Graphics Languages –Code V, PGL, MT660 IG

Applications Manual

6800 Series Printers

Trademark Acknowledgements

Acrobat® Reader is a trademark of Adobe Systems Incorporated. DEC is a trademark of Compaq Computer Corp.

Epson is a trademark of Seiko Epson Corp. Genicom is a trademark of Genicom L.L.C.

HP is a trademark of Hewlett-Packard Company.

IBM and Proprinter are trademarks of International Business Machines Corporation.

Printronix and PGL are trademarks of Printronix, Inc. QMS and Code V are trademarks of Minolta-QMS Inc. TallyGenicom brand is owned by Printronix, Inc.

Table of Contents

1 QMS® Code V™ Graphics Processing Language

Introduction

This manual describes the features and uses of the Code V Graphics Processing Language. Code V uses a Graphics Processing Language (GPL) that provides the user with total control over the printing and formatting of graphics images. This GPL is compatible with QMS® Code V™ versions 1 and 2 and includes the following features.

- Block characters, printed horizontally or vertically with height and width ranging from 0.1 to 9.9 inches. In addition to normal black on white printing, block characters can be printed in special half-tone patterns or in reverse image (white characters on a black background).
- Numerous barcodes, including Code 39 Interleaved 2 of 5, UPC, EAN, Code 128, and Codabar. All barcodes can be printed horizontally or vertically with various height, width, and ratio dimensions. Barcodes may also be printed with or without human readable characters in a wide variety of fonts.
- Solid or dashed line drawings using various line thickness.
- Box drawing with user definable size and border thickness.
- Horizontal and vertical duplication of graphics images.
- User-definable, memory resident forms, including optional font fields.
- Predefined AIAG and Primary Metals forms.
- Draft, NLQ, OCR-A, and OCR-B typefaces in various sizes.
- Predefined symbols such as copyright (©) and registered trademark (®).
- A plot mode.

Graphics Processing Language

The Graphics Processing Language (GPL) consists of a set of commands that are used to print graphics images such as block characters, barcodes, lines, and boxes, as well as provide formatting control for the correct placement of these images. Relatively complex images may be created through the use of a few simple commands.

The original version of QMS Code V (version 1) differs slightly from the current version (version 2). The primary difference between the two is in the way vertical justification commands are implemented. Specifically, version 1 used an approximation for 0.1 inch that has been made more precise in version 2. If you are running an application designed for version 1, the printer can be switched to a mode that can emulate it.

Graphics Mode

For the printer to recognize GPL commands, it must first be placed in Graphics Mode. When the printer is in Graphics Mode, the printer scans the data received from the host for GPL commands, passing non-

Graphics data to the current emulation. When the printer is not in Graphics Mode, Code V is disabled and will not search the incoming data for GPL commands.

Graphics Mode may be entered while the printer is in any emulation. Graphics Mode should not be entered while the printer is in Plot Mode. The printed output of Code V adheres to all the applicable Control Panel settings (such as LPI, CPI, Form Length, etc.).

NOTE: All escape sequences in this manual are in ASCII notation.

Turning Graphics Mode On

Graphics Mode is turned on with command ^PY, which must begin in the first column of a line, followed by one of the following six terminators.

1. ^- (caret hyphen)
2. ^* (caret asterisk)
3. ^, (caret comma)
4. Carriage Return
5. Line Feed
6. Form Feed

Once the printer is in Graphics Mode, the data it receives is read into a Pass buffer. The printer will stop looking for characters from the host and process the data held in this buffer when one of the terminators is received. If the buffer fills up before receiving a Pass Terminator, data in the buffer will be processed and printed before the printer accepts any more characters.

When Graphics Mode is on:

- The control panel displays ONLINE GRAPHICS
- All GPL commands and data are processed by Code V
- Non-GPL data is passed to the current emulation.

PY Then

This control panel option defines the action taken on characters that are on the same line as the Graphics Mode On Command (^PY). The options are All, Term, and None. The default is None. These options are defined as follows:

All: All data following Graphics Mode On Command is processed.

Term: Only the Terminator (FF, LF, or CR) is processed.

None: None of the data is processed.

If the None option is selected, any one of the following control codes (or combination of control codes) immediately following a ^PN or ^PY^- on the same line will not be printed:

- Carriage Return
- Line Feed
- New Line
- Form Feed
- Carriage Return followed by a Line Feed
- Carriage Return followed by New Line, or Carriage Return followed by Form Feed

If PY Then is set to All or Term, all characters that follow a PN or ^PN^- on the same line will print.

Turning Graphics Mode Off

The Graphics Mode Off Command is the sequence ^PN followed by a Pass Terminator. As with the Graphics Mode On Command (^PY), the Graphics Mode Off Command must be sent in the first column of a line (i.e. after a Carriage Return). This command will cause the printer to:

- Exit the Graphics Mode
- Display the ONLINE message
- Treat all further GPL commands as printable characters to be sent to the current emulation.

A few fundamental GPL commands and concepts are described in the following sections. Knowledge of these commands and concepts is essential for understanding the descriptions and examples in the remainder of this manual.

Code V Command Character (CVCC)

All GPL commands begin with a Command Character. By default, this character is an ASCII caret (^) having hexadecimal value of 5E.

Changing the CVCC

There are two ways to change the CVCC.

Using ESC Sequences

Using the ^N Command Change Sequence

This method can only be done in Graphics Mode. Send the ^N followed by the new CVCC (hex. OOto hex. FF), then terminate the ^N sequence with the New CVCC.

Example:

^N$$ would change to CVCC to the ASCII dollar sign character ($-hex, 24, Dec 36).

When you leave Graphics Mode and return to Character Mode or when the pritner is reset, the Command Character defaults to the ASCII caret character (^).

Through The Control Panel

This method can be used either in Graphics Mode or in Character Mode. To use this method take the printer offline and access the Configuration Menu. After accessing the Configuration Menu proceed through the menu levels until you access the decimal selections under the CVCC Parameter in the Graphics Category. Scroll until the desired decimal value is reached, then depress the ENTER key. This Command Character will remain valid until you exit Graphics Mode (this resets it to the default value), the printer is reset, or you change it with a ^N command or through the control panel.

NOTE: If the Command Character is set through the Control Panel outside of Graphics Mode, that character stays in effect unless changed by the

^N command. If a config. report is printed while in Graphics Mode, it will show the character chosen with the ^N command instead of the character selected through the control panel. When you exit from Graphics Mode, the Command character is reset to the value selected through the Control Panel.

However, if the Command Character is set through the control panel inside of Graphics Mode, that character stays in effect only until you exit Graphics Mode, after which the Command Character is reset to the value you selected through the control panel outside of Graphics Mode.

As with the ^N Command, if a config. report is printed while in Graphics Mode, it will show the character chosen through the control panel inside of the Graphics Mode.

All Graphics Mode commands in this manual are shown using the (default) caret symbol.

Free Format

The GPL provides a mode which ignores all data with values less than 20 (decimal 32). Data having values below decimal 32 are collectively called Control Characters and include Line Feed, Form Feed, and Escape characters. By using Free Format Mode, the user can filter out unwanted Control Characters which may be sent by some host systems. Also, a convenient way to send GPL commands to the printer is to simply type the commands into a file and send the file to the printer. In Format Mode, a command may be spread over several lines in the file, with the printer ignoring the CR and LF characters at the end of each line.

NOTE: The Free Format ON/OFF commands may not be used within the data field of a Report or a Buffered Form.

Turning Free Format On

Free Format is turned on by sending the command ^F. After the printer receives the ^F, subsequent control characters are ignored and the printer will no longer recognize CR, LF, or FF as Pass terminators. However, the GPL language will recognize the following control code equivalents as Pass terminators:

1. ^- (caret hyphen) - equivalent to the Carriage Return Command (CR)
2. ^* (caret asterisk) - equivalent to the Line Feed Command (LF)
3. ^, (caret comma) - equivalent to the Form Feed Command (FF)

Turning Free Format OFF

Free Format is turned off by sending the command ^O^-.

Control Code Equivalents

When Free Format is turned ON, GPL control code equivalents are treated as actual ASCII control codes. For example ^! functions in the same manner as the ASCII control code SOH. If control code characters need to be passed to the printer in Free Format Mode, use the control code equivalents shown in Table 1.

Table 1. Control Code Equivalents

Control Code Equivalents	Hex	ASCII
^! (caret exclamation)	01	SOH
^” (caret double quote)	02	STX
^# (caret pound sign)	03	ETX
^$ (caret dollar sign)	04	EOT
^% (caret percentage sign)	05	ENQ
^& (caret ampersand)	06	ACK
^’ (caret single quote)	07	BEL
^((caret left parenthesis)	08	BS
^) (caret right parenthesis)	09	HT

Table 1. Control Code Equivalents

Control Code Equivalents	Hex	ASCII
^* (caret asterisk)	0A	LF
^+ (caret plus sign)	0B	VT
^, (caret comma)	0C	FF
^- (caret hyphen)	0D	CR
^. (caret period)	0E	SO
^/ (caret slash)	0F	SI
^0	10	DLE
^1	11	DC1
^2	12	DC2
^3	13	DC3
^4	14	DC4
^5	15	NAK
^6	16	SYN
^7	17	ETB
^8	18	CAN
^9	19	EM
^: (caret colon)	1A	SUB
^; (caret semicolon)	1B	ESC
^< (caret left arrow)	1C	FS
^= (caret equal sign)	1D	GS
^> (caret right arrow)	1E	RS

Ignore Data

The Ignore Data command causes the printer to ignore all data (except for the command to turn Ignore Data OFF). This allows comments or other that is not to be printed to be inserted into the document for information purposes.

Format: ^X<data>^A

^X Ignore Data On Command. The printer will ignore all data after this command except for the Ignore Data Off Command, ^A.

<data> Data that the printer will ignore.

^A Ignore Data Off Command. The printer will not accept all subsequent data received from the host.

Graphics Pass

A Graphics Pass begins with a Pass Start (^M, ^U, ^V, or ^E) and ends with a Graphics Pass Terminator (Pass Start, CR, LF, FF, ^-, or ^,). Any characters placed between the Pass Start and Pass Terminator will print as block characters. Also, all GPL Graphics Image commands, such as the Bar Code and Line Drawing commands must be sent within a Graphics Pass. Each Graphics Pass is treated by the printer as a set of data and instructions to be processed and printed, starting at the current paper position, before accepting any further data from the host.

Each pass begins at the current vertical paper position and the left margin. From this point, the printer prints to the right and down the page.

NOTE: It is not possible to print above the starting point of a Graphics Pass.
It is possible, however, to issue positioning commands within the pass which position the paper at any point below the start of the pass.

NOTE: When (LF, FF, ^*, ^,) are used to terminate a Graphics Pass, they lose their normal meanings, and become Pass Terminators such as

^-. When used outside the Graphics Pass they take on their normal meaning.

Not all GPL commands and data sent to the printer need to be in a pass. In fact, many commands must be outside of a pass to function properly. For instance, the command ^H sets the form length if it is used outside of a pass, but sets the pass height if it is used within a Pass.

At this point, it may seem difficult to decide when to use a Graphics Pass. If there is any doubt about whether a command should go inside or outside of a pass, refer to Appendix E, “Command Table” on page 231. The GPL commands fall into two general categories:

Inside A Pass

- GPL positioning commands (e.g., tabs, vertical justification, etc.)
- GPL printing commands (e.g., barcodes, lines, boxes, certain fonts, etc.)
- Block characters

Outside A Pass

Global Graphics Pass commands (i.e. commands which act on the entire pass as a unit, such as those which cause the entire Pass to be repeated or stored in memory for later use).

Environmental commands such as form length setting, Graphics Mode, and Free Format.

Non-GPL Data

Any data you want to go to the current Control Panel emulation must be sent outside of a Graphics Pass. Appendix E, “Command Table” on 231 lists all GPL commands.

Pass Spacing (Modplot)

Code V output can be affected by a Plot Mode option that is available on your printer. Appendix F, “Modplot” on 235 explains the Modplot option and how it affects Code V output.

Pass Buffer Length

Data received from the host is stored in a Pass Buffer. The Printer begins to process a Graphics Pass when it receives a Pass Terminator. The printer processes one pass at a time, printing at the end of each pass. If the Pass Buffer fills up before the printer encounters a Pass Terminator, it terminates the pass and attempts to process and print it.

Buffered Form definitions and Repeat sequences cannot be processed until the Form or Repeat Terminator is received; therefore, the entire Form Definition or Repeat Sequence must fit into the Pass Buffer to be printed correctly. Buffered Form execute data can be larger than the Pass Buffer. The Printer processes this data one Pass Buffer at a time, until the Form Execute Terminator is received.

The Printer uses a Heap Buffer to process the pass for printing. If a Graphics Pass is so complex that processing it overflows the available Heap Buffer space, the Printer aborts processing and announces a Heap Overflow fault.

Pass Format

A single Graphics Pass Command is formatted like this:

Command ^ {M, U, V or E}hhwwjjd<data>^-

Arguments ^ {M, U, V, or E} the braces, { }, indicate that either M, U, V, or E must be used. This notation is used throughout the manual to indicate that a choice must be made. The braces themselves are not literally part of the command.

The M, U, V, and E determine the orientation of the data to be printed. For example:

Horizontal Orientation (^M)

Prints data across the page right side up.

Upside Down Orientation (^U)

Prints data across the page upside down.

Vertical Right Orientation (^V)

Prints data down the page and roated 90 degrees clockwise so that the baseline faces the left margin of the paper.

Vertical Left Orientation (^E)

Prints data down the page and rotated 90 degrees counterclockwise so that the baseline faces the right margin of the paper:

hh — The height of each individual barcode or block character. This dimension is measured vertically on the page and is specified in increments of 0.1 inches (0.25 cm). The allowable height for each barcode/block character ranges from 01 to 99 (0.1 to 9.9 inches or 0.3 to 25.2 cm). For example, a value of 03 means a height of 0.3 inches (0.76 cm).

ww — The width of each printed barcode/block character. This dimension is measured horizontally on the page and is specified in increments of 0.1 inches (0.25cm). The width for each barcode/ block character ranges from 01 to 99 (0.1 to 9.9 inches or 0.3 to 25.2 cm) For example, a value of 10 means a width of 1.0 inches (2.5 cm).

jjd — Data can be positioned down the page from the beginning of the Pass. This is teemed justification. the jj is the amount of justification from 00 to 99 increments of 0.1 inches (0.3 to 25.2 cm). The d allows an additional amount of justification down from 0 to 9 dot rows to fine tune the final position of the data. For example, inserting the value 118 means the data is moved down 1.1 inches (2.8 cm) plus an additional 8 dot rows. A justification value of 000 sets the current print position to the top of the current Graphics Pass.

NOTE: Any values not specified will default to 0.

<data> — The actual data can be printed. Characters in the data string will be printed as block characters or in a particular GPL font. Barcodes, lines, boxes, and formatting commands may also be placed in the pass. The data contained within the pass is printed in accordance with the orientation, height, width, and justification values of the Graphics Pass.

^- — Every Graphics Pass is terminated with a Pass Terminator, indicating the end of the Graphics Pass. When the printer is in Free Format Mode, the printer will only recognize ^-, ^*, and ^, as Pass Terminators. When the printer is not in Free Format Mode, Carriage Return (CR), Line Feed (LF), and Form Feed (FF) may be used. Throughout the manual, ^- will typically be used to indicate a Terminator.

GPL Commands

Most of the examples shown in this manual assume Graphics Mode and Free Format Mode are turned on, although the examples will not always show the ^PY and ^F commands. The complete set of commands to turn Graphics and Free Format Mode On, send GPL commands and data, then turn Free Format and Graphics Mode Off are shown below:

^PY^-	Turns on Graphics Mode
^F^-	Turns on Free Format Mode
<data>^	Sample GPL Command sequences
^O^	Turns off Free Format Mode
^PN^-	Turns off Graphics Mode

The above lines are typically entered into a file and sent to the printer, or are written to the printer using a computer language such as BASIC or C with Free Format and Graphics mode on.

The main reasons to turn Graphics mode off are:

- To print a large amount of non-GPL data.
- To reset the Code V to its default startup state.
- To send plot data to the underlying emulation.

The remaining chapters in this manual discusses how the GPL command sequences should be formatted. For all subsequent examples in this manual, the following conditions are assumed:

- The control panel displays ONLINE GRAPHICS.
- Graphics Mode is ON (^PY^-).
- The printer is in Free Format mode (^F^-).

Multiple Passes

A Graphics Pass may contain other GPL commands, including other ^M, ^U, ^V, and ^E commands. The printer does not begin printing a Graphics pass until the Pass is terminated. After a pass is printed, the next pass begins at the bottom of the printed Pass.

For example, the sequence of GPL commands

^M0101000A^M0101000B^M0101000C^- produces the same printed result as ^M0101000ABC^- which prints:

whereas the three graphics Passes ^M0101000A^-^M0101000B^-

^M0101000C^- print:

2 Character Printing

Control Panel Option

A new option has been added to the control panel under the Graphics category. The new option, Vscale, determines whether or not vertical block characters will be scaled to match the difference between the horizontal and vertical print densities. The option is necessary to be fully compatible with old QMS Code V version 1 boards that are still in the field.

Block characters are built to be printed horizontally at 60 x 72. When the characters are rotated and printed vertically, it is necessary to scale the characters to maintain the cell size of the character. Some QMS Code V version 1 boards do not handle this scaling, so that when a 5 x 7 block character is rotated and printed vertically, it will actually be 4 x 8 when the Vscale option is turned off, we will now emulate this.

Block Character

Block characters are printed by putting the characters to be printed into the

<data> portion of a Graphics Pass. Thus, the Block Character Command is the same as the Graphics Pass Command.

Each block character is printed within a window which includes an appropriate amount of space for character separation. The block character window size ranges from 0.1 inches (0.25 cm) to 9.9 inches (25.15cm) in increments of 0.1 inch (0.25 cm). the character window includes 1 dot row or column of intercharacter space for every 0.1 inches in window size. That is, if the Graphics Pass command is ^M0505000, then the character window size will be 0.5 x 0.5 inches, including a 5 dot-column space to the right of the character (see Figure 1).

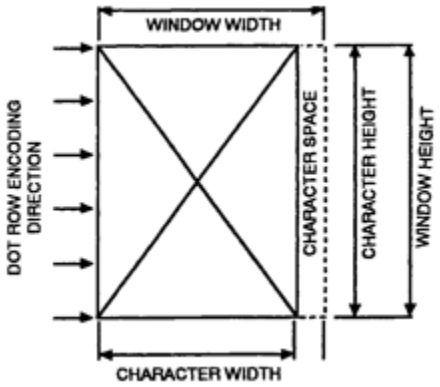


Figure 1. Character Window

When a block character is rotated 90 degrees clockwise or counterclockwise on a page, the height and width values are exchanged:

- the Graphics Pass height value determines the new character width.
- the Graphics Pass width value determines the new character height.

Command	^{M, U, V, or E}hhwwjjd<data>^-
Arguments	^{M, U, V, or E} One of the four orientation commands ^M, ^U, ^V, or ^E.
hh	The height of each printed character window. This dimension is measured vertically on the page and is specified in increments of 0.1 inches (0.25 cm). The height for each character window ranges from 01 to 99 (0.1 to 9.9 inches or 0.25 to 25.3 cm). For example, a height of 03 means a character window height of 0.3 inches (0.76cm).
ww	The width of each printed character window. This dimension is measured horizontally on the page and is specified in increments of 0.1 inches. The width for each character window ranges from 01 to 99 (0.1 to 9.9 inches or 0.3 to 25.3 cm). For example, a width of 10 means a character window width of 1 inch (2.5 cm).
jjd	The vertical justification, jj, ranges from 00 to 99 (0.0 to 9.9 inches/0.0 to 25.2 cm). The d allows an additional amount of justification down from 0 to 9 dot rows to fine tune the final positioning. For example, inserting the value 118 means the characters are moved down the page 1.1 inches (2.8 cm) plus an additional 8 dot rows. If any of the jjd is not sent, the printer assumes the missing values are 0. If any of the jjd is nto sent, the printer assumes the missing values are 0. For instance, M1010 would have a jjd value of 000.
<data>	The actual characters to be printed. This data will print in accordance with orientation, height, width, and justification values.
^-	The Pass Terminator

The following examples illustrate block character printing.

Example 1: The Graphics Pass ^M0202000Graphics ^- prints:

GraPhics

^M	Horizontal orientation of character.
02	Character height of 0.2 inches (0.51 cm).
02	Character width of 0.2 inches (0.51 cm).
00	Justified downward 0.00 inches.
0	No additional dot rows of justification down.
Graphics	Data to be printed.
^-	Pass Terminator.

Example 2: The Graphics Pass ^M0302000Graphics^- prints:

GraPhics

^M	Indicates this Graphic Pass is oriented horizontally.
03	Character height of 0.3 inches (0.76 cm).
02	Character width of 0.2 inches (0.51 cm).
00	No justification down.
0	No additional dot rows of justification down.
Graphics	Data to be printed 0.3 inches high and 0.2 inches wide.

Example 3: The Graphics Pass ^U0303000Graphics^- prints:

^U	Rotated 180 degrees from the horizontal orientation.
03	Character height of 0.3 inches (0.76 cm).
03	Character width of 0.3 inches (0.76 cm).
00	No justification down.
Graphics	Block characters to be printed in the upside down orientation.

Example 4: The Graphics Pass ^E0101000Vertical Left^_ prints:

^E	Vertical left orientation of the pass.
01	Character height of 0.1 inches (0.25 cm).
01	Character width of 0.1 inches (0.25 cm).
00	No justification down.
0	No additional dot rows of justification down.

Example 5: The Graphics Pass ^M0202000GRAPHICS ^M0203053OPTION^- prints:

- First Command - ^M0202000GRAPHICS

^M	Horizontal orientation of the Graphics Pass.
02	Character height of 0.2 inches (0.51 cm).
02	Character width of 0.2 inches (0.51 cm).
000	No vertical justification.
GRAPHICS	Data printed at 0.2 inches high and 0.2 inches wide.

- Second Command - ^M0203053OPTION^-

^M	Horizontal orientation of the Graphics Pass.
02	Character height of 0.2 inches (0.51 cm).
05	Character width of 0.3 inches (0.76 cm).
05	Justification down of 0.5 inches (1.27 cm).
3	An additional three dot rows of justification down.
OPTION	Data printed at 0.2 inches high and 0.3 inches wide with downward justification of 0.5 inches plus three dot rows.

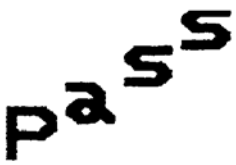
Example 6: To place OPTION directly under GRAPHICS, a pass terminator (^-^*) must be placed after the end of the first pass but before the beginning of the second pass:

^M0404000GRAPHICS^-^*M0404053OPTION^- will now print:

GRAPHICS

OPTION

Example 7: This example shows horizontal block characters with varying degrees of justification.



The Graphics Pass: ^M0303060P^M0303040a^M0303020s^M0303000s^- prints:

- First Command - ^M0303060

^M	Horizontal orientation of characters.
03	Character height of 0.3 inches (0.76 cm).
03	Character width of 0.3 inches (0.76 cm).
06	Justified down 0.6 inches (1.5 cm).
0	No additional dot rows of justification.
P	Data to be printed.

- Second Command - ^M0303040a

^M	Horizontal orientation of characters.
03	Character height of 0.3 inches (0.76 cm).
03	Character width of 0.3 inches (0.76 cm).
04	Justified down 0.4 inches (1.0 cm).
0	No additional dot rows of justification.

- Third Command - ^M0303020s

^M	Horizontal orientation of characters.
03	Character height of 0.3 inches (0.76 cm).
03	Character width of 0.3 inches (0.76 cm).
02	Justified down 0.2 inches (0.51 cm).
0	No additional dot rows of justification
s	Data to be printed.

- Fourth Command - ^M0303000s^-

^M	Horizontal orientation of characters.
03	Character height of 0.3 inches (0.76 cm).
03	Character width of 0.3 inches (0.76 cm).
00	No justification down.
0	No additional dot rows of justification
s	Data to be printed.

Descender Mode

The Descender Mode is recognized only within a Graphics Pass (^M, ^V, ^E, or ^U has been previously sent). It can be toggled on and off as often as desired within a Graphics Pass, and it is terminated when the Graphics Pass terminates. The amount of space reserved for descenders is defined as two dot rows for each 0.1 inches (0.25 cm) of character height. There are two different formats for the Descender Mode. These can be toggled from the control panel via the Descender option in the Graphics Category menu to either Fixed or Auto.

If set to Fixed the descender gap is always present whenever the Descender Mode is turned on, even if no descenders are printed.

If set to Auto, the descender gap appears only when descenders are printed. If no descenders are printed, the gap is closed.

When Descender Mode is toggled off, all lower-case characters with descenders are moved up so that the bottom of each descender rests on the baseline. Lower-case characters without descenders appear the same whether Descender Mode is ON or OFF.

NOTE: All Graphics Passes start in the default Non-Descender Mode.

Command ^D

Example 1: ^M0202000j^Dj^Dy^Dy^- prints:

Reverse Image

Reverse Image causes the printer to print white block characters on a black background. The black background extends beyond each character one dot row or column for every 0.1 inch increment of character window size. For example, a Reverse Image character window with a horizontal dimension of 03 (0.3 inches or 0.76 cm) and a vertical dimension of 04 (0.4 inches or 1.0 cm) will have the dark background extend three dot columns left and right of the character(s) and four dot rows on the top and bottom.

The Reverse Image Command is valid only if it is within a Graphics Pass. If it is outside of a Graphics Pass, it is interpreted as a Repeat Command. If the Graphics Pass is terminated while Reverse Image is ON, Reverse Image is automatically terminated.

Command ^R<data>^R

Arguments

^R	Turns ON Reverse Image.
<data>	Data to be printed in Reverse Image.
^R	Turns OFF Reverse Image.

Example 1: Sending ^M0202000^RGRAPHICS^- prints:

Example 2: Sending ^M0202000^ROPT^RION^- prints:

NOTE: The reverse image characters are a few dots lower.

Reverse Image Descender Mode

If Descender Mode (^D) is turned on with Reverse Image on, the entire dark background extends down to encompass the descenders.

Example 1: Sending the Graphics Pass ^M0303000^Rfghij^R^- with Descender Mode OFF prints:

Example 2: Sending the Graphics Pass ^M0303000^R^Dfghij^R^D^- with Descender Mode ON prints:

NOTE: The dark background of Reverse Image extends three dot rows below the descenders.

Half-Tone

Lines, boxes, and characters are printed in solid black, but they can also be printed in patterns of dots. These patterns are called half-tones. Many half tone patterns are available, such as vertical, horizontal, and diagonal lines, as well as many other designs.

The selected pattern becomes the default pattern as long as the Graphics Mode is enabled. Once Graphics Mode is disabled using the command ^PN^-

, the dot pattern defaults to the pattern corresponding to 04 hex.

The following command is used to turn on the Half-Tone Mode as well as to select a half-tone pattern:

Command ^KLxx<data> Arguments

^KL	Turns ON the Hfl-Tone Mode.
xx	Specifies the half-tone pattern, where xx equals any hex number between 00 and FF.
<data>	Data to be printed using the selected pattern.

Example: The Graphics Pass ^M0303000^KL02DOT PATTERN^- prints:

^KL	Turns ON the Half-Tone Mode.
02	Selects the half-tone pattern corresponding to 02.
DOT PATTERN	Data to be printed using the selected pattern.

Half-Tone Toggle

Once a half-tone pattern has been selected, it can be toggled ON and OFF within a Graphics Pass with the command:

Command ^KH<data>^KH

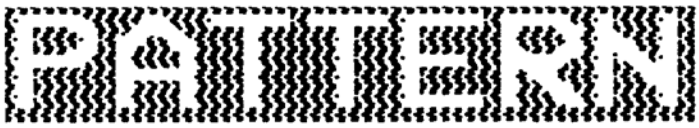
Example: The Graphics Pass ^M0303000GRA ^KL05PHI^KHSC OP^KHTIO^KHN^- prints:

GRA	Data printed as solid black.
^KL	Turns Half-Tone Mode ON.
05	Selects the 05 half-tone pattern.
PHI	Data printed using selected pattern.
^KH	Turns Half-Tone Mode OFF.
CS OP	Data printed as solid black.
^KH	Toggles Half-Tone Mode ON (half-tone pattern 05).
TIO	Data printed using the selected pattern.
^KH	Toggles Half-Tone Mode OFF.
N	Data prints as solid black.

Half-Tone Reverse Image

The dark background in Reverse Image mode can also be printed using half- tone pattern.

Example: The Graphics Pass ^M0505000^R^KL09PATTERN^KH^- prints:



^R	Turns ON the Reverse Image Mode.
^KL	Turns ON the Half-Tone Mode.
09	Selects the 09 dot pattern.
PATTERN	The characters to be printed in Reverse Image using the Half-Tone Mode.
^KH	Turns OFF the Half-Tone Mode.

Half-Dot Mode (Double Density)

The Half-Dot Mode causes the printer to print at 120 DPI horizontally (the standard print density is 60 DPI). Printing at 120 dpi results in darker printing and reduces the stairstep effect of diagonal lines. It is most often used to improve barcode readability. The Half-Do Mode can be toggled ON and OFF within a Graphics Pass using the command:

Command ^KF<data>^KF

Arguments

^KF	Toggles Half-Dot Mode ON.
<data>	Data to be printed in Half-Dot Mode.
^KF	Toggles Half-Dot Mode OFF.

Example: Half-Dot characters can be mixed with regular characters. The Graphics Pass ^M0303000^KFHALF-^KFDOT MODE^- prints:

^KF	Turns ON Half-Dot Mode.
HALF	Data to be printed in Half-Dot Mode.
^KF	Turns OFF Half-Dot Mode.
DOT MODE	Data to be printed in regular density.

Half-Dot Mode and Half-Tones

Half-Dot mode (Double Density) can be intermixed with half-tones to produce a greater variety of shades.

Example: The Graphics Passes ^M0303000^KLEAGRAPHIC^KH^ ^M0303040^KH^KFOPTION^KF^KH^- prints:

- First Graphic Pass - ^M0303000^KLEAGRAPHIC^KH^

^KL	Turns ON Half-Tone Mode.
EA	Selects EA half-tone pattern. GRAPHIC Data printed in the selected pattern.
^KH	Turns OFF Half-Tone Mode.

- Second Graphics Pass - ^M0303040^KH^KFOPTION^KF^KH^-

^KH	Toggles Half-Tone Mode ON and defaults to last pattern selected (EA).
^KF	Toggles ON Half-Dot Mode.
OPTION	Data printed with the selected pattern (EA) in Half- Tone/Half-Dot Mode.
^KF	Toggles OFF Half-Dot Mode.
^KH	Toggles OFF Half-Tone Mode.

Block Character Fonts

Block character fonts are created from the block character set in four graphic orientations, and are available in the following sizes:

- 7.5 CPI block characters, 0.2 inches high.
- 10 CPI block characters, 0.1 inch high.
- 12 CPI block characters, 0.1 inch high.
- 15 CPI block characters, 0.1 inch high.

7.5 CPI

The 7.5 CPI character set is 2/10 inches high and 0.15 inches wide (0.51 cm high by 0.38 cm wide). All characters in this set are printed in upper case, even if the characters received from the host are lower case.

Command ^{M,U,V,E}0000jjD<data>^-

Arguments

^{M,U,V,E}	One of four Graphics Pass orientations (^M for horizontal, ^V for clockwise rotation, ^E for counterclockwise rotation, or ^U for upside down and reverse order of characters.)
0000	A height/width value of 0000 specifies 7.5 cpi characters.
jjd	Justification values in increments of 0.1 inches (0.25 cm) and dot rows as discussed in the Graphics Pass Command description below. This field may be omitted entirely if the justification value is zero.
<data>	The characters to be printed.

Example: The Graphics Pass ^M00000007.5 CPI CHARACTERS^- prints:

10 CPI

The 10 CPI block character set is 0.1 inches high and is selected by using the following Graphics Pass Command.

Command ^{M,U,V,E}0101jjd<data>^-

Arguments

^{M,U,V,E}	One of four Graphics Pass orientations (^M for horizontal, ^V for clockwise rotation, ^E for counterclockwise rotation, or ^U for upside down and reverse order of characters).
0101	A height/width value of 0101 specifies 10 CPI characters.
jjd	Justification values in increments of 0.1 inches (0.25 cm) and dot rows as discussed in the Graphics Pass Command description below. This field may be omitted entirely if the justification value is zero.
<data>	The characters to be printed.

Example 1: The Graphics Pass ^M0101000HORIZONTAL 10 CPI^- prints:

Example 2: ^M0101000HORIZONTAL 10 CPI

^M0202000HORIZONTAL 5 CPI^- prints:

12 CPI

The 12 CPI character set is 0.1 inches high and is selected with the following special Graphics Pass command.

Command ^{M,U,V,E}0001jjD<data>-

Arguments

^{M,U,V,E}	One of four Graphics Pass orientations (^M for horizontal, ^V for clockwise rotation, ^E for counterclockwise rotation, or ^U for upside down and reverse order of characters).
0001	A height/width value of 0001 specifies 12 cpi characters.

jjd Justification values in increments of 0.1 inches (0.25 cm) and dot rows, as in the Graphics Pass command description. This field may be omitted entirely if the justification value is zero.

<data> The characters to be printed.

Example: The Graphics pass ^M000100012 CPI CHARACTERS^- prints:

15 CPI

The 15 CPI character set is 0.1 inches high and is selected using the following command.

Command ^{M,U,V,E}0100jjD<data>^-

Arguments

^{M,U,V,E} One of four Graphics Pass orientations (^M for horizontal, ^V for clockwise rotation, ^E for counterclockwise rotation, or ^U for upside down and reverse order of characters).

0100 A height/width value of 0100 specifies 15 cpi characters.

jjd Justification values in increments of 0.1 inches (0.25 cm) and dot rows, as in the Graphics Pass command description. This field may be omitted entirely if the justification value is zero.

<data> The characters to be printed.

Example: The Graphics Pass ^M010000015 CPI CHARACTERS^- prints:

Code V Font Selection

Since Code V runs on top of another emulation, the Code V Language has the ability to print either emulation fonts or Code V fonts. When Code V is turned on via the ^PY^- command, it sets the out of pass font to the 10 CPI Draft emulation font using the character set selected on the Control Panel. Code V has the ability to select different kinds of fonts via Code V font commands. These font commands and their effect(s) are listed below:

^IFONT,S# If issued inside a pass, the specified font is changed for the duration of the pass. If issued outside a pass, the font is changed to the specified Code V font both inside and outside passes.

^S# Valid only inside a pass, this command changes the font of the characters in the current pass to the specified Code V font.

^# Valid only outside a pass, this command changes the font of the characters outside passes to the specified Code V font.

^@C Valid only outside a pass, this command changes the emulation font, thus affecting Code V outisde of pass characters and characters printed after exiting Code V.

Code V Character Set Selection

All emulations and Code V have two character sets: upper (eighth bit high) and lower (eighth bit low). For all Code V fonts, the default lower character set (characters 20 - 7F hex) is US and the default upper character set (characters A0 - FF hex) is Latin1. Complete charts for the US and Latin1 character sets are available in Appendix E, Table 46: on page 231.

The CVCC font lower character set may be selected within Code V by using the ^IISO command. The Code V font upper character set cannot be changed and is always Latin1. Changing the character set via

the control panel does not alter the Code V character sets. The effects of the ^IISO command are described below.

- When ^IISO is sent inside a pass:
- The selected ISO character set (also called language in the Code V manual) lasts only until the end of that pass.
- When ^IISO is sent outside a pass:
- The selected ISO character set is used both inside and outside passes until Code V is exited. Also, upon re-entering Code V, the ISO character set selection made during the previous Code V session is retained.

To form an ISO character set, ^IIISO only changes 12 characters from the lower US set. These 12 character differences are shown for each ISO character set on page 38.

The upper Code V character set contains the following characters not found in the Latin1 set:

Hex Value	Printed Character
80	Š
81	š
82	Ÿ
83	■
98	*f*
99	P_t
9A	Ç
9B	£

Compressed Fonts

Code V has the following nine compressed fonts for use in normal printing outside of a Graphics Pass.

- Near Letter Quality (NLQ) character sets at 10, 12, 13.3, 15, and 17.1 CPI.
- OCR-A and OCR-B 0.10 inch character sets at 10 CPI.

Emulation control codes are supported with Compressed Fonts Mode; escape sequences are not. To print characters within a Graphics Pass, see “Graphics Pass” on page 18.

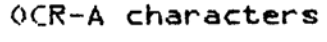
Command ^#n^-

Arguments

^#	Turns ON the Compressed Font Mode
n	Font selection.
	1 for NLQ (10 CPI)
	2 for NLQ (12 CPI)
	3 for NLQ (13.3 CPI)
	4 for NLQ (15 CPI)
	5 for NLQ (17.1 CPI)
	6 for OCR-A (10 CPI)
	7 for OCR-B (10 CPI)
	8 for Draft (12 CPI)
	9 for Draft (15 CPI)

To turn off the Compressed Fonts Mode, send the command ^#0^-

Example 1: The command ^#6OCR-A characters^- prints:



Example 2: The commands ^M0202000GRAPHICS PASS^-^*^#2^- COMPRESSED FONT^- prints:

NLQ Fonts

The NLQ fonts use 0.10 inch high Gothic characters whenever high quality print or high density is desired. NLQ fonts are limited to Horizontal Orientation only, and the NLQ Font command must be contained within a Graphics Pass. If the NLQ Font command is outside of a Graphics pass, it is interpreted as a Horizontal Repeat command (see Repeating Text later in this manual). The NLQ font is in effect only until the next command is received.

Any horizontal tabs and justification functions (see “Positioning and Repeating” on page 65) used with NLQ characters should be used prior to the

^Sn command. If a Tab or Justification Command is encountered within the NLQ characters, the selected NLQ Graphics font is terminated and a Block Character font is substituted for the remaining printable data.

To select one of the NLQ Graphics fonts, send the following command.

Command ^Sn

Arguments

^Sn	The NLQ Font Command.
n	NLQ Font
	1 for 10 CPI
	2 for 12 CPI
	3 for 13.3 CPI

4 for 15 CPI

5 for 17.1 CPI

6 for OCR-A (10 CPI)

7 for OCR-B (10 CPI)

Example 1: The Graphics Pass ^M0000000^S1A line of text^- (1/10th inch NLQ characters at 10 CPI) prints:

A line of text

Example 2: The Graphics Pass ^M0000000^S1A line^T0100of text^- (10 CPI) prints:

A line OF TEXT

Default Font Selection

This command allows selection of a default font that when issued inside a pass, selects the font used for the duration of that pass only. When used outside a pass, it selects the font used in all non-graphics printing.

Command ^IFONTS,S,n^G<data>

Arguments

^IFONTS,S,	Indicates that a new font style will be selected.
n	The number of the selected default font. This parameter can have up to five digits, although only the following values are used: 1 = Draft at 10 CPI 2 = Draft at 12 CPI 3 = Draft at 15 CPI 4 = Draft at 7.5 CPI 5 = NLQ at 10 CPI 6 = NLQ at 12 CPI 7 = NLQ at 13.3 CPI 8 = NLQ at 15 CPI 9 = NLQ at 17.1 CPI 10 = OCR-A (10 CPI) 11 = OCR-B (10 CPI) 12 = Symbol Set Low Density 13 = Symbol Set High Density
^G	Command Terminator
<data>	The data to be printed using the new font style.

Example: The human-readable characters can be printed in the OCR-A font style by sending the command ^M0202000^IFONTS,S,10^G12345^-, which prints:

12345

^M0202000	Sets up Horizontal orientation and the window size for the non-translated data characters at 0.2 inches (0.51 cm) square with no justification.

^IFONT,S,	Indicates that a new font style will be selected.
10	Selects OCR-A at 10 CPI.
^G	Command Terminator.
12345	Data to be printed using OCR-A at 10 CPI.

Draft Fonts

The draft fonts, when issued inside a pass, select the font used for the duration of that pass only. When used outside a pass, they select the font used in all non-graphics printing.

Command ^@Cnn

Arguments

^@C	Draft Font Command.
nn	One of the draft font numbers listed below: 05 = double high (0.2 inch), 7.5 CPI.
	10 = 10 CPI
	12 = 12 CPI
	13 = 13.3 CPI
	15 = 15 CPI
	17 = 17.1 CPI

Font/Quality Speed

Code V is capable of printing most of the Code V fonts at three different speeds and qualities. The slowest print speed produces the highest quality fonts, while the fastest print speed produces the lowest quality fonts. A Font Quality/Speed Command may be sent either inside or outside of a pass and remains in effect until another Font Quality/Speed Command is issued or Graphics Mode is exited.

These commands only affect the Code V Draft Font at 12, 13.3, 15, and 17.1 CPI and the Code V NLQ Font at 10, 12, 13.3, and 17.1 CPI.

The three Code V Font Quality/Speed commands are: High-Quality Font, Full- Space Font, and Half-Space Font.

High Quality Font Mode

Selects slow-speed, high quality fonts. The DPI of the font selected by a Code V Font Command depends upon the selected font and may cause the paper to back up and the shuttle to change speed. Fonts printed in this mode are positioned on character cell boundaries.

Command ^~FQ

Full Space Font Mode

Selects 120 DPI, medium speed, medium quality fonts. The paper will not back up to print any Code V font (excluding OCR) when ^~FF is selected. This is the default.

Command ^~FF

Half Space Font Mode

Selects 60 DPI, fast speed, lowest quality fonts. The paper will not back up to print any GPL font (excluding OCR) when ^~FH is selected.

Command ^~FH

Lines per Inch

The number of character lines per inch may be set with the command: Command ^@Lnn

Arguments

^@L	The Lines Per Inch Command.
nn	One of the lines per inch numbers below:
	03 = 3 LPI
	04 = 4 LPI
	06 = 6 LPI
	08 = 8 LPI

Dump Mode

The data received from the host may be printed in hexadecimal Dump Mode format using the Dump Mode command. Dump Mode is typically used to examine the data received from the host.

NOTE: Dump Modes are explained on page 37.

Command ^@Hn

Arguments

^@H	The Dump Mode Command
n	Mode selection
	0 or 1 selects Control Panel Dump Mode style 1.
	2 selects Control Panel Dump Mode style 2.
	3 selects Control Panel Dump Mode style 3.

Hex Command

The Hex Command allows the user to send single hexadecimal values to the printer. This is useful when you want to send a control code.

Command ^IHEX,xx

Arguments

^IHEX,	The Hex Command
xx	The hexadecimal value to send to the printer (00 - FF).

Control Panel Option

The following Control Panel option provides more control of speed and quality.

Block Character

This option allows you to set the size at which Block Character Smoothing begins.

NOTE: Smoothing slows down the Code V processing. The default is size 3.

ISO Character Sets

Several international standard character sets may be selected with this command. When issued inside a pass, this command selects the ISO language for the duration of that pass only. When issued outside a pass, it selects the language used in all non-pass printing.

There are 12 character positions which differ depending on the ISO language selected. See below.

NOTE: The ISO characters can only be used inside of a Graphics Pass when using the Twinax or Coax interface.

Command ^IISO,n^G

Arguments

^IISO,	Command Identifier, selects an ISO character set.
n^	The number of the desired character set.
^G	Command Terminator

ISO Character Set Table

Inside Graphics Pass

Language N	ISO	Hex	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
0	USA		#	$	@	[	\	]	^	`	{	¦	}	~
1	UK		£	$	@	[	\	]	^	`	{	¦	}	-
2	Swe/Fin		#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
3	Nor/Dan		#	$	@	Æ	Ø	Å	^	`	æ	ø	å	-
4	Japan		#	$	@	[	¥	]	^	`	{	¦	}	-
5	Germany		#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
6	France		£	$	à	°	ç	§	^	µ	é	ù	è	6
7	Italy		£	$	§	°	ç	é	^	ù	à	ò	è	ì
8	Spain		£	$	§	i	Ñ	¿	^	`	°	ñ	ç	~

Outside Graphics Pass

Language N	ISO	Hex	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
0	USA		#	$	@	[	\	]	^	`	{	¦	}	~
1	UK		£	$	@	[	\	]	^	`	{	¦	}	-
2	Swe/Fin		#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
3	Nor/Dan		#	$	@	Æ	Ø	Å	^	`	æ	ø	å	-
4	Japan		#	$	@	[	¥	]	^	`	{	¦	}	-
5	Germany		#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
6	France		£	$	à	°	ç	§	^	`	é	ù	è	6
7	Italy		£	$	@	°	\	é	^	ù	à	ò	è	ì
8	Spain		Ą	$	@	i	Ñ	¿	^	`	¨	ñ	}	~

3 *Creating Lines and Graphics*

Line Drawing

Code V allows printing of solid or dashed lines of various thicknesses and orientations. The following sections describe the line drawing commands and their usage.

