

# 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 Word Demo</b>	<b>3</b>
1 Install Template File.....	3
2 Create Single Barcode.....	5
3 Create Multiple Barcodes.....	5
4 Mail Merge.....	7
<b>Part V Excel Demo</b>	<b>8</b>
1 Change Settings.....	8
2 Create Multiple Barcodes.....	9
<b>Part VI Reference Guide</b>	<b>10</b>
1 QRAppearance Function.....	10
2 QRConfigure Function.....	11
3 QRCopyToClipboard Function.....	13
4 QRGetActualIRC Function.....	13
5 QRGetActualSize Function.....	14
6 QRGetPatternData Function.....	14
7 QRRender Function.....	15
8 QRSaveAsBMP Function.....	15
9 QRSaveAsWMF Function.....	16
10 QRSetBackColor Function.....	16
11 QRSetBarColor Function.....	17
12 QRSetDefault Function.....	17
13 QRSetMessage Function.....	17
14 QRSetSize Function.....	17
<b>Part VII Convert Size from CMs to Pixels</b>	<b>18</b>
<b>Part VIII Data Capacity Tables</b>	<b>18</b>

1	Level L.....	18
2	Level M.....	19
3	Level Q.....	20
4	Level H.....	21
<b>Part IX License</b>		<b>22</b>
	<b>Index</b>	<b>0</b>

---

# 1 Introduction

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

QRCode is designed to pack a lot of information in a very small space, our QRCode Win32 DLL supports Model 2, it is capable of encoding up to 2953 bytes, 4296 alphanumeric characters, or 7089 numeric digits.

## 2 Installation

### 2.1 Trial Version

1. UnZip MW6QRCodeWin32.ZIP, run the setup.exe to install QRCode Win32 DLL.
2. The trial version QRCode Win32 DLL appends "MW6 Demo" to the string encoded with QRCode format.

### 2.2 Full Version

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

## 3 How to Distribute It

If you want to redistribute QRCode Win32 DLL as part of your application, on the target machine, simply put **QRCodeWin32.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

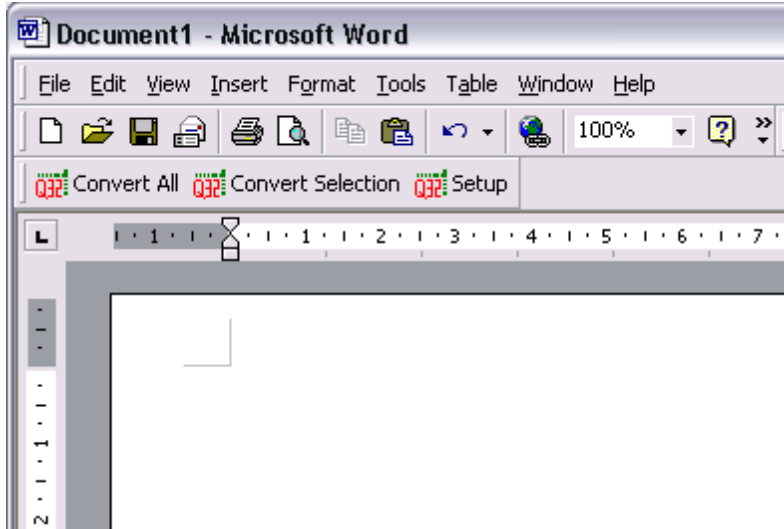
### 4.1 Install Template File

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

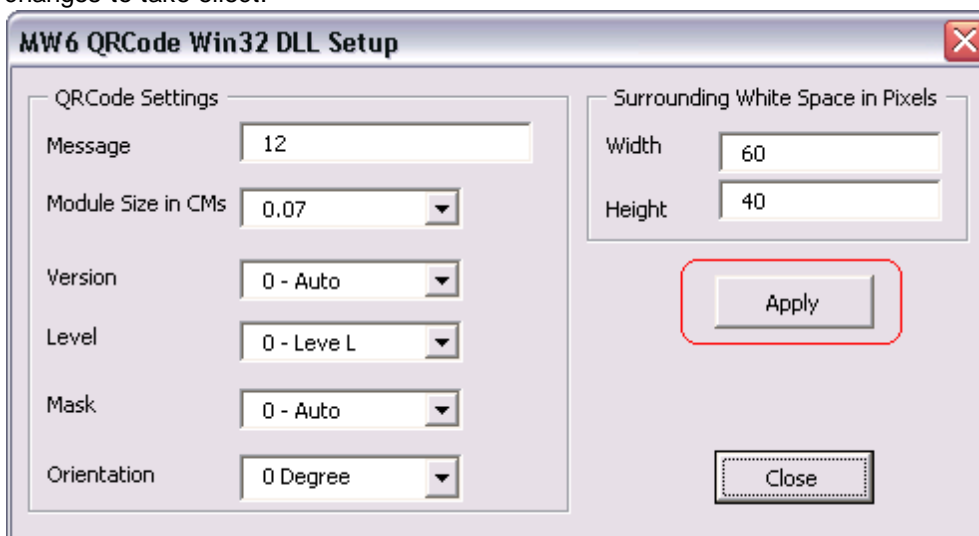
OS	Location
Windows 2000/XP	"C:\Documents and Settings\ <user name="">\Application Data\Microsoft\Word\Startup"</user>
Windows NT4	"C:\Winnt\Profiles\ <user name="">\Application Data\Microsoft\Word\Startup"</user>
Windows 95, 98, ME	Office XP: "C:\Program Files\Microsoft Office\Office10\Startup"  Office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

