

# **The Manual for Barcode .NET Standard SDK**

© 2019 MW6 Technologies, Inc.



# Table of Contents

Foreword	0
<b>Part I Introduction</b>	<b>5</b>
<b>Part II BarcodeNet Class</b>	<b>5</b>
1 Properties.....	5
CheckDigit Property .....	5
CodabarStartChar Property .....	5
CodabarEndChar Property .....	5
Data Property .....	6
SymbologyType Property .....	6
UPCESystem Property .....	6
Wide2NarrowRatio Property .....	6
2 Methods.....	7
GetPatternData Method .....	7
3 Enumerations.....	7
CodaBarChar Enumeration .....	7
SymbologyType Enumeration .....	8
UPCESystem Enumeration .....	8
<b>Part III AztecNet Class</b>	<b>8</b>
1 Properties.....	8
Data Property .....	8
HandleTilde Property .....	9
PreferredFormat Property .....	9
2 Methods.....	9
GetPatternData Method .....	9
3 Enumerations.....	10
PreferredFormat Enumeration .....	10
<b>Part IV DataMatrixNet Class</b>	<b>11</b>
1 Properties.....	11
Data Property .....	11
HandleTilde Property .....	11
Mode Property .....	12
PreferredFormat Property .....	12
2 Methods.....	12
GetPatternData Method .....	12
3 Enumerations.....	13
Mode Enumeration .....	13
PreferredFormat Enumeration .....	13
<b>Part V PDF417Net Class</b>	<b>14</b>
1 Properties.....	14
Columns Property .....	14

Data Property .....	14
ErrorCorrectionLevel Property .....	15
HandleTilde Property .....	15
Mode Property .....	15
Rows Property .....	15
TruncateSymbol Property .....	15
<b>2 Methods.....</b>	<b>16</b>
GetPatternData Method .....	16
<b>3 Enumerations.....</b>	<b>17</b>
ECLLevel Enumeration .....	17
Mode Enumeration .....	17
<b>Part VI QRCodeNet Class</b>	<b>17</b>
<b>1 Properties.....</b>	<b>17</b>
Data Property .....	17
Level Property .....	18
Mask Property .....	18
Version Property .....	18
<b>2 Methods.....</b>	<b>18</b>
GetPatternData Method .....	18
<b>3 Enumerations.....</b>	<b>19</b>
Level Enumeration .....	19
Mask Enumeration .....	19
Version Enumeration .....	20
<b>Index</b>	<b>0</b>

# 1 Introduction

Barcode .NET Standard SDK features 5 different class libraries, which can be used to generate the most popular 1D barcodes, DataMatrix, Aztec, PDF417, and QRCode. It is fully compatible with WinForms, .NET Core, Mono, Xamarin, and UWP; in other words, one DLL works fine across multiple platforms.

Please take a look at WinForms' Demo project to figure out how to use DLL properly. In your project, simply add a reference to MW6.Barcode.Std.dll and instantiate a few instances of different classes before using them, then you can leverage .NET graphics or Google SkiaSharp APIs to render barcode onto any device and create barcode images in a wide variety of formats.

## 2 BarcodeNet Class

### 2.1 Properties

#### 2.1.1 CheckDigit Property

Gets or sets a boolean flag indicating whether the check digit is required or not.

```
[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.

#### 2.1.2 CodabarStartChar Property

Gets or sets the start character of CodaBar.

```
[C#]  
public enum CodeBarChar CodabarStartChar {get; set;}
```

#### 2.1.3 CodabarEndChar Property

Gets or sets the end character of CodaBar.

```
[C#]  
public enum CodeBarChar CodabarEndChar {get; set;}
```

### 2.1.4 Data Property

Gets or sets the message to encode with BarcodeNet Class

```
[C#]
```

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

#### Remarks

The default value is "1234". For the trial version copy of library, the first character, if applicable, of this property will be replaced with "1". So for example, if your code passes "56789" to our library, "16789" will be encoded instead of "56789". If this property is blank, the replacement will not occur.

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.



### 2.1.5 SymbologyType Property

Gets or sets the barcode type.

```
[C#]
```

```
public enum SymbologyType SymbologyType {get; set;}
```

### 2.1.6 UPCESystem Property

Gets or sets the encoding system of UPCE barcode.

```
[C#]
```

```
public enum UPCESystem UPCESystem {get; set;}
```

### 2.1.7 Wide2NarrowRatio Property

Gets or sets the ratio of the wide bar to the narrow bar.