Solid Lines

Description Solid lines can be printed horizontally and vertically ranging in width and length from one dot to 13.2 inches. The solid line command is:

Command ^LShhhdvvvd

Arguments

^LS	Solid Line command
hhh	The horizontal dimensions, specified in increments of 0.1 inches (0.25 cm), ranging from 000 to 132 (0 to 13.2 inches or 0 to 33.6 cm).
d	Additional dot columns, ranging from 0 to 9 dot-columns.
vvv	The vertical dimensions, specified in increments of 0.1 inches (0.25 cm), ranging from 000 to 132 (0 to 13.2 inches or 0 to 33.6 cm).
d	Additional dot columns, ranging from 0 to 9 dot-columns.

NOTE: The length and thickness of a line must be at least one dot.

Example 1: The command ^LS04000001^- prints a horizontal line exactly 4 inches (10.2 cm) long and 1 dot row thick.

^LS	Solid line print command.
040	A horizontal dimension of 4.0 inches (10.2 cm).
0	No dot column additions to the horizontal dimensions.
000	No vertical dimensions in increments of 0.1 inch (0.25 cm).
1	Additional vertical dimensions of 1 dot row.

Example 2: The command ^LS0500011^- prints a horizontal line 3.5 inches (8.9 cm) long and 0.1 inches (0.25 cm) plus 1 dot row thick.

^LS	Solid Line Print command
035	A horizontal dimension of 3.5 inches (8.9 cm).
0	No dot column additions to the horizontal dimensions.
001	A vertical dimension of 0.1 inches (0.25 cm).
1	Additional vertical dimensions of 1 dot row.

Example 3: The command ^LS00020110^- prints a vertical line 2 dot columns wide and 1.1 inches (2.79 cm) high.

^LS	Solid Line Print command
0002	A horizontal dimension of 2 dot columns.
0110	A vertical dimension of 1.1 inches (2.79 cm).

Dashed Lines

Description Dashed lines are drawn by printing alternate tenths of an inch (every 0.25 cm). The individual dashed marks will be oriented to the line's strongest dimension (horizontal or vertical). If the horizontal and vertical lengths are equal, the dashes will be horizontal. Dashed lines can be printed using the command:

Command ^LDhhhdvvvd

Arguments

^LD	Dashed Line command
hhh	The horizontal dimension, specified in increments of 0.1 inches (0.25 cm), ranging from 000 to 132 (0 to 13.2 inches or 0 to 33.6 cm).
d	Additional dot columns, from 0 to 9.
	vvv - The vertical dimension, specified in increments of 0.1 inches (0.25 cm), ranging from 000 to 132 (0 to 13.2 inches or 0 to 33.6 cm).
	d - Additional dot rows, from 0 to 9.

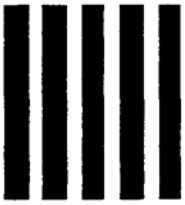
Example 1: The command ^LD02000010^- prints a horizontal dashed line exactly 2.0 inches 95.08 cm) long and 0.10 inches tall.

^LD	Dashed Line command
0200	Specifies a horizontal dimension of 2.0 inches.

Example 2: The command ^LD00100210^- prints a vertical dashed line 0.10 inch wide and 2.1 inches (5.3 cm) tall.

^LD	Dashed Line command
0010	Specifies a horizontal dimension of 0.1 inches.
0210	Specifies a vertical dimension of 2.1 inches.

Example 3: Dashed lines will print horizontally if the horizontal and vertical dimensions are the same. The command ^LD01000100 will print a series of vertical stripes 0.1 inches (0.25 cm) wide across a space that is 1.0 inches (2.5 cm) square.



^LD	Dashed Line command
0100	Horizontal dimension of 1.0 inches.
0100	Vertical dimension of 1.0 inches.

Boxes

Description Boxes can be drawn in a variety of shapes, sizes, and line weights. Maximum box size is limited to the maximum printed pass length and the width of the paper being printed upon.

NOTE: **NOTE:** When printing boxes, the minimum box length must be greater than twice the thickness of the vertical borders.

The minimum box height must be greater than twice the thickness of the horizontal borders.

The command for drawing a box is: Command ^LBhhhdvvvdhv
Arguments

^LB		The Box command.
	hhh	Horizontal length of the box in increments of 0.1 inches (0.25 cm) from 001 to 132 (0.1 to 13.2 inches or 0.25 to 33.53 cm).
	d	Additional horizontal length of the box in dot-columns, ranging from 0 to 9.
vvv		Vertical height of the box in increments of 0.1 inches (0.25 cm). d
		Additional vertical height of the box in dot columns, ranging from
		0 to 9.
	h	Thickness of the top and bottom borders of the box, ranging from 1 to 9 dot rows.
	v	Thickness of the two vertical borders of the box, ranging from 1 to 9 dot columns.

Example 1: The command ^LB0250010062^- prints the following box:

^LB	The Box command
0250	The lengths of the top of the bottom borders are 2.5 inches (1.27 cm) with no additional dot columns.
0100	Line lengths of the left and right borders are 1 inch each (2.54 cm) with no additional dot rows.
6	Line thickness for top and bottom borders is 6 dot rows.
2	Line thickness for the right and left borders is 2 dot rows.

Example 2: The command ^LB0250010025^- prints the following:

^LB	The Box command
0250	Line lengths for top and bottom borders of 2.5 inches (6.4 cm) with no additional dot columns.
0100	Line lengths for left and right borders of 1.0 inches (2.54 cm) with no additional dot rows.
2	Thickness of top and bottom borders is 2 dot rows.
4	Thickness of left and right borders is 4 dot columns.

Example 3: The command ^LB0010025046^- prints the following:

^LB	The Box command
0010	Line length for top and bottom borders is 0.1 inches (0.25 cm) with no additional dot columns.
0250	Line length for right and left borders is 2.5 inches.
4	Thickness for top and bottom borders is 4 dot rows.

Example 4: The Graphics Pass, ^M0202000^LB00900009011GRAPHICS^- prints the following box and characters:

^LB	The Box command
0090	Horizontal length of box is 0.9 inches.
0090	Vertical height of box is 0.9 inches.
1	Thickness of top and bottom borders is 1 dot row.
1	Thickness of right and left borders is 1 dot column.
GRAPHICS	Data to be printed with each character is 0.2 inches (0.51 cm) square.

Form Drawing

Description The GPL Form command is used to draw boxes with vertical lines ineside. The height of the vertical lines is restricted to the vertical dimensions of the box, and the maximum thickness of each line is limited to nine dot columns.

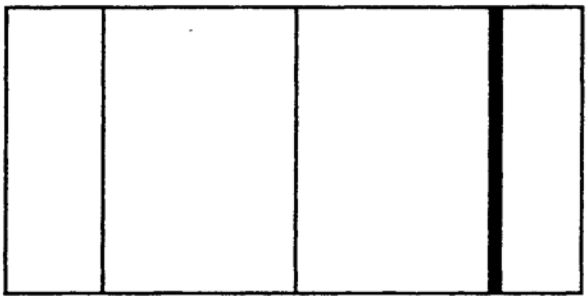
The Form command has the following format:

Command ^LFhhhdvvvdhvlllldt... ^G

Arguments

^LF	The Form command
hhh	Horizontal dimensions of the box in 0.10 inch increments from 001 to 132 (0.1 to 13.2 inches).
d	Additional horizontal dimensions of the box in dot-columns, ranging from 0 to 9.
vvv	Vertical height of the box in 0.10 inch increments.
d	Additional vertical height of the box in dot-columns,ranging from 0 to 9.
h	Thickness of the two horizontal borders of the box in dot rows, ranging from 1 to 9.
v	Thickness of the two vertical borders of the box in dot columns, ranging from 1 to 9.
lll	Horizontal positioning of the first vertical line from the left margin in 0.10 inch increments from 001 to 132 (0.1 to 13.2 inches).
d	Additional horizontal positioning of the first vertical line in dot columns, ranging from 0 to 9.
t	Thickness of the first vertical line in dot columns, ranging from 1 to 9.
...	Represents a repeat of the llldt command for each additional vertical line to be printed. The distance between each vertical line is additive, i.e., each additional vertical line is placed relative to the previous line, not from the left margin.
^G	Command Terminator

Example 1: The Graphics Pass ^LF03000150110050010100101003^G prints the following box and vertical lines.



^LF	The Form command
0300	Horizontal box dimension of 3.0 inches with no additional dot columns.
0150	Vertical box dimension of 1.5 inches with no additional dot rows.
1	Thickness of each horizontal border is 1 dot row.
1	Thickness of each vertical border is 1 dot column.
005	First vertical line is positioned 0.5 inches from left edge of box.
0	No additional dot columns added to position of first vertical line.
1	Thickness of the first vertical line is 1 dot column.
010	Second vertical line is positioned 1.0 inches from the first vertical line.
0	No additional dot columns added to position of second vertical line.
1	Thickness of second vertical line is 1 dot column.
010	Third vertical line is positioned 1.0 inches from the second vertical line.
0	No additional dot columns added to position of third vertical line.
3	Thickness of third vertical line is 3 dot columns.
^G	Command Terminator

Plot Mode

Description Plot Mode allows you to create and print your own graphics images. The Plot Mode command must be inside a Graphics Pass.

Plot data is based on vertical columns of seven dots each. Each column is described by a hex byte that specifies which of the seven dots to print. The Least Significant Bit (LSB) of the byte is mapped to the bottom of the column, while the eighth bit is ignored. See Table 2.

Table 2. Least Significant Bit

Hex Value	Bit	Dot
80	7	Not used
40	6	1 (top)
20	5	2
10	4	3

Table 2. Least Significant Bit

Hex Value	Bit	Dot
08	3	4
04	2	5
02	1	6
01	0	7 (bottom)

Command ^Q<plot data>^G

Arguments

^Q Enters Plot Mode

<plot data> The hex plot data.

^G Exits Plot Mode

Example 1: This example prints an X. The command sequence is ^M^Q41,22,14,08,14,22,41^G^-

NOTE: Commas are not required between the data and do not affect the results. Pass terminators placed between two plot commands will cause one extra blank dot row to be plotted between the plot images. To eliminate the blank row, keep subsequent plot commands within the same pass. To eliminate the blank row, keep subsequent plot commands within the same pass. Tab and justify commands can be used to position plot images anywhere in a pass.

The X is drawn dot by dot as illustrated in Figure 2

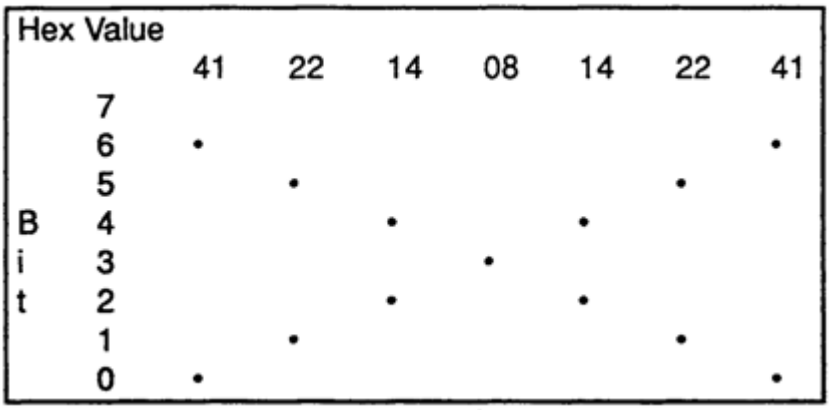


Figure 2. Plot Mode Example

Logos

A set of predefined logos come with Code V. See Appendix C, “Logos” on page 219 for a table of logos. The first twelve are available in all four orientations; the last two only work in horizontal and upside down orientations. There are two commands for printing logos: Version 1 Command and Version 2 Command. The Version 1 Command can only print the copyright and registered trademark logos, the Version 2 Command prints all the logos.

Version 1 Logos

The Version 1 logo command can only print the horizontal registered trademark and copyright logos. The logo command must be within a Graphics Pass.

Command ^Zn

Arguments

^Z	Version 1 logo command
n	Either 1 or 2. A value of 1 prints ® and a value of 2 prints ©.

Example: The command sequence ^M0404REGISTER^Z1^- prints:

REGISTER®

Version 2 logos

The Version 2 logo command prints all logos. The logo command must occur within a Graphics Pass.

Command ^ILOGO,d,nn^G

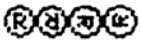
Arguments

^ILOGO	Version 2 logo command d Density: H=high, L=low
nn	A hex number indicating the logo to be used. See Appendix C, “Logos” on page 219 for a listing of logo numbers.
^G	Command Terminator

NOTE: If “d” is left out, density defaults are low.

Example: The command

^M0202000^ILOGO,L,43^G^U0202000^ILOGO,L,43^G
^V0202000^ILOGO,L43^G^E0202000^ILOGO,L,43^G^- prints:



Pixel Expansion

The Pixel Expansion command expands logos, fonts, and plot data up to 255 times their original size. The Pixel Expansion command remains in effect until a new Pixel Expansion command is received. To turn Pixel Expansion on, send the following code.

Command ^IPEXP,h,v^G

Arguments

^IPEXP	The Pixel Expansion command
h	Represents the horizontal Pixel Expansion value, which ranges from 1 to 255.
v	Represents the vertical Pixel Expansion value, which ranges from 1 to 255.
^G	Command Terminator

Example 1: The code ^M0101000^ILOGO,L45^G^- prints the UL symbol in its original size.

To enlarge the logo, send the Pixel Expansion code first, followed by the logo command.
For example, ^M0101000^IPEXP,3,3^G^ILOGO,L45^G^- prints:

Example 2: Sending the code ^M^IFONT,S,3^Gline of Text^- prints at the normal size:

Line of Text

Whereas sending the code ^M^IFONT,S,3^G^IPEXP,4,3^GLine of Text^- prints:

Line of Text

4 Barcodes

The CVCC provides the capability to print several standard barcode styles with or without accompanying human-readable characters. All barcode encoding algorithms reside in the printer. The user needs to only send the Barcode Command followed by the data to encode, and the printer will do the rest. The Barcode styles included in the CVCC are:

- Code 39
- Code 128
- Interleaved 2 of 5
- UPC A
- UPC E
- EAN
- Codabar
- Identicon 2 of 5
- MSI
- Postnet

Barcodes may be printed in all of the four standard orientations (in Version 2 only), with or without human-readable character translation in any of the available fonts, and at any height ranging from 0.3 to 9.9 inches. In addition, many of the above styles are available with or without check digits and all allow adjustment of bar/space ratios.

To print a barcode, the Barcode Command followed by the data to be encoded must be sent to the printer as part of a Graphics Pass. When the printer receives the Barcode Command, it first verifies that the data sent adheres to the established specifications for that barcode. If the data does not meet the specifications, the printer will print out the Barcode Command and data as an error message.

Some barcodes have a fixed number of digits: UPC and EAN, for instance. If less than the fixed number of digits is specified, the printer pads to the fixed number of digits with zeros on the left of the data. If more than the fixed number of digits is specified, the number is truncated and the remaining digits on the right are ignored.

Selecting Barcodes

Code V supports Version 1 and Version 2 Barcode commands. Version 1 commands allow more flexibility in the placement of the human-readable text portion of a barcode. Version 2 commands have the advantage of including the name of the barcode style in the command. Either command must be contained within a valid Graphics Pass for it to be recognized by the printer.

The Version 1 command allows the barcodes to be printed in two orientations:

Horizontal

Vertical (rotated 90 degrees clockwise from horizontal).

The orientation of a Version 1 barcode is independent of the orientation of the Graphics Pass it is in.

The orientation of Version 2 barcode is the same as the Graphics Pass in which it appears. Thus, Version 2 barcodes may be printed in any of the four Graphics orientations.

Regardless of the command style used, all Barcode commands must occur within a Graphics Pass.

Barcode Density

A control panel option (Darkbar) defines the Barcode density. If set to ON, the density will be high. The default setting of this option is OFF.

Barcode Height

The overall height of a barcode is determined by the window height (when the barcode is horizontal or upside down) or width (when the barcode is vertical right or vertical left) as specified in the ^M sequence that precedes the Barcode Command Sequence. Human-readable characters are included in the overall barcode height.

The height range is:

- 02 to 99 (0.2 to 9.9 inches) without human-readable characters (HRC).
- 03 to 99 (0.3 to 9.9 inches) with human-readable characters (HRC).

Character sizes less than 02 or 03 will cause the barcode height to assume the minimum value (i.e., 0.2 inches without HRC and 0.3 inches with HRC).

Barcode Width

The width of a barcode is a function of the number of data characters to be encoded, the style of barcode, and the bar/space ratios.

Barcode Version 1

The Version 1 command for printing a barcode is:

Command Default Ratio: ^{B,C}ax<data>^G

Variable Ratio: ^{B,C}a9x<ratio><data>^G

Arguments

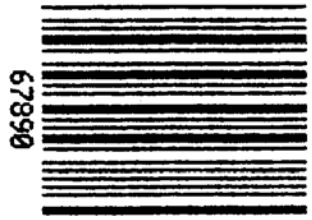
^{B,C}	B to print a horizontal barcode, or C to print a vertical barcode.
a	One of the following characters:
	B, prints the OCR-B font three dots below the barcode.
	N, turns OFF auto-printing.
	O, turns ON OCR auto-printing.
	S, prints OCR-A 0.10 inch below the barcode.
	T, prints OCR-B 0.10 inch below the barcode.
	Y, turns ON OCR-A auto-printing.
9	Selects the variable ratio feature.
x	The character from Table 3 that corresponds to the desired barcode style.
<ratio>	A sequence of bar/space ratios of the form r1r2r3r4 or r1r2r3r4r5r6r7r8, where r variables take on values 00 through 0F, hexidecimal.
<data>	The data to be encoded.
^G	Command Terminator

Example 1: The Graphics Pass ^M0515000^BNA12345^G^- prints:

^M0515000 Turns ON the Graphics Pass Mode. The barcode height is 0.5 inches and the vertical justification is 0 inches.

^B	Indicates the barcode orientation will be horizontal.
N	No human readable characters will be printed below the barcode.
A	The Code 39 barcode style (see Table 3).
12345	The data to be encoded as a barcode.

Example 2: The Graphics Pass ^M0515000^CYD67890^G^- prints:



^M0515000 Turns ON the Graphics Pass Mode. Barcode height is 1.5 inches with no vertical justification.

^C	Indicates the barcode orientation will be vertical.
Y	Non-OCR human-readable characters will print.
D	The Codabar barcode style (see Table 3).
67890	The data to be printed as a barcode and human- readable characters.

Barcode Version 2

The Version 2 command for printing a barcode is:

Command Default Ratio: ^IBARC,x,a,<data>^G

Variable Ratio: ^IBARC,x,R<ratio>,a,<data>^G

Arguments

^IBARC	Turns on the Barcode Mode.
x	The index name of one of the barcode styles selected from Table 3.
R	Specifies that a ratio specification sequence follows.
<ratio>	A sequence of bar/space ratios of the form r1r2r3r4 or r1r2r3r4r5r6r7r8, where r variables take on values 00 through 0F, hexidecimal.

NOTE: Colons must be used for Version 2 to separate the ratio parameters. a Controls the Autoprint feature (prints human-readable characters in the currently selected font). You can choose from:

N, turns off autoprinting.

E, human-readable characters are printed partially embedded into the bottom of the barcode region. The barcode is not printed in the embedded portion. This method of barcode printing is

<data>	The alphanumeric characters to be encoded.
^G	Command Terminator

Table 3 shows the available barcode styles and associated indices:

Table 3. Valid Barcode Symbologies

Version 1 Index	Version 2 Index	Description	Check Digits	Ratio*
---	AIAG	Code 39	None	1:1:3:3
---	EMBARC	Code 39	None	1:1:3:3
---	HIBCC	Code 39	Mod 43	1:1:3:3
A	C39	Code 39	None	1:1:3:3
---	LOGMAR	Code 39	None	1:1:3:3
B	C39A	Code 39	None	1:2:4:5
C	C39M43	Code 39	Mod 43	1:1:3:3
D	CBAR	Codabar	None	1:2:3:4:1:1:1:1
E	---	Identicon 2/5	None	1:2:3;2
F	MSI	MSI	None	1:1:2:2
G	MSI10	MSI	Mod 10	1:1:2:2
H	MSI1010	MSI	Mod 10/10	1:1:2:2
K	INT2/5	Interleaved 2/5	None	1:1:3:3
L	INT2/5A	Interleaved 2/5	None	1:2:4:5
P	UPCA	UPCA 11 digit	Mod 10	1:1:2:2:3:3:4:4
Q	UPCE	UPCE 10 digit	Mod 10	1:1:2:2:3:3:4:4
R	UPCE0	UPCE0 6 digit	Mod 10	1:1:2:2:3:3:4:4
S	UPCE1	UPCE1 6 digit	Mod 10	1:1:2:2:3:3:4:4
T	EAN13	EAN 13 digit	Mod 10	1:1:2:2:3:3:4:4
U	EAN8	EAN 8 digit	Mod 10	1:1:2:2:3:3:4:4
Z	C128	Code 128	Pseudo 103	1:1:2:2:3:3:4:4

* These ratio numbers are to be read in pairs. They refer to the relative ratios of the widths of the bars and gaps in barcode characters and correspond to narrow bar/narrow gap, wide bar/wide gap, respectively. For example, the first two numbers (1:2) in the INT2/5A barcode style indicates 1 unit for the narrow bar width and 2 units for the narrow gap. These units are measured in dot- columns for horizontal barcodes and dot rows for vertical barcodes.

NOTE: We cannot guarantee the barcode data with unique ratios other than those listed above will result in readable printer output. When using unique ratios, make sure that the wide and narrow bar ratios conform to requirements for readability.

Example 1: The following command is the Version 2 equivalent of Example 1 in the Version 1 barcode section: ^M0515000^IBARC,C39,N,12345^G^- prints:

^M0515000 Specifies that the barcode is horizontal with a height of 0.5 inches.

^IBARC Turns ON the Barcode mode.

C39 Selects barcode style Code 29 from Table 3, page 52.

N Turns OFF human-readable character printing.

12345 The data to be encoded into a barcode.

^G Command Terminator

Example 2: As an example of a Version 2 barcode with human-readable character translation, ^M0505000^IBARC,C39,B,12345^G^- prints:

^M0505000 Sets up horizontal orientation and sets the barcode height to 0.5 inches.

^IBARC, Turns ON the Barcode Mode.

C39, Selects the first barcode style in Table 3, page 52, (Code 39).

B, Places human-readable characters below the barcode.

12345 The data to be encoded as a barcode and printed as human readable characters.

^G Command Terminator

Example 3: Example 2 may be printed with the human-readable characters embedded in the barcode by replacing B with an E as shown below: ^M0505000^IBARC,C39,E,12345^G^- prints:

^M0505000 Sets up horizontal orientation and sets the barcode height to 0.5 inches.

^IBARC, Turns ON the Barcode Mode.

C39, Selects the first barcode style in Table 3, page 52, (Code 39).

E, Embeds human-readable characters at the bottom of the barcode.

12345 The data to be encoded as a barcode and printed as human readable characters.

^G Command Terminator

Example 4: The human-readable character translation printed below or embedded in a barcode may be printed in any of the fonts available with the IFONT command. The command sequence ^M0505000^IFONT,S,1^G^IBARC,C128,B,Rusty Old Car^G^- prints:

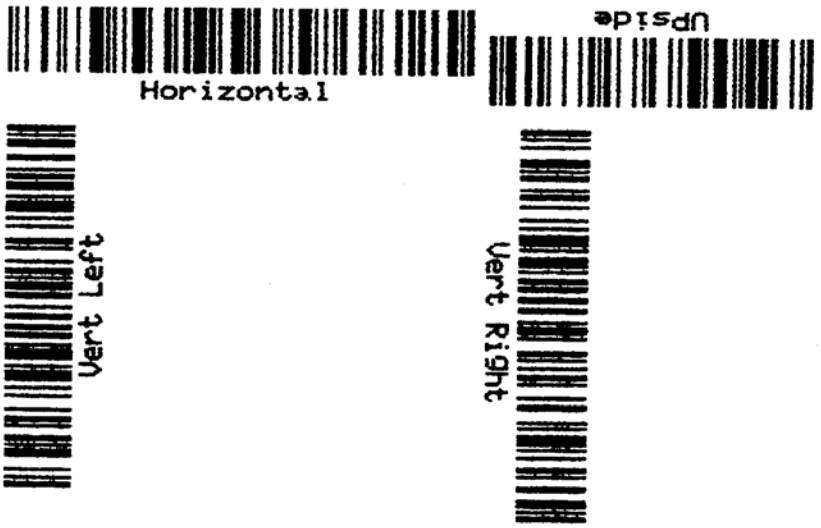
^M0505000	Sets up horizontal orientation and sets the barcode height to 0.5 inches.
^IFONT,S,1^G	Sets font to be Draft at 10 CPI.
^IBARC	Turns ON the Barcode Mode.
C128	Selects Code 128 (see Table 3, page 52).
B	Print human-readable characters below the barcode.
Rusty Old Car	The data to be encoded as a barcode with human- readable characters.
^G	Command Terminator

Example 5: The Graphics Pass ^U0505^IBARC,EAN8,E,1234567^G^- prints:

^U0505	Causes the barcode to be upside down with a height of 0.5 inches.
^IBARC	Turns ON the Barcode Mode.
EAN8	Sets the barcode style to EAN 8.
E	Embeds the human-readable character translation in the barcode.
1234567	Barcode data as well as human-readable characters.
^G	Command Terminator

Example 6: Barcodes may be printed in all four orientations. The following four Graphics Passes will print:

1. ^M0505000^IFONT,S,1^G^IBARC,C128,B,Horizontal^G
2. ^U0505000^T0250^IBARC,C128,B,Upside^G^-^*
3. ^E0505000^T0000^IBARC,C128,B,Vert Left^G
4. ^V0505000^T0250^IBARC,C128,B,Vert Right^G^-



First Command:

^M0505000^IFONT,S,1^G^IBARC,C128,B,Horizontal^G

^M0505000	Causes barcode to print horizontally with a height of 0.5 inches.

^IFONT,S,1^G	Selects Draft 10 CPI characters.
^IBARC,C128,	Turns ON Barcode Mode and selects Code 128 barcode.
B,	Prints the character translation below the barcode.
Horizontal	Data to be encoded as Code 128 barcode and printed below the barcode as Draft 10 CPI characters.
^G	Command Terminator

Second Command:

^U0505000^T0250^IBARC,C128,B,Upside^G^-^*

^U0505000	Causes barcode to print upside down with a height of 0.5 inches.
^T0250	Tabs the start of the barcode 2.5 inches from the left margin.
^IBARC,C128,	Turns ON Barcode Mode and selects Code 128 barcode.
B,	Prints the character translation below the barcode.
Upside	Data to be encoded as a Code 128 barcode and printed below the barcode as Draft 10 CPI characters.
^G	Command Terminator

Third Command:

^E0505000^T0000^IBARC,C128,B,Vert Left^G

^E0505000	Causes barcode to print vertical left with a height of 0.5 inches.
^T0000	Places the barcode at the left margin.
^IBARC,C128,	Turns ON Barcode Mode and selects Code 128 barcode.
B,	Prints the character translation below the barcode.
Vertical Left	Data to be encoded as a Code 128 barcode and printed below the barcode as Draft 10 CPI characters.
^G	Command Terminator

Fourth Command:

^V0505000^T0250^IBARC,C128,B,Vert Right^G^-

^V0505000	Causes barcode to print vertical right with a height of 0.5 inches.
^T0250	Tabs the start of the barcode 2.5 inches from the left margin.
^IBARC,C128,	Turns ON Barcode Mode and selects Code 128 barcode.
B,	Prints the character translation below the barcode.
Vertical Right	Data to be encoded as a Code 128 barcode and printed below the barcode as Draft 10 CPI characters.
^G	Command Terminator

Dark/Light Bar Ratios

The default widths of the light and dark bars that comprise each barcode style are determined by the ratios given in Table 3. These ratios are read in pairs, with the first number of the first pair being the width of the narrowest bar in dot columns, and the second number of the first pair being the width of the narrowest space in dot columns. The second pair of numbers in the ratio refer to the next wider bar and space, respectively.

For example, Code 39 has the ratio 1:1:3:3 which means:

- The narrow bars and spaces are each one dot column wide.
- The wide bars and spaces are each three dot columns wide.

If a barcode style has more than two pairs of numbers, it has more than two bar/space widths. For example, Code 128 has the ratio 1:1:2:2:3:3:4:4. Since this is comprised of four pairs of numbers, four different widths of bars and spaces are used in this particular barcode:

- The narrowest bars and spaces are one dot column wide.
- The next wider bars and spaces are two dot columns wide.
- The next wider bars and spaces are three dot columns wide.
- The widest bars and spaces are four dot columns wide.

Ratio numbers may range in value from 0 to F, hexadecimal. That is, a ratio value of A corresponds to decimal 10 and F corresponds to decimal 15.

NOTE: The printer does not verify that a given ratio sequence adheres to the standard barcode specification. Many barcode ratios will cause the barcode to fall outside the specifications, resulting in unreadability.

Example 1: Barcodes can be expanded by doubling the ratio of the light/dark bars. Code 39, which is normally 1:1:3:3 can be printed double wide by sending the Version 2 Graphics Pass,

^M0505000^IBARC,C39,R2:2:6:6,E,12345^G^- prints:

Example 2: Barcodes can also be expanded by replacing the ratio numbers in Table 3, page 52, with other values that are not exact multiples. The Version 2 Graphics Pass,

^M0505000^IBARC,UPCA,R2:3:4:6:6:8:8:9,E,1234567890^G^- prints:

High Density Barcode

Normally, barcodes print at 60 DPI horizontally. For improved readability, they may also print at 120 DPI. The 120 DPI, High Density Mode can be toggled ON and OFF by the command:

Command KF

Example: The command sequence ^M0505000^KF^IBARC,C39,B,MAU THE CAT^G^KF prints:

^KF first turns the High Density mode for barcodes ON, then the barcode is printed. The second ^KF turns High Density Mode OFF.

NOTE: Darkbar Control Panel settings overrides ^KF if Darkbar is ON.

LOGMARS Barcode

The LOGMARS barcode is a style used by the U.S. government. Although it was meant to print with the human-readable characters placed below the barcode (B), LOGMARS can also be used the with human-readable characters embedded in the barcode (E) or not selected (N).

Example: The Graphics Pass,

^M1010^IFONTS,S,10^G^IBARC,LOGMAR,E,NEWMEX^G^- prints:

Code 128 Barcode

While many barcode styles are limited to numbers, Code 128 can encode numbers, letters, and other symbols commonly found on keyboards, such as

@, #, and %. The Code 128 barcode is similar to three barcodes in one. The three barcode styles are called Code A, Code B, and Code C. Each barcode style is designed to encode certain types of data in the most compact way.

- Code A encodes upper-case alpha, numeric, and control codes.
- Code B encodes upper and lower-case alpha and numeric codes.
- Code C encodes digits in pairs.

A single Code 128 barcode may consist of all three Code 128 barcode styles. the printer chooses the styles which result in the most compact Code 128 barcode for the data to be encoded. Thus, the user does not need to be concerned about choosing the correct barcode styles since this is done automatically.

The > (greater than) symbol is a Special character prefix. If a character with a decimal value less than 32 needs to be encoded (i.e., a control character), then send a > followed by the character that is decimal 64 higher than the control code character. If control code characters are encoded, then no human-readable

Example 1: The Graphics Pass

^M0505000^IFONT,S,10^IBARC,C128,E,ABCDE^G^- prints:

Example 2: By adding the greater than symbol to the printable characters of Example 1, the Graphics Pass, ^M0505000^IFONT,S,10^IBARC,C128,E,>ABCDE^G^- prints a different barcode pattern, indicating that >A (decimal 65) now represents the control code SOH (decimal 1):

NOTE: The human-readable characters did not print under the barcode as it did in Example 1, above.

The Code 128 barcode style selection can also be done manually by adding a style selection character to the beginning of the barcode data.

Code 128 characters are uniquely identified by a Code 128 value that range from 0 to 105. For each value there is a single pattern of bars and spaces that form a Code 128 character (there are 107 Code 128 characters; the STOP character doesn't have a value). Code 128 barcode data is mapped by Code V into Code 128 values. Code V then uses the algorithm specified in the AIM USS-128 Specifications to determine the combination of codes that will produce the shortest possible barcode.

Some users may wish to control the Code 128 style used. To allow this, Code V has special commands that can be included in the command that allow the user to specify the Code 128 style that will be used. These special codes also allow the user to encode special characters or call special functions as defined in the Code 128 Specifications. Which special functions or characters will be called out is dictated by teh style of Code 128 barcode (A, B, or C) being used.

Table 4. Code 128 Special Command Characters

Special Character	Code 128 Hex Value	Code A Character	Code B Character	Code C Character
>0	30	>	>	30
>1	95	US	DEL	95
>2	96	FNC3	FNC3	96
>3	97	FNC2	FNC2	97
>4	98	Shift	Shift	98
>5	99	Code C	Code C	99
>6	100	Code B	FNC4	Code B
>7	101	FNC4	Code A	Code A
>8	102	FNC1	FNC1	FNC1

The following examples illustrate the Special Character usage

Barcode Commands	What a scanner will read
^IBARC,C128,B,>8123^G^-	F123 (all Code B)
^IBARC,C128,B,>7123^G^-	123 (all Code A)
^IBARC,C128,B,>6123^G^-	123 (all Code B)
^IBARC,C128,B,>3123^G^-	F123 (all Code B
^IBARC,C128,B,>2123^G^-	F123 (all Code B
^IBARC,C128,B,>123^G^-	123 (all Code B)
^IBARC,C128,B,>456>8123^G^-	456F,123
^IBARC,C128,B,>456>7123^G^-	456123 (456 in Code B, 123 in Code A)

Barcode Commands	What a scanner will read
^IBARC,C128,B,>456>6123^G^-	456F,123 (all Code B)
^IBARC,C128,B,>456>3123^G^-	456F,123 (all Code B)
^IBARC,C128,B,>456>2123^G^-	456F,123 (all Code B)

Code 128 Special Characters

When a style selection has been made, character data will be translated from the selected code style to the Code 128 representation as shown in Table 5.

NOTE: Codes 96 through 102 do not have corresponding ASCII character translations; these may be encoded using the special character table (Table 4, page 58).

Table 5. Code 128 Translation Table

Code 128 Value	Code A	Code B	Code C
0	(space	(space)	00
1	!	!	01
2	“	“	02
3	#	#	03
4	$	$	04
5	%	%	05
6	&	&	06
7	‘	‘	07
8	(	(	08
9	)	)	09
10	*	*	10
11	+	+	11
12	,	,	12
13	(hyphen)	(hyphen)	13
14	(period)	(period)	14
15	/	/	15
16	0	0	16
17	1	1	17
18	2	2	18

Table 5. Code 128 Translation Table

Code 128 Value	Code A	Code B	Code C
19	3	3	19
20	4	4	20
21	5	5	21
22	6	6	22
23	7	7	23
24	8	8	24
25	9	9	25
26	:	:	26
27	;	;	27
28	<	<	28
29	=	=	29
30	>	>	30
31	?	?	31
32	@	@	32
33	A	A	33
34	B	B	34
35	C	C	35
36	D	D	36
37	E	E	37
38	F	F	38
39	G	G	39
40	H	H	40
41	I	I	41
42	J	J	42
43	K	K	43
44	L	L	44
45	M	M	45
46	N	N	46

Table 5. Code 128 Translation Table

Code 128 Value	Code A	Code B	Code C
47	O	O	47
48	P	P	48
49	Q	Q	49
50	R	R	50
51	S	S	51
52	T	T	52
53	U	U	53
54	V	V	54
55	W	W	55
56	X	X	56
57	Y	Y	57
58	Z	Z	58
59	[	[	59
60	\	\	60
61	]	]	61
62	^	^	62
63			63
64	NUL	‘	64
65	SOH	a	65
66	STX	b	66
67	ETX	c	67
68	EOT	d	68
69	ENQ	e	69
70	ACK	f	70
71	BEL	g	71
72	BS	h	72
73	HT	i	73
74	LF	j	74

Table 5. Code 128 Translation Table

Code 128 Value	Code A	Code B	Code C
75	VT	k	75
76	FF	l	76
77	CR	m	77
78	SO	n	78
79	SI	o	79
80	DLE	p	80
81	DC1	q	81
82	DC2	r	82
83	DC3	s	83
84	DC4	t	84
85	NAK	u	85
86	SYN	v	86
87	ETB	w	87
88	CAN	x	88
89	EM	y	89
90	SUB	z	90
91	ESC	{	91
92	FS	|	92
93	GS	}	93
94	RS	~	94
95	US	DEL	95
96	FNC 3	FNC 3	96
97	FNC 2	FNC 2	96
98	SHIFT	SHIFT	98
99	CODE C	CODE C	99
100	CODE B	FUNC-4	CODE B
101	FNC 4	CODE A	CODE A
102	FNC 1	FNC 1	FNC 1

Postnet Barcodes

POSTNET barcodes are now available to all Code V users. You can now print POSTNET barcodes on envelopes or any printer output desired. The United States Postal Service defines POSTNET Barcodes in USPS Publication #25, A Guide To Business Mail Preparation.

Because of the strict specification requirements, POSTNET barcodes differ from other barcodes in several ways. These differences are discussed below following instructions on accessing POSTNET in Code V.

Accessing POSTNET in CVCC

To access POSTNET in Code V, use the following command:

^IBARC,POSTNET,N,98032^-

Orientation

By using a Graphics Pass command, you may position a POSTNET barcode anywhere on your printer output. (See Chapter 1, page 13, for instructions on using Graphics Passes). Since POSTNET barcodes are intended to be read by Postal Service scanners, you should position your barcodes where they may be easily read. USPS Publication #25 tells you where to position POSTNET barcodes.

Height

The Postal Service requires that POSTNET barcodes be a specific height, but you do not have to enter the correct height value when you use a Graphics Pass. When you enter a Graphics Pass command to position a POSTNET barcode, the correct height is automatically chosen for you. Any other height value you may enter is ignored in favor of the correct value.

CPI

The Postal Service requires POSTNET barcodes to be of specific resolution, as defined by dots per inch (DPI). POSTNET codes must be 105 DPI horizontally and 72 DPI vertically. For the height value, the correct POSTNET DPI values are automatically chosen. Any other DPI values you may enter in a Graphics Pass command will have no effect on the POSTNET barcode.

Other DPI values will however, effect characters other than POSTNET barcodes.

Parameters In POSTNET

When used with POSTNET barcodes, some parameters function differently than when used with other barcodes. These parameters are as follows:

^IBARC,POSTNET,B,98032^

The Below (B) parameter adds human-readable characters below the barcode, but unlike other barcodes, does not reduce the height of POSTNET codes.

^IBARC,POSTNET,E,98032^-

Normally, using the Embedded (E) parameter reduces the height of the barcode. However, because of POSTNET barcodes height requirements, characters that would normally be embedded are instead printed below the barcode, as if the B parameter were used.

^IBARC,POSTNET,N,98032-1122^-

Hyphens are ignored in POSTNET barcodes. Illegal characters (not digits or hyphens) are printed as zeros.

^IBARC,POSTNET,R1:4,N,98032^-

The required ratio for POSTNET barcodes is 1:4 (bar:gap), but may be modified using the Ratio (R) parameter. Ratios other than 1:4, however, do not meet the United States Postal Service specifications for POSTNET barcodes.

AIAG Barcodes

The AIAG barcode style is used by the automotive industry. It is actually a Code 39 style with no check digit and is encoded according to the AIAG specification.

Example: The Graphics Pass, ^M0505000^IFONTS,S,10^IBARC,AIAG,E,AIAG BAR CODE ^G^- prints:

HIBCC Barcode

The HIBCC barcode style is used by the health industry. It is actually a Code 39 barcode with a check digit. If the human-readable text is printed, it is bracketed on either side by an asterisk. Lower-case letters cannot be used with the HIBCC barcodes.

Example: The Graphics Pass, ^M0505000^IFONT,S,10^G^IBARC,HIBCC,B,HIBCC BAR CODE ^G^- prints:

EMBARC Barcodes

The EMBARC barcode style is used in the paper industry. EMBARC is a Code 39 barcode without a check digit.

Example: The Graphics Pass, ^M0505000^IFONT,S,10^IBARC<EMBARC,E,1234567^G^- prints:

5 Positioning and Repeating

Positioning Data

This chapter discusses the formatting and positioning commands provided by Code V.

Horizontal Tab

The Horizontal Tab command allows the user to specify the current print position relative to the left margin. When issued outside a pass, it sets the default left margin for all subsequent Graphics Passes. When issued inside a pass, it sets the left print position for that pass only.

The command syntax for a Horizontal Tab is:

Command ^Thhhd

Arguments

^T	Horizontal Tab Command
hhh	Horizontal distance from the left margin specified in increments of inches (0.25 cm) from 000 to 132 (00.0 to 13.2 inches or 0 to 33.5 cm).
d	Additional number of dot columns, from 0 to 9.

Example 1: The command ^M1010^T0152A^- prints an A tabbed 1.5 inches plus 2 dot columns to the right or the left margin:

^T0152	Places block character A 1.5 inches plus 2 dot columns right of the left margin.

Example 2: Since the tab command always specifies the print position relative to the left margin, it is possible to back up to previously printed positions within the line. For instance, the following example prints B before A even though A appears before B in the command sequence ^M0505^T0100A ^T0000B^-.

^T0100	Places block character A 1 inch right of the left margin.
^T0000	Places block character B at the left margin.

Vertical Justification

The print position may be changed vertically with the Vertical Justification command. This command allows the user to specify a print position below the top of the current Graphics Pass. The command syntax is:

Command ^Jxxd

Argument

^J	Vertical Justification Command
xxd	Specifies how far down from the beginning of the pass the characters are to begin printing. The xx is the justification in increments of 0.1 inches (0.25 cm) from 00 to 99. The d indicates an additional number of dot rows to justify down (from 0 to 9 dot rows).

Example: The Graphics Pass: ^M0303000OPT^J030ION^- prints:

^J030	Sets current print position to 0.3 inches below the top of the pass.

Pass Height

This function changes the height of block characters and barcodes within a Graphics Pass.

Command ^Hxx

Arguments

^H	Pass Height Command
xx	New window height in increments of 0.1 inches (0.25 cm) from 01 to 99. The actual height ranges from 0.1 to 9.9 inches (0.3 to 25.2 cm).

Example: The Graphics Pass, ^M0402000GRA ^H02PHIC^- prints:

^M	Horizontal orientation of characters.
04	Window height of 0.4 inches (1.02 cm).
02	Window width of 0.2 inches (0.51 cm).
00	No justification down.
0	No additional dot row justification down.
GRA	Data to be printed at 0.4 inches (1.02 cm) high.
^H	Indicates a change in character height.
02	New character height of 0.2 inches (0.51 cm).
PHIC	Data to be printed at 0.2 inches (0.51 cm) high.

Pass Width

This function changes the width of block characters within a Graphics Box. Command ^Wxx

Arguments

^W	Pass Width Command
xx	New window width in increments of 0.1 inches (0.25 cm) from 01 to 99. The actual width ranges from 0.1 to 9.9 inches (0.25 to 25.2 cm).

Example: The Graphics Pass, ^M0404000GRA ^W02PHIC^- prints:

^M	Horizontal orientation of characters.
04	Window height of 0.4 inches (1.02 cm).
04	Window width of 0.4 inches (1.02 cm).
00	No justification down.
0	No additional dot row justification down.
GRA	Data to be printed at 0.4 inches (1.02 cm) high.
^W	Indicates a change in character width.
02	New character height of 0.2 inches (0.51 cm).
PHIC	Data to be printed at 0.2 inches (0.51 cm) high.

Pass Density

This function changes the density (dots per inch or DPI) of barcodes and block characters. The density of lines and boxes is not changed by this command. The Pass Density command may be issued either inside or outside a Graphics Pass and remains in effect until another Pass Density command occurs or Graphics Mode is exited.

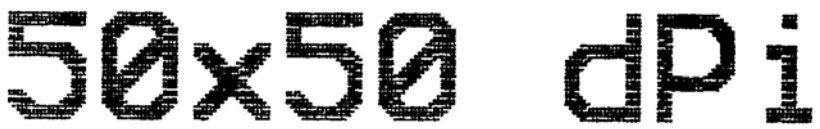
The default printing density is 60x72 DPI (60 DPI horizontally by 72 DPI vertically). The most common use of the ^~D command is to set the horizontal density to 70 DPI to decrease barcode size. Pass density is inversely proportional to size (higher density = smaller size). For example, a 2x increase in density results in a 2x size reduction.

Command ^~Dhhhvvv

Arguments

^~D	Pass Density Command
hhh	The horizontal density in dots per inch. The allowable range is from 040 to 240.
vvv	The vertical density in dots per inch. The allowable is from 040 to 240.

Example 1: The Graphics Pass, ^M0404000^~D05005050x50 dpi^- prints:



Example 2: The Graphics Pass,

^M0505000^~D070072^BYA1234567890^G^- prints:

Example 3: The Graphics Pass,

^M0505000^~D060072^BYA1234567890^G^- prints:

Dot Slew

The Dot Slew command moves the paper up by a user-specified number of dot rows. The command has the syntax:

Command ^Dnn

Arguments:

^D	The Dot Slew Command.
nn	Number of dot rows to move the paper up. the range is from 01 to 99.

