

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 Word Demo	4
1 Run Setup.....	4
2 Create Single Barcode.....	5
3 Create Multiple Barcodes.....	5
4 Mail Merge.....	7
Part V Excel Demo	8
1 Change Settings.....	8
2 Create Multiple Barcodes.....	10
Part VI Access Demo	11
Part VII Reference Guide	13
1 Properties.....	13
BackColor Property	13
BarColor Property	13
BorderStyle Property	14
Data Property	14
ImageData Property	14
Level Property	14
Mask Property	15
ModuleSize Property	15
Orientation Property	16
Version Property	16
2 Methods.....	18
CopyToClipboard Method	18
GetActualRC Method	18
GetActualSize Method	18
GetPatternData Method	19
Render Method	20
SaveAsBMP Method	20
SaveAsWMF Method	21
SetSize Method	21

Part VIII Data Capacity Tables	22
1 Level L.....	22
2 Level M.....	22
3 Level Q.....	23
4 Level H.....	24
Part IX License	25
Index	0

1 Introduction

MW6 QRCode ActiveX is a powerful ATL-based control for handling QRCode 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.

QRCode is designed to pack a lot of information in a very small space, our QRCode ActiveX control 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 MW6QRCode.ZIP, run the setup.exe to install QRCode ActiveX.
2. The trial version QRCode ActiveX appends "MW6 Demo" to the string encoded with QRCode barcode.

2.2 Full Version

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

3 How to Distribute It

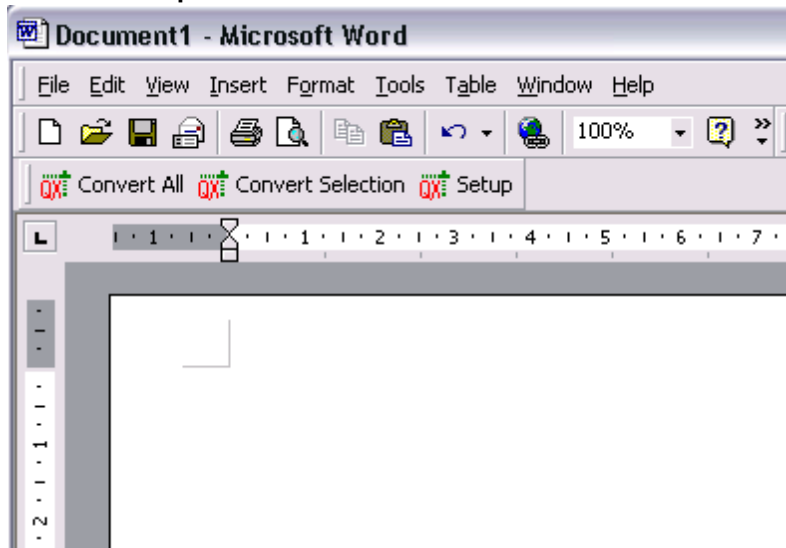
If you want to redistribute the QRCode ActiveX as part of your application, please follow the instructions below:

- 1) For 32-bit version Windows OS, put **MW6QRCode.dll** into the windows 32-bit system folder (e.g. "c:\windows\system32" or "c:\winnt\system32") on the target machine and run "regsvr32 MW6QRCode.dll" to register it.
- 2) For 64-bit version Windows OS, put **MW6QRCode.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 MW6QRCode.dll
- 3) If you want to use QRCode ActiveX in 64-bit version Office Word, Excel or Access, put 64-bit version **MW6QRCode_x64.dll** into "c:\windows\system32" folder, and run the following commands to register it:
 - cd c:\windows\system32
 - regsvr32 MW6QRCode_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".

