

Table of Contents

Foreword	0
Part I Introduction	3
Part II Installation	3
1 Trial Version.....	3
2 Full Version.....	3
Part III How to Distribute It	3
Part IV Office 2007	3
1 Word	3
Install Template File	3
Create Single Barcode	4
Create Multiple Barcodes	5
Mail Merge	6
2 Excel	9
Change Settings	9
Create Multiple Barcodes	11
Part V Office 2000 & 2003	12
1 Word.....	12
Install Template File	12
Create Single Barcode	14
Create Multiple Barcodes	14
Mail Merge	16
2 Excel.....	17
Change Settings	17
Create Multiple Barcodes	19
Part VI Reference Guide	20
1 PDF417Appearance Function.....	20
2 PDF417Configure Function.....	21
3 PDF417CopyToClipboard Function.....	22
4 PDF417GetActualSize Function.....	22
5 PDF417GetPatternData Function.....	23
6 PDF417Render Function.....	24
7 PDF417SaveAsBMP Function.....	24
8 PDF417SaveAsWMF Function.....	25
9 PDF417SetBackColor Function.....	25
10 PDF417SetBarColor Function.....	26
11 PDF417SetDefault Function.....	26

12	PDF417SetMessage Function.....	26
13	PDF417SetSize Function.....	26
	Part VII Convert Size from CMs to Pixels	27
	Part VIII License	27
	Index	0

1 Introduction

MW6 PDF417 Win32 DLL can create device independent 2D PDF417 images for your application, you can save the PDF417 as either BMP or WMF image file or copy PDF417 WMF image to the clipboard.

PDF417 is a multi-row, variable-length symbology offering high data capacity and error-correction capability, it is capable of encoding 1100 bytes, 1800 ASCII characters, or 2700 digits.

Every PDF417 symbol is composed of a stack of rows, from a minimum of 3 to a maximum of 90 rows, a PDF417 symbol character consists 17 modules arranged into 4 bars and 4 spaces.

2 Installation

2.1 Trial Version

1. UnZip MW6PDF417Win32.ZIP, run the setup.exe to install PDF417 Win32 DLL.
2. The trial version PDF417 Win32 DLL appends "MW6 Demo" to the string encoded with PDF417 format.

2.2 Full Version

1. Uninstall the trial version PDF417 Win32 DLL if applicable.
2. UnZip full version PDF417 Win32 DLL .zip file and run the setup.exe to install the full version PDF417 Win32 DLL.

3 How to Distribute It

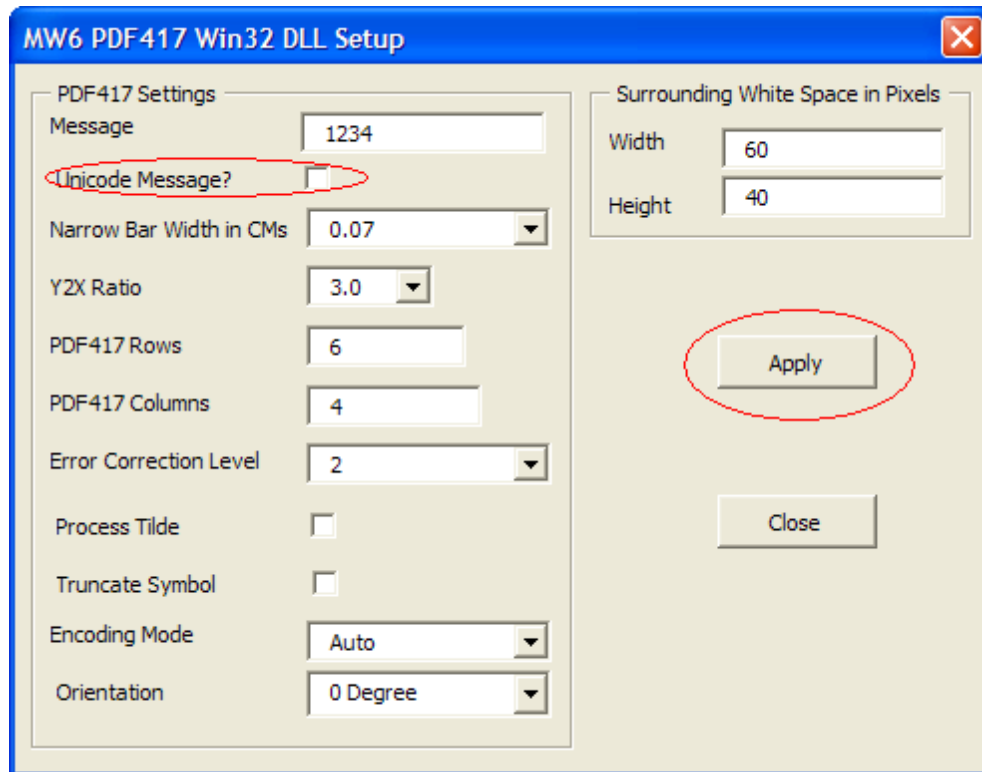
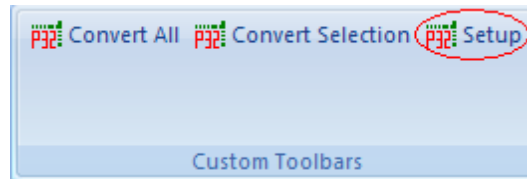
If you want to redistribute PDF417 Win32 DLL as part of your application, on the target machine, simply put **PDF417Win32.dll** into the windows 32-bit system folder (e.g. "c:\windows\system32" or "c:\winnt\system32") for 32-bit Windows OS, or the SysWow64 folder (e.g. "c:\windows\SysWow64") for 64-bit Windows OS.

4 Office 2007

4.1 Word

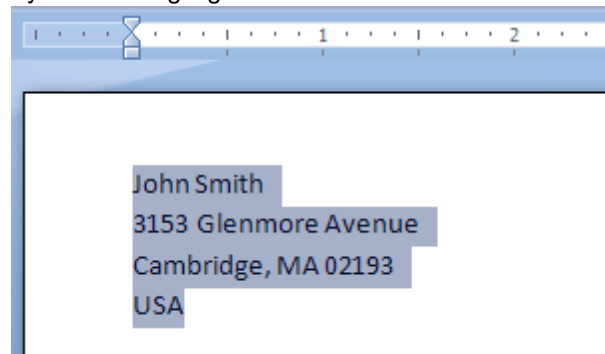
4.1.1 Install Template File

1. Locate the Office Word STARTUP folder, which usually is "C:\Documents and Settings\\Application Data\Microsoft\Word\STARTUP".
 2. Copy MW6_PDF417_Win32.dotm to this folder.
 3. Click on "**Add-Ins**", then click on "**Setup**", change the configurations for PDF417 format, 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.
-

