

Product Support Bulletin

Subject: Understanding and Using **ESCP/2** Commands and Syntax

Date: 12/4/91

PSB No: P-0082

Page(s): 1 of 14

Originator: JW

This bulletin provides detailed explanations and illustrations for some of the more commonly used printer commands used in ESC/P2 Listed below are the commands discussed in this bulletin.

Command	Command Name	PSB Page
ESC X m n1 n2	Select Pitch and Point Size	2
ESC (C m1 m2 n1 n1	Set Page Length in Defined Units	5
ESC(v m1 m2 n1 n2	Set Relative Vertical Position	6
ESC(V m1 m2 n1 n2	Set Absolute Vertical Position	8
ESCtn	Select Character Table	9
ESC (t n1 n2 d1 d2 d3	Assign Character Table	10
ESC(^ n1 n2 d₁ . . . d₂	Print Data as Characters	14

ESC X m n1 n2 (Select Pitch and Point Size)

ESC X m n1 n2 is the command for selecting pitch and point size where **m**, **n1** and **n2** are variables. The variable **"m"** is used to select the pitch, which is the character spacing. The variable **"n1"** selects the point size and the variable **"n2"**, at this time, will always be 0 (zero). The variables **"m"** and **"n1"** are explained below.

The values used for **"m"** are 0, 1, 18, 21, 24, 30, 36, 42, 48, 60, and 72. If **"0"** is used as the value for **"m"**, there will be no change in pitch. If **"1"** is used as the value for **"m"**, the character pitch will be proportional spacing. To determine the character pitch for all other values of **"m"**, divide 360 by **"m"**. For example, if **m=60** then 360 divided by 60 equals 6 and the character pitch is 6 characters per inch. If **m=30**, the character pitch is 12 characters per inch.

For the Roman and Sans Serif fonts, the values used for **"n1"** are 0, 16, 20, 21, 24, 28, 32, 36, 40, 42, 44, 48, 52, 56, 60 and 64. For all other fonts, the values used for **"n1"** are 0, 21, and 42. The actual point size that will print is **n1** divided by 2. For example, if **n1=20**, the point size is 10. If **n1=48**, the point size is 24. NOTE: When selecting a point size larger than 24 (the value for **n1** is greater than 48), select proportional spacing (**m=1**) for the character pitch.

In the Basic programming language, the command would look like this.

```
LPRINT CHR$(27);"X";CHR$(1);CHR$(48);CHR$(0);"This is 24 point proportional spaced"
```

Chr\$(27) is ESC, **chr\$(1)** selects proportional spacing, **chr\$(48)** selects 24 point character size. At this time, **n2** is always zero (0).

```

10 INPUT "Point Size";A$
20 B=VAL(A$)*2
21 RESTORE 70
22 FOR X=1 TO16
23 READ Y
24 IF B=Y THEN GOT0 30
25 NEXT X
26CLS
27 PRINT A$ " is not a valid point size choice."
28 GOT0 10
30 C=VAL(A$)
40 LPRINT CHR$(27);"X";CHR$(1);CHR$(B);CHR$(0);"These are ";A$;" point
characters"
50 LPRINT CHR$(27);"v";CHR$(2);CHR$(0);CHR$(C);CHR$(0)
60 GOT0 10
70 DATA 0,16,20,21,24,28,32,36,40,42,44,48,52,56,60,64

```

Line 10 - allows you to specify the point size desired.

Line 20 - computes the value for **"n1"** in the ESC X command.

Line 21 - check to ensure that a valid point size was input **if** the point size is valid, the program jumps to line 30. **if** the point size is not valid, the program prints a message on the screen and starts over again.

Line 30 - computes the value for **"n1"** for advancing paper using the ESC (v command.

Line 40 - sends the "Set **Pitch** and Point **Size**" command to the printer. (Note: The pitch is set for proportional spacing for **all** point sizes to ensure proper character spacing regardless of the point size.)

Line **50** - sends the "Set Relative Vertical Position" command to the printer. This command advances the paper in preparation for printing the next line.

Line 60 - causes the program to start again at the beginning.

NOTE: To halt the program, hold down the **"Ctrl"** key and press the **"C"** key.

