

# Table of Contents

Foreword	0
<b>Part I Introduction</b>	<b>3</b>
<b>Part II Installation</b>	<b>3</b>
1 Trial Version.....	3
2 Full Version.....	3
<b>Part III How to Distribute It</b>	<b>3</b>
<b>Part IV Office 2007 &amp; 2010</b>	<b>4</b>
1 Word .....	4
Run Setup .....	4
Create Single Barcode .....	5
Create Multiple Barcodes .....	5
Mail Merge .....	6
2 Excel.....	10
Change Settings .....	10
Create Multiple Barcodes .....	11
3 Access.....	12
<b>Part V Office 2000 &amp; 2003</b>	<b>16</b>
1 Word.....	16
Run Setup .....	16
Create Single Barcode .....	17
Create Multiple Barcodes .....	18
Mail Merge .....	20
2 Excel.....	21
Change Settings .....	21
Create Multiple Barcodes .....	23
3 Access.....	24
<b>Part VI Reference Guide</b>	<b>27</b>
1 Properties.....	27
BackColor Property .....	27
BarColor Property .....	27
BorderStyle Property .....	28
Data Property .....	28
HandleTilde Property .....	28
ImageData Property .....	29
Mode Property .....	29
ModuleSize Property .....	29
Orientation Property .....	30
PreferredFormat Property .....	30

<b>2</b>	<b>Methods.....</b>	<b>31</b>
	CopyToClipboard Method .....	31
	GetActualRC Method .....	32
	GetActualSize Method .....	32
	GetPatternData Method .....	33
	Render Method .....	33
	SaveAsBMP Method .....	34
	SaveAsWMF Method .....	34
	SetSize Method .....	35
	SetStructuredAppend Method .....	35
	<b>Part VII License</b>	<b>36</b>
	<b>Index</b>	<b>0</b>

---

# 1 Introduction

MW6 DataMatrix ActiveX is a powerful ATL-based control for handling DataMatrix 2D barcode and can be used in any ActiveX-compliant environment such as Word, Access, Excel, VB.NET, C#.NET, Visual Basic, Visual C++, Visual FoxPro, Delphi or C++ Builder.

DataMatrix is designed to pack a lot of information in a very small space, our DataMatrix ActiveX control 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 MW6DataMatrix.zip, run the setup.exe to install DataMatrix ActiveX.
2. The trial version DataMatrix ActiveX appends "MW6 Demo" to the string encoded with DataMatrix barcode.

### 2.2 Full Version

1. Uninstall the trial version DataMatrix ActiveX if applicable.
2. UnZip full version DataMatrix ActiveX .zip file and run the setup.exe to install the full version DataMatrix ActiveX.

## 3 How to Distribute It

If you want to redistribute the DataMatrix ActiveX as part of your application, please follow the instructions below:

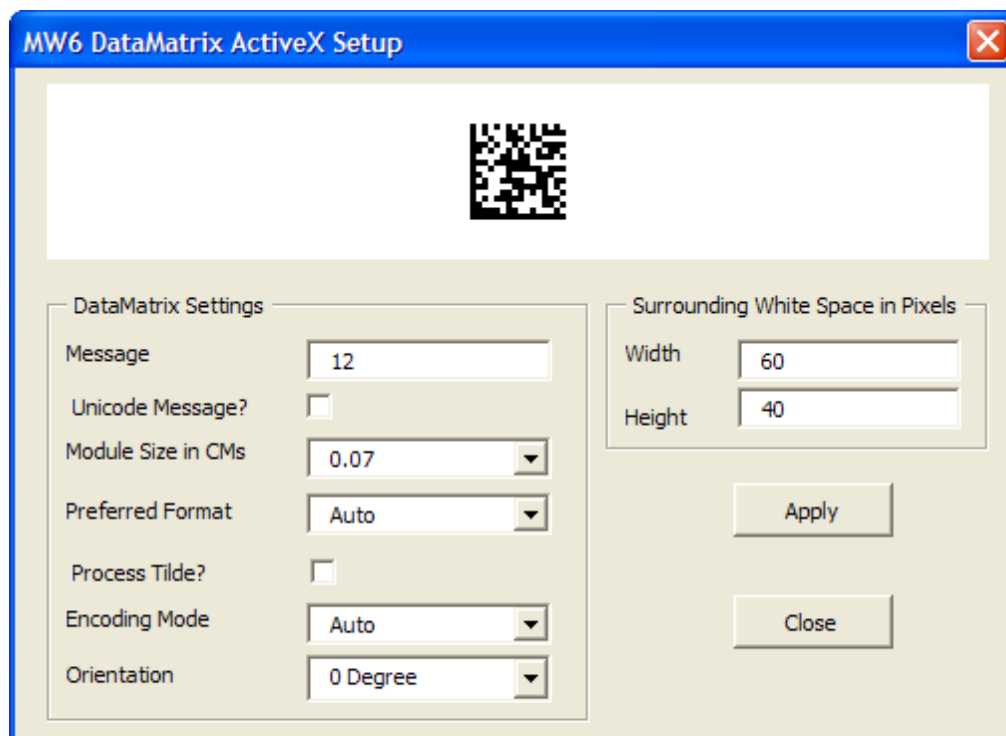
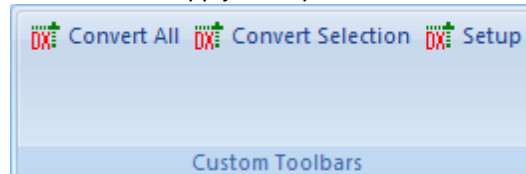
- 1) For 32-bit version Windows OS, put **DataMatrix.dll** into the windows 32-bit system folder (e.g. "c:\windows\system32" or "c:\winnt\system32") on the target machine and run "regsvr32 DataMatrix.dll" to register it.
- 2) For 64-bit version Windows OS, put **DataMatrix.dll** into the SysWOW64 folder (e.g. "c:\windows\SysWOW64") on the target machine, and run the following commands to register it:
  - cd c:\windows\SysWOW64
  - regsvr32 DataMatrix.dll
- 3) If you want to use DataMatrix ActiveX in 64-bit version Office Word, Excel or Access, put 64-bit version **DataMatrix\_x64.dll** into "c:\windows\system32" folder, and run the following commands to register it:
  - cd c:\windows\system32
  - regsvr32 DataMatrix\_x64.dll
- 4) For Windows Vista or above, you need to use an elevated Command Prompt to run *regsvr32.exe* command, click "Start" > "All Programs" > "Accessories", right-click "Command Prompt", and then click "Run as administrator".

## 4 Office 2007 & 2010

### 4.1 Word

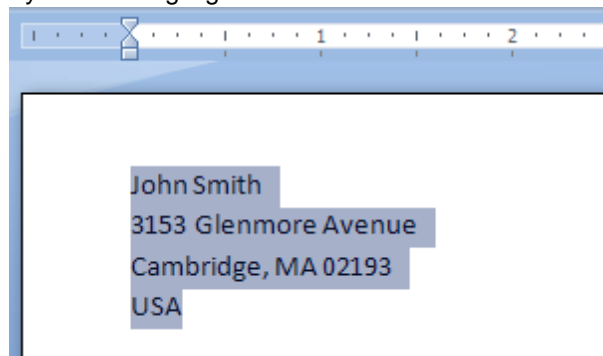
#### 4.1.1 Run Setup

1. Click on "**Add-Ins**", then click on "**Setup**", change the configurations for DataMatrix format, if the string contains some Unicode texts (Japanese, Chinese, Korean, etc), toggle on "Unicode Message?" check box, so the VBA macro code can apply the special treatments to those Unicode characters.



## 4.1.2 Create Single Barcode

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

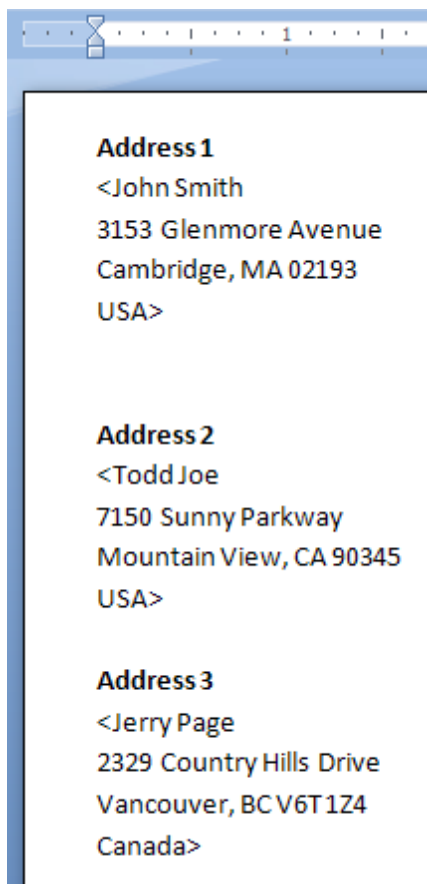


2. Click on "**Add-Ins**", then click on "**Convert Selection**" to create a DataMatrix barcode.

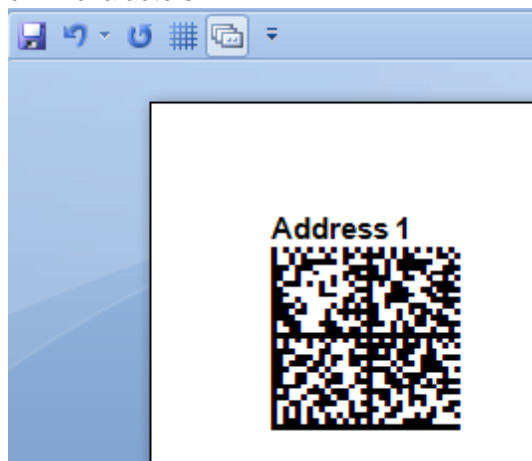


## 4.1.3 Create Multiple Barcodes

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

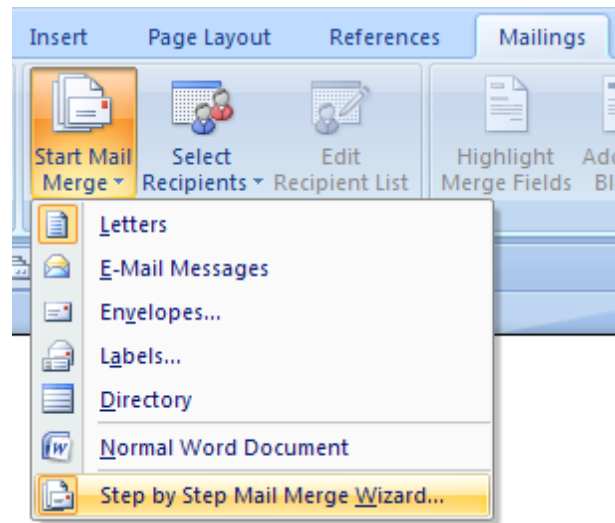


2. Click on "**Add-Ins**", then click on "**Convert All**" to create DataMatrix barcodes for the string sections surrounded with the "<" and ">" characters.

