# Table of Contents

# 1 Introduction

MW6 lightweight QRCode ASP.NET component is a 100% managed code web control which can add professional quality 2D barcode images to your ASP.NET web pages hosted on the IIS server.

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

# 2 Create IIS Mapping

If you install IIS after installing the .NET Framework, IIS will not be properly mapped to ASP.NET, you will experience unexpected behavior, you must repair IIS mappings to ASP.NET.

At the command prompt, type the following, and then press ENTER:
"<Windir>\Microsoft.NET\Framework\<Version>\aspnet_regiis.exe" -i

Where <WinDir> is the windows folder (e.g. "c:\windows" or "c:\winnt") and <Version> is the version number of the .NET Framework (e.g. "v2.0.50727").

# 3 Installation and Testing

1. The trial version QRCode ASP.NET web control appends "MW6 Demo" to the string encoded with the QRCode format.

2. Copy "MW6.ASPNET.QRCode.dll" to the bin folder of the IIS server, for example, you can copy MW6.ASPNET.QRCode.dll to the folder "c:\Inetpub\wwwroot\bin".

3. Copy Demo.html, CreateImgVB.aspx, CreateImgCS.aspx and Show.aspx to a folder of the IIS server where Active Server Pages are enabled, for example, you can create one folder "C:\Inetpub\wwwroot \MyFolder" and copy those 4 files to this folder.

4. Enter the URL of Demo.html to your browser for verifying whether QRCode ASP.NET web control is working or not, for example, you can enter http://localhost/MyFolder/Demo.html for testing it on the IIS server itself.

# 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 ASP.NET component.

```
[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 ASP.NET component 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 ASP.NET component 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 ASP.NET component 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 ASP.NET component 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 ASP.NET component 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 the computer screen.

[Visual Basic .NET]

```
Public Sub GetActualSize(ByRef ActualWidth As Integer, ByRef ActualHeight As Integer)
```

[C#]

```
public void GetActualSize(ref int ActualWidth, ref int ActualHeight);
```

**Parameters**

*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 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.

### 4.2.4 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

### 4.2.5 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 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 Server License allows the use of the software (up to 10,000 users) on ONE server with ONE CPU in your organization.

* The 2 Server License allows the use of the software (up to 10,000 users) on 2 servers (each server has only 1 CPU) in your organization.

* The 3 Server License allows the use of the software (up to 10,000 users) on 3 servers (each server has only 1 CPU) in your organization.

* The 4 Server License allows the use of the software (up to 10,000 users) on 4 servers (each server has only 1 CPU) in your organization.

* The 5 Server License allows the use of the software (up to 10,000 users) on 5 servers (each server has only 1 CPU) in your organization.

* The Unlimited Developer License allows the use of the software (unlimited number of users) on unlimited number of servers (each server has unlimited number of CPUs) in your organization.

**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.