These are 8 point characters

These are 10 point characters

These are 10.5 point characters

These are 12 point characters

These are 14 point characters

These are 16 point characters

These are 18 point characters

These are 20 point characters

These are 21 point characters

These are 22 point characters

These are 24 point characters

These are 26 point characters

These are 28 point characters

These are 30 point characters

These are 32 point characters

ESC (C m1 m2 n1 n2 (Set Page Length in Defined Units

ESC (C m1 m2 n1 n2 is the command for setting the page length in units where m1, m2, n1 and n2 are variables. The "unit" is usually 1/360 of an inch unless defined otherwise by the "ESC (U" command. Currently, the values for m1 and m2 **MUST** be m1 =2 and m2=0.

To determine the values for n1 and n2 perform the following steps.

1. Determine the desired page length in inches.
2. Multiply the page length by 380.
3. Divide the result by 256 using long division.
4. The remainder in the quotient from step 3 is the value for n1.
5. The whole number in the quotient from step 3 is the value for n2.

EXAMPLE

1. The page length is 8.5 inches.
2. $8.5 \times 380 = 3060$
3. 3060 divided by $256 = 11$ with a remainder of 244 .
4. $n1=244$
5. $n2=11$

In the Basic programming language, the command would look like this.

```
LPRINT CHR$(27);"(C";CHR$(2);CHR$(0);CHR$(244);CHR$(11)
```

This command, as listed in the line above, will set the page length to 8.5 inches.

NOTE: The current line becomes the top of form. This command cancels any previously set top and bottom margin settings. If you wish to set the top and bottom margins via software command, the commands must be issued **AFTER** the "Set Page Length in Defined Units" command. The proper page length setting can be confirmed by issuing a "Form Feed" command, chr\$(12), or by pressing the "FF button on the control panel while the printer is "Off Line".

ESC (v m1 m2 n1 n2 (Set Relative Vertical Position)

ESC (v m1 m2 n1 n2 is the command for advancing the print position vertically in units where **m1**, **m2**, **n1** and **n2** are variables. The "unit" is usually 1/360 of an inch unless defined otherwise by the "ESC (U" command. Currently, the values for **m1** and **m2** MUST be **m1 =2** and **m2=0**.

To determine the values for **n1** and **n2** perform the following steps.

1. Determine the desired vertical motion **dii** in inches.
2. Multiply the distance by 360.
3. Divide the result by 256 using long division.
4. The remainder in the quotient from step 3 is the value for **n1**.
5. The whole number in the quotient from step 3 is the value for **n2**

EXAMPLE

1. The desired distance is 1 inch.
2. **1 x 360 = 360**
3. **360 divided by 256 = 1 with a remainder of 104.**
4. **n1=104**
5. **n2=1**

In the Basic programming language, the command would look like thii.

```
LPRINT CHR$(27);"(v";CHR$(2);CHR$(0);CHR$(104);CHR$(1)
```

This command, as listed in the line above, will feed the paper up 1 inch from the current line. The current print column (horizontal position) IS NOT changed.

'Reverse" paper feed less than 1/2 inch can be performed with this command. To accomplish "reverse" paper feeding requires the following steps.

1. Determine the desired vertical motion **dice** in inches.
2. Multiply the distance by 360.
3. Divide the result by 256 using long division.
4. Subtract the remainder in the quotient computed in step 3 from 256.
5. The result from step 4 is the value for **n1**.
6. The value for **n2** will always be 255.

EXAMPLE

1. The desired distance is 1/4 inch.
2. $1/4 \times 360 = 90$
3. 90 divided by 256 = 0 with a remainder of 90.
4. $256-90 = 166$
5. $n1=166$
6. $n2=255$

In the Basic programming language, the command would look like this.

LPRINT CHR\$(27);"(v";CHR\$(2);CHR\$(0);CHR\$(166);CHR\$(255)

This command, as listed in the line above, will feed the reverse feed the paper 1/4 inch from the current line. The current print column (horizontal position) IS NOT changed.

ESC (V m1 m2 n1 n2 (Set Absolute Vertical Position)

ESC (V m1 m2 n1 n2 is the command for changing the print position vertically (with respect to the Top of Form position) in units where **m1**, **m2**, **n1** and **n2** are variables. The "unit" is usually 1/360 of an inch unless defined otherwise by the "ESC (U" command. Currently, the values for **m1** and **m2** MUST be **m1 =2** and **m2=0**.

To determine the values for **n1** and **n2** perform the following steps.