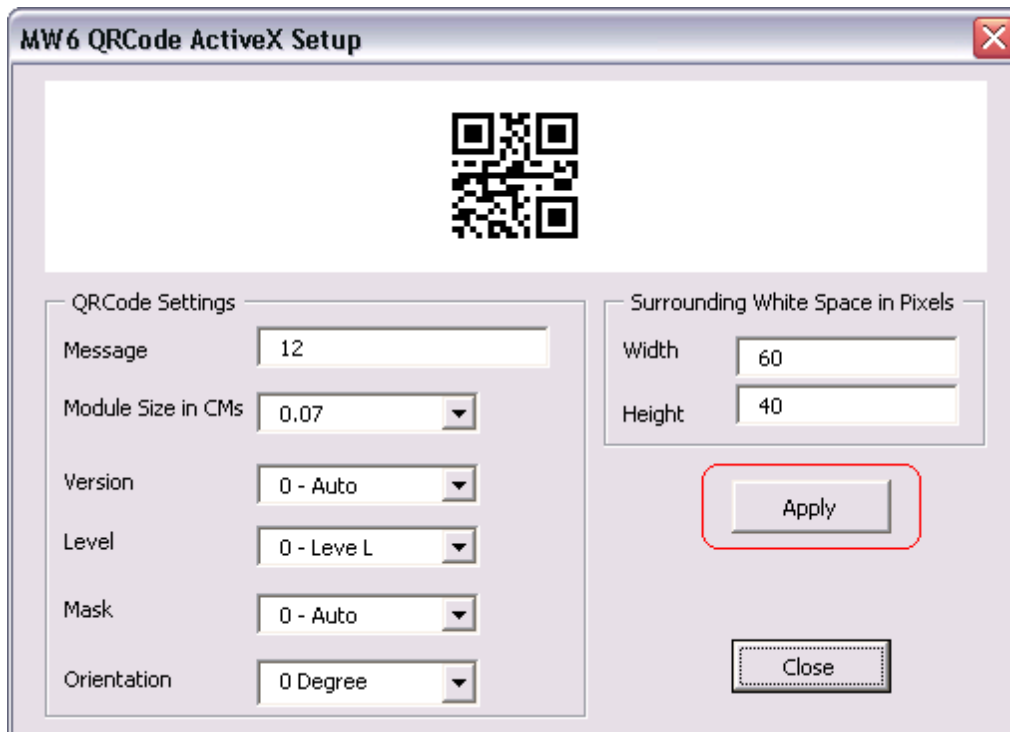
4 Word Demo

4.1 Run Setup

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

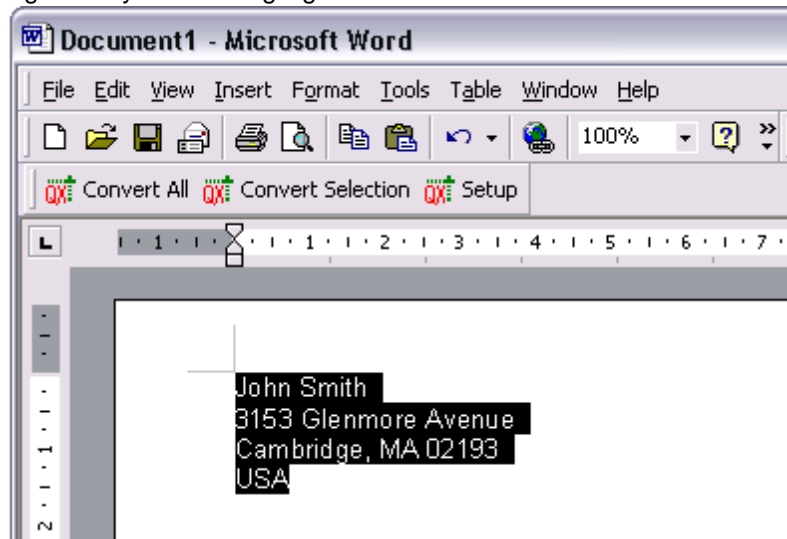


4. Choose a few appropriate values for QRCode configurations, click on "**Apply**" button to allow the changes to take effect.

