Table of Contents

1 Introduction

MW6 DataMatrix Win32 DLL can create device independent 2D DataMatrix images for your application, you can save the DataMatrix as either BMP or WMF image file or copy DataMatrix WMF image to the clipboard.

DataMatrix is designed to pack a lot of information in a very small space, our DataMatrix Win32 DLL supports 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. UnZip MW6DataMatrixWin32.ZIP, run the setup.exe to install DataMatrix Win32 DLL.

2. The trial version DataMatrix Win32 DLL appends "MW6 Demo" to the string encoded with DataMatrix format.

2.2 Full Version

1. Uninstall the trial version DataMatrix Win32 DLL if applicable.

2. UnZip full version DataMatrix Win32 DLL .zip file and run the setup.exe to install the full version DataMatrix Win32 DLL.

3 How to Distribute It

If you want to redistribute DataMatrix Win32 DLL as part of your application, on the target machine, simply put **DataMatrixWin32.dll** into the windows 32-bit system folder (e.g. "c:\windows\system32" or "c:\winnt\system32") for 32-bit Windows OS, or the SysWow64 folder (e.g. "c:\windows\SysWow64) for 64-bit Windows OS.

4 Word Demo

4.1 Install Template File

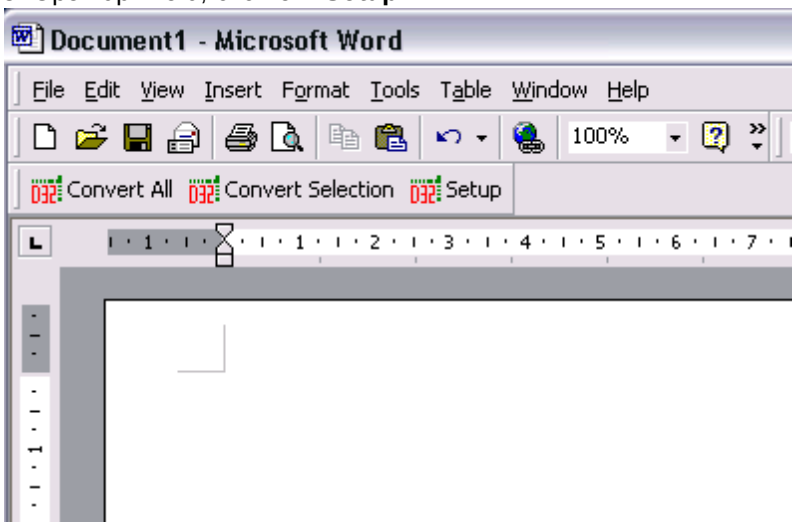
1. Locate the Word Startup folder, the Startup folder can be found in the following locations:

OS	Location
Windows 2000/XP	"C:\Documents and Settings\<user name>\Application Data\Microsoft\Word\Startup"
Windows NT4	"C:\Winnt\Profiles\<user name>\Application Data\Microsoft\Word\Startup"
Windows 95, 98, ME	Office XP: "C:\Program Files\Microsoft Office\Office10\Startup" Office 2000/97:

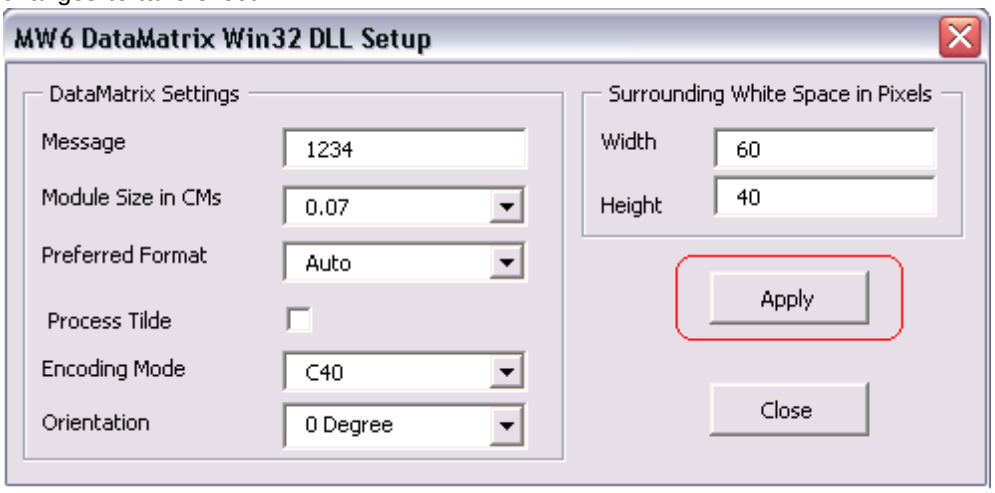
	"C:\Program Files\Microsoft Office\Office\Startup"

2. Copy MW6_DataMatrix_Win32.dot, which usually is in the folder "c:\Program Files\MW6 Win32 DLL \DataMatrix", to the Word Startup folder.

