KYOCERA

PDF417
Two Dimensional
Bar Code
Implementation Manual

For Kyocera Page Printers

The information in this manual is subject to change without notification. Additional pages may be inserted in future editions. The user is asked to excuse any technical inaccuracies or typographical errors in the present edition.

No responsibility is assumed if accidents occur while the user is following the instructions in this manual. No responsibility is assumed for defects in the printer's firmware (contents of its read-only memory) as described herein.

Table of contents

PDF417 implementation manual

...as implemented in the Kyocera Ecosys printers.

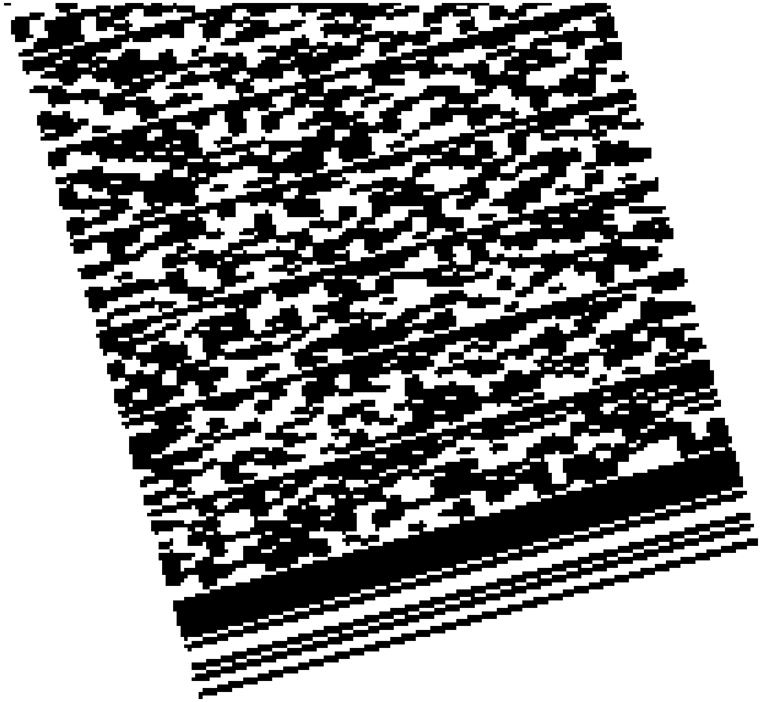
The Kyocera Ecosys printers include the capability that allows the user to implement the two-dimensional stacked bar code symbology, PDF 417, or Portable Data File 417. This expanded functionality is achieved by using PRESCRIBE language commands as explained in this manual.

This manual starts with taking a look at the overview for the PDF 417 symbology. As a whole the manual provides discussion on the following topics:

❒ PDF 417 overview

❒ PRESCRIBE commands for PDF 417

❒ Macro PDF 417



PDF417 overview

PDF 417 overview

PDF 417 is a two-dimensional stacked bar code symbology capable of encoding over a kilobyte of data per label. This is important for applications where a bar code must be more than merely an identifier, an index to reference a database.

The *portable data file* approach is well suited to applications where it is impractical to store item information in a database or where the database is not accessible when and where the item's bar code is read. Because a PDF417 symbol can store so much data, item data such as the content of a shipping manifest or equipment maintenance history can be carried on the item, without requiring access to a remote database.

Encoding data into a PDF417 bar code is a two-step process. First, data is converted into codeword values of 0-928, which represent the data. This is *high-level encoding*. Then the values are physically represented by particular bar/space patterns, which is *low-level encoding*. Decoding is the reverse process.

In addition, PDF417 is an error-correcting symbology designed for real -world applications where portions of labels can get destroyed in handling. It performs error correction by making calculations, if necessary, to reconstruct undecoded or corrupted portions of the symbol.

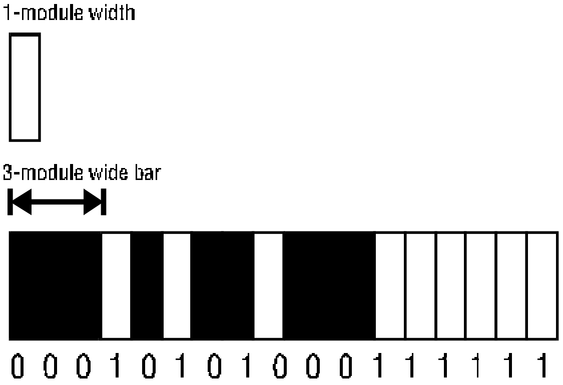
PDF417 symbol description

At first glance, a PDF417 symbol looks like a set of stacked bar codes. When we look closer to analyze how the symbol is put together, there are several key elements. These are rows, start patterns, stop patterns, codewords, and modules, as explained below.

Definitions

module The narrowest width of a bar or space in the bar code. All bars or spaces are multiples (up to six times) of this width. The nominal unit of measure.

codeword A single group of bars and spaces (or *elements*) representing one or more numbers, letters, or other symbols (i.e., codeword values for the data to be encoded). Each PDF417 codeword contains four bars and four spaces, for a total of 17 module widths. Each codeword starts with a bar and ends with a space. See figure below.

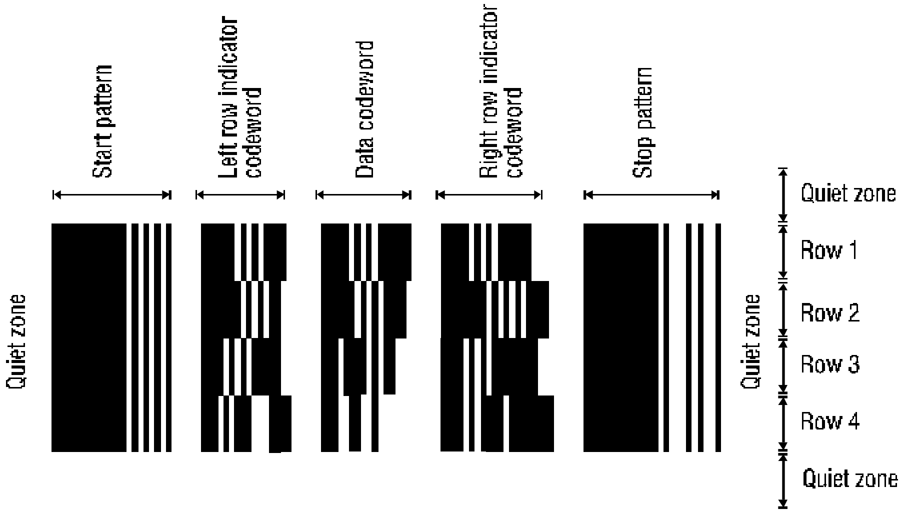


Definitions

start pattern A unique pattern of light and dark elements which indicates the leftmost part of a bar code label.

stop pattern A unique pattern of light and dark elements which indicates the rightmost part of a bar code label.

row A lateral set of elements made up of a start pattern, codewords, and a stop pattern. Each PDF417 symbol must have at least 3 rows. See figure below.