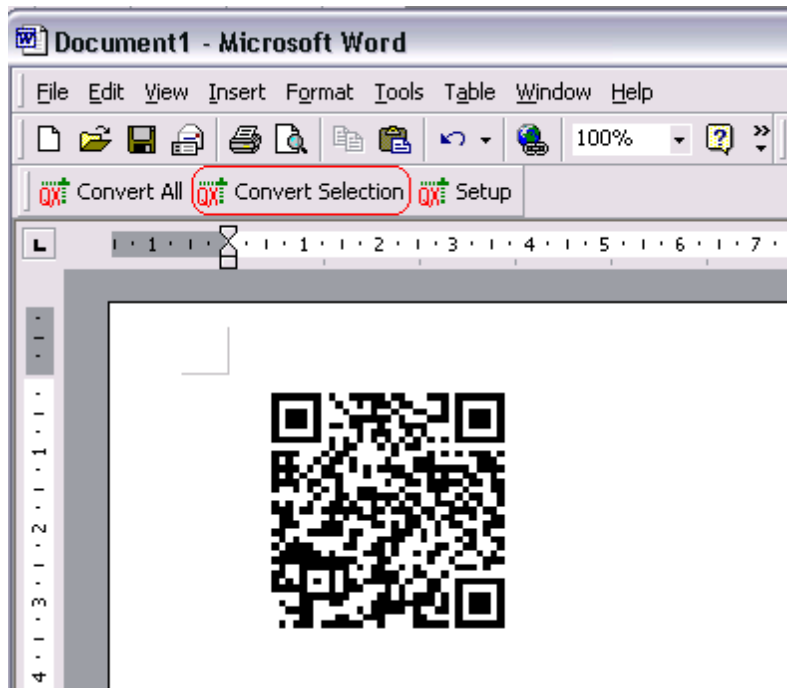


4.2 Create Single Barcode

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

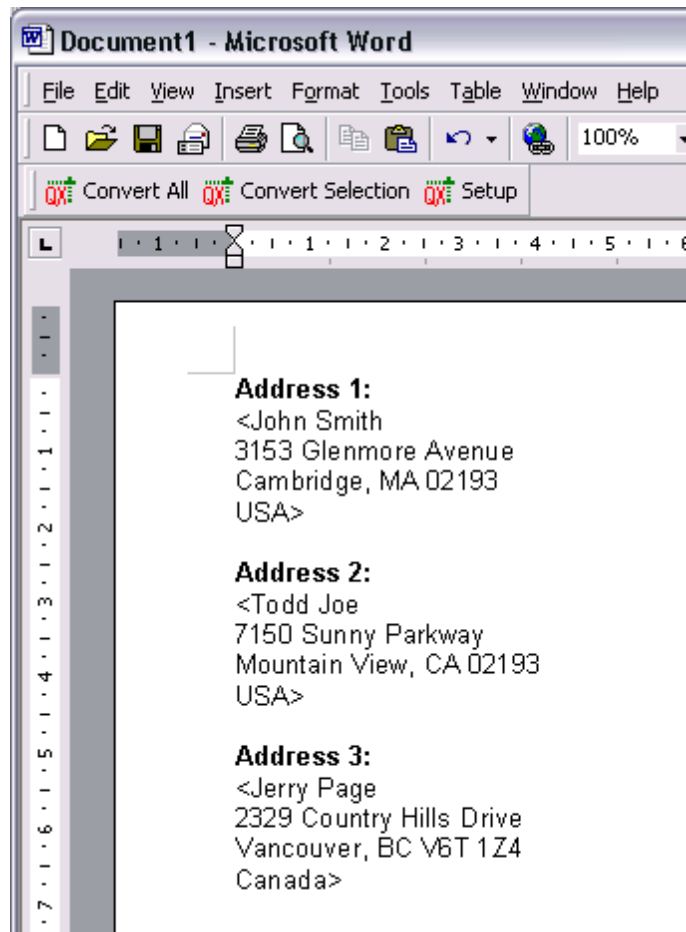


2. Click on "Convert Selection" to create a QRCode barcode.

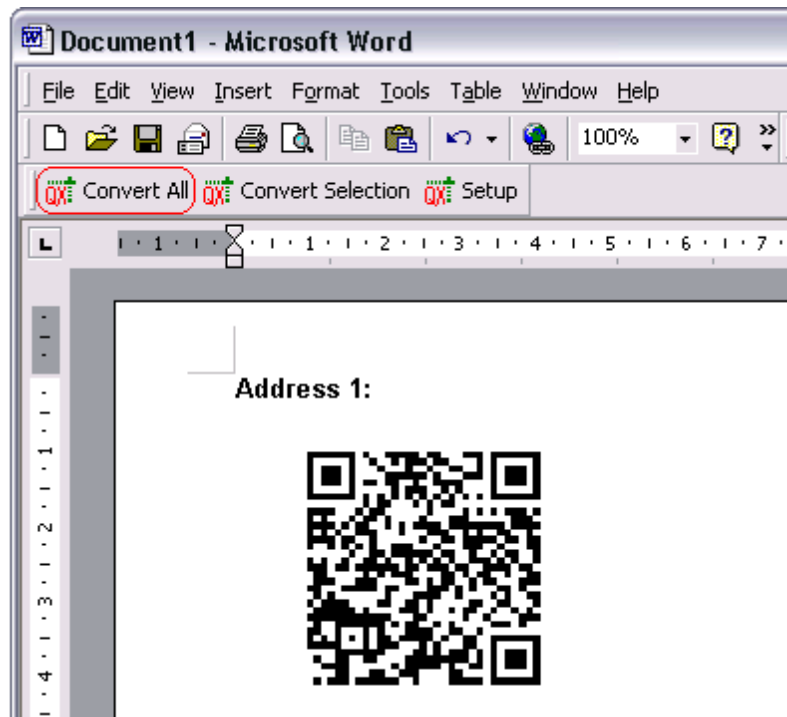


4.3 Create Multiple Barcodes

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

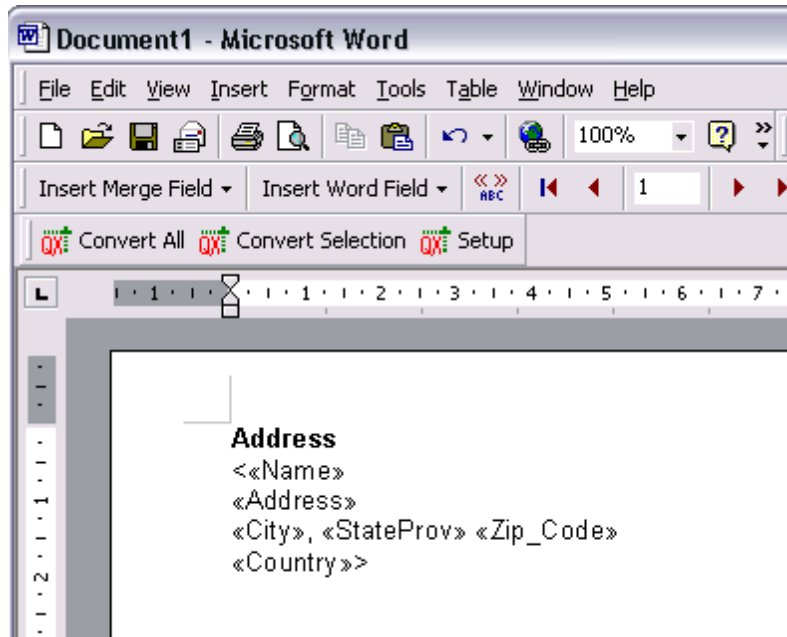


2. Click on "**Convert All**" to create QRCode barcodes for the paragraphs surrounded with the "<" and ">" characters.