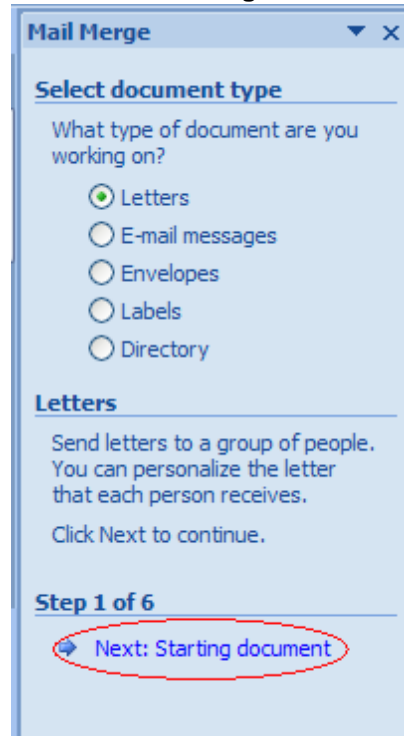


#### 4.1.4 Mail Merge

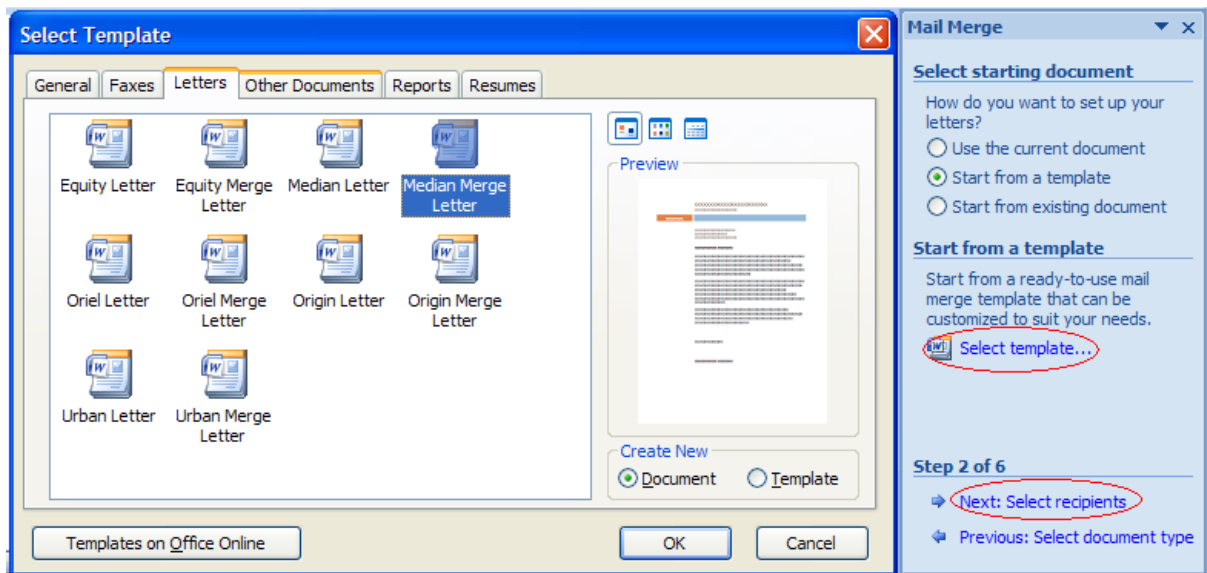
1. Click on "**Mailings**", then click on "**Start Mail Merge**". A drop-down list appears as shown below, select the last option "**Step by Step Mail Merge Wizard**".



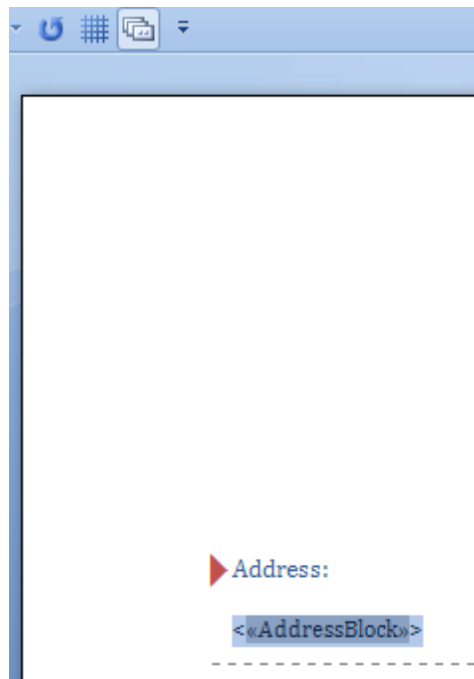
2. Select a document type and click on "**Next: Starting document**".



3. Click on "**Start from a template**", then click on the link "**Select template**", choose a template, click on "**Next: Select recipients**".

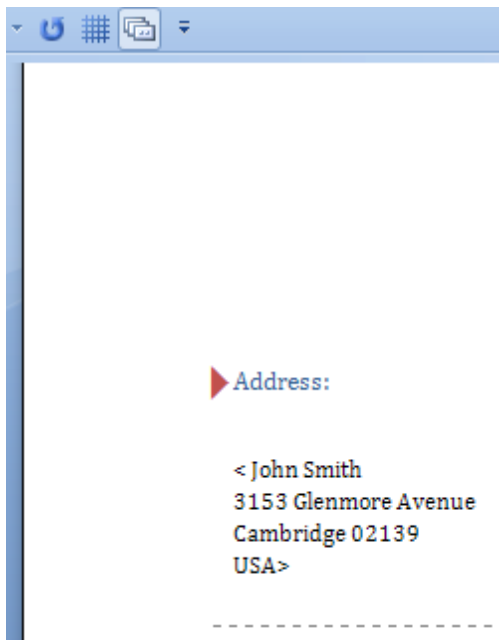


4. Select **"Use an existing list"** and click on **"Browser"** link, choose "MW6\_DataMatrix\_ActiveX.accdb" database as an existing list, click **"Next: Write your letter"**.
5. Surround the section which will be converted to the DataMatrix barcode with the "<" and ">" characters and highlight it.

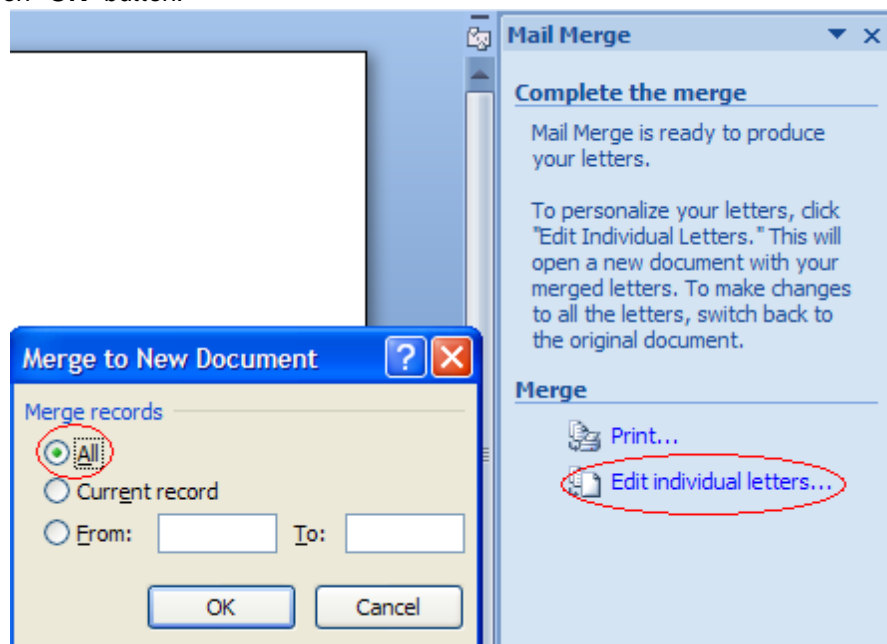


6. Click on **"Next: Preview your letters"**, then click on **"Next: Complete the merge"**.

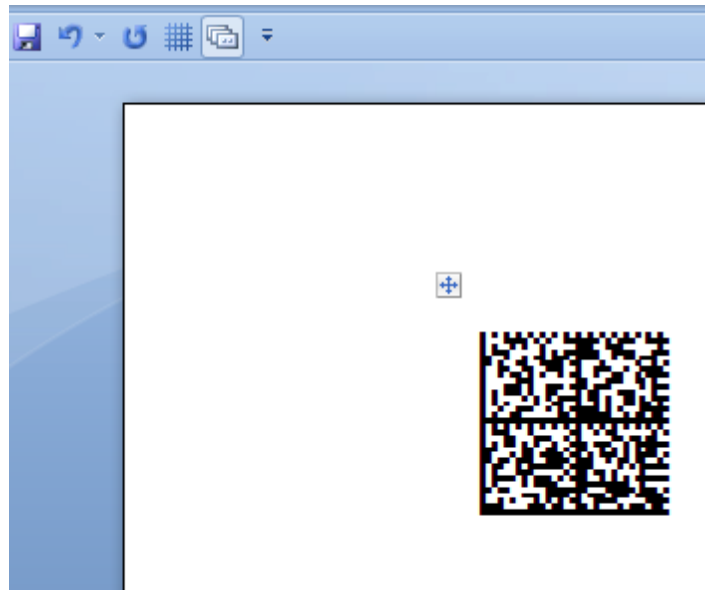




7. Click on "**Edit individual letters**", this opens "**Merge to New Document**" dialog, click on "**All**" and then click on "**OK**" button.