2. Copy MW6\_QRCode\_Win32.dot, which usually is in the folder "c:\Program Files\MW6 Win32 DLL \QRCode", to the Word Startup folder.

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

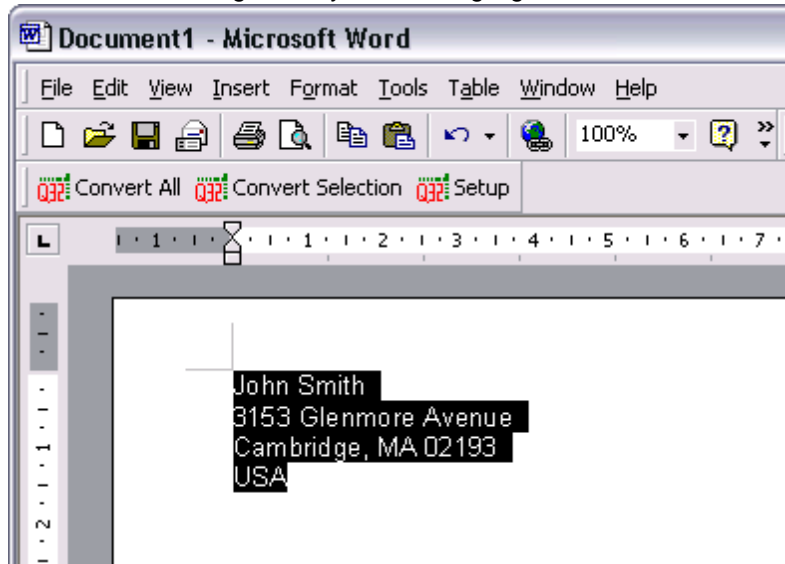


4. Choose a few appropriate values for QRCode configurations, click on "**Apply**" button to allow the changes to take effect.