4.4 Mail Merge

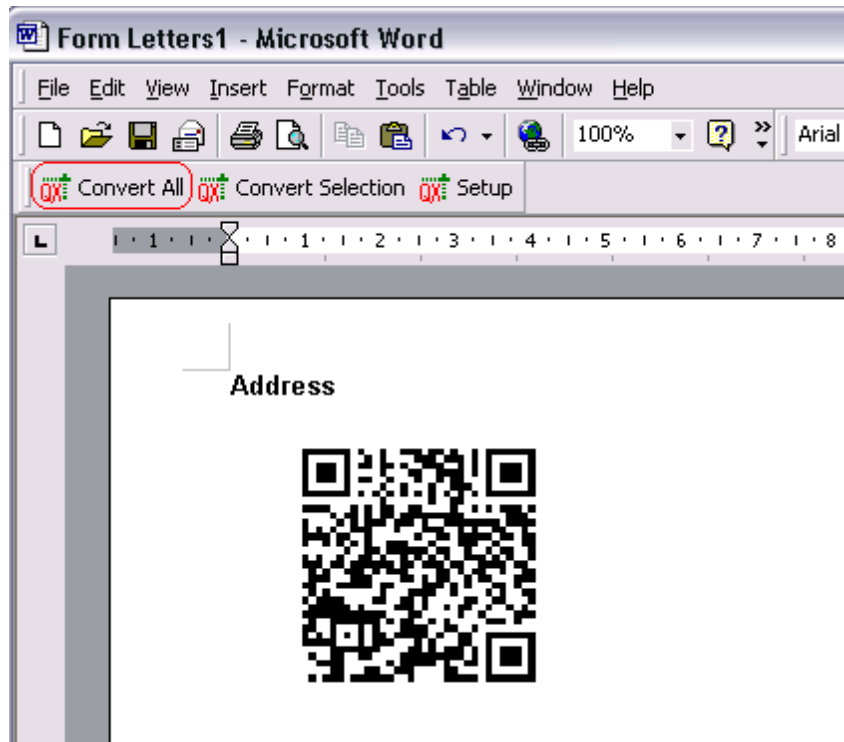
1. In Mail Merge, choose MW6_QRCode_ActiveX.mdb as the data source, surround the paragraphs which will be converted to QRCode barcode with the "<" and ">" characters.



2. Click on "Merge ..."



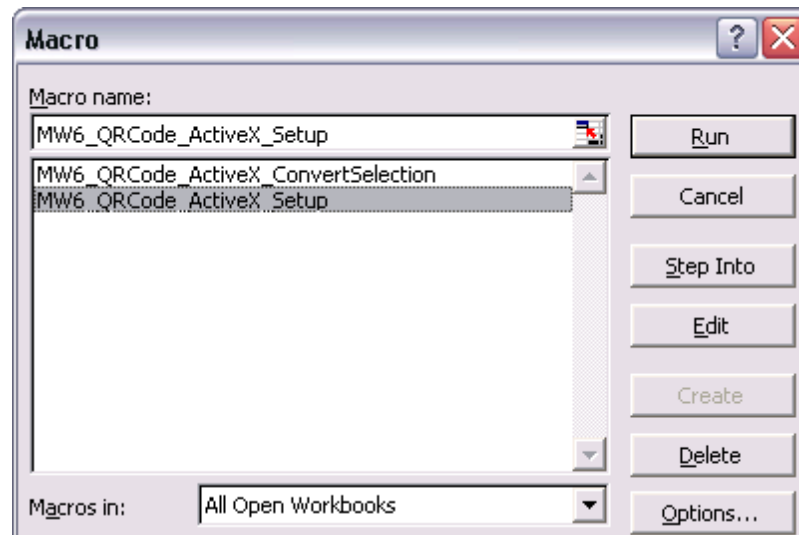
3. Click on "**Convert All**" to create QRCode barcodes for the paragraphs surrounded with the "<" and ">" characters.



