

^BX – Data Matrix Bar Code

Description The ^BX command creates a two-dimensional matrix symbology made up of square modules arranged within a perimeter finder pattern.

Format ^BXo,h,s,c,r,f,g,a

Parameters	Details
o = orientation	<p><i>Accepted Values:</i></p> <ul style="list-style-type: none"> N = normal R = rotated 90 degrees (clockwise) I = inverted 180 degrees B = read from bottom up, 270 degrees <p><i>Default Value:</i> current ^FW value</p>
h = dimensional height of individual symbol elements	<p><i>Accepted Values:</i> 1 to the width of the label</p> <p>The individual elements are square — this parameter specifies both module and row height. If this parameter is zero (or not given), the h parameter (bar height) in ^BY is used as the approximate symbol height.</p>
s = quality level	<p><i>Accepted Values:</i> 0, 50, 80, 100, 140, 200</p> <p><i>Default Value:</i> 0</p> <p><i>Quality</i> refers to the amount of data that is added to the symbol for error correction. The AIM specification refers to it as the ECC value. ECC 50, ECC 80, ECC 100, and ECC 140 use convolution encoding; ECC 200 uses Reed-Solomon encoding. For new applications, ECC 200 is recommended. ECC 000-140 should be used only in closed applications where a single party controls both the production and reading of the symbols and is responsible for overall system performance.</p>
c = columns to encode	<p><i>Accepted Values:</i> 9 to 49</p> <p>Odd values only for quality 0 to 140 (10 to 144); even values only for quality 200.</p> <p>Odd values only for quality 0 to 140 (10 to 144); even values only for quality 200. The number of rows and columns in the symbol is automatically determined. You might want to force the number of rows and columns to a larger value to achieve uniform symbol size. In the current implementation, quality 0 to 140 symbols are square, so the larger of the rows or columns supplied are used to force a symbol to that size. If you attempt to force the data into too small of a symbol, no symbol is printed. If a value greater than 49 is entered, the rows or columns value is set to zero and the size is determined normally. If an even value is entered, it generates INVALID-P (invalid parameter). If a value less than 9 but not 0, or if the data is too large for the forced size, no symbol prints; if ^CV is active, INVALID-L prints.</p>
r = rows to encode	<p><i>Accepted Values:</i> 9 to 49</p>

Parameters	Details
f = format ID (0 to 6) — not used with quality set at 200	<p><i>Accepted Values:</i></p> <ul style="list-style-type: none"> 1 = field data is numeric + space (0..9,") – No \&" 2 = field data is uppercase alphanumeric + space (A..Z,") – No \&" 3 = field data is uppercase alphanumeric + space, period, comma, dash, and slash (0..9,A..Z,“-/”) 4 = field data is upper-case alphanumeric + space (0..9,A..Z,") – no \&" 5 = field data is full 128 ASCII 7-bit set 6 = field data is full 256 ISO 8-bit set <p><i>Default Value:</i> 6</p>
g = escape sequence control character	<p><i>Accepted Values:</i> any character</p> <p><i>Default Value:</i> ~ (tilde)</p> <p>This parameter is used only if quality 200 is specified. It is the escape character for embedding special control sequences within the field data.</p> <p>Important • A value must always be specified when using the escape sequence control character. If no value is entered, the command is ignored.</p> <p>The g parameter will continue to be underscore (_) for anyone with firmware version: V60.13.0.12, V60.13.0.12Z, V60.13.0.12B, V60.13.0.12ZB, or later.</p>
a = aspect ratio	<p><i>Accepted Values:</i></p> <ul style="list-style-type: none"> 1 = square 2 = rectangular <p><i>Default Value:</i> 1</p>



The **a** parameter is only supported in V60.16.5Z and V53.16.5Z or later.