3. Open up Word, click on "**Setup**".

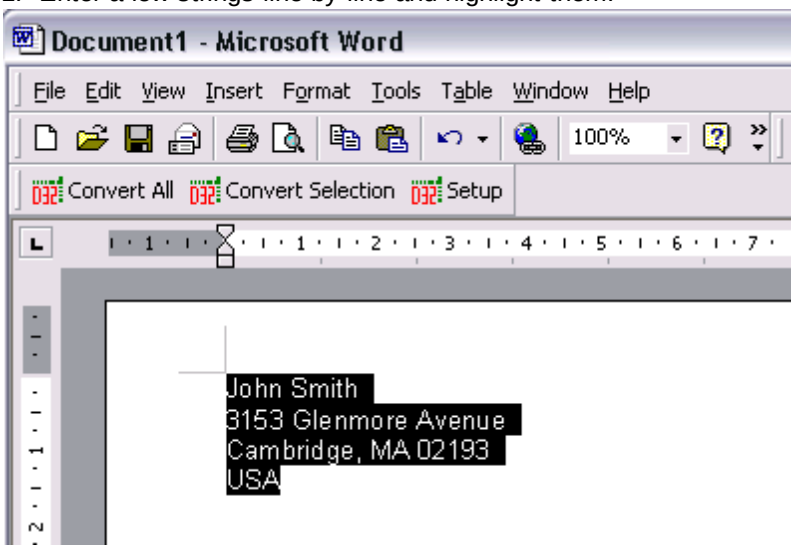


4. Choose a few appropriate values for DataMatrix configurations, click on "**Apply**" button to allow the changes to take effect.