Form Length

The Form Length command sets the number of lines in a form. This command must be used outisde a Graphics Pass. The command has the syntax:

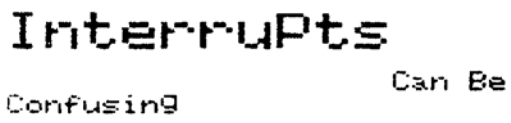
Command ^Lnn or ^Hnn

Arguments

^L or ^H	The Form Length Command. Either ^L or ^H may be used.
nn	The length of the form in lines. The valid range is from 01 to 99.

Interrupt Function

With the Interrupt function, a Graphics Pass may be interrupted at a designated vertical distance from the top of the pass to allow another Pass to begin printing at that location. The Interrupt command is not processed until a Pass Terminator is received. Any Code V commands between the Interrupt and the corresponding Pass Terminator will not be affected by the Interrupt. For example, in the Code V command sequence: ^M0202000Interrupts^I043^M0101031Can Be^-^M0101000Confusing^- prints:



The 0.2” x 0.2” characters Interrupts will print at the top of the pass. The 0.1” x 0.1” characters Can Be will print 0.3” + 1 dot row from the top of the pass. The 0.1” x 0.1” characters Confusing will print 0.4” + 3 dot rows from the top of the previous pass since Confusing is justified relative to the location of the Interrupt.

In contrast, in the Code V command sequence ^M0202000Interrupts^I043^-^M0101031Can Be^_^M0101000Confusing^- prints:

The 02” x 0.2” characters Interrupts will print at the top of the Pass. The 0.1” x 0.1” characters Can Be will print 0.7” + 4 dot rows from the top of the previous Pass since Can Be is justified relative to the location of the Interrupt (0.4” + 3 dot rows below the top of the pass). The 0.1” x 0.1” characters Confusing will print 0.8” + 4 dot rows from the top of the previous pass.

The format for an Interrupt function is:

Command ^Ixxd

Arguments

^I	Indicates the Interrupt function.
xx	Increments of 0.1 inches from the top of the Graphics Pass down to where data is inserted. The range is 00 to 99.
d	Additional distance in dot rows. The range is 0 to 9.

NOTE: ^I000 prints out all of the interrupted passes.

Repeating Data

The printed image defined by a sequence of Code V commands and data may be repeated horizontally and vertically through the use of Code V Repeat commands. This technique is often used to replicate labels, allowing the user to print numerous labels from a single command sequence. Using a Code V Repeat command typically results in faster printing since the printer only needs to process the image to be printed once, then simply duplicates the image across the page.

There are two Repeat Command versions. All of the features of the Version 1 Repeat commands are included in the Version 2 commands. For this reason, it is recommended that the more powerful Version 2 commands be used whenever possible. The Version 1 Repeat commands are discussed on page 76.

Repeating Data, Version 2

Horizontal Repeat, Version 2

Horizontal repetition can be accomplished by bracketing the commands and data to be repeated with the commands shown below.

NOTE: Non-graphics data (i.e., any data outside a Graphics Pass) will not be repeated. The Version 2 Horizontal Repeat command syntax is:

Command ^IREPH,n,hhd^G <data>^IREPE

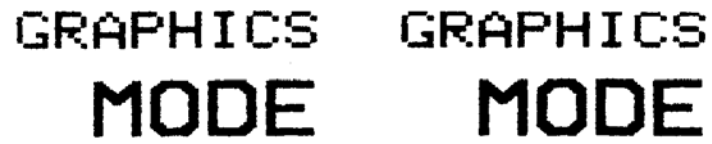
Arguments

^IREPH,	Turns ON the Horizontal Repeat Mode.
n,	The number of times the data is to be repeated (1 to 9999).
hh	Horizontal distance in increments of 0.1 inches to move the print postion relative to the start of the previously printed <data> (0 to 99).
d	Number of addtitional dot columns to mvoe the print position (0 to 9).
^G	Command Terminator
<data>	This represents a sequence of Code V commands and data.
^IREPE	Repeat Terminator, turns OFF the Horizontal Repeat Mode.

Example 1: The command ^IREPH,2,200^G^M0202000REPEAT^_^IREPE^- prints:

^IREPH,	Turns ON the Horizontal Repeat Mode.
2,	The number of times the data is to be repeated.
200	2.0 inches between the start of each repetition with no additional dot columns.
^M0202000REPEAT^	The Graphics Pass containing the character windows and the data to be repeated.
^IREPE	Repeat Terminator, turns OFF the Horizontal Repeat Mode.

Example 2: The Graphics Pass ^IREPH,2,200^G^M0202000GRAPHICS^- ^*^M0303000^T0040MODE^- ^IREPE^- prints:



^IREPH,	Turns ON the Horizontal Repeat Mode.
2,	The number of times the data is to be repeated.
200	2.0 inches between the start of each repetition with no additional dot columns.
^M0202000	Sets the orientation and size of the character windows for the first Graphics Pass.
GRAPHICS	The data to be printed.
^-^*	Functions as a carriage return and line feed.
^M0303000	Sets the orientation and size of the character windows for the second Graphics Pass.
^T0040	Tabs the beginning of the second Graphics Pass 0.4 inches from the left margin.
MODE	The data to be printed.
^IREPE	Repeat Terminator, turns OFF the Horizontal Repeat Mode.

Vertical Repeat, Version 2

Vertical repetition can be selected by bracketing the commands and data to be repeated with the commands shown below. Both Graphics and non- Graphics commands and data are repeated vertically.

Command ^IREPV,n,vd^G <data>^IREPE

Arguments

^IREPV	Vertical Repeat Command
n	The number of times the data is to be repeated (1 to 9999).
v	Vertical space between each repetition in 0.10 increments (1 to 132).
d	Additional vertical space between repetitions in dot rows (0 to 9).
^G	Command Terminator
<data>	The data to be repeated.
^IREPE	Repeat Terminator, turns OFF the Horizontal Repeat Mode.

Example 1: The Graphics Pass ^IREPV,3,12,^G^M0202000REPEAT 3 TIMES^-^IREPE^- prints:

REPEAT 3 TIMES
REPEAT 3 TIMES
REPEAT 3 TIMES

Example 2: The command sequence ^IREPV,3,13^G^M0202000GRAPHICS PASS^-^*NON-GRAPHICS SEQUENCE^*^IREPE^- mixes

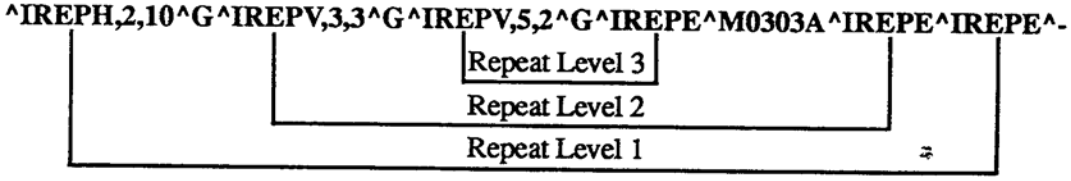
Graphics Pass text with standard text and prints the following:

GRAPHICS PASS
NON-GRAPHICS SEQUENCE
GRAPHICS PASS
NON-GRAPHICS SEQUENCE
GRAPHICS PASS
NON-GRAPHICS SEQUENCE

Multiple Repeat Commands

A Graphics Pass may contain more than one Repeat Start/Repeat End pair. When a Graphics Pass contains more than one pair of Repeat Start/Repeat End commands, the additional pairs are described as being “nested” inside the Graphics Pass. The nested pairs are matched by pairing the first Repeat End with the last Repeat Start continuing in sequence until the last Repeat End is paired with the first Repeat Start.

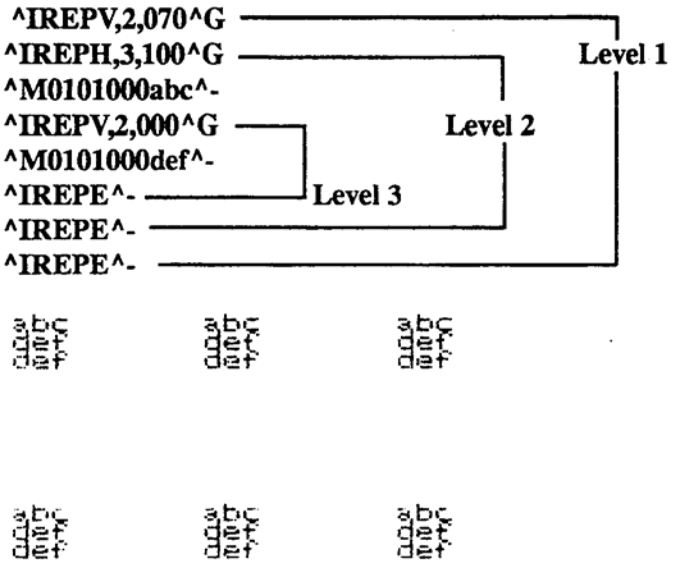
In the example below there are two Vertical Repeat pairs nested in the Graphics Pass. Each nested pair comprises a Repeat Level in the Graphics Pass. The Repeat level at a given point within a Graphics Pass is defined as the number of Repeat start commands minus the number of Repeat End commands previously encountered. For instance, the A is at the Repeat Level 2 in the following Code V command sequence:



NOTE:

- There can only be 9999 repeats per repeat level.
- The maximum number of repeat levels is 10.
- If multiple Repeat Start commands are set, each Start Repeat must be paired with an End Repeat.
- No printing occurs until all Repeat Commands are terminated.

Example 1: this Graphics Pass has three Repeat levels:



^iREPV,2,070^G

Specifies a Vertical Repeat twice at 0.7 inches separation.

^IREPH,3,100^G

Specifies a Horizontal Repeat three times with a separation of 1 inch.

^M0101000abc^-

First Graphics Pass with data and Pass Terminator.

^IREPV,2,000^G

Specifies a Vertical Repeat of two more times with no separation.

NOTE: Although the ^IREPV command specifies 0 inches of vertical separation between the repeated pass def, the def strings appear on separate lines because of the ^- Pass Terminator.

^M101000def^-

Second Graphics Pass with data and Pass Terminator.

Example 2: This example has three levels. The nested repeat commands:

^IREPV,2,050^G

^M0101000Level A^-^*

^IREPH,2,150^G

^M0101000^T003Level B^-^*

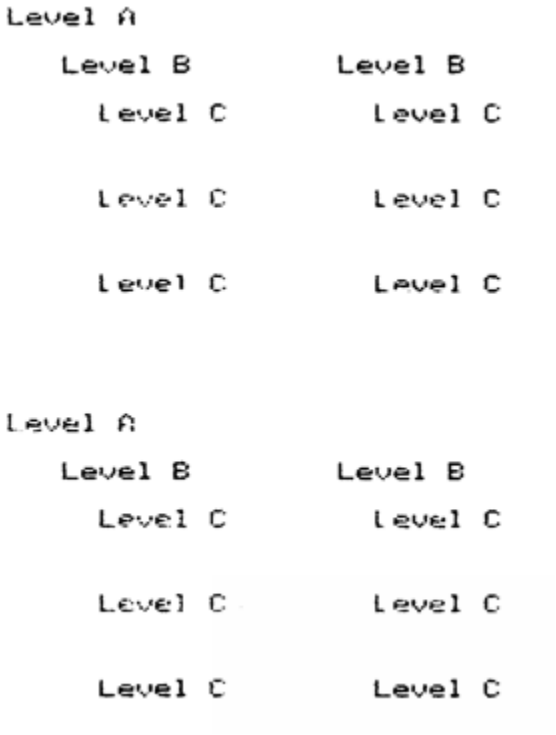
^IREPV,3,020^G

^M0101000^T005Level C^-^*

^IREPE^-

^IREPE^-

prints:



^IREPV,2,050^G

This command causes Level A and all the following passes to be repeated twice vertically, with a spacing of 0.5 inches.

Level A Printed data.

^IREPH,2,150^G

This command causes Level B and all other Graphics Passes within this level to be printed twice horizontally with a spacing of 1.5 inches.

^T003 Tabs the two columns of Level B data 0.3 inches from the left margin.

Level B Printed data.

^IREPV,3,020^G

this command causes Level C to print three times vertically with a spacing of 0.2 inches.

^T005 Tabs the two columns of Level C data 0.5 inches from the left margin.

Level C Printed data.

Automatic Increment/Decrement

Data items within a repeat sequence can be increased or decreased in predetermined amounts using the following increment or decrement syntax.

Command <Begin Repeat Command>^Yx+/-z^G<End Repeat Command>

Arguments

^Y Turns ON the Automatic Increment/Decrement Mode.

x	The operand (the value you start with in any arithmetic operation). This value can be up to twelve characters long. Valid characters are 0 to 9, the capital letters A to Z, and spaces.
+/-	Select either the + or the - sign to determine whether the operand will be incremented or decremented, respectively.
z	The operator (the amount that is incremented or decremented at each step). This must be a numeric value up to 12 digits long. If the operator exceeds 12 digits or if it contains non-numeric characters (other than spaces), the operand remains unchanged.
^G	Increment/Decrement Terminator

Incrementing or decrementing numbers and characters is similar to adding or subtracting numbers. Align the numbers and/or letters vertically in columns and add or subtract each column. For example, K + 1 means to advance 1 character position from K to the next letter (i.e., K+1 = L and K-1 = J).

The following examples show how automatic increments (addition) and decrements (subtraction) apply to numeric and character fields:

0A	0	A9	0A	(Operand)
+1	-1	+1	+11	(Operator)
0B	9	B0	1B	(Answer)

In automatic incrementing and decrementing, the alphabet wraps around itself, meaning that A is one character position beyond Z. The normal arithmetic rules of carrying or subtracting a 1 from other columns when adding or subtracting also apply. For example:

0Z	1B	(Operand)
+1	-2	(Operator)
1A	0Z	(Answer)

Below are two additional examples showing how this carry function operates:

0T	0A	(Operand)
+17	+27	(Operator)
2A	2H	(Answer)

The answer is always limited to the total number of characters and spaces in the operand, even if the operator value has the full twelve characters allowed. For example:

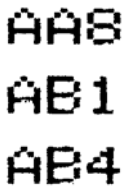
Y	2F	16A	(Operand)
+112	-105	+1234	(Operator)
A	2A	39E	(Answer)

If spaces are included in the operand value, only numeric values can appear in the answer for those same columns. No letters are permitted. For example:

(1 space) Y	2F	(Operand)
+112	-105	(Operator)
2A	2A	(Answer)

Example 1: The Auto Increment Command,

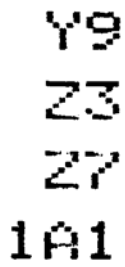
^IREPV,3,0^G^M0202000^YAA8+3^G^-^*^IREPE prints:



^Y	Turns ON the Automatic Increment/Decrement function.
AA8	The operand (the starting value).
+3	The operator. This is the amount by which the operand is incremented.
^G^-^*	Completes the automatic incremental command and performs a Carriage Return/Line Feed.
^IREPE	Turns OFF the automatic increment/decrement function.

Example 2: The Auto Increment command,

^IREPV,4,0^G^M0202000^Y Y9+4^G^- ^*^IREPE- prints:



^Y	Turns ON the Automatic Increment/Decrement function. The space before Y9 means that three characters are permitted in the answer.
+4	The operator. This is the amount by which the operand is incremented.
^G^-^*	Completes the Automatic Incremental Command and performs a carriage return/line feed.
^IREPE	Turns OFF the automatic increment/decrement function.

Example 3: This example shows how to print sequentially numbered labels.

The Automatic Increment command:

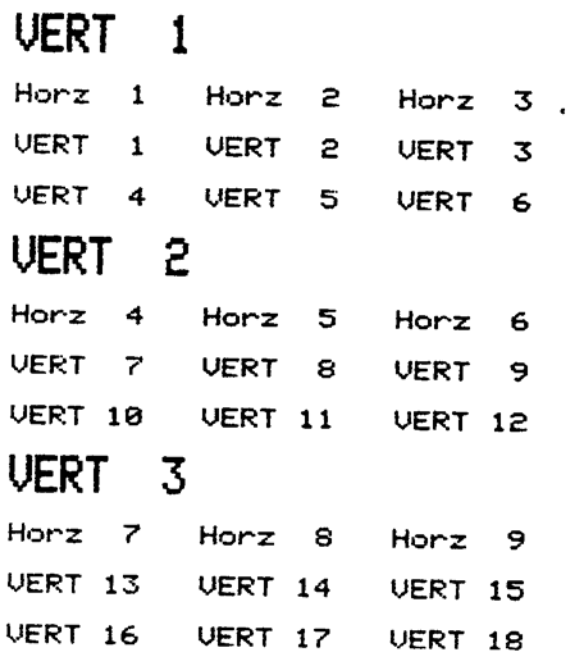
^IREPV,3,0^G^M0000Vert ^Y1+1^G^-

^IREPH,3,100^G^M0101Horz ^Y1+1^G^-

^IREPV,2,0^G^M0101VERT ^Y1+1^G^-

^IREPE^IREPE^IREPE^-

prints:



Repeating Data, Version 1

Horizontal Repeat

Horizontal Version 1 repetition can be accomplished by bracketing the commands and data to be repeated with the commands shown below:

Command ^Snntt^-<data>^-^S^-

Arguments

^S	Version 1 Horizontal Repeat Command
nn	Number of times to repeat (00 to 99).
tt	Number of 0.10 increments to move after each repeat.
<data>	Data to be repeated.
^S^-	Version 1 Horizontal Repeat Terminator.

NOTE: Auto increments/decrements are not supported within a Version 1 horizontal repeat.

Example: The command sequence, ^S0225^-^M0202Horz^_^S^- prints:

Vertical Repeat

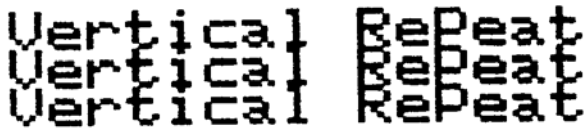
Vertical Version 1 repetition can be accomplished by bracketing the commands and data to be repeated with the commands shown below:

Command ^Rnnnn^_<data>^-^Z^-

Arguments

^R	Version 1 Repeat Command
nnnn	Number of times to repeat (0000-9999)
<data>	Data to be repeated.
^Z^-	Version 1 Vertical Repeat Terminator.

Example: The command sequence ^R0003^-^M0202Vertical Repeat^-^- prints:



NOTE: Version 1 Vertical Repeats may be nested only two levels deep (i.e., an outer repeat and an inner repeat).

6 *Buffered/Defined Forms*

Buffered Forms

Code V allows users to store commands and data in printer memory. The data stored in memory may be named and called up for use any time. Data stored in memory this way is called a buffered form. The buffered form can consist of any data and commands sent to the printer.

The Graphics Processing Language emulates both Version 1 and Version 2 of the Code V Buffered Formatting Language. Version 2 buffered formatting offers more features than Version 1 and should be used whenever possible instead. See “Version 1 Buffered Formatting” on page 90.

The printer uses a Heap Buffer space to store Buffered Forms and to process Code V data. Each Buffered Form takes up a portion of the available Heap Buffer space. The amount of work space memory available at any given time may be determined by executing the Buffered Forms List Command, see page 83.

Data Fields

The printer allows you to store the basic format for an entire form and then later send only the data that changes from one form to another. The position in each form where this data is inserted is called a data field. A field data accomplishes two purposes.

1. It specifies the number of data bytes.
2. By its physical location in the buffered form, the data field specifies where its bytes should be placed. A data field is identified by ^[nnn.

^[	Identifies the beginning of the data field.
nnn	The number of bytes in each field. This ranges from 001 to 999. One byte of data represents a single character, such as 6 or r. the actual number of bytes inserted into each field may be less than the specified maximum, but how the unused spaces are treated depends upon which control code is used as input to the buffered data field.

Buffered Form Create

Buffered forms are created by the Buffered From Create Command. When a buffered form is created, it is stored in printer memory until it is deleted or the printer exits Graphics Mode. Every Version 2 Buffered Form has a name by which it is referenced. The name is assigned to the Form by the Create Command, and the form is referenced by name by the Buffered Form Execute Command. The Execute Command is then used to print the form created by the Create command.

The Buffered Form Create command has the following syntax:

Command ^IFORM,C<form name>^G<data^]

Arguments

^IFORM,C	Buffered Form Create Command
<form name>	The name for the form being created and stored. This name can be up to twelve characters long. If this name is already taken by an existing buffered form, the newly created form will replace it.
^G	Command Terminator

<data> Commands and data to be stored (including data fields, if any).

^] Marks the end of the data sequence that follows the ^IFORM,C command.

Example 1: The command, ^IFORM,C123^G^M1010000123^-^] stores the Graphics Pass ^M1010000123^- in memory and names it 123.

Example 2: The command ^IFORM,CTEST 1^G^M0505000^[006^-^] stores the Graphics Pass ^M0505000xxxxxx^- in memory and names it TEST 1. the xxxxxx signifies six bytes in memory that are to be filled in by the data field of the Buffered Form Execute command. If the Execute command data field contains ABCDEF, then the data string ABCDEF will print as 0.5 x 0.5 inch block characters horizontally across the page in response to the ^M0505000 command.

Buffered Form Execute

The Version 2 Buffered Form Execute command is used to print command/ data sequences defined previously by a Create command. This command prints the corresponding form stored in memory and inserts the data into the appropriate data fields (if any) as the buffered form is printed. The Execute command has the following form:

Command ^IFORM,E<form name>^G<data>^G

Arguments

^IFORM,E Buffered Form Execute Command

<form name> The name of the form stored in memory.

^G Command Terminator

<data> Data to be entered into the data fields of the Buffered Form.

^G Command Terminator

Example: To print the form defined as TEST 1 in Example 2, page 79, send the following command:

^IFORM,ETEST 1^GABCDEF^-^G

This would print ABCDEF as 0.5 x 0.5 inch block characters:

Control Code Command Changes

Within the Buffered Form Execute <data> field, the ^-, ^+, ^*, and ^, commands do not perform their usual functions, but instead behave as described below:

^- Causes unfilled portions of the current data field to be padded with spaces. This is used if there are insufficient data characters to fill the current field. The entire data field is converted to spaces if this is the only input in the field.

^+ Deletes any unused portions of a data field. No data will be printed in the data field if this is the only input in the field.

^* This is identical in function to the ^- function, in that it causes the rest of the current data field to be padded with spaces. The entire data field is converted to spaces if this is the only input in the field. All remaining fields in the buffer are also converted to spaces unless the fields occur within a Copy Command (as described in the “Buffered Form Copy” on page 84).

^, This command performs a line feed unless it is used within a Copy Command. Refer to “Buffered Form Copy” on page 84.

The following examples illustrate the use of buffered formatting.

Example 1: This example is a two step process. A form with the file name EXAMPLE 1 will be created in the first step, then executed in the second step.

1. To create the buffered form, send the command:

   ^IFORM,EEXAMPLE 1^G12345678901^-^G

   This creates the basic label and specifies the size and position of the data field and how many bytes (characters) the data field will hold.

2. Call the form and send the data for the data field with the command:

   ^IFORM,EEXAMPLE1^G12345678901^-^G

   ^IFORM,EEXAMPLE 1^G

   Gets the buffered form EXAMPLE1.

   12345678901^-

   The data being inserted into form EXAMPLE 1.

   ^G Marks the end of the data (Data Terminator).

The data 12345678901 is inserted into the three byte wide data field in the buffered form, 3 bytes at a time. The first line prints “Always 3 bytes: 123” then there is a CRLF, followed by “Always 3 bytes: 456” and another CRLF, and so forth until all the data in the Execute Command has been used. The printed result of these Buffered Form commands is:

```
Always 3 bytes:123
Always 3 bytes:456
Always 3 bytes:789
Always 3 bytes:01
```

Example 2: The buffered form is a more complex design. It is a sequence of Code V commands that can be used to print an AIAG form:

1. The following commands create an AIAG buffered form in printer memory:

^F^-

^IFORM,CAIG^G

^M0504^t0100^j002^[009^T0030^J054^BNAP^[009^G

^M0503^T0100^j122^[006^T0030^J174^BNAQ^[006^G

^M0202^T0100^J232^[009^M0504^T0030^J254^BNAV^[009^G

^M0202^T0100^J312^[009^M0504^T0030^J334^BNAM^ [009^G

^T0000^J000^LB05000-40011

^T0000^J120^LS05000001

^T0000^J230^LS04000001

^T0300^J120^LS00010110

^T04400^J230^LS00010170

^T0004^J122QUANTITY^J133^T0025 (Q)

^T0004^J232 SUPPLIER^J241^T0025 (V)

^T0004^J312 SERIAL^J323^T0025 (S)

^M0100000^T0030^J386

^[055^-^*^

2. The following Execute command inserts data into the data fields and prints the AIAG form as follows:

   ^IFORM,EAIG^G 1234567^+

   1234567^+

   12345^+12345^+

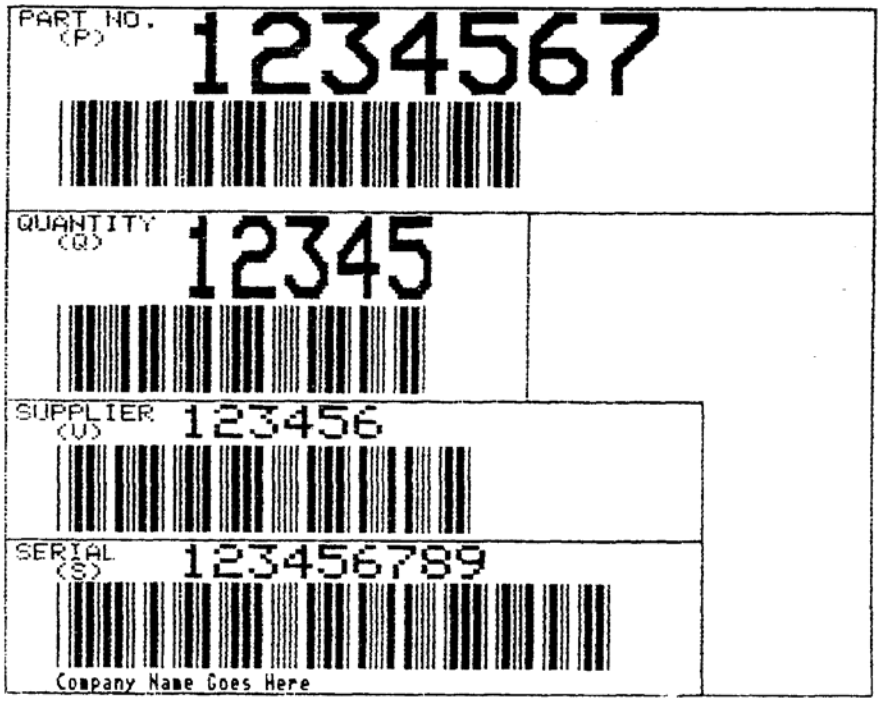
   123456^+

   123456^+

   123456789

   123456789

   Company Name Goes Here ^+^G



Buffered Form Delete

Any existing form or label can be deleted from the printer's memory if additional space is needed to create a new form. This is done through the command:

Command ^IFORM,D<name>^G

Arguments

^IFORM,D	Buffered Form Delete Command
<name>	the name of the form to be deleted.
^G	Command Terminator

Buffered Form Reset

This command deletes all user-defined forms stored in memory. The command syntax is:

Command ^IFORM,R

Buffered Form List

This command prints a list of all forms currently in memory, the number of bytes needed for each form, the total number of bytes used, and the number of bytes still available. This is done through the command:

Command ^IFORM,L

Example: Issuing the above command prints:

FORM NAME	FORM SIZE
FORM_LIST	346
TEST 1	25
EXAMPLE 1	42
AIAG_F.MT	621
AIAG	575
METALS_F.MT	1516
EXAMPLE1	82
BYTES USED	3207
BYTES AVAILABLE	40312

Buffered Form Repeats

Two types of Repeat commands are allowed within a buffered form: the Buffered Form Copy command and the Buffered Form Repeat command. Both the Copy and Repeat commands serve to repeat a sequence of Create

<data> a certain number of times. The difference between them is that the number of times a Copy command repeats data is determined in the Create command sequence, whereas the number of times the Repeat command repeats data is determined in the Execute command sequence.

Buffered Form Copy

The Buffered Form Copy command is used to repeat a sequence of commands and data within a buffered form Create <data> sequence. The command for this is:

Command ^Cnn-<data>^Z^-

Arguments

^C	Copy Command
nn	The number of times the data is to be repeated (01 - 99).
<data>	The data to be repeated.
^Z^-	End-of-Copy Terminator

Enough data must be included in the Buffered Form Execute to satisfy the copy counts in the form. Data can be different for each copy. The following control codes take on special functions within a Buffered Form Copy/Repeat command:

^-	(Caret hyphen) Fills the rest of the current data field with spaces.
^*	(Caret asterisk) Causes all unfilled character positions in all data fields to be filled with blanks up to the ^Z (End-of-Copy/Repeat) command.
^,	(Caret comma) Causes all unfilled character positions in all data fields to be filled with blanks up to the ^Z (End-of-Copy/Repeat) command, and all character positions in any remaining data fields will be filled with blanks until the copy count is satisfied.

Example 1: ^F^- Free Format ON
^IFORM,CEXAMPLE1^G
^C03^-^M0101This line prints 3 times^-^Z^-
^C02^-^M0201This line prints ^[020^-^Z^-
^C04^-^M0202This line prints ^[020^-^Z^-
^]
^IFORM,EEXAMPLE1^G
times^-2 times^-
times^0
3 times^0
3 times^0
^*
^G
^O^ Free Format OFF
prints:

```
This line Prints 3 times
This line Prints 3 times
This line Prints 3 times
This line Prints 2 times
This line Prints 2 times
This line Prints 3 times
This line Prints 3 times
This line Prints 3 times
This line Prints
```

Repeat

The Repeat command repeats a sequence of Code V commands and data a user-specified number of times. The number of times to repeat a command or data sequence is specified in the data section of a Buffered Form Execute command. The manner in which this is done is best understood by referring to Example 1 below.

The Repeat command has the following format:

Command ^R^<data>^Z^-

Arguments

^R^	Repeat Command
<data>	The data to be repeated.
^Z^-	End of Repeat Terminator

Example 1: The Repeat command always occurs within the Buffered Form Create command data sequence. The number of times to repeat a portion of the data sequence is specified within the buffered Form Execute command data sequence.

NOTE: The Repeat command is used in the following Code V sequence:

^IFORM,CEXAMPLE1^G^R^-^M0201000Repeat ^[020^-^*^*^Z^-^]

^R^-	The beginning of the Buffered Format Repeat command. Everything between the ^R^- and the ^Z will be repeated.
^[020	The data field with a maximum size of 20 bytes.
^Z^-	The end of the commands and data to be repeated.

Now we want to execute the Form EXAMPLE1. The first four bytes of the Execute data field must specify the number of times to repeat the data sequence between the ^R and the ^Z. the next 20 bytes will be read from the Execute data sequence, then the following four bytes specify the number of times to repeat the data sequence again, and so forth until all the data in the Execute command is used up.

NOTE: Each time the Create data sequence is repeated by the Repeat command, the same Execute data is used to fill the data field. The Copy command, however, inserts new Execute data into the data field each time the data field is repeated.

The Execute command sequence for EXAMPLE1 is:

^IFORM,EEXAMPLE1^G

0003 3 Times^+(Repeats 3 Times, three times.

0002 2 Times^+(Repeats 2 Times, twice)

^G

prints:

REPEAT 3 Times

REPEAT 3 Times

REPEAT 3 Times

REPEAT 2 Times

REPEAT 2 Times

Predefined Forms

Code V comes with the following two predefined forms already stored in permanent memory:

- AIAG form
- Preliminary Metals form

The barcode portions of these predefined forms can be printed at either 120 DPI high density or 60 DPI low density. However, the Halfdot Mode reduces print speed by approximately 50% during the printing of the barcode(s).

The predefined forms reside in the printer ROM. If a predefined form is used, it is copied from ROM into the Heap Buffer RAM. The amount of RAM being used ca be determined by executing the IFORM,L command. If the amount of the RAM becomes insufficient for the creation of new forms, the predefined forms can be deleted from RAM similar to user-defined forms using

^IFORM,D or ^IFORM,R.

AIAG Forms

The AIAG (Automotive Industry Action Group) form is a creation of automotive industry that utilizes both barcode and human-readable characters to create a standardized label for shipping and identification. The CVCC contains high density and low density versions of the AIAG form in ROM. The names of these forms are:

- AIAG_H.MT for low density.
- AIAG_F.MT for high density.

The following rules apply when using the AIAG forms:

1. If one of the data fields is not completely filled with data, the remaining characters for that field must end with ^-, ^*, or ^+.
2. If the data field requires both a human-readable field and a barcode field, the same value must be sent for both fields.

Partner Number

The first field (up to nine characters) is treated as human-readable characters. The second field (up to nine characters) is treated as barcode data.

Quantity

The third field (up to six characters) is treated as human-readable characters. The fourth field (up to six characters) is treated as barcode data.

Supplier Quantity

The fifth field (up to nine characters0 is treated as human-readable characters. The sixth field (up to nine characters) is treated as barcode data.

Serial Number

The seventh (up to nine characters) is treated as human readable characters. The eighth field (up to nine characters) is treated as barcode data.

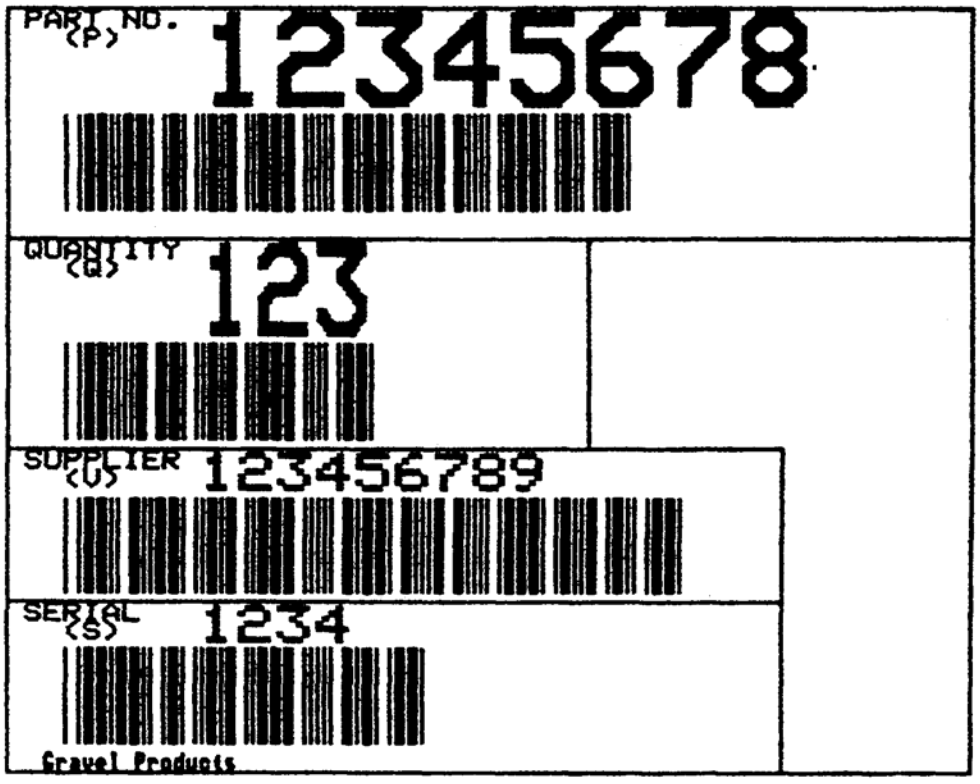
Special Data

The ninth field (up to 55 characters) is treated as human-readable characters.

Example: To print a high density AIAG form, send:

^IFORM,EAIAG_F.MT^G	Form Execution Command 12345678^+ Human-readable part number
1234567^+	Barcoded part number
123^+	Human-readable quantity 123^+ Barcoded quantity
123456789	Human-readable supplier number
123456789	Barcoded supplier number
1234^+	Human-readable serial number
1234^+	Barcoded serial number
Gravel Products^+	Human-readable special characters
^G^-	

which prints:



Primary Metals Form

Similar to the AIAG form, the Primary Metals form utilizes both barcode and human-readable characters to create a standard label for shipping and identification. The barcode portions of this form can be printed in either the High Density or Low Density mode:

The command to select Half Dot Mode is METALS_H.MT.

The command to select Full Dot Mode is METALS_F.MT.

The following rules apply when using the METALS forms:

1. If a data field is not completely filled with data, the data for that field must be terminated with ^-, ^*, or ^+.
2. If the data field requires both a human-readable field and a barcode field, the same value must be sent for both fields.

The data sent to fill the METALS forms will sequentially fill the following data fields:

Product Identification

The first field (up to 15 characters) is printed as human-readable characters. The second field (up to 15 characters) is printed as barcoded data.

Supplier Number

The third field (up to 11 characters) is printed as human-readable characters. The fourth field (up to 11 characters) is printed as barcoded data.

Serial Number

The fifth field (up to 15 characters) is printed as human-readable characters. The sixth field (up to 15 characters) is printed as barcoded data.

Customer Order Number

The seventh field (up to 15 characters) is printed as human-readable characters.

The eighth field (up to 15 characters) is printed as barcoded data.

Heat/Process Number

The ninth field (up to 12 characters) is printed as human-readable characters. The tenth field (up to 12 characters) is printed as barcoded data.

Actual Weight

The eleventh field (up to five characters) is printed as human-readable characters.

The twelfth field (up to five characters) is printed as barcoded data.

Theoretical Weight/Length

The thirteenth field (up to five characters) is printed as human-readable characters.

The fourteenth field (up to five characters) is printed as barcoded data.

Number of Pieces

The fifteenth field (up to five characters) is printed as human-readable characters.

The sixteenth field (up to five characters) is printed as barcoded data.

Size (Physical Description)

The seventeenth field (up to 15 characters) is one of the physical attributes printed as human-readable characters.

The eighteenth field (up to 15 characters) is one of the physical attributes printed as human-readable characters.

The ninteenth field (up to 15 characters) is one of the physical attributes printed as human-readable characters.

Special Data

The twentieth field (up to 14 characters) is printed as a single line of human- readable characters.

The twenty-first field (up to 14 characters) is printed as a single line of human- readable characters.

The twenty-second field (up to 14 characters) is printed as a single line of human-readable characters.

The twenty-third field (up to 14 characters) is printed as a single line of human-readable characters.

The twenty-fourth field (up to 14 characters) is printed as a single line of human-readable characters.

Supplier Address

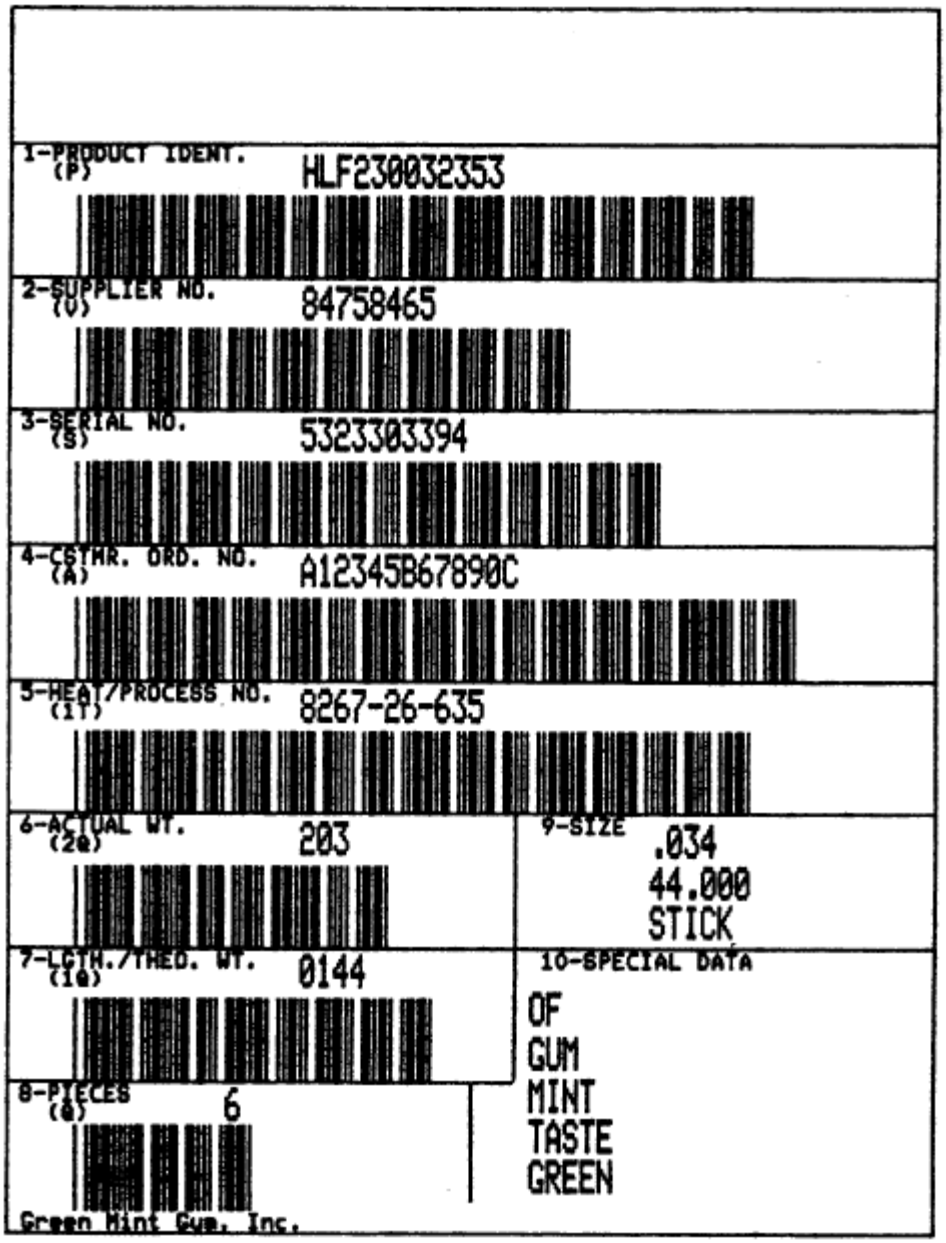
The twenty-fifth field (up to 62 characters) is printed as the supplier's name, city, state, and zip code in a single line of human-readable characters.

Example: Sending the standard high density Primary Metals data:

^IFORM,EMETALS_F.MT^G	Form Execution Command
HLF230032353^+	Human-readable Product Identification
^HLF230032353^+	Barcoded Product Identification
84758465^+	Human-readable Supplier Number
84758465^+	Barcoded Supplier Number
5323303394^+	Human-readable Serial Number
5323303394^+	Barcoded Serial Number
A12345B67890C^+	Human-readable Customer Order
A12345B67890C^+	Barcoded Customer Order
8267-26635^+	Human-readable Heat/Process Number
8267-26635^+	Barcoded Heat/Process Number
203^+	Human-readable Actual Weight
203^+	Barcoded Actual Weight
0144^+	Human-readable Theoretical Weight/Length
0144^+	Barcoded Theoretical Weight/Length
6^+	Human-readable Number of Pieces
6^+	Barcoded Number of Pieces
.034^+	Human-readable Size
44.000^+	Human-readable Size
STICK^+	Human-readable Size
OF^+	Human-readable Special Data
GUM^+	Human-readable Special Data
MINT^+	Human-readable Special Data

TASTE^+	Human-readable Special Data
GREEN^+	Human-readable Special Data
Green Mint Gum, Inc.^+	Human-readable Supplier Address

^G^- prints



Version 1 Buffered Formatting

This section discusses only the Version 1 commands for buffered formatting. Version 2 commands are discussed on page 79.

Buffered Form Create

Version 1 Buffered Form creation and execution differs from Version 2 in that there is no Version 1 Execute command. Instead, all data sent after the Create Command is plugged into the data fields of the created form until a ^G is encountered. The Version 1 Buffered Form Create command is:

Command ^B^-<data>^]<execute data> ^G

Arguments

^B^-	Indicates the beginning of a buffered format sequence.
<data>	GPL commands and data to be stored as a Buffered Form. As with Version 2 Buffered Forms, the Version 1 Buffered Form <data> may contain data fields of the form ^[nnn.
^]	Indicates the end of the buffered format sequence. All characters following the ^] are inserted into the ^[nnn data field until the printer sees ^G.

This command allows only one form to be stored in the forms buffer. For this reason, it does not require a form name.

Control Code Command Changes

The following control codes take on different functions when they are used in the data string that follows the ^] command:

^-	Fills the rest of the current data field with spaces if there are insufficient data characters to fill the current field. The entire data field is converted to spaces if this is the only input in the field.
^*	Fills the rest of the current data field with spaces if there are insufficient data characters to fill the current field. The entire data field is converted to spaces if this is the only input in the field. All remaining fields in the Buffered Form are also converted to spaces unless the fields occur within a Copy command. See “Buffered Form Copy” on page 84 more details.
^,	A form feed behaves the same as a line feed unless it is used within a Copy command. Refer to the Copy command section for more details.