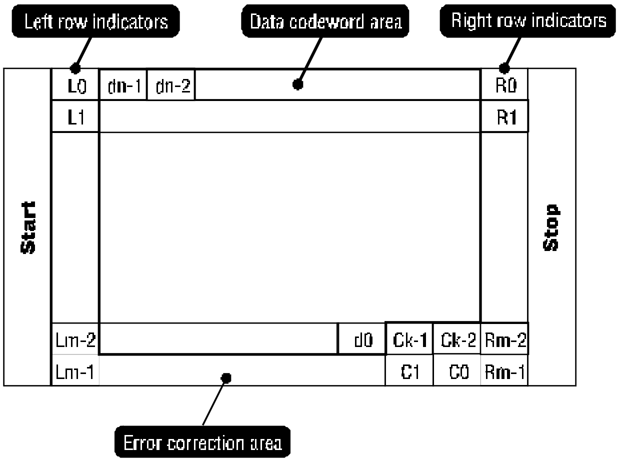


In each row, between left and right row indicators, there may be from 1 to 30 data codewords. Collectively, among all rows, these codewords form data columns.

Overall symbol structure

Any PDF417 symbol is made up of at least 3 rows, and at most 90 rows. The minimum number of codewords in a row is 3; this includes the left row indicator codeword, 1 data codeword, and the right row indicator.

Every symbol contains 1 codeword (the first data codeword in row 0) indicating the total number of data codewords within the symbol, and at least 2 error-detection codewords. General PDF417 symbol structure is indicated in the following diagram:



Printing a two-dimensional bar code

In the Kyocera printers, PRESCRIBE XBAR, XBCP, XBUF, and ENDB commands are introduced in order to support the PDF417 two dimensional bar code symbology. Each command does the following to put data together for printing a bar code.

Printer commands	Function
XBAR	Prints a two dimensional bar code from the given data string. Must be followed by an ENDB command (See below.).
XBCP	Specifies various options for the bar code to be printed using XBAR depending on the *mode* following the command (0 through 19).
XBUF	Defines a buffer name for a data input for XBAR command. See details in Appendix.
ENDB	Terminates a XBAR command sequence.

These commands are detailed in the next chapter, *PRESCRIBE commands for PDF417*.

Positioning the bar codes

The following commands are positioning commands the PRESCRIBE language provides. These commands are useful and sometimes essential for proper positioning of the bar codes onto a medium such as label. More detailed discussion on these (and some other) positioning commands can be found in the printer's manual.

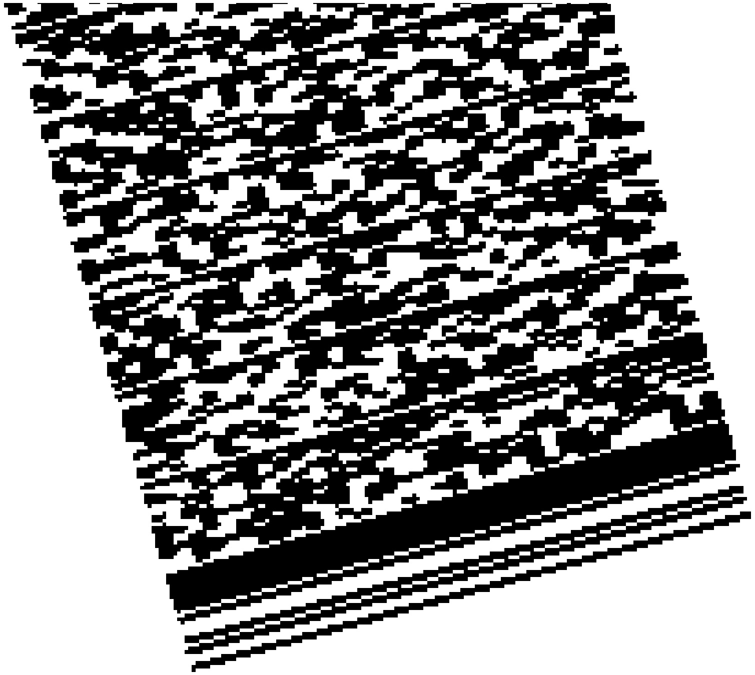
Printer commands	Function
MAP	Moves the cursor to a position relative to the top and left margins.
MRP	Moves the cursor from the current to a specified relative position.
MZP	Moves the cursor to a position relative to the top and left edge limits of the page.
UNIT	Sets the unit of measurement used in the PRESCRIBE commands including the above. The initial unit is inches. The other units are related to inches as: 1 inch=2.54 centimeters=72 points=300 or 600 dots (depending on the printer model).

By default the cursor is located at the top left corner of the bar code and stays there after printing is done.

PDF417 characteristics

Item	Description
Encodable character set	256 international characters and binary data
Code type	Continuous
Character self checking	Yes
Bi-directional decoding	Yes
Number of row indicator codewords required per row	2
Number of symbol checksum codewords required	2
Minimum number of rows per symbol	3
Maximum number of rows per symbol	90
Minimum number of data columns	1
Maximum number of data columns	30
Number of symbol length descriptors required	1
Smallest nominal element width	0.0075 inch or 0.191 mm
Smallest nominal element height	0.01 inch or 0.254 mm
Maximum data character per symbol	Assuming 928 codewords - 1 symbol length descriptor - 2 symbol checksum codewords = 925 data codewords. Binary/ASCII plus mode: 1108 bytes. Extended alphanumeric compaction mode (EXC): 1850 ASCII characters. Numeric compaction mode: 2725 digits.

For Macro PDF417, which transparently distributes information among a number of PDF417 symbols, the above storage limits are increased to more than one million bytes in Binary/ASCII Plus mode and more than 2.5 million bytes in EXC mode.



PRESCRIBE commands for PDF417

Using PRESCRIBE commands for PDF417

The XBAR, XBCP, XBUF, and ENDB commands are used to print a PDF417 two dimensional bar code. In the command format descriptions below, the command is written in `CAPITAL` letters. Parameters are indicated in lowercase *italics* . Optional parameters, which may be omitted, are enclosed in brackets []. Except for the initial !R!, PRESCRIBE commands may be written in lowercase letters if preferred.

XBAR

Print a two dimensional bar code

Format XBAR [*data_length*]; *data_string*;

Parameters *data_length* : length of data in bytes, terminated with a semicolon

data_string : data to be encoded, terminated with a semicolon immediately followed by an ENDB command (256 international characters or binary data)

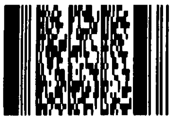
Function

This command prints a two dimensional PDF417 bar code. ENDB; must follow as a terminator for the bar code data string. Note that carriage return, line feed, space and all other characters must be counted as part of data in the data string field. If *data_length* is omitted, *0* , negative value, or non integer, then *data_string* is counted all characters until the program sees ;ENDB; . It is necessary to have no space between “; ” and “ENDB; ”.

Examples

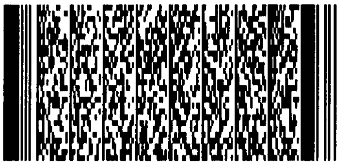
```
!R! RES;
XBCP 0;
XBAR 19;This is a XBAR test;ENDB;
EXIT;
```

Explanation : The first line puts the printer into PRESCRIBE mode and resets the printer’s temporary parameters to the defaults. The second line XBCP 0 resets the PDF417 bar code options to the defaults. Parameter 19 following XBAR is optional and counts the length of the ensuing string (19 bytes). The string must be terminated by ;ENDB; with no space placed inbetween. The EXIT; command ends PRESCRIBE mode and brings the printer back to normal printing mode. These 4 lines obtain the following printout when sent to the printer.



```
!R! RES;
XBCP 0;
MZP 1.5, 1.5;
XBAR;Visit Sea World, the world famous San Diego Zoo &
Wild Animal Park, historic Old Town, Tijuana (Mexico),
Mission Bay, the downtown Gaslamp District and all
those great beaches. And, as the natives know, the end
of September is the best time of year.;ENDB;
EXIT;
```

Explanation : In the third line, the MZP command moves the cursor to the point 1.5 inches from the left edge of limit of paper and 1.5 inches from the top edge limit of paper, where the two dimensional bar code starts at its top left corner. This example prints the following result.