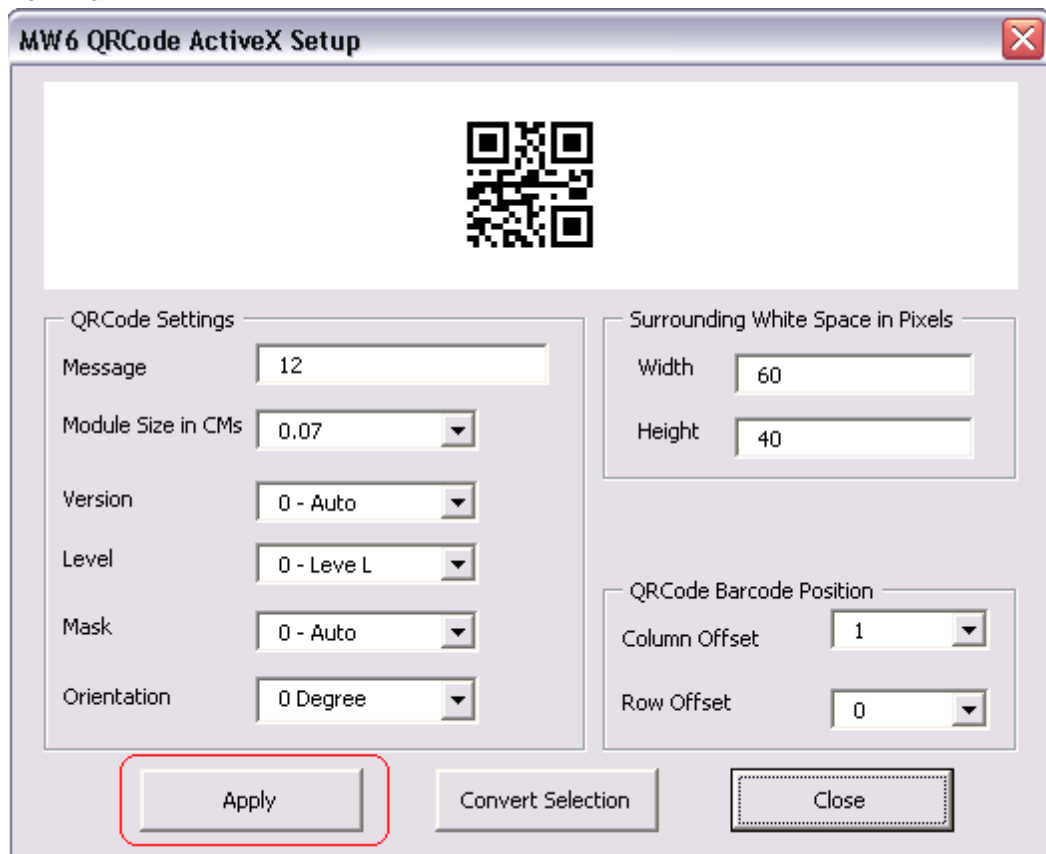
5 Excel Demo

5.1 Change Settings

1. In Excel, open MW6_QRCode_ActiveX.XLS.
2. Click on "Tools" > "Macro" > "Macros", select "MW6_QRCode_ActiveX_Setup".



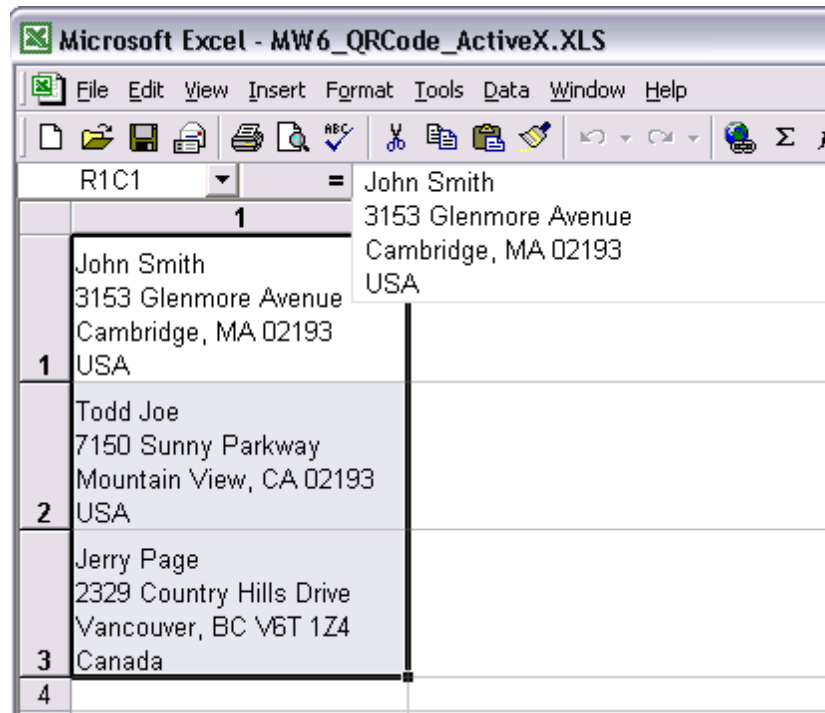
3. Click on "Run".