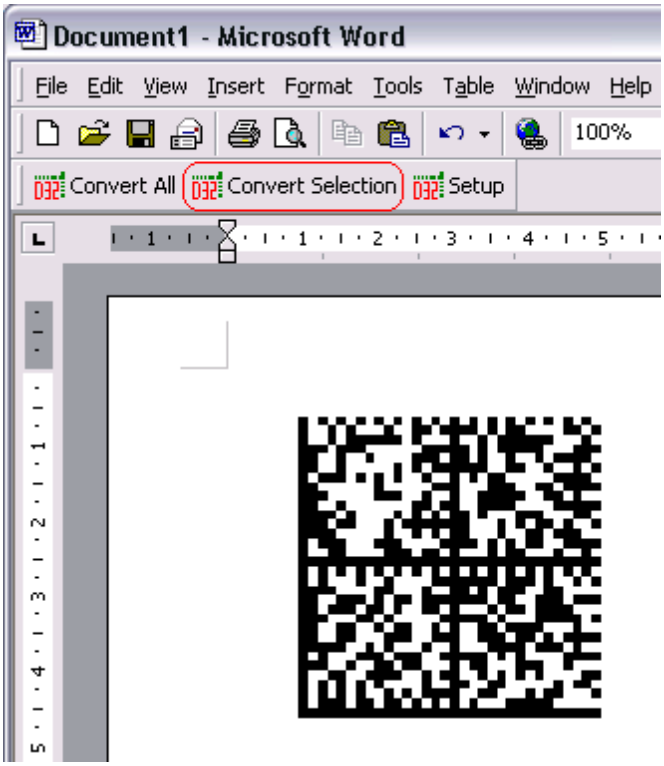


4.2 Create Single Barcode

1. Enter a few strings line by line and highlight them.

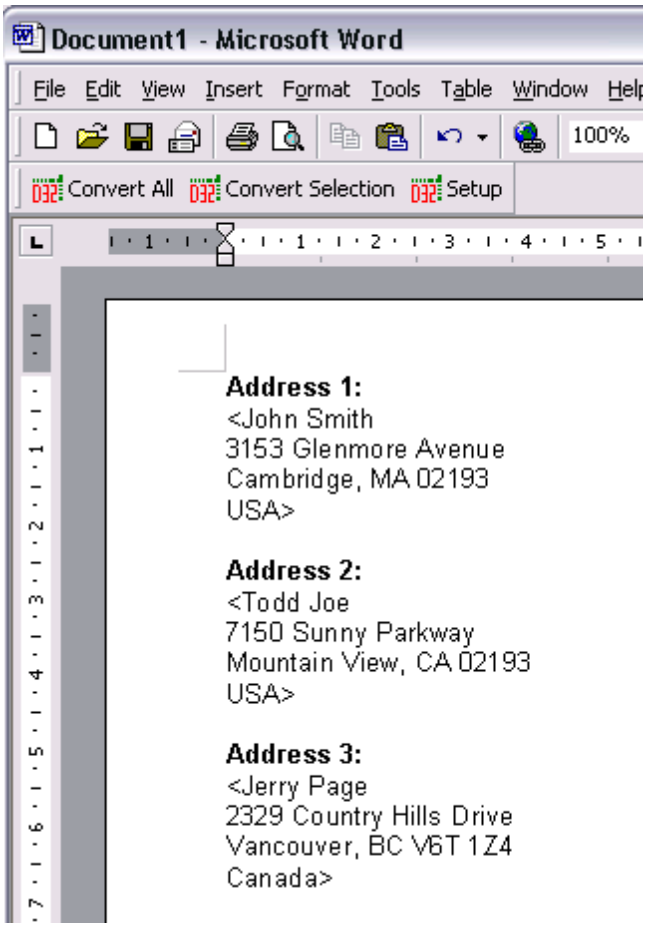


2. Click on "**Convert Selection**" to create a DataMatrix barcode.

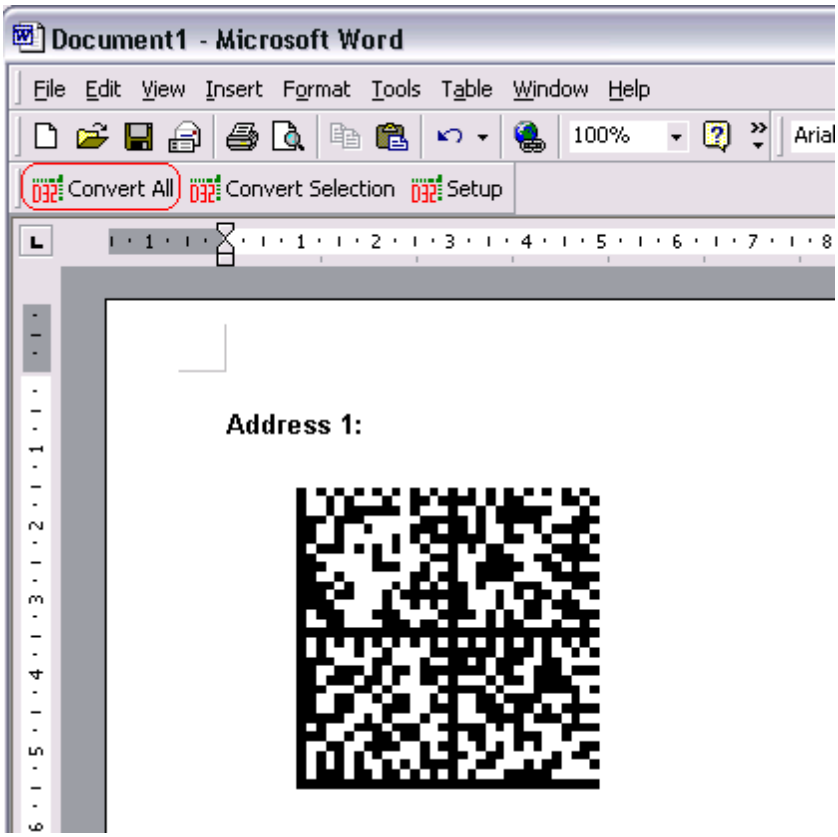


4.3 Create Multiple Barcodes

1. Enter a few paragraphs, surround those paragraphs which will be converted to DataMatrix barcodes with the "<" and ">" characters.