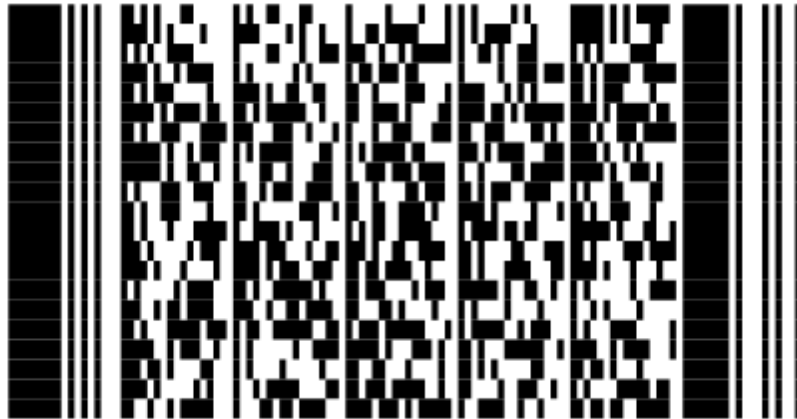


4.1.2 Create Single Barcode

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

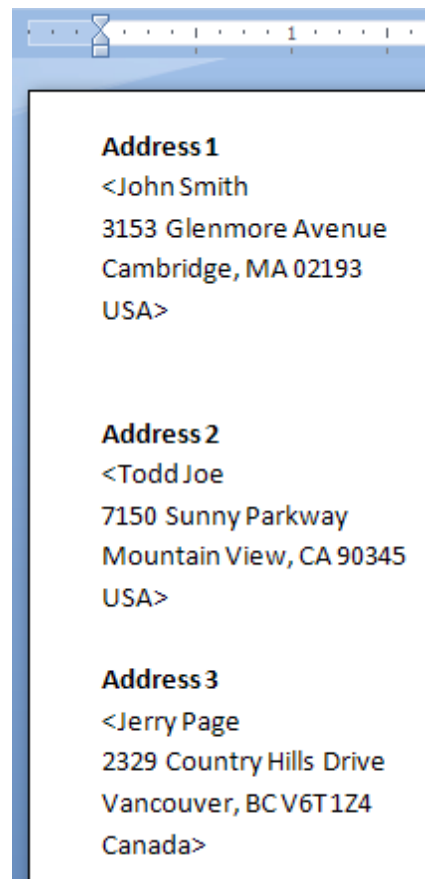


2. Click on "**Add-Ins**", then click on "**Convert Selection**" to create a PDF417 barcode.

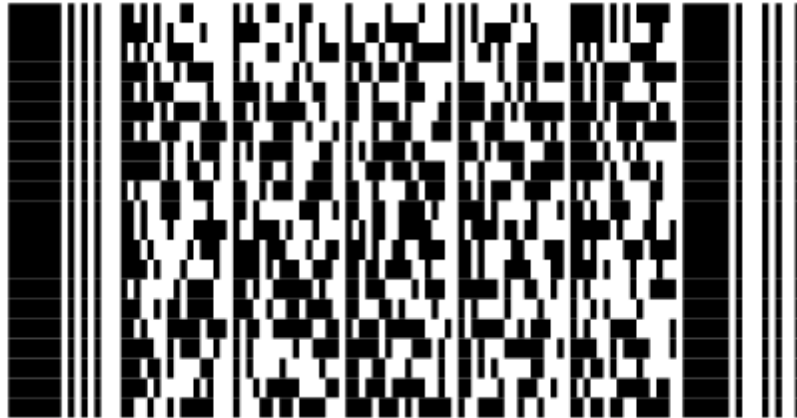


4.1.3 Create Multiple Barcodes

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

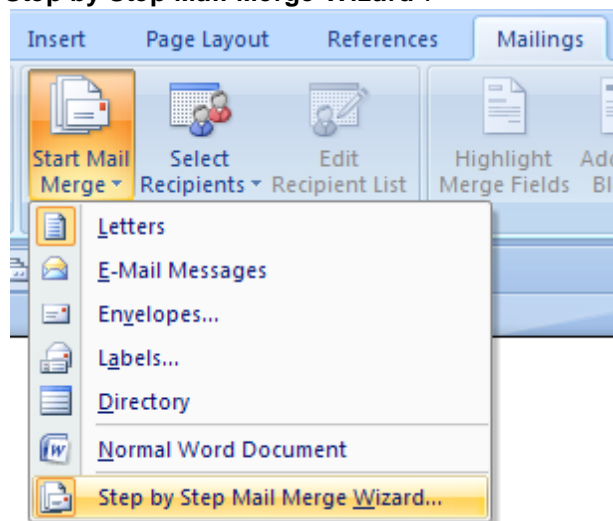


2. Click on "**Add-Ins**", then click on "**Convert All**" to create PDF417 barcodes for the string sections surrounded with the "<" and ">" characters.

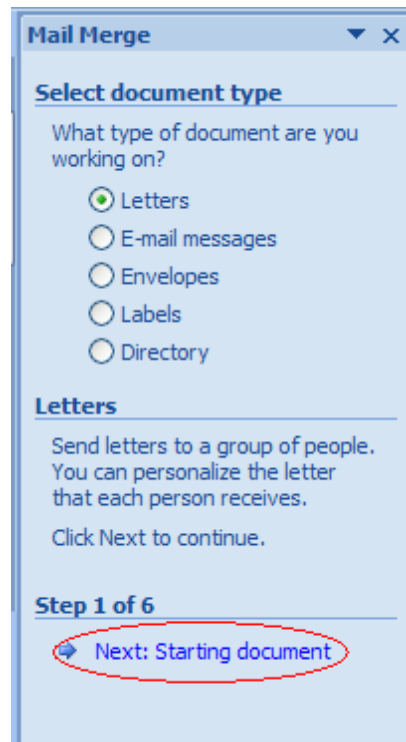


4.1.4 Mail Merge

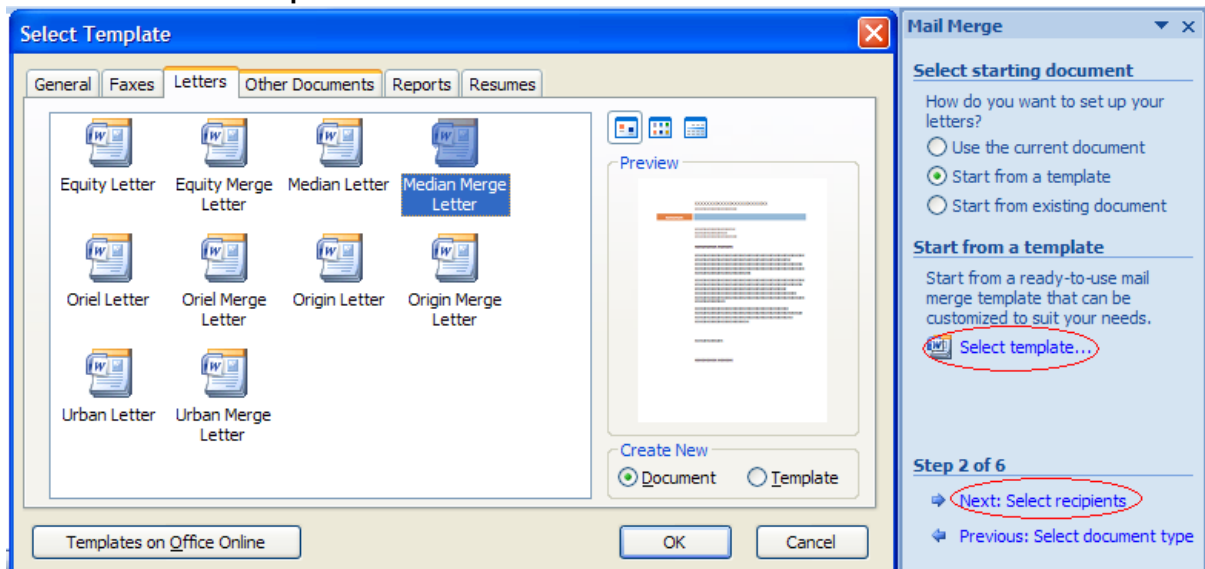
1. Click on "**Mailings**", then click on "**Start Mail Merge**". A drop-down list appears as shown below, select the last option "**Step by Step Mail Merge Wizard**".