Example 1: In this example, the words Shopping List: are part of the Graphics Pass with the data milk, eggs, and butter, transmitted afterward. Since butter contains the most letters (six characters), the data field will be 6 bytes. The command:

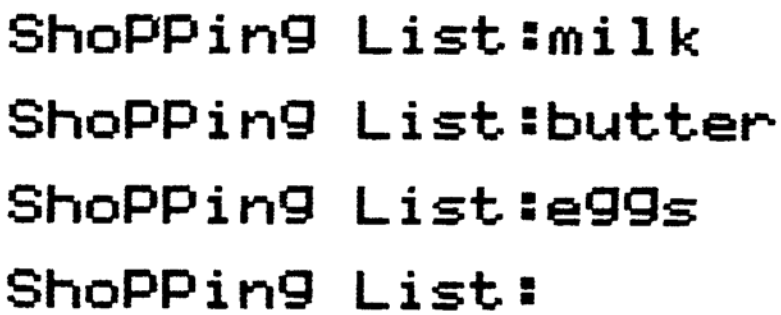
^B^-^M0202000Shopping List:

^[006^-^*^*^]

milk^-

buttereggs^-^-^G

will print:



The word butter, having six characters, completely fills up its data field and doesn't require a ^- to fill out the data field. If a ^- had been used after butter, then a blank field would have been created, as it was at the beginning of the data sequence by the second ^-.

Example 2: The Buffered Repeat command described in the Version 2 Buffered Form section can also be used with Version 1 Buffered Forms as illustrated in this example. The command sequence:

^B^-^R^-^M0202000Shopping List:

^[006^-^*^*^Z^-^]

0001milk^-

0002butter

0003eggs^-

^G^-

will print:

```
ShoPPinG List:milk
ShoPPinG List:butter
ShoPPinG List:butter
ShoPPinG List:eGGs
ShoPPinG List:eGGs
ShoPPinG List:eGGs
```

Example 3: The Buffered Copy command described in the Version 2 Buffered Form section can also be used with Version 1 Buffered forms as illustrated in this example:

^B^-^C02^-^M0202000Shopping List:

^[006^-^*^Z^-^*^]

milk^-milk^-

butterbutter

eggs^-eggs^-

^-^-^G^-

will print:

```
ShoPPinG List:milk
ShoPPinG List:milk
ShoPPinG List:butter
ShoPPinG List:butter
ShoPPinG List:eGGs
ShoPPinG List:eGGs
ShoPPinG List:
ShoPPinG List:
```

7 *Industrial Graphics Application*

Introduction

This Industrial Graphics manual describes how to print several different types of barcodes and block characters using escape sequences and command functions. This chapter contains the following sections:

Control Sequences

This section describes the escape sequences used in entering and exiting the Barcode/Block Character mode, and how to select the various functions (i.e., Normal/Double Density, barcode styles, sizes of block characters, moving the cursor for justification, etc.).

Home and Cursor

This section describes the operational differences between the Home position and the cursor in Block Character mode.

Barcodes

This section provides printed examples of the available barcodes. It also provides related encoding information such as data limits, check digits, etc.

Block Characters

This section provides printed examples of changing the sizes of block characters, and mixing characters with barcodes.

Command Summary

This section provides a summary of the escape sequences and command functions discussed in this manual.

Control Sequences

This section describes the escape sequences for entering and exiting the Industrial Graphics Barcode/Block Character mode. The commands for accessing related functions, such as selecting Normal or Double Density Block Characters, vertical or horizontal orientation of barcodes, line height, and others are also discussed. To properly use the Industrial Graphics option, the printer needs to be in the MT660 emulation.

Print Cycles

While in the Barcode/Block Character mode, print cycles occurs when either:

- the buffer fills.
- the home position is moved via a ^a Command, a Form Feed, Vertical Tab, or VFU command.

- the printer exits from the Barcode/Block Character mode (ESC[<41 where 1 = lower-case L).

Entering Barcode/Block Mode

To enter the Barcode/Block Character mode, send the following escape sequence:

ESC [<4h	ASCII
1B 5B 3C 34 68	hex

Upon each entry into the Barcode/Block Character mode, the following default conditions are established:

- Home and cursor are initialized (see Home and Cursor later in this manual).
- Normal Density printing is selected for both the Barcode Submode and the Block Character Submode.

Once you are in the Barcode/block Character mode, changing a parameter in one Submode also changes that same parameter in the other Submode. For example, selecting Double Density in Block Characters also selects Double Density in Barcode.

Invalid Commands

The following commands are invalid while in Barcode/Block Character Mode:

- Index Forward
- Index Reverse
- Save Cursor Position
- Restore Cursor Vertical Position

Exiting Barcode/Block Character Mode

To exit the Barcode/Block Character Mode, send the following escape sequence:

Command ESC {<41 (lower-case L)	ASCII
1B 5B 3C 34 6C	hex

Control Sequence Formatting

All Control Sequences in the Barcode/Block Character mode have the same basic format:

Command ^Xaa;bb...NL

Argument

^	The Control Sequence Introducer
X	The control sequence function selector. Valid Selectors are A, B, C, D, M, N, R, S, T, U, and W.
aa	First parameter field for selecting specific values using decimal numbers. An unlimited number of digits are allowed. Omitting this field leaves the current value unchanged. Setting it to “0” selects the default value. Plus (+) signs are ignored, but a minus (-) sign preceding a number signifies a negative value that is invalid.
;	Separates parameter fields. This must be present if two or more parameter fields are used or just the second field is used (i.e., ^X ; bb NL). It can be omitted if only the first field will be used (i.e., ^X aa NL).
bb	Second parameter field for selecting specific values using decimal numbers. An unlimited number of digits are allowed. Omitting this field leaves the current value unchanged. Setting it to “0” selects the default value. Plus signs are ignored, but a minus sign preceding a number signifies a negative value that is invalid.

... Additional parameter fields (if applicable).

NL Control sequence terminator. Two categories of terminators are available:

- Passive Terminators - the passive terminators Carriage Return (CR), Line Feed (LF), or CRLF act only as terminators and do not perform any further action.
- Active Terminators - the active terminators Form Feed, Vertical Tab, Reverse Form Feed, and Paper Instruction act as terminators as well as performing their normal functions.

Valid Control Sequences

The following eleven control sequences are valid in Barcode/Block Character Mode:

^A Move the Home.

^B Select Horizontal Barcode.

^C Select Vertical Barcode.

^D Select Double Density Block Characters.

^M Select Normal Density Block Characters.

^N Select New Introducer.

^R Relative Cursor Move.

^S Signed Cursor Move.

^T Absolute Cursor Move.

^U Set Unit Line.

^W Set Barcode Density.

^A - Move The Home

Moves the Home pointer and homes the cursor.

Command ^A(home)NL

Arguments

(home) The distance to move the Home pointer down in increments of 0.01 inches. Default is the current vertical position.

^B - Select Horizontal Barcode

Selects horizontal barcode and sets barcode parameters.

Command ^B(type);(height);(size);(option)NL

Arguments

^B Selects Horizontal Barcode Command.

(type) The style of Barcode (default = 1).

1 = Code 39

2 = Code 39 with Check Digit added. 3-5 = Not used.

6 = Two of five with 2:1 Ratio.

7 = Interleaved Two of Five with 2:1 Ratio. 8-9 = Not used.

10 = Two of Five with 3:1 Ratio.

11 = Interleaved Two of Five with 3:1 Ratio. 12 = UPC Version A, 10 Data Digits

13 = EAN-13

14 = EAN-8

(height) Height of the individual bars in increments of 0.01 inches, ranging from 1 - 255 (default = 80).

(size) Scale factor affecting both height and unit width of the bars, ranging from 1-8 (default = 1).

(option) Human Readable Characters option (default = 1)

1 = No characters

2 = Data Processing font

3 = OCR-A font

4 = OCR-B font

^C - Select Vertical Barcode

Selects Vertical Barcode and sets barcode parameters. Type, height, and size parameters and options are the same as for Horizontal Barcode. Defaults are the same as for horizontal (type = 1, height = 80, size = 1).

Command ^C(type);(height);(size)NL

^D - Select Double Density Block Characters

Selects 120 dots per inch and sets Block Character parameters.

Command ^D(height);(width)NL

Argument

(height) Character Height = 0.12 inches x height multiplier

or

Height Multiplier = Character Height/0.12.

Line feed distance = Unit line x height multiplier, (Unit line = unit line height set by ^U control sequence).

(width) Character width multiplier in the range 2-125 (default = 2). This also affects character cell width. For example:

Character Width = 0.08 inches x Width multiplier

or

Width Multiplier = Character Width/0.08.

Character Cell Width = 0.10 inches x Width multiplier

or

^M - Select Normal Density Block Characters

Selects 60 dots per inch and sets Block Character parameters.

Command ^M(height);(width)NL

Argument

(height) Character Height Multiplier in even numbers in the range 2-124 (default = 2).

(width) Character Width Multiplier in even numbers in the range 2-124 (default = 2).

NOTE: Odd numbers can be selected, but they will default to the next lower even number (5=>4, 7=>6, etc.).

^N - Select New Introducer

Selects a new character as the control sequence introducer where:

Command ^N(?)NL

Argument

(?) Any single graphic character, such as !, @, #, etc. The default character is ^ (caret). To avoid potential problems, the new introducer should not be an alpha-numeric character (not the letters A-Z or the numbers 0-9).

To print the Introducer character as data, send it twice.

For example, ^ ^ prints ^-.

^R - Relative Cursor Move

Moves the cursor relative to its current position in increments equal to the current block character height or width.

Command ^R(xrel);(yrel)NL

Argument

(xrel x W) The distance to move horizontally in increments of 0.01 inches, where W is the current width value. Positive values move the cursor right, and negative values move it left. Total distance must be in the range 0-1320 (default = 0).

(yrel x H) The distance to move vertically in increments of 0.01 inches, where H is the current height value. Positive values move the cursor down, and negative values move it up. Total distance must be in the range 0-9999 (default = 0).

^S - Signed Cursor Move

Moves the cursor relative to its current position in increments of fixed value.

Command ^X(xsign);(ysign)NL

Argument

(xsign) The distance to move horizontally in increments of 0.01 inches. Positive values move the cursor right, and negative values move it left. Total distance must be in the range 0-1320 (default = 0).

(ysign) The distance to move vertically in increments of 0.01 inches. Positive values move the cursor down, and the negative values move it up. Total distance must be in the range 0-9999 (default = 0).

^T - Absolute Cursor Move

Moves the cursor relative to the home position in increments of fixed positive value (negative values are not allowed).

Command ^T(xx);(yy)NL

Argument

(xx) The distance to move horizontally in increments of 0.01 inches. Total distance must be in the range 0- 1320 (default = 0).

(yy) The distance to move vertically in increments of 0.01 inches. Total distance must be in the range 0- 9999 (default = 0).

^U - Set Unit Line

Sets the distance that the paper moves in response to the line feed function. The cursor will move a distance equal to the current line height x the current height multiplier (provided that LF is not used as a Terminator).

Command ^U(x)NL

Arguments

(x) Amount of paper movement.

1, Default = (line height that was current when Barcode/Block Character mode was entered).

2, 1/8 inch line height (8 LPI).

3, 1/6 inch line height (6 LPI).

^W - Set Barcode Density

Sets the print densities and dot-making rules for Barcode printing thereby affecting print speed, symbol length, and readability through white:black and wide:narrow ratios.

For typical applications, the default values produce fully readable Barcode; however, a range of settings is provided to accommodate softer papers or drier ribbons, and to optimize size or throughput for special applications.

Command ^W (unit width);(centers);(dots)NL

Arguments

(unit width) Unit bar or space width in 600th inch increments (400th if dots = 3). Range 9 to 13 default = 11.

(centers) Center-to-center dot spacing along the length of the bars in 600th inch increments. Range 4 to 10, default = 5.

(dots) Number of dots per unit bar or space. Range 1 to 3, default = 1.

When dots = 1, the bar width is one printed dot per Barcode unit width. The bar-to-bar space is one unprinted dot per unit width.

When dots = 2, the number of dots for an N wide bar is 2N1. An N wide space is 2N + 1.

When dots = 3, the number of dots for an N wide bar is 3N1. An N wide space is 3N+1.

Printing speed is directly proportional to the values of unit width and centers, and print speed is half as great for dots = 2 or 3 as for dots = 1. The default print speed is 36

Home and Cursor

The Home position and cursor pointers in this manual are similar in function to the Home and cursor pointers in a CRT terminal:

- Upon entering the Barcode/Block Character Mode, the Home position is set to the extreme top-left dot in the current print line. This is considered the starting point for horizontal and vertical positioning.
- The cursor position is defined as the current print location.

Paper Motion Commands

The printer will lose track of the Top-of-Form position and will start printing in random locations if paper movement exceeds approximately 8 feet during a print job without finding one of the following Paper Motion commands:

Form Feed	Next TOF
Reverse Form Feed	Current TOF
Vertical Tab	Next VFU Channel 2
VFU Command	Next or Previous indicated VFU Channel

These Paper Motion commands perform three functions:

1. The Home is moved to a defined, absolute position.
2. The Cursor position is aligned to the Home.
3. Any printable data being held in the buffer is printed.

Upon entering the Barcode/Block Character Mode, the cursor is aligned to the Home position. A Carriage Return places the cursor at the left margin; a Line Feed places the cursor at the left margin and also moves it down by one line feed distance. Cursor position is not affected when the Carriage Return and/ or Line Feed are used as control sequence terminators. However, the control sequence may affect Cursor position. Use of the Line Feed switch on the printer's control panel will move the cursor forward 1/6 inch steps (in 6 LPI) or in 1/8 inch steps (in 8 LPI) and will align the Home position to the Cursor.

The Cursor also moves horizontally for each block character printed, the distance moved being dependent upon the block character cell width. The cursor does not move when printing barcodes since the length of the Barcode symbol depends upon details of the Barcode type selected. Therefore, when printing barcodes it is necessary to use the Cursor Control Sequences to properly position the cursor.

Upon exiting from the Barcode/Block Character Mode, the cursor will remain pointing to its current vertical position.

Barcodes

Barcodes can be printed either vertically or horizontally from 01 to 2.55 inches high. The Barcode styles can now be defined individually.

Code 39 (Types 1 And 2)

Character Set

0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z - . (space) $ / + %

Data Limits

No limits are imposed by your printer. However, Barcode readers are typically limited to 32 characters.

Check Digits

No limits for Type 1. Type 2 encodes one check digit at the end of the character string.

Emulation Limitations

Your printer does not support the full ASCII character set in this emulation. If a character outside of the specified character set is received, it is encoded as a solid black character. It is possible to encode the character pair representation of all characters in the full ASCII set (see Table 6). For example, if your reader is in the full ASCII Mode, a lower-case "a" may be encoded by using +A (see Table 6, top of column 4). Any of the lower-case letters may be represented by a "+" followed by the corresponding letter.

Table 6. Code 39 Character Print Equivalents, ASCII Code 39

NUL	%U	(space)	(space)	@	%V		%W
SOH	$A	!	/A	A	A	a	+A
STX	$B	“	/B	B	B	b	+B
ETX	$C	#	/C	C	C	c	+C
EOT	$D	$	$	D	D	d	+D
ENQ	$E	%	%	E	E	e	+E
ACK	$F	&	/F	F	F	f	+F
BEL	$G	‘	/G	G	G	g	+G
BS	$H	(	/H	H	H	h	+H
HT	$|	)	/I	I	I	i	+I
LF	$J	*	/J	J	J	j	+J
VT	$K	+	+	K	K	k	+K
FF	$L	,	/L	L	L	l	+L
CR	$M	-	-	M	M	m	+M
SO	$N	.	.	N	N	n	+N
SI	$O	/	/	O	O	o	+O
DLE	$P	0	0	P	P	p	+P
DC1	$Q	1	1	Q	Q	q	+Q
DC2	$R	2	2	R	R	r	+R
DC3	$3	3	3	S	S	s	+S
DC4	$T	4	4	T	T	t	+T
NAK	$U	5	5	U	U	u	+U
SYN	$V	6	6	V	V	v	+V
ETB	$W	7	7	W	W	w	+W
CAN	$X	8	8	X	X	x	+X
EM	$Y	9	9	Y	Y	y	+Y
SUB	$Z	:	/Z	Z	Z	z	+Z
ESC	%A	;	%F	[	%K	{	%P
FS	%B	<	%G	\	%L	|	%Q
GS	%C	=	%H	]	%M	}	%R

Table 6. Code 39 Character Print Equivalents, ASCII Code 39

NUL	%U	(space)	(space)	@	%V		%W
RS	%D	>	%I	^	%N	~	%S
US	%E	?	%J	_	%O	DEL	%T

Code 39 Type 1 Examples

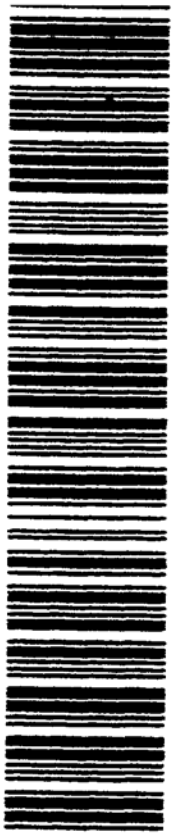
Horizontal with OCR-A characters, ^B1;80;1;3;<CRFL> 12345-67/ 890<CRLF> prints:

Vertical without OCR characters, ^C1;80;1<CRLF>12345-67/890<CRLF> prints:

Code 39 Type 2 Examples

Horizontal with OCR-A characters, ^B2;80;1;3<CRLF> 12345-67/890<CRLF> prints:

Vertical without OCR characters, ^C2;80;1<CRLF> 12345-67/890<CRLF> prints:



Two of Five with 2:1 Ratio (Type 6)

Character Set 0 1 2 3 4 5 6 7 8 9 0

Data Limits No data limits. Invalid characters are encoded as a solid black character.

Examples:

Horizontal with OCR-A Characters, ^B;80;1;3<CRLF> 2598760<CRLF> prints:

Vertical without OCR Characters, ^6;80;1<CRLF>2598760<CRLF> prints:

Interleaved Two of Five, 2:1 (Type 7)

Character Set 0 1 2 3 4 5 6 7 8 9

Data Limits An even number of digits. If an odd number of digits or invalid characters are sent, the last digit and invalid characters will be encoded as a solid black character.

Examples:

Horizontal with OCR-A characters, ^B7;80;1;3<CRLF> 012345678<CRLF> prints:

Vertical without OCr characters, ^C7;80;1<CRLF> 0123456789 <CRLF> prints:

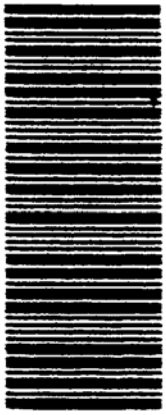
Two of Five with 3:1 Ratio (Type 10)

Character Set 0 1 2 3 4 5 6 7 8 9

Data Limits None. Invalid characters are encoded as a solid black character. Examples:

Horizontal with OCR-A Characters, ^B10;80;1;3<CRLF> 2598760<CRLF> prints:

Vertical without OCR Characters, ^C10;80;1<CRLF> 2598760<CRLF> prints:



Interleaved Two of Five, 3:1 (Type 11)

Character Set 0 1 2 3 4 5 6 7 8 9

Data Limits Any even number of digits. If an odd number of digits or invalid characters are sent, they will be encoded as a solid black character.

Examples:

Horizontal with OCR-A characters, ^B11;80;1;3<CRLF> 012345678<CRLF> prints:

Vertical without OCR characters, ^C11;80;1<CRLF> 0123456789<CRLF> prints:

UPC Version A (Type 12)

Character Set 0 1 2 3 4 5 6 7 8 9

Data Limits Data is limited to 10 digits. Each symbol as exactly 10 data digits preceded by a number system digit followed by a check digit. If 10 characters are sent, the printer supplies zero as the number system check. If 11 digits are sent, the first digit is used as the number system digit. The printer calculates and encodes the check digit. If more than 11 or less than 10 digits or any invalid characters are sent, the number system digit and the invalid characters are encoded as a solid black character.

Examples:

Horizontal with Data Processing Font,

^B12;80;1;2<CRLF> 9876543210<CRLF> prints

Vertical without Data Processing Font, ^C12;80;1<CRLF> 9876543210<CRLF> prints:

EAN-13 (Type 13)

Character Set 0 1 2 3 4 5 6 7 8 9

Data Limits Twelve digits. An EAN-13 symbol is very similar to UPC Version A, but has a thirteenth digit (a number system digit) which is encoded into a parity pattern of the left six digits. The number system digit is used with the twelth digit to define characters representing a specific country code.

If twelve digits are sent, the number system digit will automatically be zero (0). If thirteen digits are sent, the first digit will be used as the number system digit. If more than thirteen or less than twelve digits, or any invalid characters are sent, the number system digit and the invalid characters are encoded as a solid black character.

Examples:

Horizontal with Data Processing Font Characters, ^B13;80;1;2<CRLF> 509876543210<CRLF> prints:

Vertical without Data Processing Font Characters, ^C13;80;1<CRLF> 509876543210<CRLF> prints:

EAN-8 (Type 1)

Character Set 0 1 2 3 4 5 6 7 8 9

Data Limits Eight digits. The EAN-8 symbol encodes two flag digits, five data digits, and the check digit; eight total digits including the check digit. The check digit is calculated by the printer. If more or less than seven digits are sent, the sequence is printed as text rather than as Barcode. Invalid characters are encoded as a solid black character.

Examples:

Horizontal with Data Processing Font Characters, ^B14;80;1;2<CRLF> 654321<CRLF> prints:

Vertical without Data Processing Font Characters,^C14;80;1<CRLF> 6543210<CRLF> prints:

Block Characters

Block characters use the same basic formatting as described in Control Sequences earlier in this manual and can be printed in two different densities:

- Normal Density (^M)

  Characters will print 60 DPI vertically x 72 DPI horizontally.

- Double Density (^D)

  Characters will print 120 DPI vertically x 72 DPI horizontally.

In either Normal or Double Density modes, a Carriage Return/Line Feed acts simply as a command terminator. Their usual functions will not be carried out.

Block Character Sizes

Individual block characters can be printed in the following sizes:

- Height - the height of each block character ranges from 0.24 - 15 inches high.
- Width - the width of each block character ranges from 0.20 to 12.5 inches wide.

Normal Density (^M) Examples

^M2;2<CRLF> BLOCK CHARACTERS <CRLF> prints:

After data is printed and terminated with a <CRLF>, the next current print position is at the bottom right corner of the last character printed. The next data to be printed resets the cursor position to the top right corner before printing. This causes all the block characters to align along their tops.

^M6;6 <CRLF DEF ^M4;4 <CRLF> GHI^M2;2 <CRLF> JKL <CRLF> prints:

To align the baselines of these characters, the cursor position must be manipulated vertically using the Relative Cursor Move (^R) control sequence defined earlier under Control Sequence

^M6;6 <CRLF> DEF ^R0;12 <CR> ^M4;4 <CR> ^R0;-12<CR>GHI^R0;12 <CR> ^M2;2 <CR> ^R0;-12 <CR> JKL<CRLF> prints:

The ^R command defines both the top and eight of each block character.

NOTE: **NOTE:** The control sequence ^R0;XX <CR> moves the cursor up (for negative XX values) or down (for positive XX values) in multiples of 0.01 inches.

When planning the position for the next character, the cursor is always referenced at the top of the next character cell. The total distance to be moved equals a multiple of the current block character height as defined in the ^M parameter.

In this example the value 12 (from ^R0;12) multiplied by the current block character height 6 (from ^M6;6) = 72/100 inch. This is the distance that the cursor moves down from the top left corner of the character cell following the “F” to the baseline.

The value -12 (from ^R0;-12) multiplied by the current block character height 4 (from ^M4;4) -48/100 inch. This moves the cursor to the top of the left corner of the “G” for the beginning of the “GHI” print sequence.

Other Size Variations

^M2;4<CRLF> WIDE<CRLF> prints:

^M4;2<CRLF> TALL <CRLF> prints:

^M2;6 <CRLF> WIDER<CRLF> prints:

^M6;2 <CRLF> TALLER <CRLF> prints:

^M2;4 <CRLF> ^R0;36 <CRLF> WIDE ^M4;2 <CRLF> ^R0;-6 <CRLF> TALL

^M2;6 <CRLF> ^R0;12 <CRLF> WIDER ^M6;2 <CRLF> ^R0;-8 <CRLF>

TALLER<CRLF> prints:

Double Density (^D) Examples

^D2;15 <CRLF> WIDE<CRLF> prints:

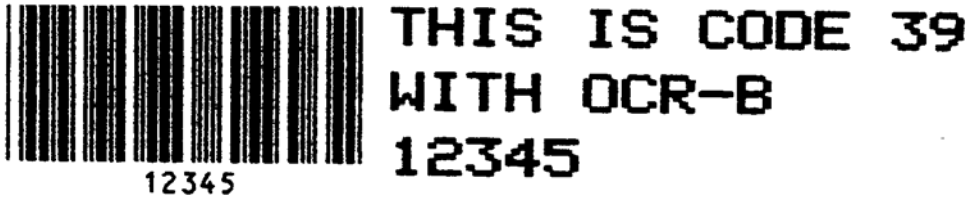
^D15;2 <CRLF> TALL <CRLF> prints:

^D15;15 <CRLF> HUGE <CRLF> prints:

Mixing Barcode/Block Characters

These following examples show the mixing of Barcode and block characters with cursor movements.

^B1;0;0;4 <CR> 12345^D2;2 <CR> ^S125;0 <CR> THIS IS CODE 39<CRLF> ^S125;30 <CR> WITH OCR-B <CRLF> ^S125;20 <CR> 12345 <CRLF> prints:



^M6;6<CR> CODE 39<CR> ^T480;0 <CR> ^B1;60;1;4 <CR> 12345 <CRLF> prints:

^D2;2 <CR> C<CRLF> o <CRLF> D<CRLF> E <CRLF <CRLF> 3 <CRLF> 9<CRLF> ^T50;0 <CR> ^C1;0;0<CR> 123 <CRLF> prints:

Command Summary Table

Command	Summary
ESC[<4h	ENTER Barcode/Block Character Mode
ESC [<41 (lower-case L)	EXIT Barcode/Block Character Mode
^A(home)NL	Move the Home
^B(type);(height);(size);(option) NL	Select Horizontal Barcode
^C(type);(height);(size) NL	Select Vertical Barcode
^D(height);(width) NL	Select Double Density Block Character Mode
^M (height);(width) NL	Select Normal Density Block Character Mode
^N(?) NL	Select New Introducer
^R (xrel);(yrel) NL	Relative Cursor Move
^S (xsign);(ysign) NL	Signed Cursor Move
^T (xx);(yy) NL	Absolute Cursor Move
^U (x) NL	Set Unit Line Height
^W (unit width);(centers);(dots) NL	Set White/Black Barcode Ratio

8 *PGL Menu Operations*

Introduction

The Printronix Graphics Language (PGL) option on your printer is designed to be compatible with the Printronix Intelligent Graphics Processor (IGP) - 100 Version 2.

On your printer some of the parameters you have to set for proper PGL operations are located in Multi-Level menus that can be accessed through the printer control panel.

PGL Menu Parameters

The PGL Graphic parameters can be accessed in the multi-level Configuration Menu. The menu has three different combinations of available parameters depending on the model of the printer. All of the available parameters are represented with applicable graphic option.

See “PGL Graphic Parameters” on page 112.

Graphic Menu Category

This category allows you to configure certain aspects of the Graphic Option on your printer.

Table 7. Configurations Menu Entries for Graphics Options

Menu Entries			Applicable Graphic Option
Menu Level 1 Category	Menu Level 2 Parameter	Menu Level 3 Selections	
Graphic	CVCC Cmd	1-255	Code V only
	Smooth	1-99	Code V and PGL
	PY Then	None All Term	Code V only
	PN Then	All None	Code V only
	Dark Bar	OFF ON	Code V and PGL
	Version	2 1	Code V only
	Descndr	Fixed Auto	Code V only
	Vscale	OFF ON	Code V and PGL
	Zero	Open Slashed	Code V and PGL
	SFCC	1-255	PGL only

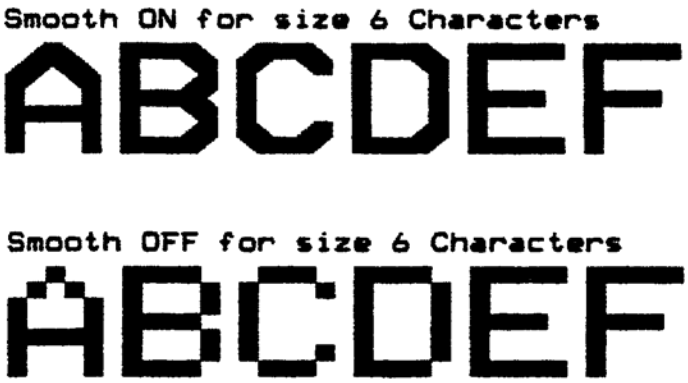
PGL Graphic Parameters

This section explains the Graphic Category parameters that applies to the PGL Graphic Option on your printer.

Smooth Parameter

The Smooth Parameter affects the block characters in PGL. Smoothing causes the stair step appearance of larger block characters to be less pronounced. This parameter dictates the character size at which smoothing takes place. The default setting for Smooth is 3.

Example:



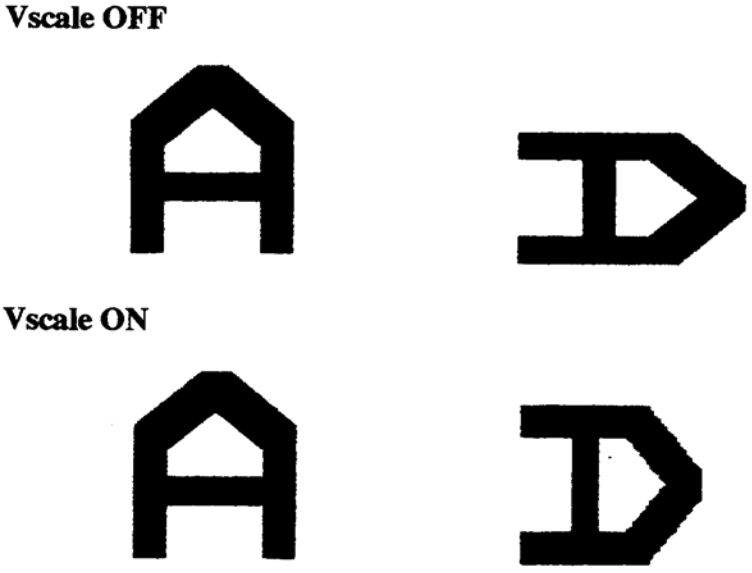
Darkbar Parameter

For barcodes to print at a heavier than normal dot density, use this parameter and set up the printer to output all barcodes at 120 DPI instead of 60 DPI. See “Half-Dot Mode (Double Density)” on page 29 for a sample of dark printing. The default for this parameter is OFF.

Vscale Parameter

Due to the difference between horizontal and vertical DPI, as the size of block characters increase, there is a greater disparity between the physical size of horizontal and vertical output. This parameter allows you to print vertical block characters larger than size 3 at approximately the same scale as horizontal block characters.

Examples: Size 10 Characters



Zero Parameter

The Zero Parameter allows you to set all zeros to print as slashed zeros (Ø). This makes it easier to distinguish zeros from the upper-case letter O. The default for the Zero Parameter is Slashed.

SFCC Parameter

This parameter gives you another method of changing the Special Function Control Character. The default is the ASCII tilde (~, HEX 7E, Decimal 126) character.

9 PGL Command Set

PGL Command Set Standards

To use the PGL commands, follow a set of standards in constructing the commands you send to your printer.

Special Function Control Character (SFCC)

The SFCC identifies Special Function Commands that are used in NORMAL Mode and Execute Mode. The character you use for the SFCC is based on the host computer interface requirements of your system. The SFCC must be a printable character and it must be placed at the beginning of a new line before the command or data. The default SFCC is ~ (tilde). This character may be changed using the SFCC Change Command or by accessing the Configuration menu through the printer control panel.

Since the SFCC can be changed to any printable character and may be different due to your host computer requirements we will use the ♦ symbol to represent the SFCC in all command sequences.

Parameter Delimiter

Each parameter on the command line must be separated from other parameters by a delimiter. The delimiter recognized by your printer is the semicolon (;).

NOTE: Do not place blank spaces between the delimiter and the next parameter. A missing or misplaced delimiter causes an error.

Case

The PGL language requires that all commands be entered in upper-case characters.

Line Terminator

In PGL each command line must be terminated by a Line Feed, Carriage Return with a Line Feed, or a Form Feed command. If you do not terminate the command line properly, the command will not be accepted by your printer. Refer to your system documentation to determine how to perform a terminator function (Line Feed, Carriage Return with Line Feed, or a Form Feed command).

Printable Data Delimiter

All printable data (excluding overlay data), alphanumeric characters, and barcode data must be enclosed by a printable data delimiter. This delimiter is always a printable character that the printer recognizes, telling it to print the data that follows until it encounters the same character again. Any printable character may be used except the SFCC or the slash (/). The same printable character must be used at both ends of the printable data. The position of the printable data delimiter in the command format and in any examples in this manual will be indicated by a. ‡

NOTE: Only characters that do not appear in the printable data field can be used as delimiters.

Spaces

Do not use spaces between the command parameters unless specifically required.

Command Parameters

The majority of the PGL commands have multiple command parameters. Some parameters are required and some are optional. There are several selections you may enter for some of the required parameters (for example, SFCC). These parameters must be separated by a semicolon.

Form and Logo Names

Names identifying forms or logos may be no longer than eight characters. The document name is used in the EXECUTE Form Mode to identify the form or in CREATE Form Mode to identify the logo to be used for printing.

NOTE: Do not use spaces in the document name.

The following characters may be used in document names.

Upper and lower-case A-Z (HEX 41-5A and 61-7A).

- Number 0-9 (HEX 30-39)
- Dollar Sign ($) (HEX 24)
- Percent Sign (%) (HEX 25)
- Dash (-) (HEX 2D)
- At Sign (@) (HEX 40)
- Braces { } (HEX 7B and 7D)
- Parenthesis ( ) (HEX 28 and 29)
- Tilde (~) (HEX 7E)
- Single Quotes (‘ ‘) (HEX 60 and 27)
- Exclamation Point (!) (HEX 21)
- Pound Sign (#) (HEX 23)
- Amersand (&) (HEX 26)

The SFCC can also be used as part of a form or logo name.

Numeric Variables

Some command parameters may include numeric variables. When it is required, an appropriate numeric value must be entered. In optional parameters where a numeric variable appears, the numeric variable needs to be included only if the option is selected.

Comments

Comments that aid in preparation or maintenance of a form or logo may be added to command lines that do not contain the SFCC. They must be preceded by a slash (/).

Print Position Location

PGL provides a method of placing the print position at any location on a page by selecting a character column or row, and a dot position within the specified (starting or ending print location) character cell. In commands that initiate printing of alphanumeric characters, barcodes, etc., there are optional methods you may use to select a vertical character column and dot position, and a horizontal character row and dot position.

When selecting a vertical character column and dot position, the format is the same as for selecting horizontal character row and dot position. The row or column is defined first followed by the dot position. The only difference between the two is the range of allowable dot positions.

Each character cell is 6 dot columns wide (0-5) by 12 dot rows high (0-11). The format for specifying character cell (in a row or column) and dot position (within the specified character cell) is as follows:

XX, YY

where:

XX = Character Column or Row

YY = Vertical or Horizontal Dot Position

NOTE: **NOTE:** A column and row position (two XX.YY selections) must be specified to locate the specific character cell where printer output will begin and/or end.

The decimal point must be present when specifying dot positions within a cell.

Within the command structure the starting row is defined first followed by the starting column.

This same format can be used to specify the distance between duplications of a specific element on a form horizontally and vertically. In this case the row and dot position indicate how far to move to the next print position.

(See “Horizontal Duplication Command” on page 128 and “Vertical Duplication Command” on page 129 for examples).

Data Types

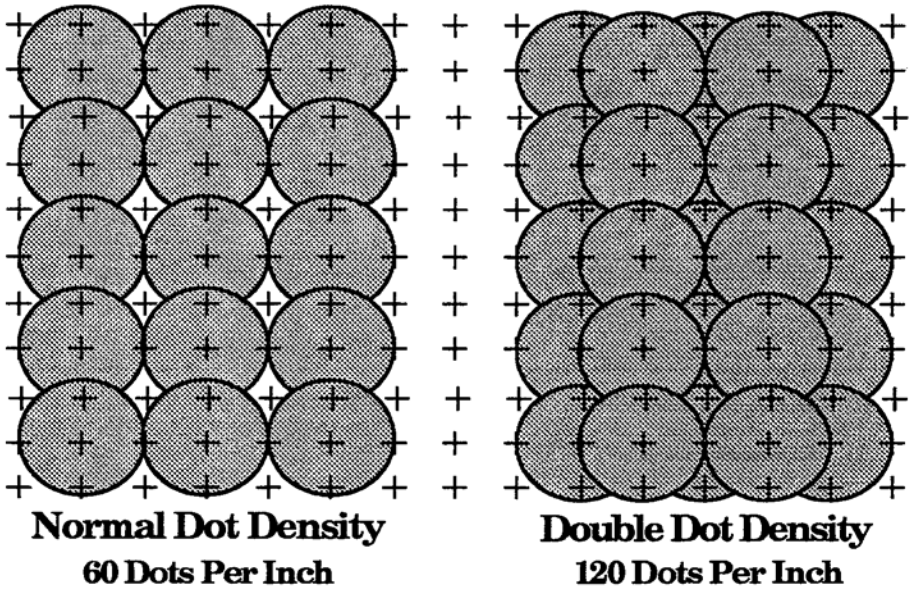
There are four data types used in PGL:

- Fixed (template/form description)
- Overlay (interactive)
- Dynamic (mailing lists)
- Incremental (automatically updated information such as check numbers, order numbers, etc.).

Double-Density Printing (DARK Parameter)

The DARK parameter is available in the Alpha, Reverse, and Barcode commands to produce darker text, backgrounds, and barcodes. When the parameter is enabled, horizontal DPI increases from 60 DPI to 120 DPI. This fills in the holes between multiple dot-columns causing darker output.

The following figure shows how Double Dot Density provides a darker print. This improves the readability of characters and barcodes, and darkness of background.



PGL Modes

Four basic modes of operation for the PGL are available: NORMAL, LOGO, CREATE, and EXECUTE.

NORMAL

In NORMAL mode, the printer functions in line printer mode. The incoming data is scanned by the printer for SFCC commands. When the printer encounters a properly formatted SFCC Command it executes the specified command.

CREATE

The CREATE mode allows you to create forms for use in EXECUTE mode. These forms can include logos, borders, and static text.

EXECUTE

The EXECUTE mode allows you to use the forms you have created.

LOGO

The LOGO mode allows you to create graphical images for use in forms.

PGL Commands

This section discusses the commands that can be used in PGL mode. Many of the commands may be used in multiple modes and in some instances there are slight differences in the construction of the command in different modes. When one of these commands is presented, the different modes used be noted. Table 8 lists all of the commands and the modes in which they can be used.

Table 8. Supported PGL Commands

PGL Command	NORMAL Mode	CREATE Mode	EXECUTE Mode
AFn			X
ALPHA		X	
BARCODE		X	
BFn			X
BOX		X	
CORNER		X	
CREATE	X		
DELETE FORM	X		
DELETE LOGO	X		
DENSITY	X		X
DIRECTORY	X		
END		X	
EXECUTE	X		

Table 8. Supported PGL Commands

PGL Command	NORMAL Mode	CREATE Mode	EXECUTE Mode
EXPAND	X		X
HDUP		X	
HORZ		X	
IAFn			X
IBFn			X
IGOFF	X	X	X
IGON	X	X	X
ISET	X	X	X
LFORM		X	
LISTEN	X		
LOGO (Call)		X	
LOGO (Create)	X		
LPI	X		X
NORMAL	X		X
PAGE		X	
QUIET	X		
RESET	X		X
REVERSE		X	
SCALE		X	
SFCC	X		X
SFOFF	X	X	X
SFON	X	X	X
USET	X		
VDUP		X	
VERT		X	

Command Presentation

Each command is presented in a table (see Figure 3) showing the command in its basic form followed by a breakdown of the command into its required and optional parameters. Required parameters are unshaded and optional parameters are shaded. All parameters must be entered in the sequence given.

Table Key

In the command and parameter tables (see Figure 3), there are a few conventions to follow to help interpret the command structure.

- ♦ = Special Function Control Character (SFCC).
- ‡ = Alphanumeric text string delimiter.
- ; = Parameter delimiter.

The tables show the individual commands, their basic construction, and valid selections where required. In addition, each command is marked as to which PGL Mode(s), NORMAL, CREATE, and EXECUTE, they can be used in.

Some commands consist of several command lines. Some command lines are long, resulting in some of the lines wrapped into two lines. When this happens the wrapped section of a command line is indented on the second line as shown below.

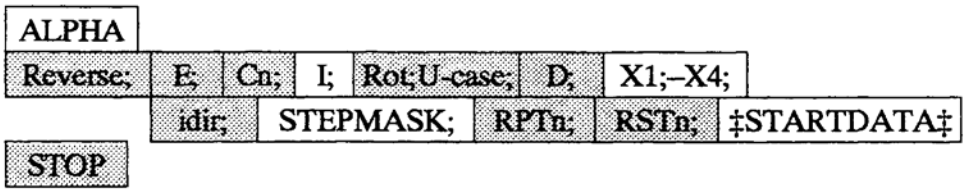


Figure 3. Table Key

Alphanumeric String Creation Commands (Create)

The command table in Figure 4 lists, describes, and explains the use of all parameters that may be used in creating alphanumeric character strings, and specifies their location on the form you are creating. You may creat static/ dynamic, incremental, and dynamic incremental alphanumeric strings using variations of this command. Table 9 shows the results of Incremental Alphanumeric String Commands.

In Dynamic Alphanumeric Creation a window is created on the form in which a Dynamic Alphanumeric String will later be entered in the EXECUTE mode.

The window is assigned a number (AFn) that will be used in EXECUTE mode to identify in which window the string will print.

The Static/Dynamic Alphanumeric Create command is:

Figure 4. Static/Dynamic Alphanumeric Create in Table Format

Table 9. Static/Dynamic Alphanumeric Creation Command

Field	Function	Option/Modifier Selections
ALPHA	Specifies Alphanumeric Data String creation command.	Enter ALPHA.
Reverse: (Printing)	Optional parameter used to enable reverse printing.	Enter: R; - for reverse printing only. RD; - for reverse/dense background RDL; - for reverse/dense/long background RL; - for reverse/long background L = long background for characters with descenders in dynamic alphanumeric printing. Not required for static alphanumeric printing. The reverse field length is adjusted for characters with descenders automatically in static alphanumeric data.
E; Elongated Printing	Elongated Character option. Selects double height and single width character output.	Enter E; If this option is selected, the X3 and X4 Parameters must be set to 0.
Cn; CPI and Type Style Selection	Optional parameter for selecting Characters Per Inch and OCR-A and OCR-B type styles.	Enter C followed by one of the values below: 10, 12, 13, 15, and 17 CPI. 10A for 10 CPI and OCR-A type style 10B for 10 CPI and OCR-B type style If this optional parameter is not used, then the X3 and X4 parameters must be set to 0.
AFn;L; Dynamic data to be entered in the EXECUTE Form Mode.	Optional field that identifies location and the length of a dynamic alphanumeric string. If this option is enabled do not enter static text data in the optional field following the X4 parameter.	Enter AF in the command string followed by a value that identifies the dynamic string. L = number of characters in the dynamic alphanumeric string. Valid range is 1 - 280. The length of the dynamic string must be less than or equal to the value signified by the L parameter.

Table 9. Static/Dynamic Alphanumeric Creation Command

Field	Function	Option/Modifier Selections
Rot; U-case	Designates print rotation and whether the data will be printed in mixed or all upper-case characters. CPI selections other than the default value of 10 CPI are not allowed in rotated output. Inserted desired rotation first, then designate upper-case printing if desired.	After this option is selected you have three rotations to select from and you can opt for all characters to be upper-case. Enter: CW; = 90 degrees clockwise rotation. CCW; = 90 degrees counterclockwise rotation. INV; = 180 degrees rotation (upside down ouput). Add: UC; = upper-case character output. All lower-case alpha characters will be converted to upper-case.
D; Dark Printing	Optional parameter for changing horizontal dot densit from 60 DPI to 120 DPI.	Enter D or DARK to select this option.
Parameters X1 and X2 define the starting point of printer output. Whether in dots or characters depends on the SCALE command. You may also use the XX.YY format for plotting start print locations.		
X1;	Specifies the starting row of the printer output.	X1 has a valid range of 1 to the current form length.
X2;	Specifies the starting column of the printer output.	X2 has a valid range from 1 to the current form width.
Parameters X3 and X4 control character expansion. The valid range of selections for both is from 0 to 113. Both must be set to zero if the Elongated parameter is used or any CPI other than the default is specified. These parameters have no effect on OCR and Compressed characters. For selections $\geq$ 2, a block character style will be used for alphanumeric characters. See NOTE below.		
X3;	Specifies vertical expansion of the printer output.	Enter a value from 0 to 113. Zero (0) specifies the standard size.
X4;	Specifies horizontal expansion of the printer output.	Enter a value from 0 to 113. Zero (0) specifies the standard size.