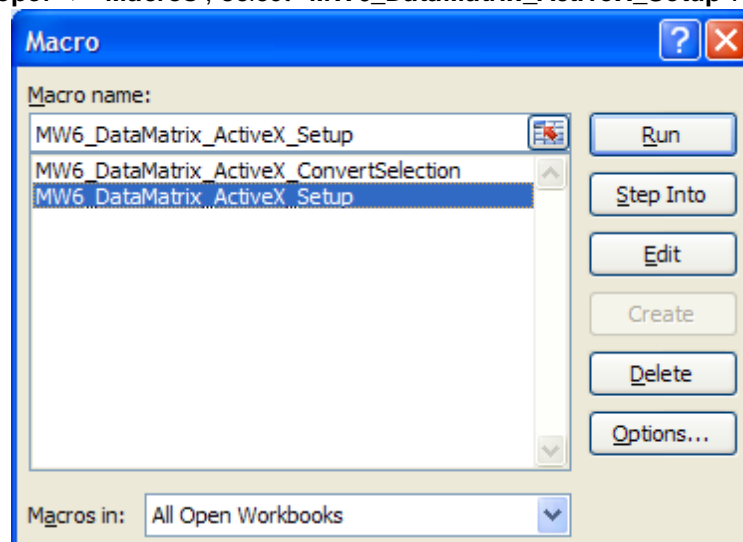
8. Click on "**Add-Ins**", then click on "**Convert All**" to create DataMatrix barcodes.



## 4.2 Excel

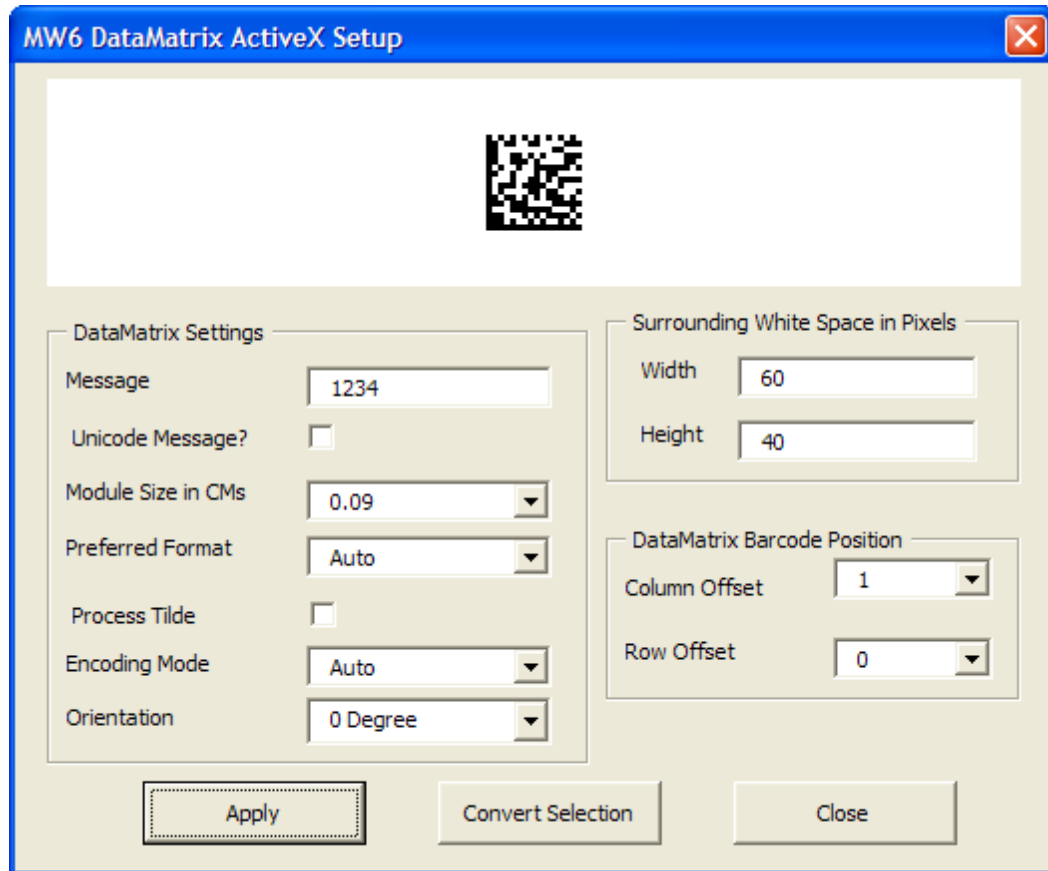
### 4.2.1 Change Settings

1. In Excel, open MW6\_DataMatrix\_ActiveX.xlsm.
2. If you see "**Security Warning, Macros have been disabled**", click on "**Options**" to open "**Microsoft Office Security Options**" dialog, toggle on "**Enable this content**" check box.
3. Click on "**Developer**" > "**Macros**", select "**MW6\_DataMatrix\_ActiveX\_Setup**".



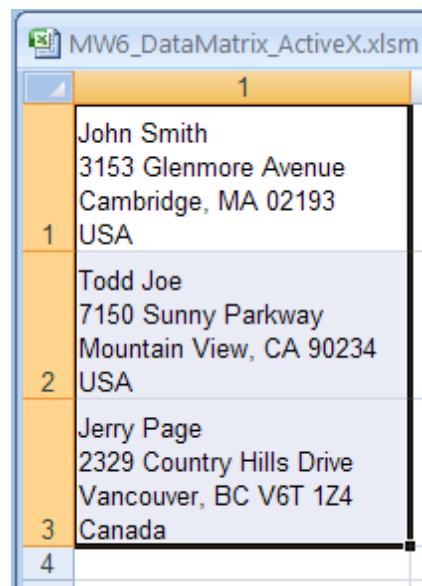
4. Choose a few appropriate values for DataMatrix configurations, "**Column Offset**" and "**Row Offset**" are used to specify the barcode position relative to the position of a cell which contains the regular string. If the string contains some Unicode texts (Japanese, Chinese, Korean, etc), toggle on "Unicode Message?" check box, so the VBA macro code can apply the special treatments to those

Unicode characters.

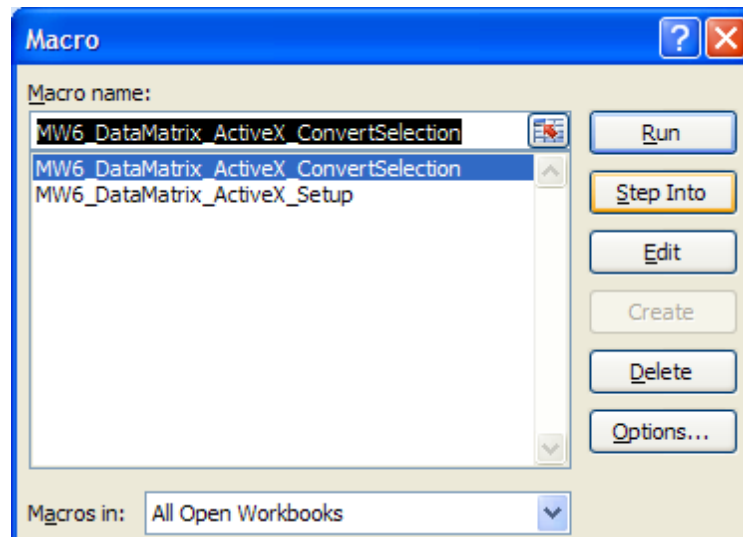


## 4.2.2 Create Multiple Barcodes

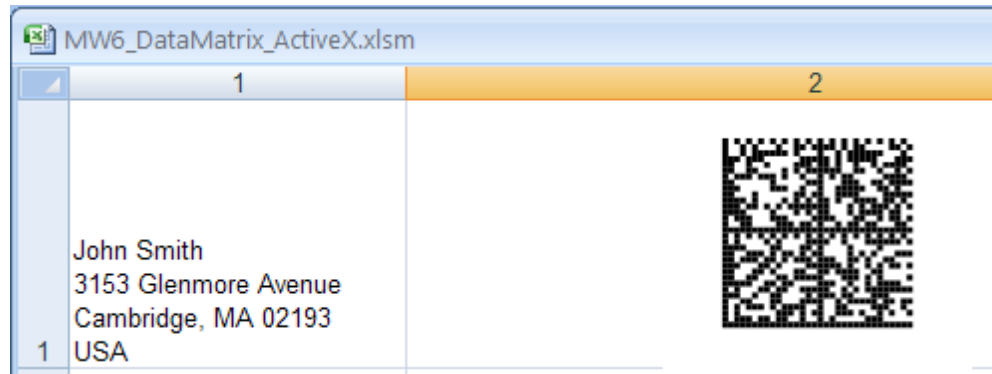
1. Select a few cells.