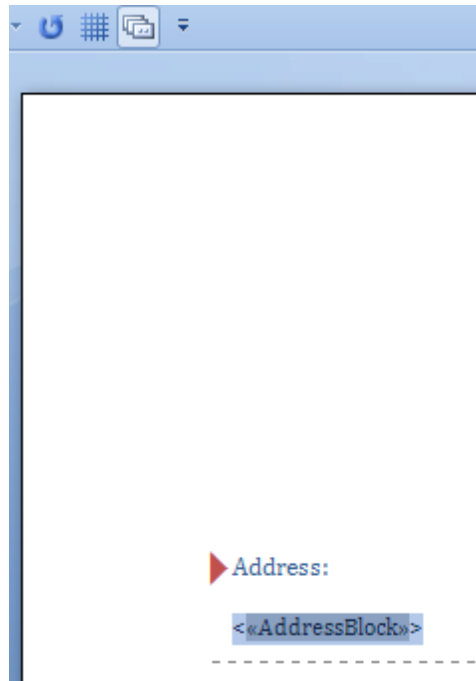
2. Select a document type and click on "**Next: Starting document**".
-



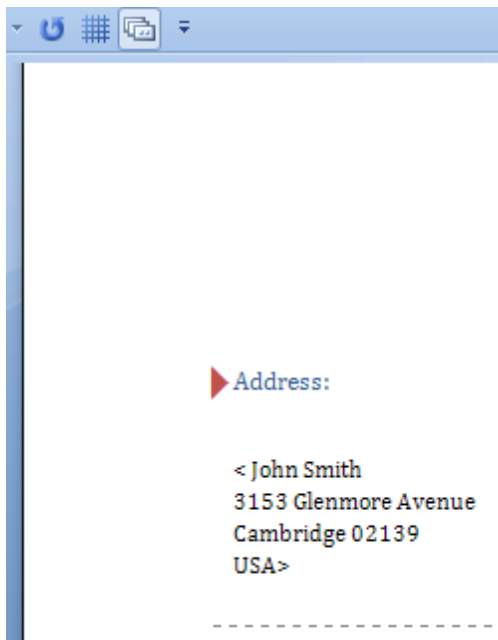
3. Click on "**Start from a template**", then click on the link "**Select template**", choose a template, click on "**Next: Select recipients**".



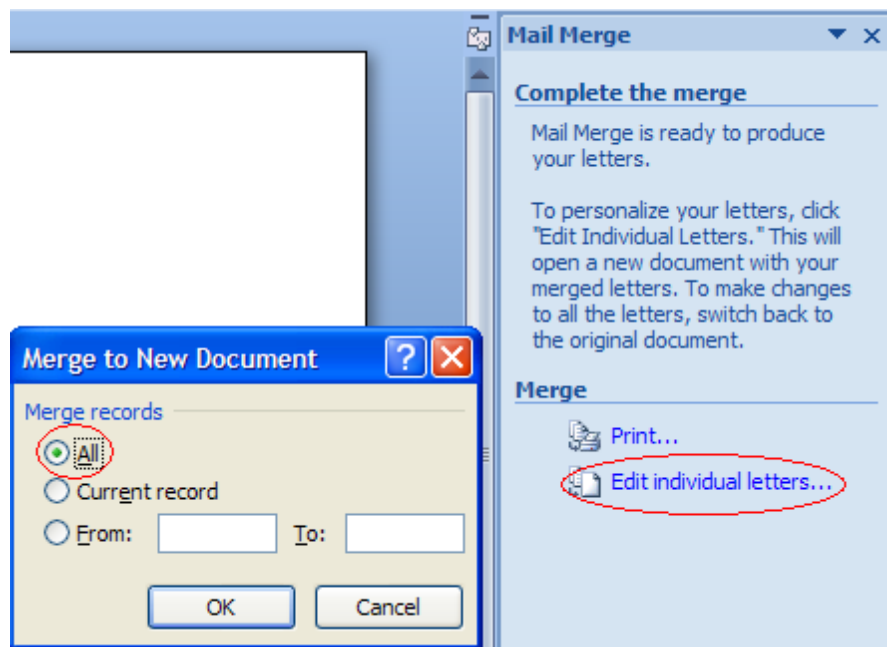
4. Select "**Use an existing list**" and click on "**Browser**" link, choose a database as an existing list, click "**Next: Write your letter**".
5. Surround the section which will be converted to the PDF417 barcode with the "<" and ">" characters and highlight it.



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.



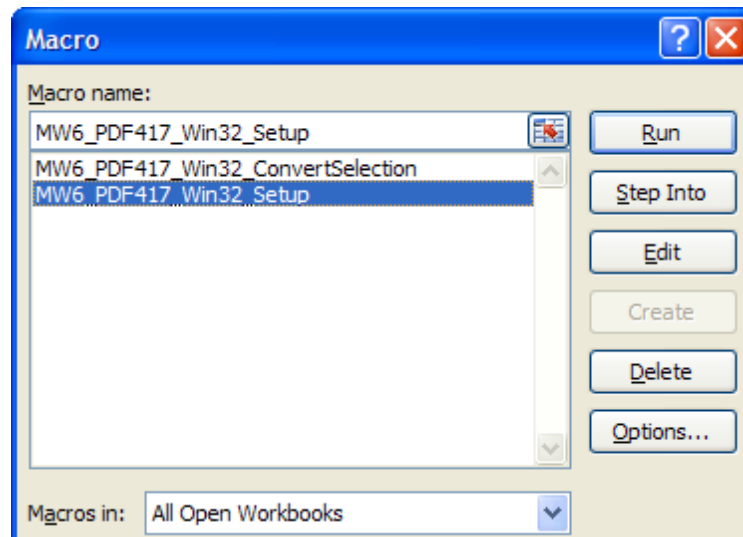
8. Click on "Add-Ins", then click on "Convert All" to create PDF417 barcodes.



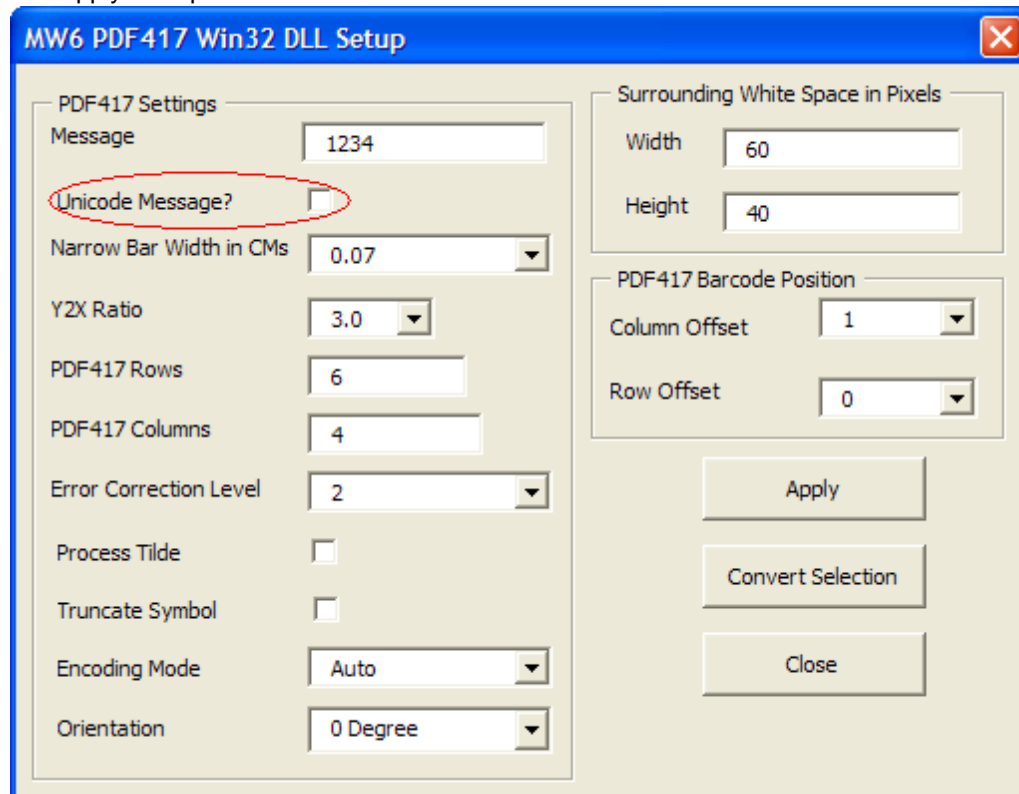
4.2 Excel

4.2.1 Change Settings

1. In Excel, open MW6_PDF417_Win32.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_PDF417_Win32_Setup".



4. Choose a few appropriate values for PDF417 configurations, click on "**Apply**" button to allow the changes to take effect, "**Column Offset**" and "**Row Offset**" are used to specify the barcode position relative to the position of 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.

