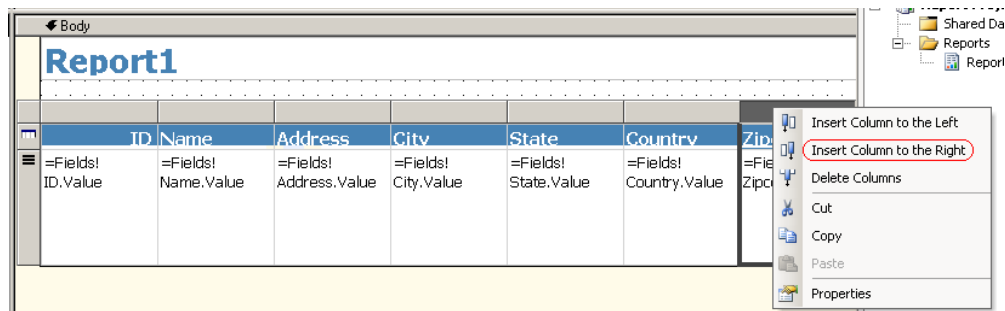


4. In the following dialogs, choose the appropriate options for the type of the report, the way of how to group the data in the table and the table style, then click the "**Finish**" button.

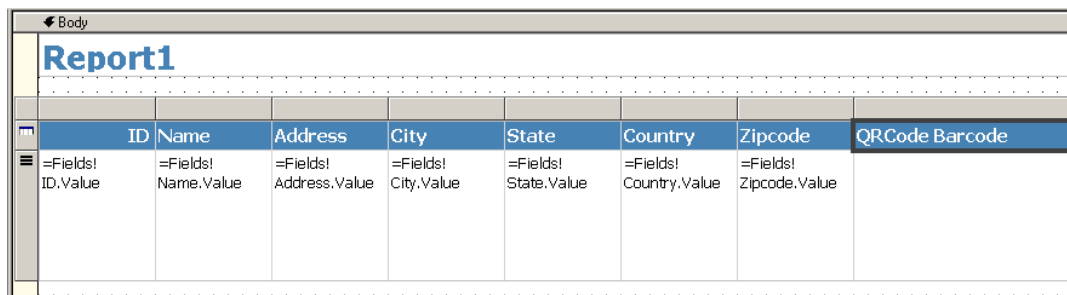
5.2 Add QRCode Barcodes


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

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

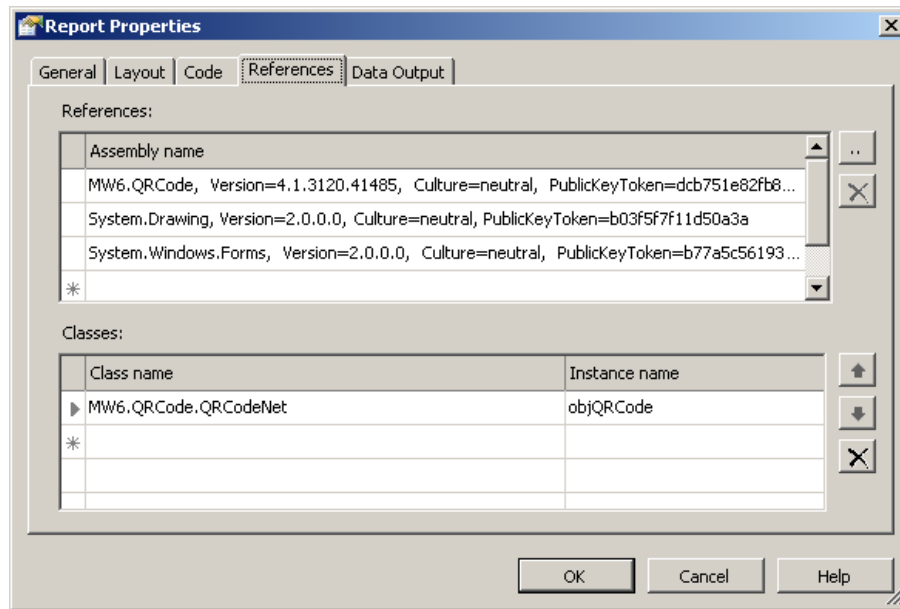


3. Change the column title to the "QRCode Barcode".



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.QRCode.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.QRCode.QRCodeNet" in the "Class name" box, enter "objQRCode" in the "Instance name" box to create an assembly object to use in the code to retrieve the QRCode 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 QRCode 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

