Comtec®

Printing Systems

Programming Manual

Comtec Information Systems, Inc.

Part No. PRO-MAN

Rev. F July,199

Proprietary Statement
This manual contains proprietary information of Comtec Information Systems, Inc. It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the expressed written permission of Comtec Information Systems, Inc.

Inasmuch as every effort has been made to supply accurate information in this manual, Comtec is not liable for any erroneous information or omissions.

Copyrights

Comtec Information Systems, Inc.
30 Plan Way, Warwick, Rhode Island 02886, USA 401/739-5800
800/556-7266
Fax: 401/732-0145
EMail: support@comtecis.com

Comtec Europe B.V.
Nuenenseweg 167, 5667 KP Geldrop,The Netherlands +31 40 286 29 80
Fax: +31 40 286 27 36

Visit Our Website
www.comtecinfosys.com

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

Programming Section

PROGRAMMING INTRODUCTION

Your printer has built in text, graphics and bar code printing capability. This manual will introduce these commands to you using the following conventions:

{ } Required item

[ ] Optional item

( ) Abbreviated command

< > Literal item

A space character is used to delimit each field in a command line.

After the word 'Input' in each example, the file name of that example will appear in parentheses. These files can be found on the Miniature Printer Utility disk provided with this manual.

GETTING PRINTER INFORMATION

The printer can produce a report containing information about the application resident in the printer. A copy similar to the example printout shown here can be obtained from your printer by doing the following:

1. Turn the printer OFF.
2. While holding the FEED key down, turn the printer ON.
3. When printing begins, release the feed key.

The printer will now issue two reports similar to the example showing current information about fonts, baud rate settings, and time-outs. The Label Height value shown is the amount of label memory available in dots. For example, if Label Height is 1400, you can print labels up to 7 inches (178 mm) in length. This value is derived from a printer resolution of 203 dots/inch (8 dots/mm).

GETTING PRINTER INFORMATION EXAMPLE

Output:

First Report:

```
Comtec MP5022 V60.05 12/11/95
  CHK: 0FD9  XXMP99-99-9999
  AW11820-005
  Testing Memory...
  Memory tested and OK
  Baud Rate: 19200 BPS
  In-activity Timeout: 120 Secs
  Low battery Shut-down: 190
  RESIDENT FONTS:
    Font      Size(s)
    --------  -------
     0         0-6
     1         0
     2         0-1
     4         0-7
     5         0-3
     6         0
     7         0-1

  End of report.
```

Second Report:

```
Comtec MP5022 V60/24 06/30/97
  CHK: AB55
```

```
  XXMP99-99-9999
  AW11820-005
  MP STD Appl v24k
  R60.05.U47.B7.T14.A24
  Baud Rate: 19200 BPS
  In-activity Timeout: 120 Secs
  Low battery Shut-down: 190
  Remote(DTR) pwr-off: Enabled
  CCL Key '!'[21]
  Label width : 384
  Label height: 3624
  RESIDENT FONTS:
    Font      Size(s)
    --------  -------
     0         0-6
     1         0
     2         0-1
     4         0-7
     5         0-3
     6         0
     7         0-1
  Directory
   854000 Bytes Free

  End of report.

  Press FEED key to
  enter DUMP mode.


  Dump mode not entered.
```

COMMUNICATIONS DIAGNOSTICS MODE

To aid the user in diagnosing communications problems, the printer features a Communications Diagnostics Mode (Dump Mode). In the dump mode, the printer will print the ASCII hex codes of the data sent to it, and their text representation (or the period '.', if not a printable character). As a test of the printer the “ALL CHRS.LBL” file may be sent.

To enter Communications Diagnostics Mode:

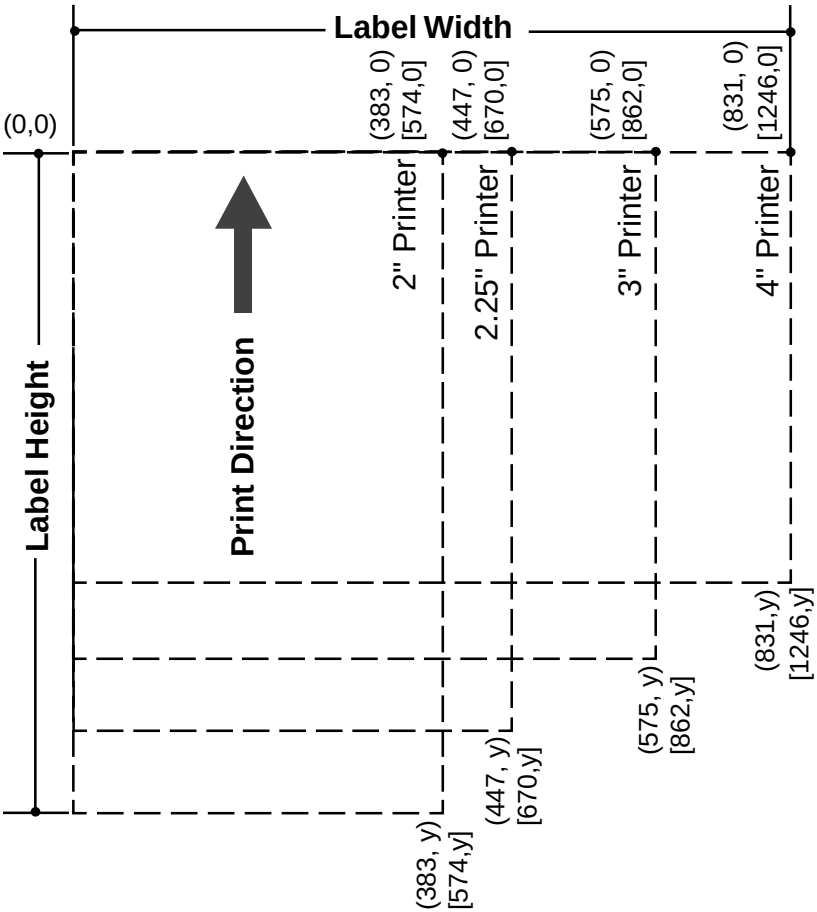
1. Turn the printer OFF.
2. Hold FEED key down.
3. Turn the printer ON.
4. Release FEED key when printer starts printing the diagnostics.
5. Wait for the 2nd diagnostics report.
6. At the end of 2nd diagnostics report, the printer will print: "Press FEED key to enter DUMP mode".
7. Now press the FEED key. The printer will print: "Entering DUMP mode".
   Note: You have 3 seconds to press the FEED key. If the FEED key is not pressed within 3 seconds, the printer will print "DUMP mode not entered" and will resume normal operation.
8. At this point, the printer is in DUMP mode and will print the ASCII hex codes of the data sent to it, and their text representation (or "." if not a printable character).

To cancel Communications Diagnostics Mode:

1. Turn the printer OFF.
2. Wait 5 seconds.
3. Turn the printer ON.

LABEL COORDINATE SYSTEM

The x and y coordinates are expressed here in terms of dots. Coordinates in ( ) are for 200 dot per inch printers. On 200 d.p.i. printers, 8 dots (either horizontally or vertically) equal 1 millimeter and 203 dots equate approximately to 1 inch.
Coordinates in [ ] are for 300 dot per inch printers. On 300 d.p.i. printers 12 dots equal 1 millimeter, and 305 dots equate approximately to 1 inch.



Note that these coordinates refer to the actual printing area of the printers.

"y" =the available label height which varies with the resident application.
(See Getting Printer Information, Page P1-2.)

Label Vista™

Label Vista is a stand-alone program running in Windows™95/NT that allows users with little or no programming background to design labels which can be printed on the full range of Comtec portable and desktop printers. It combines an intuitive graphically based user environment with powerful, but easily mastered, editing tools.

The Premier Edition of Label Vista allows the creation of printable, fixed-size (pre-scaled) fonts derived from an included library of TrueType™ fonts, which greatly enhances the versitility of this program.

In addition, Label Vista allows the easy creation of format files which can remain resident in the printer and be merged with variable data files sent from the host. This provides a very efficient method of printing labels that have a mixture of data fields that change from label to label and elements that remain constant. Refer to Section 8 of this Manual for more information on format files.

Label Vista utilizes a subset of the full Comtec Printer Control Language. Label files created in Label Vista are fully compatible with other labels created using the complete set of Printer Control Language commands.

Label Vista requires an IBM compatable personal computer, running Windows 95 or Windows NT. A PC with the minimum configuration to run Windows 95/NT will have sufficient memory to run Label Vista.

Note: Label Vista has proven to be compatable with Windows 98 in informal testing, however, compatibility problems with certain unusual combinations of hardware and software may arise.

PRINTER COMMANDS

A label file always begins with the "!" character followed by an "x" offset parameter, "x" and "y" axis resolutions, a label length and finally a quantity of labels to print. The line containing these parameters is referred to as the Command Start Line.

A label file always begins with the Command Start Line and ends with the "PRINT" command. The commands that build specific labels are placed between these two commands.

A space character is used to delimit each field in a command line.

Note: Every line in the command session must be terminated with both carriage-return <u>and</u> line-feed characters. All Printer Commands must be in uppercase characters ONLY.

PRINTER Commands

Format:
<!> {offset} <200> <200> {height} {qty}

where:

<!>: Use '!' to begin a control session.

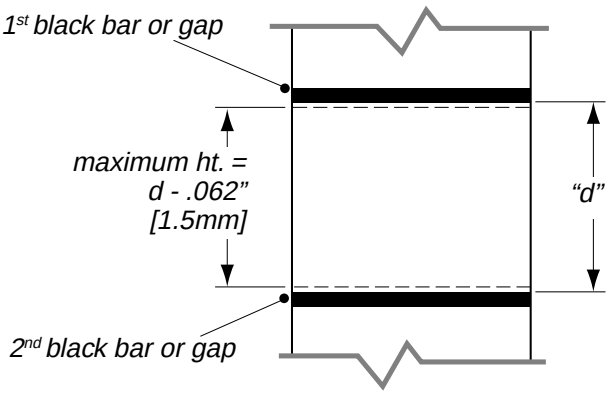
{offset}: The horizontal offset for the entire label. This value causes all fields to be offset horizontally by the specified number of UNITS.

<200>: Horizontal resolution (in dots-per-inch).

<200>: Vertical resolution (in dots-per-inch).

{height}: The maximum height of the label.

The maximum label height is calculated by measuring from the bottom of the first black bar (or label gap) to the top of the next black bar (or label gap). Then 1/16" [1.5mm] is subtracted from this distance to obtain the mximum height. (In dots: subtract 12 dots on 203 d.p.i printers; 18 dots on 306 d.p.i. printers)



{qty}: Quantity of labels to be printed. Maximum = 1024.

PRINTER Command Example

Input (HELLO.LBL):

```
! 0 200 200 210 1
TEXT 4 0 30 40 Hello World
FORM
PRINT
```

Output:

PRINT Command

The PRINT command terminates and prints the file. This must always be the last command (except when in Line Print Mode). Upon execution of the PRINT command, the printer will exit from a control session. Be sure to terminate this and all commands with both carriage-return and line-feed characters.

Format:

{command}

where:

{command}: PRINT

FORM Command

The FORM command instructs the printer to feed to top of form after printing.

Format:

{command}

where:

{command}: FORM

In the following example, the printer will execute a form feed after the label is printed. See the SETFF (set form feed) command for information on setting how the printer will behave when the FORM command is executed.

JOURNAL Command

By default, the printer will check for correct media alignment if it encounters the eye-sense mark (black horizontal bars on back of media) during a print cycle (LABEL mode). If necessary, the JOURNAL command can be used to disable this automatic correction feature. The user's program is responsible for checking and assuring presence of paper in JOURNAL mode. Please refer to the status inquiry command for details on checking for out-of-paper condition.

Format:

{command}

where:

{command}: JOURNAL

UNITS Commands

The units commands are used to specify a measurement system for all subsequent command fields in a control session. Coordinates, widths, and heights for all control commands can be entered with precision to four decimal places. By placing a units command immediately after the first line in a control session, the specified measurement system will also apply to the offset and height fields. The printer measurement system will default to dots until a units command is issued.

Format:

{command}

where:

{command}: Choose from the following:

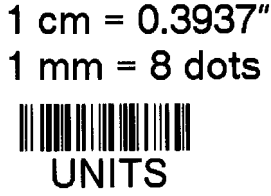
IN-INCHES	Measurement in inches.
IN-CENTIMETERS	Measurement in centimeters.
IN-MILLIMETERS	Measurement in millimeters.
IN-DOTS	Measurement in dots.

UNITS Examples

Input 1 (UNITS1.LBL):

```
! 0.3937 200 200 1 1
IN-INCHES
T 4 0 0 0 1 cm = 0.3937"
IN-DOTS
T 4 0 0 48 1 mm = 8 dots
B 128 1 1 48 16 112 UNITS
T 4 0 48 160 UNITS
FORM
PRINT
```

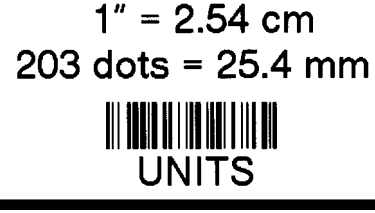
Output 1:



Input 2 (UNITS2.LBL):

```
! 0 200 200 2.54 1
IN-CENTIMETERS
T 4 0 1 0 1" = 2.54 cm
IN-MILLIMETERS
T 4 0 0 6 203 dots = 25.4 mm
B 128 0.125 1 6 12 14 UNITS
T 4 0 16 20 UNITS
FORM
PRINT
```

Output 2:



USING COMMENTS

Comments can be added between the first line of a command session and the “PRINT” command.

A comment is placed in the file by starting a line with the ';' character in the first column. Any remaining text to the end of the line will be ignored. Comments are illegal between the CONCAT and ENDCONCAT commands.

COMMENTS Example

Input (COMMENT.LBL):

```
! 0 200 200 25 1
IN-MILLIMETERS
JOURNAL
; Center justify text
CENTER
; Print the words 'A COMMENT'
TEXT 5 1 0 5 A COMMENT
; Print the label and go to top of next form
FORM
PRINT
```

Output:

RESIDENT FONT SAMPLES

Font#: 0 Size: 0

Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll Mm Nn Oo

$01.23456789¢ $0123456789 $0123456789 $0123

Font#: 0 Size: 1

Aa Bb Cc Dd Ee Ff Gg

$01.23456789¢ $0123

Font#: 0 Size: 2

Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll Mm Nn Oo

$01.23456789¢ $0123456789 $0123456789 $0123

Font#: 0 Size: 3

Aa Bb Cc Dd Ee Ff Gg

$01.23456789¢ $0123

Font#: 0 Size: 4

Aa Bb Cc Dd

$01.2345678¢

Font#: 0 Size: 5

Aa Bb Cc Dd Ee Ff Gg

$01.23456789¢ $0123

Font#: 0 Size: 6

Aa Bb Cc Dd

$01.2345678¢

Font#: 1 Size: 0

Aa Bb Cc Dd Ee Ff Gg

$01.23456789¢ $0123

Font#: 2 Size: 0

A C D

$01.23456789

Font#: 2 Size: 1

A C D

$01.23456789

RESIDENT FONT SAMPLES (continued)

Font#: 4 Size: 0

Aa Bb Cc Dd Ee Ff
$01.23456789¢

Font#: 4 Size: 1

Aa Bb Cc $123

Font#: 4 Size: 2

$120.34

Font#: 4 Size: 3

$120.34

Font#: 4 Size: 4

$120.34

Font#: 4 Size: 5

$120.34

Font#: 4 Size: 6

$120.34

Font#: 4 Size: 7

$120.34

RESIDENT FONT SAMPLES (continued)

Font#: 5 Size: 0

Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj

$01.23456789¢ $01.234567890

Font#: 6 Size: 0

Font#: 5 Size: 1

Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj

$01.23456789¢ $01.234567890

Font#: 7 Size: 0

Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj

$01.23456789¢ $01.234567890

Font#: 5 Size: 2

Aa Bb Cc Dd Ee Ff

$01.23456789¢

Font#: 7 Size: 1

Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj

$01.23456789¢ $01.234567890

Font#: 5 Size: 3

Aa Bb Cc $1.23 10¢

TEXT Commands

The TEXT command is used to place text on a label. This command and its variants control the specific font number and size used, the location of the text on the label, and the orientation of this text. Standard resident fonts can be rotated to 90˚ increments as shown in the example.