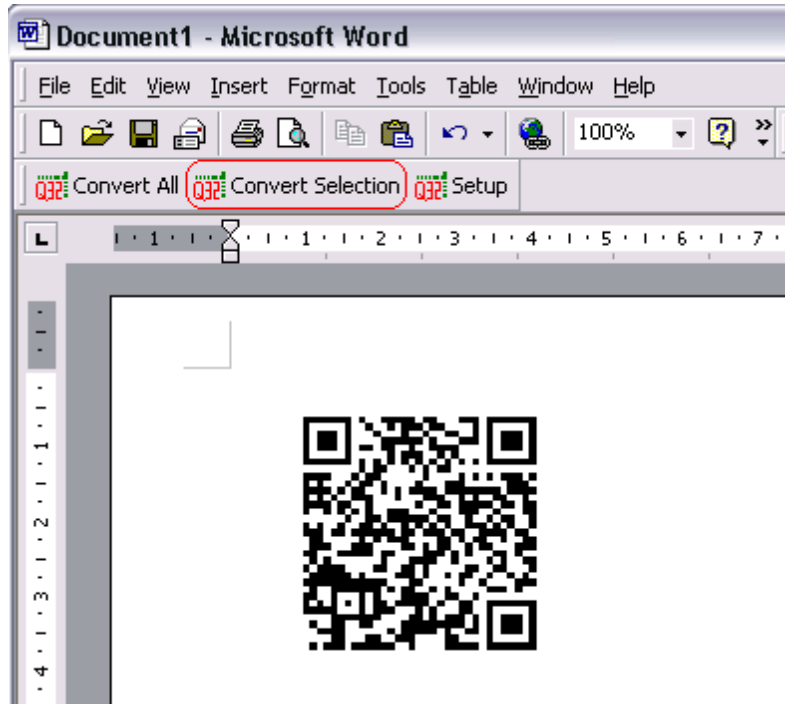


## 4.2 Create Single Barcode

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

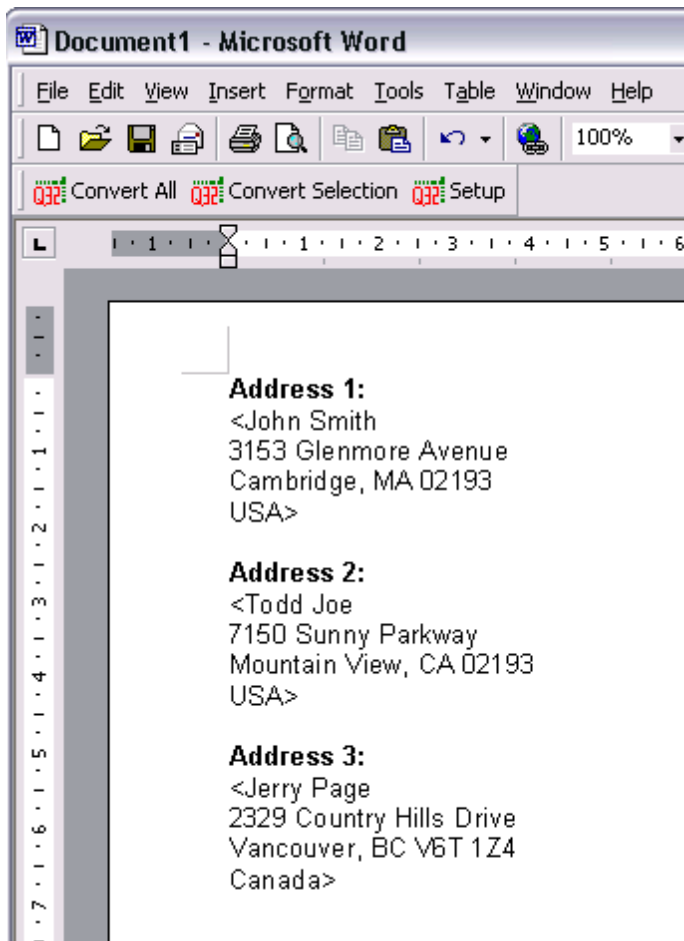


2. Click on "Convert Selection" to create a QRCode barcode.

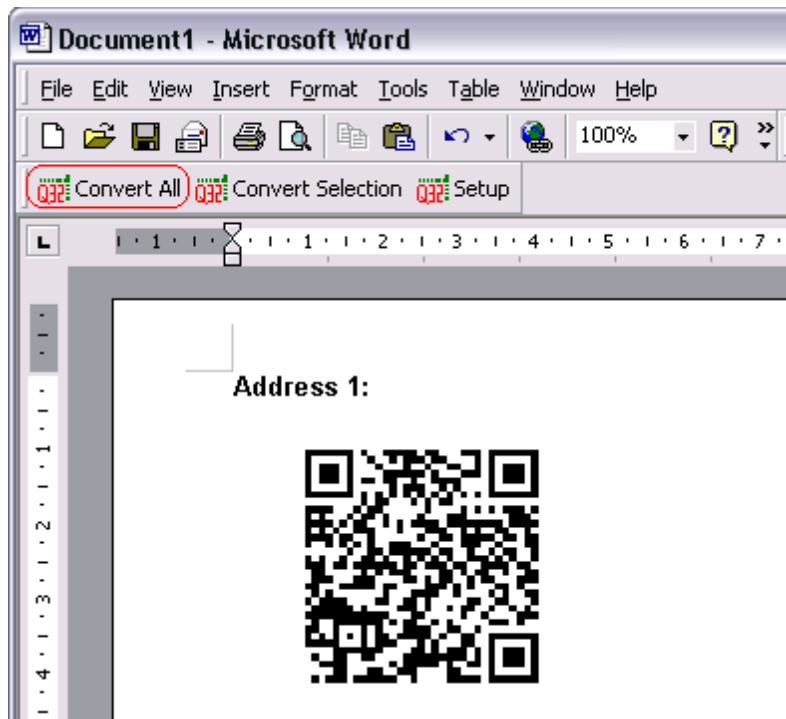


## 4.3 Create Multiple Barcodes

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

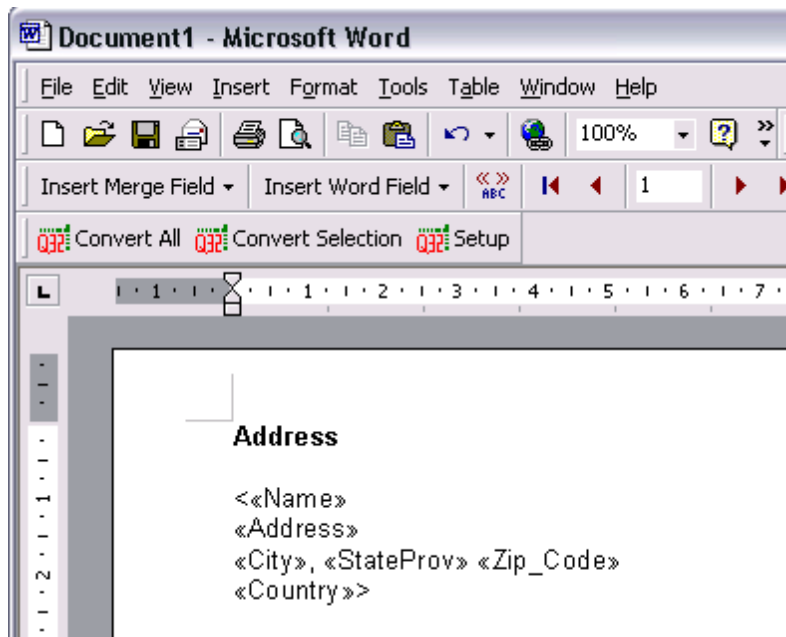