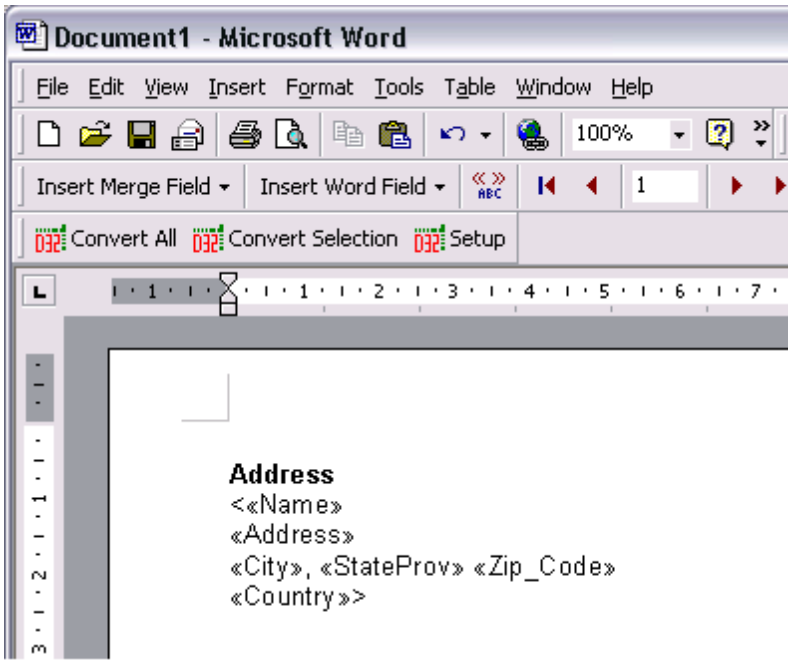


2. Click on "**Convert All**" to create DataMatrix barcodes for the paragraphs surrounded with the "<" and ">" characters.



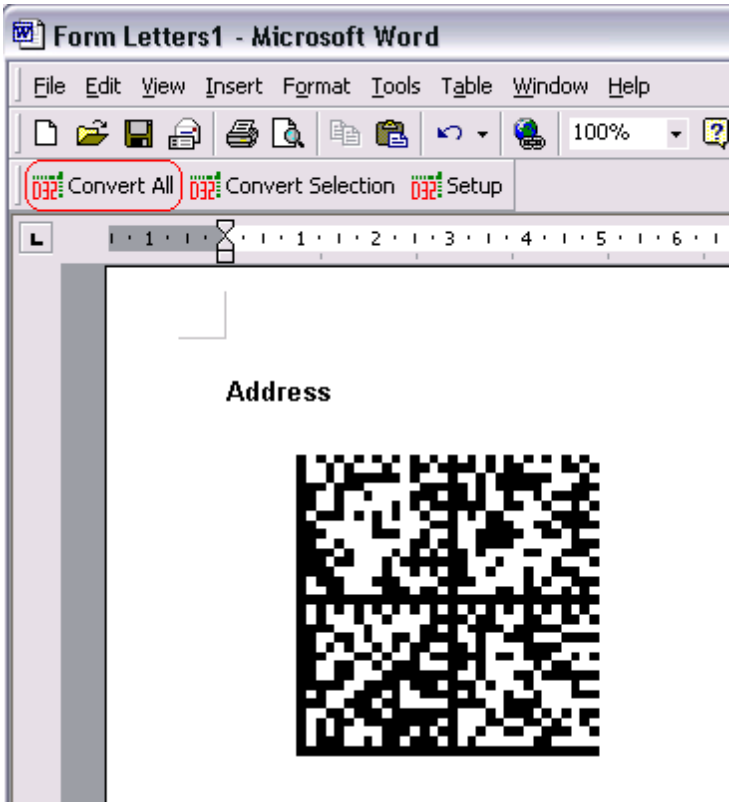
4.4 Mail Merge

1. In Mail Merge, surround the paragraphs which will be converted to DataMatrix barcodes with the "<" and ">" characters.



2. Click on "**Merge ...**"

3. Click on "**Convert All**" to create DataMatrix barcodes for the paragraphs surrounded with the "<" and ">" characters.