Format:

{command} {font} {size} {x} {y} {data}

where:

{command}: Choose from the following:

TEXT (or T)	Prints text horizontally.
VTEXT (or VT)	Prints text (vertically) rotated 90 degrees counterclockwise.
TEXT90 (or T90)	(Same as VTEXT above.)
TEXT180 (or T180)	Prints text (upside down) rotated 180 degrees counterclockwise.
TEXT270 (or T270)	Prints text (vertically) rotated 270 degrees counterclockwise.

{font}: Name/number of the font.

{size}: Size identifier for the font.

{x}: Horizontal starting position.

{y}: Vertical starting position.

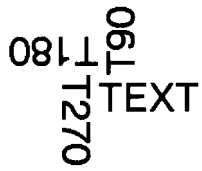
{data}: The text to be printed.

TEXT Example

Input (TEXT.LBL):

```
! 0 200 200 210 1
TEXT 4 0 200 100 TEXT
TEXT90 4 0 200 100 T90
TEXT180 4 0 200 100 T180
TEXT270 4 0 200 100 T270
FORM
PRINT
```

Output:



USING FONT GROUPS

FONT-GROUP (FG) Command

The FG command gives a user the ability to group up to 10 pre-scaled font files into a single group. A user can later specify the font group in a TEXT command. If a font group is used in a text command, the printer will use the *largest* font specified in the font group that will produce the required text data and still remain within the available width of the label for the text. When specified in the TEXT command, the {font} parameter is specified as FG, and the {size} parameter is specified as the {fg}. Note that a user can also specify an FG command within a CONCAT/ENCONCAT command.

Format:

{command} {fg fn fs} [fn fs] ...

where:

{command}: FG

{fg}: Font group number. Up to 10 font groups can be specified. Valid font groups range from 0 to 9.

{fn}: Name/number of the font.

{fs}: Size identifier for the font.

NOTE: Up to 10 font number/font size pairs can be assigned to a font group.

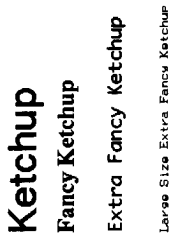
FG Example

In the example, the descriptions will be printed with the largest font in the specified font group that is capable of fitting the requested text in a 250 dot label field.

Input (FG.LBL):

```
! 0 200 200 250 1
; Specify fonts 0-0, 7-0, 5-0, 4-0 as members
; of font group 3.
FG 3 0 0 7 0 5 0 4 0
VT FG 3 10 250 Ketchup
VT FG 3 70 250 Fancy Ketchup
VT FG 3 120 250 Extra Fancy Ketchup
VT FG 3 180 250 Large Size Extra Fancy Ketchup
FORM
PRINT
```

Output:



TEXT CONCATENATION COMMANDS

Text concatenation allows you to assign different character styles to strings, printing them with uniform spacing on the same text line. This command can be used in combination with scalable fonts. See Scalable Concatenation Commands

Format:

{command} {x} {y}
{font} {size} {offset} {data}

" " " "

{font} {size} {offset} {data}
<ENDCONCAT>

where:

{command}: Choose from the following:

CONCAT	Horizontal concatenation.
VCONCAT	Vertical concatenation.

{x}: Horizontal starting position.

{y}: Vertical starting position.

{font}: Name/number of the font.

{size}: Size identifier for the font.

{offset}: Unit-value to offset text from the starting position. Used to align individual text strings or create superscript/subscript characters.

{data}: Text to be printed.

<ENDCONCAT>: Terminates concatenation.

TEXT CONCATENATION Example

Input (CONCAT.LBL):

```
! 0 200 200 210 1
CONCAT 75 75
4 2 5 $
4 3 0 12
4 2 5 34
ENDCONCAT
FORM
PRINT
```

Output:

MULTILINE (ML) Commands

MULTILINE (ML) allows you to print multiple lines of text using the same font and line-height.

Format:

{command} {height}
{text} {font} {size} {x} {y}
{data}
“
{data}
<ENDMULTILINE>

where:

{command}: Choose from the following:

MULTILINE (or ML) Prints multiple lines of text.

{height}: Unit-height for each line of text.

{text}: Text command (TEXT, VTEXT, etc.).

{font}: Name/number of the font.

{size}: Size identifier for the font.

{x}: Horizontal starting position.

{y}: Vertical starting position.

{data}: Text to be printed.

<ENDMULTILINE>: Terminates MULTILINE. *(or ENDML)*

MULTILINE Example

Input (MULTIL.LBL):

```
! 0 200 200 210 1
ML 47
TEXT 4 0 10 20
1st line of text
2nd line of text
:
Nth line of text
ENDML
FORM
PRINT
```

Output:

1st line of text
2nd line of text
:
Nth line of text

COUNT Command

The COUNT command is used for printing multiple labels where a numeric text field or numeric data encoded in a bar code is to be incremented or decremented for each label. The TEXT/BARCODE command string must contain this numeric data as the last characters of the string. The numeric data portion can be up to 20 characters, and can be preceded by the '-' sign. Incrementing or decrementing the numeric data thru '0' is not allowed. Leading zeros will be retained. Up to 3 COUNT commands can be used in a label file.

The numeric data incremented/decremented is contained in the TEXT or BARCODE command that immediately preceded the COUNT command.

Format:

{command} {numeric value}

where:

{command}: COUNT

{numeric value}: Any integer value up to 20 characters. The value can be preceded by a '-' sign if decrementing of the TEXT/BARCODE value is desired. Leading zeros will be retained in the output.

COUNT Example

Input (COUNT.LBL):

```
! 0 200 200 210 3
; Print 3 labels
CENTER
TEXT 4 0 0 50 TESTING 001
COUNT 1
TEXT 7 0 0 100 Barcode Value is 123456789
COUNT -10
BARCODE 128 1 1 50 0 130 123456789
COUNT -10
FORM
PRINT
```

Output:

SETMAG Command

The SETMAG command magnifies a resident font to the magnification factor specified.

Format:

{command} {w} {h}

where:

{command}: SETMAG

{w}: Width magnification of the font.
Valid magnifications are 1 thru 16.

{h}: Height magnification of the font.
Valid magnifications are 1 thru 16.

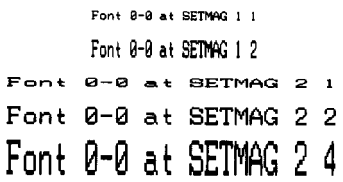
NOTE: The SETMAG command stays in effect after printing a label. This means that the next label printed will use the most recently set SETMAG values. To cancel any SETMAG values and allow the printer to use its default font sizes, use the SETMAG command with magnifications of 0,0.

SETMAG Example

Input (SETMAG.LBL):

```
! 0 200 200 210 1
CENTER
SETMAG 1 1
TEXT 0 0 0 10 Font 0-0 at SETMAG 1 1
SETMAG 1 2
TEXT 0 0 0 40 Font 0-0 at SETMAG 1 2
SETMAG 2 1
TEXT 0 0 0 80 Font 0-0 at SETMAG 2 1
SETMAG 2 2
TEXT 0 0 0 110 Font 0-0 at SETMAG 2 2
SETMAG 2 4
TEXT 0 0 0 145 Font 0-0 at SETMAG 2 4
; Restore default font sizes
SETMAG 0 0
FORM
PRINT
```

Output:



SCALABLE TEXT

Scalable text allows a user to print text at any point size. Point size can be specified for both the X and Y directions to produce characters that are "stretched" in either their width or height. Point sizes specified and text produced will print at 72 points equating to one inch (25.4mm).

The printer can contain scalable font files as part of the application, or scalable font files can be downloaded to the printer using one of the utilities on the supplied disk. A scalable text file must be present in your printer in order to use scalable text features.

SCALE-TEXT Commands

The SCALE-TEXT commands allow the user to specify the point size of *both* the width and height of the font.

Format:

{command} {name} {width} {height} {x} {y} {data}

where:

{command}: Choose from the following:

SCALE-TEXT: Prints scaled text horizontally. (or ST)

VSCALE-TEXT: Prints scaled text vertically. (or VST)

{name}: Font name.

{width}: Font width (point size).

{height}: Font height (point size).

{x}: Horizontal starting position.

{y}: Vertical starting position.

{data}: Text to be printed.

SCALE-TEXT Example

Input (SCALE.LBL):

```
! 0 200 200 300 1
CENTER
; Print using x and y scales of 10 points
SCALE-TEXT PLL_LAT.CSF 10 10 0 10 10 POINT FONT
; Print using x scale of 20 points and y scale
; of 10 points
SCALE-TEXT PLL_LAT.CSF 20 10 0 80 WIDER FONT
; Print using x scale of 10 points and y scale
; of 20 points
SCALE-TEXT PLL_LAT.CSF 10 20 0 150 TALLER FONT
FORM
PRINT
```

Output:

SCALE-TO-FIT Commands

The SCALE-TO-FIT commands automatically calculate the scale in order to fit text inside a window.

Format:

{command} {name} {width} {height} {x} {y} {data}

where:

{command}: Choose from the following:

SCALE-TO-FIT: (or STF) Prints scaled text horizontally.

VSCALE-TO-FIT: (or VSTF) Prints scaled text vertically.

{name}: Font name.

{width}: Unit-width of the window.

{height}: Unit-height of the window.

{x}: Horizontal starting position.

{y}: Vertical starting position.

{data}: Text to be printed.

SCALE-TO-FIT Example

Input (STF.LBL):

```
! 0 200 200 100 1
IN-MILLIMETERS
CENTER
; Fit a text string into an area 40mm wide by 10mm
; high
SCALE-TO-FIT PLL_LAT.CSF 40 10 0 10 SALE
; Fit a longer text string into the same 40mm wide
; by 10mm high area
SCALE-TO-FIT PLL_LAT.CSF 40 10 0 20 SALE PRICE
; Fit “SALE” text into a 40mm wide by 20mm high
; area
SCALE-TO-FIT PLL_LAT.CSF 40 20 0 30 SALE
FORM
PRINT
```

Output:

SCALABLE CONCATENATION Commands

Scalable concatenation allows you to assign different character styles to strings, printing them with uniform spacing on the same text line. Both scalable and bitmap text can be combined between a CONCAT/ENCONCAT command. See also Text Concatenation Commands

Format:

{command} {x} {y}
<ST> {name} {width} {height} {offset} {data}
" " " " " "
<ST> {name} {width} {height} {offset} {data}
<ENDCONCAT>

where:

{command}: Choose from the following:

CONCAT	Horizontal concatenation.
VCONCAT	Vertical concatenation.

{x}: Horizontal starting position.

{y}: Vertical starting position.

{name}: Font name.

{width}: Font width point size.

{height}: Font height point size.

{offset}: Unit-value to offset text from the starting position. Used to align individual text strings or create superscript/subscript characters.

{data}: Text to be printed.

<ENDCONCAT>: Terminates concatenation.

SCALABLE CONCATENATION Example

Input (STC.LBL):

```
! 0 200 200 210 1
CENTER
; Concatenate 3 scalable font strings and 1
; Resident font string
CONCAT 0 20
4 1 0 2/
ST PLL_LAT.CSF 20 20 15 $
ST PLL_LAT.CSF 40 40 0 22
ST PLL_LAT.CSF 20 20 0 99
ENDCONCAT
FORM
PRINT
```

Output:

ROTATE Commands

ROTATE commands are used to rotate all subsequent scalable text fields at a specified angle. Rotation direction is counter-clockwise about the center point of the text. This rotation remains in effect until another ROTATE command is issued. Default angle is zero degrees.

Format:

{command} {angle}

where:

{command}: Choose from the following:

ROTATE (or R)	Rotates scalable fonts.

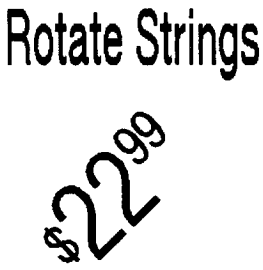
{angle}: Degree of rotation (ccw).

ROTATE Example

Input (ROTATE.LBL):

```
! 0 200 200 440 1
CENTER
TEXT 4 1 0 50 Rotate Strings
ROTATE 45
CONCAT 50 300
ST PLL_LAT.CSF 20 20 20 $
ST PLL_LAT.CSF 40 40 0 22
ST PLL_LAT.CSF 20 20 0 99
ENDCONCAT
FORM
PRINT
```

Output:



LINEAR BAR CODES

Introduction

Bar codes allow easy implementation of automated identification, cataloging and processing of almost any object. They have been successfully used on items ranging in size from boxcars to bumblebees.

This overview of barcode symbologies will help when programming Comtec printers and/or designing labels with Label Vista software. All of the information in this section is derived from the barcode standards established by AIM and the Uniform Code Council as of 11/97. If you plan to create software using these bar codes, we recommend ordering the uniform symbology specification from AIM or the UCC to determine the uses and limitations pertaining to that type of barcode. The information in this document is in no way complete.

The following discussions contain basic information and some suggested applications for each type of barcode. The quick reference table in Appendix E lists specific data for each barcode in one location. All the information on ideal widths and ratios comes directly from the uniform symbology specification. Please note that all measurements contained in this document are in printer dots. On 200 dot per inch (dpi) printers, one dot is equal to 0.005” or 0.13 millimeters, on 300 dpi printers one dot is equal to 0.003” or 0.07 millimeters.

Resident Linear Bar Code Samples

UPC-A

UPC-E

EAN-13

EAN-8

UPC/EAN Plus 2 Extender

UPC/EAN Plus 5 Extender

CODE 39

CODE 93

I 2 OF 5

CODE 128

UCC-128

CODABAR

PLESSY MS110

POSTNET

UPC and EAN/JAN Bar Codes

UPC and EAN/JAN bar codes are typically used for marking products with a unique code used to look up prices and to track inventories of goods sold. They are also used for store coupons, periodicals, and paperback books. UPC and EAN/JAN bar codes are generally rectangular, contain a fixed amount of data, and in most cases are accompanied by human readable text printed below them. For best results, this text should be an OCR-A (resident font 2), a sans serif font (resident font 7) or an OCR-B font.

The first number in the UPC/EAN barcode is the number system character. The specification lists use of characters 0 through 9 as follows.

0 Regular UPC codes (UPC-A and UPC-E)

1 Reserved

2 Random weight items, like store packaged meat. (UPC-A only)

3 National Drug Code and National Health Related Items Code in current 10-digit code length (UPC-A only)*

4 In-store marking of non food items without code format restriction and with check digit protection (UPC-A only)

5 Coupons (UPC-A only)

6 Regular UPC codes (UPC-A only)

7 Regular UPC codes (UPC-A only)

8 Reserved

9 Reserved

* Number system 3 has the following note in the specification. "…the symbol is not affected by the various internal structures possible with the NDC or HRI codes." The users should determine what effect this statement may have on their program. It will not change how bar codes are printed.