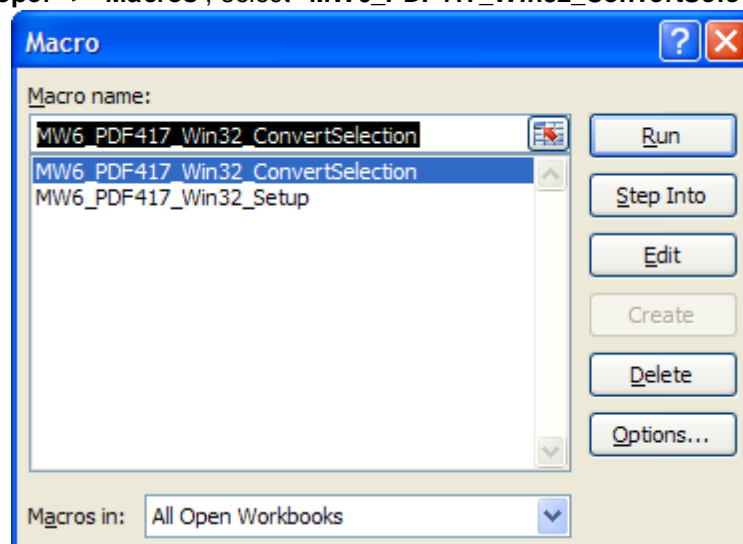


4.2.2 Create Multiple Barcodes

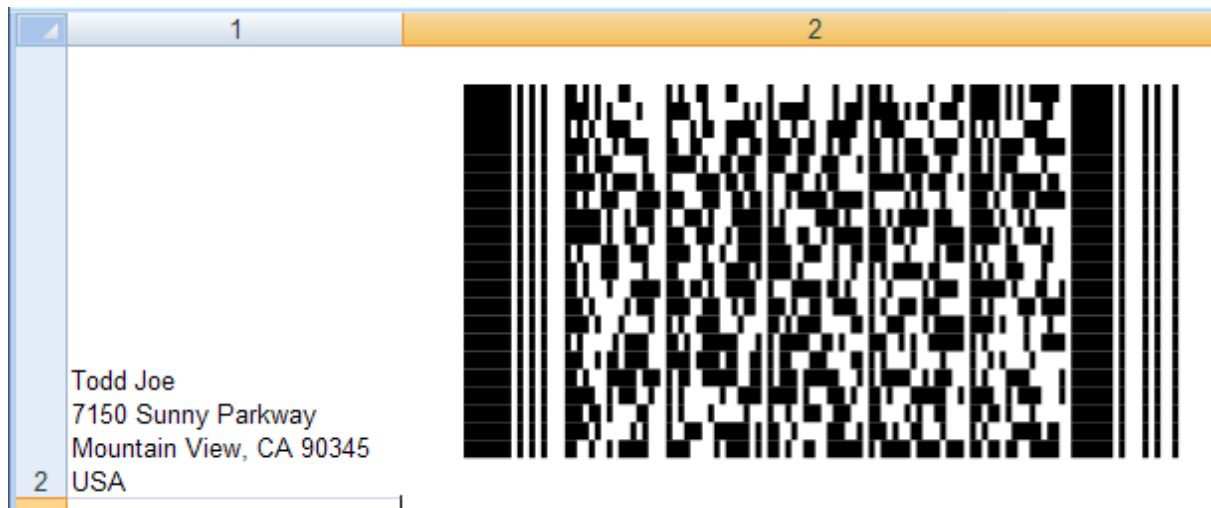
1. Select a few cells.



2. Click on "Developer" > "Macros", select "MW6_PDF417_Win32_ConvertSelection".



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



5 Office 2000 & 2003

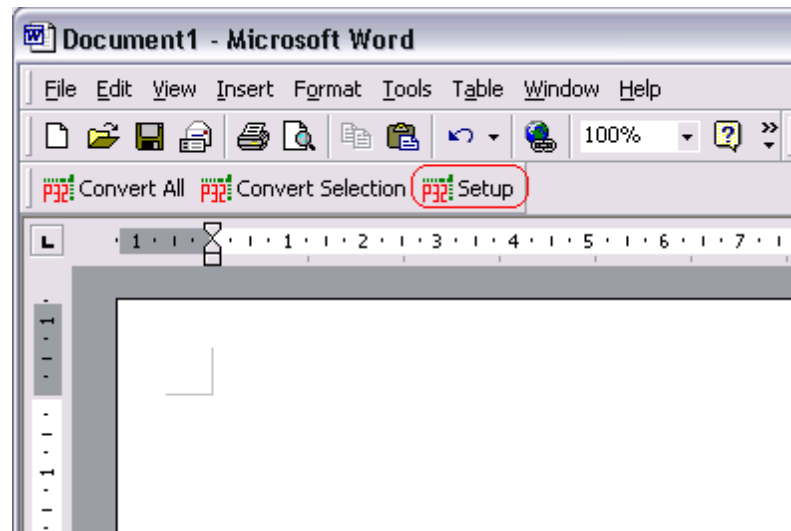
5.1 Word

5.1.1 Install Template File

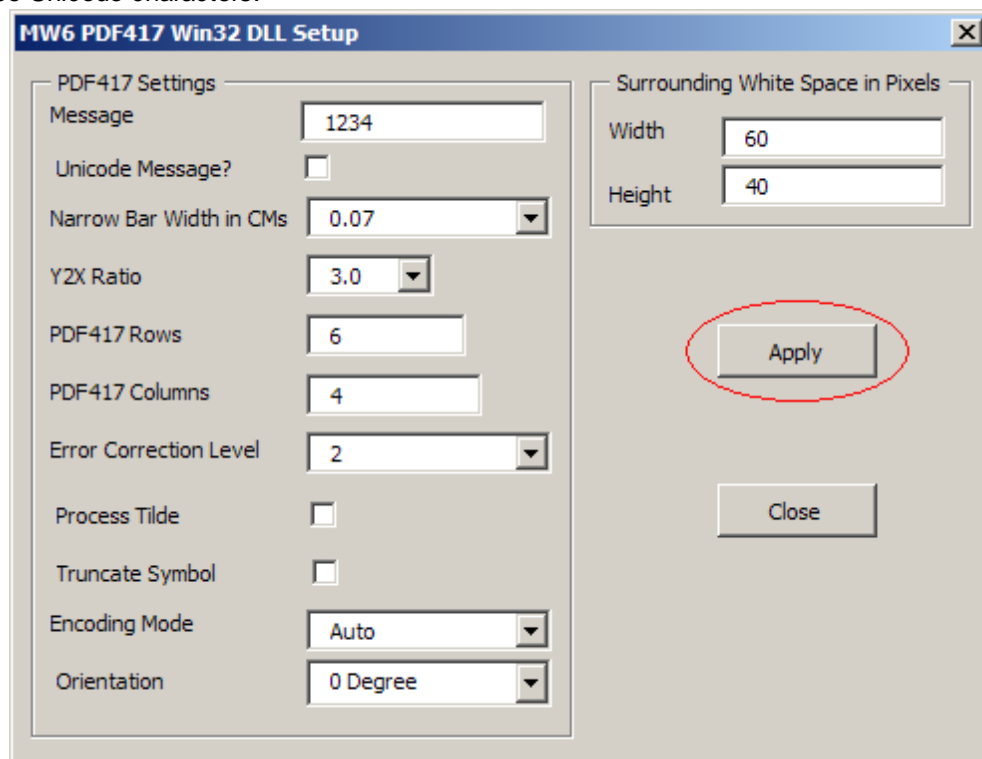
1. Locate the Word Startup folder, the Startup folder can be found in the following locations:

OS	Location
Windows 2000/XP	"C:\Documents and Settings\ <user name="">\Application Data\Microsoft\Word\Startup"</user>
Windows NT4	"C:\Winnt\Profiles\ <user name="">\Application Data\Microsoft\Word\Startup"</user>
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_PDF417_Win32.dot, which usually is in the folder "c:\Program Files\MW6 Win32 DLL \PDF417", to the Word Startup folder.
3. Open up Word, click on "**Setup**".