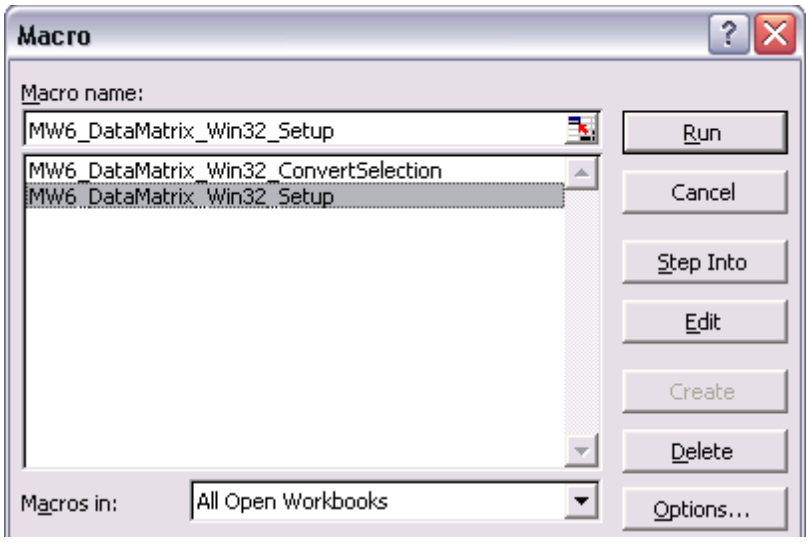


5 Excel Demo

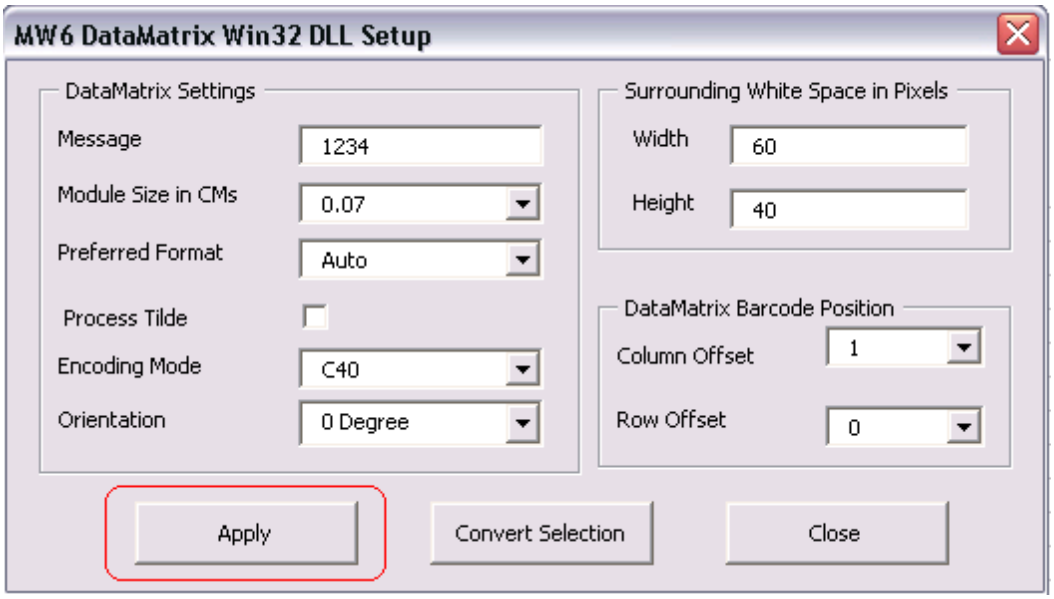
5.1 Change Settings

1. In Excel, open MW6_DataMatrix_Win32.XLS.

2. Click on "**Tools**" > "**Macro**" > "**Macros**", select "**MW6_DataMatrix_Win32_Setup**".



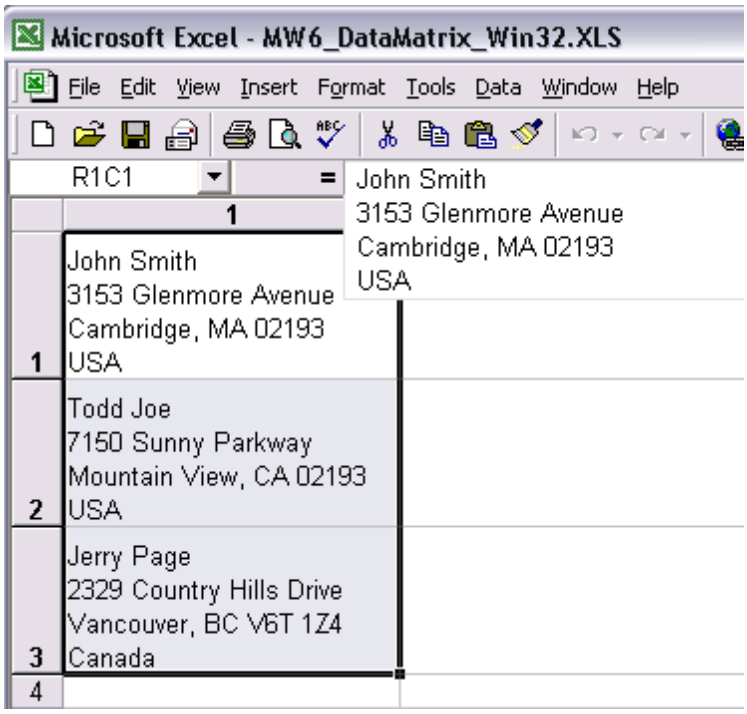
3. Click on "**Run**".