1. Determine the desired vertical **distance** from the Top of Form in inches.
2. **Multiply the d i c e by 360.**
3. **Divide the result by 258 using long division.**
4. The remainder in the quotient from step 3 is the value for **n1**.
5. **The whole number in the quotient from step 3 is the value for n2.**

EXAMPLE

1. **The desired dice from the Top of Form is 1 inch.**
2. **1 x 360 = 360**
3. **360 divided by 256 = 1 with a remainder of 104.**
4. **n1=104**
5. **n2=1**

In the Basic programming language, the command would look like this.

```
LPRINT CHR$(27);"V";CHR$(2);CHR$(0);CHR$(104);CHR$(1)
```

This command, as listed in the line above, will position the paper 1 inch from the current Top of Form setting. For example, if the Top of Form setting is the top edge of the paper, the paper will be positioned so that the printer is ready to print 1 inch down from the top edge of the paper.

ESC t n (Select Character Table)

ESC t n is the command for selecting a character table to be used for printing where n is a variable. The following table lists the values for n and the character table associated with the value.

n	T a b l e
0	Italic
1	PC437 (US)
2	User-defined Characters
3	PC437 (US)

Table 1

- NOTE: 1. Character tables other than those listed above can be assigned by using the ESC (t command (Assign Character Table).
- 2 The value used for n can be either the ASCII value OR the ASCII character (i.e. chr\$(1) or "1").

In the Basic programming language, the command would look like this.

```
LPRINT CHR$(27);"t";CHR$(0);
```

CHR\$(27) is ESC and CHR\$(0) selects table #0 (italic).

ESC (t n1 n2 d1 d2 d3 (Assign Character Table)

ESC (t n1 n2 d1 d2 d3 is the command for assigning character tables where n1, n2, d1, d2, and d3 are variables. Currently, variables n1, n2 and d3 **MUST** be n1=3, n2=0, and d3=0. Variable d1 is a number corresponding to one of the four tables selectable with the ESC t n command. This is the table that will be replaced. Valid values for d1 are 0, 1, 2, and 3 (see Table 1 below). Variable d2 is a number corresponding to a registered table. This is the table that will replace the table indicated by Variable d1. Valid Values for d2 are 0, 1, 3, 7, 8, and 9 (see Table 2 below).

d1	Selectable Table
0	Italic
1	PC437 (US)
2	User-defined Characters
3	PC437 (US)

Table 1

d2	Registered Table
0	Italic
1	PC437 (US)
3	PC850 (International)
7	PC860 (Portugal)
8	PC863 (Canada-French)
9	PC865 (Norway)

Table 2

After using the ESC (t command, the ESC t n command **MUST** be sent to select the table before printing from the new table. For example, the command ESC (t 3 0 1 3 0 is sent (assign Registered Table 3 (**PC850**) to Selectable Table 1 (**PC437**)). In order to print the \$ character (ASCII 245), the command ESC t 1 must be sent. In the Basic programming language, the sequence would be as follows:

```
LPRINT CHR$(27);"(T);CHR$(3);CHR$(0);CHR$(1);CHR$(3);CHR$(0);
LPRINTCHR$(27);"t";CHR$(1);
LPRINT CHR$(245)
```

Some ESC/P2 documentation may illustrate as many as 22 different Registered Tables. The Registered Tables are a part of the character set, therefore, they are contained in the Character Generator ROM. The CG ROM differs based upon the destination country. For this reason, only the above Registered Tables are available on LQ-570, LQ-1070, LQ-870 and LQ-1170 printers sold for the U.S. market. Foreign Character Generator ROMs ARE NOT available in the United States. The following

page contains a print sample of the Registered Tables available in printers sold in the U.S. In order to print the first 31 characters in Tables 1 through 9, the ESC (^ command must be sent.