2. Click on "**Convert All**" to create QRCode barcodes for the paragraphs surrounded with the "<" and ">" characters.



## 4.4 Mail Merge

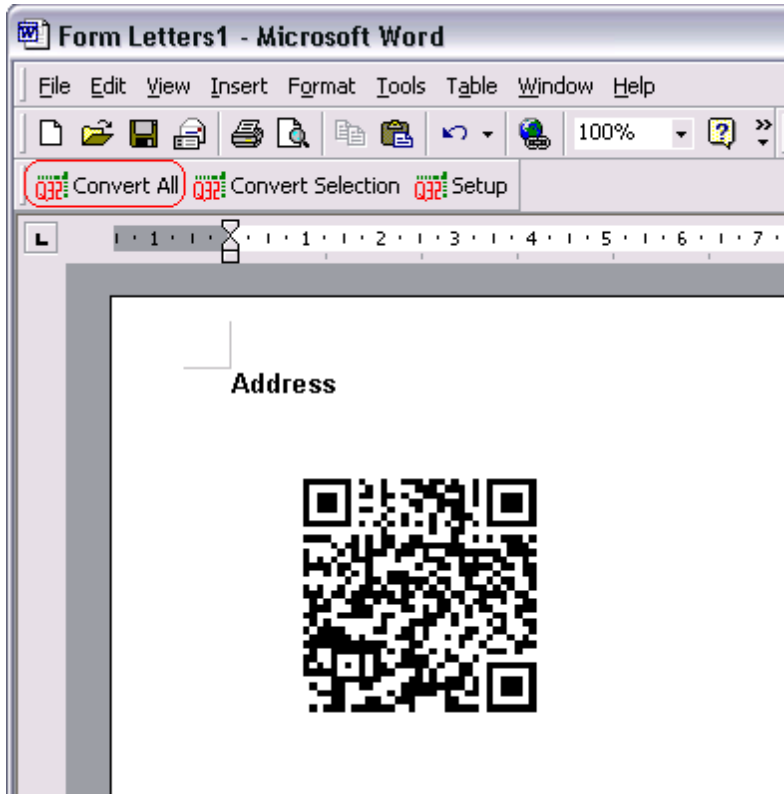
1. In Mail Merge, surround the paragraphs which will be converted to QRCode barcodes with "<" and ">" characters.



2. Click on "Merge ..."



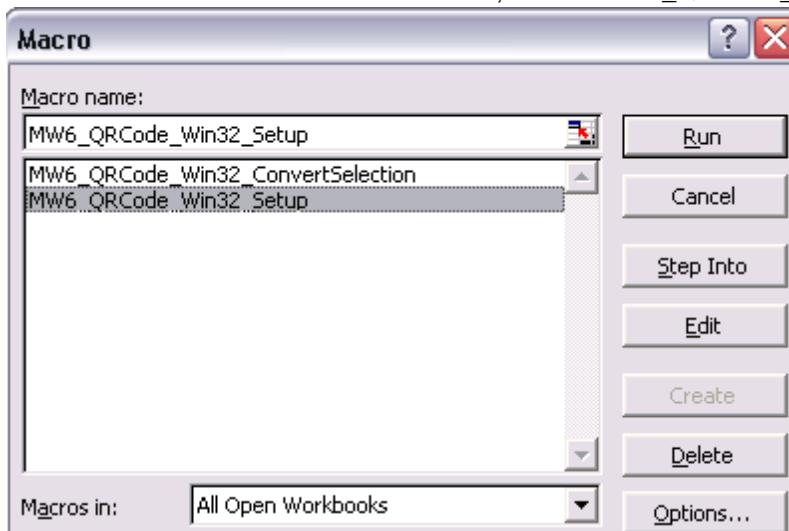
3. Click on "**Convert All**" to create QRCode barcodes for the paragraphs surrounded with "<" and ">" characters.