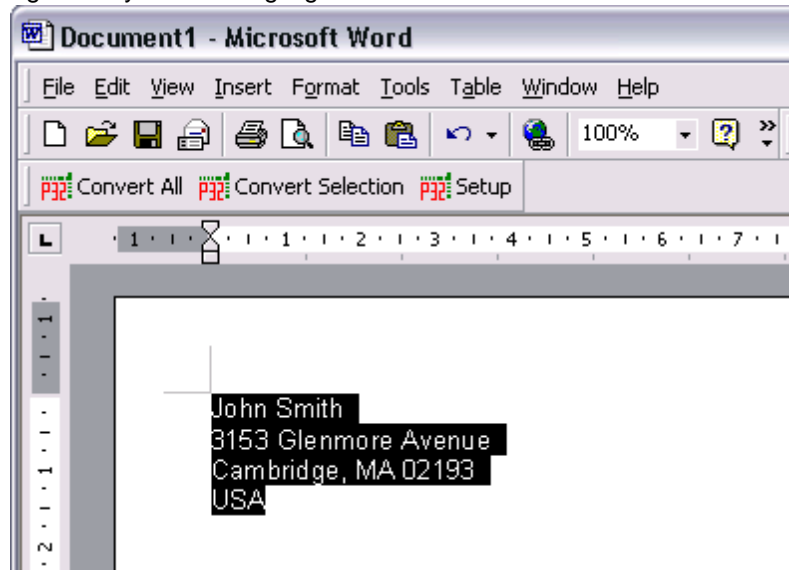


4. Choose a few appropriate values for PDF417 configurations, click on "**Apply**" button to allow the changes to take effect. If the string contains some Unicode texts (Japanese, Chinese, Korean, etc), toggle on "Unicode Message" check box, so the VBA macro code can apply the special treatments to those Unicode characters.

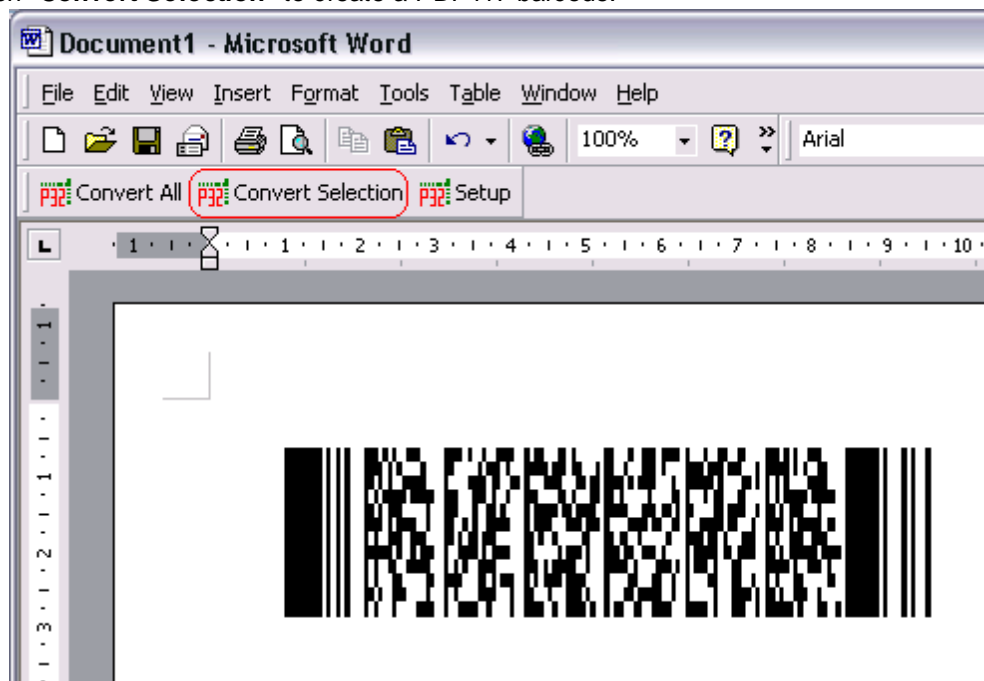


5.1.2 Create Single Barcode

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

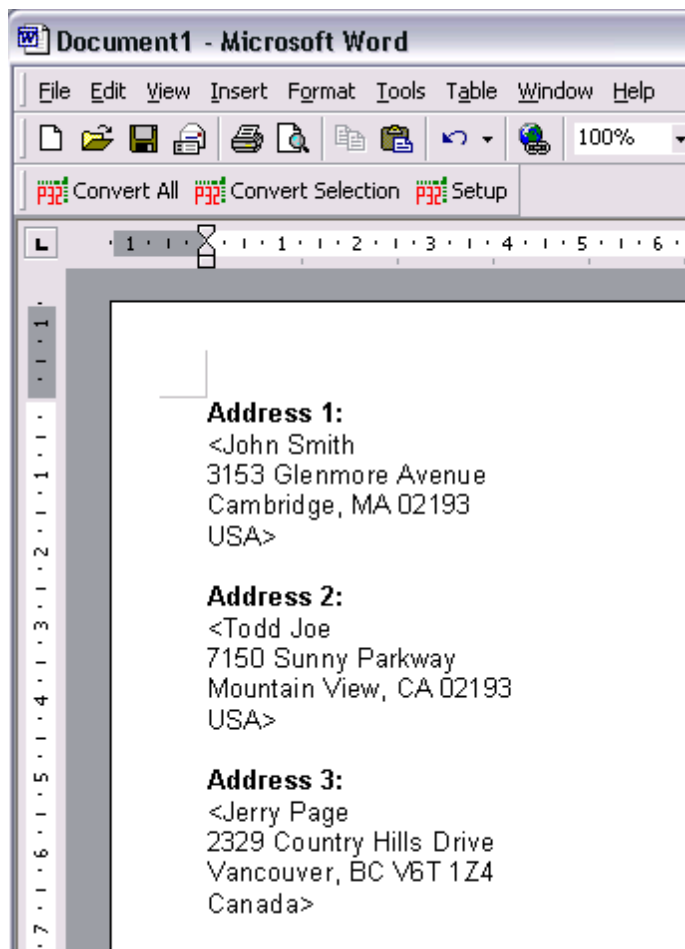


2. Click on "Convert Selection" to create a PDF417 barcode.

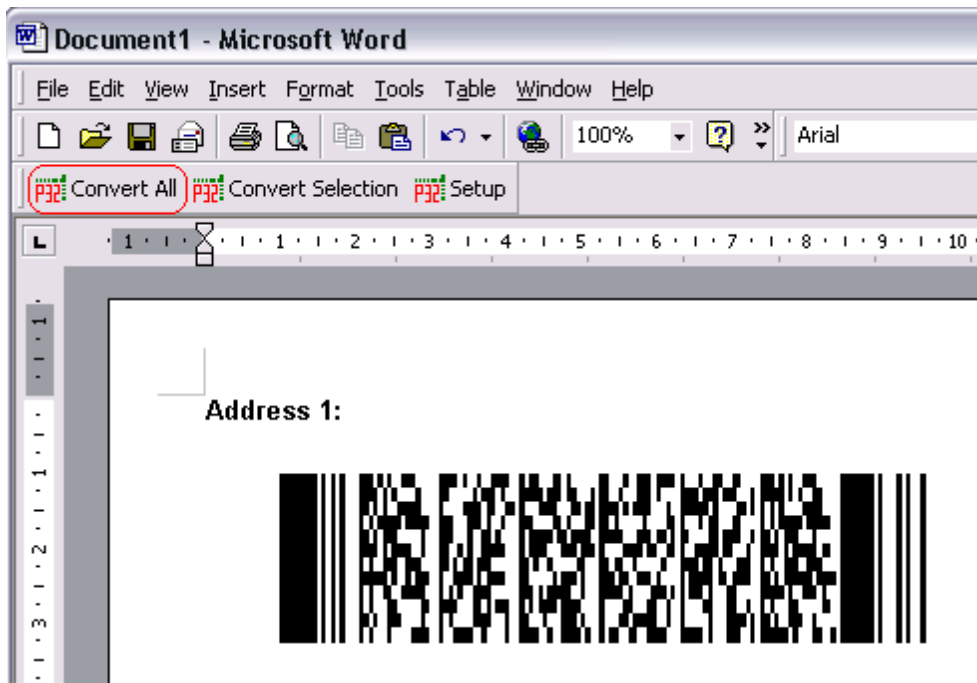


5.1.3 Create Multiple Barcodes

1. Enter a few paragraphs, surround those paragraphs which will be converted to PDF417 barcodes with the "<" and ">" characters.

