



ExpoBadge®

Defining the Lead...

ExpoBadge Badge Data Standard

STANDARD (version ExpoBadge4.5) PDF417 ENCODING FORMAT FOR TRADE SHOWS

The following is the standard encoding format for registration service contractors and show managers wishing to encode PDF417 data on badges at trade shows. Companies who follow this standard encoding format will be assured of compatibility with off the shelf devices designed for use in the electronic lead collection and qualification process commonly used at trade shows. This standard encoding format is now regularly used by more than 15 different registration contractors and 30 show management companies at an estimated 200 events per year. This encoding format is, by far, the most widely used standard in the trade show industry.

Like any other standard this document changes from time to time and relies on the work of people in the industry who use the standard to report problems with the standard or suggestions for improving the standard. Your opinions count, so if you are using a device (printer) which uses any of the control characters in this standard as reserved words or you have difficulty getting your application to produce these control characters or any other technical difficulties for which you need to find a 'work-around', please don't keep it to yourself! Report it to us verbally at 800-490-9941, or fax us at 714-712-7351, or email us at channigan@expobadge.com. This document is also available on the EXPOBADGE website at www.ExpoBadge.Com

START SEQUENCE

1. Start each new record with a unique combination of non-printable characters which will signify the start of a record. Where {XXX} denotes a single decimal value

- The three byte string {002} {002} {002} is standard.

FIELD DELIMITER

2. Use a unique non-printable character to delimit the fields on each line. The one byte non-printable character {003} is the standard. The rule to remember here is if a field is the first characters on a line then there only need to be a delimiter after the field, but if the

field data does not begin the line then wrap the field data in the same delimiter. For example:

```
Suite 104 {003}
Carrollton {003} {003} TX {003} {003} 75007 {003}
```

If there is a case where there is more than 1 field on a line and the first field is blank then just put in the delimiter and close up the trailing space. For example:

```
Mr.{003} {003}Joe{003} {003}Smith{003}
```

becomes

```
{003}{003}Joe{003} {003}Smith{003}
```

END OF RECORD DELIMITER

3. To be absolutely certain of where the record ends it is mandatory to have an end of record character, or sequence, after the entire record has been printed.

- The one byte non-printable character {026} is recommended as an end of record delimiter.

DATA STREAM

The following is a sample output with the appropriate delimiters. In the following examples, {XXX} denotes a single decimal value that is encoded in the data stream.

```
{002}{002}{002}01234{003}Greg{003} {003}McMullen III{003}Director of
Technology Development{003}ARI, Inc{003}142 MacArthur Dr{003}Suite
104{003}Carrollton{003}, {003}TX{003} {003}75007{003}United States of
America{003}214/245-8600{003}214/245-
8700{003}greg_mcmullen@ari.com{003}{026}
```

Example #2: (Note: Address 2 information absent)

```
{002}{002}{002}01234{003}Greg{003} {003}McMullen III{003}Director of
Technology Development{003}ARI, Inc{003}142 MacArthur
Dr{003}{003}Carrollton{003}, {003}TX{003} {003}75007{003} United States of
America{003}214/245-4600{003}214/345-
8700{003}greg_mcmullen@ari.com{003}{026}
```

DATA STREAM (hex)

The following is a sample output with the appropriate delimiters. In the following examples, {XX} denotes a single hex value that is encoded in the data stream.