    objQRCode.Data = DataStr

    ' Module Size
    objQRCode.ModuleSize = 0.07

    ' Version
    objQRCode.Version = 0

    ' Level
    objQRCode.Level = 0

    ' Mask
    objQRCode.Mask = 1

    'Orientation
    objQRCode.Orientation = 0

    ExtraWidth = 60
    ExtraHeight = 70

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

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

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

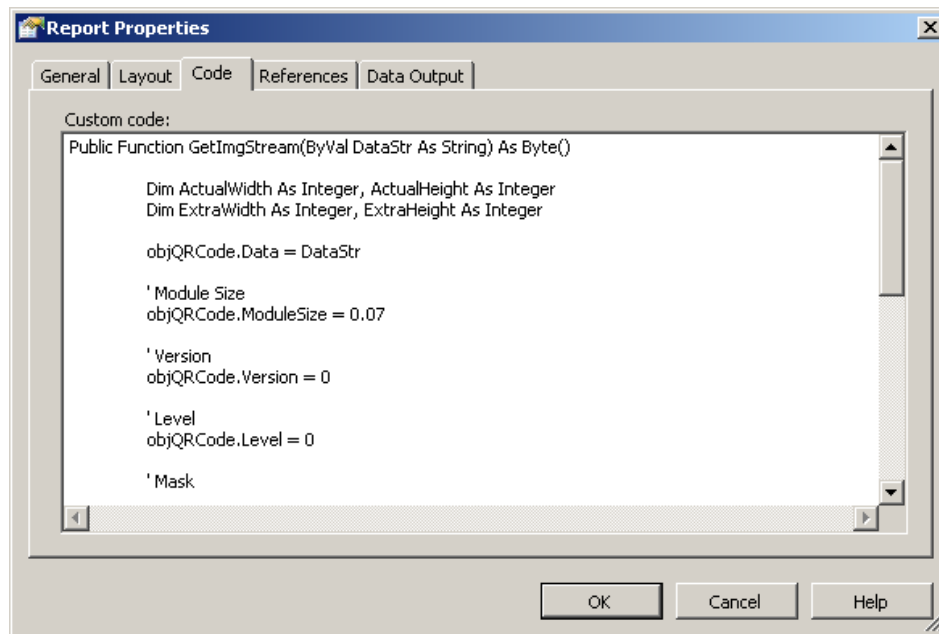
    objQRCode.SaveAsMemory(MS, System.Drawing.Imaging.ImageFormat.Jpeg)

    ImgStream = MS.ToArray

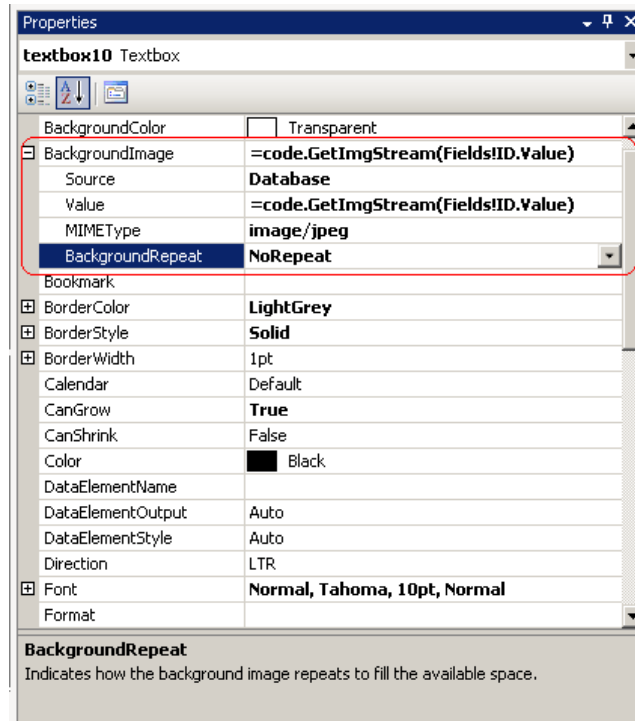
    MS.Close()