The checksum is the last number in the barcode and can be used to make certain that the barcode is decoded properly. This digit is *automatically* calculated by the printer. The UPC barcode specification has the full instructions for calculating this checksum. The methodology is as follows:

For this example, the barcode will be 01234567890.

Step 1: Starting at the left, including the number system character, add up all the numbers in the ODD positions. (0 + 2+ 4 + 6 + 8 + 0 = 20)

Step 2: Multiply this sum by 3. (20 x 3 = 60)

Step 3: Starting at the left again, add up all the numbers in the EVEN positions. (1 + 3 + 5 + 7 + 9 = 25).

Step 4: Add the results from step 2 and step 3. (60 + 25 = 85)

Step 5: The checksum is the smallest number when added to step 4 will equal a multiple of ten. In our example: 85 + 5 = 90, which is a multiple of 10. Therefore, the check digit should be 5. It is called a modulo checksum because you take the modulo, or remainder, of the sum. For the programmers, it is:

10 - (85 mod 10) = the checksum.

UPC-A and EAN13 bar codes can be created with and without a checksum supplied. If the programmer supplies a checksum digit, the printer will create the barcode with that check digit, *even if it is incorrect*. Most laser scanning devices will not be able to decode the barcode if the check digit is incorrect,

UPC-E bar codes, useful for small items like candy and gum, are created through the method of “zero suppression.” For example, if you were to encode 01000000567, the resulting barcode would be a compressed barcode

that only contains the data, the compression scheme, and the checksum without all the extra zeros. For our example, the barcode would decode to 1056707. Please refer to the UPC Symbol Specification Manual from the Uniform Code Council for more information on zero suppression.

UPC-E and EAN8 bar codes have a few restrictions. First, the number system character must be set to 0. Number systems 1 through 9 do not support UPC-E and EAN8 bar codes and may not be decoded by a laser scanning device. In case your application requires it, the number system may be set to something other than 0. Second, if the programmer supplies a checksum digit, the printer will create the barcode with that check digit, *even if it is incorrect*. If the check digit is incorrect, most laser scanning devices will not be able to decode the barcode. Therefore, the programmer may send six digits (no number system, no checksum), seven digits (number system, no checksum), or eight digits (number system and checksum) and create a barcode.

Plus 2 and plus 5 barcode extensions are only used for periodicals and paperback books. Specifically, the bar code specification states that the plus 2 extension should only be used for a periodical issue number. The plus 2 and plus 5 extensions do not contain any checksum according to the barcode specification.

To create an extended barcode, place a space between the data that should go into the UPC/EAN barcode and the data that should go into the extension. You can also use the PLUS2 and PLUS5 barcode types to create the extension separately. Remember to leave ample space (about 9 times the ratio) between the UPC/EAN barcode and the extension.

UPC/EAN Specifications

Barcode Symbology	Barcode Type	Input Length	Characters	Ideal Wide/Narrow Ratio	Ideal Narrow Dot Width	Checksum Calculation
UPC-A	UPCA	11 or 12 digits	0-9 only	2:1	2	mod 10
UPC-A plus 2	UPCA2	13 digits	0-9 only	2:1	2	mod 10 (UPC-A)
UPC-A plus 5	UPCA5	16 digits	0-9 only	2:1	2	mod 10 (UPC-A)
UPC-E	UPCE	6, 7 or 11 digits	0-9 only	2:1	2	mod 10
UPC-E plus 2	UPCE2	8 or 13 digits	0-9 only	2:1	2	mod 10 (UPC-E)
UPC-E plus 5	UPCE5	11 or 16 digits	0-9 only	2:1	2	mod 10 (UPC-E)
EAN/JAN-13	EAN13	12 or 13 digits	0-9 only	2:1	2	mod 10
EAN/JAN-13 plus 2	EAN132	14 digits	0-9 only	2:1	2	mod 10 (EAN13)
EAN/JAN-13 plus 5	EAN135	17 digits	0-9 only	2:1	2	mod 10 (EAN13)
EAN/JAN-8	EAN8	6, 7 or 8 digits	0-9 only	2:1	2	mod 10
EAN/JAN-8 plus 2	EAN82	9 digits	0-9 only	2:1	2	mod 10 (EAN8)
EAN/JAN-8 plus 5	EAN85	12 digits	0-9 only	2:1	2	mod 10 (EAN8)

Code 39 or Code 3 of 9 Bar Codes

The Code 39 barcode is used for many applications including inventories, hospital applications, or any other place where the code length may change between items being scanned (e.g. a bar code stating there were 420 pieces in one box and 20004 pieces in another box would have a different physical length). This barcode can use the characters 0 through 9, A through Z, '-' (dash), "." (period), space, "$" (dollar sign), "/"(forward slash), "+" (plus) and "%" (percent). There is also a special character called "S/S" used as a start/ stop character. The F39 and F39C types allow the use of carriage return, line feed, and null characters.

The checksum for this barcode is located as the last (or least significant) digit of the decoded barcode. To assure data integrity in your application, use a barcode with a checksum. The printer will *automatically* supply this digit

if the user selects a 39C or a F39C barcode.

CheckCharacterNumericalValueTable

Char	Value	Char	Value	Char	Value	Char	Value
0	0	C	12	O	24	-	36
1	1	D	13	P	25	.	37
2	2	E	14	Q	26	SPACE	38
3	3	F	15	R	27	$	39
4	4	G	16	S	28	/	40
5	5	H	17	T	29	+	41
6	6	I	18	U	30	%	42
7	7	J	19	V	31	$ (full)	43*
8	8	K	20	W	32	% (full)	44*
9	9	L	21	X	33	/(full)	45*
A	10	M	22	Y	34	+ (full)	46*
B	11	N	23	Z	35		

* Full represents F39 or F39C for Full ASCII

Refer to the full barcode symbology specification for complete information on checksum calculation. For a short example, take an example barcode with the data “CODE 39” .

Step 1: Assign a value to each character per the Character Numerical Value Table above . C=12, O=24, D=13, E=14, space = 38, 3=3, 9=9.

Step 2: Add the values 12+24+13+14+38+3+9=113.

Step 3: Divide this number by 43. The remainder or modulo, 27, is the checksum.

Step 4: Referring to the table, 27 is the character R. Therefore, the checksum in the barcode should be R. The final code reads “CODE 39R” when it is decoded.

Code 39 (3 of 9) Specifications

Barcode Symbology	Barcode Type	Input Length	Characters	Ideal Wide/Narrow Ratio	Ideal Narrow Dot Width	Checksum Calculation
Code 39	39	Variable	Refer text	2.5:1	2	none
	39C	Variable	Refer text	2.5:1	2	mod 43
	F39	Variable	Refer text	2.5:1	2	none
	F39C	Variable	Refer text	2.5:1	2	mod 43

Code 93 or Code 9 of 3 Bar Codes

The Code 93 barcode is used for applications that require heavy error checking capabilities. To accomplish this, the Code 93 barcode contains two separate error checking checksums that are automatically calculated and placed into the barcode. This barcode is used for inventories, hospital applications, or any other place where the length may change between items being scanned. (See Code 39 above.) This barcode type can use the entire ASCII 128 character set. It is useful for encoding data and phrases like “Code 93”.

The two checksums in this barcode are located as the last and second to last characters in the decoded bar code. Code 93 has a complex checksum calculation. Please see the barcode symbology specification for information on how to create and decode this checksum. Please also note that the barcode symbology specification does not state any ideal values for the ratio and the width of the narrow bar.

Code 93 Specifications

Barcode Symbology	Barcode Type	Input Length	Characters	Ideal Wide/Narrow Ratio	Ideal Narrow Dot Width	Checksum Calculation
Code 93	93	Variable	128 ASCII	1.5:1	1	two mod 47

Interleaved 2 of 5 Bar Codes

The Interleaved 2 of 5 (or ITF) Barcode is used for applications that have a fixed data length for all items scanned. A date, telephone number, or a SKU of fixed length would be a good application for this barcode. The symbology specification states that a ITF barcode may be partially decoded without any recognizable difference. Therefore, to prevent this problem, you must keep the length of data to a constant and perform an error checking routine on the decoding program to determine if the data is correct.

Only the digits 0-9 can be encoded, and there should

be an even number of digits in the data. If there is an odd number of digits, the printer will automatically insert a zero (0) at the beginning of the barcode.

There is no checksum automatically added to the bar code. You can, however, manually calculate a checksum and append it as the last digit (or least significant digit) in the barcode.

Step 1: To calculate the checksum, first ensure that you are starting with an odd number of digits in the data. If not, add a zero (0) to the beginning of the data.

Step 2: Multiply every other digit by 3, and add up the numbers. So, if your data was "43827" your calculation should be (4 x 3) + 3 + (8 x 3) + 2 + (7 x 3) = 62.

Step 3: Divide this number by 10, resulting in 6 with a remainder of 2. Subtract the remainder from 10. In our example, 10 - 2 = 8. The checksum is this final number, 8. Append this to the end of your data. Note that if the remainder was a zero, your checksum should be zero.

Interleaved 2 of 5 Specifications

Barcode Symbology	Barcode Type	Input Length	Characters	Ideal Wide/Narrow Ratio	Ideal Narrow Dot Width	Checksum Calculation
Interleaved 2 of 5	I2OF5	Varies	0-9 only	2.5:1	2	See text

Code 128 and the UCC-128 Shipping Standard

Code 128 is used for applications that need to contain a large amount of data such as shipping applications, marking blood donations, and any other application that can vary in length between bar codes being scanned. The barcode also contains a checksum as the last character in the code which ensures that data remains intact.

Code 128 can use the entire ASCII 128 character set as well as other subsets available in the universal symbology specification. The three start and stop characters

determine which character set to use. The checksum for this barcode is located immediately before the stop character. The barcode symbology specification contains all the information on calculating this checksum. For a short example, we desire to encode “BAR128” in a barcode. We will use “A” as our start and stop character in this example.

Step 1: The symbology specification assigns a numerical value for each character. Find the values of all the characters in the data.

Step 2: Add the value of the start character and all the data characters multiplied by its position in the bar code. For our example, the calculation would be 103 + (34 x 1) + (32 x 2) + (50x 3) + (17x 4) + (18x 5) + (24 x 6) = 672.

Step 3: Divide this number by 103. The remainder or modulo, 54, is the checksum. The character that assigned to 54 in the specification is “V”. Our final code will look like “ABAR128V” where “A” is the start character, “BAR128” is the data, and “V” is the checksum.

The UCC-128 Shipping Standard is part of a document called *Application Standard for Shipping Container Codes* available from the Uniform Code Council. This 90-page guide contains the entire specification on marking any shipment sent anywhere in the United States. Seventeen pages are dedicated to the technical considerations of using, placing, and printing these bar codes. We highly recommend getting this information if your application involves shipping.

Code 128/ UCC-128 Specifications

Barcode Symbology	Barcode Type	Input Length	Characters	Ideal Wide/Narrow Ratio	Ideal Narrow Dot Width	Checksum Calculation
Code 128 /A/B/C/Auto	128	Variable	Refer text	1.5:1	1	mod 103
UCC-128Std.	UCCEAN 16	Refer text	Refer text	3:1	3	mod 103

Codabar

Codabar is ideal for applications that contain mostly numeric symbols that may vary in length from barcode to barcode. It can encode the digits from 0 to 9, the characters "-"(dash), "$"(dollar sign), ":" (colon), "/" (forward slash), "." (period), and "+" (plus) as well as start/stop characters A through D.

One optional checksum is automatically appended as the least significant digit in the barcode data directly before the stop character. The barcode symbology specification contains all the information on calculating the checksum. As a short example, our data will be "A37859B" where A and B are start/stop characters. The characters 0 through 9 are assigned the numerical values 0 through 9 respectively. "-" is 10, "$" is 11, ":"is 12, "/" is 13, "." is 14, "+" is 15, and start/stop characters A B C and D are 16, 17, 18, and 19 respectively.

Step 1: Add the numerical value of all the characters. 16 + 3 + 7 + 8 + 5 + 9 + 17 = 65.

Step 2: Divide this number by 16 and use the remainder, or modulo. In our example, this is 1

Step 3: Subtract the modulo from 16. This is the smallest number that can be added to the sum in step 1 to make a multiple of 16. (65 + 15 =80. 80 /16= 5) Therefore, the check sum for our example is 15.

Step 4: The character that corresponds to 15 is "+" and is added in before the stop character. Our final bar code looks like "A37859+B".

The barcode type NW7 is for reverse compatibility only. We do not recommend using this command for new systems. There is no difference between CODABAR and NW7.

Codabar Specifications

Barcode Symbology	Barcode Type	Input Length	Characters	Ideal Wide/Narrow Ratio	Ideal Narrow Dot Width	Checksum Calculation
Codabar	CODABAR	Variable	0-9,A-D symbol	2.5:1	2	none
	CODABAR 16	Variable	0-9,A-D, symbol	2.5:1	2	mod 16

MSI Plessey Bar Codes

The MSI Plessey barcode is a fixed length code that uses only numerical characters. It is primarily used for grocery applications. Three different types of encoding exist with different levels of data protection. Please see the bar code symbology specification for more information on how to calculate these checksums.

The barcode type “PLESSEY” is used for reverse compatibility only. We do not recommend using this command for new systems. The PLESSEY type will force a 2:1 ratio of the wide to narrow bar width.

MSI Plessy Specifications

Barcode Symbology	Barcode Type	Input Length	Characters	Ideal Wide/Narrow Ratio	Ideal Narrow Dot Width	Checksum Calculation
MSI Plessey	MSI	13 digits max	0-9 only	2:1	2	none
	MSI10	13 digits max	0-9 only	2:1	2	mod 10
	MSI1010	13 digits max	0-9 only	2:1	2	two mod 10
	MSI1110	13 digits max	0-9 only	2:1	2	mod 11 mod 10

Postnet and Facing Identification Marks

The US Postnet barcode is used only to help automate mail delivery. To comply with postal regulations, set the height of the barcode to 30 dots, the wide/narrow bar ratio at 3.5:1, and the width of the narrow bar to 3 dots on a 200 dpi printer. The data sent to the barcode can be 5, 9, or 11 digits long. For example, to send mail to 30 Plan Way, Warwick, RI 02886-1234, the data would be

5 digits- ZIP Code only: 02886

9 digits - ZIP + 4 Code: 028861234

11 digits - ZIP + 4 Code and last two digits in address: 02886123430

The Postnet barcode also contains an automatically calculated checksum as the last character in the decoded barcode. As a short example, our data will be “02881123430”

Step 1: Add the numerical value of all the characters. 0+2+8+8+1+1+2+3+4+3+0 = 32.

Step 2: Divide this number by 10 and use the remainder, or modulo. In our example, this is 2

Step 3: Subtract the remainder (or modulo) from 10 to get the check sum. The check sum for our example would be 8 (10 - 32 mod 10 for programmers).

A Facing Identification Mark (FIM) is the bar in the upper right corner of an envelope near the stamp. To comply with postal regulations, set the height of the barcode to 125 dots, the ratio to 1.5:1 dots, and the width of the narrow bar to 6 dots. There are three different characters you can send as data: A, B, and C.

FIM A: Courtesy Reply Mail with Postnet Barcode

FIM B: Business Reply Mail, Penalty Mail, or Franked

Mail without Postnet Barcode

FIM C: Business Reply Mail, Penalty Mail, or Franked Mail with Postnet Barcode.

For more information, please see Publication 25 from the USPS Postal Business Center. If you are making a label with an address, try using resident font 7 or font 4 for best results with the optical character recognition software used by the post office.