ENDB

End a two dimensional bar code string

Format `ENDB;`

Parameters None

Function

The ENDB command terminates a bar code data string. The program counts all characters or binary data in the data string until it encounters "`;ENDB;` ". It is necessary to have no space between "`;` " and "`ENDB;` ". See XBAR above for an example of ENDB.

XBCP

Specify the two dimensional bar code options [General format of XBCP]

Format `XBCP` *mode* `[,`*parameter 1* `[,` *parameter 2* `]];`

Parameters
mode : integer between 0 and 19, as explained on the following pages
parameter(s) : new value(s) for the option, as also explained on the following pages.

Function

This command specifies the two dimensional bar code options. As of today, only PDF417 encoding options and its Macro PDF options are supported. *mode* identifies the option being set as listed below and explained on the following pages. *parameter* holds the new value for the option. Note that the options set by XBCP are valid until a RES command, a XBCP 0 command or until changed by the same XBCP command itself.

PDF417 two-dimensional bar code modes

XBCP mode	Meaning
XBCP 0	Reset
XBCP 1	Narrowest element width
XBCP 2	Error correction (by percentage)
XBCP 3	Error correction (by predetermined level)
XBCP 4	Number of rows

XBCP 5	Number of columns
XBCP 6	Aspect ratio of height and width
XBCP 7	Bar height by a ratio of element width
XBCP 8	Number of rows and columns (XBCP 4 and 5)
XBCP 9	Truncation
XBCP 10 through 19	These modes give additional control options used to support Macro PDF417 bar codes. See more details in section Macro PDF417.

XBCP 0

Select a bar code type/Reset all other XBCP parameters to the defaults

Format `XBCP 0[,` *type*`];`

Parameters *type* : two dimensional bar code type, must be 1, meaning PDF417 (only type currently supported)

Function

The XBCP command set to mode 0 selects a bar code type and resets all parameters to the defaults.

See XBAR 1 below for an example of XBCP 0.

XBCP 1

Specify the narrowest element width

Format `XBCP 1,` *element_width* `;`

Parameters *element_width* : narrowest bar and space width (Default is 3 dots.)

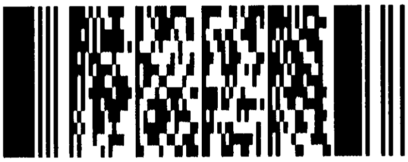
Function

PDF417 supports 8 different bar and space widths. The each width is a multiple of 1 to 8 times of the narrowest element width. The XBCP command set to mode 1 specifies the narrowest element width (narrowest bar width) in the current unit as designated by a UNIT command.

Example

```
!R! RES;
MZP 1.5, 1.5;
XBCP 1, .02;
XBAR;Min. element width is .02 inches;ENDB;
MZP 1.5, 2.5;
XBCP 0;
XBAR;Min. element width is reset (to 3 dots);ENDB;
EXIT;
```

Explanation : The element width is set to .02 inches as the default unit is inches. Minimum width allowed by the printer is 1 dot. Minimum readable width is probably higher. Minimum width meeting AIM recommendation for open systems is 3 dots. After the first example has been printed, the printer moves cursor 1 inch down to print another example in which the minimum element width is reverted to its default (3 dots).



Minimum element width is 0.02 inches.

Default minimum element width

XBCP 2

Specify the error correction level by percentage

Format XBCP 2, *percentage* ;

Parameters *percentage* : value between 0 and 400 (An illegal value defaults to 10.)

Function

PDF417 supports an error correction scheme. The XBCP command set to mode 2 specifies the error correction level as a percentage.

Note that XBCP 3, as explained below, sets the error correction level by predetermined values, and the execution of XBCP 3 disables the setting of XBCP 2 even if XBCP 2 is executed after XBCP 3, until RES or XBCP 0.

Example

To set 50% error correction level:

```
!R! RES;
XBCP 2, 50;
XBAR;This is an error correction level test;ENDB;
EXIT;
```

XBCP 3

Set the error correction level by predetermined values

Format XBCP 3, *error_corr_level*;

Parameters *err_corr_level* : integer between 0 and 8 (An illegal value defaults to 0.)

Function

The XBCP command in mode 3 sets the error correction level by predetermined values. Note that the execution of XBCP 3 overrides the setting of XBCP 2 (See above.) and remains valid until RES or XBCP 0.

Example

To sets the error correction level to level 4:

```
!R! RES;
XBCP 3, 4;
XBAR;Error correction can also be set by level;ENDB;
EXIT;
```

XBCP 4

Set the number of rows in the PDF bar code

Format `XBCP 4, *num*;`

Parameters *num*: number of rows, ranges between 3 and 90 (Illegal values are ignored.)

Function

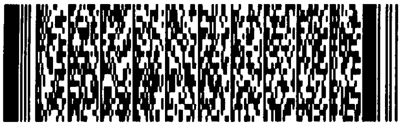
The XBCP command in mode 4 sets number of rows in the PDF417 bar code. XBCP 6 default provides default.

Note that this command is not effective if XBCP 8 is issued. Precedence of XBCP modes is: 4<8, 4>6, 4+8< 6.

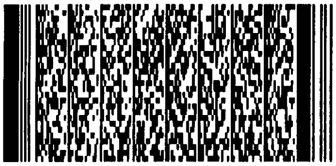
Example

To set 20 rows of bar code:

```
!R! RES;
XBCP 4, 20;
MZP 1.5, 1.5;
XBAR;Visit Sea World, the world famous San Diego Zoo &
Wild Animal Park, historic Old Town, Tijuana (Mexico),
Mission Bay, the downtown Gaslamp District and all
those great beaches. And, as the natives know, the end
of September is the best time of year.;ENDB;
EXIT;
```



Number of rows is set to 20.



Number of rows is normal.

The second bar code sample above represents the normal number of rows for comparison.

XBCP 5

Set the number of data code word columns in the PDF417 bar code

Format `XBCP 5,` *`num`*`;`

Parameters	*num*:	number of data code word columns, ranges between 1 and 30 inclusive (Illegal values are ignored.)

Function

The XBCP command in mode 5 sets the number of data codeword columns in the PDF417 bar code. There are also row indicator codewords at each end of the row which are not counted as data codewords. XBCP 6 default provides default.

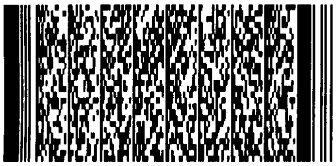
Note that this command is not effective if XBCP 8 is issued. Precedence of XBCP modes is: 5<8, 5>6, 5+8>6.

