Table of Contents

1 Introduction

MW6 lightweight DataMatrix 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.

DataMatrix is designed to pack a lot of information in a very small space, our DataMatrix ASP.NET web control supports the ECC-200 version, it is capable of encoding 1556 bytes, 2335 alphanumeric characters, or 3116 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 DataMatrix ASP.NET web control appends "MW6 Demo" to the string encoded with the DataMatrix format.

2. Copy "MW6.ASPNET.DataMatrix.dll" to the bin folder of the IIS server, for example, you can copy MW6.ASPNET.DataMatrix.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 DataMatrix 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 DataMatrix 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 DataMatrix 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 DataMatrix ASP.NET web control.

```
[Visual Basic .NET]
Public Property Data As String
```

```
[C#]
public string Data {get; set;}
```

Remarks

The default value is "12".

4.1.4 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 DataMatrix 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).

"~1" is used to indicate FNC1. For example, "**~1**0107612345678900**~1**17100503" can be used to generate GS1 DataMatrix "(01)0107612345678900(17)100503", and "**~1**10AC34563G3" can be used to generate GS1 DataMatrix "(10)AC34563G3".

"~5" is used to indicate Macro 5. For example, "**~5**ABCDEF[GS]123456" can be used to generate DataMatrix "[)>[RS]05[GS]ABCDEF[GS]123456[RS][EOT]".

"~6" is used to indicate Macro 6. For example, "**~6**ABCDEF[GS]123456" can be used to generate DataMatrix "[)>[RS]06[GS]ABCDEF[GS]123456[RS][EOT]".

[RS] is the record separator with ASCII value 30, [GS] is the group separator with ASCII value 29, and [EOT] is the end of transmission with ASCII value 4.

4.1.5 Mode Property

Gets or sets the encoding mode of the DataMatrix barcode.

```
[Visual Basic .NET]
Public Property Mode As enumMode
```

```
[C#]
public enumMode Mode {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 DataMatrix 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, 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 DataMatrix barcode.

```
[Visual Basic .NET]
Public Property Orientation As enumOrientation
```

```
[C#]
public enumOrientation Orientation {get; set;}
```

4.1.8 PreferredFormat Property

Gets or sets the format of the DataMatrix 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 DataMatrix 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 DataMatrix 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 DataMatrix 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 DataMatrix barcode.

ActualCols

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

Remarks

If you set PreferredFormat to pfAuto (Auto format), DataMatrix 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, DataMatrix 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 DataMatrix 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 DataMatrix barcode (in pixels).

ActualHeight

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

4.2.3 Render Method

Renders the DataMatrix 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 DataMatrix barcode.

4.2.4 SetSize Method

Sets the size of the image which contains the DataMatrix 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 DataMatrix barcode, then use this method to set image size by adding surrounding white space around the DataMatrix 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 DataMatrix 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 16, which means that up to 16 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
mdAscii	ASCII mode for mainly encoding ASCII characters (0-127)
mdC40	C40 mode for mainly encoding numeric and upper case characters
mdText	Text mode for mainly encoding numeric and lower case characters
mdBase256	Base256 mode for mainly 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	Data Capacity		
		Numeric	Alphanumeric	Byte
pfAuto	Auto format			
pf10X10	10 X 10 format	6	3	1
pf12X12	12 X 12 format	10	6	3
pf14X14	14 X 14 format	16	10	6
pf16X16	16 X 16 format	24	16	10
pf18X18	18 X 18 format	36	25	16
pf20X20	20 X 20 format	44	31	20
pf22X22	22 X 22 format	60	43	28
pf24X24	24 X 24 format	72	52	34
pf26X26	26 X 26 format	88	64	42
pf32X32	32 X 32 format	124	91	60
pf36X36	36 X 36 format	172	127	84
pf40X40	40 X 40 format	228	169	112
pf44X44	44 X 44 format	288	214	142
pf48X48	48 X 48 format	348	259	172
pf52X52	52 X 52 format	408	304	202
pf64X64	64 X 64 format	560	418	278
pf72X72	72 X 72 format	736	550	366
pf80X80	80 X 80 format	912	682	454
pf88X88	88 X 88 format	1152	862	574
pf96X96	96 X 96 format	1392	1042	694
pf104X104	104 X 104 format	1632	1222	814
pf120X120	120 X 120 format	2100	1573	1048
pf132X132	132 X 132 format	2608	1954	1302
pf140X140	144 X 144 format	3116	2335	1556
pf8X18	8 X 18 format	10	6	3
pf8X32	8 X 32 format	20	13	8
pf12X26	12 X 26 format	32	22	14
pf12X36	12 X 36 format	44	31	20
pf16X36	16 X 36 format	64	46	30
pf16X48	16 X 48 format	98	72	47

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.