Postnet and FIM Specifications

Barcode Symbology	Barcode Type	Input Length	Characters	Ideal Wide/Narrow Ratio	Ideal Narrow Dot Width	Checksum Calculation
Postnet	POSTNET	5, 9, 11 digits	0-9 only	3.5:1	3	mod 10
Facing Ident Mark	FIM	A, B, or C only	A, B, or C	1.5:1	6	N/A

BAR CODE COMMANDS

The following commands are used for the creation and formatting of bar codes on labels. Insure that the bar code symbology chosen agrees with its intended use, and that it conforms to the guidelines in the previous section.

A Quick Reference Guide for the linear bar code symbologies discussed in this manual can be found in Appendix “E”.

Note that the “COUNT” command is also discussed in Section 3 of this manual.

BARCODE Commands

The BARCODE command prints bar codes in both vertical and horizontal orientations at specified widths and heights.

Standard Bar Codes

Format:

{command} {type} {width} {ratio} {height} {x} {y} {data}

where:

{command}: Choose from the following:

BARCODE (or B) Prints bar code horizontally.

VBARCODE (or VB) Prints bar code vertically.

{type}: Choose from the following table:

Symbology:	use:
UPC-A	UPCA, UPCA2, UPCA5
UPC-E	UPCE, UPCE2, UPCE5
EAN/JAN-13	EAN13, EAN132, EAN135
EAN/JAN-8	EAN8, EAN82, EAN 85
Code 39	39, 39C, F39, F39C
Code 93/Ext. 93	93
Interleaved 2 of 5	I2OF5
Code 128 (Auto)	128
UCC EAN 128	UCCEAN128
Codabar	CODABAR, CODABAR16
MSI/Plessy	MSI, MSI10, MSI1010, MSI1110
Postnet	POSTNET
FIM	FIM

{width}: Unit-width of the narrow bar.

{ratio}: Ratio of the wide bar to the narrow bar. Refer to the table in Appendix “E” for appropriate settings.

0 = 1.5 : 1	20 = 2.0:1	26 = 2.6:1
1 = 2.0 : 1	21 = 2.1:1	27 = 2.7:1
2 = 2.5 : 1	22 = 2.2:1	28 = 2.8:1
3 = 3.0 : 1	23 = 2.3:1	29 = 2.9:1
4 = 3.5 : 1	24 = 2.4:1	30 = 3.0:1
	25 = 2.5:1	

Note: The ratios in the Appendix are suggested for best results, however any ratio can be assigned.

{height}: Unit-height of the bar code.

{x}: Horizontal starting position.

{y}: Vertical starting position.

{data}: Bar code data.

BAR CODE Example

Input (BARCODE.LBL):

```
! 0 200 200 210 1
BARCODE 128 1 1 50 150 10 HORIZ.
TEXT 7 0 210 60 HORIZ.
VBARCODE 128 1 1 50 10 200 VERT.
VTEXT 7 0 60 140 VERT.
FORM
PRINT
```

Output:

BARCODE-TEXT Command

The BARCODE-TEXT command is used to label bar codes with the same data used to create the bar code. The command eliminates the need to annotate the bar code using separate text commands. The text will be centered below the bar code.
Use BARCODE-TEXT OFF (or BT OFF) to disable.

Format:

{command} {font number} {font size} {offset}

where:

{command}:	BARCODE-TEXT (or BT)
{font number}:	The font number to use when annotating the bar code.
{font size}:	The font size to use when annotating the bar code.
{offset}:	Unit distance to offset text away from the bar code.

BARCODE-TEXT Example

Input (BARTEXT.LBL):

```
! 0 200 200 400 1
JOURNAL
CENTER
; Annotate bar codes using font 7 size 0
; and offset 5 dots from the bar code.
BARCODE-TEXT 7 0 5
BARCODE 128 1 1 50 0 20 123456789
VBARCODE 128 1 1 50 40 400 112233445
BARCODE-TEXT OFF
FORM
PRINT
```

Output:

COUNT Command

The COUNT command is used for printing multiple labels where a numeric text field or numeric data encoded in a bar code is to be incremented or decremented for each label. The TEXT/BARCODE command string must contain this numeric data as the last characters of the string. The numeric data portion can be up to 20 characters, and can be preceded by the '-' sign. Counts of 9 → 0 will wrap to 9. Counts of 00 → 99 will wrap to 00. Leading zeros will be retained. Up to 3 COUNT commands can be used in a label file.

The numeric data incremented/decremented is contained in the TEXT or BARCODE command that immediately preceded the COUNT command.

Format:

{command} {numeric value}

where:

{command}: COUNT

{numeric value}: Any integer value up to 20 characters. The value can be preceded by a '-' sign if decrementing of the TEXT/BARCODE value is desired. Leading zeros will be retained in the output.

COUNT Example

Input (COUNT.LBL):

```
! 0 200 200 210 3
; Print 3 labels
CENTER
TEXT 4 0 0 50 TESTING 001
COUNT 1
TEXT 7 0 0 100 Barcode Value is 123456789
COUNT -10
BARCODE 128 1 1 50 0 130 123456789
COUNT -10
FORM
PRINT
```

Output:

TWO-DIMENSIONAL BAR CODES

Introduction

A two dimensional barcode can be regarded as a "portable database." For example, if a package has a serial number encoded with a linear barcode, you could scan the serial number, look up the number in a computer system, and gather the information about that package. If the computer system was unavailable for any reason the information you were looking for would also be unavailable A 2D barcode, however, can contain several different fields of information in them, essentially a database attached to a package.

The first 2D barcode was just an extension of a one-dimensional barcode. The user could stack several Code 39 barcodes on top of another. These types of codes are called "stacked symbology" or "multi-row codes". Development of scanners capable of 2D scanning allowed use of more compact and useful symbologies like PDF417 from Symbol and MaxiCode from the United Parcel Service.

Comtec printers have capabilities to print PDF417 and MaxiCode symbols. If your application requires the use of PDF417 or MaxiCode symbols, we highly recommend acquiring the universal symbol specification to assist your programming. The specifications include suggestions on how to structure your data in the code to make it easier to use.

Please note that you must load a special application before using PDF417 or MaxiCode on your Comtec printer. The standard application loaded in your printer does not support the ability to print these barcodes. Contact the Help Desk for more information on how to load this application.

PDF417

The PDF417 barcode is a 2D barcode that can contain a very large amount of data in a small space. If you look at a PDF417 barcode carefully, it is actually a stack of smaller barcodes. The number and height of the stacks are controllable by the user. The barcode can contain the entire ASCII 255 set of characters, and has the capability to use different encoding schemes and different levels of security to correct errors. The largest amount of data that can be encoded is 2725 characters

MaxiCode

MaxiCode was originally designed by the United Parcel Service to help them automate package delivery and sort packages coming down a quick conveyor belt. It consists of a bulls eye to help the imaging system “target” the barcode and an array of hexagons to represent the data stored in the barcode. It can contain a maximum of 93 alphanumeric characters or 138 numeric characters, has two different error correction modes, seven different storage modes, the capability to use character sets other than ASCII, and the capability to “link” several MaxiCodes together. Since there are so many different modes, it is suggested that you contact AIM for the symbol specification. If you are developing software to be used with the UPS MaxiCode shipping system, contact UPS for information on how to order the information on the Maxicode shipping system.

TWO-DIMENSIONAL BAR CODE COMMANDS

The following commands are used for the creation and formatting of two-dimensional bar codes. Insure that the bar code symbology chosen agrees with its intended use. We recommend acquiring the specification for the desired code to supplement the discussions in this manual.

Note that Codablock “A” and “F” are not fully supported in this release of the manual.

PDF417 (PORTABLE DATA FILE)

Format:

{command} {type} {x} {y} [XD n] [YD n] [C n] [S n]
{data}
<ENDPDF>

where:

{command}: Choose from the following:

BARCODE (or B) Prints bar code horizontally.

VBARCODE (or VB) Prints bar code vertically.

{type}: PDF-417

{x}: Horizontal starting position.

{y}: Vertical starting position.

[XD n]: Unit-width of the narrowest element.
Range is 1 to 32.
Default is 2.

[YD n]: Unit-height of the narrowest element.
Range is 1 to 32.
Default is 6.

[C n]: Number of columns to use. Data columns do not include start/stop characters and left/right indicators.
Range is 1 to 30.
Default is 3.

PDF 417 (PORTABLE DATA FILE) continued

(continued)

[S n]: Security level indicates maximum amount of errors to be detected and/or corrected.
Range is 0 to 8.
Default is 1.

{data}: Bar code data.

<ENDPDF>: Terminates PDF-417.

Note: The BARCODE-TEXT command does not work with the PDF-417 barcode type. Any desired human readable text must be entered seperately with the TEXT command as in the example below.

PDF417 Example

Input (PDF.LBL):

```
! 0 200 200 210 1
B PDF-417 10 20 XD 3 YD 12 C 3 S 2
PDF Data
ABCDE12345
ENDPDF
T 4 0 10 120 PDF Data
T 4 0 10 170 ABCDE12345
FORM
PRINT
```

Output:

PDF Data
ABCDE12345

MAXICODE

Revised: Application Version 25 or higher

The Maxicode bar code now handles all the symbols defined by the United Parcel Service as well as the basic fields supported in the standard bar code. Maxicode supports all standard printable characters with automatic conversion of all lowercase letters in the secondary message to uppercase. This revision supports only Mode 2 bar codes.

Format:
{command} {type} {x} {y}
{tag} {options}
…
{tag} {options}
<ENDMAXICODE>

where:

{command}:	BARCODE or B	Prints bar code.
{type}:	MAXICODE	
{x}:	Horizontal starting position	
{y}:	Vertical starting position	
{tag}:	Tags not supplied will be filled with default values. Use only the tags that you require. Tags can be in any order.	

<ENDMAXICODE> Final tag in Maxicode barcode.

Tags encoded in the high priority message of all Maxicodes:

Tag	Definition	Default Value
POST	Postal or ZIP Code	-empty-
CC	Country Code (from ISO 3166)	840 (USA)
SC	Service Class	1

Tags to control the type of barcode created:

Tag	Definition	Default Value
UPS5	Use UPS5 tags to create the low priority message. (On: 1, Off: 0)	0
ZIPPER	Turn the zipper and contrast patterns on or off. (On: 1, Off: 0)	0
FILLC	Low priority message fill character (Messages shorter than 84 characters will be padded with this character.)	!

Tags used when UPS5 is turned off:

Tag	Definition	Default Value
MSG	Low priority message field (maximum of 84 characters, overwritten by UPS5 tags)	-empty-

Tags used when UPS5 is turned on:

Tag	Definition	Default Value
LPMS	Low priority message header	[)>[RS]
HEAD	Transportation data format header	01[GS]98
TN	Tracking Number	[GS]
SCAC	Standard Carrier Alpha Code	UPSN
SHIPPER	UPS Shipper Number	[GS]
PICKDAY	Julian day of pickup	[GS]
SHIPID	Shipment ID Number	[GS]
NX	Package N of X (n/x)	[GS]
WEIGH	Package weight	[GS]
VAL	Address validation (Y or N)	[GS]
STADDR	Ship to street address	[GS]
CITY	Ship to city	[GS]
ST	Ship to state	[GS]
EXTRA	Extra user defined fields	- empty -
EOT	End of transmission character	0x004h
GS	Field separator character [GS]	0x01Dh
RS	Format type separator [RS]	0x01Eh

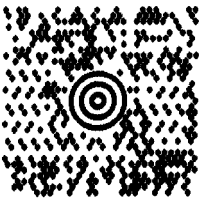
Please see the document "*Guide to Bar Coding with UPS OnLine: for Customers Generating Bar Code Labels, Version 5*" available from the United Parcel Service for more information on creating labels for the UPS shipping system.

MAXICODE Examples

Basic example with minimal required fields to print a barcode:

```
! 0 200 200 600 1
JOURNAL
B MAXICODE 20 20
CC 12345
MSG This is a MAXICODE low pri-
ority message.
SC 12345
POST 02886
ENDMAXICODE
PRINT
```

Output:

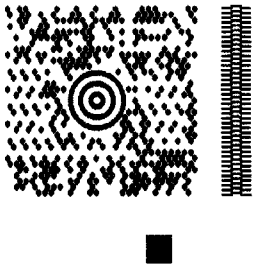


Decodes to:
028860000[GS]057[GS]057[GS]This is a MAXICODE low priority message.

Same example with zipper and contrast patterns:

```
! 0 200 200 600 1
JOURNAL
B MAXICODE 20 20
CC 12345
MSG This is a MAXICODE low pri-
ority message.
SC 12345
POST 02886
ZIPPER 1
ENDMAXICODE
PRINT
```

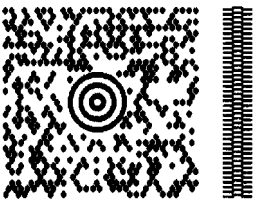
Output:



Example using tags for UPS:

```
! 0 200 200 600 1
JOURNAL
B MAXICODE 20 20
VAL Y
STADDR 30 PLAN WAY
WEIGH 210
SHIPID 42
PICKDAY 193
SHIPPER 12345
TN 1Z12345675
CC 860
SC 1
POST 02886
ZIPPER 1
SHIPPER 12345E
NX 1/2
UPS5 1
CITY WARWICK
ST RI
ENDMAXICODE
PRINT
```

Output:



Decodes to:

[)>[RS]01[GS]98028860000[GS]860[GS]001[GS]1Z12345675[GS]USPN[GS]12345E[GS]193[GS]42[GS]1/2[GS]210[GS]Y[GS]30 PLAN WAY[GS]WARWICK[GS]RI[RS][EOT]

GRAPHICS

BOX Command

The BOX command provides the user with the ability to produce rectangular shapes of specified line thickness.

Format:

{command} *{x_0}* *{y_0}* *{x_1}* *{y_1}* *{width}*

where:

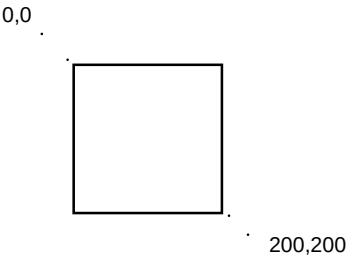
{command}:	BOX
{x_0}:	X-coordinate of the top left corner.
{y_0}:	Y-coordinate of the top left corner.
{x_1}:	X-coordinate of the bottom right corner.
{y_1}:	Y-coordinate of the bottom right corner.
{width}:	Unit-width (or thickness) of the lines forming the box.

BOX Example

Input (BOX.LBL):

```
! 0 200 200 210 1
BOX 0 0 200 200 1
FORM
PRINT
```

Output:



Note: Text coordinates (in output) are shown for illustration purposes only.

LINE Commands

Lines of any length, thickness, and angular orientation can be drawn using the LINE command.

Format:

{command} {x_0} {y_0} {x_1} {y_1} {width}

where:

{command}: Choose from the following:

LINE (or L)	Prints a line.

{x_0}: X-coordinate of the top-left corner.

{y_0}: Y-coordinate of the top-left corner.

{x_1}: X-coordinate of:
- top right corner for horizontal.
- bottom left corner for vertical.

{y_1}: Y-coordinate of:
- top right corner for horizontal.
- bottom left corner for vertical.

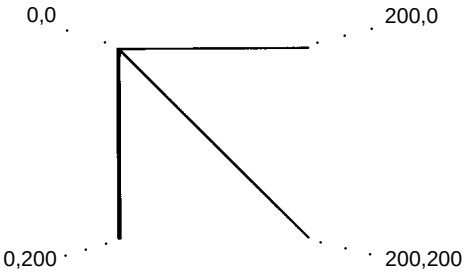
{width}: Unit-width (or thickness) of the line.