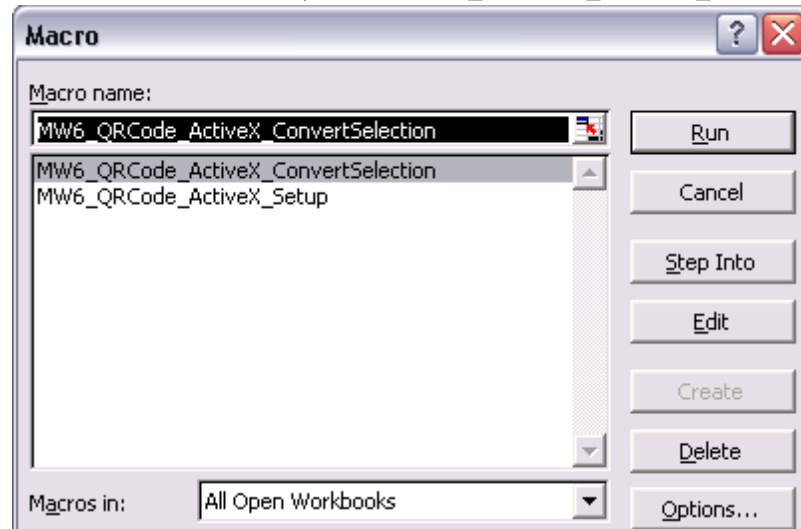
4. Choose a few appropriate values for QRCode configurations, click on "Apply" button to allow the changes to take effect, "Column Offset" and "Row Offset" are used to specify QRCode barcode position relative to the position of the cell which contains the regular string.

5.2 Create Multiple Barcodes

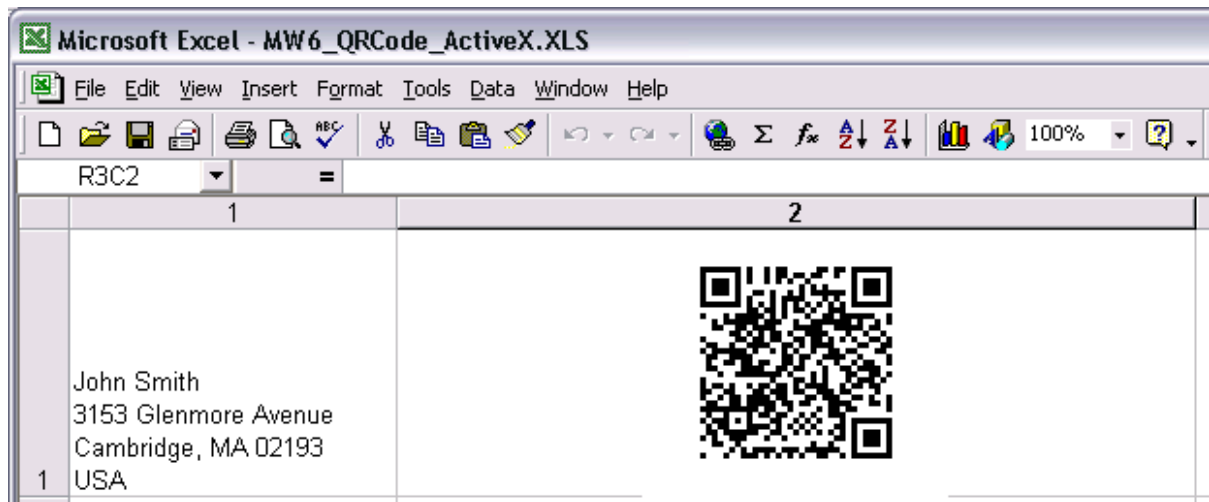
1. Select a few cells.



