

Table of Contents

Foreword	0
Part I Introduction	2
Part II Create IIS Mapping	2
Part III Installation and Testing	2
Part IV Reference Guide	2
1 Properties.....	2
BackColor Property	2
BarColor Property	3
CorrectionLevel Property	3
Data Property	3
HandleTilde Property	4
Mode Property	4
ModuleSize Property	4
Orientation Property	5
PreferredFormat Property	5
2 Methods.....	6
GetActualRC Method	6
GetActualSize Method	6
Render Method	7
SetSize Method	7
SetStructuredAppend Method	8
3 Enumerations.....	8
Mode Enumeration	8
Orientation Enumeration	8
PreferredFormat Enumeration	9
Part V License	10
Index	0

1 Introduction

MW6 lightweight Aztec 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.

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 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 Aztec ASP.NET web control appends "MW6 Demo" to the string encoded with the Aztec format.
2. Copy "MW6.ASPNET.Aztec.dll" to the bin folder of the IIS server, for example, you can copy MW6.ASPNET.Aztec.dll to the folder "c:\inetpub\wwwroot\bin".
3. Copy Demo.html, CreatelmVB.aspx, CreatelmCS.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 Aztec 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 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 ASP.NET web 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 ASP.NET web 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 ASP.NET web 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, $\text{size_in_pixels} = 0.06 * 96 / 2.54 = 2.2677$, round down 2.2677 to 2, so actual module size is 2 pixels.

(3) Set ModuleSize property to 0.07, $\text{size_in_pixels} = 0.07 * 96 / 2.54 = 2.6456$, 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 enumPreferredFormat PreferredFormat {get; set;}
```

Remarks

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

ActualHeight

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

4.2.3 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.

4.2.4 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

4.2.5 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 digits)	Capacity (in alphanumeric characters)	Capacity (in 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 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.