LINE Example

Input (LINE.LBL):

```
! 0 200 200 210 1
LINE 0 0 200 0 1
LINE 0 0 200 200 2
LINE 0 0 0 200 3
FORM
PRINT
```

Output:



Note: Text coordinates (in output) are shown for illustration purposes only.

INVERSE-LINE Commands

The INVERSE-LINE command has the same syntax as the LINE command. Previously created objects that lie within the area defined by the INVERSE-LINE command will have their black areas re-drawn white, and white areas re-drawn black. These objects can include text, barcodes and/or graphics, including downloaded .pcx files. INVERSE-LINE has no effect on objects created after its location, even if they fall within its covered area. In example INVERSE2.LBL, portions of the text field created after the INVERSE-LINE command remain black, hence invisible, even though placed within the IN-VERSE-LINE area.

Format:

{command} {x_0} {y_0} {x_1} {y_1} {width}

where:

{command}: Choose from the following:

INVERSE-LINE (or IL)	Prints a line over an existing field to invert the image.

{x_0}: X-coordinate of the top-left corner.

{y_0}: Y-coordinate of the top-left corner.

{x_1}: X-coordinate of:
- top right corner for horizontal.
- bottom left corner for vertical.

{y_1}: Y-coordinate of:
- top right corner for horizontal.
- bottom left corner for vertical.

{width}: Unit-width (or thickness) of the inverse-line.

INVERSE-LINE Examples

Input 1 (INVERSE1.LBL):

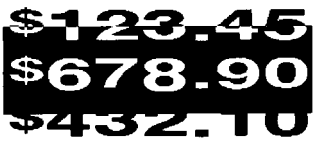
```
! 0 200 200 210 1
CENTER
TEXT 4 0 0 45 SAVE
TEXT 4 0 0 95 MORE
INVERSE-LINE 0 45 145 45 45
INVERSE-LINE 0 95 145 95 45
FORM
PRINT
```

Output 1:

Input 2 (INVERSE2.LBL):

```
! 0 200 200 210 1
T 4 2 30 20 $123.45
T 4 2 30 70 $678.90
IL 25 40 350 40 90
T 4 2 30 120 $432.10
FORM
PRINT
```

Output 2:



PATTERN Command

The PATTERN command is used with the LINE and SCALE-TEXT commands to change the patterns used to fill these shapes. Valid pattern values are listed below.

Format:

{command} {pattern number}

where:

{command}: PATTERN

{pattern number}: Choose from the following:

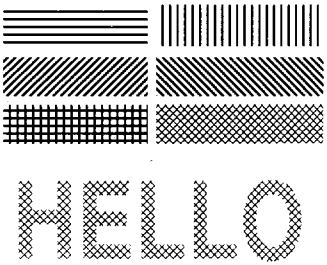
100	Filled (solid black/default pattern).
101	Horizontal lines.
102	Vertical lines.
103	Right rising diagonal lines.
104	Left rising diagonal lines.
105	Square pattern.
106	Cross hatch pattern.

PATTERN Example

Input (PATTERN.LBL):

```
! 0 200 200 700 1
; Draw horizontal and vertical patterns
PATTERN 101
LINE 10 10 160 10 42
PATTERN 102
LINE 170 10 350 10 42
; Draw left and right diagonal patterns
PATTERN 103
LINE 10 65 160 65 40
PATTERN 104
LINE 170 65 350 65 40
; Draw square and cross hatch patterns
PATTERN 105
LINE 10 115 160 115 40
PATTERN 106
LINE 170 115 350 115 40
; Draw a scalable text character with cross
; hatch pattern
PATTERN 106
ST PLB_LAT.CSF 40 40 20 180 HELLO
FORM
PRINT
```

Output:



GRAPHICS Commands

Bit-mapped graphics can be printed by using graphics commands. ASCII hex (hexadecimal) is used for expanded graphics data (see example). Data size can be reduced to one-half by utilizing the COMPRESSED-GRAPHICS commands with the equivalent binary character(s) of the hex data.

Format:

{command} {width} {height} {x} {y} {data}

where:

{command}: Choose from the following:

EXPANDED-GRAPHICS (or EG)	Prints expanded graphics horizontally.
VEXPANDED-GRAPHICS (or VEG)	Prints expanded graphics vertically.
COMPRESSED-GRAPHICS (or CG)	Prints compressed graphics horizontally.
VCOMPRESSED-GRAPHICS (or VCG)	Prints compressed graphics vertically.

{width}: Byte-width of image.

{height}: Dot-height of image.

{x}: Horizontal starting position.

{y}: Vertical starting position.

{data}: Graphics data.

GRAPHICS Example

Input (GRAPHIC.LBL):

```
! 0 200 200 210 1
EG 2 16 90 45 F0F0F0F0F0F0F0F00F0F0F0F0F0F0F0F
F0F0F0F0F0F0F0F00F0F0F0F0F0F0F0F
FORM
PRINT
```

Output:

Note: Graphic output has been magnified. Actual size is 1/4 of output shown.

PCX Commands

The PCX command gives a user the ability to send ".PCX" graphics formatted images to the printer. The .PCX image MUST be encoded as a black and white image.

Format:

{command} {x} {y}
{data}

where:

{command}: PCX

{x}: X-coordinate of the top-left corner.

{y}: Y-coordinate of the top-left corner.

{data}: PCX image data.

PCX Examples

EXAMPLE 1

In the example below, the image is sent in three steps. First, the printer is sent commands to expect a .PCX formatted file. The second input to the printer is the .PCX image. This image must be a 2 color (black and white) image. The last step is to tell the printer to print the label.

Input 1 (STARTPCX.LBL):

```
! 0 200 200 500 1
PCX 0 30
```

Input 2 (IMAGE.PCX)

Input 3 (ENDPCX.LBL)

```
FORM
PRINT
```

Output:

EXAMPLE 2

In this example , the PCX image has been loaded into the printer's flash file system and given the name "IMAGE.PCX". The "!<" operator can now be used to instruct the printer to get the data stored in the file "Image.PCX" and use it for building the image.

Input 2 (IMAGE.LBL):

```
! 0 200 200 500 1
PCX 0 30 !<IMAGE.PCX
FORM
PRINT
```

Output:

ADVANCED COMMANDS

CONTRAST Command

The CONTRAST command is used to specify the print darkness for the entire label. The lightest printout is at contrast level 0. The darkest contrast level is 3. The printer defaults to contrast level 0 on power up. Contrast level must be specified for each label file.
In order to maximize printing efficiency, always use the lowest contrast level possible .

Format:

{command} {level}

where:

{command}: CONTRAST

{level}: Contrast level.

0 = Default
1 = Medium
2 = Dark
3 = Very Dark

TONE Command

The TONE Command can be used instead of the CONTRAST command to specify the print darkness for all labels. The lightest printout is at tone level -99. The darkest tone level is 200. The printer defaults to tone level 0 on power up. Tone level settings remain in effect for all printing tasks until changed. The TONE and CONTRAST commands *cannot* be used in combination with one another.

Format:

{command} {level}

where:

{command}: TONE

{level}: select a value from -99→200.

Contrast to Tone level equivalents.
Contrast 0 = Tone 0 Contrast 1 = Tone 100
Contrast 2 = Tone 200 Contrast 3 = No equivalent

JUSTIFICATION Commands

Alignment of fields can be controlled by using the justification commands. By default, the printer will left justify all fields. A justification command remains in effect for all subsequent fields until another justification command is specified.

Format:

{command} [end]

where:

{command}: Choose from the following:

CENTER	Center justifies all subsequent fields.
LEFT	Left justifies all subsequent fields.
RIGHT	Right justifies all subsequent fields.

[end]: End point of justification. If no parameter is entered, justification commands use the print head's width for horizontal printing or zero (top of form) for vertical printing.

JUSTIFICATION Example

Input (JUSTIFY.LBL):

```
! 0 200 200 210 1
CENTER 383
TEXT 4 0 0 75 C
LEFT
TEXT 4 0 0 75 L
RIGHT 383
TEXT 4 0 0 75 R
FORM
PRINT
```

Output:

PAGE-WIDTH Command

The printer assumes that the page width is the full width of the printer. The maximum height of a print session is determined by the page width and the available print memory. If the page width is less than the full width of the printer, the user can increase the maximum page height by specifying the page width.

Note: This command should be issued at the beginning of a print session.

Format:

{command} {width}

where:

{command}: Choose from the following:

PAGE-WIDTH (or PW)	Specifies page width.

{width}: Unit-width of the page.

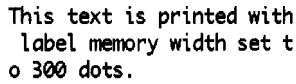
PAGE-WIDTH Example

Input 1 (PW1.LBL):

```
! UTILITIES
SETLP 7 0 15
PW 300
PRINT


This text is printed with label memory width
set to 300 dots.
```

Output 1:

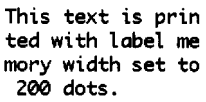


Input 2 (PW2.LBL):

```
! UTILITIES
SETLP 7 0 15
PW 200
PRINT


This text is printed with label memory width
set to 200 dots.
```

Output 2:



PACE Command

This command can be used with batch printing. When PACE is activated, the user must depress the feed key to print additional labels until the batch quantity is exhausted. By default, pacing is disabled on power up.

Format:

{command}

where:

{command}: PACE

PACE Example

In the following example, the command file shown was sent to the printer once. The two additional printouts were produced by pressing the 'FEED' key once for each additional printout.

Input (PACE.LBL):

```
! 0 200 200 210 3
; Tell printer to print a label
; after each 'FEED' key press
; until all 3 labels are printed
PACE
; Printer holds journal stock
JOURNAL
; Center the text
CENTER
TEXT 4 1 0 10 Print 3 labels
TEXT 4 1 0 90 Using PACE
PRINT
```

Output:

AUTO-PACE Command

This command can be used to instruct a printer equipped with a label presentation sensor to delay printing until the previously printed label is removed.

Format

{command}

Where:

{command}: AUTO-PACE

AUTO-PACE Command Example

This example instructs the printer to print 10 labels. The printer prints a label, and waits for that label to be removed before printing the next label.

Input (APACE.LBL):

```
! 0 200 200 250 10
CENTER
TEXT 7 0 0 10 AUTO-PACE EXAMPLE
AUTO-PACE
FORM
PRINT
```

NO-PACE Command

This command cancels the PACE and AUTO-PACE mode, if the printer is already in PACE orAUTO-PACE. The printer defaults to NO-PACE on power up.

format: {command}

where: {command}: NO-PACE

NO-PACE Command Example

This example instructs the printer to print 10 labels. The printer prints a label, waits for the label to be removed before printing the next label. The second set of 10 labels will be printed in batch mode and the printer will not wait for the operator to remove the labels.

```
Input (NO-PACE.LBL):

! 0 200 200 250 10
TEXT 7 0 0 10 AUTO-PACE EXAMPLE
AUTO-PACE
FORM
PRINT
! 0 200 200 250 10
TEXT 7 0 0 10 NO-PACE EXAMPLE
NO-PACE
FORM
PRINT
```

WAIT Command

This command is used to introduce a delay after a label is printed.

Format
{command} {delay-time}

where:

{command}: WAIT

{delay-time}: Delay time in 1/8 seconds.

WAIT Command Example

In the example below, the printer will pause 10 seconds (10 * 8 = 80) after printing each label.

```
INPUT (WAIT.LBL):

! 0 200 200 150 5
WAIT 80
TEXT 5 0 0 20 DELAY 10 SECONDS
FORM
PRINT
```

REWIND Command

This command is used to turn the rewind (or take-up) motor on or off. The printer defaults to REWIND-ON on power up. The REWIND command is ignored by printers that are not equipped with a motorized rewind.

format: {command}

where: {command}: Choose one of the following:

REWIND-OFF

REWIND-ON

REWIND Command Example

```
INPUT (REWIND.LBL):

! 0 200 200 150 1
REWIND-OFF
TEXT 5 0 0 20 TURNS REWIND OFF
PRINT
```

TENSION Commands

The tension commands are used to adjust the liner tension before and/or after printing a label by running the rewind motor for a pre-specified length. This adjustment improves peeler performance of printers equipped with a motorized rewind mechanism. The TENSION commands are ignored by printers not equipped with a motorized rewind.

Format

{command} {length}

where:

{command}: Choose one of the following:

PRE-TENSION: Perform tension adjustment prior to printing the label.

POST-TENSION: Perform tension adjustment after printing the label.

{length}: The unit length the rewind motor should advance to tighten the liner tension. The re wind motor will slip once tension is adjusted (it will not pull the stock out of adjustment for the next print cycle).

TENSION Command Example

In this example, the printer is instructed to run the rewind motor for 30 dot-lines, removing any slack in the liner to adjust the liner tension prior to printing the label.

```
Input (TENSION.LBL):
! 0 200 200 150 1
PRE-TENSION 30
TEXT 5 0 0 20 ADJUSTS TENSION
```

SPEED Command

This command is used to set the highest motor speed level. Each printer model is programmed with a minimum and maximum attainable speed. The SPEED command selects a speed level, on a scale of 0 to 5, level 0 being the slowest within this range. The maximum speed programmed into each printer model is attainable under ideal conditions. The battery or power-supply voltage, stock thickness, print darkness, applicator usage, peeler usage, and label length are among the factors that would further limit the maximum attainable print speed.

By exercising this command, the user will override the factory programmed speed for the label being printed.

Format
{command} {speed level}

where:

{command}: SPEED

{speed level}: A number between 0 and 5, 0 being the slowest speed.

SPEED Command Example

```
Input (SPEED.LBL):

! 0 200 200 150 1
SPEED 4
TEXT 5 0 0 20 PRINTS AT SPEED 4
FORM
PRINT
```

SETSP Command

The SETSP command is used to change spacing between text characters.

Format:

{command} {spacing}

where:

{command}: SETSP

{spacing}: Unit measurement between characters. The default for spacing is zero.

SETSP CommandExample

Input (SETSP.LBL):

```
! 0 200 200 210 1
T 4 0 0 10 Normal Spacing
SETSP 5
T 4 0 0 50 Spread Spacing
SETSP 0
T 4 0 0 90 Normal Spacing
FORM
PRINT
```

Output:

ON-OUT-OF-PAPER Command

This command can be issued to instruct the printer as to the course of action to take when it encounters an error while printing the label (such as running out of stock)

format: {command} {action} {number of retries}

where: { command}: ON-OUT-OF-PAPER

{action}: Choose one of the following:

PURGE: Discard the label if print error is encountered after the specified number of attempts.

WAIT: Do not discard the label if print error is encountered. In this mode the printer will wait for the error to be corrected before making the next print attempt.

{number of retries}: n: Specify how many times the printer should attempt to print the label.

The default printer configuration is:
ON-OUT-OF-PAPER PURGE 2

ON-OUT-OF-PAPER Command Example

This example instructs the printer to attempt to print the label twice.

```
Input (OOP.LBL):

! 0 200 200 150 1
ON-OUT-OF-PAPER WAIT 2
TEXT 5 0 0 20 MAKE TWO ATTEMPT
FORM
PRINT
```

ON-FEED Command

Your printer can be configured to ignore, form-feed, or reprint the last label when the feed key is pressed or when it receives a form-feed character (0x0c).

Format:

{command} {action}

where:

{command}: ON-FEED

{action}: Choose from the following:

IGNORE	Don't take any action when the feed key is pressed or when the form-feed character is received.
FEED	Feed to top-of-form when the feed key is pressed or when the form-feed character is received.
REPRINT	REPRINT the last label when the feed key is pressed or when the form-feed character is received.