    Return ImgStream

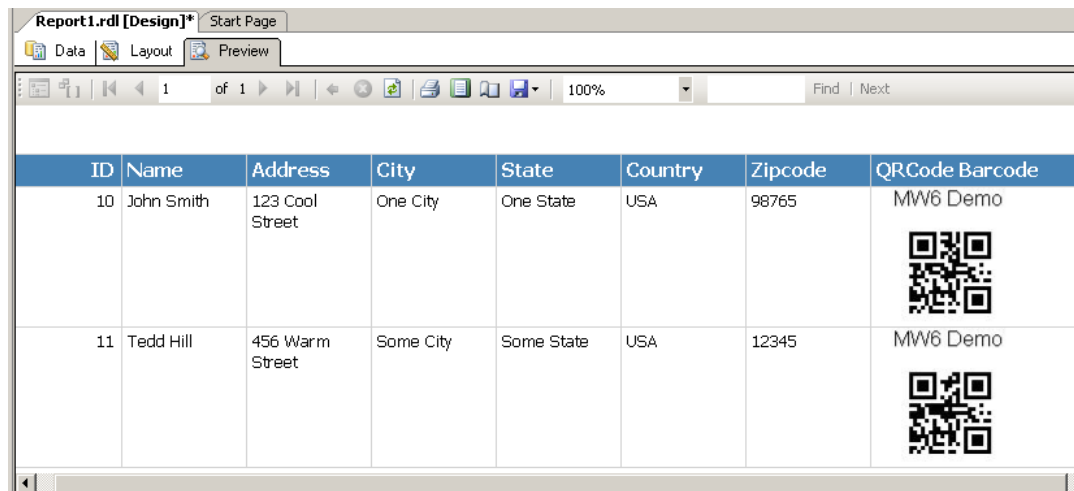
End Function
```



6. Change the "**BackgroundImage**" property of the text box in the "**QRCode Barcode**" column to display the QRCode 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 QRCode 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 QRCode 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 QRCode 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.QRCode.dll assembly to run properly.

6 Data Capacity Tables

6.1 Level L

Version	Capacity (in digits)	Capacity (in alphanumeric characters)	Capacity (in bytes)
1	41	25	17
2	77	47	32
3	127	77	53
4	187	114	78
5	255	154	106
6	322	195	134
7	370	224	154
8	461	279	192
9	552	335	230
10	652	395	271
11	772	468	321
12	883	535	367
13	1022	619	425
14	1101	667	458
15	1250	758	520
16	1408	854	586
17	1548	938	644
18	1725	1046	718
19	1903	1153	792
20	2061	1249	858
21	2232	1352	929
22	2409	1460	1003
23	2620	1588	1091
24	2812	1704	1171
25	3057	1853	1273
26	3283	1990	1367
27	3517	2132	1465
28	3669	2223	1528
29	3909	2369	1628
30	4158	2520	1732
31	4417	2677	1840
32	4686	2840	1952
33	4965	3009	2068
34	5253	3183	2188
35	5529	3351	2303
36	5836	3537	2431
37	6153	3729	2563
38	6479	3927	2699
39	6743	4087	2809
40	7089	4296	2953

6.2 Level M

Version	Capacity (in digits)	Capacity (in alphanumeric)	Capacity (in bytes)
---------	----------------------	----------------------------	---------------------

		characters)	
1	34	20	14
2	63	38	26
3	101	61	42
4	149	90	62
5	202	122	84
6	255	154	106
7	293	178	122
8	365	221	152
9	432	262	180
10	513	311	213
11	604	366	251
12	691	419	287
13	796	483	331
14	871	528	362
15	991	600	412
16	1082	656	450
17	1212	734	504
18	1346	816	560
19	1500	909	624
20	1600	970	666
21	1708	1035	711
22	1872	1134	779
23	2059	1248	857
24	2188	1326	911
25	2395	1451	997
26	2544	1542	1059
27	2701	1637	1125
28	2857	1732	1190
29	3035	1839	1264
30	3289	1994	1370
31	3486	2113	1452
32	3693	2238	1538
33	3909	2369	1628
34	4134	2506	1722
35	4343	2632	1809
36	4588	2780	1911
37	4775	2894	1989
38	5039	3054	2099
39	5313	3220	2213
40	5596	3391	2331

6.3 Level Q

Version	Capacity (in digits)	Capacity (in alphanumeric characters)	Capacity (in bytes)
1	27	16	11
2	48	29	20
3	77	47	32
4	111	67	46
5	144	87	60
6	178	108	74

7	207	125	86
8	259	157	108
9	312	189	130
10	364	221	151
11	427	259	177
12	489	296	203
13	580	352	241
14	621	376	258
15	703	426	292
16	775	470	322
17	876	531	364
18	948	574	394
19	1063	644	442
20	1159	702	482
21	1224	742	509
22	1358	823	565
23	1468	890	611
24	1588	963	661
25	1718	1041	715
26	1804	1094	751
27	1933	1172	805
28	2085	1263	868
29	2181	1322	908
30	2358	1429	982
31	2473	1499	1030
32	2670	1618	1112
33	2805	1700	1168
34	2949	1787	1228
35	3081	1867	1283
36	3244	1966	1351
37	3417	2071	1423
38	3599	2181	1499
39	3791	2298	1597
40	3993	2420	1663

6.4 Level H

Version	Capacity (in digits)	Capacity (in alphanumeric characters)	Capacity (in bytes)
1	17	10	7
2	34	20	14
3	58	35	24
4	82	50	34
5	106	64	44
6	139	84	58
7	154	93	64
8	202	122	84
9	235	143	98
10	288	174	119
11	331	200	137
12	374	227	155
13	427	259	177

14	468	283	194
15	530	321	220
16	602	365	250
17	674	408	280
18	746	452	310
19	813	493	338
20	919	557	382
21	969	587	403
22	1056	640	439
23	1108	672	461
24	1228	744	511
25	1286	779	535
26	1425	864	593
27	1501	910	625
28	1581	958	658
29	1677	1016	698
30	1782	1080	742
31	1897	1150	790
32	2022	1226	842
33	2157	1307	898
34	2301	1394	958
35	2361	1431	983
36	2524	1530	1051
37	2625	1591	1093
38	2735	1658	1139
39	2927	1774	1219
40	3057	1852	1273

7 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.