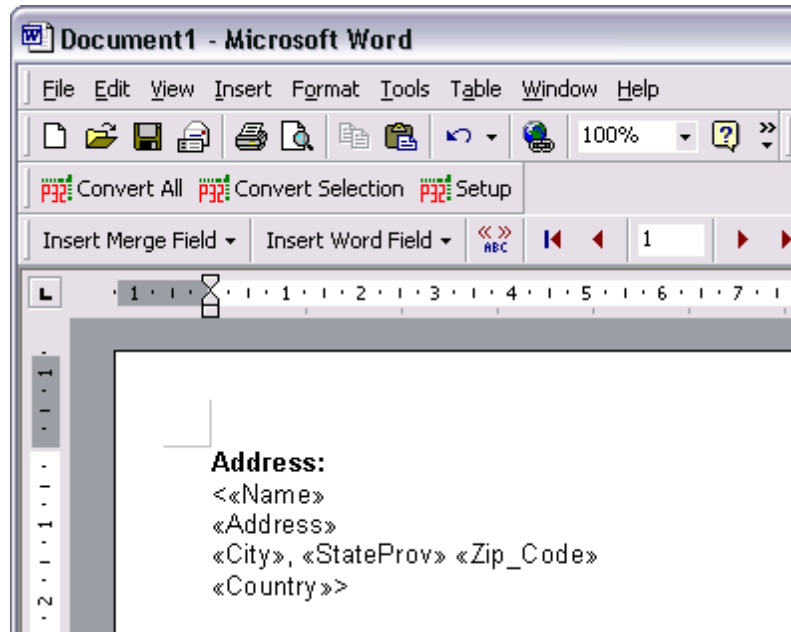


2. Click on "**Convert All**" to create PDF417 barcodes for the paragraphs surrounded with the "<" and ">" characters.



5.1.4 Mail Merge

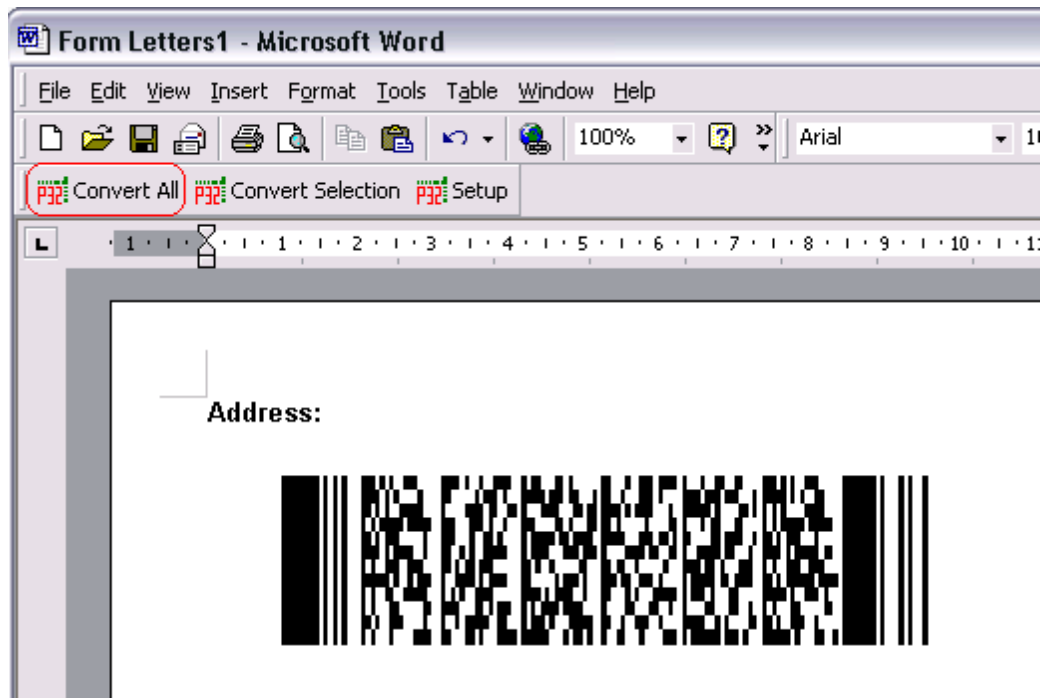
1. In Mail Merge, surround the paragraphs which will be converted to PDF417 barcodes with the "<" and ">" characters.



2. Click on "Merge ..."



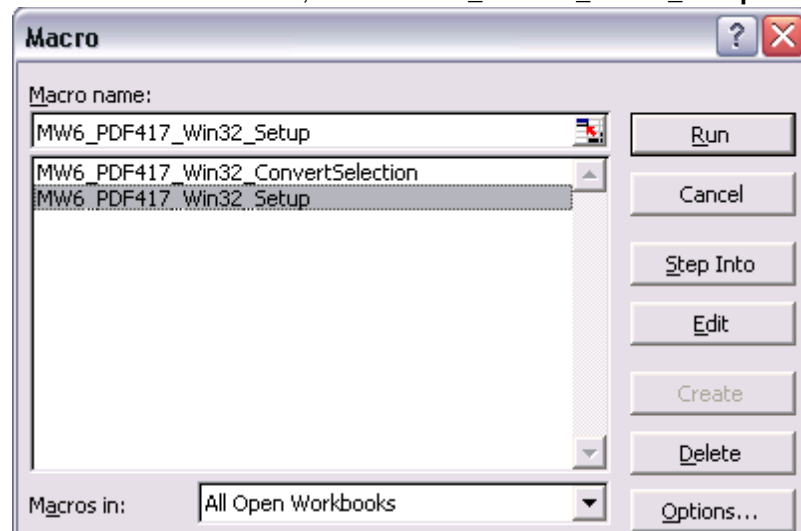
3. Click on "Convert All" to create PDF417 barcodes for the paragraphs surrounded with the "<" and ">" characters.