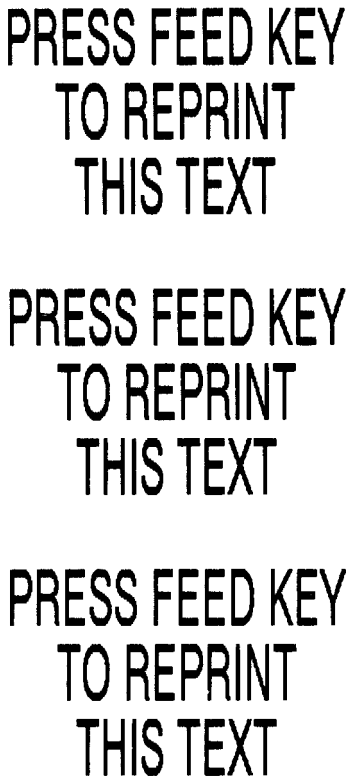
In the following example, the command file shown was sent to the printer only once. The two additional labels were produced by pressing the printer 'FEED' key once for each additional label.

ON-FEED Example

Input (ONFEED.LBL):

```
! 0 200 200 300 1
ON-FEED REPRINT
CENTER
JOURNAL
TEXT 4 1 0 20 PRESS FEED KEY
TEXT 4 1 0 100 TO REPRINT
TEXT 4 1 0 180 THIS TEXT
PRINT
```

Output:



PREFEED Command

The PREFEED command instructs the printer to advance the media a specified amount prior to printing.

Format:

{command} {length}

where:

{command}: PREFEED

{length}: Unit length the printer advances media prior to printing.

PREFEED Command Example

The following example sets up the printer for pre-feeding 40 dot-lines prior to printing.

Input (PREFEED.LBL):

```
! 0 200 200 210 1
PREEFEED 40
TEXT 7 0 0 20 PREFEED EXAMPLE
FORM
PRINT
```

POSTFEED Command

The POSTFEED command instructs the printer to advance the media a specified amount after printing.

Format:

{command} {length}

where:

{command}: POSTFEED

{length}: Unit length the printer advances media after printing.

POSTFEED Command Example

The following example sets up the printer for post-feeding 40 dot-lines after printing.

Input (POSTFEED.LBL):

```
! 0 200 200 210 1
TEXT 7 0 0 20 POSTFEED EXAMPLE
FORM
POSTFEED 40
PRINT
```

PRESENT-AT Command

The PRESENT-AT command can be used to position the media at the tear bar of the printer or at a location where the printed label can be easily removed by the operator. When a PRESENT-AT command is issued, the printer will print a label and then, after a delay period, advance the media a specified distance. It will then retract the media the same distance before starting a new print job.
The “delay” parameter is used to avoid unnecessary advance/retract operations when printing a batch of print jobs. The PRESENT-AT command can be issued in a label file or in a utilities command session (!UTILITIES…PRINT)

Format:

{command} {length} {delay}

where:

{command}:	PRESENT-AT
{length}:	Unit length in dot-lines the media is advanced after printing and retracted before printing the next label.
{delay}:	The interval after printing the label the printer waits prior to advancing the media. Incrementa are in 1/8 of a second. A delay of “1” is equivalent to 1/8th of a second. a delay of “4” is equivalent to 1/2 second, etc.

PRESENT-AT Command Example

The following example instructs the printer to wait 1/4 second and if there is no printer activity within this interval to then advance the media 80 dot-lines. The printer will retract the media by the same amount before printing the next label.

```
Input (PRESENT-AT.LBL):

! 0 200 200 250 1
TEXT 7 0 0 10 PRESENT-AT EXAMPLE
PRESENT-AT 80 2
FORM
PRINT
```

COUNTRY / CODE PAGE Command

The COUNTRY control command substitutes the appropriate character set for the specified country.

Format:

{command} {name}

where:

{command}: COUNTRY

{name}: Choose from the following:

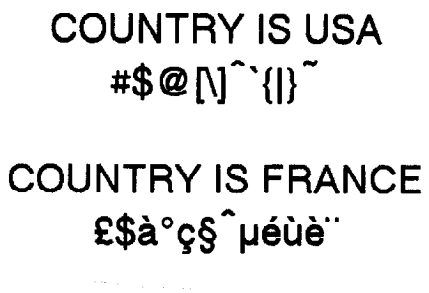
USA
GERMANY
FRANCE
SWEDEN
SPAIN
NORWAY
CHINA
ITALY
CP850
UK

COUNTRY / CODE PAGE Example

Input (COUNTRY.LBL):

```
! 0 200 200 80 1
IN-MILLIMETERS
JOURNAL
CENTER
; Set the country as USA
COUNTRY USA
; Now Print Text From ISO sub-
stitution Table
TEXT 4 0 0 8 COUNTRY IS USA
TEXT 4 0 0 15 #$@[\]^‘{|}~
; Set country for France and
print the same text
COUNTRY FRANCE
TEXT 4 0 0 28 COUNTRY IS FRANCE
TEXT 4 0 0 35 #$@[\]^‘{|}~
PRINT
```

Output:



USING FORMAT FILES

The DEFINE-FORMAT and USE-FORMAT commands are used to identify format and data respectively.

Format files eliminate having to re-send the same format information for every label printed. By using a pre-loaded format, only variable data (such as descriptions, price, etc.) is sent to the printer.

Input (NOFORMAT.LBL):

```
! 0 200 200 210 1
CENTER
TEXT 4 3 0 15 $22.99
TEXT 4 0 0 95 SWEATSHIRT
BARCODE UPCA 1 1 40 0 145 40123456784
TEXT 7 0 0 185 40123456784
FORM
PRINT
```

Output:

The following pages illustrate separating the above example into a format file and data.

USING FORMAT FILES (continued)

Defining a label format file is accomplished using the DEFINE-FORMAT (or DF) command to mark the beginning of the format, and PRINT to mark the end. A '\\' (double-backslash) acts as a place holder for data.

```
Input (FORMATF.LBL):

! DF SHELF.FMT
! 0 200 200 210 1
CENTER
TEXT 4 3 0 15 \\
TEXT 4 0 0 95 \\
BARCODE UPCA 1 1 40 0 145 \\
TEXT 7 0 0 185 \\
FORM
PRINT
```

The USE-FORMAT (or UF) command instructs the printer to use a specified format file. The label will be created using that format file with data supplied following the USE-FORMAT command. After accessing the specified format file, the printer substitutes the '\\' delimiters with the data supplied, producing the desired label.

```
Input (FORMATD.LBL):

! UF SHELF.FMT
$22.99
SWEATSHIRT
40123456784
40123456784
```

USING FORMAT FILES (continued)

As with all print commands, each line in a format file and its accompanying variables must be terminated with the carriage-return **and** line-feed character sequence.

Once defined, a format will remain in the printer's non-volatile memory for future reference. An existing format can be changed by over writing the file. By using the DEL command, the format file can be deleted.

Format file names can consist of no more than 8 letters or digits, and format file extensions can be no more than 3 letters or digits. Any lowercase letter in the format file name or extension will be converted to upper case.

Note: Every time a file is created on the printer using the "! DEFINE-FORMAT...", "! DF..." or the Visual Label Utility application, for example, the file information is written to flash memory. Unlike RAM, flash memory does not require battery for retaining data, and is immune to data corruption due to static discharge. Although flash memory is superior to RAM for safe-guarding file contents, it is limited to an average of 10,000 write cycles (i.e. file creations). For this reason, the user should exercise the file creation commands such that the stated limit is not exceeded.

BEEP Command

This command instructs the printer to sound the beeper for a given time length. Printers not equipped with a beeper will ignore this command.

Format:
{command} {beep_length}

Where:
{command}: BEEP

{beep_length}: Duration of beep, specified in .125 (1/8th) second increments.

BEEP Command Example

This example instructs the printer to beep for two seconds (16 x .125 seconds = 2 seconds)

Input (BEEP.LBL):

```
! 0 200 200 210 1
CENTER
TEXT 5 0 0 10 BEEPS FOR TWO SECONDS
BEEP 16
FORM
PRINT
```

CUT Command

On printers equipped with cutters, this command will cut the label after it is printed.

Format:

{command}

where:

{command}: CUT

CUT Example

Input (CUT.LBL):

```
! 0 200 200 1.5 1
IN-INCHES
; Journal label 300 dots long
CENTER
; Print some text
TEXT 4 0 0 .15 CUT COMMAND
TEXT 4 0 0 .5 EXAMPLE
; After we print the label, cut it.
CUT
PRINT
```

PARTIAL-CUT Command

On printers equipped with cutters, this command will cut the label after it is printed, leaving a portion of the label uncut to facilitate easily tearing the remainder of the label.

Format:

{command}

where:

{command}: PARTIAL-CUT

PARTIAL-CUT Example

Input (PCUT.LBL):

```
! 0 200 200 1.5 1
IN-INCHES
; Journal label 300 dots long
JOURNAL
CENTER
; Print some text
TEXT 4 0 0 .15 PARTIAL CUT
TEXT 4 0 0 .5 EXAMPLE
; After we print the label, partially cut the
label.
PARTIAL-CUT
PRINT
```

CUT-AT Command

This command is used on printers equipped with a cutter, in conjunction with the CUT or PARTIAL-CUT commands. This command will instruct the printer to retract the stock by a specified length. Printers not equipped with a cutter will ignore this command. This command should not be used on printers that use a thermal transfer ribbon.

Format: {command} {length}

Where: {command}:
CUT-AT

{length}: The unit length the stock should be retracted after performing a cut or partial cut.

CUT-AT Command Example

This example instructs the printer to print a label, perform a form-feed, cut the label, and retract the stock by 100 dot-lines.

```
Input (CUTAT.LBL):

! 0 200 200 250 1
CENTER
TEXT 7 0 0 20 CUT-AT EXAMPLE
CUT
CUT-AT 100
FORM
PRINT
```

MCR Commands

These commands (MCR, MCR-QUERY and MCR-CAN) can be used to configure and activate the optional Magnetic Card Reader (MCR). The MCR commands will be ignored by printers that are not equipped with a Magnetic Card Reader.
The MCR command can be issued in a label file (`! 0 200 … PRINT`), or in a utilities command session (`! UTILITIES … PRINT`). Refer to the discussion of the MCR commands in Section 10 (Advanced Utilitiies) of this manual for complete information on using the MCR option.

DESIGNING RECEIPTS AND LISTS

Introduction

Besides printing labels, the Comtec printer can make receipts, lists, and other variable length documents in line printer mode. When the printer receives raw ASCII text, it will print out your document in raw text. In Windows 95, this is called the “Generic/Text Only” printer.

While in line print mode, the printer can interpret special commands to change the font, to change the spacing of characters, or even to print out bar codes and graphics. Receipts can be just as ornate and detailed as your most complex label design.

In line print mode, the printer will use the spacing, fonts, and form-feed instructions in a raw text ASCII file. The printer will interpret line feed and carriage returns as well as form feed characters. It will **not** print tab characters. In label mode, the user must provide a X and Y coordinate for every item on the label. In line print mode, the printer can automaticaly calculate these coordinates or use the ones the user provides.

This section explains how to use line print mode to its full advantage. It covers the basics of how to use utility commands and of the most commonly used commands to create receipts. The end of this section includes sample files and results as well as how to design a unique and professional receipt.

It is assumed that the user knows how to communicate with the printer and how to create a basic ASCII file using a text editor like DOS ‘EDIT’ or Windows™ Notepad.

Special Commands Using the Utility Function

The printer can execute several utility commands at once or one at a time.

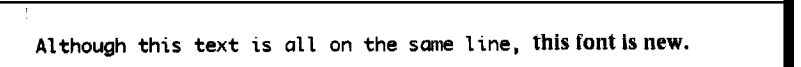
```
! U
SETLP 7 0 24
PAGE-WIDTH 720
PRINT
This line is printed as raw text.
These lines are in font 7 size 0

! U1 SETLP 7 0 24
! U1 PAGE-WIDTH 720
This line is printed as raw text.
These lines are in font 7 size 0.
```

The "! UTILITIES" command, or "! U" for short, must be ended by the terminator "PRINT" followed by a CR/LF (Carriage Return/Line Feed, or "Enter") to end the utility session. The "! U1" command only executes one utility command and must be followed by a CR/LF. Also, the "! U1" command can be placed ANYWHERE in a text line to execute the command.

Example:

```
Although this text is all on the same
line, ! U1 SETLP 5 0 24 (CR/LF)
this font is new.
```



To change the default settings for the printer, any of these commands can be placed in an AUTOEXEC.BAT file. Please see the "Batch Files" section in "Printer Configuration and Setup" in the Programming Manual section 12.

UNITS Commands

The units commands specify a measurement system for all subsequent command fields in a utility session. Coordinates, widths, and heights for all utility commands can be entered with precision to four decimal places. The printer measurement system will default to dots until a units command is issued.

Format:

<!> <UTILITIES>
{command}
<PRINT>

where:

{command}: Choose from the following:

IN-INCHES	Measurement in inches.
IN-CENTIMETERS	Measurement in centimeters.
IN-MILLIMETERS	Measurement in millimeters.
IN-DOTS	Measurement in dots. The default unit of measurement is in dots.

SETLP Command

Selecting the line printer font (the SETLP command) will change the font the printer uses for line print mode. It also chooses the amount of space the printer will move down when the printer recieves a carriage return (hex value 0x0d).

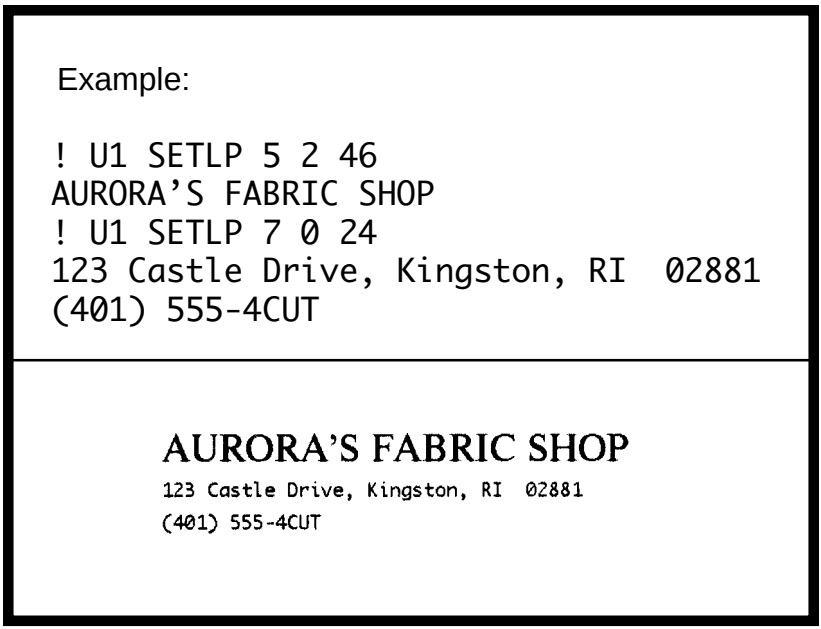
! U1 SETLP {font name or number} {size} {unit height}

SETLP allows you to use either the resident fonts or pre-scaled fonts downloaded to the flash memory. The Label Vista design software can create and upload a font for the printer from any available TrueType[1] font. Appendix D contains a table of all resident font heightsand their proper unit height.

You can set the printer font multiple times when using the line printer to make a receipt. For example, to put the company name in a larger font at the top of a label, change to font 5 size 2 and then to font 7 size 0.

Example:

```
! U1 SETLP 5 2 46
AURORA'S FABRIC SHOP
! U1 SETLP 7 0 24
123 Castle Drive, Kingston, RI  02881
(401) 555-4CUT
```



[1] TrueType is a registered trademark of Apple Computers, Inc.