## 5 Excel Demo

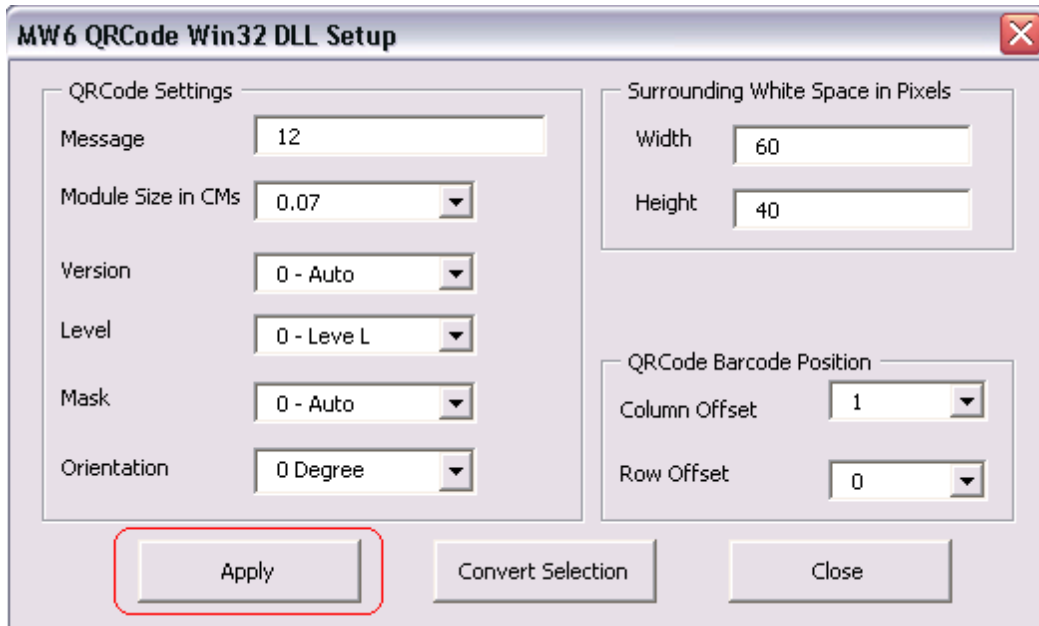
### 5.1 Change Settings

1. In Excel, open MW6\_QRCode\_Win32.XLS.
2. Click on "**Tools**" > "**Macro**" > "**Macros**", select "**MW6\_QRCode\_Win32\_Setup**".





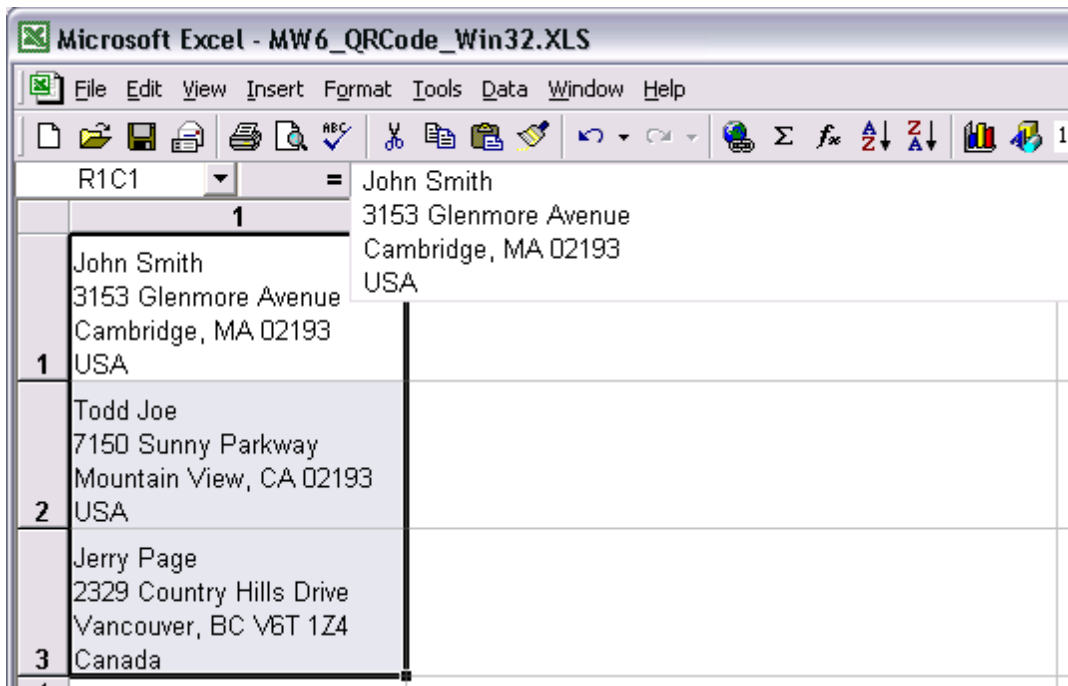
3. Click on "Run".