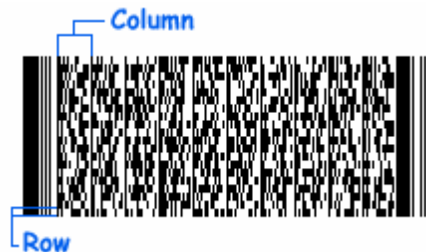
```
{02}{02}{02}01234{03}Greg{03} {03}McMullen III{03}Director of Technology  
Development{03}ARI, Inc{03}142 MacArthur Dr{03}Suite 104{03}Carrollton{03},  
{03}TX{03} {03}75007{03}United States of America{03}214/245-8600{03}214/245-  
8700{03}greg_mcmullen@ari.com{03}{1A}
```

Example #2: (Note: Address 2 information absent)

```
{02}{02}{02}01234{03}Greg{03} {03}McMullen III{03}Director of Technology  
Development{03}ARI, Inc{03}142 MacArthur Dr{03}{03}Carrollton{03},  
{03}TX{03} {03}75007{03}United States of America{03}214/245-4600{03}214/345-  
8700{03}greg_mcmullen@ari.com{03}{1A}
```

Parameters to set for the PDF417 DLL

The most important parameter to set, it is the number of rows and columns. Those will affect the size of the bar code. The number of columns will be the number you send to the DLL plus 2. The bar code below is the result when we set the number of rows to 20 and columns to 8. So, we have 8 columns plus 2 extra columns and 20 rows. This will create a bar code with the same dimension whatever the number of characters encoded. But there is a point you must pay a special attention to: The number of characters allowed will be approximately the number of rows multiplied by the number of columns. So in the example, we send 20 rows and 8 columns to the DLL. This will allow around 160 characters in the bar code. You must be sure to put the correct numbers corresponding to the maximum characters you can have in your bar code. We suggest using dynamic rows and 8 columns to accommodate a standard badge.



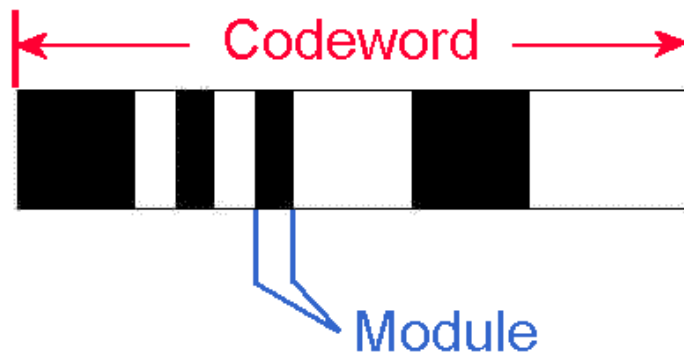
PDF417 - 2D barcode Ratio

What is the Ratio?

The barcode is composed of rows and columns. Each column has many rows so 1 row inside of a column is a Codeword composed of 17 vertical bars (white and black). Each bar is called a module. The height versus the width of one module is the ratio.



Above: 4:1 aspect ratio



EXPOBADGE Specifications

This ratio is very important for the scanner. Each type of scanner reacts a little bit differently to this ratio, but it is still critical for the scanning speed and capability. We want a ratio between 3:1 and 5:1. EXPOBADGE doesn't have a tool to measure the module but we can visually see a difference between each barcode ratio. The best ratio for EXPOBADGE equipment is 4:1.

Zebra ZPL Barcode Creation

It is suggested that for programming the Zebra printer that you use the following guidelines for formatting your string of text to send to the printer.

```
^XA
^FO200,0
^FB800,1,,
^FD{First + Last}^FS
^BY2,3
^FO300,200
^B7,7,5,,8,N
^FH^FD|02|02|02\&
{ID number}|03\&
{First}|03 |03{Last}|03\&
{Title}|03\&
{Company}|03\&
{Address1}|03\&
{Address2}|03\&
{City}|03 |03{ST}|03 |03{Zip}|03\&
{Country}|03\&
{Phone}|03\&
{Fax}|03\&
{email address}|03\& \&\&\&\&\&\&\&\&
|1A^FS
^FO500,0
^FB800,1,,C
^FD{ID number}^FS
^XZ
```

This will print the attendees name along with the PDF barcode and then the attendees ID number below. The command for setting up the PDF bar code is

```
^B7,7,5,,8,N
```

This will yield the most efficient width:height ratio for optimum scanning and will allow the Zebra printer to determine how many rows to print based upon the specified column amount of 8.