SETLF Command

Use the SETLF command to change the height of each line without changing the font.

! U1 SETLF {unit height}

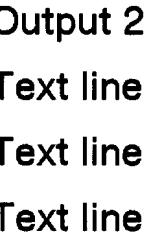
The command "! U1 SETLF 40" will advance the paper 40 dots for every LF (line feed, hex value 0x0a) character it receives.

SETLF Command Example

```
Input (SETLF2.LBL):

   ! U SETLP 4 0 40
   SETLF 40
   PRINT
   Output 2
   Text line
   Text line
   Text line
```

Output :



Moving With X and Y Coordinates

Even though the printer is in a line print mode, it can still move down and across the paper using X and Y values.

! U1 X {unit value}
! U1 Y {unit value}
! U1 XY {x unit value} {y unit value}
! U1 RX {unit x value to move relative to present position}
! U1 RY {unit y value to move relative to present position}
! U1 RXY{unit x value to move relative to present position} {unit y value to move relative to present position}

This command is useful for moving across the paper without using extra spaces or moving down the paper without needing to set the SETLF command to a specific value.

Negative values cannot be used for “Y” coordinates.

LMARGIN Command

The LMARGIN command sets the left margin in line print mode. Instead of issuing several X commands or inserting spaces, the LMARGIN command moves everything over the number of dots you choose.

! U1 LMARGIN {dots to offset from left}

This function can be used with the PAGE-WIDTH command. LMARGIN will move the left margin over the set number of dots from the automatically calculated side of the paper.

SETBOLD Command

The SETBOLD command will make text bolder and slightly wider. The SETBOLD command takes one operand to set how black the text should be made.

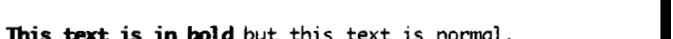
! U1 SETBOLD {value}

where {value} is a number from 0 to 5.

Be sure to issue a "! U1 SETBOLD 0" command to turn the bolding off when done.

Example:

```
! U1 SETBOLD 2
This text is in bold ! U1 SETBOLD 0
but this text is normal.
```



SETSP Command

The SETSP command is used to change spacing between text characters. Spreading out characters on a line makes the font appear wider. The SETSP command can also be used to spread out the text across the line.

! U1 SETSP {unit to separate characters}

For example, "! U1 SETSP 5" will put five dots between each character on the line. Try this command to make fonts look larger for emphasis.

Example:

```
Normal Text ! U1 SETSP 5
SPREAD OUT TEXT
```

Normal Text SPREAD OUT TEXT

PAGE-WIDTH Command
PAGE-HEIGHT Command

The printer lets you control both the width and height of the page through use of the PAGE-WIDTH and PAGE-HEIGHT commands. The Bravo four inch printers, for example, will take any size stock up to four inches wide. If the stock is three inches wide, use the PAGE-WIDTH, or PW command.

! U1 PW {unit width}

If the size of the receipts should remain a constant and the paper stock does not have a black bar to denote the top of form, use the PAGE-HEIGHT, or PH command. The printer will then partition the data you send into fixed page sizes.

! U1 PH {unit height}

Special ASCII Characters

Form Feed

ASCII Character (0x0c) will advance the paper to either the next index mark, or the length specified by the PAGE-HEIGHT, SETFF or SET-TOF commands. (The index mark is either a black line on the back of the stock, or the gap between labels. See GAP-SENSE or BAR-SENSE in Section 12 of this manual.)

Backspace

ASCII Character (0x08) acts as a non-destructive back-space. The character after the backspace character will appear on top of the previous character.

SETFF Command

The SETFF command is used to align top of media to print head. Once this command is executed, the alignment will occur when :

- feed key is pressed.
- form-feed character (0x0c) is issued.
- FORM command is issued.

Format:

<!> <UTILITIES>
{command} {max-feed} {skip-length}
<PRINT>

where:

{command}:	SETFF
{max-feed}:	Maximum unit-length the printer advances searching for the next eye-sense mark to align top of form.
{skip-length}:	Unit-length printer advances past top of form.

SETFF Command Example

The following example programs the printer to advance the paper until the eye-sense mark is found, or the paper has been advanced a maximum of 25 millimeters. If an eye-sense mark is found, the paper will be advanced an additional 2.5 millimeters.

```
Input (SETFF.LBL):

! UTILITIES
IN-MILLIMETERS
SETFF 25 2.5
PRINT
```

SET-TOF Command

This command is used to program the distance between the top-of-form and the end of the next (positive value) or previous (negative value) eye-sense mark or gap. The eye-sense-mark or gap that is closer to the top-of-form should be used for top-of-form setting.

Format

{command} {d}

where:

{command}: SET-TOF

{d}: The distance between the top of form and the end of the next or previous eye-sense mark or gap, which ever is closer. The specified value should be negative if the previous eye-sense mark is used as reference, or positive if the next eye-sense mark is used.
The following are the maximum values (in DOTS) that can be specified for each model:

MP5022 & MP5033	101
MP5044	134
RP3	147
Encore 2/4	120
Bravo Series (200 d.p.i)	300
Bravo Series (300 d.p.i)	563
M8	222

SET-TOF Command Example 1, label with eye-sense mark

The following example sets the top-of-form to end of next eye-sense mark to 101 dots (measured from the top-of-form to the end of the next (lower) label's eye-sense mark.

```
Input (SETTOF.LBL):

! UTILITIES
SET-TOF 101
PRINT
```

SET-TOF Command Example 2, label with gaps

The following example sets the top-of-form to end of next eye-sense-mark/gap to 0 dots (measured from the top-of-form to the end of the next (lower) label's gap.

```
Input (SETTOF2.LBL):

! UTILITIES
SET-TOF 0
PRINT
```

Tearing or Cutting the Paper

PRESENT-AT Command
CUT-AT Command
CUT Command
PARTIAL-CUT Command

Remember to add a few extra Carriage Return/Line Feeds (CR/LF, or Enter) to the end of your receipt. This will advance the paper enough to allow the receipt to be torn off without ripping through the last line of text. After the printer is done advancing the paper, tear the paper off.

If your printer can move stock bi-directionally, the PRESENT-AT command will advance the paper enough to rip off the receipt and not tear through the last line of text. Then, when the printer starts on the next item, it will automatically retract the paper before printing to save on paper. Using PRESENT-AT without an argument will set the units to advance to the default for that printer.

```
! U1 PRESENT-AT {units to advance after print}
! U1 PRESENT-AT
```

For printers with a cutter, the CUT-AT command will advance the paper, cut, then retract back to not waste paper. Using CUT-AT without any argument will set the units to advance to the default for that printer.

```
! U1 CUT-AT {units to advance after print}
! U1 CUT-AT
```

Alternatively, the printer can simply cut the paper or partially cut the paper. (Partial cut is useful when the receipt may fall to the floor when the user doesn't grab it immediately.) At the end of the receipt, put in a sufficient

amount of CR/LF characters, then issue the CUT or PARTIAL-CUT command.

```
! U1 CUT
! U1 PARTIAL-CUT
```

These commands are discussed more fully in Section 8 of this manual)

Bar Codes, Graphics and Lines

Bar Codes:

The printer can make any 1-D barcode in line print mode. The BARCODE command works just like it would in a label file. See Section 5 of this manual on the BARCODE command for more details.

The BARCODE command is affected by the justification commands like "! U1 CENTER". See JUSTIFICATION Commands in section 8 of the manual for more information.

```
! U1 BARCODE {type} {width} {ratio} {height} {x} {y}
{data}
```

Graphics:

The printer can print PCX formatted graphics files in line print mode. This is not recommended, however, in order to keep print times to a minimum. The graphic should be loaded into the flash file system on the printer to achieve maximum printing speed. Please see Section 7 of this manual for more information on graphics, lines, boxes and PCX Commands.

Example:

```
! U1 PCX {x coordinate} {y coordinate} !<
{filename.pcx}
```

SETLP-TIMEOUT Command

If the printer does not receive any characters after a set time, it will begin to print. This delay can be set with the SETLP-TIMEOUT command.

! U1 SETLP-TIMEOUT {time in 1/8 second units}

Multiply the seconds to wait by 8 to get the correct time for the command.

Designing a Receipt

It is now possible to create a receipt using the commands just described. First, list all the fields that you want on the receipt. Will there be different types of receipts used in the business? Is this a receipt for picking up goods, or is this an itemized list of products ordered and paid for? Will there be a need to differentiate quickly between different types of receipts to prevent errors? In a nutshell, what do you need this thing to look like?

A sales receipt should contain a few basic fields to prevent confusion. First, place the business name on the top of the receipt, preferably in a font to differentiate it from everything else. Choose a large font which is either resident in the printer or custom created with the Label Vista software package. Resident font 4 size 0 or 1 and font 5 size 2 or 3 are perfect for this function. If you desire to keep the receipt size small, font 7 size 1 is tall enough to set the text apart from everything else, yet it conserves space. Set a little space between your business name and the next line.

Next, put the address and telephone number of the business under the name. Customers will appreciate not having to flip through the phone book the next time they want to buy something from you. Also, this helps track which stores sell which items when dealing with returns and special sales promotions. Set at least two lines of

space between the header and the rest of the receipt.

Consider other important items on the receipt. Add the date and time of the sale, the ID number of the cashier or salesperson. Is the action a sale, return, price adjustment, or a sales quote?

Next, the itemized bill of sale contains a SKU or UPC code, a product description, and a price. Consider using a fixed width font, or a monospaced font, for this section. A monospaced font keeps the characters an even width for every character. For example, an 'I' is the same width as an 'M' character. The resident fonts 0 and 7 are monospaced as well as other monospaced fonts available with the Label Vista software package.

Finally, at the end, many companies like to put a slogan or advertisement of coming events. Be certain to put a few CR/LF characters at the end after your final line to ensure that the bottom of the receipt does not get torn off through a printed part of the receipt.

Receipt Examples

The program must set up the file with all the spacing already inserted. Use only spaces, not tab characters, to make things line up correctly. Note that when you issue a command like "! U1 SETSP 0," it must be followed by a CR/LF, or enter. This will not advance the printer to the next line; it will only execute the utility command.

Example 1:

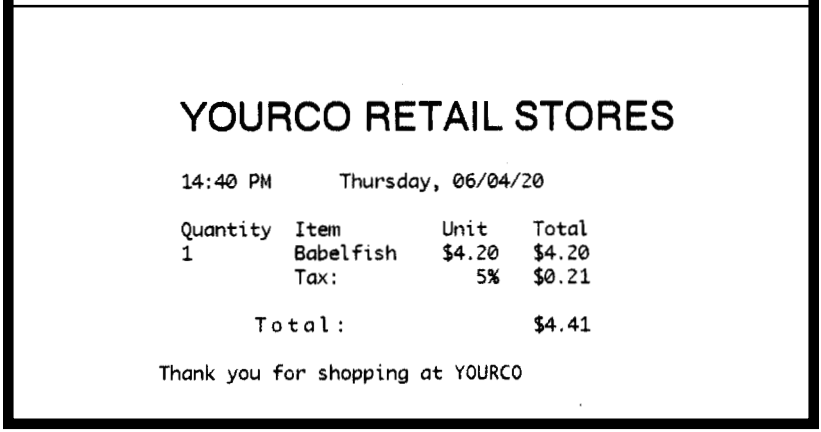
```
! U1 JOURNAL
! U1 SETLP 4 0 47
  YOURCO RETAIL STORES

! U1 SETLP 7 0 24
  14:40 PM      Thursday, 06/04/20

  Quantity  Item          Unit    Total
  1         Babelfish     $4.20   $4.20
            Tax:             5%   $0.21

! U1 SETSP 5
      Total:! U1 SETSP 0
                 $4.41

Thank you for shopping at YOURCO
```



YOURCO RETAIL STORES

14:40 PM Thursday, 06/04/20

Quantity	Item	Unit	Total
1	Babelfish	$4.20	$4.20
	Tax:	5%	$0.21
	Total:		$4.41

Thank you for shopping at YOURCO

The following example is a more complex design for a fabric shop. This receipt would print at the cutting table where a salesperson measures out the fabric. The cashier then scans the barcode at the bottom of the receipt to complete the sale.

Except for the utility commands, this receipt will print out much as it looks. Your program must provide all the correct number of spaces and text alignment.

Example 2 Input:

```
! U1 JOURNAL
! U1 SETLP 5 2 46
             AURORA'S FABRIC SHOP
! U1 SETLP 5 0 24
         123 Castle Drive, Kingston, RI  02881
               (401) 555-4CUT
! U1 SETLP 7 0 24

  4:20 PM Thursday, June 04, 2020  Store: 142
  Order Number:  #59285691
  Status:  ! U1 SETSP 10
INCOMPLETE ! U1 SETSP 0


Item  Description  Quant.   Price  Subtotal  Tax

1211  45" Buckram  5 yds @  $3.42/yd $17.10   Y
Z121  60" Blue Silk 10 yds@ $15.00/yd $150.00  N
Z829  60" Muslin    20 yds@ $1.00/yd $20.00   Y

             SUBTOTAL:  $187.10
   RHODE ISLAND SALES TAX 7.00%:      $2.60

             TOTAL:     $189.70

! U1 SETLP 7 1 48
   PLEASE BRING THIS RECEIPT TO THE CASHIER
       WITH THE REST OF YOUR PURCHASES.

! U1 CENTER
! U1 B 128 1 2 100 0 0 59285691 ST 187.10 T 2.60
```

Example 2 Output:

AURORA'S FABRIC SHOP

123 Castle Drive, Kingston, RI 02881
(401) 555- 4CUT

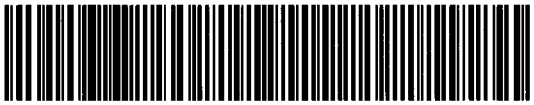
4:20 PM Thursday, June 04, 2020 Store: 142
Order Number: #59285691
Status: I N C O M P L E T E

Item	Description	Quant.	Price	Subtotal	Tax
1211	45" Buckram	5 yds @	$3.42/yd	$17.10	Y
Z121	60" Blue Silk	10 yds@	$15.00/yd	$150.00	N
Z829	60" Muslin	20 yds@	$1.00/yd	$20.00	Y

SUBTOTAL: $187.10
RHODE ISLAND SALES TAX 7.00%: $2.60

TOTAL: $189.70

PLEASE BRING THIS RECEIPT TO THE CASHIER
WITH THE REST OF YOUR PURCHASES.



ADVANCED UTILITIES

The Advanced Utilities are used to manage the flash file system, obtain information about firmware and printer applications, configure the printer for use in other countries, and to set several operating parameters.

The following example illustrates the use of some of the commands you will find in this section. The example assumes that the printer is connected to a host computer capable of full duplex serial communication. Comments to the right are not part of the session. Comments in UPPERCASE are commands sent from the host to the printer. Comments in lower case are printer responses to the host. These commands are further explained as you read through this section.

