

---

# Table of Contents

Foreword	0
<b>Part I Introduction</b>	<b>2</b>
<b>Part II Installation and Testing</b>	<b>2</b>
<b>Part III Reference Guide</b>	<b>3</b>
<b>1 Properties</b> .....	<b>3</b>
BackColor Property .....	3
BarColor Property .....	3
BorderStyle Property .....	4
CorrectionLevel Property .....	4
Data Property .....	4
HandleTilde Property .....	5
Height Property .....	5
Mode Property .....	5
ModuleSize Property .....	6
Orientation Property .....	6
PNGImage Property .....	6
PreferredFormat Property .....	7
Width Property .....	8
<b>2 Methods</b> .....	<b>8</b>
GetActualRC Method .....	8
GetActualSize Method .....	9
SetStructuredAppend Method .....	9
<b>Part IV License</b>	<b>10</b>
<b>Index</b>	<b>0</b>

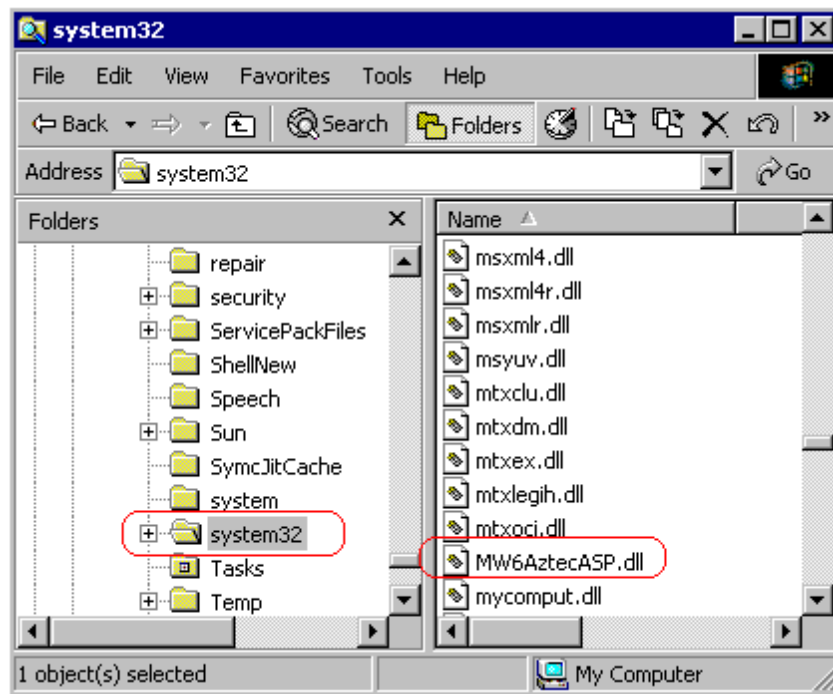
## 1 Introduction

MW6 Aztec ASP component is an ATL COM product which can add professional quality 2D Aztec PNG format images to your web pages hosted on the IIS server.

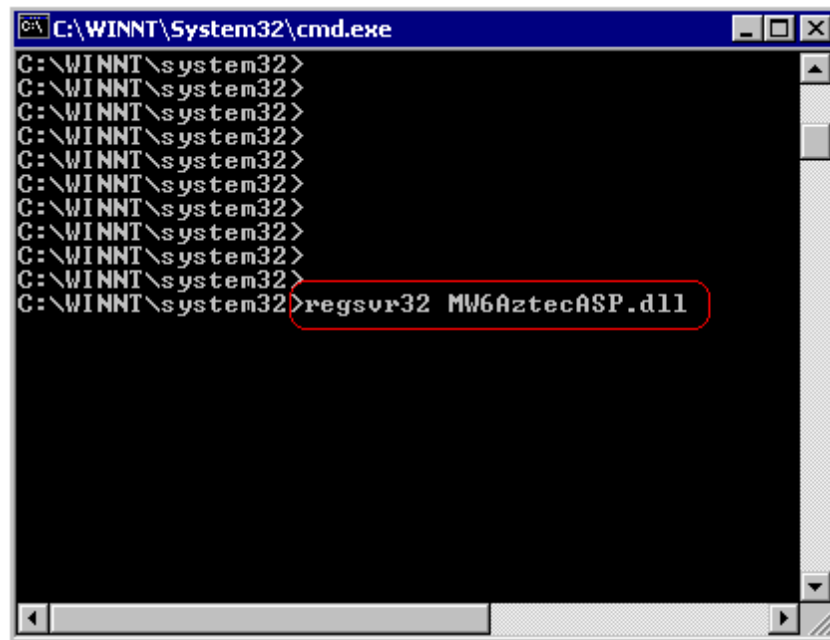
Aztec is designed to pack a lot of information in a very small space, it is capable of encoding 1914 bytes, 3067 alphanumeric characters, or 3832 numeric digits.