2. Click on "Developer" > "Macros", select "MW6\_DataMatrix\_ActiveX\_ConvertSelection".

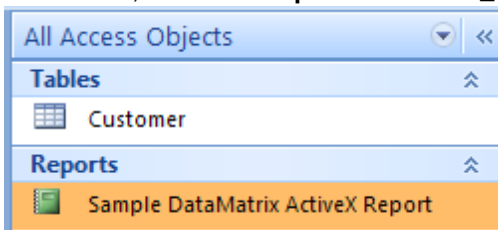


3. Click on "Run" to create the barcodes for the selected cells.

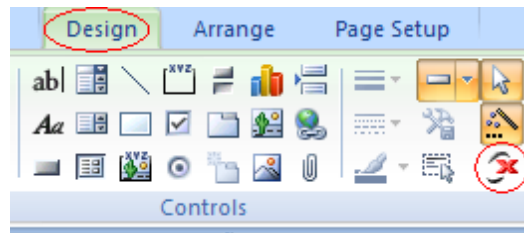


## 4.3 Access

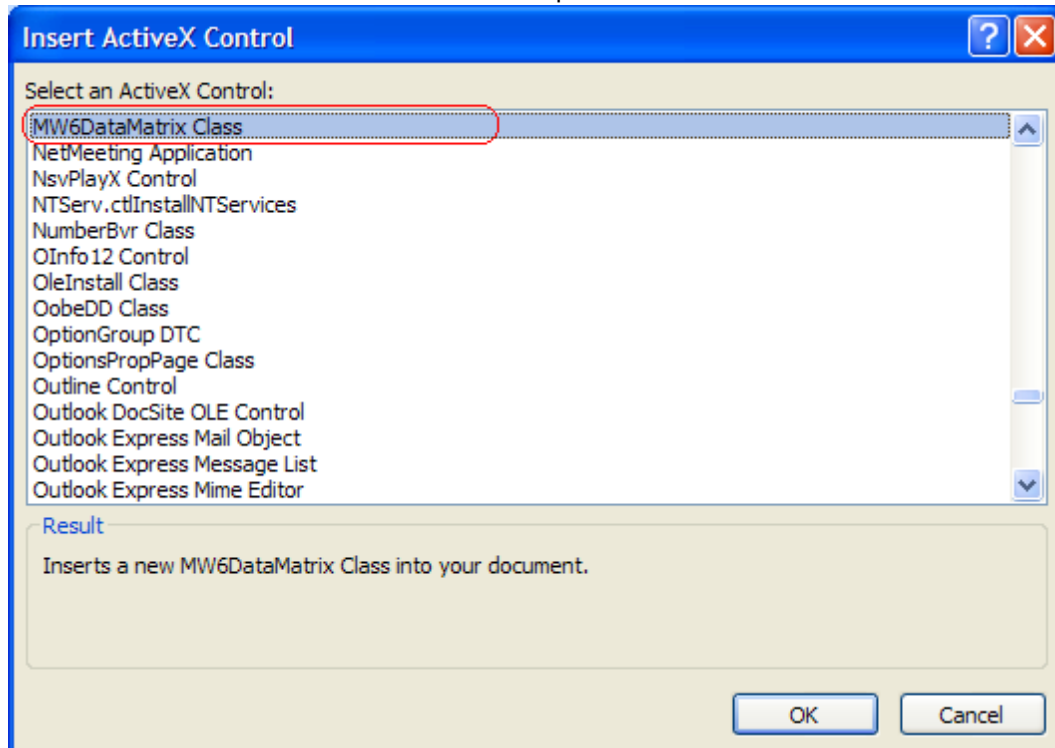
1. Open MW6\_DataMatrix\_ActiveX.accdb, select "Sample DataMatrix\_ActiveX Report".



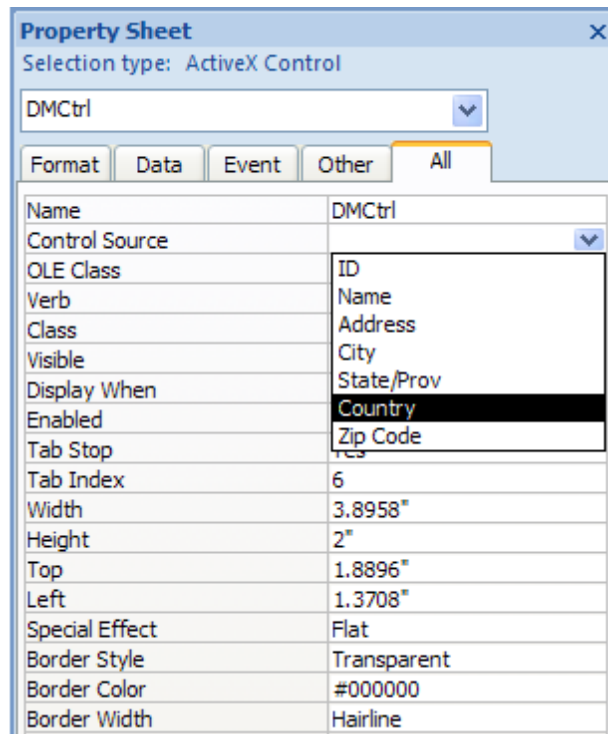
2. If you see "Security Warning, Certain content in the database has been disabled", click on "Options" to open up "Microsoft Office Security Options" dialog, toggle on "Enable this content" check box.
3. Click on "Design View", click "Design" > "Insert ActiveX Control".



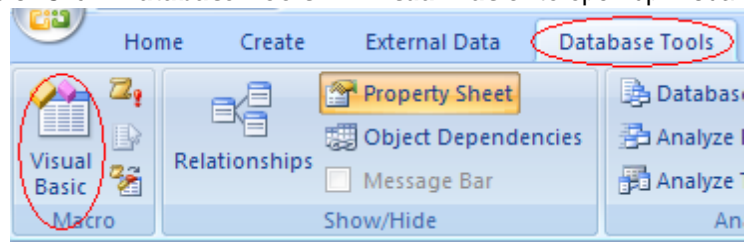
4. Insert a MW6 DataMatrix ActiveX control into the report.



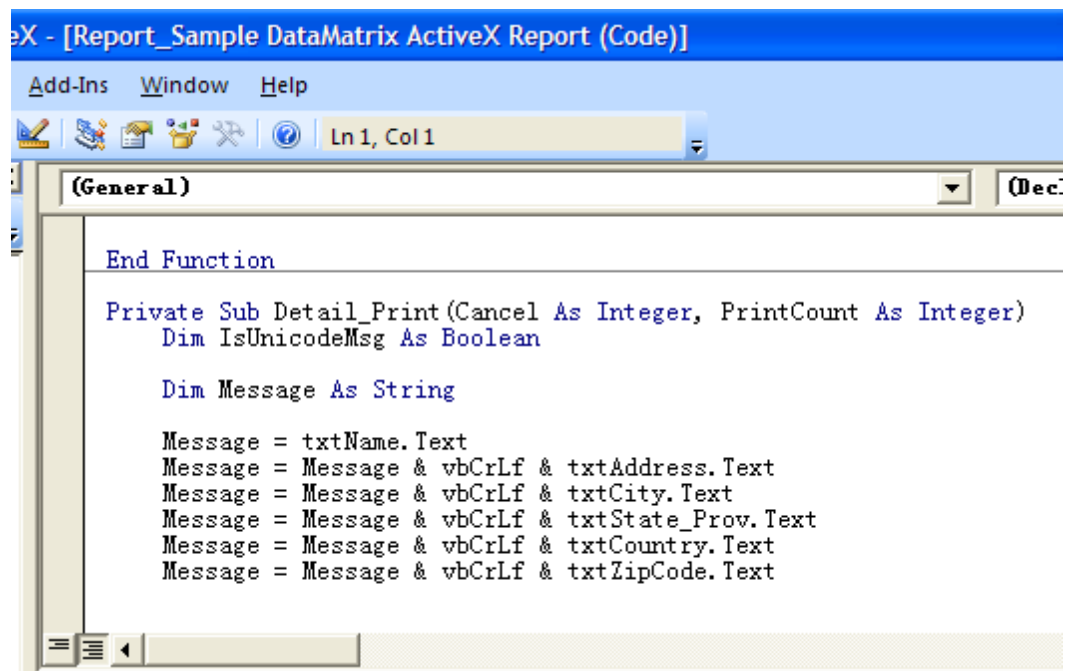
5. Change its properties to meet your application requirements, our DataMatrix ActiveX supports the data binding so you can bind a field in a database to the control and generate a barcode for each data record automatically, there is an arrow on the right side of the "**Control Source**" property, click on the arrow, a list opens with all fields, select the field you want for the control.



6. If the string contains some Unicode texts (Japanese, Chinese, Korean, etc), use "*Private Sub Detail\_Print(Cancel As Integer, PrintCount As Integer)*" to apply the special treatments to those Unicode characters. Click "**Database Tools**" > "**Visual Basic**" to open up Visual Basic Editor.