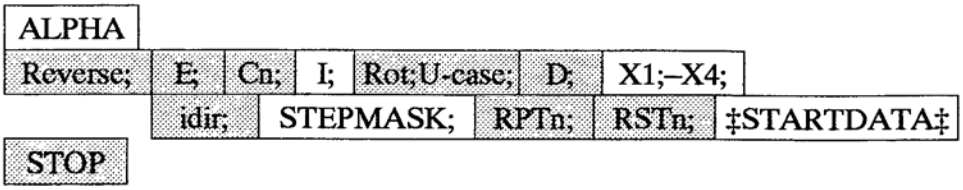
Table 9. Static/Dynamic Alphanumeric Creation Command

Field	Function	Option/Modifier Selections
‡text‡	Static alphanumeric data that will print at the position specified by X1 and X2. If you have specified a dynamic character string (AFn or IAFn) in this command do not use this parameter.	This alphanumeric string must be set off by the Character String Delimiter (‡). Any standard ASCII printable character may be used except for the Character String Delimiter character. To change this string, you must redefine the form.
STOP	**Optional parameter that indicates the end of the Alphanumeric command and leaving the Create mode.**	Enter STOP. If you do not enter STOP, the printer will wait for another set of Alphanumeric command parameters.

NOTE: See Appendix B, page 217, for the approximate size of available expansion factor for block characters (both horizontal and vertical print orientation).

Incremented Static (Fixed) Alphanumeric Strings

To create a fixed incremental alphanumeric string use the following basic command:



This command is identical to the Static/Dynamic Alphanumeric string command except for the following added parameters.

Table 10. Fixed Incremental Alphanumeric String Command

Field	Function	Option/Modifier Selections
I;	Specifies that this command is an incremental alphanumeric command.	Enter I;
idir;	Optional field specifying whether the data field will be incremented or decremented.	Enter a +; or leave this field blank to increment in a positive direction. Enter a -; to decrement the data.

Table 10. Fixed Incremental Alphanumeric String Command

Field	Function	Option/Modifier Selections
STEPMASK;	This parameter is the increment information field that defines the number of character positions in the data field, the increment amount for each position, and which positions will be incremented.	See the explanation following this table for how to construct the increment information field.
RPTn;	Optional parameter that specifies the number of times a field value is repeated before it is incremented.	Enter RPT followed by a numeric value from 1 to 65,535. The default repeat amount is 1, which means that if you do not enter this parameter or enter only 1, then the field will be incremented each time it prints.
RSTn;	Optional parameter that specifies the number of times an incremented field will print before it is reset to the starting point.	Enter RST followed by a numeric value from 1 to 65,535. The default reset value is 0, which means that the field will not be reset.
‡STARTDATA‡	This parameter defines the starting incremental field.	See the explanation following this table on how to construct the increment information field.

STEPMASK (Increment Information Field)

The STEPMASK parameter defines the size, amount of increment and which positions in the field will be incremented. Valid entries for the STEPMASK field are 0-9, L, and any other character.

0-9

These values dictate the amount of the increment for a selected position in the incremented field.

L

This character indicates that the selected position will not be incremented but the positions on the right and left of this character remain linked and will increment the specified amount.

Any Other Character

These characters indicate that the selected field position will not be incremented and that any position to the left and right of this character are not linked. Fields to the left and right of this character can only be incremented independently.

There is one exception to these rules. If the first character after the semicolon preceding the STEPMASK parameter is a minus sign (-) or a plus (+) sign, the printer does not include it as part of the STEPMASK. It translates it as an increment direction command. After the first minus sign additional minus signs will be treated as any other character.

STARTDATA (Incremented Field)

The STARTDATA parameter defines the starting incremental field. This field can be no larger than the STEPMASK field. It can, however, be smaller than the STEPMASK field. If it is smaller, the printer will right justify the field and insert enough spaces to equalize field sizes. Valid entries for each position in this field depend on the increment information in the STEPMASK.

If STEPMASK = 0-9, Valid STARTDATA is:

- A-Z (upper and lower-case)
- 0-9
- Space character

If the STEPMASK is not 0-9, then any character may be used for STARTDATA.

The following examples show various ways the STEPMASK may be set up and the result each setup has on a field of data.

Example Key:

SM = STEPMASK

SD = STARTDATA

RPT = SD Repeat Count

RST = SD Reset Count

Example 1: Incremental, linked alphanumeric Subfields: Default RPT (1) and RST (0)

ALPHA		Printer Output
I;1;1;0;0;00001;‡ABC01‡		ABC01
SM = 00001	RPT = 1	ABC02
SD = ABC01	RST = 0	ABC03
		...
		ABC09
		ABC10
		...
		ABC99
		ABD00
		...
		ZZZ99
		AAA00

Example 2: Non-incremental position with nonlinked subfields: Default RPT (1) and RST (0)

ALPHA		Printer Output
I;1;1;0;0;0001X0;‡ABC0-1‡		ABC0-1
SM = 0001X0	RPT = 1	ABC1-1
SD = ABC0-1	RST = 0	ABC2-1
		...
		...
		ABC9-1
		ABD0-1
		...
		...
		ZZZ9-1
		AAA0-1

Example 3: Non-incremental subfield with Linked subfields, Default RPT (1) and RST (999)

ALPHA		Printer Output
I;1;1;0;0;00LL1;RST999‡00001‡		00001
SM = 00LL1	RPT = 1	00002
SD = 00001	RST = 999	00003
		...
		...
		00009
		01000
		...
		...
		99009
		00001

In example 3 above, if the subfield was 0000A (alphabetic), the subfield to the left of the nonincremental linking subfield would increment alphabetically.

0000A

0000B

0000C

...

... 0000Z

0A00A

...

...

ZZ00Z 00000A

Example 4: Single STARTDATA position, Default RPT (1) and RST (0)

ALPHA		Printer Output
I;1;1;0;0;00001;‡1‡		----1
SM = 00001	RPT = 1	----2
SD = 1	RST = 0	----3
		...
Notice that the printer		...
automatically inserts the proper		----9
number of spaces (-) to make the		---10
STARTDATA field equal in size		...
to the STEPMASK.		...
		99999
		00000

Example 5: Non-incremental subfield with incremented linked subfields, Default RPT (1) and RST (5000)

ALPHA		Printer Output
I;1;1;0;0;XXX000L1;‡RST5000;ACME000-1‡		ACME000-1
SM = XXX000L1	RPT = 1	ACME000-2
SD = ACME000-1	RST = 5000	ACME000-3
		...
		...
		ACME000-9
		ACME001-0
		...
		...
		ACME499-9
		ACME500-0
		ACME000-1

Example 6: Linked Alphanumeric subfields with a Decrementing subfield and Non-incrementing subfield: Default RPT (2) and RST (0)

ALPHA		Printer Output
I;1;1;0;0;XXX01;RPT2‡TOP99‡		TOP99
SM = –XXX01	RPT = 2	TOP99
SD = TOP99	RST = 0	TOP98
		TOP98
		TOP97
		TOP97
		....
		TOP55
		TOP55
		TOP54
		TOP54
		....
		...
		TOP02
		TOP02
		TOP01`
		TOP01
		TOP99

Dynamic Incremented Alphanumeric Strings

This command creates a window on the form in which a Dynamic Incremented Alphanumeric String will later be entered in the EXECUTE mode. The window is assigned a number (IAFn) that will be used in EXECUTE mode to identify in which window the string will print. To create a window on a form, use the following basic command when creating a form:

ALPHA						
Reverse;	E;	Cn;	IAFn;L;	Rot;U-case;	D;	X1;–X4
STOP						

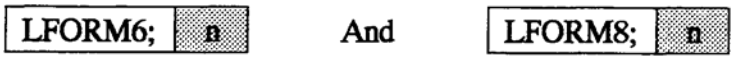
This command is identical to the Static/Dynamic Alphanumeric string command except for the following parameters.

Table 11. Dynamic Incremental Alphanumeric String Command

Field	Function	Option/Modifier Selections
IAFn;	Defines the field as an incremental alphanumeric string and assigns it an identification number that will be used to call it up for use in EXECUTE mode.	Enter IAF to signify that the field is incremental dynamic alphanumeric data, then replace n with the identification number (1-255).
L;	Specifies the maximum number of characters in the dynamic alphanumeric field.	Enter a value from 1-280.

Form Length Commands (Create)

When you originally set up your form, one of the optional parameters sets the form length in dot rows. Using the following commands you can change the Form Length to a specific number of lines:



NOTE: Each command specifies an LPI as well as a Form Length. However, these commands do not change the LPI. They simply specify a form length in lines measured in different dot row heights. The LFORM6 command sets form length in lines measured in 12 dot rows and LFORM8 sets from length in lines measured in 9 dot rows.

Valid Ranges:

1 - 5461 lines for 6 LPI

1-7281 lines for 8 LPI

NOTE: Make sure to send the proper command for the LPI. If you send the LFORMS command when the printer is set for 6 LPI, the results are not desirable.

Duplication Commands (Create)

There are two commands that can be used to duplicate elements on a form: HDUP (Horizontal Duplication) and VDUP (Vertical Duplication). Each specifies the number of times the element will be printed and the spacing between each duplication. These commands function with any element on a form except Overlay Data.

Horizontal Duplication Command

This command allows you to duplicate specific elements horizontally on a form, such as lines, logos, barcodes, etc. The basic command is:

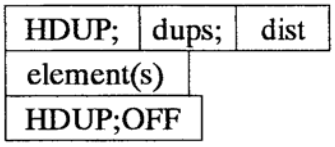


Table 12. Horizontal Duplication Command

Field	Function	Option/Modifier Selections
HDUP;	Command Specifier	Enter **HDUP;**
dups;	Specifies number of duplications.	Enter the desired number of duplications. The valid range of values is 1-255. Make sure there is enough room on the form for the duplication(s) to print.
dist	Specifies the distance from the starting point of one element to the next duplication. This parameter may specify columns or dots or a combination of columns and dot positions.	Enter a column or dot distance or a combination of columns and dots. Whether in columns or dots depends on the **SCALE** Command. You may also use the **XX.YY** format for plotting duplication print locations explained earlier in the **Standards Section**.
element	Defines the element that will be duplicated. You may define a line, call a logo or a barcode, etc.	Enter print information in proper format.
HDUP;OFF	Termines Horizontal duplication mode.	Enter HDUP;OFF

NOTE: You may embed Horizontal Duplication commands within Vertical Duplication commands, but you cannot embed Horizontal Duplication commands within Horizontal Duplication commands.

Vertical Duplication Command

This command allows you to duplicate specific elements vertically on a form, such as lines, logos, barcodes, etc. The basic command is:

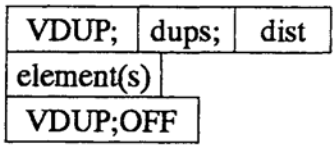


Table 13. Vertical Duplication Command

Field	Function	Option/Modifier Selections
VDUP;	Command Specifier	Enter **VDUP;**
dups;	Specifies number of duplications.	Enter the desired number of duplications. The valid range of values is 1-255. Make sure there is enough room on the form for the duplication(s) to print.

Table 13. Vertical Duplication Command

Field	Function	Option/Modifier Selections
dist	Specifies the distance from the starting point of one element to the next duplication. This parameter may specify columns or dots or a combination of columns and dot positions.	Enter a column or dot distance or a combination of columns and dots. Whether in columns or dots depends on the **SCALE** Command. You may also use the **XX.YY** format for plotting duplication print locations explained earlier in the **Standards Section**.
element	Defines the element that will be duplicated. You may define a line, call a logo or a barcode, etc.	Enter print information in proper format.
VDUP;OFF	Termines Vertical duplication mode.	Enter VDUP;OFF

NOTE: You may embed Horizontal Duplication commands within Vertical Duplication commands, but you cannot embed Vertical Duplication commands within Vertical Duplication commands.

Using Duplication Commands

Both the duplication commands use the same basic format. For these examples we will use the HDUP and VDUP commands to duplicate the BOMB Logo, which we create in a later section, on a form. The following program would print the bomb logo four times horizontally on a form:

Example 1: Horizontal Duplication

```
~CREATE;BOMB HDUP;4;10 LOGO 10;10;BOMB STOP HDUP;OFF
END
~EXECUTE;BOMB
~NORMAL
```

Printer Output:

Example 2: Vertical Duplication

The following program would print the bomb logo four times vertically.

```
~CREATE;BOMB VDUP;4;10 LOGO 10;10;BOMB STOP VDUP;OFF
END
~EXECUTE;BOMB
~NORMAL
```

Printer Output:

Page Number Command (Create)

This command allows you to place an automatically incremented page number on each page in a specific location. The format for this command is:

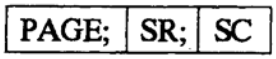


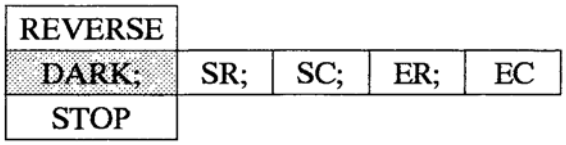
Table 14. Page Number Command

Field	Function	Option/Modifier Selections
PAGE;	Command Specifier	Enter **PAGE**
SR and SC parameters define the starting location of the page number. Whether in columns or dots depends on the **SCALE** Command. You may also use the **XX.YY** format for plotting duplication print locations explained earlier in the **Standards Section**.		
SR;	Plots the starting row of the page number.	Enter a value ranging from 1 to one less than the maximum form length.
SC	Plots the starting column of the page number.	Enter a value ranging from 1 to one less than the maximum form width.

The printer automatically increments the page numbers starting from the page number specified in the EXECUTE command. If a page number is not specified in EXECUTE command, no page number will print.

Reverse Print Command (Create)

This command allows you to define an area on a form where reverse printing (white ouput on black background) will be output.



Reverse printing cannot be used with OCR Characters or Overlay Data.

Table 15. Reverse Command

Field	Function	Option/Modifier Selections
REVERSE	This is the command specifier. Using the following parameters a field of reverse printing will be defined for a location on a form.	Enter REVERSE
D; Dark Printing	Optional parameter for changing horizontal dot density from 60 DPI to 120 DPI.	Enter **D;** or **DARK;** to select this option.
The SR, SC, ER, and EC parameters use the same methods for defining the REVERSE Print area. Whether in columns or dots depends on the **SCALE** Command. You may also use the **XX.YY** format for plotting duplication print locations explained earlier in the **Standards Section**.		
SR;	Plots the starting ROW of the field.	Enter a value ranging from 1 to one less than the maximum form length and the delimiter.
SC;	Plots the starting COLUMN of the field.	Enter a value ranging from 1 to one less than the maximum form width and the delimiter.
ER;	Plots the ending ROW of the field. The ending row must be greater than the starting row.	Enter a range ranging from 2 to one less than the maximum form length and the delimiter.
EC	Plots the ending COLUMN of the field. The ending column must be greater than the starting column.	Enter a value ranging from 2 to the maximum form width.
STOP	Command Terminator.	Enter STOP and the printer waits for a new command. Leave it out and the printer will wait for another REVERSE Command.

Exit CREATE Mode (Create)

To exit the CREATE mode after creating your image, enter the following command:

```
END
```

When the printer receives this command it stores the work you have done and returns to NORMAL Mode.

EXECUTE Mode (Normal)

To use the commands listed above to execute printing of a form you must first enter EXECUTE Mode. To assist in composing the various parameters that can be used with the EXECUTE Form Command, the basic command is provided again.

The basic command to enter EXECUTE Mode is:

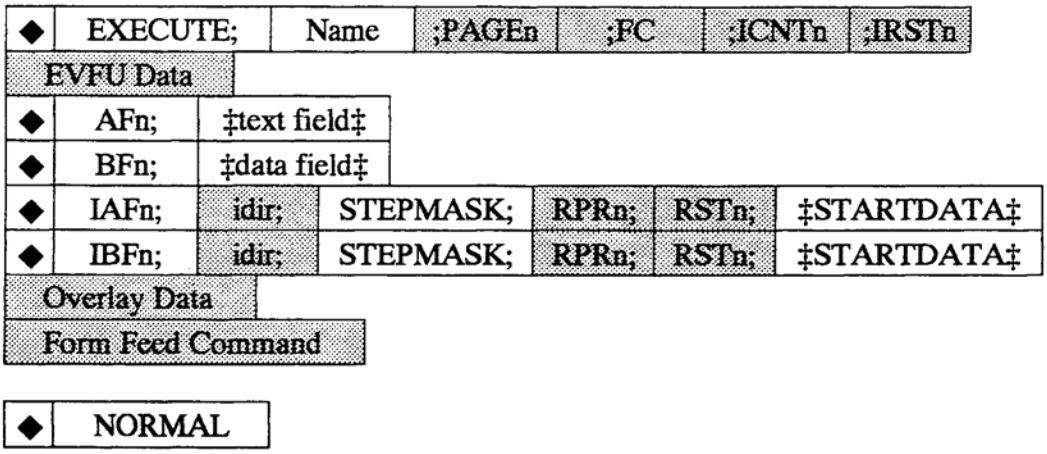


Table 16. EXECUTE Form Command

Field	Function	Option/Modifier Selections
+EXECUTE;	Command Specifier. Instructs printer to enter EXECUTE Mode.	Enter the current Special Function Control Character followed by **EXECUTE;**
Name;	Specifies Form to print.	Enter form name exactly as assigned when it was created.
PAGEn;	Optional parameter that sets the beginning page number for an incremental page number. If a page number location has not been specified in the CREATE Form Mode, the page number will print in the upper-left corner of the form.	Enter PAGE followed by a decimal number ranging from 0 - 99999999, then the parameter delimiter. Upon reaching the 99999999th page, the page number resets to 0. If no page number is specified no page number will print, even if a space was designated in the Form Definition.
FC	Optional parameter that specifies the number of forms to print. This parameter cannot be used if dynamic incremental, overlay, or EVFU data will be used.	Enter the desired number of forms. You can print a total of 4,294,967,296 forms.

After the EXECUTE command is sent, several different types of data can be defined and printed using the following parameters and SFCC commands.

Table 17. EXECUTE Form Command Optional Parameters

Field	Function	Option/Modifier Selections
The following parameters are used in forms where Dynamic Incremental Data is included.		
ICNTn;	Optional parameter that specifies the number of forms that will be printed with incremental data fields.	Enter ICNT followed by a value ranging from 1 - 65,535 then the parameter delimiter.
IRSTn;	Optional parameter that specifies how many forms will print before the incremental data fields will be reset to their original values.	Enter IRST followed by a value ranging from 1 - 65,535 then the parameter delimiter. This value must be less than or equal to the ICNT amount or the incremental data fields will not reset.
The following information represents an EVFU Form Download sequence.		
EVFU Data		Enter the Form Definition Sequence. Refer to your Operator/Application Manual for information on composing the download sequence.
The following parameters are used informs where Dynamic Alphanumeric Data Fields and Barcodes have been defined. You may enter as many of the following fields as were originally defined on the form.		
♦ AFn;	Optional parameter that specifies one of the previously defined Dynamic Alphanumeric Data fields.	Enter the Special Function Control Character followed by **AF**, then the assigned number of the desired Alphanumeric Data Field. End the entry with the parameter delimiter.
‡text field‡	The ASCII character string that will be printed in the specified location.	Enter the data parameter delimiter followed by the alphanumeric string and end the entry with the data parameter delimiter. The delimiter cannot be used in the alphanumeric string.
♦ BFn;	Optional parameter that specifies one of the previously defined Dynamic Dynamic Barcode Data fields.	Enter the Special Function Control Character followed by BF, then by the assigned number of the desired Barcode Data Field and end the entry with the parameter delimiter.

Table 17. EXECUTE Form Command Optional Parameters

Field	Function	Option/Modifier Selections
‡data field‡	The barcode data that will be printed in the specified location.	Enter the data parameter delimiter followed by the barcode data string and end the entry with the data parameter delimiter.
The following parameters are used in forms where incremental Dynamic Alphanumeric Data fields have been defined. You may enter as many of the following fields as were originally defined on the form.		
♦ IAFn;	Optional parameter that specifies one of the previously defined Incremental Dynamic Alphanumeric Data fields.	Enter the Special Function Control Character followed by IAF, then by the assigned number of the desired Incremental Alphanumeric Data Field. End the entry with the parameter delimiter.
idir;	Optional field specifying whether the data field will be incremented or decremented.	Enter a plus sign **+;** or leave this field blank to increment in a positive direction. Enter a minus sign **-;** to decrement the data.
STEPMASK;	This parameter is the increment information field that defines the number of character positions in the data field, the increment amount for each position, and which positions will be incremented.	See the previous explanation on Incremented Static Alphanumeric Strings for how to construct the STEPMASK parameter.
RPTn;	Optional parameter that specifies the number of times this field value is repeated before it is incremented.	Enter **RPT** followed by a numeric value from 1 to 65,535, then the parameter delimiter. The default repeat amount is 1, which means that if you do not enter this parameter or enter only 1, then the field will be incremented each time it prints.
RSTn;	Optional parameter that specifies the number of times this incremented field will print before it is reset to the starting point.	Enter **RST** followed by a numeric value from 1 to 65,535, then the parameter delimiter. The default reset value is 0, which means that the field will not be reset.
‡STARTDATA‡	This parameter defines the ASCII character string that will be printed in the specified location.	See the explanation on Incremented Static Alphanumeric Strings for how to construct the starting field. Enter the delimiter followed by the alphanumeric string and end the entry with the data delimiter. **NOTE:** The delimiter cannot be used in the alphanumeric string.

Table 17. EXECUTE Form Command Optional Parameters

Field	Function	Option/Modifier Selections
The following parameters are used in forms where Incremental Dynamic Barcode Data fields have been defined. You may enter as many of the following fields as were originally defined on the form.		
♦ IBFn;	Optional parameter that specifies one of the previously defined Incremental Dynamic Barcode Data fields.	Enter the Special Function Control Character followed by IBF, then by the assigned number of the desired Barcode Data Field. End the entry with the parameter delimiter.
idir;	Optional field specifying whether the data field will be incremented or decremented.	Enter a plus sign **+;** or leave this field blank to increment in a positive direction. Enter a minus sign **-;** to decrement the data.
STEPMASK;	This parameter is the increment information field that defines the number of character positions in the data field, the increment amount for each position, and which positions will be	See the previous explanation on Incremented Static Alphanumeric Strings for how to construct the STEPMASK parameter.
RPTn;	Optional parameter that specifies the number of times this field value is repeated before it is incremented.	Enter **RPT** followed by a numeric value from 1 to 65,535, then the parameter delimiter. The default repeat amount is 1, which means that if you do not enter this parameter or enter only 1, then the field will be incremented each time it prints.
RSTn;	Optional parameter that specifies the number of times this incremented field will print before it is reset to the starting point.	Enter **RST** followed by a numeric value from 1 to 65,535, then the parameter delimiter. The default reset value is 0, which means that the field will not be reset..
‡STARTDATA‡	This parameter defines the starting incremental bar code field.	Enter the delimiter followed by the barcode data string and end the entry with the data delimiter. See the previous explanation on Incremented Static Alphanumeric Strings for how to construct the starting field. Also see the information on individual barcodes.
If you are using interactive EXECUTE Mode, use the following code to initiate printing.		

Table 17. EXECUTE Form Command Optional Parameters

Field	Function	Option/Modifier Selections
Overlay Data	Overlay data can be entered while in EXECUTE mode onto predefined forms. It is variable interactive Alphanumeric data that is entered into predefined locations by using tabs and spaces to position it horizontally and Line Feeds, Form Feeds, and EVFU programs to position it vertically. Paper Movement Commands initiate printing of the buffer contents.	Enter tabs, spaces, line feeds, form feeds, and EVFU channel selections to position the data then enter the data string. Refer to your Operator/Application Manual for information on EVFU operations.
Form Feed	This command is used to print the remainder of the current form and move the print location to the top of the next form. When you use this command the	Enter Form Feed Command.
Use the following commands to initiate printing of the last form and leave EXECUTE Mode.		
Line Feed	A Line Feed command must separate the Execute Sequence from the NORMAL command.	Enter Line Feed Command
+NORMAL	Initiates printing of last form and returns printer to NORMAL Mode.	Enter the current Special Function Control Character followed by NORMAL. **NOTE:** Do not forget to include a Line Feed before this command.

Box Command (Create)

This command is used to construct boxes at different locations on the form. The basic command for creating a box is:

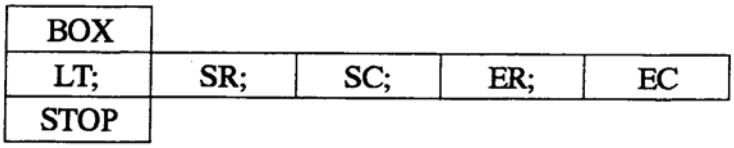


Table 18. Box Command

Field	Function	Option/Modifier Selections
BOX	This is the command specifier. Using the following parameters creates a box that expands down and to the right from the starting point at a specific line thickness.	Enter **BOX**
LT;	Dictates the line thickness.	Line thickness is based on dot increments (1/72″) horizontally and vertically. Line thickness is only limited by your specifications. Do not forget the parameter delimiter.
The SR, SC, ER, and EC parameters use the same methods for defining the BOX Print area. Whether in columns or dots depends on the **SCALE** Command. You may also use the **XX.YY** format for plotting duplication print locations explained earlier in the **Standards Section**.		
SR;	Plots the starting ROW of the box.	Enter a value ranging from 1 to one less than the maximum form length and the parameter delimiter.
SC;	Plots the starting COLUMN of the box.	Enter a value ranging from 1 to one less than the maximum form width and the parameter delimiter.
ER;	Plots the ending ROW of the box.	Enter a range ranging from 2 to one less than the maximum form length and the parameter delimiter.
EC	Plots the ending COLUMN of the box.	Enter a value ranging from 2 to one less than the maximum form width.
STOP	Command Terminator.	Enter STOP and the printer waits for a new command. Leave STOP out and the printer will wait for another BOX Command.

Using The Box Command

This command allows you to create a series of boxes, each smaller than the previous box. Make the first box 4” square and make each successive box smaller by approximately .25”. Each box will have a line thickness of 3 points. Start the first box at row 1 and column 1 (SCALE = DOT). The finished program would be:

~RESET

~CREATE;BOX SCALE;DOT BOX

3;1;1;288;240

3;22;19;267;222

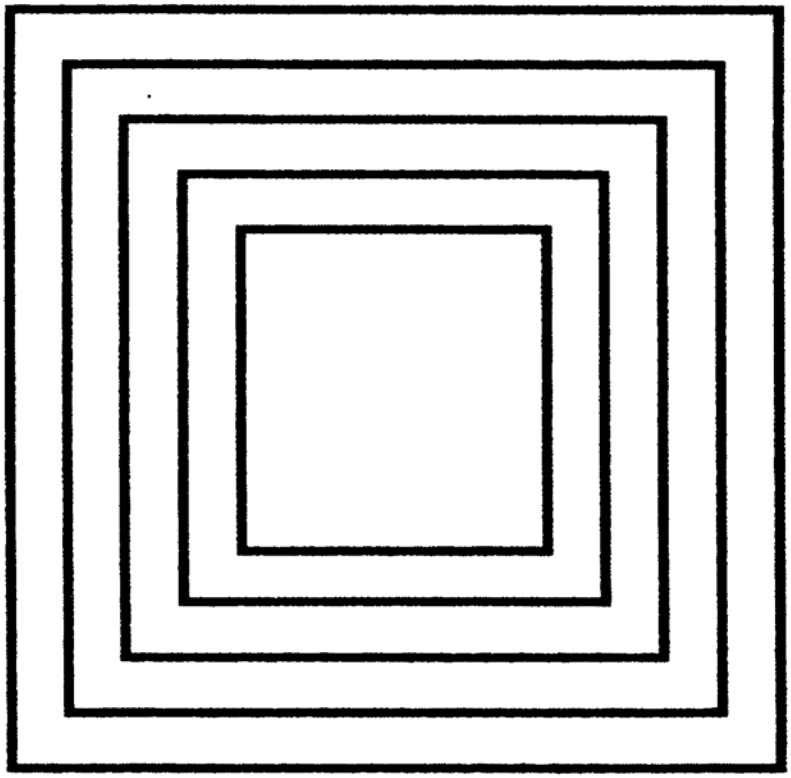
3;43;37;246;186

3;85;73;206;168

STOP END

~EXECUTE;BOX

~NORMAL



Box Samples

If you do not want a closed box around text or the form, but you still want to set off specific areas on the form, use the Corner command to print corner frames on your form.

Call LOGO Command (Create)

This command allows you to call a previously defined logo for printing at a specified place on the form you are creating. The format of this command is:

LOGO		
SR;	SC;	Name
STOP		

Table 19. Call LOGO Command

Field	Function	Option/Modifier Selections
LOGO	Initiates Create LOGO mode.	Enter **LOGO**
The SR and SC parameters use the same methods for defining the LOGO Print area. Whether in columns or dots depends on the **SCALE** Command. You may also use the **XX.YY** format for plotting duplication print locations explained earlier in the **Standards Section**.		
SR;	Plots the starting ROW of the logo.	Enter a value ranging from 1 to one less than the maximum form length, then end the entry with the parameter delimiter.
SC;	Plots the starting COLUMN of the logo.	Enter a value ranging from 1 to one less than the maximum form width, then end the entry with the parameter delimiter.
Name	Specifies a previously defined logo for printing.	Enter the logo name exactly as entered when defining and storing the logo.
STOP	Command Terminator.	Enter STOP and the printer waits for a new command. Leave STOP out and the printer will wait for another LOGO Command.

Change LPI Command (Normal and Execute)

This command selects an LPI setting for printer output. This command affects all printing in PGL mode except Overlay data. It will not reset the value selected in the printer menu system. when you are not using the PGL option, printer output is configured according to the settings in the pritner multi-level menus.

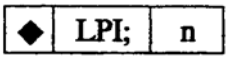


Table 20. Change LPI Command

Field	Function	Option/Modifier Selections
+LPI;	Command Specifier. Change LPI Command.	Enter the current Special Function Control Character followed by **LPI**, then end the entry with the parameter delimiter.
n	Designates the new LPI setting.	Enter desired selection from the following range: 6, 8, 9, and 10.

Change SFCC Command (Normal and Execute)

This command allows you to change the current/default Special Function Control Character to one of your choosing. SFCC command characters may be selected from ASCII characters in Hex address 00-FF. The SFCC command is:

Table 21. Change SFCC Command

Field	Function	Option/Modifier Selections
+SFCC;	Special Function Control Character (SFCC)	Enter the current Special Function Control Character followed by SFCC, then end the entry with the parameter delimiter.
n	Designates the new SFCC Command Character.	Enter desired ASCII character in decimal format. The valid range of values is from 0 - 255. You may also use the optional method of actually specifying the desired character, if it is a printable character, by using the format 'n' to designate the character. If you use this method enter the desired printable character between single quote characters.

NOTE: In addition to being able to change the SFCC using this command we have added the ability to change it through the printer control panel. The SFCC Parameter is located in the Configurations Menu.

Example: To change the SFCC from the default character (~, tilde) to the @ character use either the following commands:

~	SFCC;	64

OR

~	SFCC;	'@'

Create Corner Frames (Create)

This command allows you to crate Corner Frames for your forms. Similar to the BOX command, you define starting and ending points but in addition, you specify the length of the horizontal and vertical arms of each corner frame.

The basic command for creating Corner Frames is:

CORNER						
LT;	SR;	SC;	ER;	EC;	VL;	HL
STOP						

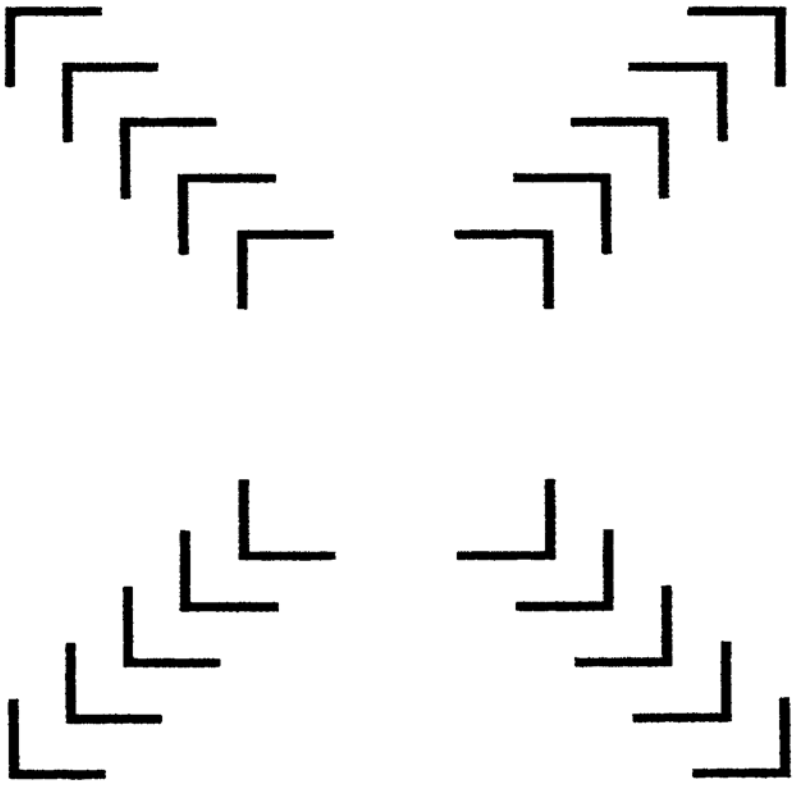
Table 22. Corner Command

Field	Function	Option/Modifier Selections
CORNER	This is the command specifier. Using the following parameters creates a set of corners around a specified area of the form at a specific line thickness and length of vertical and horizontal arms of each corner.	Enter **CORNER**
LT;	Dictates the line thickness.	Line thickness is based on dot increments (1/72″) horizontally and vertically. Line thickness is only limited by your specifications. Do not forget the parameter delimiter.
The SR, SC, ER, and EC parameters use the same methods for defining the CORNER Print area. Whether in columns or dots depends on the **SCALE** Command. You may also use the **XX.YY** format for plotting duplication print locations explained earlier in the **Standards Section**.		
SR;	Plots the starting ROW of the corner.	Enter a value ranging from 1 to one less than the maximum form length and the parameter delimiter.
SC;	Plots the starting COLUMN of the corner.	Enter a value ranging from 1 to one less than the maximum form width and the parameter delimiter.
ER;	Plots the ending ROW of the corner.	Enter a range ranging from 2 to one less than the maximum form length and the parameter delimiter.
EC	Plots the ending COLUMN of the corner.	Enter a value ranging from 2 to one less than the maximum form width.
The VL and HL parameters specify the arm length of each corner piece in dots or characters depending on the **SCALE** Command. You may also use the **XX.YY** format.		
VL;	Specifies the length of the vertical arms of each corner.	Enter a value of 1 or greater. Do not forget the parameter delimiter.
HL	Specifies the length of the horizontal arms of each corner.	Enter a value of 1 or greater.
STOP	Command Terminator.	Enter STOP and the printer waits for a new command. Leave STOP out and the printer will wait for another CORNER Command.

Using the Corner Command

The corner command is used and set up similar to the BOX Command, with a few additional parameters. Use the same basic program for this parameter and add two specifications for controlling arm lengths. Create a set of corner frames on our form with the same starting and ending points (SCALE = DOT) used in the BOX program presented previously, and add arm lengths of 30 dots. The finished program would be:

```
~RESET
~CREATE;CORNER SCALE;DOT CORNER
3;1;1;288;240;30;30
3;22;19;267;222;30;30
3;43;37;246;204;30;30
3;64;55;225;186;30;30
3;85;73;206;168;30;30
STOP END
~EXECUTE;CORNER
~NORMAL
```



Corner Coordinates

Create Form Command (Normal)

This command is used to enter CREATE Form Mode. The command structure for entering CREATE Mode is:

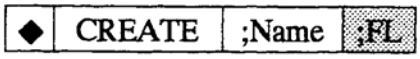


Table 23. Enter CREATE Mode Command

Field	Function	Option/Modifier Selections
+CREATE	Indicates CREATE Mode.	Enter the current Special Function Control Character followed by **CREATE**.
;Name	Enter the name of the form in this field.	Form names are allowed a maximum of eight characters. The followig ASCII characters may be used in form names: • Upper and lower-case A - Z (HEX 41 - 5A and 61 - 7A) • Number 0 - 9 (HEX 30 - 39) • Dollar sign $ (HEX 24) • Percent Sign % (HEX 25) • Dash - (HEX 2D) • At sign @ (HEX 40) • Braces { } (HEX 7B and 7D) • Parens ( ) (HEX 28 and 29) • Tilde ~ (HEX 7E) • Single quotes ‘ ‘ (Hex 60 and 27) • Exclamation point ! (HEX 21) • Pound sign # (HEX 23) • Amersand & (HEX 26)
;FL Form Length	This parameter sets the Form Length in three ways: • Automatic form length • Specific form length in dot rows. • Default form length in dot rows (Also see LFORM 6 and LFORM 8 Commands)	Enter 0 to set the Form Length to end automatically after the longest element. Enter a value from 1 - 65,535 dot rows to set the Form Length to a specified amount. To compute the desired number of dot rows multiply length in inches by 72. To set form length to the default (792 dot rows) do not enter a value in this field.

Using the Create Form Command

This example shows how to design an employment application form for an Explosives Handler for a Demolition Company. To enter CREATE Mode and design a form called JobAppl, send the following command to your printer:

- CREATE;JobAppl

To enter CREATE Mode and design the same form, send the following command to your printer:

- CREATE;JobAppl

To enter CREATE Mode and assign a Form Length of 10 inches, send the following command to your printer:

- CREATE;JobAppl;**720**

After entering CREATE Form Mode send the required commands to complete the form.

Now that you have informed the printer that you wish to CREATE a form called JobAppl that is 10 inches long, you can finish creating your form using the appropriate commands in the CREATE Form Mode.

Create Logo Command (Normal)

This command allows you to create logos for your forms. It initiates the Logo CREATE Mode, which enables you to create and store logos that will be called in CREATE Mode for use in creating forms. Logos are defined using specific dot placement within a grid.

This command names the logo then specifies a grid size for the logo and which dot positions within the grid will be printed. The grid is 72 DPI vertically and 60 DPI horizontally. The maximum size of the grid is 252 dot rows (3.5 inches) by 240 dot columns (4 inches). When defining the grid, it is best not to exceed the actual requirements needed to produce the logo. For example, if your logo is going to be 1 inch high by 1.5 inches tall, the grid would be the same size in dots (72 by 90). The basic command for creating a logo is:

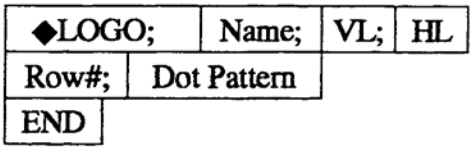


Table 24. CREATE LOGO Mode Command

Field	Function	Option/Modifier Selections
◆LOGO;	Initiates Create LOGO Mode.	Enter the current Special Function Control Character followed by **LOGO**, then end the entry with the parameter delimiter.
Name;	Enter the name of the Logo in this field.	LOGO names are allowed a maximum of eight characters. The following ASCII characters may be used in form names: • Upper and lower-case A - Z (HEX 41 - 5A and 61 - 7A) • Number 0 - 9 (HEX 30 - 39) • Dollar sign $ (HEX 24) • Percent Sign % (HEX 25) • Dash - (HEX 2D) • At sign @ (HEX 40) • Braces { } (HEX 7B and 7D) • Parens ( ) (HEX 28 and 29) • Tilde ~ (HEX 7E) • Single quotes ‘ ‘ (Hex 60 and 27) • Exclamation point! (HEX 21) • Pound sign # (HEX 23) • Ampersand & (HEX 26) **NOTE:** Make sure your LOGO name is not identical to one that is already stored in memory. The new LOGO will replace the old LOGO if this happens.
VL;	Vertical size of the Grid at 72 DPI.	Enter a value from 1-252, followed by the parameter delimiter. Do not enter an amount larger than necessary to print the Logo.
HL	Horizontal size of the Grid at 60 DPI.	Enter a value from 1-240, followed by the parameter delimiter. Do not enter an amount larger than necessary to print the Logo.
Dot Pattern	Specifies dot(s) that will be printed in the specified row.	This information is entered in two ways: single dots or sequences of dots. Several single or sequences of dots can be defined for each row. Proper format for entering sequences of dots is beginning dot - ending dot (23-44). Each of the entries must be separated by a semicolon.

Table 24. CREATE LOGO Mode Command

Field	Function	Option/Modifier Selections
END	Command Terminator.	Enter **END**.

Using the LOGO Command

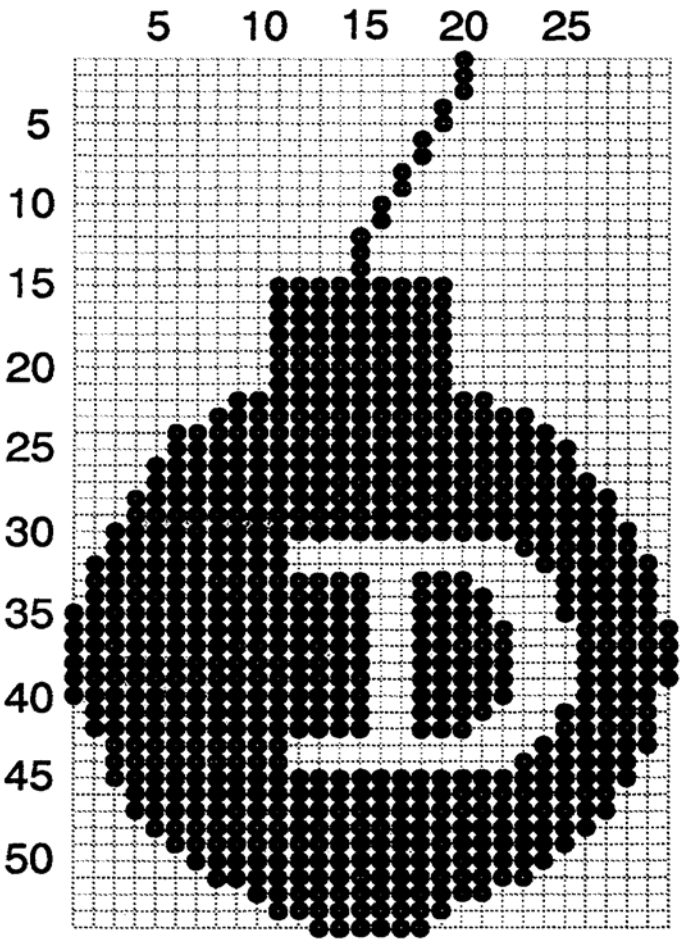
This section illustrates the use of the LOGO Command to create a logo for the Job Application Form we created earlier for a demolition company. Since this is a demolition company, we will create the logo as a Bomb with the uppercase letter D superimposed on it. The dimensions of the grid will be

.5 x .75 inches.

First plot the logo on a grid so you know which dots are necessary for printing. Then compose the dot pattern information for each row of the graphic. Use the following graphic as the logo for the company. The syntax on below creates the BOMB logo plotted.

◆ LOGO;BOMB;54;30 1;20
2;20
3;20
4;19
5;19
6;18
7;18
8;17
9;17
10;16
11;16
12;15
13;15
14;15
15;11-19
16;11-19
17;11-19
18;11-19
19;11-19
20;11-19
21;11-19
22;9-21
23;8-23
24;6-24
25;6-25
26;5-25
27;5-26
28;4-27
29;4-27
30;3-28
31;3-11;23-28

32;2-11;24-29
33;2-15;18-20;25-29
34;2-15;18-21;25-29
35;1-15;18-22;26-29
36;1-15;18-22;26-30
37;1-15;18-22;26-30
38;1-15;18-22;26-30
39;1-15;18-22;26-30
40;1-15;18-22;26-29
41;2-15;18-21;25-29
42;2-15;18-20;25-29
43;1-11;24-29
44;3-11;23-28
45;3-28
46;4-27
47;4-27
48;5-26
49;6-25
50;7-24
51;8-23
52;10-21
53;11-19
54;13-18
END



Delete Form Command (Normal)

This command is used in NORMAL Mode to delete a stored form from memory.

Table 25. Delete Form Command

Field	Function	Option/Modifier Selections
◆ DELETE FORM;	Command Specifier	Enter the current Special Function Control Character, followed by **DELETE FORM**, then end the entry with the parameter delimiter.
NAME	Parameter that specifies desired Form to delete.	Enter the name of a specific form you want to delete or ***ALL** to delete all forms.

Delete Logo Command (Normal)

This command is used in NORMAL Mode to delete a stored logo from memory.

