

# Table of Contents

Foreword	0
<b>Part I Introduction</b>	<b>3</b>
<b>Part II Create IIS Mapping</b>	<b>3</b>
<b>Part III Installation and Testing</b>	<b>3</b>
<b>Part IV Reference Guide</b>	<b>3</b>
1 Properties.....	3
BackColor Property .....	3
BarColor Property .....	4
BarHeight Property .....	4
BearerBarType Property .....	5
BorderWidth Property .....	5
CheckDigit Property .....	5
CheckDigitToText Property .....	5
CodabarStartChar Property .....	6
CodabarEndChar Property .....	6
Data Property .....	6
NarrowBarWidth Property .....	7
Orientation Property .....	7
ShowText Property .....	8
Supplement Property .....	8
SupplementGap Property .....	8
SupplementType Property .....	9
SymbologyType Property .....	9
TextFont Property .....	9
UPCESystem Property .....	9
Wide2NarrowRatio Property .....	9
2 Methods.....	10
CodeOne Method .....	10
GetActualSize Method .....	10
MicroPDF417 Method .....	11
MicroQRCode Method .....	11
Render Method .....	12
SetSize Method .....	12
3 Enumerations.....	13
BearerBarType Enumeration .....	13
CodaBarChar Enumeration .....	13
Orientation Enumeration .....	13
SupplementType Enumeration .....	13
SymbologyType Enumeration .....	14
UPCESystem Enumeration .....	16
<b>Part V License</b>	<b>16</b>

**Index**

**0**

# 1 Introduction

MW6 barcode ASP.NET web 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 100% managed code web control which can add the professional quality barcode images to your ASP.NET web pages hosted on the IIS server.

## 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 Barcode ASP.NET web control adds "MW6 Demo" at the top of the barcode.
2. Copy "MW6.ASPNET.Barcode.dll" to the bin folder of the IIS server, for example, you can copy MW6.ASPNET.Barcode.dll to the folder "c:\inetpub\wwwroot\bin".
3. Copy Demo.html, CreatelmVB.aspx, CreatelmGS.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 Barcode 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 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 ASP.NET web 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 ASP.NET web 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.

---



(01)12345678901234(10)12345(11)080101

#### 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 ASP.NET web control converts narrow bar width from centimeters to pixels based on the device resolution, round up or round down float pixel value to the nearest integer.

The centimeter to pixel conversion formula is :

$$size\_in\_pixels = size\_in\_centimeters * device\_resolution / 2.54$$

For example, if you render barcode on computer screen and the screen resolution is 96dpi.

(1) Set NarrowBarWidth property to 0.04,  $size\_in\_pixels = 0.04 * 96 / 2.54 = 1.5118$ , round up 1.5118 to 2, so actual narrow bar width is 2 pixels.

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

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

Different NarrowBarWidth property values might end up with same narrow bar width in pixels due to performing 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 ASP.NET web 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 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 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 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 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
<b>1D Barcodes</b>				
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			+H123ABC01234567890D
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			
<b>Postal Code Barcodes</b>				
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
<b>2D Barcodes</b>				
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 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.

---