7. Change the code in the sub "*Detail\_Print( ... )*" to meet your application requirements.



The screenshot shows a code editor window titled "eX - [Report\_Sample DataMatrix ActiveX Report (Code)]". The window has a menu bar with "Add-Ins", "Window", and "Help". Below the menu bar is a toolbar with icons for undo, redo, save, print, and a status bar showing "Ln 1, Col 1". The main area is a code editor with a tab labeled "(General)". The code is as follows:

```
End Function

Private Sub Detail_Print(Cancel As Integer, PrintCount As Integer)
    Dim IsUnicodeMsg As Boolean

    Dim Message As String

    Message = txtName.Text
    Message = Message & vbCrLf & txtAddress.Text
    Message = Message & vbCrLf & txtCity.Text
    Message = Message & vbCrLf & txtState_Prov.Text
    Message = Message & vbCrLf & txtCountry.Text
    Message = Message & vbCrLf & txtZipCode.Text
```

8. Click on "**Preview**" to view the barcodes.

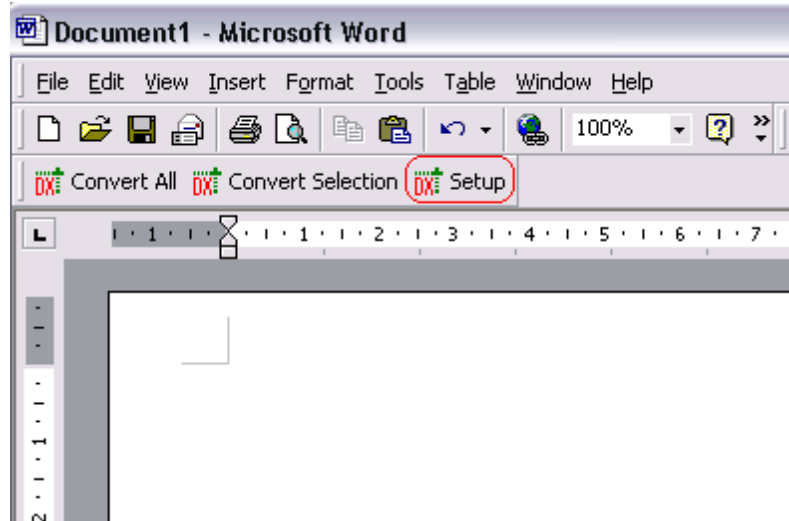


## 5 Office 2000 & 2003

### 5.1 Word

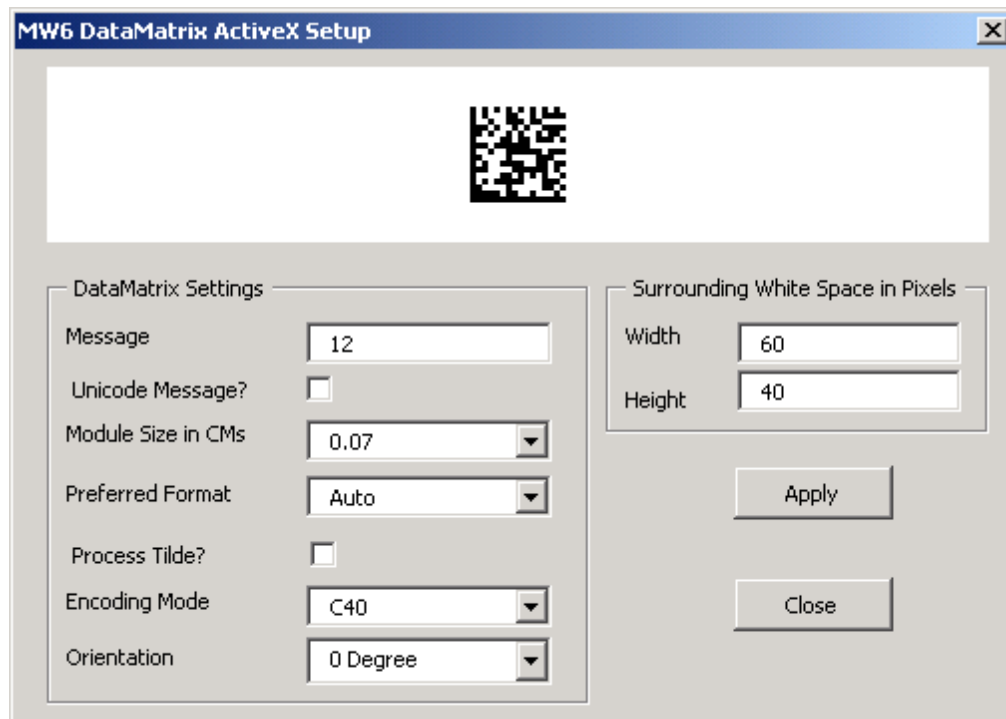
#### 5.1.1 Run Setup

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



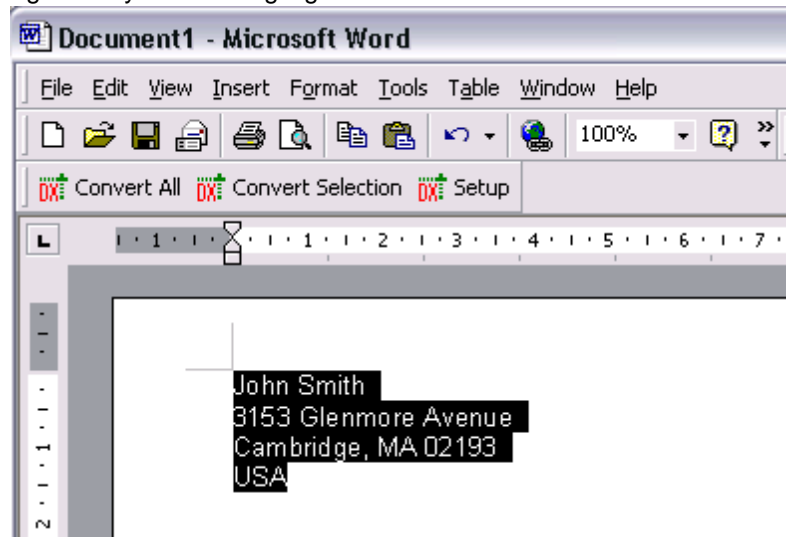
2. Choose a few appropriate values for DataMatrix configurations, click on "**Apply**" button to allow the changes to take effect. If the string contains some Unicode texts (Japanese, Chinese, Korean, etc), toggle on "Unicode Message?" check box, so the VBA macro code can apply the special treatments to those Unicode characters.
-