Table 26. Delete Logo Command

Field	Function	Option/Modifier Selections
◆ DELETE LOGO;	Command Specifier	Enter the current Special Function Control Character, followed by **DELETE LOGO**, then end the entry with the parameter delimiter.
NAME	Parameter that specifies desired Logo to delete.	Enter the name of a specific form you want to delete or ***ALL** to delete all logos.

Density Command (Normal and Execute)

This command selects a Characters per Inch for print output.

Table 27. Density (CPI) Command

Field	Function	Option/Modifier Selections
+DENSITY;	Command Specifier	Enter the current Special Function Control Character, followed by **DENSITY**, then end the entry with the parameter delimiter.
n	Parameter that specifies desired CPI setting.	Valid selections are 10, 12, 13, 15, 17. 10A to select 10 CPI OCR-A and 10B to select 10 CPI OCR-B.

Directory Command (Normal)

This command is used in NORMAL Mode to print a directory of all defined forms and logos, logo assignments to forms, available memory, and how memory is currently used for form and logo storage.

To print the directory, enter the current Special Function Control Character (SFCC) followed by **DIRECTORY**.

Expanded Print Command (Normal and Execute)

This command allows you to expand alphanumeric characters horizontally and vertically.

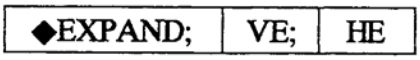


Table 28. Expanded Print Command

Field	Function	Option/Modifier Selections
+EXPAND;	Command Specifier. Instructs printer to enter Expanded Print mode.	Enter the current Special Function Control Character, followed by **EXPAND**, then end the entry with the parameter delimiter.
VE;	Vertical expansion amount.	Enter a value specifying the desired expansion factor, then end the entry with the parameter delimiter. Valid selections range from 0-113. A selection of 0 returns you to standard size characters.
HE	Horizontal expansion amount.	Enter a value specifying the desired expansion factor, then end the entry with the parameter delimiter. Valid selections range from 0-113. A selection of 0 returns you to standard size characters.

Horizontal and Vertical Line Commands (Create)

In addition to boxes and corners, you can print horizontal and vertical lines on your forms. The following commands control location, length, and thickness of horizontal and vertical lines.

Horizontal Line Command

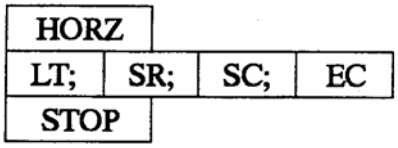


Table 29. Horizontal Line Command

Field	Function	Option/Modifier Selections
HORZ	This is the command specifier. The following parameters define print location and line thickness of a horizontal line.	Enter **HORZ**

Table 29. Horizontal Line Command

Field	Function	Option/Modifier Selections
LT;	Dictates the line thickness.	Horizontal line thickness is based on dot increments (1/72″). Line thickness is only limited by your specifications.
HE	Horizontal expansion amount.	Enter a value specifying the desired expansion factor, then end the entry with the parameter delimiter. Valid selections range from 0-113. A selection of 0 returns you to standard size characters.
The SR, SC, and EC parameters control placement, length, and thickness of horizontal lines on a form. Whether in columns or dots depends on the **SCALE** Command. You may also use the **XX.YY** format for plotting duplication print locations explained earlier in the **Standards Section**.		
SR;	Plots the starting ROW of the line.	Enter a value ranging from 1 to one less than the maximum form length and the parameter delimiter.
SC;	Plots the starting COLUMN of the line.	Enter a value ranging from 1 to one less than the maximum form width and the parameter delimiter.
EC	Plots the ending COLUMN of the line.	Enter a value ranging from 2 to one less than the maximum form width.
STOP	Command Terminator.	Enter **STOP** and the printer waits for a new command. Leave STOP out and the printer will wait for another Horizontal Line Command.

Vertical Line Command

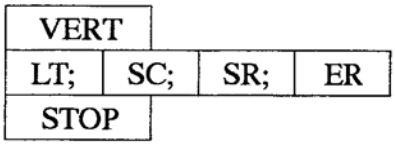


Table 30. Vertical Line Command

Field	Function	Option/Modifier Selections
VERT	Command Specifier.	Enter **VERT**
The following parameters define print location and line thickness of a vertical line. The SR, SC, and EC parameters control placement, length, and thickness of vertical lines on a form. Whether in columns or dots depends on the **SCALE** Command. You may also use the **XX.YY** format for plotting duplication print locations explained earlier in the **Standards Section**.		

Table 30. Vertical Line Command

Field	Function	Option/Modifier Selections
LT;	Dictates the line thickness.	Vertical line thickness is based on dot increments (1/60″), then end the entry with the parameter delimiter. Line thickness is only limited by specifications.
SC;	Plots the starting COLUMN of the line.	Enter a value ranging from 1 to one less than the maximum form width and the parameter delimiter.
SR;	Plots the starting ROW of the line.	Enter a value ranging from 1 to one less than the maximum form length and the parameter delimiter.
ER	Plots the ending ROW of the line.	Enter a value ranging from 2 to one less than the maximum form length.
STOP	Command Terminator.	Enter **STOP** and the printer waits for a new command. Leave STOP out and the printer will wait for another Vertical Line Command.

Ignore Commands (Normal, Create, and Execute)

Good parameters document what they are doing in different sections of their code by including things such as Comment Fields or REM statements. In PGL the same type of information can be included in PGL files by bracketing them with the IGON and IGOFF commands.

In addition, some host systems pad the data stream with characters and spaces before the SFCC. You can use these commands to cause the printer to ignore these additional characters by inserting the IGON at the end of the last PGL file and IGOFF at the beginning of the next PGL file. After sending an IGON Command the printer will ignore ALL characters until it receives an IGOFF Command. The commands that control the Ignore Mode are:

◆ IGON AND ◆ IGOFF

Table 31. Ignore Commands

Field	Function	Option/Modifier Selections
IGON		
◆ IGON	Turns Ignore Mode on. The printer ignores all characters it receives until it receives the IGOFF sequence.	Enter the current Special Function Control Character followed by **IGON**.
IGOFF		
◆ IGOFF	Turns Ignore Mode off. The printer begins processing normally.	Enter the current Special Function Control Character followed by **IGOFF**.

Normal Command (Normal and EXECUTE Mode)

This command can be used in all modes. If the printer is in CREATE mode or EXECUTE mode, this command places the printer back into NORMAL Mode. If you are already in NORMAL Mode, sending this command will reset the printer to its Font and LPI defaults. The command format is:

Table 32. NORMAL Mode Command

Field	Function	Option/Modifier Selections
◆ NORMAL	Initiates NORMAL Mode.	Enter the current Special Function Control Character followed by **NORMAL**.

PGL Mode Control

Quiet Command (Normal)

This command disables the PGL Mode of your printer. The only PGL command the printer will respond to is the Listen Command. The Quiet Command format is:

Table 33. Quiet Command

Field	Function	Option/Modifier Selections
◆ QUIET	Command Specifier. Turns off Graphics Mode.	Enter the current Special Function Control Character followed by **QUIET**.

Listen Command (Normal)

This command re-enables the PGL Mode of your printer after a Quiet Command. The printer will respond appropriately to PGL Commands it receives. The following command enables Graphics Mode:

Table 34. Listen Command

Field	Function	Option/Modifier Selections
◆ LISTEN	Command Specifier. Turns off Graphics Mode.	Enter the current Special Function Control Character followed by **LISTEN**.

Reset Command (Normal and Execute)

This command deletes ALL forms and logos from printer memory and resets configuration parameters to their default values. If forms and logos have been loaded by other users they will be deleted also. You may also eliminate forms and logos by sending a Delete Form or Delete Logo Command. The Reset Command is:

Table 35. Reset Command

Field	Function	Option/Modifier Selections
♦ RESET	Erases ALL forms and logos from printer memory.	Enter the current Special Function Control Character followed by **RESET**.

Scale Command (Create)

There are two commands that define the format that will be used in specifying where various elements of the form will be placed.

Dot Scale Placement	
SCALE;	DOT

Character Scale Placement			
SCALE;	CHAR	;LPI	;CPI;

Table 36. Scale Command

Field	Function	Option/Modifier Selections
Dot Scale Placement		
SCALE;	Command Specifier.	Enter **SCALE;**
DOT	Specifies that placement will be done using a dot-scale.	Enter DOT to specify a dot measurement scale (dot-columns and dot-rows).
Character Scale Placement		
SCALE;	Command Specifier.	Enter **SCALE;**
CHAR	Specifies that placement will be done using a character scale.	Enter **CHAR;** to specify a character measurement scale (columns and rows).
;LPI	An optional parameter that can be used to specify Lines Per Inch setting for print output.	Selections that can be used are 6, 8, 9, and 10. No value will result in the default setting of 6 LPI. Do not forget the parameter delimiter.
;CPI	An optional parameter that can be used to specify Character Per Inch setting for print output.	Selections that can be used are 10, 12, 13, 15, and 17. No value will result in the default setting of 10 LPI.

Select Format (SF) Command (Normal, Create, and Execute)

These commands allow you to configure PGL to ignore characters from 00 to 0F Hex. This enables the printer to ignore all host-generated paper movement commands.

♦ SFOFF AND ♦ SFON

Table 37. Set Format Commands

Field	Function	Option/Modifier Selections
SFON		
♦ SFON	Turns Select Format Mode on. The printer ignores all characters from 00 to 0F Hex until it receives the SFOFF sequence.	Enter the current Special Function Control Character followed by **SFON**.
SFOFF		
♦ SFOFF	Turns Select Format Mode off. The printer begins processing normally.	Enter the current Special Function Control Character followed by **SFOFF**.

When SF Mode is activated you may still send paper movement commands by using one of the following commands.

Enter Result

♦ CR Carriage Return

♦ LF Line Feed

♦ FF Form Feed

These commands will have no effect after the SFOFF Command has been sent.

Practice

Earlier in this chapter we presented some sample commands for creating various graphical elements for a form.

Earlier we created a logo named BOMB and a form called JobAppl. Now let's modify our earlier work and put them together with some Alphanumerical commands, line commands, and a BOX command to create a Job Application for a company called "Demolitions Are US."

The logo we designed is fine the way it is, but to create a form that will fit on a page of this manual we will have to modify the BOX and CREATE Form command in the examples we used earlier.

The overall form dimensions will be 5.6 inches wide by 8.5 inches long. It will be printed at 10 CPI and 6 LPI. These figures give us a maximum line length of 70 columns and form length of 51 lines (form length in dot rows would be 612).

We have already created and stored the logo for the form. Now we will create the form itself. First enter the CREATE Form Mode.

```
~CREATE;JobAppl;612
```

After we enter the CREATE Mode, create a border for our form by sending a BOX Command specifying a box of 3 points thickness that prints three columns from the left and right edge and three lines from the top and bottom edge.

```
BOX 3;3;3;48;52 STOP
```

We started our form creation and added a border, now we will compose the items that will go inside the box.

Remember that the BOMB logo we created earlier has a superimposed "D" on it. The bomb is going to be part of the company title on the form. To finish the form header, create an Alphanumeric string with the

remaining characters of the title to align properly with the bomb making a form header. Also, we will create a form title, the instructions for the form, and end with the first question.

```
ALPHA
4;30;0;0;*Job Application*
5;30;0;0*Explosives Handler*
E;D;7;10;0;0;*emolitions Are US*
10;6;0;0;*Answer ALL Questions. If you do not under-*
11;6;0;0;*stand a question or have no answer for a *
12;6;0;0;*question, stop where you are and deposite your *
13;6;0;0;*application in the receptacle by the front *
14;6;0;0;*door as you leave the premises*
16;6;0;0;*1. *
16;10;0;0;*Are you afraid of loud noises?*
STOP
```

Follow this section with a horizontal line command to place a line where the answer to the first question should go.

```
HORZ
1;16;10;41;46
STOP
```

Continue this procedure until all questions with answering spaces composed and in place. Also include a closing logo.

The following program combined with the program for creating the BOMB log would produce the form in Figure 5 on page 158.

```
~CREATE;JobAppl;612
BOX
3;3;3;48;52
STOP
LOGO
4;5;BOMB
42;5;BOMB
STOP
ALPHA
4;30;0;0;*Job Application*
5;30;0;0;*Explosives Handler*
E;D;7;10;0;0;*emolitions Are US*
10;6;0;0;*Answer ALL Questions.  If you do not under-*
11;6;0;0;*stand a question or have no answer for a *
12;6;0;0;*question, stop where you are and deposit your *
13;6;0;0;*application in the receptacle by the front *
14;6;0;0;*door as you leave the premises.*
16;6;0;0;*1.*
16;10;0;0;*Are you afraid of loud noises?*
17;10;0;0;*If answer is yes, go no farther and depo-*
18;10;0;0;*sit your application in the receptacle by *
19;10;0;0;*the front door as you leave the premises.*
21;10;0;0;*1a.*
21;14;0;0;*Can you hear at all?*
23;10;0;0;*1b.*
23;14;0;0;*If yes, continue on.*
24;14;0;0;*If no, continue on.*
26;6;0;*2.*
26;10;0;0*Name (no Xs please)*
28;6;0;0;*3.*
28;10;0;0;*Address:*
33;6;0;0;*4.*
33;10;0;0;*Do you have all of your fingers?*
35;6;0;0;*5.*
35;10;0;0;*Have you ever been convicted of any*
36;10;0;0;*crime(s) involving explosives?*
E;D;45;10;0;0;*emolitions can be YOU TOO.*
STOP
HORZ
1;16.10;41;46
1;21.10;34;42
1;26.10;30;50
1;29.10;15;50
1;30.10;15;50
1;31.10;15;50
1;33.10;42;50
1;36.10;42;50
STOP
END
~EXECUTE;JobAppl

~NORMAL
```

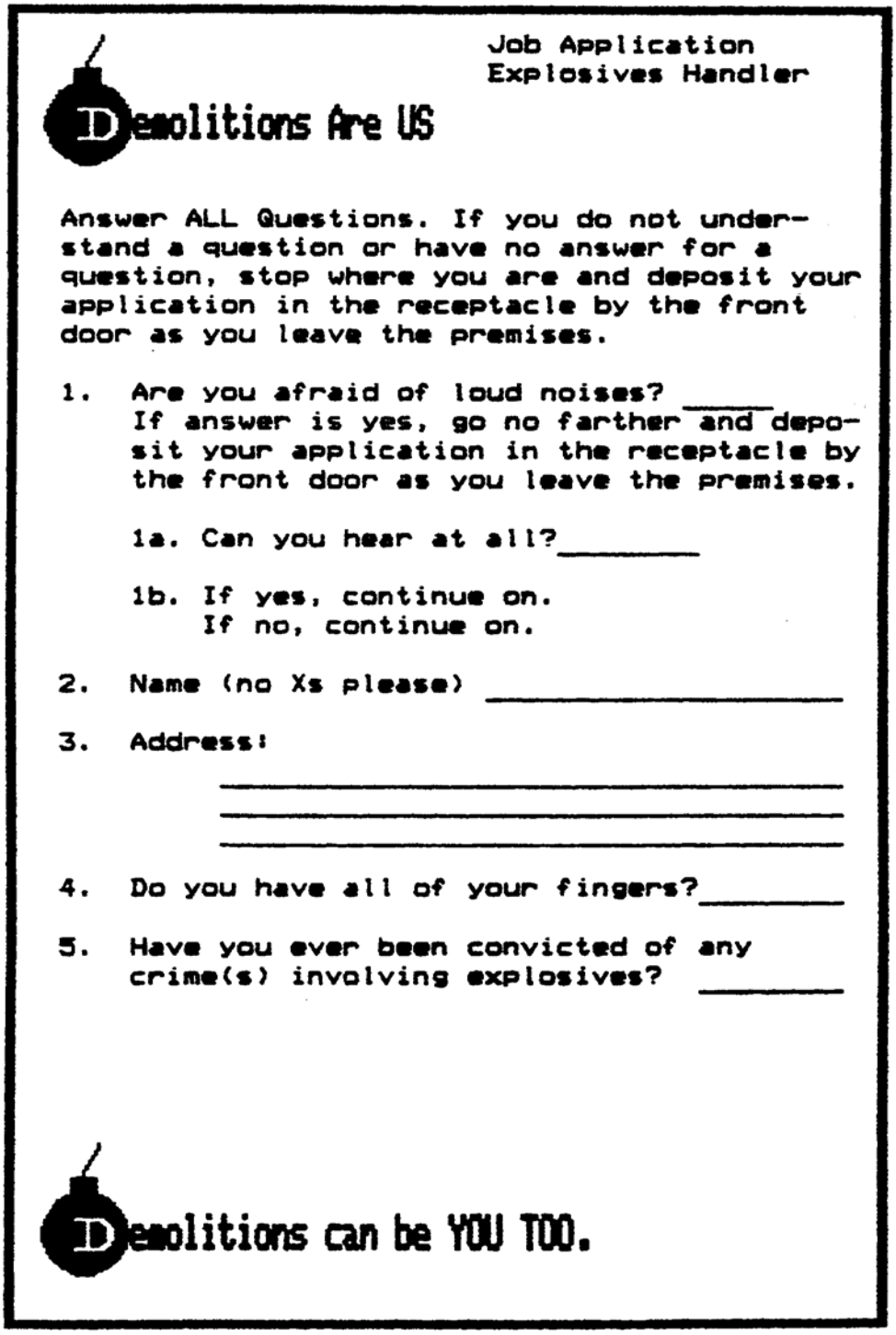


Figure 5. Sample Form Printout

10 *PGL Barcodes*

PGL Barcodes (Used in CREATE Mode Only)

PGL incorporates several barcode styles in its command set. Most barcodes use the same basic set of parameters. Barcode commands follow the same requirements as other PGL commands as far as how they are entered, ability to change orientation of printer output, adjust darkness of output, etc. They can also be printed in incremented format.

For this section of the manual we first explain all of the barcode parameters, followed by a table showing which parameters each barcode uses. After the table we present the exceptions for each barcode and how to increment barcodes.

NOTE:

‡ = Alphanumeric text string delimiter.

; = Parameter delimiter.

Barcode Command Parameters

BARCODE

This is the Barcode Command. It notifies the printer that the data stream that follows is to be used to format and define a specific barcode type.

Enter: BARCODE

Name/CD

This parameter specifies the type of barcode that will be printed. Table 38 on page 159 lists barcodes that

Table 38. PGL Barcodes

Enter	Function
C3/9CD	Specifies a Code 39 type bacode. CD is an optional parameter that instructs the printer to calculate and plot the optional Modulo-43 check digit.
C128B	Specifies a Code 128B Barcode.
C128C	Specifies a Code 128C Barcode.
EAN8	Specifies an EAN 8 Barcode.
EAN13	Specifies an EAN 13 Barcode.
I-2/5CD	Specifies an Interleaved 2/5 Barcode. CD is an optional parameter that instructs the printer to calculate and plot the optional Modulo-10 check digit.

Table 38. PGL Barcodes

Enter	Function
MSIn	Specifies an MSI Barcode. The n value specifies the type of check digit combinations. If n = A = Single Digit modulo-10 followed by a second modulo-10 digit. B = Single Digit modulo-11 followed by a second modulo-11 digit. C = Single Digit modulo-10 D = Single Digit modulo-11
POSTNET	Specifies a POSTNET Barcode.
UPC-A	Specifies a UPC-A Barcode.
UPC-E	Specifies a UPC-E Barcode.

VSCAN (Vertical Scan)

Optional parameter that changes barcode print orientation. When included in the command structure it causes the printer to print the barcode in vertical orientation (90 degrees counterclockwise rotation).

Enter: VSCAN

Magnification Value (X-Factor)

Optional parameter that specifies a horizontal expansion amount (X-Factor) for the barcode output. When used this value causes an increase in the physical size of the barcode. Table 39 lists the expandable barcodes and allowable expansion factors.

Table 39. X-Factor VS Barcode Style

ENTER	**Barcode Style**				
X-Factor	C3/9	C128B	C128C	I2/5	MSI A, B, C, & D
X1	✓	✓	✓	✓	✓
X1A	✓			✓	
X1B	✓			✓	
X1.5		✓	✓		
X2	✓	✓	✓	✓	✓
X2A				✓	
X3	✓	✓	✓	✓	✓
X4	✓	✓	✓	✓	✓

Hn (Height Expansion)

Optional parameter that specifies height expansion of barcode and human readable character output. Adjustments are made in 0.1 inch increments. The valid range of selections is 4 to 99 (0.4 to 9.9 inches). The default value is

0.9 inches.

Enter: H followed by the desired expansion amount.

Examples:

H1 = Expansion of 0.1”

H20 = Expansion of 2.0”

H45 = Expansion of 4.5”

BFn;L (Dynamic Barcode Data Field)

Optional field that identifies location and the length of a dynamic Barcode field. To select this parameter:

Enter BF followed by:

n — to identify the barcode data that will be located at the coordinates specified by the SR and SC parameters. The valid range of values is 1-255.

L — to specify the number of characters in the dynamic barcode field. The length of the dynamic string must be less than or equal to the value signified by the L parameter. This value is not used for Barcodes with fixed length data fields (EAN and UPC barcodes).

If this option is enabled, do not enter static barcode data in the optional data field.

Dark (Dark Printing)

Optional parameter that enables DARK Printing. See “Half-Dot Mode (Double Density)” on page 29.

Enter: DARK or D to initiate double-density printing.

SR (Staring Row)

This parameter specifies the Starting Row of the barcode.

Plots the starting row of the field. Whether in dots or characters depends on the SCALE Command. You may also use XX.YY format for plotting starting print locations explained earlier in the Standards Section.

Enter a value ranging from 1 to less than the maximum form length.

SC (Starting Column)

Plots the starting column of the field. Whether in dots or characters depends on the SCALE Command. You may also use XX.YY format for plotting starting print locations explained earlier in the Standards Section.

Enter a value ranging from 1 to less than the maximum form width.

‡ data field ‡

Static barcode data that will be printed at the position specified by SR and SC.

This alphanumeric string must be set off by the Character String Delimiter (‡). See “Data Field Characters” on page 163.

NOTE: If you have specified dynamic barcode fields earlier in this command, do not use this parameter.

Data Field length and allowable characters depend on the barcode being defined. The Character String Delimiter character cannot be used as printable data, but must be included at the beginning and end of the data string.

PDF

Optional parameter that manipulates printing of human readable characters when it is present. In UPC and EAN barcode this parameter is not necessary to print human-readable characters (human readable characters are the default condition for these barcodes).

Enter: PDF

NOTE: Do not use this parameter if a null data field is specified.

LOC (Location of Human Readable Characters)

Optional parameter that changes the location of the human readable characters.

Enter: No value or B to cause human readable characters to print below the barcode.

Enter: A to cause human-readable characters to print above the barcode.

FONT

Optional parameter that selects a font for the human readable characters or suppress printing of human readable characters and trailing portions of the barcode.

Enter:

N = 10 CPI ASCII

O = 10 CPI OCR-A

X = 10 CPI OCR-B

S = Suppress printing the human readable characters and trailing portions of the barcode.

Examples:

PDF;A;X	Selects human readable characters above the barcode in 10 CPI OCR-B.
PDF;S	Suppresses printing of human readable characters in the barcode.

STOP

This parameter ends the Barcode Command. The printer remains in Create Mode.

Enter: STOP

This must be present at the end of each individual barcode. For every BARCODE command you must have a STOP command.

Command Parameters VS Barcode Types

Figure 6 provides a listing of the Barcode Command Parameters which are used by each Barcode Type.

NOTE: Since all Barcode Types use the BARCODE, Name, and STOP parameters they are not present in the figure. The first parameter is the CD Parameter.

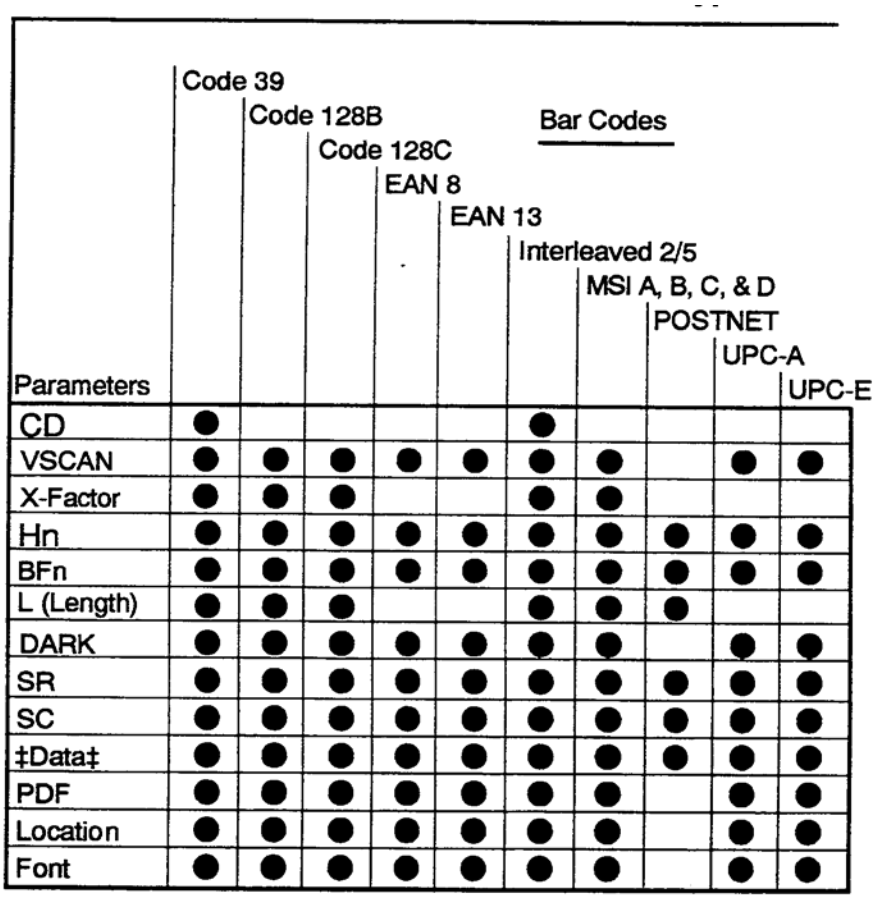


Figure 6. Command Parameters VS Barcode Types

Data Field Characters

All of the barcodes presented in this chapter use the numeric characters 0-9 (HEX 30-39). The only exceptions to this rule are the Code 39 Barcode and the Code 128B Barcode. The Code 39 and Code 128B barcodes will also print alpha characters and some symbols.

None of the barcodes can use the SFCC as part of a data field.

Code 39 Data Field Characters

The data field length for Code 39 Barcodes is variable. It is best to limit field length to no more than 32 characters to minimize the potential for scanning errors.

Table 40 lists the available characters for Code 39 Barcodes.

Table 40. Code 39 Character Set

Character	HEX Value	Character	HEX Value	Character	HEX Value
Space	20	8	38	M	4D
$	24	9	39	N	4E
%	25	A	41	O	4F
+	2B	B	42	P	50

Table 40. Code 39 Character Set

Character	HEX Value	Character	HEX Value	Character	HEX Value
-	2D	C	43	Q	51
.	2D	D	44	R	52
/	2F	E	45	S	53
0	30	F	46	T	54
1	31	G	47	I	55
2	32	H	48	V	56
3	33	I	49	W	57
4	34	J	4A	X	58
5	35	K	4B	Y	59
6	36	L	4C	Z	5A
7	37				

Code 128B and 128C Data Field Characters

The data field length for 128B and 128C Barcodes is variable. It is best to limit field length to no more than 32 characters to minimize the potential for scanning errors.

While Code 128C Barcodes will only print paired numeric values, you can send unpaired numeric, alpha characters, and symbols to the printer at the same time. The printer will shift from 128C to 128B and vice versa as necessary to print all of the printable data.

Table 41 lists the available characters for Code 128B Barcodes.

Table 41. Code 128B Character Set

Character	HEX Value	Character	HEX Value	Character	HEX Value
Space	20	G	47	g	67
!	21	H	48	h	68
“	22	I	49	i	69
#	23	J	4A	j	6A
$	24	K	4B	k	6B
%	25	L	4C	l	6C
&	26	M	4D	m	6D
‘	27	N	4E	n	6E
(	28	O	4F	o	6F
)	29	P	50	p	70

Table 41. Code 128B Character Set

Character	HEX Value	Character	HEX Value	Character	HEX Value
*	2A	Q	51	q	71
+	2B	R	52	r	72
,	2C	S	53	s	73
-	2D	T	54	t	74
.	2E	U	55	u	75
/	2F	V	56	v	76
0	30	W	57	w	77
1	31	X	58	x	78
2	32	Y	59	y	79
3	33	Z	5A	z	7A
4	34	[	5B	{	7B
5	35	\	5C	|	7C
6	36	]	5D	}	7D
7	37	^	5E	~	7E
8	38	_	5F	DEL	7F
9	39	‘	60	FNC1	21
A	41	a	61	FNC2	22
B	42	b	61	FNC3	23
C	43	c	63	FNC4	24
D	44	d	64	CODEA	25
E	45	e	65	CODEC	27
F	46	f	66	SHIFT	28

To access the codes in the greyscale area of the table you must send a SO (OE) character followed by the HEX value for the desired code.

Code 128C character set includes Start B code (HEX 26).

Codes EAN 8 and EAn 13 Data Fields

The required field length for EAN 8 Barcodes is seven characters and the required field length for EAn 13 Barcodes is 12 characters. If the data field is too long or too short, an error message along with the offending program code will print in the place of the barcode. The data fields for EAN 8 and EAN 13 Barcodes can be comprised of the numeric characters 0-9 and special start, stop, and center codes.

Interleaved 2/5 (I-2/5) Data Fields

The data field length for I-2/5 Barcodes is variable, however it is best to limit field length to no more than 32 characters to minimize the potential for scanning errors. In addition, the I-2/5 Barcode requires an even number of characters in its data field. If an odd number of characters is sent the printer automatically adds a leading zero to even out the field. The data field for Interleaved 2/5 Barcodes can be comprised of the numeric characters 0-9.

MSI Data Fields

MSI data fields are variable in length up to a maximum of 14 if one check digit is used and 13 if two check digits are used. The data field for MSI Barcodes can be comprised of the numeric characters 0-9.

POSTNET Data Fields

POSTNET Barcodes use two data field lengths. A 9 digit data field for Zip + 4 and an 11 digit data field for Advanced Barcode format. The data field for POSTNET Barcodes can be comprised of the numeric characters 0-9.

UPC-A Codes

This barcode requires exactly 11 digits. The first character is translated as the number system character. The remaining 10 characters constitute the barcode data. The data field for UPC A Barcodes can be comprised of the numeric characters 0-9.

UPC-E Codes

This command requires exactly 11 digits. The first character is always 0. The next five characters constitute the manufacturer's number and the five ending characters represent the item number. The data field for UPC E Barcodes can be comprised of the numeric characters 0-9.

Incrementing Barcode Data

The basic format for incrementing barcodes are as follows.

For Fixed Barcode Data Fields

The bold characters indicate the added parameters that control incrimination of barcode data.

```
BARCODE
TYPE;VSCAN;MAG;Hn;I;DARK;SR;SC
idir;STEPMASK;RPTn;RSTn;*STARTDATA*
PDF;LOC;FONT
STOP
```

For Dynamic Bar Code Data Fields

The bold characters indicate the added parameters that control incrimination of barcode data.

```
BARCODE
TYPE;VSCAN;MAG;Hn;IBFn;L;DARK;SR;SC
PDF;LOC;FONT
STOP
```

NOTE: The actual base commands change depending on the barcode being defined.

PGL Barcode Commands and Output

This section shows samples of all PGL supported barcodes and what the output will be on your printer. For each barcode, the basic format is provided followed by a CREATE program and sample printout of barcode data. Where possible each barcode will print in a horizontal and vertical (VSCAN) orientation.

Code 39

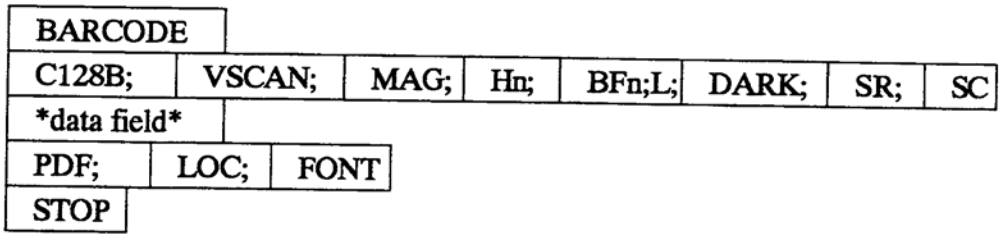
BARCODE							
C3/9CD;	VSCAN;	MAG;	Hn;	BFn;L;	DARK;	SR;	SC
data field							
PDF;	LOC;	FONT					
STOP							

Sample CREATE Code 39 Program

```
~CREATE;C39
BARCODE
C3/9;DAR;5;1
*CODE-39*
PDF
STOP
BARCODE
C3/9;VSCAN;H10;DARK;5;50
*CODE-39*
PDF
STOP
END
```

Output:

Code 128B

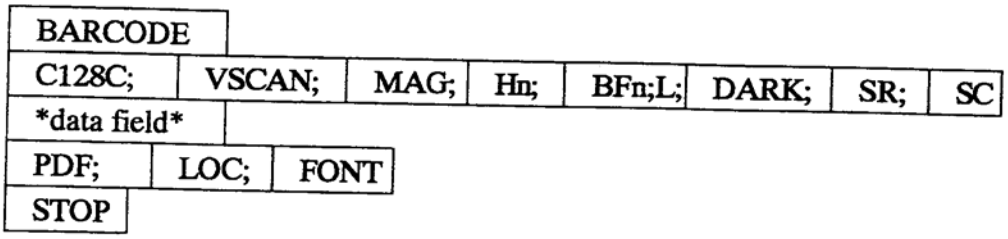


Sample CREATE Code 128B Program

```
~CREATE;128B
BARCODE
C128B;DARK;5;1
*CODE-128B*
PDF
STOP
BARCODE
C128B;VSCAN;H10;DARK;5;50
*CODE-128B*
PDF
STOP
END
```

Output:

Code 128C

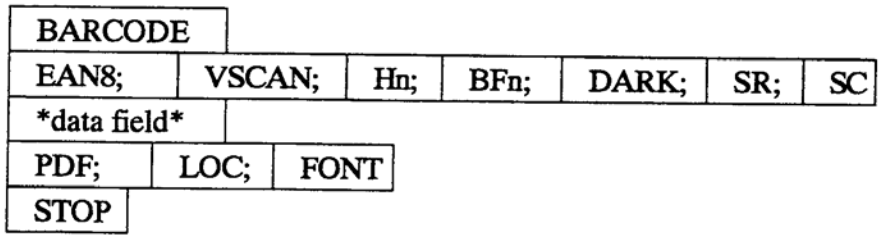


Sample CREATE Code 128C Program

```
~CREATE;128C
BARCODE
C128C;DARK;5;5
*1234567891*
PDF
STOP
BARCODE
C128C;VSCAN;H10;DARK;5;50
*1281281281*
PDF
STOP
END
```

Output:

EAN 8



Sample CREATE Code EAN 8 Program

```
~CREATE;CODEEAN8
BARCODE
EAN8;DARK;5;1
*1234567*
PDF
STOP
BARCODE
EAN8;VSCAN;H10;DARK;5;50
*1234567*
PDF
STOP
END
```

Output:

EAN 13

BARCODE						
EAN13;	VSCAN;	Hn;	BFn;	DARK;	SR;	SC
data field						
PDF;	LOC;	FONT				
STOP						

Sample CREATE Code EAN 13 Program

```
~CREATE;CODEEAN13
BARCODE EAN13;DARK;5;1
*1234567*
PDF
STOP
BARCODE
EAN13;VSCAN;H10;DARK;5;50
*1234567*
PDF
STOP
END
```

Output:

Interleaved 2/5

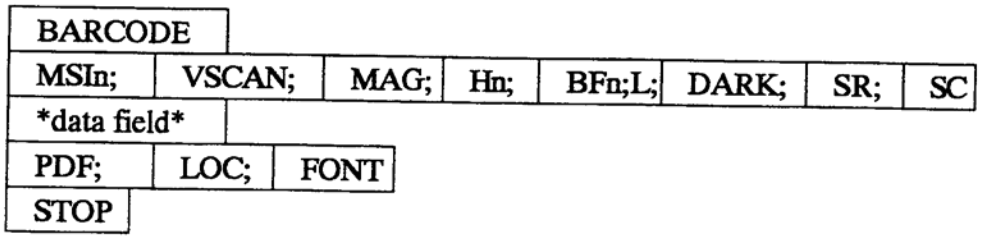
BARCODE							
I–2/5;	VSCAN;	MAG;	Hn;	BFn;L;	DARK;	SR;	SC
data field							
PDF;	LOC;	FONT					
STOP							

Sample CREATE Interleaved 2/5 Program

```
~CREATE;I2/5
BARCODE
I-2/5;DARK;5;1
*123456789*
PDF
STOP
BARCODE
I-2/5;VSCAN;H10;DARK;5;50
*123456789*
PDF
STOP
END
```

Output:

MSI

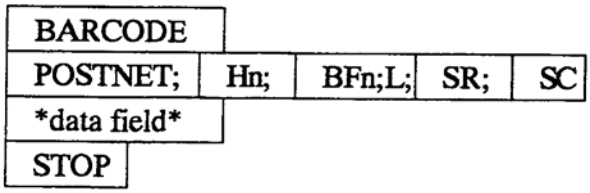


Sample CREATE MSI Program

```
~CREATE;MSIA
BARCODE
MSIA;DARK;5;1
*123456789*
PDF
STOP
BARCODE
MSIA;VSCAN;H10;DARK;5;50
*123456789*
PDF
STOP
END
```

Output:

POSTNET



POSTNET Samples

- ZIP + 4 Program

```
~CREATE;POSTNET
BARCODE
POSTNET;5;1
*123456789*
STOP
END
```

- Advaned Barcode Format

```
~CREATE;POSTNET
BARCODE
POSTNET;5;1
*12345678901*
STOP
END
```

UPC-A

BARCODE						
UPC-A;	VSCAN;	Hn;	BFn;	DARK;	SR;	SC
data field						
PDF;	LOC;	FONT				
STOP						

Sample CREATE Code UPCA Program

```
~CREATE;EAN13
BARCODE
UPC-A;DAR;5;1
*12345654321*
PDF
STOP
BARCODE
UPC-a;VSCAN;H10;DARK;5;50
*12345654321*
PDF
STOP
END
```

Output:

UPC-E

BARCODE						
UPC-E;	VSCAN;	Hn;	BFn;	DARK;	SR;	SC
data field						
PDF;	LOC;	FONT				
STOP						

Sample CREATE Code UPC-E Program

```
~CREATE;UPCE
BARCODE
UPC-E;DARK;5;1
*012345678900*
PDF
STOP
BARCODE
UPC-E;VSCAN;H10;DARK;5;50
*012345678900*
PDF
STOP
END
```

Output:

11 *Barcode and LCP Printing*

Introduction

This chapter describes all barcode and LCP sequences and control codes, regardless of your printer's special barcode and LCP (Large Character Printing) implementation.

NOTE: Be sure to observe the notes in this chapter regarding the special barcode/LCP Types implemented in your printer and which of the sequences described are not available as a result.

Selecting the Barcode Mode automatically selects the corresponding character set.

To print barcode or LCP characters, the following sets must be carried out in most cases:

- Activate barcode
- Activate barcode/unsecured mode (see "Secured/Unsecured Mode" on page 179)
- Transfer barcode and/or LCP header (see "Barcode Header" on page 181 and "LCP Header" on page 210)
- Calculate check number (barcode only, see "Check Number Calculation" on page 258)
- Transfer barcode and/or LCP data. (see "Data Formats of Barcode Types" on page 181 and "LCP Header" on page 210)
- Deactivate barcode (if necessary)

The following commands are used to print barcode and LCP characters: SUB Start Character Barcode header

DLE	Start Character LCP header
EM	Stop Character Barcode and LCP header
DC4	Barcode brackets (start and end characters for barcode data)
SI	LCP brackets (start and end characters for LCP data)
ESC P ... ESC \	Settings for Barcode and LCP

This may cause conflict with other emulations, since the above commands may have different functions in these emulations. For example:

SI	Condensed print
DC4	Reset expanded print
ESC P <n>	Proportional spacing ON/OFF
ESC P	Pica

In Barcode Mode, the barcode sequences have priority.

The barcode interpreter can be swited on or off with the MTPL sequences ESC [? 11 ~ (Barcode ON) and ESC [? 10 ~ (Barcode OFF). The typical transmission procedure should take place as follows:

- Barcode Interpreter ON
- Transmit mode specification (barcode header, LCP header, secured/ unsecured mode,...)

- Transmit barcode / LCP
- Barcode OFF

NOTE: The mode specifications are saved temporarily and must only be sent to the printer once. The specification remains valid until the printer is switched off.

Note the following explanatory information:

Every sequence description begins with a header in which the function and short form of the sequence of the barcode Types are listed, e.g.:

2/5 Matrix	Type A

The header is followed by the data structure in ASCII, hexadecimal and decimal representation with the necessary parameters, e.g.:

ASCII	DC4	start code	n ... n	stop code	DC4
hex.	14	start code	n ... n	stop code	14
dec.	20	start code	n ... n	stop code	20

The syntax for the parameters, the start, separate and stop code is represented as follows:

Table 42. Syntax Parameters

	Figures (n)	Start Code	Stop Code
ASCII	0 to 9	;	;
hex.	30 to 39	3A	3A
dec.	48 to 57	58	58

For the parameter (n), a distinction must be made between two types of representation:

- if the parameter is in pointed parentheses, the decimal value must be transmitted
- if the parameter is not in pointed parentheses, the ASCII value must be transmitted

Example:

Parameter representation: <n>, with n=0

to be transmitted: dec.0 (hex.00)

Parameter representation: n, with n=0

to be transmitted: ASCII "0" (hex.30, dec.48)

Character explanation and symbol descriptions

Information/important notes

[ ] Optional, must be not necessarily be transmitted

9 9-needle printer

24 24-needle printer

Special Features

If you did not receive the required Barcode control codes from the ASCII code table columns 0 and 1 from your computer, use a set of MTPL sequences to generate these control codes by printable ASCII characters. The Barcode programs will remain fully compatible in spite of the additional MTPL sequences.

The following control codes in Barcode strings can be substituted by the appropriate ANSI sequences:

Table 43. Barcode Strings Control Codes

Control Code	ANSI Sequence (CSI = Hex 9B or Hex 1B 5B or ESC[)	Example (Hex)
SUB	CSI 26 Space s	1B 5B 32 36 20 73
EM	CSI 25 Space s	1B 5B 32 35 20 73
DC4	CSI 20 Space s	1B 5B 32 30 20 73
DLE	CSI 16 Space s	1B 5B 31 36 20 73
SI	CSI 15 Space s	1B 5B 31 35 20 73
ANSI sequences without matching parameters will be ignored.		

NOTE: These features are not available on all printers with MTPL emulation. Special firmware may be necessary. Ask your service representative for details.

Secured/Unsecured Mode

Unsecured mode	**ESC PSC0 ESC **
Secured mode	**ESC PSC1 ESC **

Data Structure

ASCII	ESC	“P”	“S”	“C”	“0”	ESC	“\”	Unsecured
hex.	1B	50	53	43	30	1B	5C	Mode
dec.	27	80	83	67	48	27	92	
ASCII	ESC	“P”	“S”	“C”	“1”	ESC	“\”	Secured
hex.	1B	50	53	43	31	1B	5C	Mode
dec.	27	80	83	67	49	27	92	

Description

In Secured Mode, the amount of space the barcode or LCP character requires is secured. In each line, additional barcode and normal characters can be printed.

These additional characters are printed in the current line and in the following lines without influencing the barcode or LCP character. As a result, normal characters can be printed to the right or left of the barcode or LCP character in each line.

To guarantee successful barcode and LCP character printing, it is important to insert the correct paper feed commands, so that the paper feed is ensured to the end of the barcode and LCP height.

In Unsecured Mode, the paper feed necessary for barcode and LCP printing is automatic and is not possible to print mroe than one line with normal characters in the barcode and LCP line.

All characters in the mixed line are printed, so that the bottom edges are aligned in a straight line.

Example 1

```
10      REM LCP unsecured mode
20      LPRINT CHR$(27); "[?11~";
30      REM select unsecured mode
40      LPRINT CHR$(27); "PSC0"; CHR$(27); "\";
50      REM set character size to 5
60      LPRINT CHR$(16); "5"; CHR$(25);
70      LPRINT "Example for "
80      LPRINT CHR$(15); "LCP"; CHR$(15); " unsecured mode"
```

Example 2

```
10      REM LCP secured mode
20      LPRINT CHR$(27); "[?11~";
30      REM select secured mode
40      LPRINT CHR$(27); "PSC1"; CHR$(27); "\";
50      REM set character size to 5
60      LPRINT CHR$(16); "5"; CHR$(25);
70      LPRINT "This is "; CHR$(15); "LCP"; CHR$(15);
80      LPRINT " an ex-"
90      LPRINT "ample of an"
100     LPRINT "expres- sion"
110     LPRINT "in the secured mode"
```

Barcode Printing

Barcode Header

Before the data, which contains the barcode information, are transmitted to the printer, the barcode header must be sent. Otherwise the standard parameter values are used (see “Header Format” below). In the header (the printing parameters), the barcode size and type are defined. This header only needs to be transferred once, unless settings are to be changed or the printer has been turned off.