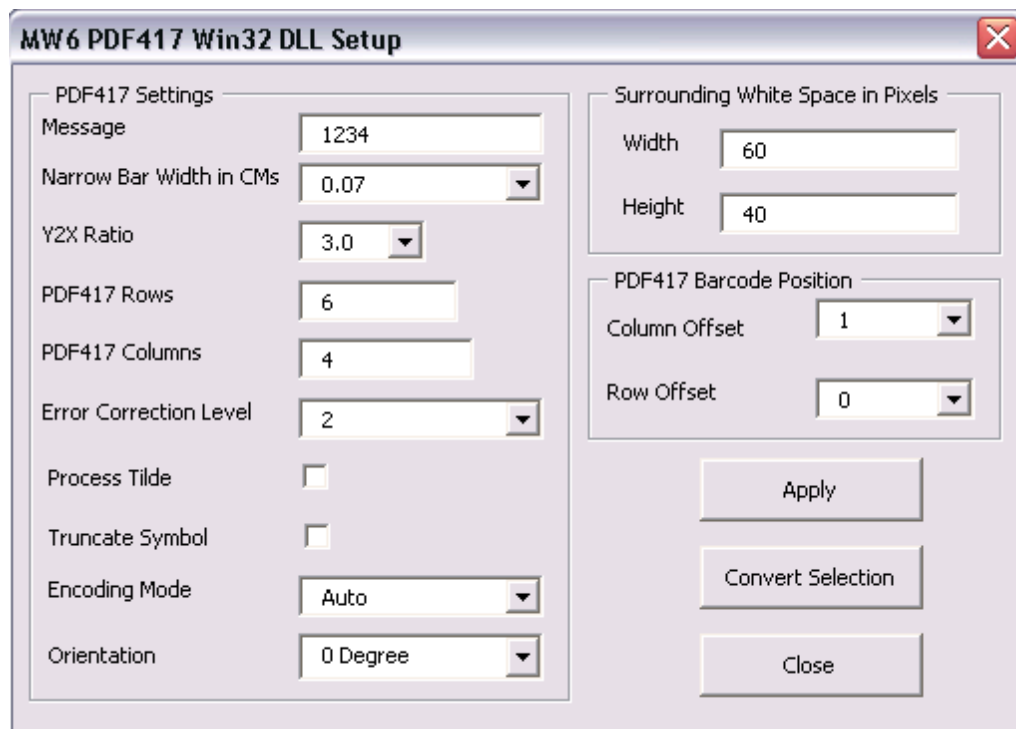
5.2 Excel

5.2.1 Change Settings

1. In Excel, open MW6_PDF417_Win32.XLS.
2. Click on "Tools" > "Macro" > "Macros", select "MW6_PDF417_Win32_Setup".



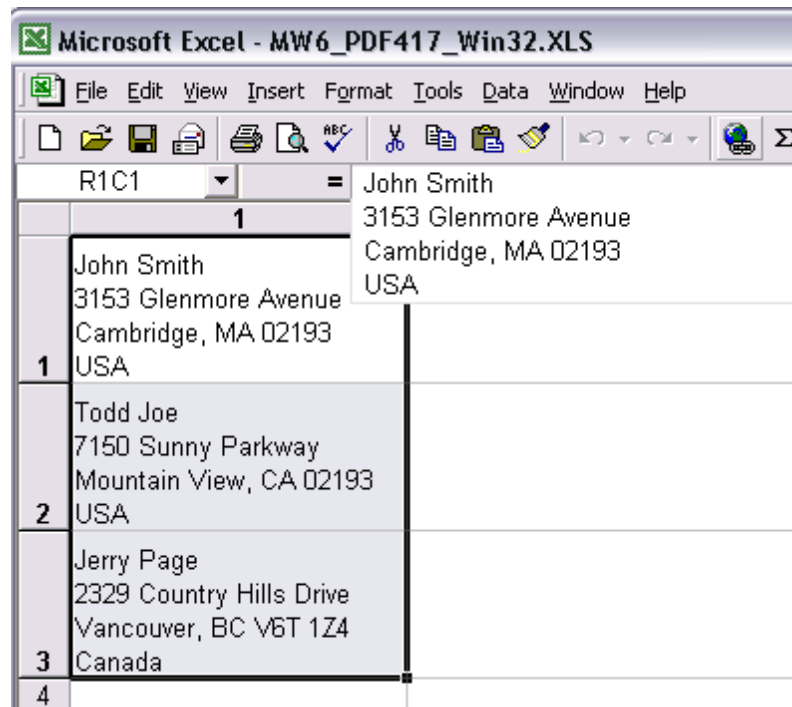
3. Click on "Run".



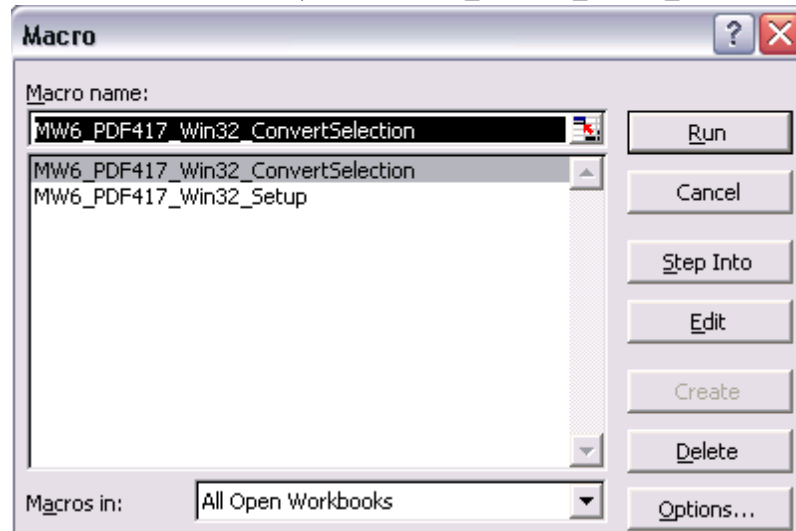
4. Choose a few appropriate values for PDF417 configurations, click on "**Apply**" button to allow the changes to take effect, "Column Offset" and "Row Offset" are used to specify PDF417 barcode position relative to the position of the cell which contains the regular string. If the string contains some 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

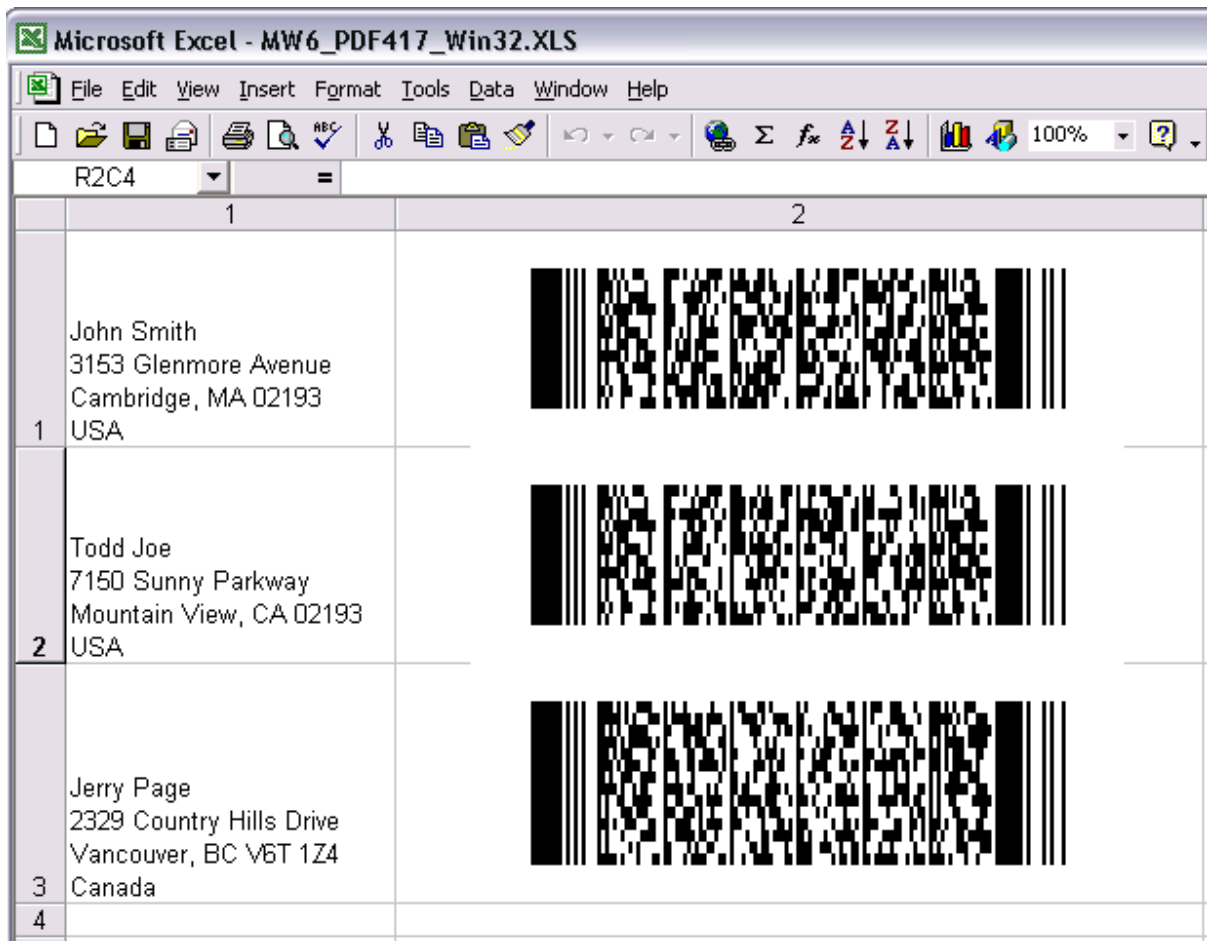
1. Select a few cells.