## 2 Installation and Testing

1. The trial version Aztec ASP component appends "MW6 Demo" to the string encoded with the Aztec format.
2. For 32-bit version Windows OS, go to the 32-bit system folder (e.g. "c:\winnt\system32" or "c:\windows\system32") of the IIS server. For 64-bit version Windows OS, go to the SysWOW64 folder (e.g. "c:\windows\SysWOW64") of the IIS server. Copy "MW6AztecASP.dll" to the current folder.



3. From the current folder, run "regsvr32 MW6AztecASP.dll" to register it. 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.



```
C:\WINNT\System32\cmd.exe
C:\WINNT\system32>
C:\WINNT\system32>
C:\WINNT\system32>
C:\WINNT\system32>
C:\WINNT\system32>
C:\WINNT\system32>
C:\WINNT\system32>
C:\WINNT\system32>
C:\WINNT\system32>
C:\WINNT\system32>
C:\WINNT\system32>
C:\WINNT\system32>
C:\WINNT\system32>regsvr32 MW6AztecASP.dll
```

4. Copy Demo.html, Createlmng.asp and Show.asp to a folder of the IIS server where Active Server Pages are enabled, for example, you can create one folder "C:\inetpub\wwwroot\MyFolder" and copy those 3 files to this folder.
5. Enter the URL of Demo.html to your browser for verifying whether Aztec ASP component is working or not, for example, you can enter <http://localhost/MyFolder/Demo.html> for testing it on the IIS server itself.

## 3 Reference Guide

### 3.1 Properties

#### 3.1.1 BackColor Property

Sets the background color of the Aztec barcode.

##### Property Data Type

OLE\_COLOR

##### Remarks

The default value is white color.

#### 3.1.2 BarColor Property

Sets the color of the Aztec barcode.

**Property Data Type**

OLE\_COLOR

**Remarks**

The default value is black color.

**3.1.3 BorderStyle Property**

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

**3.1.4 CorrectionLevel Property**

Sets the percentage of errors which can be recovered.

**Property Data Type**

short

**Remarks**

The recommended value is 23 percent, the valid value should be between 5 percent and 95 percent.

**3.1.5 Data Property**

Sets the message to encode with Aztec ASP component.

**Property Data Type**

BSTR

**Remarks**

The default value is "12".

---

### 3.1.6 HandleTilde Property

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 Aztec ASP component 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).

### 3.1.7 Height Property

Sets the height, in pixels, of the image which contains the Aztec barcode.

#### Property Data Type

short

#### Remarks

First call GetActualSize() method to obtain the actual size of the Aztec barcode, then use this property and Width property to set image size by adding surrounding white space around the Aztec barcode.

#### See Also

GetActualSize() Method

### 3.1.8 Mode Property

Sets the encoding mode of the Aztec barcode.

#### Property Data Type

short

#### Remarks

This parameter can be one of the following values:

This parameter can be one of the following values:

Value	Description
0	Auto mode for mainly encoding ASCII characters (0-127)
1	Binary mode for mainly encoding bytes of data

### 3.1.9 ModuleSize Property

Sets the size (width/height) of the square-shaped module.

#### Property Data Type

float

#### Remarks

The default value is 0.07, internally our Aztec ASP 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.