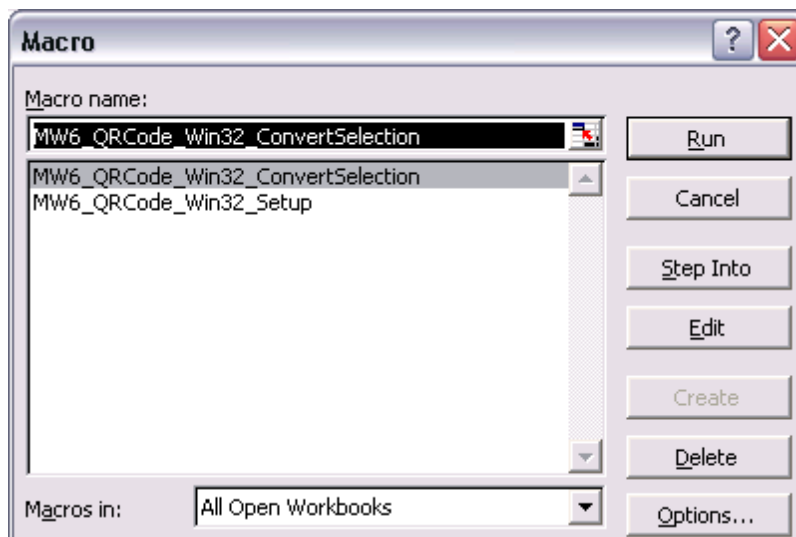
4. Choose a few appropriate values for QRCode configurations, click on "**Apply**" button to allow the changes to take effect, "Column Offset" and "Row Offset" are used to specify QRCode barcode position relative to the position of the cell which contains the regular string.

## 5.2 Create Multiple Barcodes

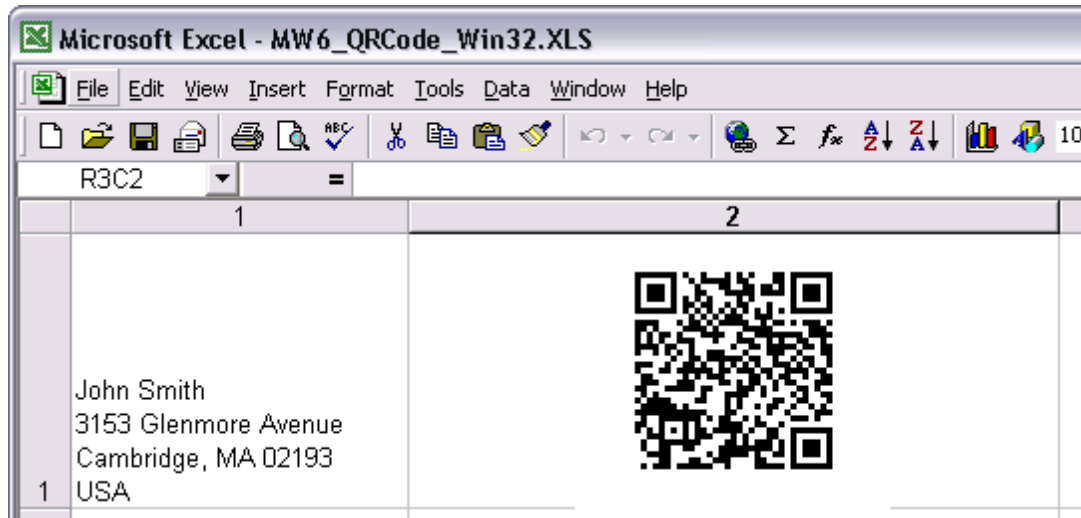
1. Select a few cells.



2. Click on "Tools" > "Macro" > "Macros", select "MW6\_QRCode\_Win32\_ConvertSelection".



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



## 6 Reference Guide

### 6.1 QRAppearance Function

Sets up the parameters of QRCode barcode appearance.

```
void QRAppearance(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 QRCode 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 QRConfigure Function

Sets up the parameters of QRCode barcode.