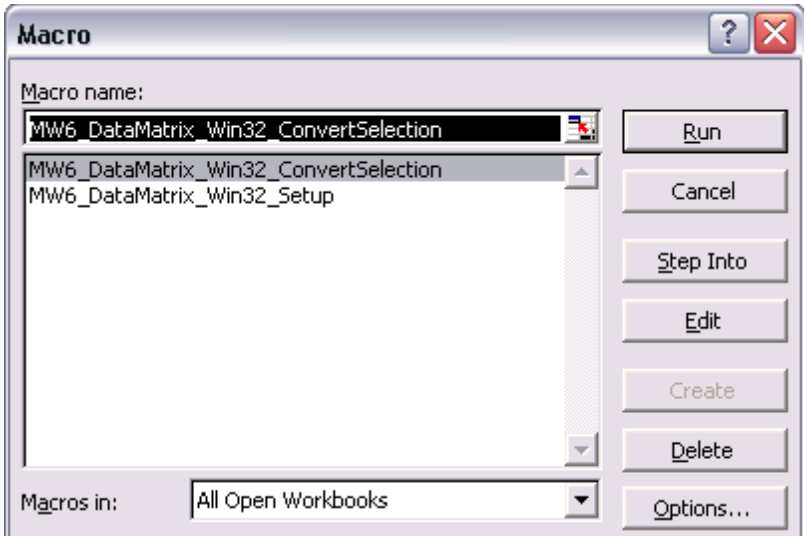
4. Choose a few appropriate values for DataMatrix configurations, click on "**Apply**" button to allow the changes to take effect, "Column Offset" and "Row Offset" are used to specify DataMatrix barcode position relative to the position of the cell which contains the regular string.

5.2 Create Multiple Barcodes

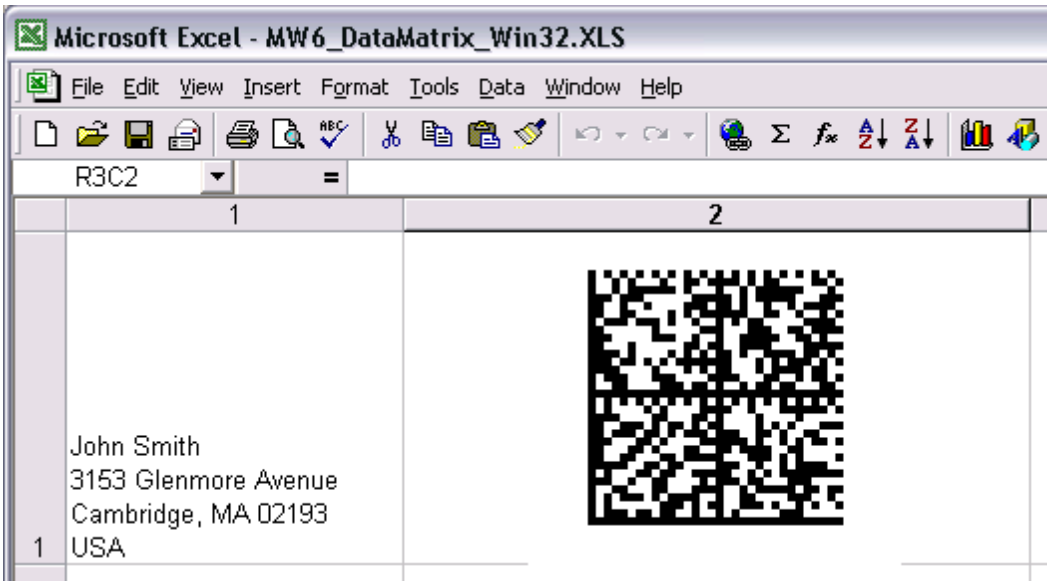
1. Select a few cells.



2. Click on "**Tools**" > "**Macro**" > "**Macros**", select "**MW6_DataMatrix_Win32_ConvertSelection**".



3. Click on "**Run**" to create DataMatrix barcodes for the selected cells.



6 Reference Guide

6.1 DMAppearance Function

Sets up the parameters of DataMatrix barcode appearance.

```
void DMAppearance(double ModuleSize, WORD Orientation, WORD BorderStyle);
```

Parameters

ModuleSize

Specifies the size (width/height) of the square-shaped module, please refer to this note for more information.