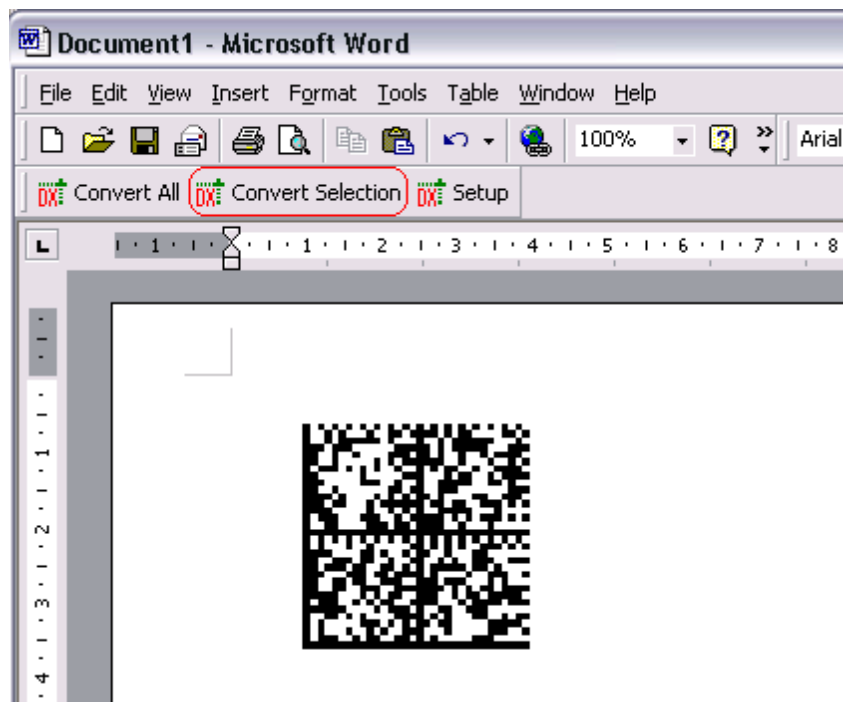


## 5.1.2 Create Single Barcode

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

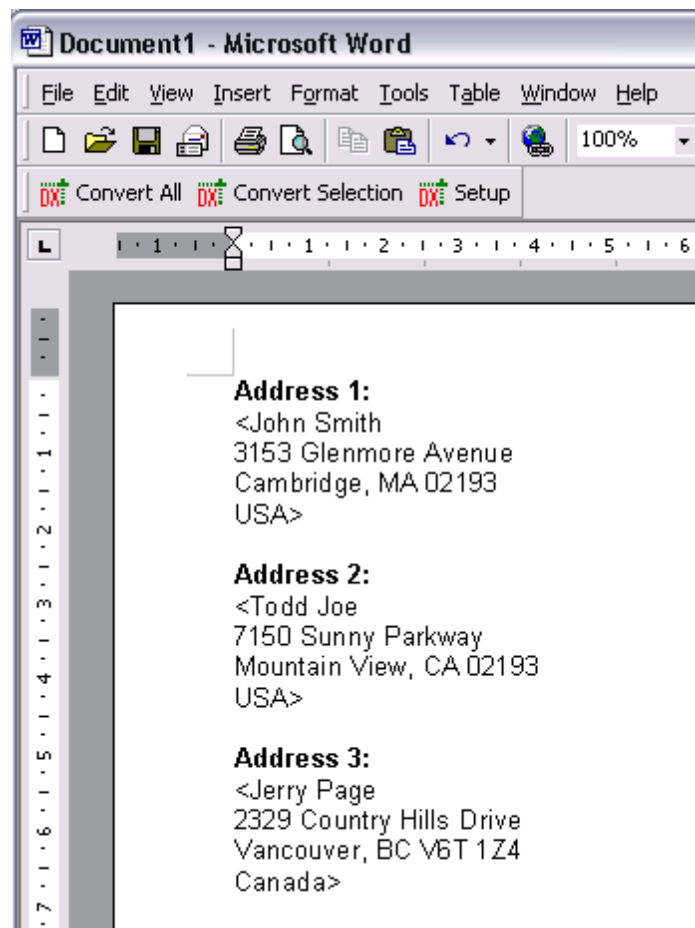


2. Click on "**Convert Selection**" to create a DataMatrix barcode.

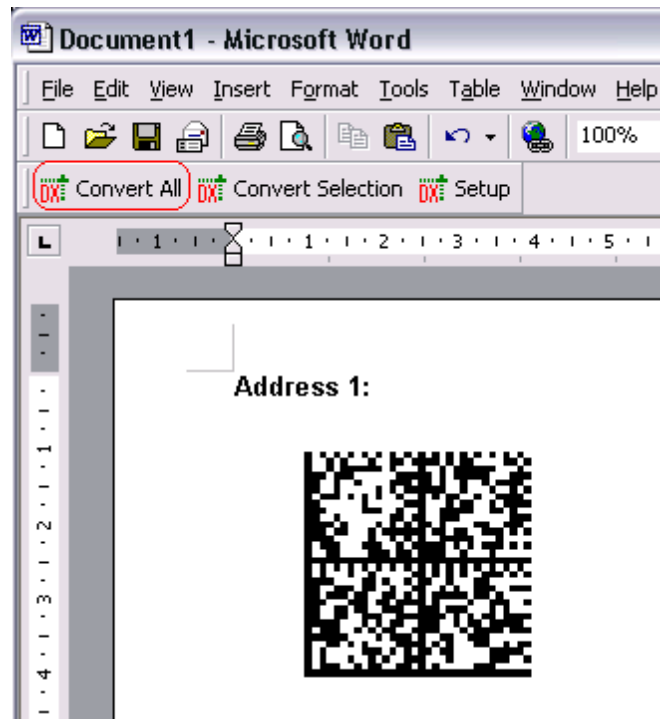


### 5.1.3 Create Multiple Barcodes

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

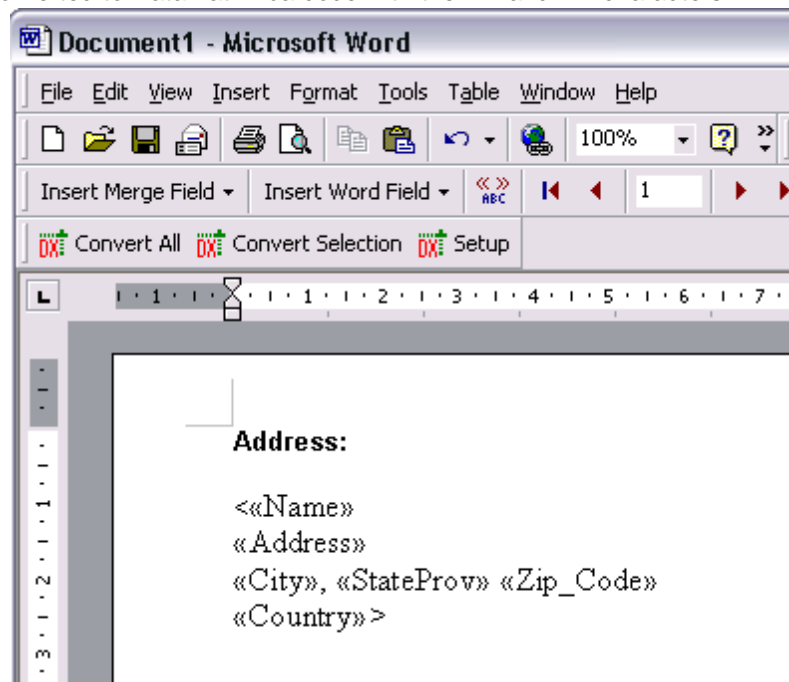


2. Click on "**Convert All**" to create DataMatrix barcodes for the paragraphs surrounded with the "<" and ">" characters.



### 5.1.4 Mail Merge

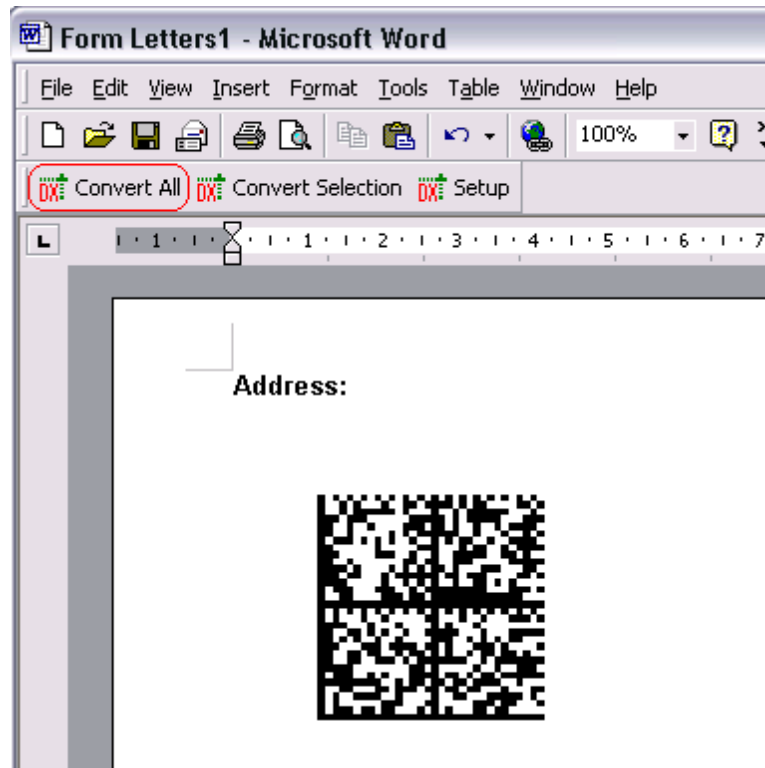
1. In Mail Merge, choose MW6\_DataMatrix\_ActiveX.mdb as the data source, surround the paragraphs which will be converted to DataMatrix barcode with the "<" and ">" characters.



2. Click on "Merge ..."



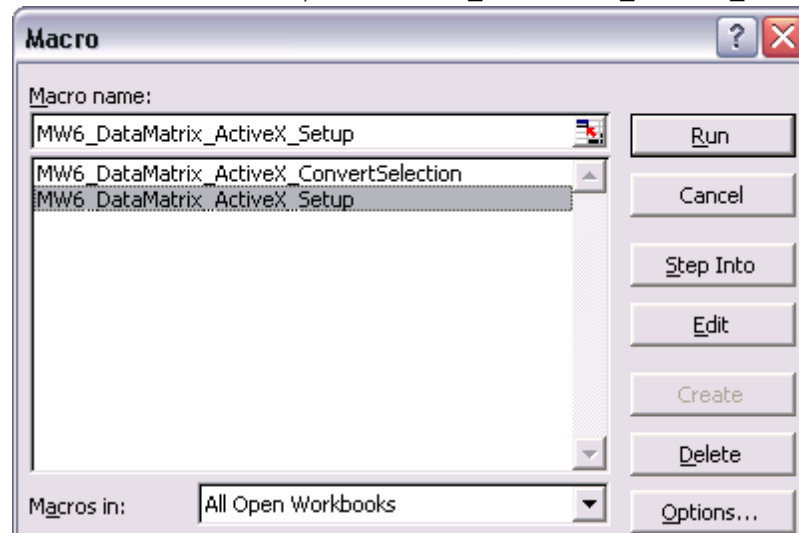
3. Click on "**Convert All**" to create DataMatrix barcodes for the paragraphs surrounded with the "<" and ">" characters.



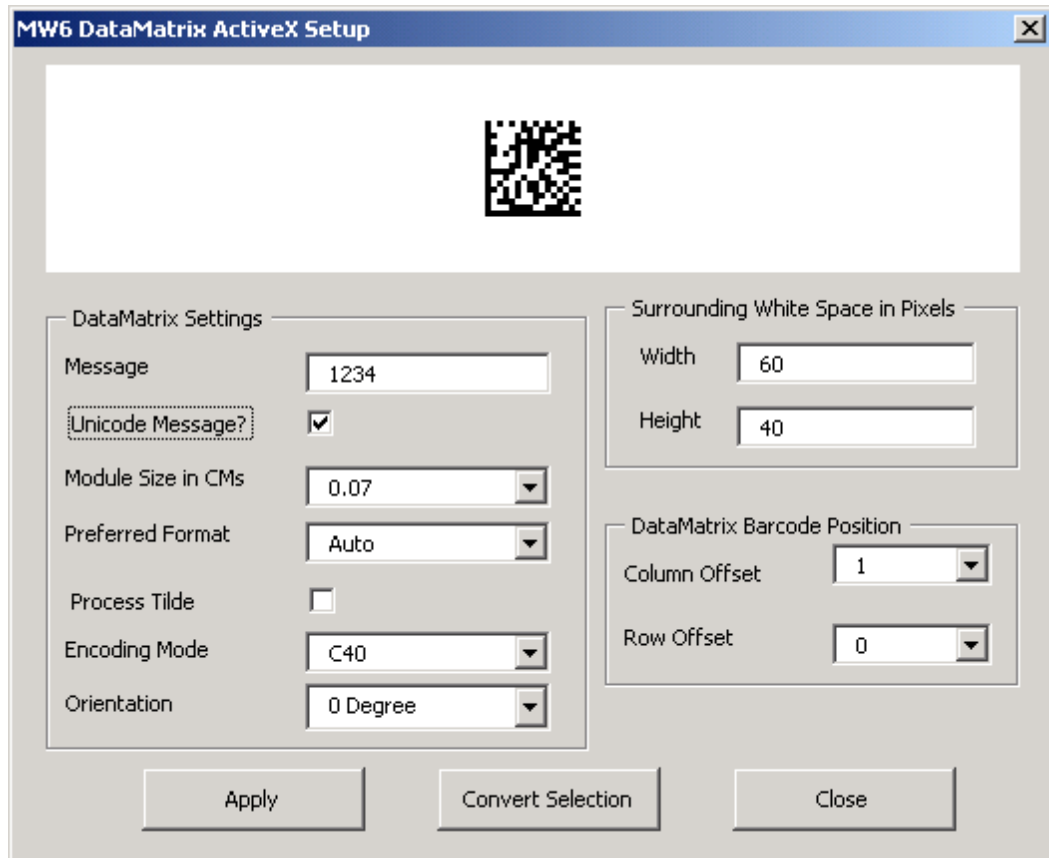
## 5.2 Excel

### 5.2.1 Change Settings

1. In Excel, open MW6\_DataMatrix\_ActiveX.XLS.
2. Click on "Tools" > "Macro" > "Macros", select "MW6\_DataMatrix\_ActiveX\_Setup".