```
void QRConfigure(WORD Version, WORD Level, WORD Mask);
```

### **Parameters**

#### *Version*

Indicates the version of the QRCode barcode, this parameter can be one of the following values.

Value	Description
0	Auto
1	21 X 21
2	25 X 25
3	29 X 29
4	33 X 33
5	37 X 37
6	41 X 41
7	45 X 45
8	49 X 49
9	53 X 53
10	57 X 57
11	61 X 61
12	65 X 65
13	69 X 69
14	73 X 73
15	77 X 77
16	81 X 81
17	85 X 85
18	89 X 89
19	93 X 93

20	97 X 97
21	101 X 101
22	105 X 105
23	109 X 109
24	113 X 113
25	117 X 117
26	121 X 121
27	125 X 125
28	129 X 129
29	133 X 133
30	137 X 137
31	141 X 141
32	145 X 145
33	149 X 149
34	153 X 153
35	157 X 157
36	161 X 161
37	165 X 165
38	169 X 169
39	173 X 173
40	177 X 177

*Level*

Indicates the level of error correction allowing recovery, this parameter can be one of the following values.

Value	Comment
0	Level L
1	Level M
2	Level Q
3	Level H

*Mask*

Indicates the mask pattern for improving the readability, this parameter can be one of the following values.

Value	Comment
0	Auto
1	Mask 0
2	Mask 1
3	Mask 2
4	Mask 3
5	Mask 4
6	Mask 5
7	Mask 6
8	Mask 7

## 6.3 QRCopyToClipboard Function

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

```
BOOL QRCopyToClipboard();
```

### 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 `QRGetActualSize()` function to obtain the actual size of the QRCode barcode and use `QRSetSize()` function to set the image size by adding surrounding white space around the QRCode barcode.

### See Also

`QRGetActualSize()` Function | `QRSetSize()` Function

## 6.4 QRGetActualRC Function

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

```
void QRGetActualRC(WORD *ActualRows, WORD *ActualCols);
```

### Parameters

*ActualRows*

A pointer to the variable that receives the final number of rows for the QRCode barcode.

*ActualCols*

A pointer to the variable that receives the final number of columns for the QRCode barcode.

### Remarks

If you set *Version* to 0 (Auto version), QRCode Win32 DLL will automatically choose an appropriate version 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 *Version* to other values and the data capacity of the selected version is not big enough to encode the string, QRCode Win32 DLL will also automatically choose an appropriate version 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 *Version* parameter.

### See Also

QRConfigure() Function

## 6.5 QRGetActualSize Function

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

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

### Parameters

*ScreenIsTarget*

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

*TargetHDC*

Device context on which to render the QRCode 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 QRCode barcode (in pixels).

*ActualHeight*

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

## 6.6 QRGetPatternData Function

Gets the QRCode barcode pattern matrix data.

```
BOOL QRGetPatternData(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 QRCode 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 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 function to obtain the QRCode barcode pattern matrix data and render the QRCode barcode onto any device such as the printer, call *QRSetMessage()* and *QRConfigure()* functions before calling this function, other functions don't affect the output of QRCode barcode pattern matrix.

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

## **6.7 QRRender Function**

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

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

#### **Parameters**

##### *hDC*

Device context on which to render the QRCode barcode.

##### *x*

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

##### *y*

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

## **6.8 QRSaveAsBMP Function**

Saves the QRCode barcode image as a BMP file.

```
BOOL QRSaveAsBMP(LPCTSTR FileName);
```

#### **Parameters**

##### *FileName*

A string that contains the name of the file to which to save BMP format QRCode 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 `QRGetActualSize()` function to obtain the actual size of the QRCode barcode and use `QRSetSize()` function to set image size by adding surrounding white space around the QRCode barcode.

**See Also**

`QRGetActualSize()` Function | `QRSetSize()` Function

## 6.9 QRSaveAsWMF Function

Saves the QRCode barcode image as a WMF file.

```
BOOL QRSaveAsWMF(LPCTSTR FileName);
```

**Parameters**

*FileName*

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

**See Also**

`QRGetActualSize()` Function | `QRSetSize()` Function

## 6.10 QRSetBackColor Function

Specifies the RGB triplet of the background color.

```
void QRSetBackColor(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 QRSetBarColor Function

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

```
void QRSetBarColor(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 QRSetDefault Function

Initializes the QRCode barcode parameters with the default values.

```
void QRSetDefault();
```

## 6.13 QRSetMessage Function

Specifies the message to encode using the appropriate parameters.

```
void QRSetMessage(LPCTSTR Message);
```

### Parameters

*Message*

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

## 6.14 QRSetSize Function

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

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

**Parameters***Width*

The width, in pixels, of the image.