Orientation

Specifies the orientation of the DataMatrix barcode, this parameter can be one of the following values:

Value	Description
0	0 degree
1	90 degrees
2	180 degrees
3	270 degrees

BorderStyle

Specifies the style of the border rectangle, this parameter can be one of the following values:

Value	Description
0	No Border
1	Dash Border

2	Solid Border

6.2 DMConfigure Function

Sets up the parameters of DataMatrix barcode.

```
void DMConfigure(WORD Mode, WORD PreferredFormat, BOOL HandleTilde);
```

Parameters

Mode

Indicates which encoding mode is used, this parameter can be one of the following values.

Value	**Description**
0	ASCII mode for mainly encoding ASCII characters (0-127)
1	C40 mode for mainly encoding numeric and upper case characters
2	Text mode for mainly encoding numeric and lower case characters
3	Base256 mode for mainly encoding bytes of data

PreferredFormat

Specifies the format of DataMatrix barcode, please refer to this note for all possible formats.

HandleTilde

Indicates whether to process the tilde character "~" or not, if this parameter is set to TRUE, non-printable characters can be passed to the encoder 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.

6.3 DMCopyToClipboard Function

Copies the DataMatrix barcode WMF format image into the system clipboard.

```
BOOL DMCopyToClipboard();
```

Return Value

If the function succeeds, the return value is a nonzero value, otherwise the return value is zero.

Remarks

Before you call this function, use DMGetActualSize() function to obtain the actual size of the DataMatrix barcode and use DMSetSize() function to set the image size by adding surrounding white space around the DataMatrix barcode.

See Also

DMGetActualSize() Function | DMSetSize() Function

6.4 DMGetActualRC Function

Gets the actual numbers of rows and columns for the DataMatrix barcode.

```
void DMGetActualRC(WORD *ActualRows, WORD *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 0 (Auto format), DataMatrix Win32 DLL will automatically choose an appropriate format with enough data capacity to encode the string, use this function 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 Win32 DLL 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* parameter.

6.5 DMGetActualSize Function

Gets the actual size of the DataMatrix barcode which is rendered onto either computer screen or other devices such as printers.

```
void DMGetActualSize(BOOL ScreenIsTarget,
                     DWORD TargetHDC,
                     DWORD *ActualWidth,
                     DWORD *ActualHeight);
```

Parameters

ScreenIsTarget

Indicates whether the DataMatrix barcode is rendered onto computer screen or not.

TargetHDC

Device context on which to render the DataMatrix barcode, if the parameter *ScreenIsTarget* is set to TRUE, set this parameter to NULL.

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).

6.6 DMGetPatternData Function

Gets the DataMatrix barcode pattern matrix data.

```
BOOL DMGetPatternData(LPVOID lpBuffer,
                      DWORD *Size,
                      WORD *Rows,
                      WORD *Columns);
```

Parameters

lpBuffer

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 *lpBuffer*. 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 function to obtain the DataMatrix barcode pattern matrix data and render the DataMatrix barcode onto any device such as the printer, call *DMSetMessage()* and *DMConfigure()* functions before calling this function, other functions don't affect the output of DataMatrix barcode pattern matrix.

Based on the *Orientation* parameter value, rotate the pattern matrix accordingly before you render the DataMatrix barcode onto a device.

6.7 DMRender Function

Renders the DataMatrix barcode onto the device such as computer screen or printers.

```
void DMRender(DWORD hDC, WORD x, WORD y);
```

Parameters

hDC

Device context on which to render the DataMatrix barcode.

x

The x coordinate, in pixels, of the top left corner of the DataMatrix barcode .

y

The y coordinate, in pixels, of the top left corner of the DataMatrix barcode.

6.8 DMSaveAsBMP Function

Saves the DataMatrix barcode image as a BMP file.

```
BOOL DMSaveAsBMP(LPCTSTR FileName);
```

Parameters

FileName

A string that contains the name of the file to which to save BMP format DataMatrix barcode image.

Return Value

If the function succeeds, the return value is a nonzero value, otherwise the return value is zero.

Remarks

Before you call this function, use DMGetActualSize() function to obtain the actual size of the DataMatrix barcode and use DMSetSize() function to set image size by adding surrounding white space around the DataMatrix barcode.

See Also

DMGetActualSize() Function | DMSetSize() Function

6.9 DMSaveAsWMF Function

Saves the DataMatrix barcode image as a WMF file.

```
BOOL DMSaveAsWMF(LPCTSTR FileName);
```

Parameters

FileName

A string that contains the name of the file to which to save WMF format DataMatrix barcode image.

Return Value

If the function succeeds, the return value is a nonzero value, otherwise the return value is zero.

Remarks

Before you call this function, use DMGetActualSize() function to obtain the actual size of the DataMatrix barcode and use DMSetSize() function to set image size by adding surrounding white space around the DataMatrix barcode.

See Also

DMGetActualSize() Function | DMSetSize() Function

6.10 DMSetBackColor Function

Specifies the RGB triplet of the background color.