### 3.1.10 Orientation Property

Sets the orientation of the Aztec 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

### 3.1.11 PNGImage Property

Gets PNG format image data stream of the Aztec barcode.

**Property Data Type**

VARIANT

**Remarks**

Use ASP Response.BinaryWrite() method and this property to display an Aztec barcode image on a web page.

**3.1.12 PreferredFormat Property**

Sets the format of the Aztec barcode.

**Property Data Type**

short

**Remarks**

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

Value	Description	Capacity (in digits)	Capacity (in alphanumeric characters)	Capacity (in bytes)
0	Auto format			
1	15 X 15 compact format	13	12	6
2	19 X 19	18	15	8
3	19 X 19 compact format	40	33	19
4	23 X 23	49	40	24
5	23 X 23 compact format	70	57	33
6	27 X 27	84	68	40
7	27 X 27 compact format	110	89	53
8	31 X 31	128	104	62
9	37 X 37	178	144	87
10	41 X 41	232	187	114
11	45 X 45	294	236	145
12	49 X 49	362	291	179
13	53 X 53	433	348	214
14	57 X 57	516	414	256
15	61 X 61	601	482	298
16	67 X 67	691	554	343
17	71 X 71	793	636	394
18	75 X 75	896	718	446
19	79 X 79	1008	808	502
20	83 X 83	1123	900	559
21	87 X 87	1246	998	621
22	91 X 91	1378	1104	687
23	95 X 95	1511	1210	753
24	101 X 101	1653	1324	824
25	105 X 105	1801	1442	898
26	109 X 109	1956	1566	976
27	113 X 113	2116	1694	1056
28	117 X 117	2281	1826	1138
29	121 X 121	2452	1963	1224

30	125 X 125	2632	2107	1314
31	131 X 131	2818	2256	1407
32	135 X 135	3007	2407	1501
33	139 X 139	3205	2565	1600
34	143 X 143	3409	2728	1702
35	147 X 147	3616	2894	1806
36	151 X 151	3832	3067	1914

If you set *PreferredFormat* to 0 (Auto format), our Aztec ASP 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 Aztec ASP control will also automatically choose an appropriate format with bigger data capacity to encode the string.

### 3.1.13 Width Property

Sets the width, in pixels, of the image which contains the Aztec barcode.

#### Property Data Type

short

#### Remarks

First call `GetActualSize()` method to obtain the actual size of the Aztec barcode, then use this property and Height property to set image size by adding surrounding white space around the Aztec barcode.

#### See Also

`GetActualSize()` Method

## 3.2 Methods

### 3.2.1 GetActualRC Method

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

```
void GetActualRC(VARIANT *ActualRows, VARIANT *ActualCols);
```

#### Parameters

*ActualRows*

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

*ActualCols*

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

#### Remarks

---



If you set PreferredFormat to 0 (Auto format), Aztec ASP 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, Aztec ASP 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.

### 3.2.2 GetActualSize Method

Gets the actual size of the Aztec barcode which is rendered onto computer screen.

```
void GetActualSize(VARIANT *ActualWidth, VARIANT *ActualHeight);
```

#### Parameters

##### *ActualWidth*

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

##### *ActualHeight*

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

### 3.2.3 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 Aztec 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 26, which means that up to 26 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.

## 4 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 Server License allows the use of the software (up to 10,000 users) on ONE server with ONE CPU in your organization.

\* The 2 Server License allows the use of the software (up to 10,000 users) on 2 servers (each server has only 1 CPU) in your organization.

\* The 3 Server License allows the use of the software (up to 10,000 users) on 3 servers (each server has only 1 CPU) in your organization.

\* The 4 Server License allows the use of the software (up to 10,000 users) on 4 servers (each server has only 1 CPU) in your organization.

\* The 5 Server License allows the use of the software (up to 10,000 users) on 5 servers (each server has only 1 CPU) in your organization.

\* The Unlimited Developer License allows the use of the software (unlimited number of users) on unlimited number of servers (each server has unlimited number of CPUs) in your organization.

#### 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.

---