*Height*

The height, in pixels, of the image.

**Remarks**

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

**See Also**

QRGetActualSize() Function

## 7 Convert Size from CMs to Pixels

Internally our QRCode Win32 DLL 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 the QRCode 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 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 in pixels due to performing rounding operations.

## 8 Data Capacity Tables

### 8.1 Level L

Version	Capacity (in digits)	Capacity (in alphanumeric characters)	Capacity (in bytes)
1	41	25	17

2	77	47	32
3	127	77	53
4	187	114	78
5	255	154	106
6	322	195	134
7	370	224	154
8	461	279	192
9	552	335	230
10	652	395	271
11	772	468	321
12	883	535	367
13	1022	619	425
14	1101	667	458
15	1250	758	520
16	1408	854	586
17	1548	938	644
18	1725	1046	718
19	1903	1153	792
20	2061	1249	858
21	2232	1352	929
22	2409	1460	1003
23	2620	1588	1091
24	2812	1704	1171
25	3057	1853	1273
26	3283	1990	1367
27	3517	2132	1465
28	3669	2223	1528
29	3909	2369	1628
30	4158	2520	1732
31	4417	2677	1840
32	4686	2840	1952
33	4965	3009	2068
34	5253	3183	2188
35	5529	3351	2303
36	5836	3537	2431
37	6153	3729	2563
38	6479	3927	2699
39	6743	4087	2809
40	7089	4296	2953

## 8.2 Level M

Version	Capacity (in digits)	Capacity (in alphanumeric characters)	Capacity (in bytes)
1	34	20	14
2	63	38	26
3	101	61	42
4	149	90	62
5	202	122	84
6	255	154	106
7	293	178	122
8	365	221	152

9	432	262	180
10	513	311	213
11	604	366	251
12	691	419	287
13	796	483	331
14	871	528	362
15	991	600	412
16	1082	656	450
17	1212	734	504
18	1346	816	560
19	1500	909	624
20	1600	970	666
21	1708	1035	711
22	1872	1134	779
23	2059	1248	857
24	2188	1326	911
25	2395	1451	997
26	2544	1542	1059
27	2701	1637	1125
28	2857	1732	1190
29	3035	1839	1264
30	3289	1994	1370
31	3486	2113	1452
32	3693	2238	1538
33	3909	2369	1628
34	4134	2506	1722
35	4343	2632	1809
36	4588	2780	1911
37	4775	2894	1989
38	5039	3054	2099
39	5313	3220	2213
40	5596	3391	2331

### 8.3 Level Q

Version	Capacity (in digits)	Capacity (in alphanumeric characters)	Capacity (in bytes)
1	27	16	11
2	48	29	20
3	77	47	32
4	111	67	46
5	144	87	60
6	178	108	74
7	207	125	86
8	259	157	108
9	312	189	130
10	364	221	151
11	427	259	177
12	489	296	203
13	580	352	241
14	621	376	258
15	703	426	292

16	775	470	322
17	876	531	364
18	948	574	394
19	1063	644	442
20	1159	702	482
21	1224	742	509
22	1358	823	565
23	1468	890	611
24	1588	963	661
25	1718	1041	715
26	1804	1094	751
27	1933	1172	805
28	2085	1263	868
29	2181	1322	908
30	2358	1429	982
31	2473	1499	1030
32	2670	1618	1112
33	2805	1700	1168
34	2949	1787	1228
35	3081	1867	1283
36	3244	1966	1351
37	3417	2071	1423
38	3599	2181	1499
39	3791	2298	1597
40	3993	2420	1663

## 8.4 Level H

Version	Capacity (in digits)	Capacity (in alphanumeric characters)	Capacity (in bytes)
1	17	10	7
2	34	20	14
3	58	35	24
4	82	50	34
5	106	64	44
6	139	84	58
7	154	93	64
8	202	122	84
9	235	143	98
10	288	174	119
11	331	200	137
12	374	227	155
13	427	259	177
14	468	283	194
15	530	321	220
16	602	365	250
17	674	408	280
18	746	452	310
19	813	493	338
20	919	557	382
21	969	587	403
22	1056	640	439

23	1108	672	461
24	1228	744	511
25	1286	779	535
26	1425	864	593
27	1501	910	625
28	1581	958	658
29	1677	1016	698
30	1782	1080	742
31	1897	1150	790
32	2022	1226	842
33	2157	1307	898
34	2301	1394	958
35	2361	1431	983
36	2524	1530	1051
37	2625	1591	1093
38	2735	1658	1139
39	2927	1774	1219
40	3057	1852	1273

## 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.

---