Example

To set 30 columns of bar code, command:

```
!R! RES;
XBCP 5, 30;
MZP 1.5, 1.5;
XBAR;Visit Sea World, the world famous San Diego Zoo &
Wild Animal Park, historic Old Town, Tijuana (Mexico),
Mission Bay, the downtown Gaslamp District and all
those great beaches. And, as the natives know, the end
of September is the best time of year.;ENDB;
EXIT;
```

Number of columns is set to 30.



This second bar code sample represents the normal number of columns for

XBCP 6

Determines aspect ratio of vertical height and horizontal width of the bar code

Format `XBCP 6,` *ver, hor*`;`

Parameters *ver /hor:* parts of the aspect ratio: vertical and horizontal, ranging between 10:1 (XBCP 6, 10, 1) and 1:10 (XBCP 6, 1, 10) (Illegal values default to 1:2. Default is 1:2.)

Function

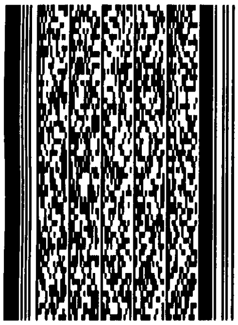
The XBCP command in mode 6 determines aspect ratio of vertical height and horizontal width of the bar code.

Note that this command is not effective if XBCP 4 and/or 5 is issued, however, this has the highest priority if both XBCP 8 and XBCP 4 and/or 5 are issued together. Precedence of XBCP modes is: 4/5<8, 4/5>6, 4/5+8<6.

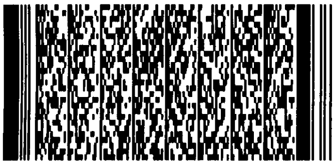
Example

To set aspect ratio 3:2 in vertical and horizontal direction, command as follows:

```
!R! RES;
XBCP 6, 3, 2;
XBAR;Visit Sea World, the world famous San Diego Zoo &
Wild Animal Park, historic Old Town, Tijuana (Mexico),
Mission Bay, the downtown Gaslamp District and all
those great beaches. And, as the natives know, the end
of September is the best time of year.;ENDB;
EXIT;
```



Aspect ratio is 3:2 in vertical and horizontal directions.



This second bar code sample represents the normal aspect ratio for comparison.

XBCP 7

Determines the bar height by a ratio to the narrowest element width

Format XBCP 7, *mod_ratio* ;

Parameters *mod_ratio* : ratio of module height to element width, ranging from 1 to 10 (An illegal value defaults to 3.)

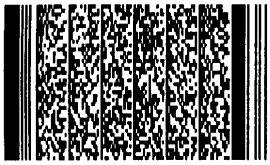
Function

PDF417 determines the bar height by a ratio to the narrowest element width. The XBCP command in mode 7 sets this bar height ratio. For discussion on the definition of “module,” refer to page 7.

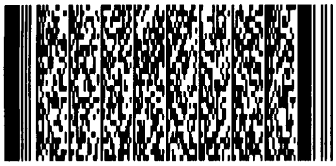
Example

To set the module height twice as high as the element width, command as follows:

```
!R! RES;
XBCP 7, 2;
XBAR;Visit Sea World, the world famous San Diego Zoo &
Wild Animal Park, historic Old Town, Tijuana (Mexico),
Mission Bay, the downtown Gaslamp District and all
those great beaches. And, as the natives know, the end
of September is the best time of year.;ENDB;
EXIT;
```



The bar height is twice as high as the element width. Since the default bar height is a triple of the element width, the bar code aspect ratio is now in ratio of 2:3. (See XBCP 6.)



Original (default) bar code for comparison

XBCP 8

Permit the system to automatically set rows and columns

Format `XBCP 8;`

Parameters None

Function

PDF417 sets number of rows and columns from XBCP 4 and XBCP 5. The XBCP command in mode 8 permits the system to automatically set rows and columns so that it satisfies the requirement that the multiplication of rows times columns is less than or equal to 928. Default is disabled.

Note that this command overrides XBCP 4 and/or 5, however, this command is not effective if both XBCP 6 and XBCP 4 and/or 5 are issued together. Precedence of XBCP modes is: 4/5<8, 4/5>6, 4/5+8<6.

XBCP 9

Enable the truncation

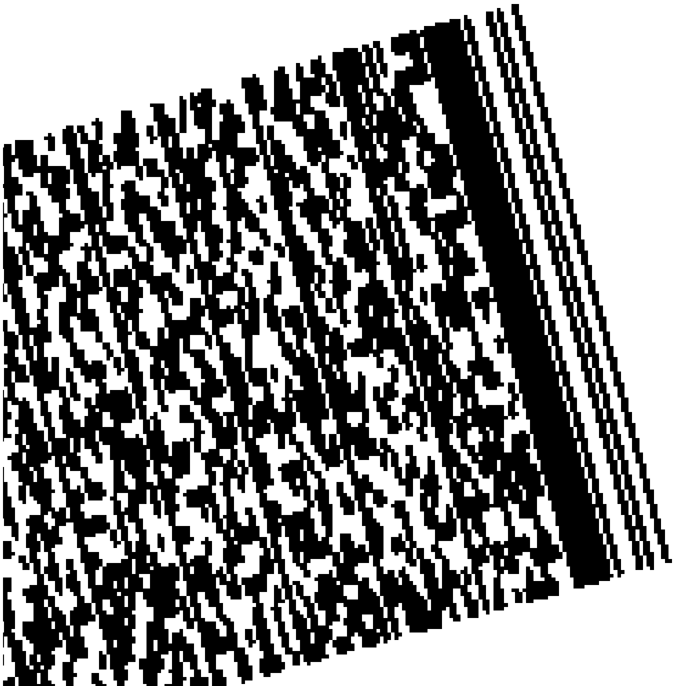
Format `XBCP 9;`

Parameters None

Function

PDF417 can generate a truncated PDF symbol that omits the right row indicator and the stop pattern. The XBCP command in mode 9 enables the truncation. Default is non-truncation.

Macro PDF417



Macro PDF417

Macro PDF417 provides a powerful mechanism for creating a distributed representation of files too large to be presented by a single PDF417 bar code. Macro PDF417 bar codes differ from ordinary PDF417 bar codes in that they contain additional control options which are added by modes 10 through 19 of the XBCP command. This allows a reader to make use of this information to correctly reconstruct and verify the file, independent of the bar code scanning order.

Note The terminology Prescribe Macro and Macro PDF are not same. Refer the printer manual for the Prescribe Macro.

The following modes of XBCP are used for Macro PDF417 to represent additional control options for XBAR.

XBCP modes for Macro PDF control options

XBCP mode	Meaning
XBCP 10	File name
XBCP 11	Block count
XBCP 12	Time stamp
XBCP 13	Sender ID
XBCP 14	Addressee ID
XBCP 15	File size
XBCP 16	Check sum
XBCP 17	File ID
XBCP 18	Macro PDF417 execution
XBCP 19	Distributed bar codes positioning

Note that implementation of these parameters are optional except XBCP 17 and XBCP 18.

Creating a Macro PDF417 representation

A Macro PDF417 creation begins with giving “fileid” using XBCP 17 for the ensuing Macro PDF417 sequences. Each one XBAR command sequence is needed for each divided bar code which is succeeded by a XBCP 18 statement that defines a separate *block index* for each bar code. The block index is needed to ensure that the divided bar codes are reconstructed in the correct order as the whole file when the bar codes are read.