Listed below is a Basic language program and an explanation of the program that will print the output illustrated on page 13 of this bulletin.

```

10 FOR Z=1 TO 6
20  READ Y,Y$
30 LPRINT "Table";Y;" ";Y$
40 LPRINTCHR$(27);"(";CHR$(3);CHR$(0);CHR$(0);CHR$(Y);CHR$(0);
50 LPRINTCHR$(27);";";CHR$(0)
60 IF Y=0 THEN GOT0 100
70 LPRINT CHR$(27);"^";CHR$(31);CHR$(0);
80 FOR X=1 TO 255:LPRINT CHR$(X);:NEXT X
90 GOT0 120
100 FOR X=33 TO 126:LPRINT CHR$(X);:NEXT X
110 FOR X=161 TO 254:LPRINT CHR$(X);:NEXT X
120 LPRINT CHR$(10):LPRINT CHR$(10)
130 NEXT Z
140 DATA 0,Italic,1,PC437 (US),3,PC850 (International),7,PC860 (Portugal)
150 DATA 8, PC863 (Canada-French),9,PC865 (Norway)

```

Program Description

- 10 Begin loop to print the 6 Registered Tables
- 20 Read Registered Table number and Table Name
- 30 Print Registered Table number and Table Name
- 40 Send the Assign Character Table command to the printer assigning the Registered Character Table read in line 20 to Selectable Character Table 0.
- 50 Send the Select Character Table command to the printer selecting Selectable Table number 0 as the current Character Table for printing.
- 60 If the Registered Character Table number is 0, do not execute commands in lines 70 through 90. This is because the Italic Character Table duplicates control codes and non-displayable characters in ASCII values 127 through 160. Program lines 100 and 110 prevent the attempted printing of this range of ASCII values.

Note: lines 70 through 90 will execute ONLY if the Registered Character Table to be printed is NOT table number 0. Refer to the conditional command in line 60.

- 70 If the Registered Character Table number is not 0, send the Print Data as Characters command to the printer. Specifically, this command, as listed, tells the printer to print the next 31 bytes as data, not control codes.

- 60 If the Registered Character Table number is not 0, print ASCII values 1 through 255 inclusive. Remember, line 70 allows the first 31 ASCII values to print as characters, not control codes.
- 90 If the Registered Character Table number is not 0, do not execute commands in lines 100 and 110.
- 100 If the Registered Character Table number is 0, send ASCII values 33 through 126 to the printer.
- 110 If the Registered Character Table number is 0, send ASCII values 161 through 254 to the printer.
- 120 Send 2 line feed commands to the printer to provide separation between tables.
- 130 If program lines 20 through 120 have not executed 6 times, return to line 20 and start again.
- 140 Data that is read by line 20. This line and line 150 contain the Registered Table number and name that will be used by this program.

Table 0 Italic

'#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~!"#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

Table 1 PC437 (us)

'#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~!"#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

Table 3 PC850 (International)

'#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~!"#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

Table 7 PC860 (Portugal)

'#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~!"#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

Table 8 PC863 (Canada-French)

'#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~!"#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

Table 9 PC865 (Norway)

'#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~!"#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

ESC (^ n1 n2 d₁ . . . d_k (Print Data as Characters)

ESC (^ n1 n2 d₁ . . . d_k is the command for printing control codes as data where **n1** and **n2** are variables and **d₁ . . . d_k** represent the data that is to be printed as characters. The variables **n1** and **n2** tell the printer "the next **xx** ASCII codes are to be interpreted as characters instead of control codes". This command is used primarily to print displayable characters such as ASCII values 0 through 31 in Registered Tables 1 through 9 which are normally used as control codes (LF, FF, ESC, etc.)

To determine the values for **n1** and **n2** perform the following steps.

1. Determine how many ASCII codes will be sent sequentially immediately following **n1** and **n2**.
2. Using long division, divide the quantity of ASCII codes that will be sent sequentially immediately following **n1** and **n2** by **255**.
3. The remainder in the quotient is the value for **n1**.
4. The whole number in the quotient is the value for **n2**.

EXAMPLE

1. The symbols for the different suits of a card deck (hearts, diamonds, clubs, and spades - ♥, ♦, ♣, and ♠) are to be printed. These are each separate ASCII codes (ASCII codes 2, 3, 4, and 5 when using the PC437 Character table).
2. 4 divided by 255 = 0 with a remainder of 4.
3. **n1 = 4**
4. **n2 = 0**

In the Basic programming language, the command would look like this.

```
LPRINT CHR$(27);"(^";CHR$(4);CHR$(0);CHR$(2);CHR$(3);CHR$(4);CHR$(5)
```

This command, as listed in the line above, will cause the printer to print the symbols for the different suits of a card deck (hearts, diamonds, clubs, and spades ♥, ♦, ♣, and ♠). Another example of using the ESC (^ command is illustrated in the explanation of the ESC (t (Assign Character Table) command.