2. Click on "Tools" > "Macro" > "Macros", select "MW6_QRCode_ActiveX_ConvertSelection".

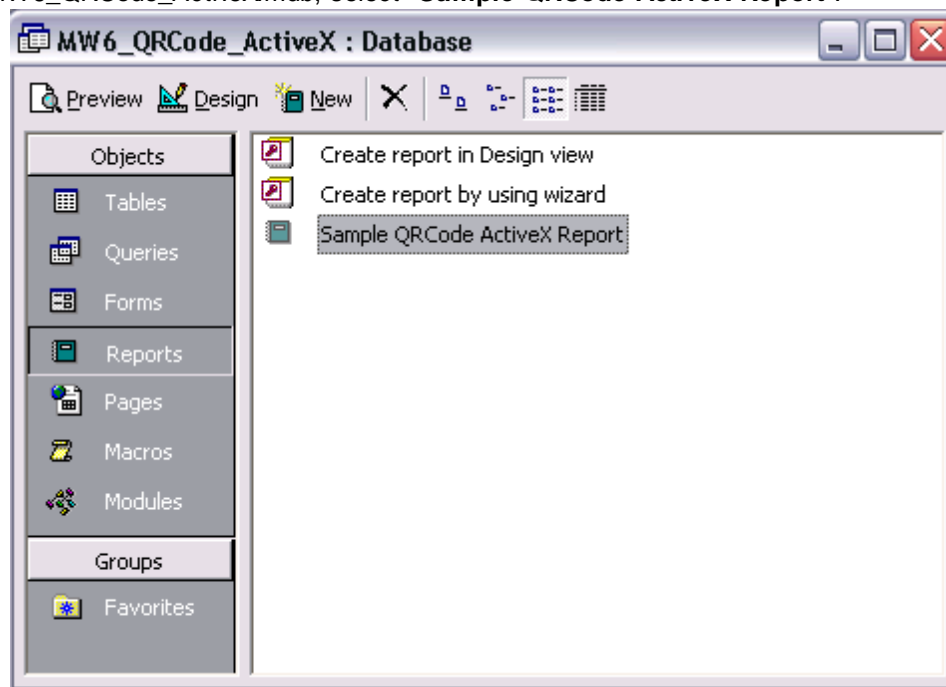


3. Click on "Run" to create QRCode barcodes for the selected cells.

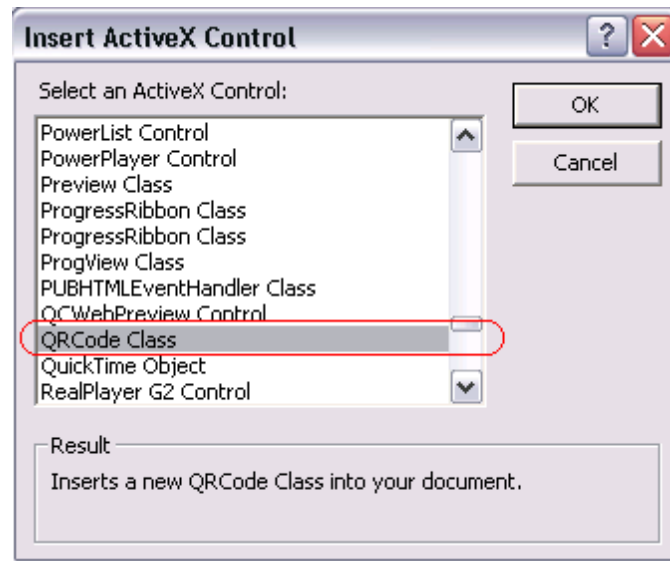


6 Access Demo

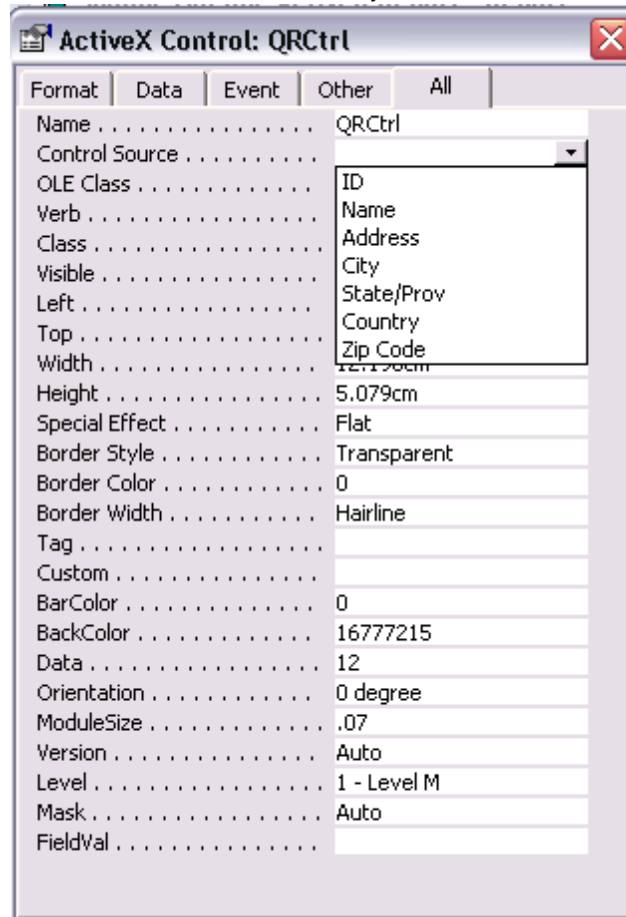
1. Open MW6_QRCode_ActiveX.mdb, select "**Sample QRCode ActiveX Report**".