The basic command sequence for Macro PDF417 is as follows.

```
...
XBCP 17, "fileid";          Gives the same file ID to all distributed bar codes.
MZP x1, y1;                 Position the first bar code at (x1, y1).
XBCP 18, 0;                 Block index for the first bar code.
XBAR; data_string;ENDB;     Encode and print the first divided bar code.
MZP x2, y2;                 Position the second bar code at (x2, y2).
XBCP 18, 1;                 Assign block index 1 to the second bar code.
XBAR; data_string;ENDB;     Encode and print the second divided bar code.
MZP x3, y3;                 Position the second bar code at (x3, y3).
XBCP 18, 2;                 Assign block index 2 to the second bar code;
...
```

The largest allowed block index is 99,998. Thus, up to 99,999 Macro PDF417 bar codes may comprise the distributed representation of a data file.

Explanation on each XBCP mode for Macro PDF417 follows. Some samples for Macro PDF417 are provided at the end of this chapter.

XBCP 10

Enable the file name field and define its contents for Macro PDF417

Format `XBCP 10,` *file_name* `;`

Parameters *file_name* : string containing a file name

Function

A Macro PDF symbol contains an optional file name field. This command enables this field and defines its contents.

The string must be surrounded by single or double quote marks, be less than 200 characters long and only contain EXC mode compaction characters (20H~7FH). This command is ignored if the string does not satisfy these conditions. The execution of this command automatically sets Macro PDF symbol mode.

Example

```
!R! RES;
XBCP 10, 'A00254';
MZP 1,0; XBCP 18,0;
XBAR; ...
...
EXIT;
```

XBCP 11

Enable the field and define its contents

Format `XBCP 11;`

Parameters None

Function

A Macro PDF symbol contains an optional block count field. This command enables this function. Default is not enabled.

The execution of this command also automatically sets Macro PDF symbol mode.

Example

```
!R! RES;
XBCP 11;
MZP 1,0; XBCP 18,0;
XBAR; ...
...
EXIT;
```

XBCP 12

Enable the time stamp field in Macro PDF

Format `XBCP 12,` *time*`;`

Parameters *time* : total number of seconds since 00:00:00AM January 1, 1970 Greenwich standard time

Function

A Macro PDF symbol contains an optional time stamp field. The XBCP command in mode 12 enables this function. The execution of this command also automatically sets Macro PDF symbol mode.

Example

To set time stamp at 06:15:30AM on Jan. 2, 1970, command:

```
!R! RES;
XBCP 12, 108930;
MZP 1,0; XBCP 18,0;
XBAR; ...
...
EXIT;
```

XBCP 13

Enable the sender ID field in Macro PDF

Format XBCP 13, *sender_ID* ;

Parameters *sender_ID* : string containing sender ID

Function

A Macro PDF symbol contains an optional Sender ID field. This command enables this field and defines its contents.

The string must be surrounded by single or double quote marks, be less than 200 characters long and only contain EXC compaction mode characters (20H~7FH). This command is ignored if the string does not satisfy these conditions. The execution of this command automatically sets Macro PDF symbol mode.

Example

```
!R! RES;
XBCP 13, "BETTERDAYSINC"
MZP 1,0; XBCP 18,0;
XBAR; ...
...
EXIT;
```

XBCP 14

Enable the addressee ID field in Macro PDF

Format XBCP 14, *addressee_ID* ;

Parameters *addressee_ID* : string containing addressee ID

Function

A Macro PDF symbol contains an optional Addressee ID field. The XBCP command in mode 14 enables this field and defines its contents.

The string must be surrounded by single or double quote marks, be less than 200 characters long and only contain EXC compaction mode characters (20H~7FH). This command is ignored if the string does not satisfy these conditions. The execution of this command also automatically sets Macro PDF symbol mode.

Example

```
!R! RES;
XBCP 14, 'BAREFOOTINC';
MZP 1,0; XBCP 18,0;
XBAR; ...
...
EXIT;
```

XBCP 15

Enable the file size field in Macro PDF

Format `XBCP 15;`

Parameters None

Function

A Macro PDF symbol contains an optional file size field. The XBCP command in mode 15 enables this function. Note that the execution of this command also automatically sets Macro PDF symbol mode. This function is disabled in default condition.

Example

```
!R! RES;
XBCP 15;
MZP 1,0; XBCP 18,0;
XBAR; ...
...
EXIT;
```

XBCP 16

Enable the check sum in Macro PDF

Format `XBCP 16;`

Parameters None

Function

A Macro PDF symbol contains an optional check sum. This command enables this function. Note that the execution of this command automatically sets Macro PDF symbol mode. This function is disabled in default condition.

Example

```
!R! RES;
XBCP 16;
MZP 1, 0; XBCP 18, 0;
```

```
XBAR; ...
EXIT;
```

XBCP 17

Allows control of file ID for Macro PDF417

Format `XBCP 17, file_ID;`

Parameters *file_ID* : string containing a file ID

Function

A Macro PDF symbol contains a required file ID. This command allows control of this field. A random file ID is generated when XBCP 10 through 16 or XBCP 18 are executed without executing XBCP 17.

The string must be surrounded by single or double quote marks, be less than 200 characters long and only contain EXC compaction mode characters (20H~7FH). This command is ignored if the string does not satisfy these conditions. The execution of this command automatically sets Macro PDF symbol mode.

Example

```
!R! RES;
XBCP 17, XMD1503;
MZP 1,0; XBCP 18,0;
XBAR; ...
...
EXIT;
```

XBCP 18

Enable Macro PDF417 symbol mode

Format `XBCP 18[, block index];`

Parameters *block index*: 0 to k-1, and k is the number of blocks

Function

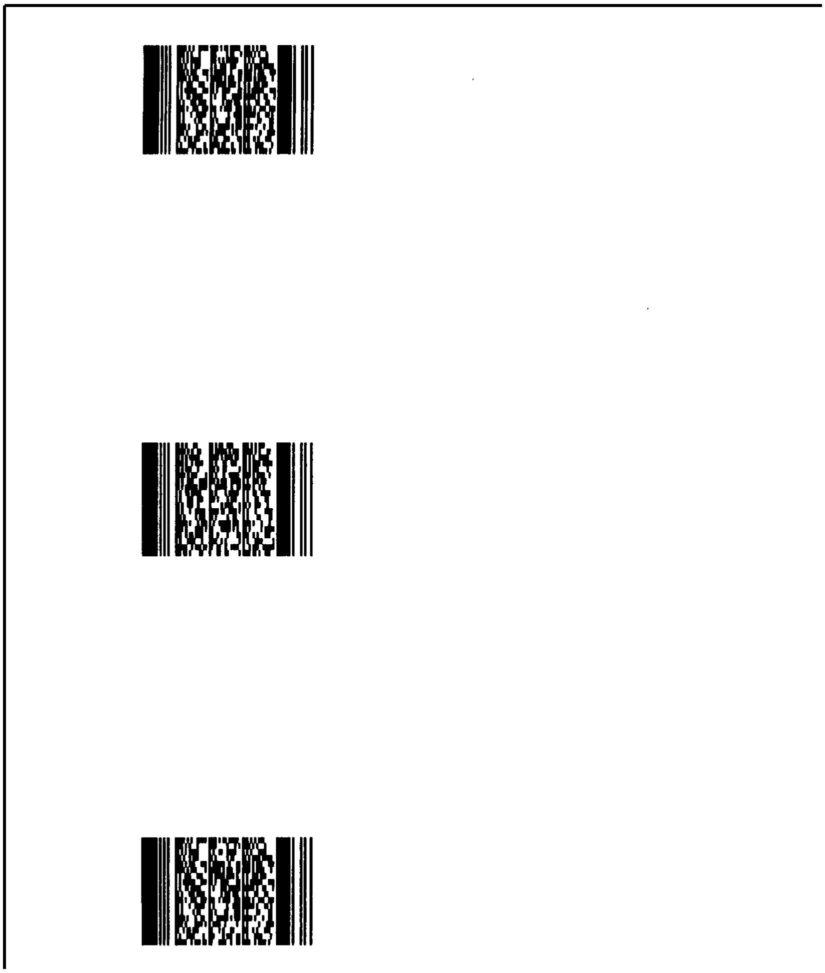
The Macro PDF symbol mode is enabled by executing this command. The Macro PDF symbol mode directs the reader to separate large data and allows multiple PDF two dimensional bar codes with a single "fileid".