2. Click on "Tools" > "Macro" > "Macros", select "MW6_PDF417_Win32_ConvertSelection".



3. Click on "Run" to create PDF417 barcodes for the selected cells.



6 Reference Guide

6.1 PDF417Appearance Function

Sets up the parameters of PDF417 barcode appearance.

```
void PDF417Appearance(double NarrowBarWidth,
                     double Y2XRatio,
                     WORD Orientation,
                     WORD BorderStyle);
```

Parameters

NarrowBarWidth

Specifies the width, in centimeters, of the narrow bar element, please refer to this note for more information.

Y2XRatio

Specifies the ratio of the height of the row to the width of the narrow element.

Orientation

Specifies the orientation of the PDF417 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 PDF417Configure Function

Sets up the parameters of PDF417 barcode.

```
void PDF417Configure(WORD Rows,  
                    WORD Columns,  
                    WORD Mode,  
                    WORD ErrorCorrectionLevel,  
                    BOOL TruncateSymbol,  
                    BOOL HandleTilde);
```

Parameters

Rows

Indicates how many rows are used to encode the string, the valid value should be between 3 and 90.

Columns

Indicates how many columns are used to encode the string, the valid value should be between 3 and 30.

Mode

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

Value	Comment
0	Binary encoding mode
1	Text encoding mode
2	Auto mode for optimized encoding

ErrorCorrectionLevel

Indicates which error correction level is used, the valid value should be between 0 and 8, the value of 2 is recommended.

TruncateSymbol

Indicates whether to truncate the right side of the symbol or not, the recommended value is FALSE.

HandleTilde

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

6.3 PDF417CopyToClipboard Function

Copies the PDF417 barcode WMF format image into the system clipboard.

```
BOOL PDF417CopyToClipboard();
```

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 PDF417GetActualSize() function to obtain the actual size of the PDF417 barcode and use PDF417SetSize() function to set the image size by adding surrounding white space around the PDF417 barcode.

See Also

PDF417GetActualSize() Function | PDF417SetSize() Function

6.4 PDF417GetActualSize Function

Gets the actual size of the PDF417 barcode which is rendered onto either computer screen or other devices such as printers.

```
void PDF417GetActualSize(BOOL ScreenIsTarget,  
                        DWORD TargetHDC,  
                        DWORD *ActualWidth,  
                        DWORD *ActualHeight);
```

Parameters

ScreenIsTarget

Indicates whether the PDF417 barcode is rendered onto computer screen or not.

TargetHDC

Device context on which to render the PDF417 barcode, if the parameter *ScreensIsTarget* is set to TRUE, set this parameter to NULL.

ActualWidth

A pointer to the variable that receives the width of the PDF417 barcode (in pixels).

ActualHeight

A pointer to the variable that receives the height of the PDF417 barcode (in pixels).

6.5 PDF417GetPatternData Function

Gets the PDF417 barcode pattern matrix data.

```
BOOL PDF417GetPatternData(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 PDF417 barcode pattern matrix data row by row from the top left matrix corner, '1' indicates the narrow bar and '0' indicates the narrow space.

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 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 function to obtain the PDF417 barcode pattern matrix data and render the PDF417 barcode onto any device such as the printer, call *PDF417SetMessage()* and *PDF417Configure()* functions before calling this function, other functions don't affect the output of PDF417 barcode pattern

matrix.

Based on the *Orientation* parameter value, rotate the pattern matrix accordingly before you render the PDF417 barcode onto a device.

If the *Orientation* parameter value is 0 (no rotation) or 2 (180-degree rotation), the ratio of the height to the width for the rectangles (the narrow bar and the narrow space) must be equal to the *Y2XRatio* parameter value, the width of the rectangles is specified by the *NarrowBarWidth* parameter.

If the *Orientation* parameter value is 1 (90-degree rotation) or 3 (270-degree rotation), the ratio of the width to the height for the rectangles (the narrow bar and the narrow space) must be equal to the *Y2XRatio* parameter value, the height of the rectangles is specified by the *NarrowBarWidth* parameter.

6.6 PDF417Render Function

Renders the PDF417 barcode onto the device such as computer screen or printers.

```
void PDF417Render(DWORD hDC, WORD x, WORD y);
```

Parameters

hDC

Device context on which to render the PDF417 barcode.

x

The x coordinate, in pixels, of the top left corner of the PDF417 barcode .

y

The y coordinate, in pixels, of the top left corner of the PDF417 barcode.

6.7 PDF417SaveAsBMP Function

Saves the PDF417 barcode image as a BMP file.

```
BOOL PDF417SaveAsBMP(LPCTSTR FileName);
```

Parameters

FileName

A string that contains the name of the file to which to save BMP format PDF417 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 PDF417GetActualSize() function to obtain the actual size of the

PDF417 barcode and use PDF417SetSize() function to set image size by adding surrounding white space around the PDF417 barcode.

See Also

PDF417GetActualSize() Function | PDF417SetSize() Function

6.8 PDF417SaveAsWMF Function

Saves the PDF417 barcode image as a WMF file.

```
BOOL PDF417SaveAsWMF(LPCTSTR FileName);
```

Parameters

FileName

A string that contains the name of the file to which to save WMF format PDF417 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 PDF417GetActualSize() function to obtain the actual size of the PDF417 barcode and use PDF417SetSize() function to set image size by adding surrounding white space around the PDF417 barcode.

See Also

PDF417GetActualSize() Function | PDF417SetSize() Function

6.9 PDF417SetBackColor Function

Specifies the RGB triplet of the background color.

```
void PDF417SetBackColor(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.

6.10 PDF417SetBarColor Function

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

```
void PDF417SetBarColor(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 PDF417SetDefault Function

Initializes the PDF417 barcode parameters with the default values.

```
void PDF417SetDefault();
```

6.12 PDF417SetMessage Function

Specifies the message to encode using the appropriate parameters.

```
void PDF417SetMessage(LPCTSTR Message);
```

Parameters

Message

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

6.13 PDF417SetSize Function

Sets the size of the image which contains the PDF417 barcode.

```
void PDF417SetSize(DWORD Width, DWORD Height);
```

Parameters

Width

The width, in pixels, of the image.

Height

The height, in pixels, of the image.

Remarks

First call PDF417GetActualSize() function to obtain the actual size of the PDF417 barcode, then use this function to set image size by adding surrounding white space around the PDF417 barcode.

See Also

PDF417GetActualSize() Function

7 Convert Size from CMs to Pixels

Internally our PDF417 Win32 DLL converts the narrow bar 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 PDF417 barcode onto the computer screen and the screen resolution is 96dpi.

(1) Set *NarrowBarWidth* parameter to 0.04, $size_in_pixels = 0.04 * 96 / 2.54 = 1.5118$, round up 1.5118 to 2, so actual narrow bar width is 2 pixels.

(2) Set *NarrowBarWidth* parameter to 0.06, $size_in_pixels = 0.06 * 96 / 2.54 = 2.2677$, round down 2.2677 to 2, so actual narrow bar width is 2 pixels.

(3) Set *NarrowBarWidth* parameter to 0.07, $size_in_pixels = 0.07 * 96 / 2.54 = 2.6456$, round up 2.6456 to 3, so actual narrow bar width is 3 pixels.

Different *NarrowBarWidth* parameter values might end up with same narrow bar width in pixels due to performing rounding operations.

8 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 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.