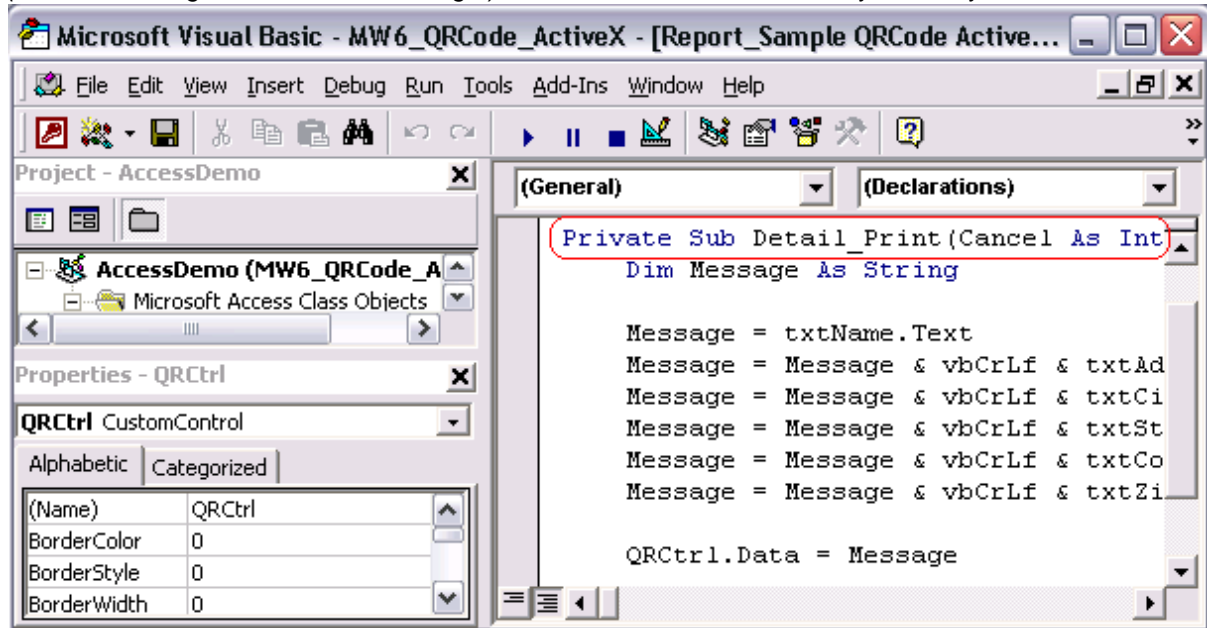
2. Click on "**Design**", insert a MW6 QRCode ActiveX control into the report.



3. Change its properties to meet your application requirements, our QRCode ActiveX supports the data binding so you can bind a field in a database to the control and generate QRCode 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 the 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 QRCode barcodes dynamically.



4. Click on "Preview" to view QRCode barcodes.

7 Reference Guide

7.1 Properties

7.1.1 BackColor Property

Gets or sets the background color of the QRCode barcode.

Property Data Type

OLE_COLOR

Remarks

The default value is white color.

7.1.2 BarColor Property

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

Property Data Type

OLE_COLOR

Remarks

The default value is black color.

7.1.3 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

7.1.4 Data Property

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

Property Data Type

BSTR

Remarks

The default value is "12".

7.1.5 ImageData Property

Gets WMF format data stream of the QRCode barcode.

Property Data Type

IPictureDisp*

7.1.6 Level Property

Gets or sets the level of error correction allowing recovery.

Property Data Type

short

Remarks

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

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

7.1.7 Mask Property

Gets or sets the mask pattern for improving the readability.

Property Data Type

short

Remarks

The default value is 0, this property 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

7.1.8 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 QRCode 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.

7.1.9 Orientation Property

Gets or sets the orientation of the QRCode 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

7.1.10 Version Property

Gets or sets the version of the QRCode barcode.

Property Data Type

short

Remarks

The default value is 0, this property 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

If you set *Version* to 0 (Auto version), our QRCode ActiveX control will automatically choose an appropriate version with enough data capacity to encode the string.

If you set *Version* to other values and the data capacity of the selected version is not big enough to encode the string, our QRCode ActiveX control will also automatically choose an appropriate version with bigger data capacity to encode the string.

7.2 Methods

7.2.1 CopyToClipboard Method

Copies the QRCode barcode image into the system clipboard.

```
void CopyToClipboard();
```

Remarks

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

See Also

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

7.2.2 GetActualRC Method

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

```
void GetActualRC(short *ActualRows, short *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 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 *Version* to other values and the data capacity of the selected format is not big enough to encode the string, QRCode 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 *Version* property.

7.2.3 GetActualSize Method

Gets the actual size of the QRCode 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 QRCode barcode, if the parameter *ScreenIsTarget* is set to TRUE, this parameter is irrelevant.

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).

7.2.4 GetPatternData Method

Gets the QRCode 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 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 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 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..

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 QRCode barcode pattern matrix data and render the QRCode barcode onto any device such as the printer, only *Data*, *Level*, *Mask* and *Version* properties affect the pattern matrix data output.

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

7.2.5 Render Method

Renders the QRCode 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 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.

7.2.6 SaveAsBMP Method

Saves the QRCode 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 QRCode barcode and use `SetSize()` method to set image size by adding surrounding white space around the QRCode barcode.

See Also

`GetActualSize()` Method | `SetSize()` Method

7.2.7 SaveAsWMF Method

Saves the QRCode 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 QRCode barcode and use `SetSize()` method to set image size by adding surrounding white space around the QRCode barcode.

See Also

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

7.2.8 SetSize Method

Sets the size of the image which contains the QRCode 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 QRCode barcode, then use this method to set image size by adding surrounding white space around the QRCode barcode.

See Also

[GetActualSize\(\) Method](#)

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)	Capacity (in bytes)
---------	----------------------	----------------------------	---------------------

		characters)	
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.