Example

To bar-code data “abcdXYZ12345” as a Macro PDF417 using block indices 0, 1, and 2:

```
!R!
UNIT C;      CMNT Set unit to centimeters;
XBCP 17,"fileid"; CMNT Set file ID to fileid;
MZP 1,0;     CMNT Move cursor to (1, 0) [centimeters];
XBCP 18,0;   CMNT Set block index 0 to next data;
XBAR; abcd;ENDB;  CMNT Encode and print data "abcd";
MZP 1,5;     CMNT Move cursor to (1, 5) [centimeters];
XBCP 18,1;   CMNT Set block index 1 to next data;
XBAR; XYZ;ENDB;   CMNT Encode and print data "XYZ";
MZP 1,10;    CMNT Move cursor to (1, 10) [centimeters];
XBCP 18,2;   CMNT Set block index 2 to next data;
XBAR;12345;ENDB;  CMNT Encode and print data "12345";
EXIT;
```

Example



XBCP 19

Position multiple Macro PDF417 symbol at the specified locations

Format `XBCP 19[, x1, y1, x2, y2, x3, y3, ...];`

Parameters		
	x1, y1, x2, y2, x3, x3, ...	x and y coordinates that represent the positions of the divided bar codes measured from the current cursor position

Function

The multiple PDF 417 symbol from one large data can be positioned at the specified locations. *X1* , *Y1* is the relative location of second PDF symbol to the first symbol, *X*2 , *Y2* for third, and so on as measured from the current coordinate (like the MRP command described on the printer's manual). The remaining pairs of positioning parameters are ignored if there are no more PDF symbol to print. The relative location parameters, *X*n and *Y*n can not define the location beyond first page. The positioning command such as MZP and ENDB should be used to specify the macro PDF location if there are data beyond first page. PDF symbol will be positioned a half inch below the previous PDF symbol if there are no positioning parameter exist or insufficient number of positioning parameters exist. The combination with multiple number of ENDB command allows to print multiple number of macro PDF on multiple pages. Note RES, XBCP0, and XBAR command will delete the remaining macro PDF data if all data has not been printed. Therefore, the page eject should not be done by RES command.

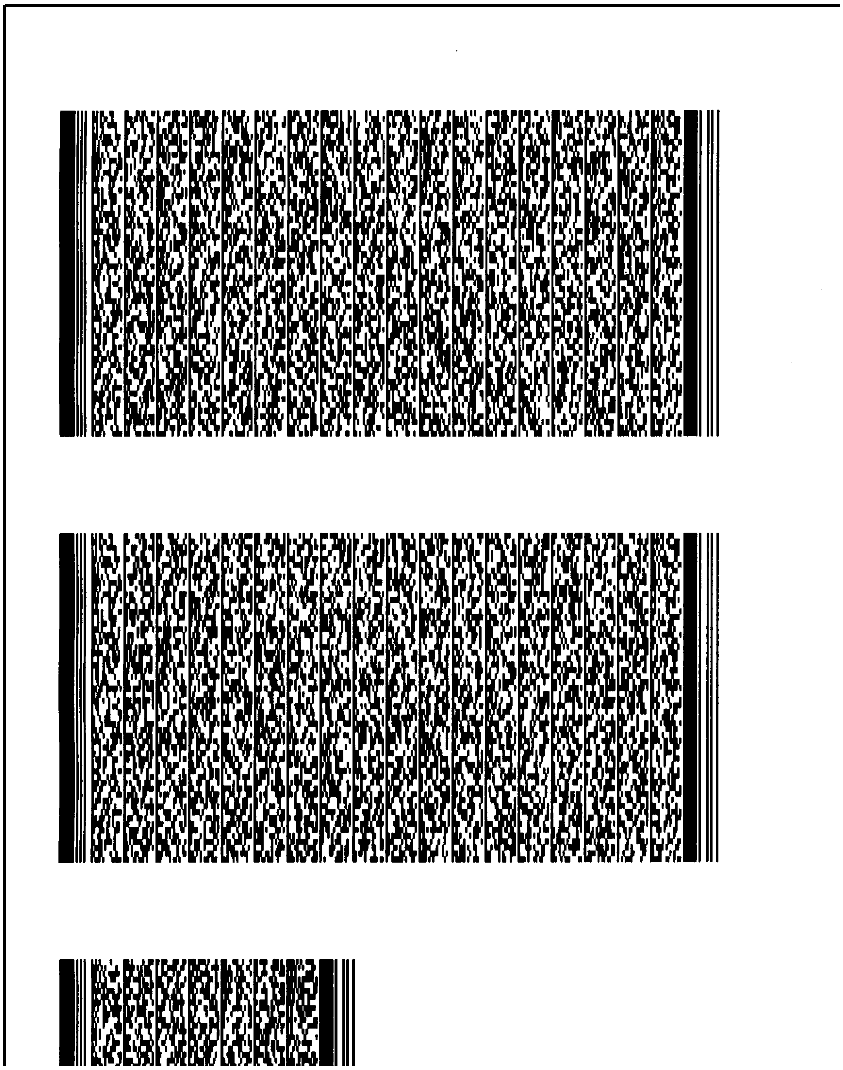
See the samples on the following pages.

Example 1

In Macro PDF mode, data such as 3000 bytes print 3 90-column-by-30 -row PDF symbols a half inch apart vertically:

```
!R! RES;
XBCP 17,"BookOne";
XBAR; large data such as 3000 bytes;ENDB;
EXIT;
```

Explanation : Execution of XBCP 17 automatically sets Macro PDF mode. Note that PDF bar code is positioned a half inch below the previous PDF bar code if there are no positioning parameters exist or insufficient number of parameters exist.