Header Format

Format:	SUB [F] a [n] [;xyz] EM
[ ]	Specification is optional
x, y	unregarded at EAN/UPC-Barcode! For Code 128 and EAN 128 (Type S+T) only the X parameter is valid. This is automatically used for the Y parameter. The Z parameter is not evaluated.

Character Meaning:

SUB	(hex.1A, dec.26)	Start header
F		Print Feature (see section “Barcode Print Feature F” to select F codes, page 182)
a	ASCII a = “A”…”S”	Barcode Types (see section “Barcode Types, page 183)
n	ASCII n = “0”…”90”	Barcode height in n/6 inch. At n = “0” the barcode height equals 1/12 inch.
;	ASCII	Separation character
x	ASCII x = “0”…”3”	Width of the narrow bar (see ‘Barcode Width” on page183)
y	ASCII y = “0”…”3”	Width of the narrow space (see ‘Barcode Width” on page 183)
z	ASCII z = “0”…”3”	Ratio of wide to narrow (see “Barcode Width” on page 183)
EM	(hex.19, dec.25)	End of header

NOTE: For encoding ASCII values to decimal or hexadecimal values refer to “Character Sets” on page 247.

The default parameter values are the following:

- Unsecured mode (see section "Secured/Unsecured mode")
- HRI OFF, Normal Print, Double Pass (F = SP)
- Barcode Type 2/5 matrix (a = "A")
- Barcode height 1/6 inch (n = "1")
- Narrow bar (x = "0")
- Narrow space (y = "0")
- Ratio of wide to narrow 2 to 1 (z = "0")

When only parts of the header are to be changed, copy the header up to the parameter which must be changed, and then close the header with the end- of-header character. If a header error was detected the total previous features are still active.

The "Barcode brackets" (hex.14, dec.20), initiate and terminate the printing of the barcodes.

If the printer is switched OFFLINE, all defined barcodes are printed out completely.

NOTE: The barcode remains resident in the background and can be activated again by the barcode bracket. Text justification and centering are both permitted. With activated barcodes these functions are not carried out, since they lead to conflicts with the barcodes.

Transparent Barcode/LCP Commands

Following control codes in Barcode strings can be substituted by the appropriate MTPL sequences (see “Special Features” on page 179).

Table 44. MTPL Sequences Barcode Strings Substitution

Control Code	ANSI Sequence (CSI = Hex 9B or ESC [ )
SUB	CSI 26 Space s
EM	CSI 25 Space s
DC4	CSI 20 Space s
DLE	CSI 16 Space s
SI	CSI 15 Space s
MTPL sequences without matching parameters will be ignored.	

These sequences may only be used with activated barcodes.

Barcode Print Feature F for Selection of F-Code

HRI or normal/compressed as well as single or double pass is switched via character F according to the following table.

ASCII Char.	Hex-Value	HRI [2]		Print		Pass [3]		Direction [2]	
		On	Off	Normal	Compr.	Double	Single	Unidir.	Bidir.
SP	20	–	x	x	–	x	–	x	–
!	21	–	x	x	–	x	–	–	x
"	22	x	–	x	–	x	–	x	–
#	23	x	–	x	–	x	–	–	x
$	24	–	x	–	x	x	–	x	–
%	25	–	x	–	x	x	–	–	x
&	26	x	–	–	x	x	–	x	–
‘	27	x	–	–	x	x	–	–	x
(	28	–	x	x	–	–	x	x	–
)	29	–	x	x	–	–	x	–	x
*	2A	x	–	x	–	–	x	x	–
+	2B	x	–	x	–	–	x	–	x
,	2C	–	x	–	x	–	x	x	–
-	2D	–	x	–	x	–	x	–	x
.	2E	x	–	–	x	–	x	x	–
/	2F	x	–	–	x	–	x	–	x
0 [1]	30	–	x	x	–	x	–	x	–
1 [1]	31	x	–	x	–	x	–	x	–

1) It is recommended to avoid using ASCII Characters 0 and 1 when possible. They are reserved for future functions.
2) Human Readable Index
3) Whether the printer performs “Double Pass” with two physical print passes or special print modes (i.e. emphasized), depends on the used printer type.

Barcode Types

A = 2/5 matrix (default)

B = 2/5 industrial

C = 2/5 interleaved

D = Code 11

E = Code BCD matrix

F = Code 39

G = Codabar

H = EAN 8 with HRI

I = EAN 8 without HRI

J = 2/5 matrix (default)

K = EAN 13 with HRI

L = EAN 13 without HRI

M = MSI/modified Plessey

N = UPC A with HRI

O = UPC A without HRI

P = UPC E with HRI

Q = UPC E without HRI

R = Delta Distance (IBM)

S = Code 128

T = EAN 128

All commercial barcodes (for labeling systems) of the H, I, K, L, N, O, P, Q Types can be extended using the barcodes Add-On 2 or Add-On 5 (see “Add-On Barcodes” on page 206).

HRI

HRI = Human Readable Index

HRI characters are centered if enough space is left. If the barcode printout is smaller than the HRI character field, smaller character density (CPI) is used. Start and stop codes are not printed as HRI; a space character (SP) will be stored.

Barcode Width

By specifying an ASCII value from 0 to 3, the barcode width can be defined. This allows adaption to the scanner specifications, particularly for long-range scanners.

Table 1	Header Para-meter	Normal			Compressed[1])		
		9	24	older printer types (e.g.MT230)	9	24	older printer types (e.g.MT230)
Width of the narrow bar	x = 0	0.53 mm	0.54 mm	0.48 mm	0.32 mm	0.33 mm	0.27 mm 3)
	x = 1	0.74 mm	0.67 mm	0.69 mm	0.53 mm	0.43 mm	0.37 mm 3)
	x = 2	1.16 mm	1.09 mm	1.16 mm	0.74 mm	0.65 mm	0.59 mm 3)
	x = 3	1.38 mm	1.30 mm	1.33 mm	0.95 mm	0.88 mm	0.90 mm 3)
Width of the narrow space	y = 0	0.53 mm	0.54 mm	0.48 mm	0.32 mm	0.33 mm	0.27 mm 3)
	y = 1	0.74 mm	0.67 mm	0.69 mm	0.53 mm	0.43 mm	0.37 mm 3)
	y = 2	1.16 mm	1.09 mm	1.16 mm	0.74 mm	0.65 mm	0.59 mm 3)
	y = 3	1.38 mm	1.30 mm	1.33 mm	0.95 mm	0.88 mm	0.90 mm 3)
Enlargement factor	z = 0			2.0 : 1			2.0 : 1 2)
	z = 1			2.5 : 1			2.5 : 1 2)
	z = 2			3.0 : 1			3.0 : 1 2)
	z = 3			3.5 : 1			3.5 : 1 2)

EAN/UPC Barcode

(X, Y = unregarded):

Table 2	Header Parameter	Normal	Compressed [1])
Enlargement factor	z = 0	1.95 : 1	1.30 : 1
	z = 1	1.60 : 1	0.95 : 1

1) These values are true, if “Compressed Print” is selected in the menu (see print feature [F]).

2) It is recommended to set the bar width equal to the space width (x=y).

3) Printer-dependent reference value.

All values are only valid when a new color ribbon is used. The values change depending on the degree of wear:

- **narrow bar:** approx. -0.05 mm
- **narrow space**: approx. +0.05 mm.

Code EAN 128 and EAN/UPC are based on module widths. Only the X parameter is valid for this type. This parameter is used for the Y parameter (narrow bar width). The Z parameter has no meaning. In combination with the normal/condensed feature, 8 widths result (see Table 1 on page 183).

Error Code

Wrong characters in a control code or in a barcode test (e.g. an undefined character in a certain barcode type) cause the barcode error sign to print.

Data Formats of Barcode Types

Code 2/5 Matrix

Syntax

	Figures (n)	Start Code	Stop Code
ASCII	“0” to “9”	“:”	“:”
hex.	30 to 39	3A	3A
dec.	48 to 57	58	58

Data Structure

ASCII	DC4	Start Code	n…n	Stop code	DC4
hex.	14	Start code	n…n	Stop code	14
dec.	20	Start code	n…n	Stop code	20

Example

```
10 REM code 2/5 matrix barcode
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" A3;111";CHR$(25);
40 LPRINT CHR$(20);":123:";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code 2/5 Industrial

Syntax

	Figures (n)	Start Code	Stop Code
ASCII	“0” to “9”	“:”, “<“ or “>”	“:”, “=” or “?”
hex.	30 to 39	3A, 3C, or 3E	3B, 3D, or 3F
dec.	48 to 57	58, 60, or 62	59, 61, or 63

Data Structure

ASCII	DC4	Start Code	n…n	Stop code	DC4
hex.	14	Start code	n…n	Stop code	14
dec.	20	Start code	n…n	Stop code	20

Example

```
10 REM code 2/5 industrial barcode
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" B3;111";CHR$(25);
40 LPRINT CHR$(20);":123;";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code 2/5 Interleaved

Syntax

	Figures (n)	Start Code	Stop Code
ASCII	“0” to “9”	“:” or “<“	“:” or “=”
hex.	30 to 39	3A or 3C	3B or 3D
dec.	48 to 57	58 or 60	59 or 61

Data Structure

ASCII	DC4	Start Code	n…n	Stop code	DC4
hex.	14	Start code	n…n	Stop code	14
dec.	20	Start code	n…n	Stop code	20

Example

```
10 REM code 2/5 interleaved barcode
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" C3;111";CHR$(25);
40 LPRINT CHR$(20);":123;";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

NOTE: Due to the “interleaved mechanism”, data stream (n...n) should only be transferred with an even number of digits e.g., not 398 but 0398

If an odd number of digits are transferred the printer adds a leading zero to the printed barcode.

Code 11

Syntax

	Figures/Characters (n)	Start Code	Stop Code
ASCII	“0” to “9” and “-”	“:”	“:”
hex.	30 to 39 and 2D	3A	3A
dec.	48 to 57 and 45	58	58

Data Structure

ASCII	DC4	Start Code	n…n	Stop code	DC4
hex.	14	Start code	n…n	Stop code	14
dec.	20	Start code	n…n	Stop code	20

Example

```
10 REM code 11 barcode
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" D3;111";CHR$(25);
40 LPRINT CHR$(20);":123:";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code BCD Matrix

Syntax

	Figures (n)	Start Code	Stop Code
ASCII	“0” to “9”	“:”	“:”
hex.	30 to 39	3A	3A
dec.	48 to 57	58	58

Data Structure

ASCII	DC4	Start Code	n…n	Stop code	DC4
hex.	14	Start code	n…n	Stop code	14
dec.	20	Start code	n…n	Stop code	20

Example

```
10 REM BCD-matrix-code barcode
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" E3;111";CHR$(25);
40 LPRINT CHR$(20);":123:";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code 39

Syntax

	Figures/Characters (n)	Start Code	Stop Code
		Not fixed, recommended:	Not fixed, recommended:
ASCII	"0" to "9", "A" to "Z" and "$", "%", "*", "+", "-", ".", "/"	“*”	“*”
hex.	30 to 39, 41 to 5A and 24, 25, 2A, 2B, 2D, 2E, 2F	2A	2A
dec.	48 to 57, 65 to 90 and 36, 37, 42, 43, 45, 46, 47	42	42

Data Structure

ASCII	DC4	Start Code	n…n	Stop code	DC4
hex.	14	Start code	n…n	Stop code	14
dec.	20	Start code	n…n	Stop code	20

Example

```
10 REM code 39 barcode
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" F3;111";CHR$(25);
40 LPRINT CHR$(20);"*123*";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

CODABAR

Syntax

	Figures/Characters (n)	Start Code	Stop Code
		Not fixed, recommended:	Not fixed, recommended:
ASCII	"0" to "9", "A" to "D" and "$", "+", "-", ".", "/", ":"	"a" to "e" and "n", "t", "*"	"a" to "e" and "n", "t", "*"
hex.	30 to 39 and 24, 2B, 2D, 2E, 2F, 3A	61 to 65 and 6E, 74, 2A	61 to 65 and 6E, 74, 2A
dec.	48 to 57 and 36, 43, 45, 46, 47, 58	97 to 101 and 110, 116,42	97 to 101 and 110, 116,42

Data Structure

ASCII	DC4	Start Code	n…n	Stop code	DC4
hex.	14	Start code	n…n	Stop code	14
dec.	20	Start code	n…n	Stop code	20

Example

```
10 REM codabar barcode
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" G3;111";CHR$(25);
40 LPRINT CHR$(20);"*123*";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code EAN 8 with HRI

Syntax

	Figures (n)	Check Number (c)	Start Code	Separation Code	Stop Code
ASCII	"0" to "9"	Calculation see “Check Number Calculation” on page 207.	":"	":"	":"
hex.	30 to 39		3A	3A	3A
dec.	48 to 57		58	58	58

Data Structure

ASCII DC4 Start code nnnn Separation code nnnc Stop code DC4

hex. 14 Start code nnnn Separation code nnnc Stop code 14

dec. 20 Start code nnnn Separation code nnnc Stop code 20

Example

```
10 REM code EAN 8 with HRI
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" H3;111";CHR$(25);
40 LPRINT CHR$(20);":0123:4567:";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code EAN 8 without HRI

Syntax

	Figures (n)	Check Number (c)	Start Code	Separation Code	Stop Code
ASCII	"0" to "9"	Calculation see “Check Number Calculation” on page 207.	":"	":"	":"
hex.	30 to 39		3A	3A	3A
dec.	48 to 57		58	58	58

Data Structure

ASCII DC4 Start code nnnn Separation code nnnc Stop code DC4

hex. 14 Start code nnnn Separation code nnnc Stop code 14

dec. 20 Start code nnnn Separation code nnnc Stop code 20

Example

```
10 REM code EAN 8 without HRI
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" I3;111";CHR$(25);
40 LPRINT CHR$(20);":0123:4567:";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code EAN 13 with HRI

Syntax

	Figures (n)	Check Number (c)	Start Code	Separation Code	Stop Code
ASCII	"0" to "9"	Calculation see “Check Number Calculation” on page 207.	":"	":"	":"
hex.	30 to 39		3A	3A	3A
dec.	48 to 57		58	58	58

Data Structure

ASCII DC4 Start code nnnnnnn Separation code nnnnnc Stop code DC4

hex. 14 Start code nnnnnnn Separation code nnnnnc Stop code 14

dec. 20 Start code nnnnnnn Separation code nnnnnc Stop code 20

Example

```
10 REM code EAN 13 with HRI
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" K3;111";CHR$(25);
40 LPRINT CHR$(20);":0123456:789012:";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code EAN 13 without HRI

Syntax

	Figures (n)	Check Number (c)	Start Code	Separation Code	Stop Code
ASCII	"0" to "9"	Calculation see “Check Number Calculation” on page 207.	":"	":"	":"
hex.	30 to 39		3A	3A	3A
dec.	48 to 57		58	58	58

Data Structure

ASCII DC4 Start code nnnnnnn Separation code nnnnnc Stop code DC4

hex. 14 Start code nnnnnnn Separation code nnnnnc Stop code 14

dec. 20 Start code nnnnnnn Separation code nnnnnc Stop code 20

Example

```
10 REM code EAN 13 without HRI
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" L3;111";CHR$(25);
40 LPRINT CHR$(20);":0123456:789012:";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code MSI/Modify Plessey

Syntax

	Figures (n)	Start Code	Stop Code
ASCII	“0” to “9”	“:”	“:”
hex.	30 to 39	3A	3A
dec.	48 to 57	58	58

Data Structure

ASCII	DC4	Start Code	n…n	Stop code	DC4
hex.	14	Start code	n…n	Stop code	14
dec.	20	Start code	n…n	Stop code	20

Example

```
10 REM code MSI/plessey modified
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" M3;111";CHR$(25);
40 LPRINT CHR$(20);":0123;";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code UPC A with HRI

Syntax

	Figures (n)	Check Number (c)	Start Code	Separation Code	Stop Code
ASCII	"0" to "9"	Calculation see “Check Number Calculation” on page 207.	":"	":"	":"
hex.	30 to 39		3A	3A	3A
dec.	48 to 57		58	58	58

Data Structure

ASCII DC4 Start code nnnnnn Separation code nnnnnc Stop code DC4

hex. 14 Start code nnnnnn Separation code nnnnnc Stop code 14

dec. 20 Start code nnnnnn Separation code nnnnnc Stop code 20

Example

```
10 REM code UPC A with HRI
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" N3;111";CHR$(25);
40 LPRINT CHR$(20);":012345:678901:";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code UPC A without HRI

Syntax

	Figures (n)	Check Number (c)	Start Code	Separation Code	Stop Code
ASCII	"0" to "9"	Calculation see “Check Number Calculation” on page 207.	":"	":"	":"
hex.	30 to 39		3A	3A	3A
dec.	48 to 57		58	58	58

Data Structure

ASCII DC4 Start code nnnnnn Separation code nnnnnc Stop code DC4

hex. 14 Start code nnnnnn Separation code nnnnnc Stop code 14

dec. 20 Start code nnnnnn Separation code nnnnnc Stop code 20

Example

```
10 REM code UPC A without HRI
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" O3;111";CHR$(25);
40 LPRINT CHR$(20);":012345:678901:";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code UPC E with HRI

Syntax

	Figures (n)	Check Number (c)	Start Code	Stop Code
ASCII	"0" to "9"	Calculation see “Check Number Calculation” on page 207.	":"	":"
hex.	30 to 39		3A	3A
dec.	48 to 57		58	58

Data Structure

ASCII	DC4	Start Code	nnnnnnnc	Stop code	DC4
hex.	14	Start code	nnnnnnnc	Stop code	14
dec.	20	Start code	nnnnnnnc	Stop code	20

Example

```
10 REM code UPC E with HRI
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" P3;111";CHR$(25);
40 LPRINT CHR$(20);":01234567:";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code UPC E without HRI

Syntax

	Figures (n)	Check Number (c)	Start Code	Stop Code
ASCII	"0" to "9"	Calculation see “Check Number Calculation” on page 207.	":"	":"
hex.	30 to 39		3A	3A
dec.	48 to 57		58	58

Data Structure

ASCII	DC4	Start Code	nnnnnnnc	Stop code	DC4
hex.	14	Start code	nnnnnnnc	Stop code	14
dec.	20	Start code	nnnnnnnc	Stop code	20

Example

```
10 REM code UPC E without HRI
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" Q3;111";CHR$(25);
40 LPRINT CHR$(20);":01234567:";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code Delta Distance (IBM)

Syntax

	Figures/Characters (n)	Start Code	Stop Code
ASCII	"0" to "9" and "A" to "F"	“F”	“D”
hex.	30 to 39 and 41 to 46	46	44
dec.	48 to 57 and 65 to 70	70	68

Data Structure

ASCII	DC4	Start Code	n…n	Stop code	DC4
hex.	14	Start code	n…n	Stop code	14
dec.	20	Start code	n…n	Stop code	20

Example

```
10 REM code delta distance (IBM) without HRI
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" R3;111";CHR$(25);
40 LPRINT CHR$(20);"F0123D";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Code 128

Syntax

	Figures/Characters (n)	Start Code	Stop Code
ASCII	Table 45 on page 203.	None	None
hex.			
dec.			

Data Structure

ASCII	DC4	Start Code	n…n	Stop code	DC4
hex.	14	Start code	n…n	Stop code	14
dec.	20	Start code	n…n	Stop code	20

Description

While many barcode styles are limited to numbers, Code 128 can encode numbers, letters, and other symbols commonly found on keyboards, such as

@, # and %. The Code 128 barcode is similar to three barcodes in one. The three barcode styles are called Code A, Code B, and Code C. Each barcode style is designed to encode certain types of data in the most compact way.

- Code A encodes uppercase alpha, numeric and control codes.
- Code B encodes upper and lower case alpha and numeric codes.
- Code C encodes digits in pairs.

A single Code 128 barcode may consist of all three Code 128 barcode styles A, B and C. The printer chooses the styles which result in the most compact Code 128 barcode for the data to be encoded. Do not be concerned about choosing the correct barcode styles as this is done automatically.

The ">" (greater than) symbol is a special character prefix. If a character with decimal value less then 32 needs to be encoded (i.e., a control character), then send a ">" followed by the character that is decimal 64 higher then the control code character.

NOTE: Control characters conflict with DC4, SUB. These characters are encoded by using the ">" sign following a number that is 64 dec. higher than the control character.

Example

If you need to print a Carriage Return (hex.0D, dec.13), as part of a barcode, send ">M" (13 + 64 = 77, see character set table to verify that decimal 77 equals the capital letter "M"). This substitution process can be carried out throughout the range of decimal values, i.e., ">d" (decimal 100) equates to the dollar symbol (decimal 36). If the greater-than symbol must be transmitted, send ">0" (the 0 will not be part of the printed data).

The Code 128 barcode style selection can also be done manually. This is done by adding a style selection character to the beginning of the barcode data. Choose code A, B, or C from the table of special characters below:

Table 45. Special Characters

Special Character	Code 128
“>0	30 (“<“)
“>1”	95
“>2”	96
“>3”	97
“>4”	98
“>5”	99 (Code C)
“>6”	100 (Code B)
“>7”	101 (Code A)
“>8”	102

When a style selection has been made, character data will be translated from the selected code style to the Code 128 representation. See Figure 7 on page 204.

Wert	Code A	Code B	Code C
0	Space	Space	0
1	!	!	1
2	"	"	2
3	#	#	3
4	$	$	4
5	%	%	5
6	&	&	6
7	‘	‘	7
8	(	(	8
9	)	)	9
10	*	*	10
11	+	+	11
12	‘	‘	12
13	hyphen	hyphen	13
14	period	period	14
15	/	/	15
16	0	0	16
17	1	1	17
18	2	2	18
19	3	3	19
20	4	4	20
21	5	5	21
22	6	6	22
23	7	7	23
24	8	8	24
25	9	9	25
26	:	:	26
27	;	;	27
28	<	<	28
29	=	=	29
30	>	>	30
31	?	?	31
32	@	@	32
33	A	A	33
34	B	B	34
35	C	C	35

Wert	Code A	Code B	Code C
36	D	D	36
37	E	E	37
38	F	F	38
39	G	G	39
40	H	H	40
41	I	I	41
42	J	J	42
43	K	K	43
44	L	L	44
45	M	M	45
46	N	N	46
47	O	O	47
48	P	P	48
49	Q	Q	49
50	R	R	50
51	S	S	51
52	T	T	52
53	U	U	53
54	V	V	54
55	W	W	55
56	X	X	56
57	Y	Y	57
58	Z	Z	58
59	[	[	59
60	\	\	60
61	]	]	61
62	^	^	62
63	_	_	63
64	NUL	‘	64
65	SOH	a	65
66	STX	b	66
67	ETX	c	67
68	EOT	d	68
69	ENQ	e	69
70	ACK	f	70
71	BEL	g	71

Figure 7. Code 128 Translation Table

Wert	Code A	Code B	Code C
72	BS	h	72
73	HT	i	73
74	LF	j	74
75	VT	k	75
76	FF	l	76
77	CR	m	77
78	SO	n	78
79	SI	o	79
80	DLE	p	80
81	DC1	q	81
82	DC2	r	82
83	DC3	s	83
84	DC4	t	84
85	NAK	u	85
86	SYN	v	86
87	ETB	w	87
88	CAN	x	88
89	EM	y	89
90	SUB	z	90
91	ESC	{	91
92	FS	|	92
93	GS	}	93
94	RS	~	94
95	US	DEL	95
96	FNC3	FNC3	96
97	FNC2	FNC2	97
98	SHIFT	SHIFT	98
99	CODE	CODE	99
100	CODE	FUNC4	CODE
101	FNC4	CODE	CODE
102	FNC1	FNC1	FNC1

Figure 8. Code 128 Translation Table (continued)

Codes 96 through 102 do not have corresponding ASCII character translations; these may be encoded using the special character table above

Example

```
10 REM code 128
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" S3;111";CHR$(25);
40 LPRINT CHR$(20);"ABCD0123";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

EAN 128

Syntax

	Figures/Characters (n)	Start Code	Stop Code
ASCII	Table 45 on page 203.	None	None
hex.			
dec.			

Data Structure

ASCII	DC4	n…n	DC4
hex.	14	n…n	14
dec.	20	n…n	20

Description

The only difference between both types of barcodes is the initial sequence. Code 128 starts with Code A, Code B or Code C followed by character information. Barcode EAN 128 requires the code FNC1 between Startcode and character information. The Code 128 translation table remains valid. The checksum of EAN 128 is calculated using Code 128 algorithm.

Example

```
10 LPRINT CHR$(27);"[?11~";
20 LPRINT CHR$(26);" T3;111";CHR$(25);
30 LPRINT CHR$(20);"1234ABCD";CHR$(20);
40 LPRINT CHR$(27);"[?10~"
```

Add-On Barcodes

UPC and EAN barcodes (commercial barcodes) can be extended with Add- On barcodes.

The following barcodes can be extended with Add-On barcodes: Barcode Types H, I, K, L, N, O, P and Q

In the data formats the numbers are generally given in the sequence they are printed from left to right in the barcode, i.e. in the case of the EAN13 code, the 13. figure is transferred first and the 1. figure last. For the formats for the Add- On barcodes, the printer expects the check number as the first figure. This is not printed in the barcode. The following numbers (2 or 5) are printed from left to right in Add-On barcode.

Example 1

EAN13 barcode with HRI and Add On 2 extention

Format: DC4:nnnnnnn:nnnnnn:cnn:DC4

```
10 REM code EAN 13 with add-on-2 extention
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" K3;111";CHR$(25);
40 LPRINT CHR$(20);":0123456:789012:012:";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Example 2

EAN13 barcode with HRI and Add On 5 extention

Format: DC4:nnnnnnn:nnnnnn:cnnnnn:DC4

```
10 REM code EAN 13 with add-on-5 extention
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(26);" K3;111";CHR$(25);
40 LPRINT CHR$(20);":0123456:789012:012345:";CHR$(20);
50 LPRINT CHR$(27);"[?10~"
```

Check Number Calculation

The following barcode types must be given a check number for transfer to the printer:

- Type H and I (EAN 8)
- Type K and L (EAN 13)
- Type N and O (UPC A)
- Type P and Q (UPC E)

The check number (c) is transferred after the barcode information (n). For the "EAN" type, this is printed as the last figure in the HRI data line. For the "UPC" type, the check number is not printed in the HRI data

The commercial barcodes can be extended with the following user-specific additional barcodes (see “Add-On Barcodes” on page 206).

- Add-On 2 barcode
- Add-On 5 barcode

Here the check number (c) is transferred before the barcode information (n).

EAN 13

12 numbers are transferred for the EAN 13 code. The 13. digit represents the check number. The even figures of the information have a factor of "3", the uneven figures a factor of "1". Counting begins from the left and the resulting cross sum is divided by 10. The remaining figure is subtracted from the modul (10). If the remainder is "0", the check number is also "0".

Example

Information	4	0	1	2	3	4	5	6	7	8	9	0	1
Factor	1	3	1	3	1	3	1	3	1	3	1	3	
Product	4	+ 0	+ 1	+ 6	+ 3	+ 12	+ 5	+ 18	+ 7	+ 24	+ 9	+ 0	
Cross sum	89												

10 (modul) - 9 (remainder) = 1 (check number)

For remainder “0” the check number is also “0”.

EAN 8

Seven numbers are transferred for the EAN 8 code. The 8. digit represents the check number. The even figures of the information have a factor of "3", the uneven figures a factor of "1". Counting begins from the left and the resulting cross sum is divided by 10. The remaining figure is subtracted from the modul (10). If the remainder is "0", the check number is also "0".

Example

Information	4	0	1	2	3	4	6	2
Factor	3	1	3	1	3	1	3	
Product	12	+ 0	+ 3	+ 2	+ 9	+ 4	+ 18	
Cross sum	48							

10 (modul) - 8 (remainder) = 2 (check number)

For remainder “0” the check number is also “0”.

UPCA

11 numbers are transferred for the UPCA code. The 12. digit represents the check number. The even figures of the information have a factor of "3", the uneven figures a factor of "1". Counting begins from the left. The resulting cross sum is divided by 10. The remaining figure is subtracted from the modul (10). If the remainder is "0", the check number is also "0".

Example

Information	4	0	1	2	3	4	5	6	7	8	9	3
Factor	3	1	3	1	3	1	3	1	3	1	3	
Product	12 +	0	+ 3	+ 2	+ 9	+ 4	+ 15	+ 6	+ 21	+ 8	+ 27	
Cross sum	107											

10 (modul) - 7 (remainder) = 3 (check number)

For remainder “0” the check number is also “0”.

UPCE

For the UPCE barcode type, seven numbers are transmitted. The 8. digit is the check number. The uneven figures of the information have a factor of "3", the even figures a factor of "1". Counting begins from the left and the resulting cross sum is divided by 10. The remaining integer value is subtracted from the modul (10). The result is transmitted as the check number (c).

Example

Information	0	1	2	3	4	5	6	5
Factor	3	1	3	1	3	1	3	
Product	0 +	1 +	6 +	3 +	12 +	5 +	18	
Cross sum	45							

10 (modul) - 5 (remainder) = 5 (check number)

For remainder “0” the check number is also “0”.

Add-On 5

Six numbers are transmitted: check number (c) + 5 information items (n). The uneven figures of the information have a factor of "3", the even figures a factor of "1". Counting begins from the right. The resulting cross sum is divided by

10. The remaining integer value is subtracted from the modul (10). The result is transmitted as the check number (c).

Example

Information	3	8	6	1	0	4
Factor		3	9	3	3	3
Product		24 +	54 +	3 +	0 +	12
Cross sum		93				

3 (remainder) = 3 (check number)

10 (modul) - 5 (remainder) = 5 (check number)

The check number is not printed in the clear data line (HRI).

Add-On 2

Three numbers are transmitted: check number (c) and 2 information (n). The check number results from the remaining integer value of modul (4).

Example 1

Information	0	0	4
Remainder (0)		0	

Example 2

Information	2	0	6
Remainder (2)		2	

Example 3

Information	3	9	9
Remainder (3)	└───	3	

Example 4

Information	1	0	9
Remainder (1)	└───	1	

NOTE:

NOTE: The check number always lies between “0” and “3”; it is not printed in the HRI line.

US Postnet Barcode

Data Structure

ASCII	ESC	“[“	“1”	SP	“p”
hex	1B	5B	31	20	70
dec	27	91	49	32	112

Description

After receiving this sequence numeric characters from 0 (hex. 30) to

9 (hex. 39) are interpreted as barcode figures. Other Alpha characters are ignored. Control characters from hex.00 (dec.9) to hex.1F (dec.31) terminates this barcode mode.

You may use this sequence in every emulation. The printer can print barcode in NLQ and LQ. If draft print quality (DPQ) is selected the printout is performed in NLQ.

NOTE: The character (e.g. CR = carriage return, hex. 0D), which terminates barcode mode will neither be printed nor carried out. A tab command (hex.09, dec.9) is carried out in this barcode.

```
10 REM US postnet barcode
20 LPRINT CHR$(27);"[?11~";
30 LPRINT CHR$(27);"[1 p";"1234567";CHR$(13);
40 LPRINT CHR$(27);"[?10~"
```

Large Character Printing (LCP)

Before you can transfer the LCP data (Large Character Printing) data to the printer, the LCP header must be sent. The LCP header is made up of a series of max. five characters. In the header, the printing parameters and the LCP character size are defined. This header only needs to be transferred once, unless settings are to be changed or the printer has been turned off.

LCP Header

Format: DLE [!] n EM

[ ] optional, does not need to be transmitted

Character meaning:

DLE	(hex. 10, dec. 16)	start character
!	ASCII	bidirectional printing (if transferred)
n	ASCII n = “2”…”99”	enlargement factor
EM	(hex. 19, dec.25)	end character

With the help of the header, the enlargement factor is fixed to the original size. A character in LCP mode with the size n occupies a horizontal space for n normal character, depending on the selected character distance. At a character distance of 10 characters/inch and an enlargement factor of 6, the LCP font width is 6/10 inch.

The height of a LCP character is n times 1/12 inch. It is independent of the selected line spacing. The LCP characters are printed, so that their lower edges lie flush with the next available ground line. The result is that only the upper edges of characters with an uneven enlargement factor can lie flush with the line.

LCP Data

The existence of a LCP header does not mean that all subsequent characters are printed in LCP size. The LCP mode must begin and end with the LCP brackets SI (hex.0F, dec.15). All characters inside these brackets are printed as LCP characters. The LCP mode is ended by all characters from hex.00 to hex.1F.

LCP characters can be printed with character densities 10, 12, 15, 17.1 and 20 cpi. If there is a LCP line overflow, the printing procedure is automatically started. All characters which caused the overflow are then printed as normal characters.

Refer to the section "Secured/Unsecured Mode" at the front of the Manual for detailed description of the secured/unsecured mode.

Data Structure

ASCII	SI	n…n	SI
hex.	0F	n…n	0F
dec.	15	n…n	15

n = all printable characters (> hex.1F, dec.31)

Example

```
REM Example Character Densities
OPEN "lpt1:" FOR RANDOM AS #1
WIDTH #1, 255
PRINT #1, CHR$(27); "[?11~";
PRINT #1, CHR$(16); "4"; CHR$(25);
PRINT #1, CHR$(27); "[6w"; :REM 15CPI
PRINT #1, CHR$(15); "LCP ";CHR$(15);
PRINT #1, CHR$(27); "[5w"; :REM 12CPI
PRINT #1, CHR$(15); "LCP ";CHR$(15);
PRINT #1, CHR$(27); "[4w"; :REM 10CPI
PRINT #1, CHR$(15); "LCP ";CHR$(15);
PRINT #1, STRING$ (5, 10);
END
```

LCP LCP LCP

LCP Character Set US-ASCII LCP Character Set German

Data Structure

US-ASCII Character Set

ASCII	ESC	“P”	“L“	“S”	“0”	“1”	ESC	“\”
hex	1B	50	4C	53	30	31	1B	5C
dec	27	80	76	83	48	49	27	92

German Character Set

ASCII	ESC	“P”	“L“	“S”	“0”	“2”	ESC	“\”
hex	1B	50	4C	53	30	32	1B	5C
dec	27	80	76	83	48	50	27	92

Example 1

```
10    LPRINT "Example for different LCP character sizes"
20    LPRINT CHR$(27); "[?11~";
30    LPRINT CHR$(27); "PSC0"; CHR$(27); "\";
40    REM set character size to 3
50    LPRINT CHR$(16); "3"; CHR$(25);
60    LPRINT CHR$(15); "3"; CHR$(15);
70    REM set character size to 4
80    LPRINT CHR$(16); "4"; CHR$(25);
90    LPRINT CHR$(15); "4"; CHR$(15);
100   REM set character size to 5
110   LPRINT CHR$(16); "5"; CHR$(25);
120   LPRINT CHR$(15); "5"; CHR$(15);
130   REM set character size to 6
140   LPRINT CHR$(16); "6"; CHR$(25);
150   LPRINT CHR$(15); "6"; CHR$(15);
160   LPRINT CHR$(27); "[?10~"
170
END
```

Example for Different LCP Character Sizes

3456

Example 2

```
REM Example 2 for unsecured printing
OPEN "lpt1:" FOR RANDOM AS #1
WIDTH #1, 255
PRINT #1, CHR$(27); "[?11~";
PRINT #1, CHR$(27); "PSCO"; CHR$(27); "\";
PRINT #1, CHR$(16); "5"; CHR$(25);
PRINT #1, "Example for ";
PRINT #1, CHR$(15); "LCP"; CHR$(15);" unsecured mode";
PRINT #1, CHR$(10); CHR$(13);
END
```

Example 3

