
Table of Contents

Foreword	0
Part I Introduction	2
Part II Installation	2
1 Trial Version.....	2
2 Full Version.....	3
Part III Reference Guide	3
1 Properties.....	3
Data Property	3
HandleTilde Property	3
Mode Property	4
PreferredFormat Property	4
2 Methods.....	4
GetPatternData Method	4
SetStructuredAppend Method	5
3 Enumerations.....	6
Mode Enumeration	6
PreferredFormat Enumeration	6
Part IV License	7
Index	0

1 Introduction

MW6 DataMatrix .NET CF package has three 100% managed code libraries for Pocket PC, Windows CE and Smartphone platforms, the libraries can generate DataMatrix format pattern data and your applications can use this pattern data to render DataMatrix barcode onto any device.

DataMatrix is designed to pack a lot of information in a very small space, our DataMatrix .NET CF libraries support the ECC-200 version, it is capable of encoding 1556 bytes, 2335 alphanumeric characters, or 3116 numeric digits.

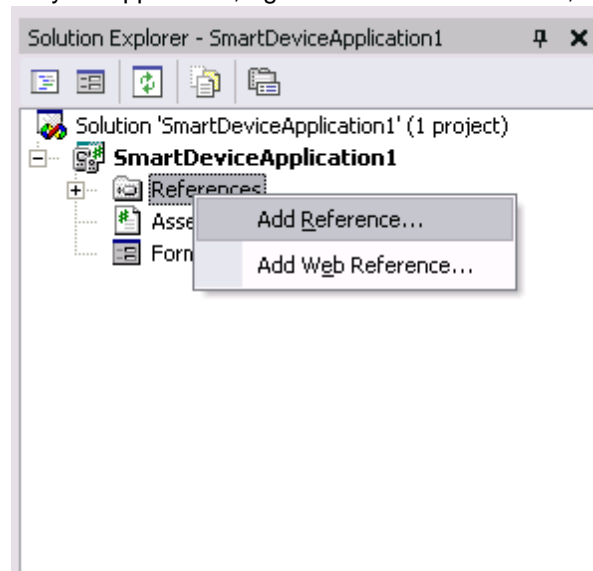
2 Installation

2.1 Trial Version

1. The trial version DataMatrix .NET CF libraries append "MW6 Demo" to the string encoded with the DataMatrix format.
2. After the installation for the trial version is finished, depending on your application platform, copy corresponding DataMatrix .NET CF library file in VB.NET or C# demo sub folder of the destination folder (e.g. "C:\Program Files\MW6 .NET CF\DataMatrix\Platform_Name") to your project folder.

Platform Name	Library Name
Pocket PC	MW6.DataMatrixPC.dll
Windows CE	MW6.DataMatrixCE.dll
Smartphone	MW6.DataMatrixSP.dll

3. In "**Solution Explorer**" for your application, right-click on "**References**", click on "**Add Reference**".



4. In "**Add Reference**" dialog, click on "**Browse**", locate DataMatrix .NET CF library file for your platform, click on "**Open**" to return to "**Add Reference**" dialog, click on "**OK**". If a message box ("The reference MW6.DataMatrixXX.DLL may or may not be valid for the active platform...") pops up, click on "**OK**" and continue.
-

5. Add DataMatrix .NET CF library related code to your application.

2.2 Full Version

1. Uninstall the trial version DataMatrix .NET CF libraries if applicable.
2. Complete the installation for the full version DataMatrix .NET CF libraries, copy MW6DataMatrixNet.dll in VB.NET demo project sub folder or C# demo project sub folder in the destination folder (e.g. "C:\Program Files\MW6 .NET Components\DataMatrix") to your project folder to replace the trial version DataMatrix .NET CF libraries .dll file.

3 Reference Guide

3.1 Properties

3.1.1 Data Property

Gets or sets the message to encode with DataMatrix .NET CF library.

```
[Visual Basic .NET]
```

```
Public Property Data As String
```

```
[C#]
```

```
public string Data {get; set;}
```

Remarks

The default value is "12".

3.1.2 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 .NET CF library 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).

3.1.3 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;}
```

3.1.4 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 .NET CF libraries 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 .NET CF libraries will also automatically choose an appropriate format with bigger data capacity to encode the string.

3.2 Methods

3.2.1 GetPatternData Method

Gets the DataMatrix 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 DataMatrix 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 DataMatrix 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 DataMatrix barcode pattern matrix data and render the DataMatrix barcode onto any device such as the printer.

If you want to create a rotated barcode (90 degrees, 180 degrees or 270 degrees), rotate pattern matrix accordingly before you render the DataMatrix barcode onto a device.

3.2.2 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.

3.3 Enumerations

3.3.1 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

3.3.2 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

4 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 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.