```
void DMSetBackColor(WORD red, WORD green, WORD blue);
```

Parameters

red

Specifies the value of red component for a RGB triplet, the valid value should be between 0 and 255.

green

Specifies the value of green component for a RGB triplet, the valid value should be between 0 and 255.

blue

Specifies the value of blue component for a RGB triplet, the valid value should be between 0 and 255.

6.11 DMSetBarColor Function

Specifies the RGB triplet of the color for the narrow bar.

```
void DMSetBarColor(WORD red, WORD green, WORD blue);
```

Parameters

red

Specifies the value of red component for a RGB triplet, the valid value should be between 0 and 255.

green

Specifies the value of green component for a RGB triplet, the valid value should be between 0 and 255.

blue

Specifies the value of blue component for a RGB triplet, the valid value should be between 0 and 255.

6.12 DMSetDefault Function

Initializes the DataMatrix barcode parameters with the default values.

```
void DMSetDefault();
```

6.13 DMSetMessage Function

Specifies the message to encode using the appropriate parameters.

```
void DMSetMessage(LPCTSTR Message);
```

Parameters

Message

A string that contains the message to encode using the appropriate parameters.

6.14 DMSetSize Function

Sets the size of the image which contains the DataMatrix barcode.

```
void DMSetSize(DWORD Width, DWORD Height);
```

Parameters

Width

The width, in pixels, of the image.

Height

The height, in pixels, of the image.

Remarks

First call DMGetActualSize() function to obtain the actual size of the DataMatrix barcode, then use this function to set image size by adding surrounding white space around the DataMatrix barcode.

See Also

DMGetActualSize() Function

6.15 DMSetStructuredAppend Function

Specifies which symbol this is in a sequence and the total number of symbols in the sequence.

```
void DMSetStructuredAppend(BOOL AllowSA, WORD SymbolID, WORD 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 function if you don't need the structured append feature.

7 Convert Size from CMs to Pixels

Internally our DataMatrix Win32 DLL converts the module size 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 the DataMatrix barcode onto the computer screen and the screen resolution is 96dpi.

(1) Set *ModuleSize* parameter to 0.04, size_in_pixels = 0.04 * 96 / 2.54 = 1.5118, round up 1.5118 to 2, so actual module sizewidth is 2 pixels.

(2) Set *ModuleSize* parameter to 0.06, size_in_pixels = 0.06 * 96 / 2.54 = 2.2677, round down 2.2677 to 2, so actual module size width is 2 pixels.

(3) Set *ModuleSize* parameter 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* parameter values might end up with same module size width in pixels due to performing rounding operations.

8 DataMatrix Formats

This table lists all possible formats of the DataMatrix barcode and their data capacity.

Value	Description	Data Capacity		
		Numeric	Alphanumeric	Byte
0	Auto format			
1	10 X 10 format	6	3	1
2	12 X 12 format	10	6	3
3	14 X 14 format	16	10	6
4	16 X 16 format	24	16	10
5	18 X 18 format	36	25	16
6	20 X 20 format	44	31	20
7	22 X 22 format	60	43	28
8	24 X 24 format	72	52	34
9	26 X 26 format	88	64	42
10	32 X 32 format	124	91	60
11	36 X 36 format	172	127	84
12	40 X 40 format	228	169	112
13	44 X 44 format	288	214	142
14	48 X 48 format	348	259	172
15	52 X 52 format	408	304	202
16	64 X 64 format	560	418	278
17	72 X 72 format	736	550	366
18	80 X 80 format	912	682	454
19	88 X 88 format	1152	862	574
20	96 X 96 format	1392	1042	694
21	104 X 104 format	1632	1222	814
22	120 X 120 format	2100	1573	1048
23	132 X 132 format	2608	1954	1302
24	144 X 144 format	3116	2335	1556
25	8 X 18 format	10	6	3
26	8 X 32 format	20	13	8
27	12 X 26 format	32	22	14
28	12 X 36 format	44	31	20
29	16 X 36 format	64	46	30
30	16 X 48 format	98	72	47

If you set *PreferredFormat* to 0 (Auto format), our DataMatrix Win32 DLL 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 Win32 DLL will also automatically choose an appropriate format

with bigger data capacity to encode the string.

9 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 User License allows the use of the software on ONE computer by ONE person in your organization.

* 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.