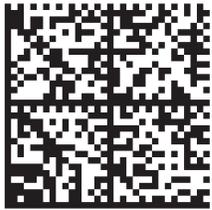
Table 10 •

ECC LEVEL	ID = 1	ID = 2	ID = 3	ID = 4	ID = 5	ID = 6
0	596	452	394	413	310	271
50	457	333	291	305	228	200
80	402	293	256	268	201	176
100	300	218	190	200	150	131
140	144	105	91	96	72	63

Maximum Field Sizes

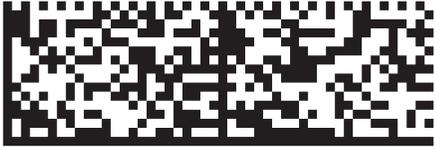


Example 1 • This is an example of a square Data Matrix bar code:

ZPL II CODE	DATA MATRIX BAR CODE
<pre> ^XA ^FO100,100 ^BXN,10,200 ^FDZEBRA TECHNOLOGIES CORPORATION 333 CORPORATE WOODS PARKWAY VERNON HILLS, IL 60061-3109^FS ^XZ </pre>	



Example 2 • This is an example of a rectangle Data Matrix bar code:

ZPL II CODE	DATA MATRIX BAR CODE
<pre> ^XA ^FO100,100 ^BXN,10,200,,,,,2 ^FDZEBRA TECHNOLOGIES CORPORATION 333 CORPORATE WOODS PARKWAY ^FS ^XZ </pre>	

Effects of ^BY on ^BX

w = **module width** (no effect)

r = **ratio** (no effect)

h = **height of symbol**

If the dimensions of individual symbol elements are not specified in the ^BY command, the height of symbol value is divided by the required rows/columns, rounded, limited to a minimum value of one, and used as the dimensions of individual symbol elements.

Field Data (^FD) for ^BX

Quality 000 to 140

- The \& and || can be used to insert carriage returns, line feeds, and the backslash, similar to the PDF417. Other characters in the control character range can be inserted only by using ^FH. Field data is limited to 596 characters for quality 0 to 140. Excess field data causes no symbol to print; if ^CV is active, INVALID-L prints. The field data must correspond to a user-specified format ID or no symbol prints; if ^CV is active, INVALID-C prints.
- The maximum field sizes for quality 0 to 140 symbols are shown in the table in the g parameter.

Quality 200

- If more than 3072 bytes are supplied as field data, it is truncated to 3072 bytes. This limits the maximum size of a numeric Data Matrix symbol to less than the 3116 numeric characters that the specification would allow. The maximum alphanumeric capacity is 2335 and the maximum 8-bit byte capacity is 1556.
- If ^FH is used, field hexadecimal processing takes place before the escape sequence processing described below.
- The underscore is the default escape sequence control character for quality 200 field data. A different escape sequence control character can be selected by using parameter g in the ^BX command.

The information that follows applies to firmware version: V60.13.0.12, V60.13.0.12Z, V60.13.0.12B, V60.13.0.12ZB, or later. The input string escape sequences can be embedded in quality 200 field data using the ASCII 95 underscore character (`_`) or the character entered in parameter g:

- `_X` is the shift character for control characters (e.g., `_@=NUL`, `_G=BEL`, `_0 is PAD`)
- `_1` to `_3` for FNC characters 1 to 3 (explicit FNC4, upper shift, is not allowed)
- FNC2 (Structured Append) must be followed by nine digits, composed of three-digit numbers with values between 1 and 254, that represent the symbol sequence and file identifier (for example, symbol 3 of 7 with file ID 1001 is represented by `_2214001001`)
- `5NNN` is code page NNN where NNN is a three-digit code page value (for example, Code Page 9 is represented by `_5009`)
- `_dNNN` creates ASCII decimal value NNN for a code word (must be three digits)
- `_` in data is encoded by `__` (two underscores)

The information that follows applies to all other versions of firmware. The input string escape sequences can be embedded in quality 200 field data using the ASCII 7E tilde character (`~`) or the character entered in parameter g:

- `~X` is the shift character for control characters (e.g., `~@=NUL`, `~G=BEL`, `~0 is PAD`)
- `~1` to `~3` for FNC characters 1 to 3 (explicit FNC4, upper shift, is not allowed)
- FNC2 (Structured Append) must be followed by nine digits, composed of three-digit numbers with values between 1 and 254, that represent the symbol sequence and file identifier (for example, symbol 3 of 7 with file ID 1001 is represented by `~2214001001`)
- `5NNN` is code page NNN where NNN is a three-digit code page value (for example, Code Page 9 is represented by `~5009`)
- `~dNNN` creates ASCII decimal value NNN for a code word (must be three digits)
- `~` in data is encoded by a `~` (tilde)