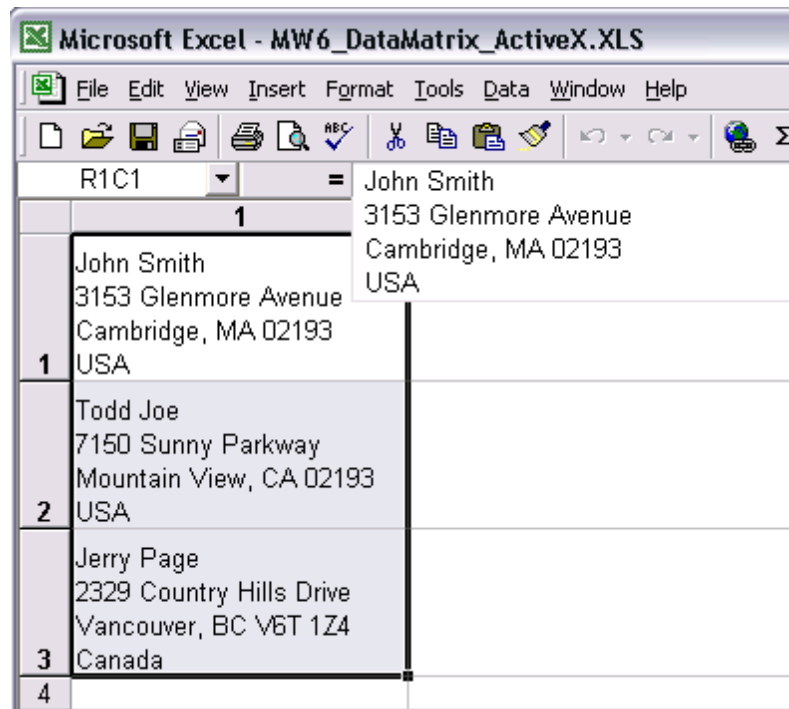
3. Click on "Run".



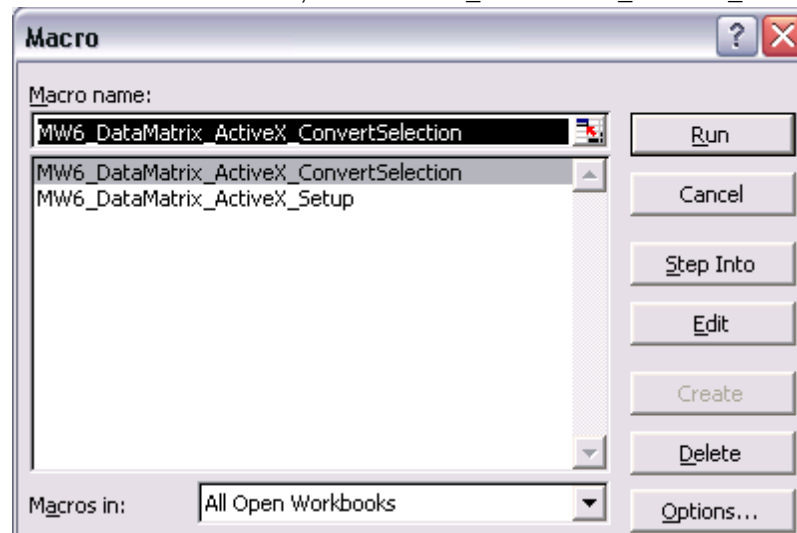
4. Choose a few appropriate values for DataMatrix configurations, click on "Apply" button to allow the changes to take effect, "Column Offset" and "Row Offset" are used to specify DataMatrix barcode position relative to the position of a cell which contains the regular string. If the string contains some Unicode texts (Japanese, Chinese, Korean, etc), toggle on "Unicode Message?" check box, so the VBA macro code can apply the special treatments to those Unicode characters.

## 5.2.2 Create Multiple Barcodes

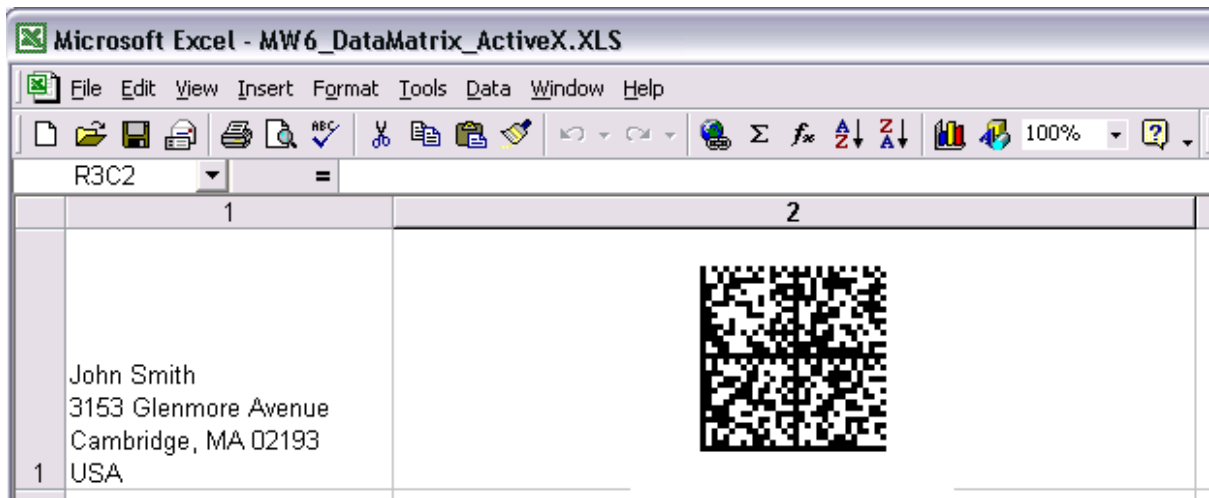
1. Select a few cells.