```
REM Example 3 for secured printing
OPEN "lpt1:" FOR RANDOM AS #1
WIDTH #1, 255
PRINT #1, CHR$(27); "[?11~";
PRINT #1, CHR$(27); "PSC1"; CHR$(27); "\";
PRINT #1, CHR$(16); "5"; CHR$(25);
PRINT #1, "Example for ";
PRINT #1, CHR$(15); "LCP"; CHR$(15);
PRINT #1, " secured"; CHR$(10); "mode";
PRINT #1, STRING$(5, 10);
END
```

Example 4

Unsecured Mode, vertical spacing with various enlargement factors

Legend

A – a single line Feed (1/6")

B – Factor 6 character; height 6 x 1/12 = 3 x 1/6 LF

C – Factor 7 character; height 7 × 1/12 additional space is added to gain a full Line Feed

D – Factor 4 character; height 4 x 1/12 = 2 x 1/6 LF

E – Factor 5 character; height 5 x 1/12 additional space is added to gain a full Line Feed

 – start, actual print (cursor)-position

 – end, actual print-position

Description

The room is occupied to fit the highest character in one line. Characters which do not meet the 1/6" grid* have additional space added to fill the room to the next possible line.

*Value may change due to actual line density setting.

A Customer Support

TallyGenicom Customer Support Center

IMPORTANT **Please have the following information available prior to calling the TallyGenicom Customer Support Center:**

- Model number
- Serial number (located on the back of the printer)
- Installed options (i.e., interface and host type if applicable to the problem)
- Configuration printout: (See “Printing A Configuration”) in your printers Administrator’s manual.
- Is the problem with a new install or an existing printer?
- Description of the problem (be specific)
- Good and bad samples that clearly show the problem (faxing or emailing of these samples may be required)

Americas	(714) 368-2686
Europe, Middle East, and Africa	(31) 24 6489 311
Asia Pacific	(65) 6548 4114
China	(86) 800-999-6836

http://www.tallygenicom.com/support.aspx

TallyGenicom Supplies Department

Contact the TallyGenicom Supplies Department for genuine TallyGenicom supplies.

Americas	(800) 733-1900
Europe, Middle East, and Africa	(33) 1 46 25 19 07
Asia Pacific	(65) 6548 4100
China	(86) 400-886-5598
India	1800 102 7869

http://www.tallygenicom.com/supplies-parts.aspx

Corporate Offices

Printronix, LLC.
15345 Barranca Parkway
Irvine, CA 92618
U.S.A.
Phone: (714) 368-2300
Fax: (714) 368-2600

Printronix LLC.
c/o Printronix Nederland BV
Bijsterhuizen 11-38
6546 AS Nijmegen
The Netherlands
Phone: (31) 24 6489489
Fax: (31) 24 6489499

Printronix Schweiz GmbH
3 Changi Business Park Vista
#04-05 AkzoNobel House
Singapore 486051
Phone: (65) 6548 4100
Fax: (65) 6548 4111

Printronix Commercial (Shanghai) Co. Ltd.
Room 903, 9th Floor
No. 199, North Xizang Road
200070 Shanghai P.R. China
Phone: (86) 400 886 5598
Fax: (8621) 61171256

Printronix India Pvt Ltd
B-808/809, BSEL Tech Park
8th Floor, Sector 30A
Vashi Navi Mumbai 400705
India
Toll Free No.: 1800 102 7896
Fax: (9122) 4158 5555

Visit the TallyGenicom web site at www.tallygenicom.com

B *ASCII Chart*

KEY Hex | NUL 0 0 | ASCII Character / Decimal

B7 B6 B5 / BITS B4 B3 B2 B1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		NUMBERS SYMBOLS		UPPER-CASE		LOWER-CASE	
0 0 0 0	NUL 00 0	Data Link Escape 10 DLE 16	SPACE 20 32	0 30 48	@ 40 64	P 50 80	` 60 96	p 70 112
0 0 0 1	Start of Header SOH 01 1	Device Control One 11 DC1 17	! 21 33	1 31 49	A 41 65	Q 51 81	a 61 97	q 71 113
0 0 1 0	Start of Text STX 02 2	Device Control Two 12 DC2 18	" 22 34	2 32 50	B 42 66	R 52 82	b 62 98	r 72 114
0 0 1 1	End of Text ETX 03 3	Device Control Three 13 DC3 19	# 23 35	3 33 51	C 43 67	S 53 83	c 63 99	s 73 115
0 1 0 0	End of Transmission 04 EOT 4	Device Control Four 14 DC4 20	$ 24 36	4 34 52	D 44 68	T 54 84	d 64 100	t 74 116
0 1 0 1	Enquiry ENQ 05 5	Negative Acknowledge 15 NAK 21	% 25 37	5 35 53	E 45 69	U 55 85	e 65 101	u 75 117
0 1 1 0	Positive Acknowledge 06 ACK 6	Synchronous Idle 16 SYN 22	& 26 38	6 36 54	F 46 70	V 56 86	f 66 102	v 76 118
0 1 1 1	Alarm BEL 07 7	End Text Block ETB 17 23	' 27 39	7 37 55	G 47 71	W 57 87	g 67 103	w 77 119
1 0 0 0	Back Space BS 08 8	Cancel CAN 18 24	(28 40	8 38 56	H 48 72	X 58 88	h 68 104	x 78 120
1 0 0 1	Horizontal Tab HT 09 9	End of Message 19 EM 25	) 29 41	9 39 57	I 49 73	Y 59 89	i 69 105	y 79 121
1 0 1 0	Line Feed LF 0A 10	Substitute SUB 1A 26	* 2A 42	: 3A 58	J 4A 74	Z 5A 90	j 6A 106	z 7A 122
1 0 1 1	Vertical Tab VT 0B 11	Escape ESC 1B 27	+ 2B 43	; 3B 59	K 4B 75	[5B 91	k 6B 107	{ 7B 123
1 1 0 0	Form Feed FF 0C 12	Field Separator 1C FS 28	, 2C 44	< 3C 60	L 4C 76	\ 5C 92	l 6C 108	| 7C 124
1 1 0 1	Carriage Return 0D CR 13	Group Separator 1D GS 29	- 2D 45	= 3D 61	M 4D 77	] 5D 93	m 6D 109	} 7D 125
1 1 1 0	Shift Out SO 0E 14	Record Separator 1E RS 30	. 2E 46	> 3E 62	N 4E 78	^ 5E 94	n 6E 110	~ 7E 126
1 1 1 1	Shift In SI 0F 15	Unit Separator 1F US 31	/ 2F 47	? 3F 63	O 4F 79	_ 5F 95	o 6F 111	RUBOUT DEL 7F 127

C *Logos*

41 

42 

43 

44 

45 

46 

47 TM

48 TM

49 

4A 

4B 

4C 

61 

62 

D *Fonts*

^~FQ Draft

DRAFT, 7.5 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789

Draft, 10 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

Draft, 12 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

Draft, 13.3 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

Draft, 15 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

^~FQ NLQ

NLQ, 10 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

NLQ, 12 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

NLQ, 13.3 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

NLQ, 15 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

NLQ, 17.1 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

^~FQ OCR-A & B

OCR-A, 10 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

OCR-B, 10 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

^~FF Draft

DRAFT, 7.5 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789

Draft, 10 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

Draft, 12 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

Draft, 13.3 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

Draft, 15 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

Draft, 17.1 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

^~FF NLQ

NLQ, 10 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

NLQ, 12 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

NLQ, 13.3 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

NLQ, 15 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

NLQ, 17.1 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

^~FF OCR-A & B

OCR-A, 10 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

OCR-B, 10 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

^~FH Draft

DRAFT, 7.5 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789

Draft, 10 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

Draft, 12 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

Draft, 13.3 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

Draft, 15 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

Draft, 17.1 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

^~FH NLQ

NLQ, 10 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

NLQ, 12 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

NLQ, 13.3 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

NLQ, 15 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

NLQ, 17.1 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

^~FH OCR-A & B

OCR-A, 10 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

OCR-B, 10 CPI

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789

E Commands

Table 46: Command Table

Command	Description	Inside Pass	Outside Pass
^#n	Compressed Fonts		X
^@Cnn	Draft Fonts	X	X
^@Hn	Dump Mode		X
^@Lnn	Lines Per Inch (LPI)	X	X
^[nnn	Data Field, Buffered Forms	X	X
^A	Ignore Data Off	X	X
^B^-<data>^]	Buffered Formatting, Version 1		X
^B{B,N,O,S,T,Y}9x<ratio><data>^G	Barcode Style 1, Horz Var. Ratio	X	
^B{B,N,O,S,T,Y}x<data>^G	Barcode Style 1, Horizontal	X	
^Cnn<data>^Z	Buffered Form Copy		X
^C{B,N,O,S,T,Y}9x<ratio><data>^G	Barcode Style 1, Vert Var. Ratio	X	
^C{B,N,O,S,T,Y}x<data>^G	Barcode Style 1, Vertical	X	
^D	Descender Mode	X	
^~Dhhhvvv	Pass Density	X	X
^Dnn	Dot Slew		X
^Ehhwwjjd<data>	Graphics Pass, Vertical Left	X	X
^F	Free Format		X
^~FF	Full Space Font Mode	X	X
^~FH	Half Space Font Mode	X	X
^~FQ	Quality Font Mode	X	X
^G	Command Terminator	X	X
^Hnn	Form Length		X
^Hnn	Pass Height	X	

Table 46: Command Table

Command	Description	Inside Pass	Outside Pass
^IBARC,x,R<ratio>,{N,E,B},<data>^G	Barcode Style 2, Var. Ratio	X	
^IBARC,x,{N,E,B},<data>^G	Barcode Style 2	X	
^IFONT,S,n^G<data>	Default Font		X
^IFORM,C...	Buffered Form Create		X
^IFORM,D<name>^G	Buffered Form Delete		X
^IFORM,E<form name>^G<data>^G	Buffered Form Execute		X
^IFORM,L	Buffered Form List		X
^IFORM,R	Buffered Form Memory Reset		X
^IHEX,nn	Hex Command	X	X
^ILOGO,nn^G	Logos	X	X
^Innn	Interrupt Function	X	X
^IPEXP,h,v^G	Pixel Expansion	X	X
^IREPE	Repeat Terminator		X
^IREPH,n,hhd^G	Horizontal Repeat, Version 2		X
^IREPH,n,vvvd^G	Vertical Repeat, Version 2		X
^IISO,n^G	ISO Character Set	X	X
^Jvvd	Vertical Justification	X	
^KF	Half-Dot Mode	X	
^KH	Half-Tone Toggle	X	
^KLnn	Half-Tone		X
^Knn	Line Slew		X
^LBhhhdvvvdhv	Boxes	X	
^LDhhhdvvvd	Dashed Lines	X	
^LFhhhdvvvdhvlllldt... ^G	Form Drawing	X	
^Lnn	Form Length		X
^LShhhdvvvd	Solid Lines	X	
^Mhhwwjjd<data>	Graphics Pass, Horizontal	X	X
^Nx	Command Change		X

Table 46: Command Table

Command	Description	Inside Pass	Outside Pass
^O	Free Format Off		X
^PN	Graphics Mode Off		X
^PY	Graphics Mode On		X
^Q<plot data>^G	Plot Mode	X	
^R	Reverse Image Toggle	X	
^R^-	Buffered Form Repeat		X
^Rnnnn^-<data>^-^Z^-	Vertical Repeat, Version 1		X
^Sn	NLQ fonts	X	
^Snntt^-<data>^-^S^-	Horizontal Repeat, Version 1		X
^Thhhd	Horizontal Tab	X	
^Uhhwwjjd<data>	Graphics Pass, Upside Down	X	X
^Vhhwwjjd<data>	Graphics Pass, Vertical Down	X	X
^Wnn	Line Slew		X
^Wnn	Pass Width	X	
^X	Ignore Data On		X
^Yx{+,-}z	Automatic Increment/Decrement	X	X
^Z	Buffered Copy/Repeat Terminator		X

F *Modplot*

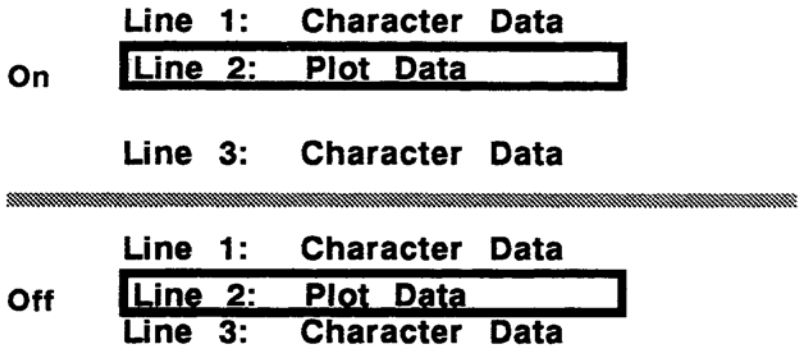
The Modplot option is designed to give the user a choice in spacing between Graphics and Text when leaving Graphics or Plot mode.

Modplot ON: A single LF advances the print postion one full line. This is the default setting.

Modplot OFF: A single LF advances the print position to the next line position relative to the top of form.

Modplot can be found as a print option in Configuration Menu, under the Printer section.

Example:



G Control Panel Selection

Printer Control Panel: Graphics Category

This Appendix lists the Graphics Category parameters in the Configuration Menu of your printer. It includes changes and additions that were made to the menu.

Graphic Category (Menu Level 1)

This category allows you to configure certain aspects of the Graphics Processing Options on your printer. The Graphic Category contains parameters CVCC (Code V Command Character), Smooth, PY Then, PN Then, DarkBar, Version, Descndr, Zero, ModPlot, Vscale (Vertical Scale), and SFCC (PGL Special Function Command Character). Complete explantions for these parameters are located in the Graphic Application Manual.

Code V Command Character (CVCC) Parameter (Level 2 Menu)

This parameter allows you to change the Graphic Command Character. The default for this parameter is the ASCII caret (^, Decimal 94, HEX 5E) character. (Refer to ^N Explanation in the Graphic Application Manual).

CVCC Cmd Selections (Menu Level 3)

1-255 (Decimal designators for ASCII characters)

Smooth Parameter (Menu Level 2)

This parameter controls the size at which block characters are smoothed. The default size is 3.

Smooth Selections (Menu Level 3)

2-99

PY Then Parameter (Menu Level 3)

This command defines printer response to characters following the ^PY Command on the same line.

PY Then Selections (Menu Level 3)

None (Default) All

Term

NOTE: How your printer responds to data and control codes that follow a ^PY Command or a ^PN Command on the same line is shown in the Code V Application section of the Graphics Application Manual in Appendix H.

PN Then Parameter (Menu Level 2)

This command defines printer response to characters following the ^PN Command on the same line.

NOTE: How your printer responds to data and control codes that follow a ^PY Command or a ^PN Command on the same line is shown in the Code V Application section of the Graphics Application Manual in Appendix H.

PN Then Selections (Menu Level 3)

None (Default) All

DarkBar Parameter (Menu Level 2)

This parameter allows you to increase the darkness of the barcodes by changing horizontal dot density from 60 DPI to 120 DPI.

DarkBar Selections (Menu Level 3)

OFF (Default) ON

Version Parameter (Menu Level 2)

This parameter controls the version of Code V the printer emulates.

Version Selections (Menu Level 3)

2 (Version 2) (Default selection)

1 (Version 1)

Descender (Descndr) Parameter (Menu Level 2)

This parameter controls insertion of the character descender gap between print lines.

Descndr Selections (Menu Level 2)

Fixed: The descender gap is always inserted after the line regardless of the setting of Descender Mode (Default selection).

Auto: The descender gap is only inserted after lines containing characters with descenders.

Zero Parameter (Menu Level 2)

As an aid in distinguishing zeros from the upper-case letter O you can choose to have your zeros slashed (Ø).

Zero Selections (Menu Level 3)

Open (O)

Slashed (Ø) (Default selection)

Vscale Parameter (Menu Level 2)

This parameter determines whether vertical block characters will be scaled to the same dimensions as horizontal block characters.

Vscale Selections (Menu Level 3)

OFF

ON (Default selection)

PGL Special Function Command Character (SFCC) Parmeter (Menu Level 2)

This parameter allows you to change the SFCC. The default for this parameter is the ASCII tilde (~, Decimal 126, HEX 7E) character. This parameter is only applicable to the PGL Graphics option. (See “Change SFCC Command (Normal and Execute)” on page 140.

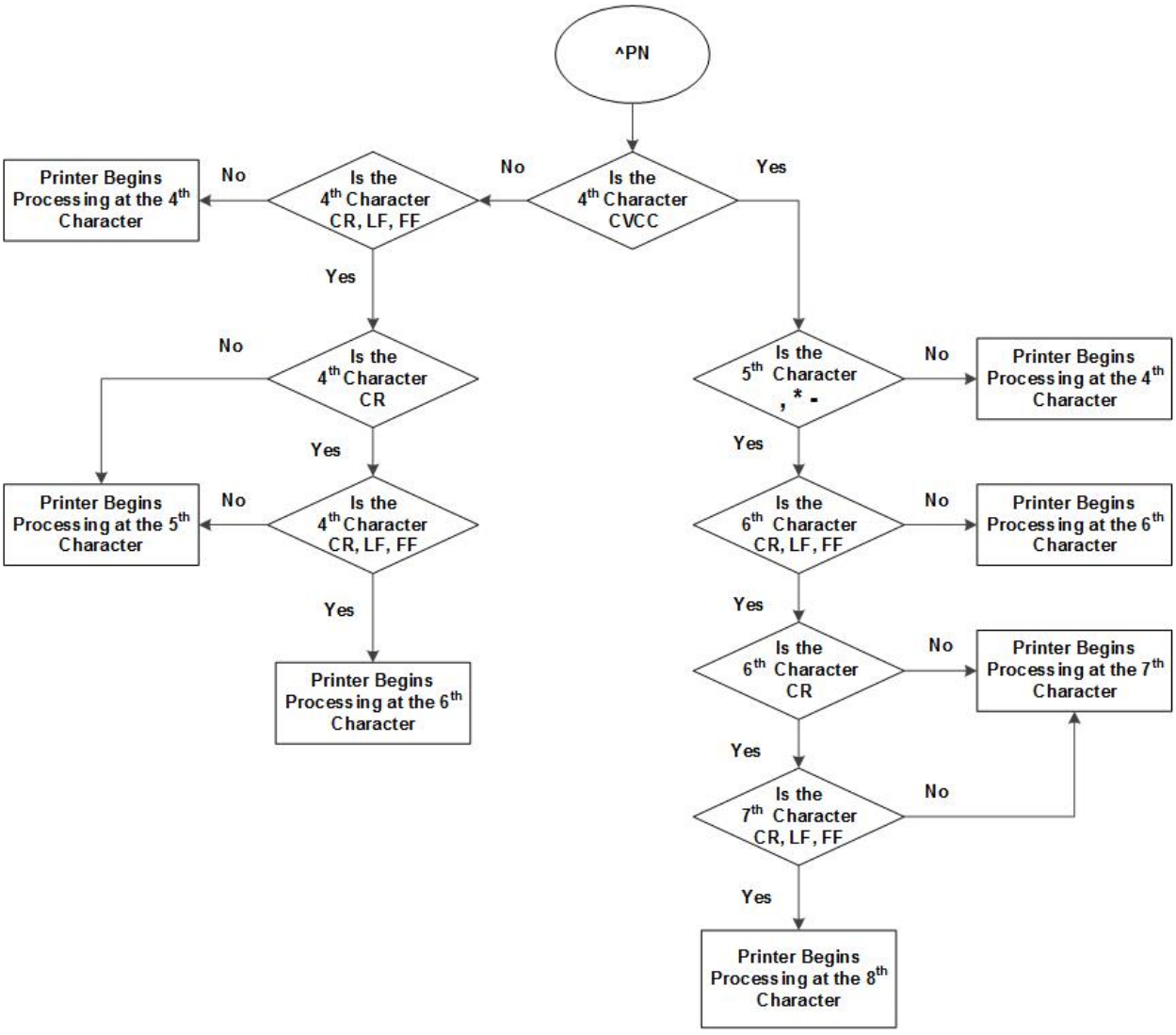
SFCC Selections (Menu Level 3)

1-255 (Decimal designators for ASCII characters).

H *PY/PN Data Processing*

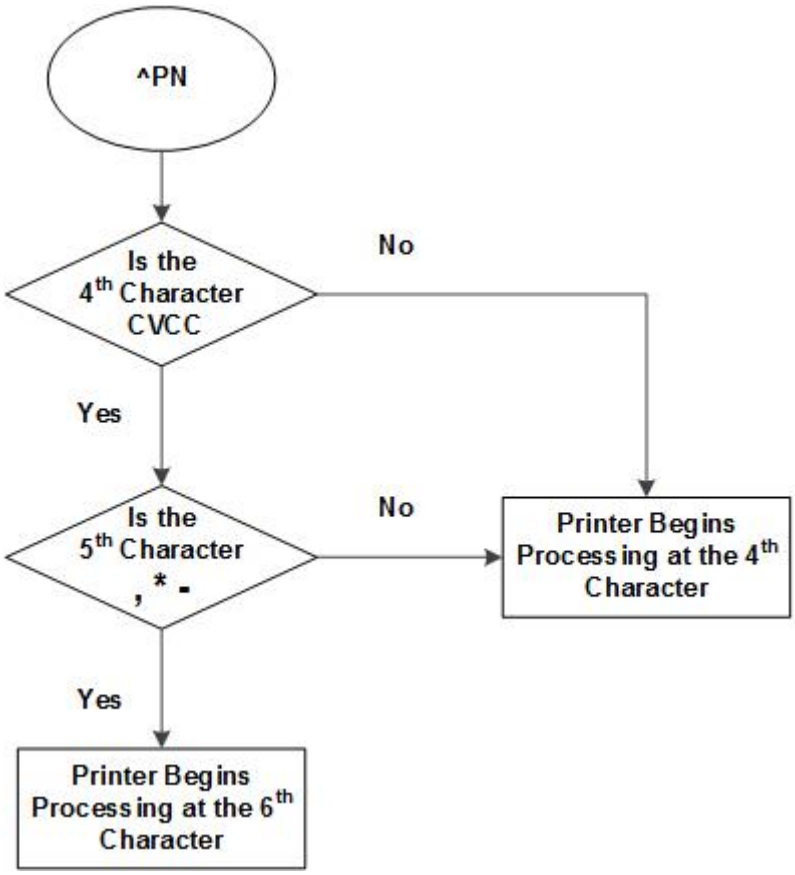
PN Then Data Processing When PN Then = NONE

Character Number	1	2	3	4	5	6	7	8	9	.....
Data Stream	^	P	N	X	O	X	O	X	O	.....

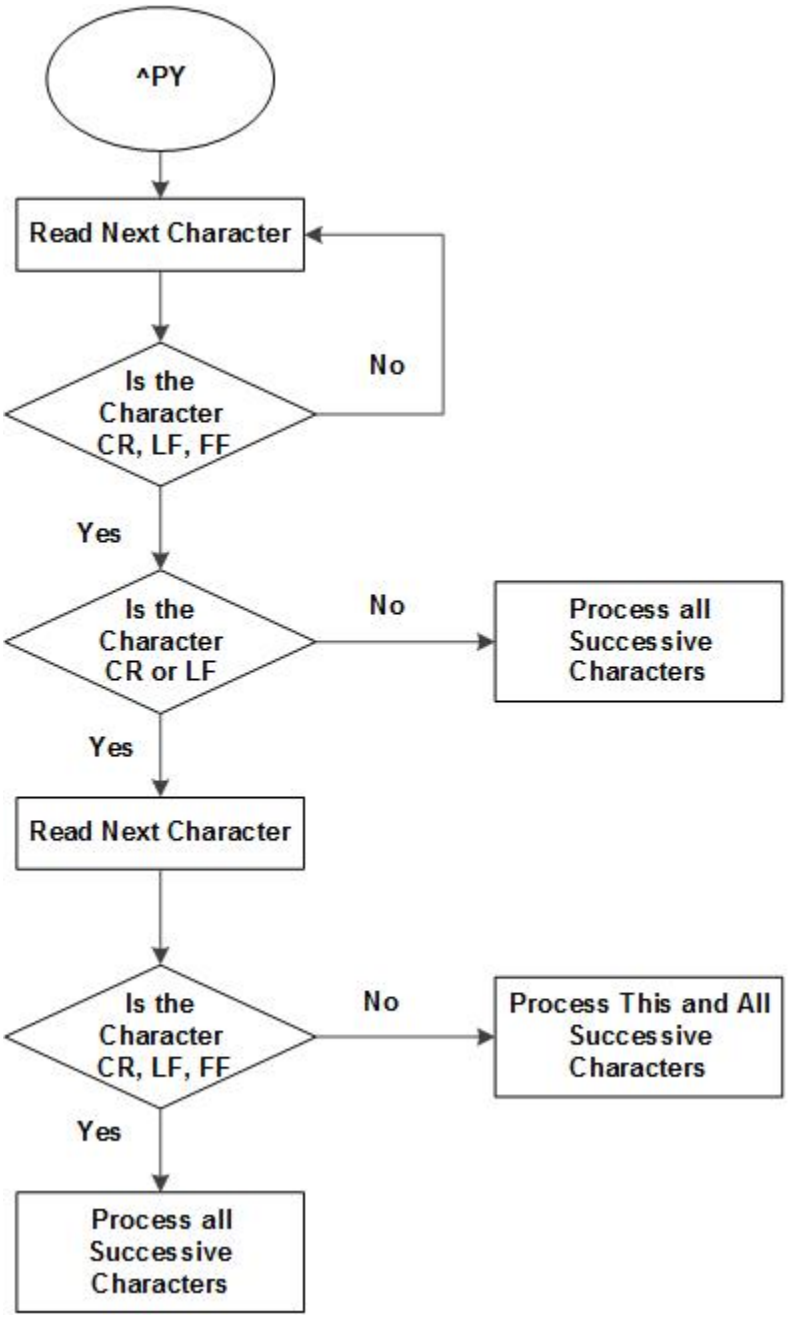


PN Then Data Processing When PN Then = ALL

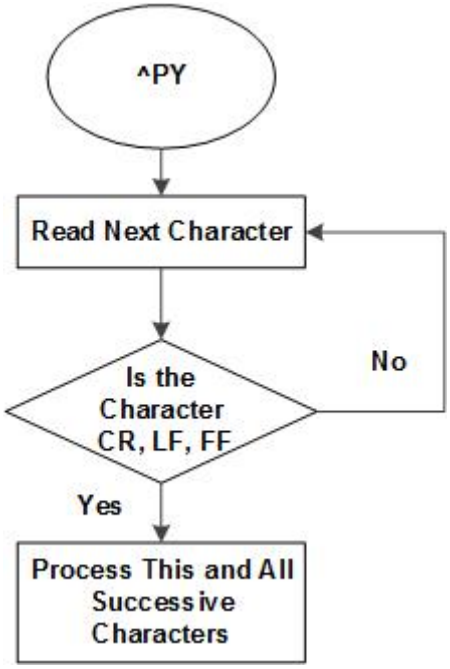
Character Number	1	2	3	4	5	6	7	8	9	.....
Data Stream	^	P	N	X	O	X	O	X	O	.....



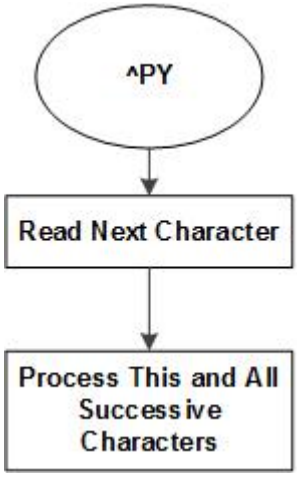
PY Then Data Processing When PY Then = NONE



PY Then Data Processing When PY Then = TERM



PY Then Data Processing When PY Then = ALL



I Block-Character Size Tables

Block Character Size Table

The following table lists the approximate physical size of expanded Block Characters specified in the Alphanumerics String Creation Command.

Table 47: Block Character Size

Expansion Factor	Character Height	Character Width	Expansion Factor	Character Height	Character Width
0	0	0	17	1.75"	1.67"
1	0.12″	0.08″	18	1.75″	1.83″
2	0.23″	0.17″	19	1.87″	1.92″
3	0.35″	0.33″	20	1.98″	2.00″
4	0.35″	0.42″	21	2.10″	2.08″
5	0.47″	0.50″	22	2.22″	2.13″
6	0.58″	0.58″	23	2.33″	2.33″
7	0.70″	0.67″	24	2.45″	2.42″
8	0.82″	0.83″	25	2.45″	2.50″
9	0.93″	0.92″	26	2.57″	2.58″
10	1.05″	1.00″	27	2.68″	2.67″
11	1.05″	1.08″	28	2.80″	2.83″
12	1.17″	1.17″	29	2.92″	2.92″
13	1.28″	1.33″	30	3.03″	3.00″
14	1.40″	1.42″	31	315″	3.08″
15	1.52″	1.50″	32	3.15″	3.17″
16	1.63″	1.58″	33	3.27″	3.33″
34	3.38″	3.42″	62	6.18″	6.17″
35	3.50′	3.50′	63	6.30″	6.33″
36	3.62″	3.58″	64	6.42″	6.42″
37	3.73″	3.67″	65	6.53″	6.50″
38	3.85″	3.92″	66	6.65″	6.58″
39	3.97″	4.00″	67	6.65″	6.67″
40	4.08″	4.08″	68	6.77″	6.83″
41	4.20″	4.17″	69	6.88″	6.92″
42	4.20″	4.33″	70	7.00″	7.00″
43	4.43″	4.42″	71	7.12″	7.08″

Table 47: Block Character Size

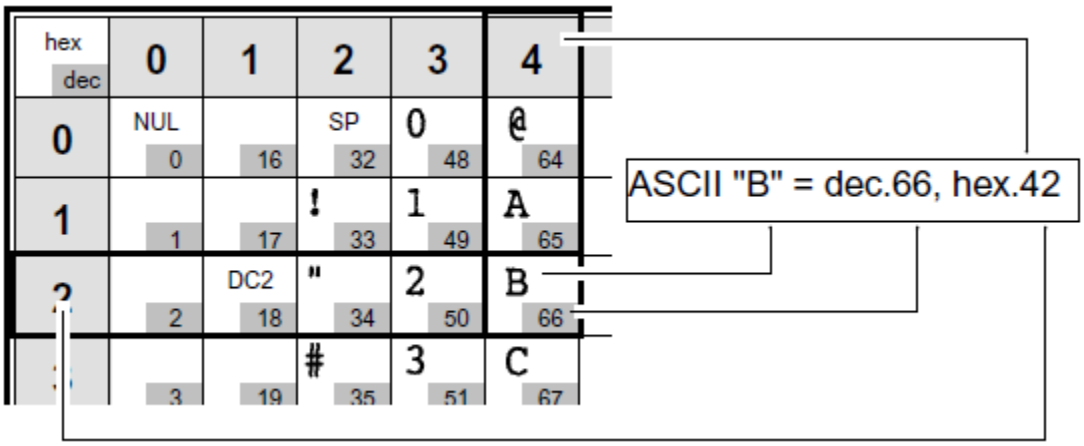
Expansion Factor	Character Height	Character Width	Expansion Factor	Character Height	Character Width
44	4.55″	4.50″	72	7.23″	7.17″
45	4.55″	4.58″	73	7.35″	7.33″
46	4.67″	4.67″	74	7.35″	7.42″
47	4.78″	4.83″	75	7.47″	7.50″
48	4.90″	4.92″	76	7.58″	7.58″
49	5.02″	5.00″	77	7.70″	7.67″
50	5.13″	5.08″	78	7.82″	7.83″
51	5.25″	5.17″	79	7.93″	7.92″
52	5.25″	5.17″	80	8.05″	8.00″
53	5.37″	5.42″	81	8.05″	8.08″
54	5.48″	5.50″	82	8.17″	8.17″
55	5.60″	5.58″	83	8.28″	8.33″
56	5.72″	5.67″	84	8.40″	8.42″
57	5.72″	5.57″	85	8.52″	8.50″
58	5.83″	5.83″	86	8.63″	8.58″
59	5.95″	5.92″	87	8.75″	8.67″
60	5.95″	6.00″	88	8.75″	8.83″
61	6.07″	6.08″	89	8.87″	8.92″
90	8.98″	9.00″	102	10.15″	10.17″
91	9.10″	9.08″	103	10.27″	10.33″
92	9.22″	9.17″	104	10.38″	10.42″
93	9.33″	9.33″	105	10.50″	10.50″
94	9345″	9.42″	106	10.63″	10.58″
95	9.45″	9.42″	107	10.73″	10.67″
96	9.57″	9.58″	108	10.85″	10.83″
97	9.68″	9.67″	109	10.85″	10.92″
98	9.80″	9.83″	110	10.97″	11.00″
99	9.92″	9.92″	111	11.08″	11.08″
100	10.03″	10.00″	112	11.20″	11.17″
101	10.15″	10.08″	113	11.32″	11.33″

J Character Sets

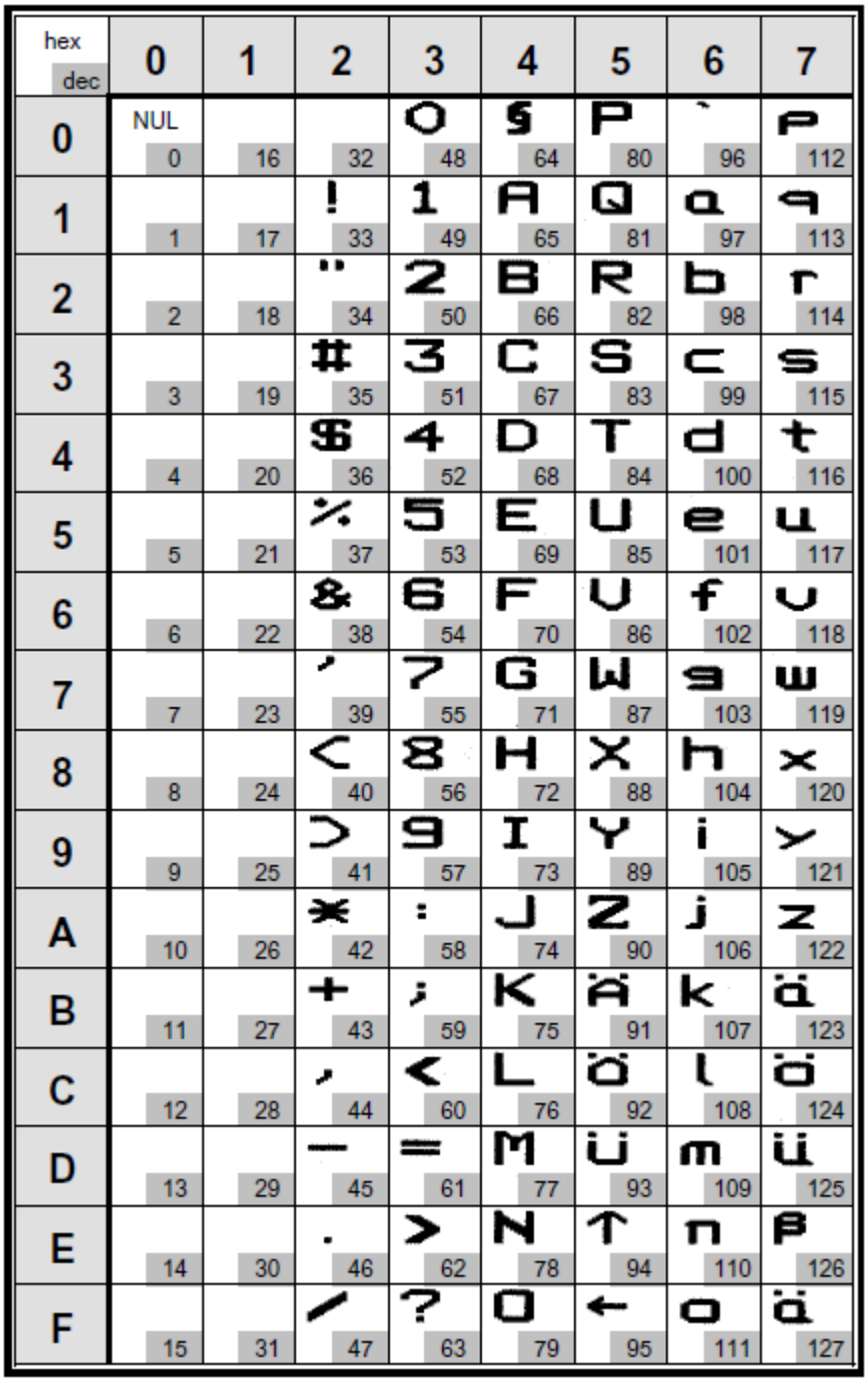
The following symbol sets are available in the barcode mode.

The LCP symbol sets can only be selected via Escape sequences.
The OCR-A and OCR-B fonts can be selected via control panel or by Escape sequences, which correspond to the selected emulation mode. The codes hex.A0 up to hex.FE correspond to the selected character set.

The following example shows you how to find the hexadecimal value for a character from the character set table.

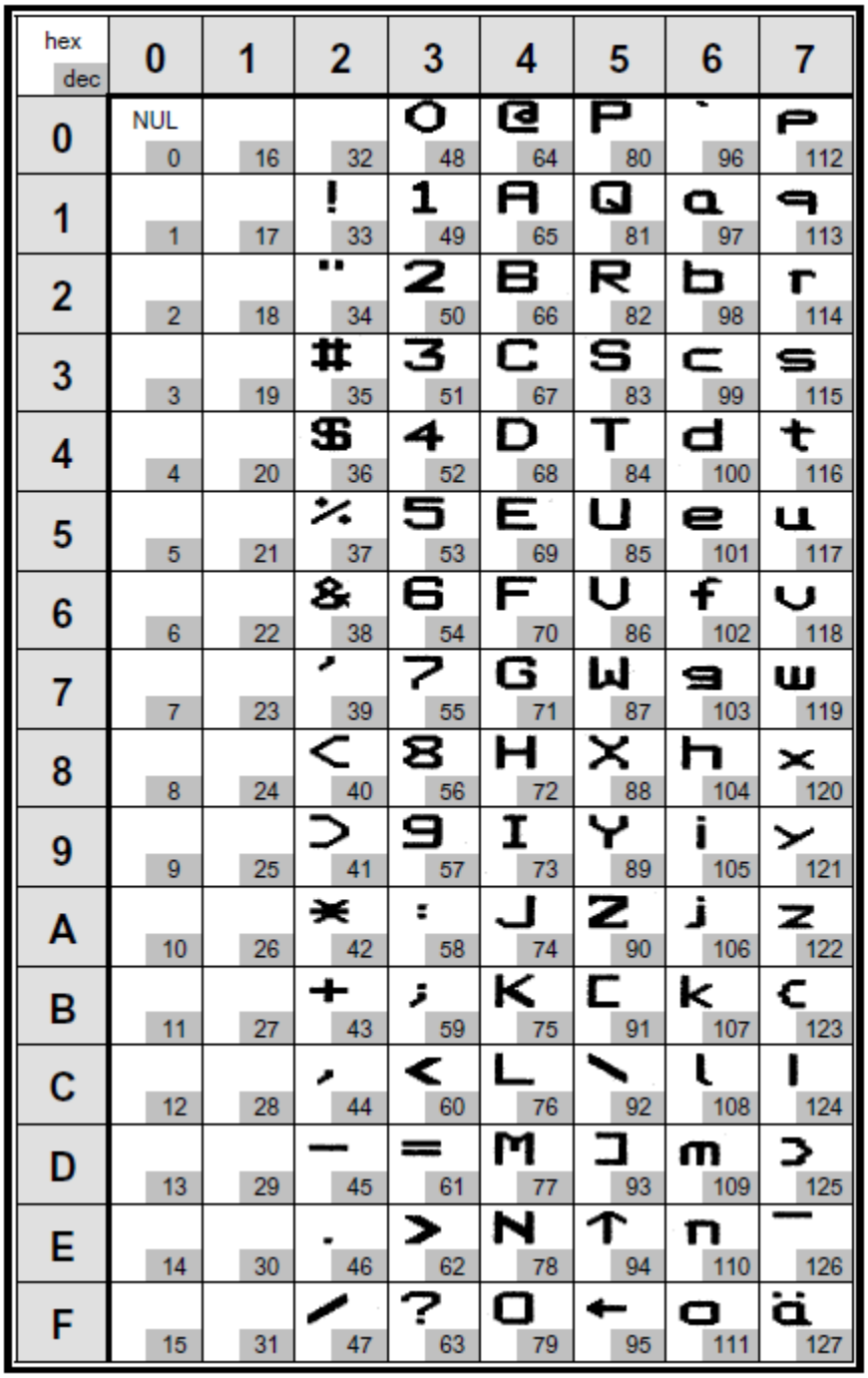


German



hex / dec	0	1	2	3	4	5	6	7
0	NUL 0	16	32	48	64	80	96	112
1	1	17	33	49	65	81	97	113
2	2	18	34	50	66	82	98	114
3	3	19	35	51	67	83	99	115
4	4	20	36	52	68	84	100	116
5	5	21	37	53	69	85	101	117
6	6	22	38	54	70	86	102	118
7	7	23	39	55	71	87	103	119
8	8	24	40	56	72	88	104	120
9	9	25	41	57	73	89	105	121
A	10	26	42	58	74	90	106	122
B	11	27	43	59	75	91	107	123
C	12	28	44	60	76	92	108	124
D	13	29	45	61	77	93	109	125
E	14	30	46	62	78	94	110	126
F	15	31	47	63	79	95	111	127

US-ASCII



hex / dec	0	1	2	3	4	5	6	7
0	NUL 0	16	32	48	64	80	96	112
1	1	17	! 33	1 49	A 65	Q 81	a 97	q 113
2	2	18	" 34	2 50	B 66	R 82	b 98	r 114
3	3	19	# 35	3 51	C 67	S 83	c 99	s 115
4	4	20	$ 36	4 52	D 68	T 84	d 100	t 116
5	5	21	% 37	5 53	E 69	U 85	e 101	u 117
6	6	22	& 38	6 54	F 70	V 86	f 102	v 118
7	7	23	' 39	7 55	G 71	W 87	g 103	w 119
8	8	24	(40	8 56	H 72	X 88	h 104	x 120
9	9	25	) 41	9 57	I 73	Y 89	i 105	y 121
A	10	26	* 42	: 58	J 74	Z 90	j 106	z 122
B	11	27	+ 43	; 59	K 75	[91	k 107	{ 123
C	12	28	, 44	< 60	L 76	\ 92	l 108	| 124
D	13	29	- 45	= 61	M 77	] 93	m 109	} 125
E	14	30	. 46	> 62	N 78	↑ 94	n 110	~ 126
F	15	31	/ 47	? 63	O 79	← 95	o 111	ä 127

OCR-A (Code Page 437)

hex / dec	0	1	2	3	4	5	6	7
0	NUL 0	16	32	0 48	@ 64	P 80	` 96	p 112
1	1	17	! 33	1 49	A 65	Q 81	a 97	q 113
2	2	18	" 34	2 50	B 66	R 82	b 98	r 114
3	3	19	# 35	3 51	C 67	S 83	c 99	s 115
4	4	20	$ 36	4 52	D 68	T 84	d 100	t 116
5	5	21	% 37	5 53	E 69	U 85	e 101	u 117
6	6	22	& 38	6 54	F 70	V 86	f 102	v 118
7	7	23	' 39	7 55	G 71	W 87	g 103	w 119
8	8	24	(40	8 56	H 72	X 88	h 104	x 120
9	9	25	) 41	9 57	I 73	Y 89	i 105	y 121
A	10	26	* 42	: 58	J 74	Z 90	j 106	z 122
B	11	27	+ 43	; 59	K 75	[91	k 107	{ 123
C	12	28	, 44	< 60	L 76	\ 92	l 108	| 124
D	13	29	- 45	= 61	M 77	] 93	m 109	} 125
E	14	30	. 46	> 62	N 78	^ 94	n 110	~ 126
F	15	31	/ 47	? 63	O 79	_ 95	o 111	127

OCR-B (Code Page 437)

hex / dec	0	1	2	3	4	5	6	7
0	NUL 0	16	32	0 48	@ 64	P 80	` 96	p 112
1	1	17	! 33	1 49	A 65	Q 81	a 97	q 113
2	2	18	" 34	2 50	B 66	R 82	b 98	r 114
3	3	19	# 35	3 51	C 67	S 83	c 99	s 115
4	4	20	$ 36	4 52	D 68	T 84	d 100	t 116
5	5	21	% 37	5 53	E 69	U 85	e 101	u 117
6	6	22	& 38	6 54	F 70	V 86	f 102	v 118
7	7	23	' 39	7 55	G 71	W 87	g 103	w 119
8	8	24	(40	8 56	H 72	X 88	h 104	x 120
9	9	25	) 41	9 57	I 73	Y 89	i 105	y 121
A	10	26	* 42	: 58	J 74	Z 90	j 106	z 122
B	11	27	+ 43	; 59	K 75	[91	k 107	{ 123
C	12	28	, 44	< 60	L 76	\ 92	l 108	| 124
D	13	29	- 45	= 61	M 77	] 93	m 109	} 125
E	14	30	. 46	> 62	N 78	^ 94	n 110	~ 126
F	15	31	/ 47	? 63	O 79	_ 95	o 111	127

K Addendum to the Line Printer Graphics Applications Manual

Postal Barcodes

Barcode	Code V	PGL
U.S. POSTNET	^IBARC,POSTNET	barcode=POSTNET
U.S. PLANET	^IBARC,PLANET	barcode=PLANET
U.K. 4-State	^IBARC,UKPOST	barcode=UKPOST barcode=ROYALBAR
Dutch KIX 4-State	^IBARC,KIX	barcode=KIX barcode=ROYALBAR;KIX
PostBar (generic POSTNET-style barcode)		barcode=POSTBAR

PDF417 2-D Barcode

PDF417 is a multi-row, variable length symbology offering high data capacity and error correction capability. Linear scanners, rastering laser scanners, or two-dimensional imaging devices can scan PDF417. One PDF417 symbol is capable of encoding more than 1100 bytes, 1800 ASCII characters or 2700 digits, depending on the selected data compaction mode.

Every PDF417 symbol is composed of a stack of rows, from a minimum of 3 to a maximum of 90 rows. Each PDF417 row contains start and stop patterns, left and right row indicators, and from 1 to 30 data symbol characters. Since both the number of rows and their length are selectable, the aspect ratio of a PDF417 symbol can be varied to suit spatial requirements for printing.

PDF417 in Code V

^IBARC,PDF417,<1-99>,<0-9>,<data>^G

Where:

<1-99> is the number of desired data columns in the barcode.

<0-9> designates the error correction level.

Error Correction Level	Number of Error Correction Codewords
0	2
1	4
2	8
3	16
4	32
5	64
6	128
7	256
8	512
9	Auto

If error correction level 9 (auto) is selected, the error level will be selected as follows:

Number of Data Codewords	Error Correction Level
1-40	2
40-160	3
161-320	4
321-863	5

NOTE: This table corresponds to the recommended error correction levels.

The height and width of the bars correspond to the magnification of the current pass.

Example:

^M1103000^T0010^IBARC,PDF417,2,3,This is an example^G

Produces a barcode with bars 11 dots high and 3 dots wide, containing 2 data columns and 16 error correction codewords (error correction level 3). The encoded data is “This is an example”.

PDF417 in PGL

BARCODE

PDF417;<X<D>n;><Y<D>n;><(Hn)|(Wn);><ASPECT;h:w;><(Rn)| (Cn);><Sn;>

<BFn;L;><DARK;>SR;SC(T)

STOP(T)

X<(D)>n	Specifies the horizontal size of the narrow bar in 1/60ths of an inch, if D is present. If D is not present the size is specified in dots.
Y<D>n	Specifies the vertical size of the narrow bar in 1/60ths of an inch, if D is present. If D is not present the size is specified in dots.
Hn	Specifies the overall height of the barcode in 1/10ths of an inch. The valid range for n is 4-99.
Wn	Specifies the overall width of the barcode in 1/10ths of an inch. The valid range for n is 4-99.
ASPECT;h:w	Specifies the overall height to width aspect ratio of the barcode. The default setting is 1:2.
Rn	Specifies the number of rows in the barcode. The valid range for n is 3-90.
Cn	Specifies the number of data columns in the barcode. The valid range for n is 1-90.
Sn	Specifies the security level, which determines the number of error correction codewords contained in the barcode. The valid range for n is 0-9.

Since many of the formatting commands for PDF417 barcodes conflict with each other, most of them cannot be used in conjunction with each other. Table 48 on page 255 shows commands that cannot be used together.

Table 48: Commands That Cannot Be Used Together

	X	Y	H	W	ASPECT	R	C	X
X	-							
Y		-						
H			-	X	X	X	X	
W			X	-	X	X	X	
ASPECT			X	X	-	X	X	
R			X	X	X	-	X	
C			X	X	X	X	-	
S								-

Code 128A Barcodes in PGL

The PGL graphics emulation now supports selection of Code 128A barcodes from within a bar code command. Enter C128A as the barcode name.

Codabar Codes in PGL

The PGL graphics emulation now supports selection of Codabar barcodes from within a bar code command. Enter CODABAR as the bar codename.

UCC/EAN-128 Barcodes in PGL

The PGL graphics emulation now supports selection of UCC-128 (EAN-128) barcodes from within a bar code command. Enter EAN128, UCC128, or UCC-128 as the barcode name.

UPC-E0 Bar Codes in PGL

The PGL graphics emulation now supports selection of UPC-E0 barcodes from within a barcode command. Enter UPC-E0 as the bar code name.

Barcode Rotations in PGL

Barcodes can now be rotated in four different orientations instead of the former two orientations. VSCAN is still supported to create a 90 degree counterclockwise barcode. The new parameters are use in place of VSCAN to allow new rotations. The new parameters are CW to create a 90 degree clockwise bar code, INV to create an inverted (180 degree) bar code, and CCW to create a 90 degree counterclockwise bar code.

User-defined Barcode Ratios in PGL

Bar code ratios can now be defined to control bar code size. The XR and XRD parameters are used in the same location as the current magnification value parameter. The XR and XRD parameters are followed by four ratio values (eight for Code 128, UPC, and EAN bar codes).

These values specify the narrow bar, narrow space, wide bar, and wide space widths in printer dots (60 dpi for XR and 120 dpi for XRD), each separated by a colon. The following example illustrates the usage of the XR parameter:

```
BARCODE
C3/9;XR1:1:2:2;H10;5;1
*Code 39*
STOP
```

The previous values for magnification continue to be valid.

New Barcode Fonts in PGL

New values have been added to the FONT parameter of the BARCODE command. The following table identifies the valid values.

Value	Description
N	10 CPI ASCII
P	12 CPI ASCII
Q	13 CPI ASCII
R	15 CPI ASCII
T	17 CPI ASCII
V	20 CPI ASCII

Value	Description
O	10 CPI OCR-A
X	10 CPI OCR-B
S	Supresses printing of human readable characters.

ISET/USET Commands in PGL

The PGL graphics emulation now supports the ISET and USET commands. The ISET command can be used in NORMAL, CREATE, and EXECUTE modes and is used to select standard or user-defined character sets. User- defined character sets are created with the USET command in NORMAL mode. These commands have the following format:

```
<SFCC>ISET;n (Do not use the SFCC in CREATE mode)
<SFCC>USETn
ca;fa (one to sixteen ca;fa pairs)
END
```

Field	Description
<SFCC>	The special function control character.
n	The character set number. See the following table for descriptions of the character sets.
ca	The ASCII character cell hex address which will be replaced by the character represented by the font address (fa) parameter. The ca parameter must be equal to one of the following hex values: 21, 22, 23, 24, 25, 26, 40, 5B, 5C, 5D, 5E, 60, 7B, 7C, 7D, 7E.
fa	The font address of a character that will replace the character located in the character address (ca). This parameter must be between the hex values 20 and FF.

ISET Parameter	USET Parameter	Character Set
0		US ASCII
1		German
2		Swedish
3		Danish
4		Norwegian
5		Finnish
6		English

ISET Parameter	USET Parameter	Character Set
7		Dutch
8		French
9		Spanish
10		Italian
11		Turkish
24	1	User-defined character set 1
25	2	User-defined character set 2
26	3	User-defined character set 3
27	4	User-defined character set 4
28	5	User-defined character set 5
29	6	User-defined character set 6
30	7	User-defined character set 7
31	8	User-defined character set 8

POINT Parameter in PGL

An optional POINT parameter has been added to the ALPHA command. The structure of this command is now the following:

```
ALPHA
R;E;Cn;Afn;L;Rot;U-case;D;POINT;X1;X2;X3;X4;‡text‡
STOP
```

The POINT parameter is used to change the units of the horizontal and vertical expansion parameters (X3 and X4). When POINT is specified, the X3 and X4 parameters are interpreted in 1/72 of an inch (points). If this parameter is specified, the E and Cn parameters should not be used.

20 CPI Support in PGL

20 CPI text can now be selected from the ALPHA and DENSITY commands. Use C20 in the ALPHA command and 20 in the DENSITY command.

SCALE Command Modifications in PGL

The SCALE command has been modified in the following manner:

Field	Function	Option/Modifier Selections
Dot Scale Placement		
SCALE;	This is the command specifier.	Command Identifier
DOT	Specifies that placement will be done using a dot-scale.	Enter DOT to specify a dot measurement scale (dot-columns and dot-rows).
;HORZ	An optional parameter that specifies the horizontal DPI used to place elements on the form.	Enter the horizontal DPI. No value will result in the default setting of 60.
;VERT	An optional parameter that specifies the vertical DPI used to place elements on the form.	Enter the vertical DPI. No value will result in the default setting of 72.
Character Scale Placement		
SCALE;	This is the command specifier.	Command Identifier
CHAR	Specifies that placement will be done using a character-scale.	Enter CHAR to specify a character measurement scale (columns and rows).
;LPI	An optional parameter that can be used to specify Lines Per Inch settings for print output.	Selections that can be used are 6, 8, 9, 10, 12, and 15. No value will result in the default setting of 6 LPI.
;CPI	An optional parameter that can be used to specify Characters Per Inch settings for print output.	Selections that can be used are 10, 12, 13, 15, 17, and 20. No value will result in the default setting of 10 CPI.