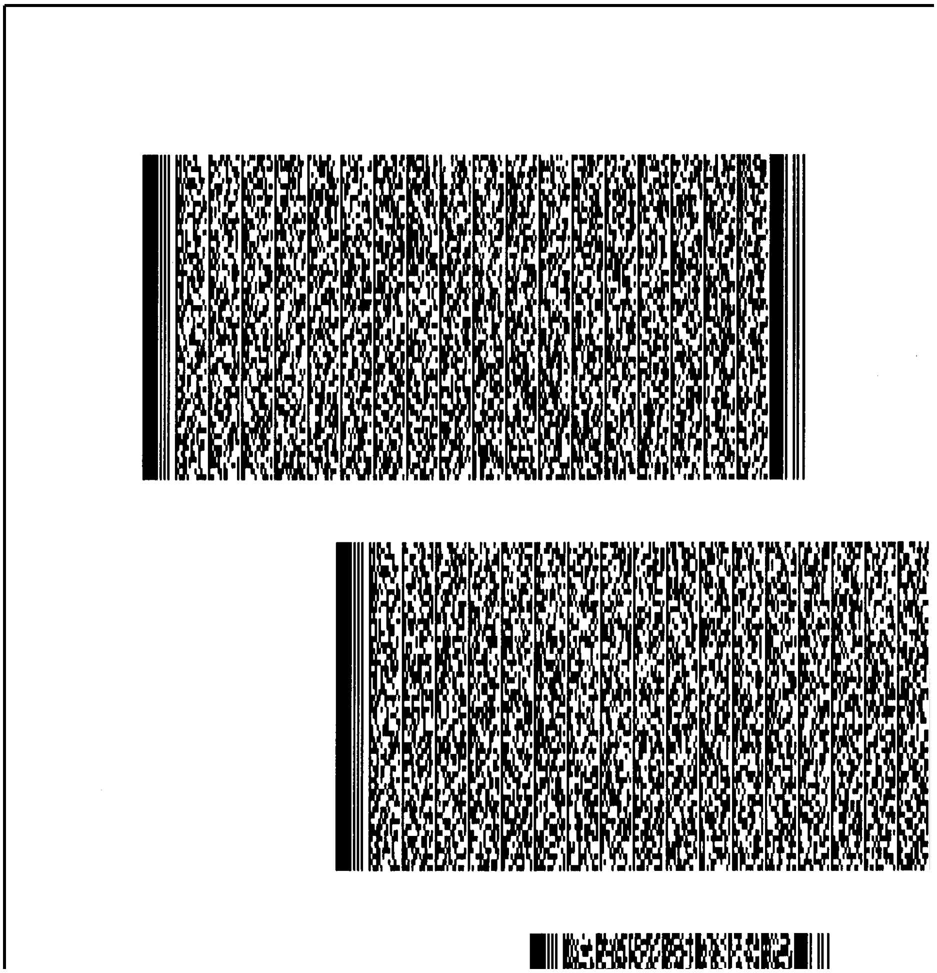


Example 2

In Macro PDF mode, data such as large as 3000 bytes print 3 90-column -by-30-row PDF symbols at specified locations specified by XBCP 19:

```
!R! RES;
XBCP 17,"QualitySt";
MZP 1, 1;
XBCP 19,1,2,1,2;
XBAR; large data such as 3000 bytes;ENDB;
EXIT;
```

Explanation : The second line XBCP 17 activates Macro PDF mode and specifies *QualitySt* for *fileid* . The third line MZP command positions the first bar code at 1 inch each from the left and right paper edge limits. The fourth line XBCP 19 moves the cursor 1 inch from the left and 2 inches below the point where the first bar code started and prints the second bar code; then moves the cursor 1 inch from the left and 2 inches below the point where the second bar code started and prints the third bar code.

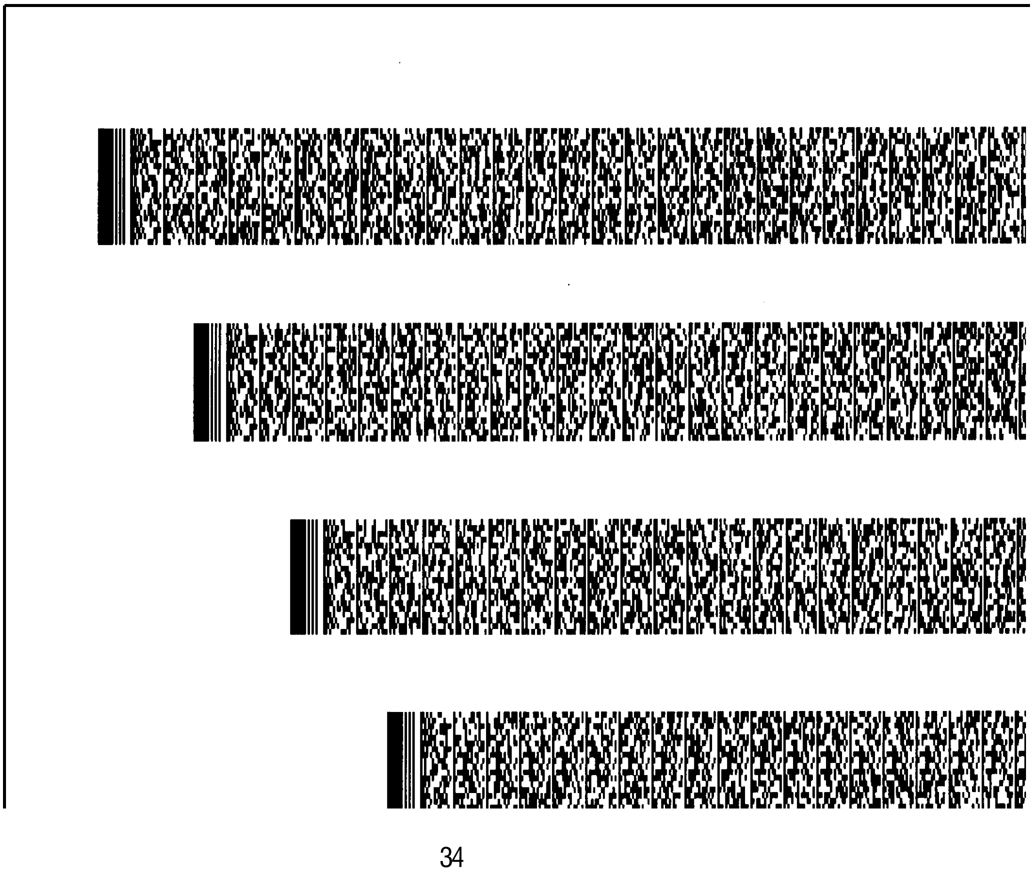


Example 3

To print 4 30-column-by-20-row PDF symbols at specified locations in Macro PDF mode:

```
!R! RES;
XBCP 0,1;
XBCP 4,20;
XBCP 5,30;
XBCP 17,"TrnOvr9";
MZP 0.5, 0.5;
XBCP 19,0.5,1,0.5,1,0.5,1,0.5,1;
XBAR; large data such as 3000 bytes;ENDB;
EXIT;
```

Explanation : The first line XBCP command resets all XBCP parameters to the defaults and selects the bar code type for PDF417 (though it is the only type currently available); the second line XBCP sets the number of rows in the bar code to 20 (rows); the third line XBCP sets the number of data codeword in the bar code to 30 (codewords). Note that with the number of rows and codewords of 20 and 30 respectively, the data the XBAR encodes are split into 4 bar codes which can be positioned separately at different coordinates as specified by the seventh line XBCP command.