```
[C#]
```

```
public int Wide2NarrowRatio {get; set;}
```

#### Remarks

The default value is 2, and typically this property's value is between 2 and 3.

## 2.2 Methods

### 2.2.1 GetPatternData Method

Gets the pattern data of 1D barcode.

```
[C#]  
public bool GetPatternData(ref char[] Buffer,  
                           ref long Size);
```

#### Parameters

##### *Buffer*

Pointer to a buffer that receives a character stream ('1's and '0's) storing the pattern data - '1' indicates a black bar and '0' indicates a white bar.

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 1D barcode pattern matrix ('1's and '0's).

#### 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 pattern data of 1D barcode and render barcode onto any device such as a printer.

## 2.3 Enumerations

### 2.3.1 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'

### 2.3.2 SymbologyType Enumeration

An enumeration type for all possible symbology type values.

Value	Barcode Description	Sample Barcode String
1	Codabar	
2	Code 11	
3	Code 128	1234ABCD+ /
4	Code 128 (Set A)	
5	Code 128 (Set B)	
6	Code 128 (Set C)	
7	Code 39	1234ABCD
8	Code 39 Extended	
9	Code 93	
10	Data Logic 2/5	
11	EAN128/UCC (GS1-128)	(21)95FNC1(11)090101
12	EAN 13	123456789012
13	EAN 8	1234567
14	Interleaved 2 of 5 Barcode	
15	ITF-14 or UPC Shipping Container Symbol	
16	JAN 13	
17	JAN 8	
18	UK Plessey	
19	UPC-A	1234567890
20	UPC-E	1234567

### 2.3.3 UPCESystem Enumeration

An enumeration type for all possible UPCE system values.

#### Members

Name	Comment
ustSystem0	System 0
ustSystem1	System 1

## 3 AztecNet Class

### 3.1 Properties

#### 3.1.1 Data Property

Gets or sets the message to encode with AztecNet Class.

[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.

```
[C#]
```

```
public bool HandleTilde {get; set;}
```

#### Remarks

If this property is set to TRUE, non-printable characters can be passed to AztecNet Class 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 PreferredFormat Property

Gets or sets the format of the Aztec barcode.

```
[C#]
```

```
public enum PreferredFormat PreferredFormat {get; set;}
```

#### Remarks

If you set PreferredFormat to pfAuto (Auto format), our AztecNet Class 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 AztecNet Class 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 pattern data of Aztec barcode.

```
[C#]
```

```
public bool GetPatternData(ref char[] Buffer,  
                           ref long Size,  
                           ref int Rows,  
                           ref int Columns);
```

## Parameters

### Buffer

Pointer to a buffer that receives a character stream ('1's and '0's) storing the pattern data row by row from the top left corner - '1' indicates a black module and '0' indicates a 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 Aztec 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 pattern data of Aztec barcode and render Aztec barcode onto any device such as a printer, only *Data*, *HandleTilde*, and *PreferredFormat* properties affect the pattern data.

## 3.3 Enumerations

### 3.3.1 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

## 4 DataMatrixNet Class

### 4.1 Properties

#### 4.1.1 Data Property

Gets or sets the message to encode with DataMatrixNet Class.

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

#### Remarks

The default value is "12".

#### 4.1.2 HandleTilde Property

Gets or sets a boolean flag indicating whether to process the tilde character "~" or not.

```
[C#]  
public bool HandleTilde {get; set;}
```

**Remarks**

If this property is set to TRUE, non-printable characters can be passed to DataMatrixNet Class 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).

**4.1.3 Mode Property**

Gets or sets the encoding mode of the DataMatrix barcode.

```
[C#]  
public enum Mode {get; set;}
```

**4.1.4 PreferredFormat Property**

Gets or sets the format of the DataMatrix barcode.

```
[C#]  
public enum PreferredFormat PreferredFormat {get; set;}
```

**Remarks**

If you set PreferredFormat to pfAuto (Auto format), our DataMatrixNet Class 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 DataMatrixNet Class will also automatically choose an appropriate format with bigger data capacity to encode the string.

**4.2 Methods****4.2.1 GetPatternData Method**

Gets the pattern data of DataMatrix barcode.