2. Click on "Tools" > "Macro" > "Macros", select "MW6\_DataMatrix\_ActiveX\_ConvertSelection".

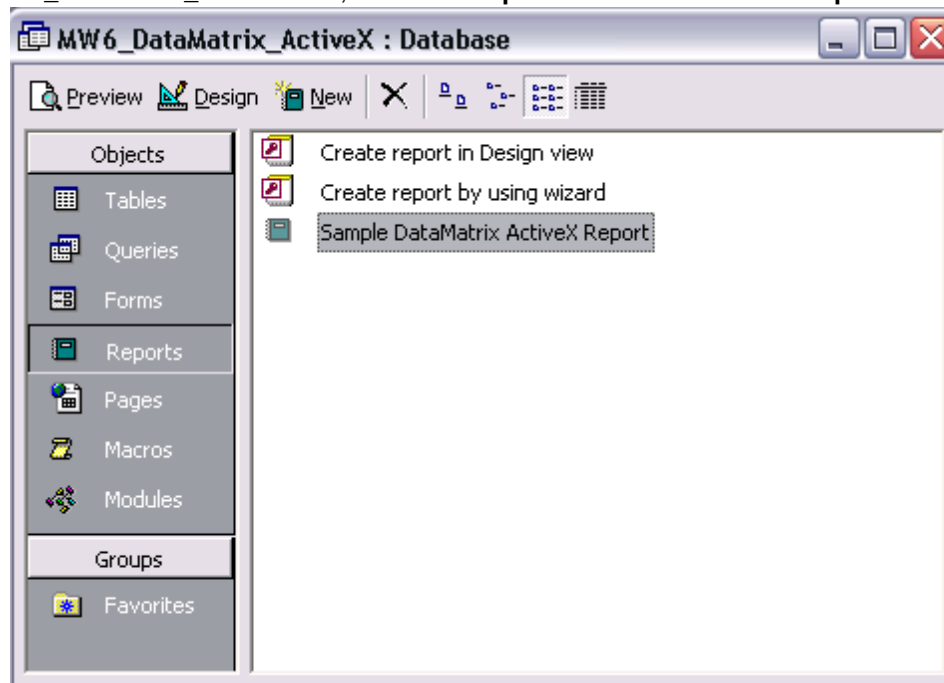


3. Click on "Run" to create DataMatrix barcodes for the selected cells.



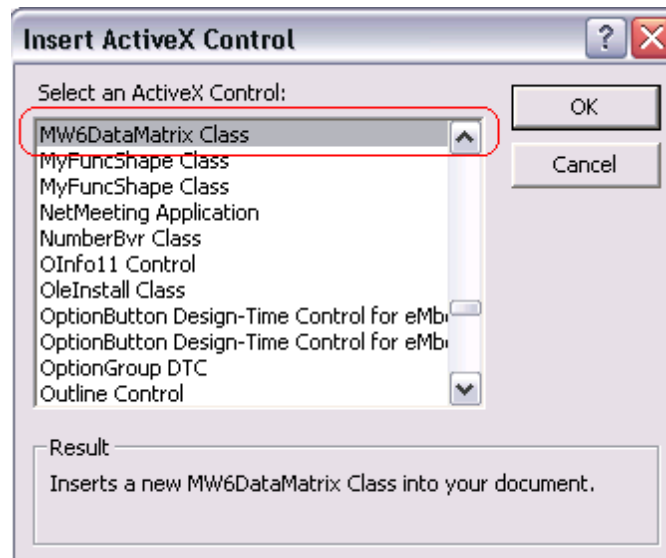
## 5.3 Access

1. Open MW6\_DataMatrix\_ActiveX.mdb, select "**Sample DataMatrix ActiveX Report**".

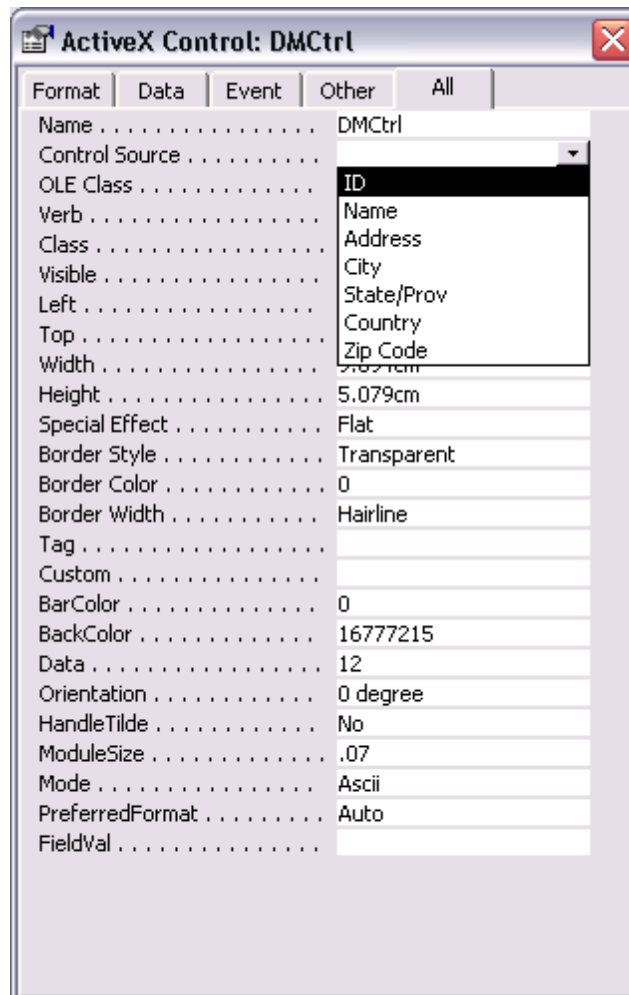


2. Click on "**Design**", insert a MW6 DataMatrix ActiveX control into the report.

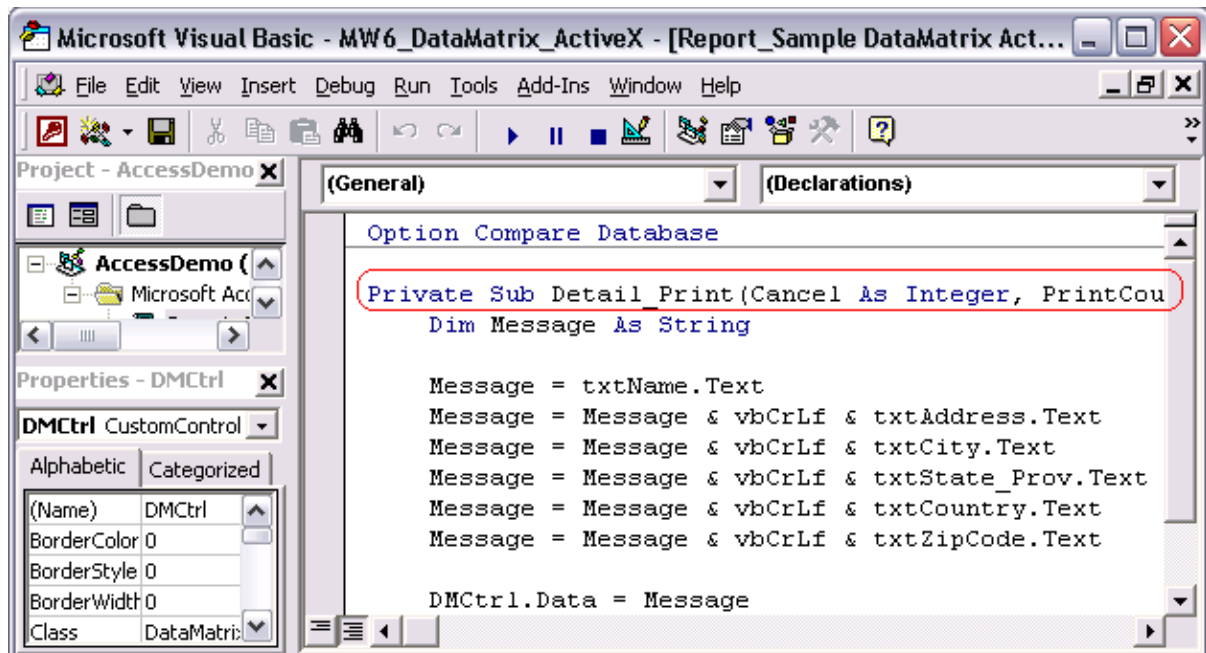




3. Change its properties to meet your application requirements, our DataMatrix ActiveX supports the data binding so you can bind a field in a database to the control and generate DataMatrix barcodes for each data record automatically, there's an arrow on the right side of the "**Control Source**" property, click on the arrow, a list opens with all fields, select the field you want for the control.



4. If you do not want to use the data binding feature, you can customize "*Private Sub Detail\_Print (Cancel As Integer, PrintCount As Integer)*" to create DataMatrix barcodes dynamically.



5. Click on "**Preview**" to view DataMatrix barcodes.

## 6 Reference Guide

### 6.1 Properties

#### 6.1.1 BackColor Property

Gets or sets the background color of the DataMatrix barcode.

##### Property Data Type

OLE\_COLOR

##### Remarks

The default value is white color.

#### 6.1.2 BarColor Property

Gets or sets the color of the DataMatrix barcode and text.

##### Property Data Type

OLE\_COLOR

##### Remarks

The default value is black color.

### 6.1.3 BorderStyle Property

Gets or sets the style of the border rectangle.

#### Property Data Type

short

#### Remarks

The default value is 0, this property can be one of the following values:

Value	Description
0	No Border
1	Dash Border
2	Solid Border

### 6.1.4 Data Property

Gets or sets the message to encode with DataMatrix barcode ActiveX.

#### Property Data Type

BSTR

#### Remarks

The default value is "12".

### 6.1.5 HandleTilde Property

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

#### Property Data Type

VARIANT\_BOOL

#### Remarks

If this property is set to TRUE, non-printable characters can be passed to DataMatrix ActiveX 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, "~10107612345678900~117100503" can be used to generate GS1 DataMatrix "(01)0107612345678900(17)100503", and "~110AC34563G3" can be used to generate GS1 DataMatrix "(10)AC34563G3".

"~5" is used to indicate Macro 5. For example, "~5ABCDEF[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, "~6ABCDEF[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.

### 6.1.6 ImageData Property

Gets WMF format data stream of the DataMatrix barcode.

#### Property Data Type

IPictureDisp\*

### 6.1.7 Mode Property

Gets or sets the encoding mode of the DataMatrix barcode.

#### Property Data Type

short

#### Remarks

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

### 6.1.8 ModuleSize Property

Gets or sets the size (width/height) of the square-shaped module.

#### Property Data Type

float

#### Remarks

The default value is 0.07, internally our DataMatrix ActiveX 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.

### 6.1.9 Orientation Property

Gets or sets the orientation of the DataMatrix barcode.

#### Property Data Type

short

#### Remarks

The default value is 0, this property can be one of the following values:

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

### 6.1.10 PreferredFormat Property

Gets or sets the format of the DataMatrix barcode.

#### Property Data Type

short

#### Remarks

The default value is 0, this property can be one of the following values.

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 ActiveX 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 ActiveX control will also automatically choose an appropriate format with bigger data capacity to encode the string.

## 6.2 Methods

### 6.2.1 CopyToClipboard Method

Copies the DataMatrix barcode image into the system clipboard.

```
void CopyToClipboard();
```

#### Remarks

Before you call this method, use *GetActualSize()* method to obtain the actual size of the DataMatrix barcode and use *SetSize()* method to set image size by adding surrounding white space around the DataMatrix barcode.

#### See Also

[GetActualSize\(\) Method](#) | [SetSize\(\) Method](#)

### 6.2.2 GetActualRC Method

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

```
void GetActualRC(short *ActualRows, short *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 ActiveX 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 ActiveX 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.

### 6.2.3 GetActualSize Method

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

```
void GetActualSize(VARIANT_BOOL ScreenIsTarget, long TargetHDC, long *ActualWidth, long *ActualHeight);
```

#### Parameters

*ScreenIsTarget*

Indicates whether 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).

---



## 6.2.4 GetPatternData Method

Gets the DataMatrix barcode pattern matrix data.

```
void GetPatternData(short *Buffer,  
                   long *Size,  
                   short *Rows,  
                   short *Columns,  
                   VARIANT_BOOL *Result);
```

### 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 method fails and the variable pointed to by *Size* returns the required buffer size, in 16-bit integers.

#### *Size*

[in/out] On input, specifies the size, in 16-bit integers, of the *Buffer*. On output, receives the size, in 16-bit integers, 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..

#### *Result*

If the method succeeds, the value of the variable pointed to by *Result* is VARIANT\_TRUE, otherwise the value is VARIANT\_FALSE.

### 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, only *Data*, *HandleTilde*, *Mode* and *PreferredFormat* properties affect the pattern matrix data output.

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

## 6.2.5 Render Method

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

```
void Render(long hDC, int x, int 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.

**6.2.6 SaveAsBMP Method**

Saves the DataMatrix barcode image as a BMP file.

```
void SaveAsBMP(BSTR FileName);
```

**Parameters***FileName*

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

**Remarks**

Before you call this method, use `GetActualSize()` method to obtain the actual size of the DataMatrix barcode and use `SetSize()` method to set image size by adding surrounding white space around the DataMatrix barcode.

**See Also**

[GetActualSize\(\) Method](#) | [SetSize\(\) Method](#)

**6.2.7 SaveAsWMF Method**

Saves the DataMatrix barcode image as a WMF file.

```
void SaveAsWMF(BSTR FileName);
```

**Parameters***FileName*

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

**Remarks**

Before you call this method, use `GetActualSize()` method to obtain the actual size of the DataMatrix barcode and use `SetSize()` method to set image size by adding surrounding white space around the DataMatrix barcode.

---

**See Also**

GetActualSize() Method | SetSize() Method

**6.2.8 SetSize Method**

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

```
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

**6.2.9 SetStructuredAppend Method**

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

```
void SetStructuredAppend(VARIANT_BOOL AllowSA, short SymbolID, short 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.

## 7 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.

\* 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.

---