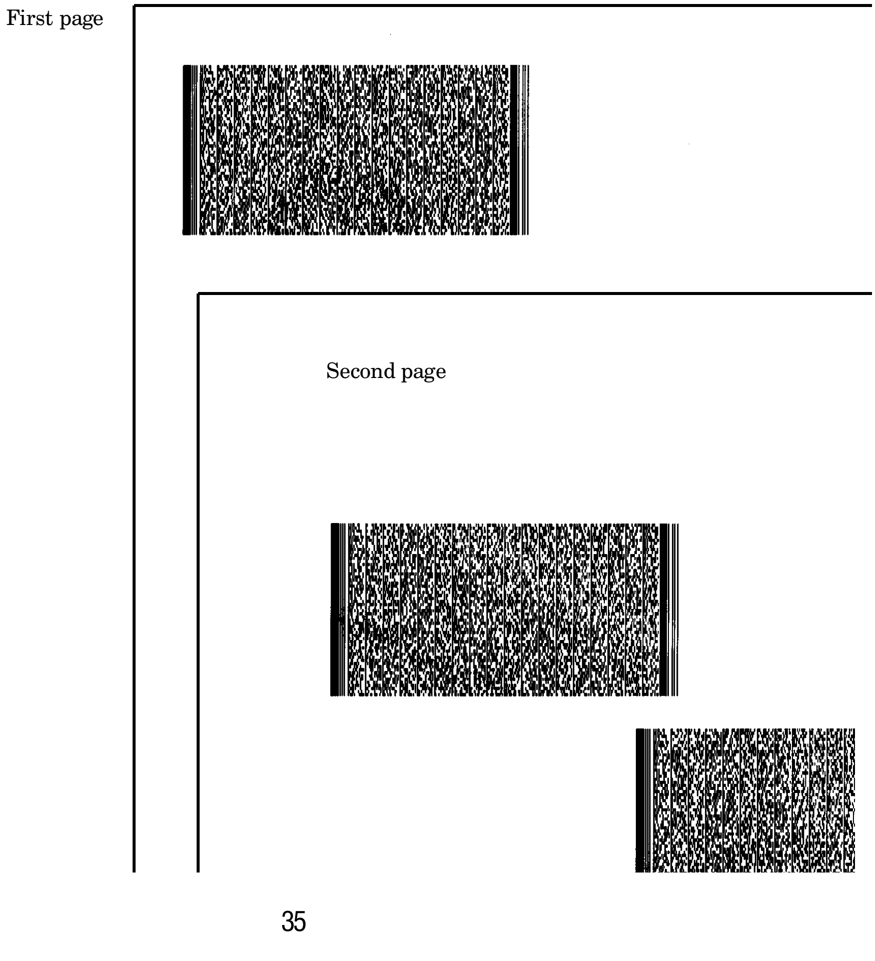


Example 4

To print multiple Macro PDF bar codes beyond the current page:

```
!R! RES;
XBCP 0,1;
XBCP 17,"TrnOvr0";
XBCP 19;
XBAR; large data such as 3000 bytes;ENDB;
PAGE;
MZP 1,2; ENDB;
MZP 4,4; ENDB;
EXIT;
```

Explanation : The fourth line XBCP 19 has no coordinating parameters specified. This allows the first segment bar code of the ensuing PDF data to be positioned at the current cursor position and the remaining data on the subsequent pages if there are data beyond the first page. The PAGE command on the sixth line starts the new page. The combinations of MZP and ENDB allow the second and third PDF bar codes printed at the specific coordinates on the following page.



]

Appendices

XBAR, XBCP, XBUF and PRESCRIBE macro

Two dimensional bar code commands, XBAR and XBCP, may be used within PRESCRIBE macro definitions. Limitations are:

- ❐ Length of a variable is limited to 256 characters. A variable may not contain control codes. Contiguous space characters inside a variable will be treated as one character. The space character is always treated as a part of data.
- ❐ If XBAR must encode control codes or more than 256 characters, then use it outside of a macro definition or use the XBUF command to define a buffer name, then call a buffer name from inside the defined macro. Refer to page 38 for details on XBUF.

Example 1—XBAR and XBCP commands outside of macro definition

```
....
XBCP x;
XBAR [length];data........;ENDB;
```

where *length* is any integer and *data* is any characters including binary control codes. If *length* is omitted, 0, negative number, or non integer, then all the data until the program sees ENDB; are counted.

Example 2—XBAR and XBCP commands inside of macro definition

```
....
MCRO PDF1,%1,%2,%3,%4,%5,%6,%7;
....
MAP%1,%2;
XBCP %3,%4,%5;
XBAR %6;%7;ENDB;
MRP2,2;
TEXT'%7';
ENDM;
```

where %6 is an integer between 1 and 256 and %7 is a string of ASCII characters of that length.

```
CALL PDF1,2,2,6,2,1,120,0123456789ABCDEF This is a test .....;
```

This will execute the following PRESCRIBE commands:

```
MAP2,2; XBCP 6,2,1;
XBAR 120;0123456789ABCDEF This is a test .....;ENDB;
MRP2,2; TEXT'0123456789ABCDEF This is a test ....';
```

XBUF command

Defining a buffer name, the XBUF command is introduced to fully support the two dimensional bar code command within a macro definition. The buffer name is called both in the macro or without a macro definition.

Limitation of the current Prescribe implementation

The XBUF command provides a partial support of allowing non ASCII characters (and a limit of 256 bytes in any string or command) to be used in the PRESCRIBE macros for supporting PDF417 commands.

XBUF

Define/undefine data input for XBAR command

Format XBUF *buffer_name* [, [*data_length*]]; *data_string* ;

Parameters

buffer_name	name of the buffer that contains XBAR data
data_length	length of data in bytes, terminated with a semicolon
data_string	data to be encoded, terminated with a semicolon (256 international characters or binary data)

Once the XBUF command is implemented to contain actual data for the two dimensional bar code, XBAR command can specify the buffer name and can be called both inside and outside of a macro statement recursively. Syntax is as follows:

XBAR *buffer_name* ;

where *buffer_name* is previously defined by XBUF command that contains actual data for the two dimensional bar code.

As the above Format instruction implies, the XBUF command sequence can have two other formats implementing the different functions as follows:

XBUF *buffer_name* , ; *data_string* ;ENDB;

Note the comma (,) placed between the *buffer_name* and "; " . As is the case of XBAR, this format allows that *data_length* can be omitted, 0, negative value, or non integer, the comma after *buffer_name* must be left in place, however. Then *data_string* is counted all characters until the program encounters " ;ENDB; ". It is necessary to have no space or line-break between "; " and "ENDB; ".

The XBUF command sequence with no comma and *data_length* specified

XBUF *buffer_name*;

is used to delete the buffer name from the printer memory.

Example 1

```
...
XBCP 0,1;
XBUF PDF 1, 25;This is a PDF417 example.;ENDB;
XBUF PDF 2, 15;This is a test.;ENDB;
...
XBAR PDF 1;
MRP 1,1;
XBAR PDF 2;
```

Explanation : The sixth line prints PDF417 bar code containing the data string buffered in the third line and the last line prints PDF417 bar code containing the data string buffered in the fourth line.

Example 2

```
...
XBCP 0, 1;
XBUF PDF 1, 25;This is a PDF417 example.;ENDB;
XBUF PDF 2, 15;This is a test.;ENDB;
...
MCRO PDF;
XBAR PDF 1;
MRP 1, 1;
XBAR PDF 2;
ENDM;
CALL PDF;
```

Explanation : The last line prints the two PDF417 bar codes defined in the macro which is defined by MCRO in the sixth line.

Part No.
Printed in Japan/Imprimé au Japan