```
[C#]  
public bool GetPatternData(ref char[] Buffer,  
                           ref long Size,  
                           ref int Rows,  
                           ref int Columns);
```

**Parameters***Buffer*

Pointer to a buffer that receives a character stream ('1's and '0's) storing the pattern data row by row from the top left corner - '1' indicates a black module and '0' indicates a 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 pattern data of DataMatrix barcode and render DataMatrix barcode onto any device such as printer, only *Data*, *HandleTilde*, *Mode*, and *PreferredFormat* properties affect the pattern data.

## 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 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 PDF417Net Class

### 5.1 Properties

#### 5.1.1 Columns Property

Gets or sets the preferred number of the columns for the PDF417 barcode.

[C#]

```
public int Columns {get; set;}
```

#### Remarks

Typically this property value should be between 3 and 30.

#### 5.1.2 Data Property

Gets or sets the message to encode with PDF417Net Class

[C#]

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

**Remarks**

The default value is "12".

**5.1.3 ErrorCorrectionLevel Property**

Gets or sets the error correction level of the PDF417 barcode.

```
[C#]  
public enum ECLLevel ErrorCorrectionLevel {get; set;}
```

**5.1.4 HandleTilde Property**

Gets or sets a boolean flag indicating whether to process the tilde character "~" or not.

```
[C#]  
public bool HandleTilde {get; set;}
```

**Remarks**

If this property is set to TRUE, non-printable characters can be passed to PDF417Net Class 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).

**5.1.5 Mode Property**

Gets or sets the encoding mode of the PDF417 barcode.

```
[C#]  
public enum Mode Mode {get; set;}
```

**5.1.6 Rows Property**

Gets or sets the preferred number of the rows for the PDF417 barcode.

```
[C#]  
public int Rows {get; set;}
```

**Remarks**

Typically this property value should be between 3 and 90.

**5.1.7 TruncateSymbol Property**

Gets or sets a boolean flag indicating whether the right side of the PDF417 barcode should be removed or not.

```
[C#]  
public bool TruncateSymbol {get; set;}
```

## 5.2 Methods

### 5.2.1 GetPatternData Method

Gets the pattern data of PDF417 barcode.

```
[C#]  
public bool GetPatternData(ref char[] Buffer,  
                           ref long Size,  
                           ref int Rows,  
                           ref int Columns);
```

#### Parameters

##### *Buffer*

Pointer to a buffer that receives a character stream ('1's and '0's) storing the pattern data row by row from the top left corner - '1' indicates a black bar and '0' indicates a white bar.

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 PDF417 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 pattern data of PDF417 barcode and render PDF417 barcode onto any device such as a printer, only *Columns*, *Data*, *ErrorCorrectionLevel*, *HandleTilde*, *Mode*, *Rows* and *TruncateSymbol* properties affect the pattern data.



## 5.3 Enumerations

### 5.3.1 ECLevel Enumeration

An enumeration type for all possible error correction level values.

#### Members

Name	Comment
ecl_0	Error Correction Level 0
ecl_1	Error Correction Level 1
ecl_2	Error Correction Level 2
ecl_3	Error Correction Level 3
ecl_4	Error Correction Level 4
ecl_5	Error Correction Level 5
ecl_6	Error Correction Level 6
ecl_7	Error Correction Level 7
ecl_8	Error Correction Level 8

### 5.3.2 Mode Enumeration

An enumeration type for all possible encoding mode values.

#### Members

Name	Comment
mdBinary	Binary mode
mdText	Text mode
mdAuto	Auto mode

## 6 QRCodeNet Class

### 6.1 Properties

#### 6.1.1 Data Property

Gets or sets the message to encode with QRCodeNet Class.

[C#]

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

#### Remarks

The default value is "12".

### 6.1.2 Level Property

Gets or sets the level of error correction allowing recovery.

```
[C#]  
public enumLevel Level {get; set;}
```

### 6.1.3 Mask Property

Gets or sets the mask pattern for improving the readability.

```
[C#]  
public enumMask Mask {get; set;}
```

### 6.1.4 Version Property

Gets or sets the version of the QRCode barcode.

```
[C#]  
public enumVersion Version {get; set;}
```

#### Remarks

If you set *Version* to *vrAuto* (Auto version), our QRCodeNet Class 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 QRCodeNet Class will also automatically choose an appropriate version with bigger data capacity to encode the string.

## 6.2 Methods

### 6.2.1 GetPatternData Method

Gets the pattern data of QRCode barcode.

```
[C#]  
public bool GetPatternData(ref char[] Buffer,  
                           ref long Size,  
                           ref int Rows,  
                           ref int Columns);
```

#### Parameters

##### *Buffer*

Pointer to a buffer that receives a character stream ('1's and '0's) storing the pattern data row by row from the top left corner - '1' indicates a black module and '0' indicates a 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 QRCode 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 pattern data of QRCode barcode and render QRCode barcode onto any device such as a printer, only *Data*, *Level*, *Mask* and *Version* properties affect the pattern data.

## 6.3 Enumerations

### 6.3.1 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

### 6.3.2 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

### 6.3.3 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