`! UTILITIES`	*START A UTILITIES SESSION*
`VERSION`	*GET THE FIRMWARE VERSION*
`6001`	**firmware version is 60.01**
`CHECKSUM`	*GET APPLICATION CHECKSUM*
`F723`	**checksum is F723**
`DIR`	*GET DIRECTORY OF THE FLASH FILE SYSTEM*
`Directory`	**directory has the following 3 files**
`PLL_LAT.CSF  17306`	**17306 byte file**
`PLL_LAT.CSF  18423`	**18423 byte file**
`AUTOEXEC.BAT  96`	**96 byte file**
`TYPE AUTOEXEC.BAT`	*WHAT'S IN AUTOEXEC.BAT?*
`! UTILITIES`	**autoexec.bat line 1**
`SETLP 5 1 40`	**autoexec.bat line 2**
`PRINT`	**autoexec.bat line 3**
`DEL AUTOEXEC.BAT`	*DELETE THE AUTOEXEC.BAT FILE*
`DIR`	*GET A DIRECTORY OF THE FLASH FILE SYSTEM*
`Directory`	**directory has the following 2 files**
`PLL_LAT.CSF  17306`	
`PLB_LAT.CSF  18423`	
`PRINT`	*CLOSE THE UTILITIES SESSION*

Note: *Text printed in italic type refers to data sent to the printer.*
Text printed in bold type refers to data sent from the printer.

VERSION Utility

This command reports the firmware version as a four character null-terminated ASCII string.

Format:

<!> <UTILITIES>
{command}
<PRINT>

where:

{command}: VERSION

VERSION Example

```
Input (VERSION.LBL):

! UTILITIES
VERSION
PRINT
```

CHECKSUM Utility

This command reports the application checksum as a four character null-terminated ASCII string.

Format:

<!> <UTILITIES>
{command}
<PRINT>

where:

{command}: CHECKSUM

CHECKSUM Example

```
Input (CHECKSUM.LBL):

! UTILITIES
CHECKSUM
PRINT
```

DEL Utility

The DEL command deletes the specified file.

Format:

<!> <UTILITIES>
{command} {name.ext}
<PRINT>

where:

{command}: DEL

{name.ext}: Name of file to be deleted. Note: DEL*.* may be used to globally delete all files.

DIR Utility

The DIR command sends the file directory to a host.

Format:

<!> <UTILITIES>
{command}
<PRINT>

where:

{command}: DIR

DEFINE-FILE (DF) Utility

The DF command defines a file name for a file to be loaded into the printer. If a file with the same name already exists in the printer, it will be overwritten with the new file. The contents of the file must contain ASCII characters. To transfer binary files to the printer, use the utility provided in the Visual Label Utility

Format:

<!> {command} {filename.ext}
{data}
{terminator}

where:

{command}: DF

{filename.ext}: Name of file to be created.

{data}: The contents of the file. The file must be ASCII and cannot contain any {terminator} keyword.

{terminator}: Choose from the following:

PRINT: If the PRINT terminator is used, it is also written to the file.

END: If the END terminator is used, it is not written to the file.

DEFINE-FILE Example

```
Input (DF.LBL):

! DF AUTOEXEC.BAT
! UTILITIES
SETFF 200 20
PRINT
```

In this example the file AUTOEXEC.BAT is created in the printer flash file system. The file contents are:

```
! UTILITIES
SETFF 200 20
PRINT
```

Note: Every time a file is created on the printer using the "! DEFINE-FORMAT...", "! DF..." or the Visual Label Utility application, for example, the file information is written to flash memory. Unlike RAM, flash memory does not require battery for retaining data, and is immune to data corruption due to static discharge. Although flash memory is superior to RAM for safe guarding file contents, it is limited to an average of 10,000 write cycles (i.e. file creations). For this reason, the user should exercise the file creation commands such that the stated limit is not exceeded.

TYPE Utility

The TYPE command allows you to read a text file by sending it from the printer to a host.

Format:

<!> <UTILITIES>
{command} {name.ext}
<PRINT>

where:

{command}: TYPE

{name.ext}: Name of text file to be sent to host.

BAUD Utility

The BAUD command enables you to set the printer serial port baud rate. Note: this command will take effect immediately and the requested baud rate will remain in effect when the printer is powered down.

Format:

<!> <UTILITIES>
{command} {baud}
<PRINT> *;<PRINT> must be sent at the new baud rate*

where:

{command}: BAUD

{baud}: Choose from the following:

1200
4800
9600
19200

BAUD Example

```
Input (BAUD.LBL):

! UTILITIES
BAUD 19200
PRINT
```

COUNTRY / CODE-PAGE Utility

The COUNTRY utility command substitutes the appropriate character set for the specified country.

Format:

<!> <UTILITIES>
{command} {name}
<PRINT>

where:

{command}: COUNTRY

{name}: Choose from the following:

USA
GERMANY
FRANCE
UK
ITALY
SWEDEN
SPAIN
NORWAY
CHINA
CP850

COUNTRY / CODE-PAGE Utility Example

Input (COUNTRY1.LBL):

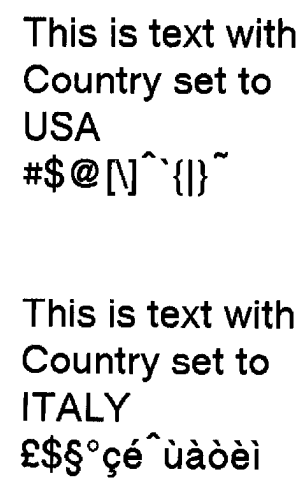
```
! UTILITIES
; Tell the printer to use font 4 size 0
; for line printer mode with 5 millime-
ters
; line spacing
IN-MILLIMETERS
SETLP 4 0 5
COUNTRY USA
PRINT

This is text with
Country set to
USA
#$@[\]^‘{|}~

! UTILITIES
COUNTRY ITALY
PRINT

This is text with
Country set to
ITALY
#$@[\]^‘{|}~
```

Output:



Note: The printer must be configured with fonts that contain the extended character sets used in the selected country.

ANNOUNCE Utility

The ANNOUNCE command is used to activate pre-programmed sounds in the printer. A space character is required between each code. This command will be ignored by printers that are not equipped with a speaker and voice circuit.

Format:

<!> <UTILITIES>
{command} {message}
<PRINT>

where:

{command}: ANNOUNCE

{message}: Choose from the following:

Code:	Spoken Messages:
' '	pauses between sounds
'.'	"point"
'0'	"zero"
'1'	"one"
'2'	"two"
'3'	"three"
'4'	"four"
'5'	"five"
'6'	"six"
'7'	"seven"
'8'	"ϵ
'9'	"ı
'^'	"version"
'{'	"go to aisle"
'|'	"latch is open"
'}'	"battery is low"
'~'	"out of paper"

ANNOUNCE Utility Example

Input (ANNOUNCE.LBL):

```
! UTILITIES
ANNOUNCE 1 2 . 2 5
PRINT
```

Output:

The speaker will play the message "one two point two five.

TIMEOUT Utility

The TIMEOUT command allows you to set the time the printer will remain on without receiving data. If no data is received after the specified timeout, the printer will turn itself off to save energy and preserve battery life. You can disable the timeout feature by setting the timeout value to 0.

Format:

<!> <UTILITIES>
{command} {time}
<PRINT>

where:

{command}: TIMEOUT

{time}: Time in 1/8 seconds of inactivity before printer will turn itself off.

TIMEOUT Utility Example

```
Input (TIMEOUT.LBL):

! UTILITIES
TIMEOUT 960
PRINT
```

This example sets the printer to turn off after 2 minutes of inactivity (120 seconds X 8 = 960).

BEEP Command

This command instructs the printer to sound the beeper for a given time length. Printers not equipped with a beeper will ignore this command.

Format:
{command} {beep_length}

Where:

{command}:	BEEP
{beep_length}:	Duration of beep, specified in .125 (1/8th) second incre ments.

BEEP Command Example

This example instructs the printer to beep for two seconds (16 x .125 seconds = 2 seconds)

Input (BEEP.LBL):

```
! UTILITIES
BEEP 16
PRINT
```

LT Command

This command specifies the command line terminator character(s). The default terminator characters are CR/LF or LF. The printer can be programmed to accept a different character sequence to terminate command lines. This command does not apply to data sent to the printer when it is in line print mode.

Format
{command} {mode}

\

Where:
{command}: LT
{mode}: Choose from the following.

CR: Carriage-return (0x0D) character is the line terminator.

LF : Line-feed (0x0A) character is the line terminator.

CR-LF: Carriage-return/line-feed (0x0D 0x0A) characters are the line terminator.

CR-X-LF: Line terminator is a carriage-return (0x0D) followed by any number of characters followed by the line-feed (0x0A) character. Characters found between the carriage-return and line-feed characters are discarded.

LT Command Examples

Set the printer to require line-feeds only as a terminating character.

```
! UTILITIES LT LF PRINT
```

Set the printer to ignore any characters found between a carriage return character and a line-feed character.

```
! UTILITIES LT CR-X-LF PRINT
```

SET-TIME Utility

This command sets the time in the real time clock module. Time should be a valid time and be given in the specified format. This command will be ignored by printers that are not equipped with a real time clock module.

Format:

<!> <UTILITIES>
{command} {time-stamp}
<PRINT>

where:

{command}: SET-TIME

time-stamp}: hh:mm:ss

hh = hours (00 – 23)
mm = minutes (00 – 59)
ss = seconds (00 – 59)

GET-TIME Utility

This command reports the current time, if valid, as an eight character null-terminated ASCII string. This command will be ignored by printers that are not equipped with a real time clock module.

Format:

<!> <UTILITIES>
{command}
<PRINT>
where:

{command}:	GET-TIME
Printer output:	hh:mm:ss\0
	hh = hours (00 – 23) mm = minutes (00 – 59) ss = seconds (00 – 59) \0 = null terminator (00H)

SET-DATE Utility

This command sets the date in the real time clock module. Date should be a valid date and be given in the specified format. This command will be ignored by printers that are not equipped with a real time clock module.

Format:

<!> <UTILITIES>
{command} {date-stamp}
<PRINT>

where:

{command}:	SET-DATE
{date-stamp}:	mm-dd-yyyy
	mm = month (01 – 12) dd = day (01 – 31) yyyy = year (1990 – 2089)

GET-DATE Utility

This command reports the current date, if valid, as an eight character null-terminated ASCII string. This command will be ignored by printers that are not equipped with a real time clock module.

Format:

<!> <UTILITIES>
{command}
<PRINT>
where:

{command}:	GET-DATE
Printer output:	mm:dd:yyyy\0
	mm = month (01 – 12) dd = day (01 – 31) yy = year (1990 – 2089) \0 = null terminator (00H)

Printing a Time Stamp

To print a time stamp on a label, use any text command and insert
[!<TIME] in place of the text to be printed.

Example:

```
Input (TIME.LBL)

! 0 200 200 210 1
TEXT 4 0 0 100 !<TIME
FORM
PRINT
```

Output:

14:47:23

Printing a Date Stamp

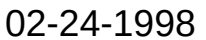
To print a date stamp on a label, use any text command and insert
[!<DATE] in place of the text to be printed.

Example:

Input (DATE.LBL)

```
! 0 200 200 210 1
TEXT 4 0 0 100 !<DATE
FORM
PRINT
```

Output:



Magnetic Card Reader (MCR) Command

This command can be used to configure and activate the Magnetic Card Reader (MCR). The MCR command will be ignored by printers not equipped with a Magnetic Card Reader.
The MCR command can be issued in a label file (`! 0 200 … PRINT`), or in a utilities command session (`! UTILITIES … PRINT`). This command activates the MCR. When the printer is turned on , the MCR is not active by default until the MCR command is received. To make the MCR active on power-up, the MCR command can be issued from autoexec.bat or run.bat files.

The RXD LED of the printer will blink while the MCR is active. When the MCR times out or after a successful read (provided the MCR is not in MULTIPLE read mode,) the MCR will deactivate, thereby returning the RXD LED back to its normal state.
Format

{command} {time-out} {options}

Where:

{command}:	MCR
{time-out}:	time-out is the inactivity shut-down time, in 1/8th of a sec ond. For example, time-out=80 for a 10-second inac-tivity time-out(10 * 8), or time-out=160 for a 20-second inactivity time-out(20 seconds * 8). The MCR timer starts ticking once the last option of MCR command is received by the printer. When MCR times

out, it will not read a new card until a new MCR command is issued. The only exception to this rule is when MCR is in MULTIPLE read mode. See below for description of MULTIPLE read mode. A time-out of 0 will instruct the printer to never time-out, and wait for a successful read.

{options}: The options listed below can be specified in any order, and must be separated with spaces. The last option must be terminated with cr/lf (carriage-return / line-feed) characters. Options are accumulative, meaning that “MCR 10 ERRORS T1” and “MCR 40 T2” are equivalent to “MCR 40 ERRORS T1 T2”. The time-out field is not accumulative, and only the last specified time-out is used. Select from the following options:

Track Options:

T1 :Read Track 1. (can specify one or more tracks to read).

T2 :Read Track 2.

T3: Read Track 3. (Currently not supported)

Frequency Options:

MULTIPLE: Read multiple swipes. MCR will continue to read and report card swipes until the MCR times out. SINGLE read is the default mode.

SINGLE: Read and report one successful card swipe. No data will be reported if the MCR times out before a successful read. SINGLE read is the default read frequency mode.

Data Reporting Options

QUERY: Report MCR data when queried (In response to MCR-QUERY command, see MCR-QUERY.). By default, the printer will report MCR data when valid MCR data is decoded before the MCR times out.

Debugging Options

ECHO: The printer will print the MCR data by internally routing it to the line-printer module, forcing the data to be printed, as well as being transmitted to the host computer.

Track Data Transmit Options

PREFIX prefix: This option specifies the track data prefix. The printer defaults to no prefix. The PREFIX command should be followed by the data that needs to be sent from the printer to the host, up to 10 characters, and terminated with space or carriage-return/ line-feed characters. An example of a typical prefix option is “PREFIX START:”.

POSTFIX postfix: This option specifies the track data postfix. The printer defaults to no postfix. The POSTFIX command should be followed by the data that needs to be sent from the printer to the host after all track data have been transmitted, up to 10 characters, and terminated with space or carriage-return/ line-feed characters. An example of a typical postfix option is “POSTFIX END”.

DEL lr: Defines the track number delimiters, where “l” is the left delimiter character, and “r” is the right delimiter character. Default delimiters for the printer are “DEL T:”, meaning that the printer will transmit T, followed by the

track number if track number reporting is specified via the TN option, and the ':'. The transmitted data will be, "T1:" or "T:", or "T2:", depending on the options selected.

DELAY nnn: This option specifies the inter-character delay of data sent from the printer to the host, in milliseconds. This command is typically used where the host computer can not collect data transmitted from the printer fast enough, resulting in missed characters. The default DELAY is 0. An example of the DELAY option is: "DELAY 15" which instructs the printer to observe a 15-millisecond delay before sending the next character data to the host.

TN: This option enables the reporting of track number between the delimiters (see "DEL lr" command). The printer will report the track number by default.

NTN:This options disables the reporting of the track number, inserted between the delimiters. The printer enables track number reporting by default.

Error Reporting Options

ERRORS: This option turns error reporting on. Error reporting is turned off by default. The following is a list of the error messages (Please note that the following error messages will be transmitted only if error reporting is turned on, via the ERRORS option in the MCR command):

READ ERROR: This error is reported when the card could not be read due to errors such as parity check, LRC check-sum, no end-sentinel, or invalid characters. It indicates that either: (1) the card is bad, or (2) the card was not swiped correctly. The printer will leave the MCR on and will continue to try to read future card swipe retries by the user until it times out or until a successful read, whichever occurs first.

CANCEL: This error is reported when a MCR-CAN command is received. This error message will confirm that the MCR is turned off.

TIME-OUT: This error is reported when the printer MCR times out before a successful read.

EPREFIX: This option is similar to the PREFIX command, but applies to error messages. Error messages, if error reporting is turned on, will be preceded by the specified eprefix. For example, “EPREFIX ERROR:” instructs the printer to prefix error messages with “ERROR:”. The default error prefix is “Error:”

MCR Command Examples

Example 1: This example sets the MCR in the printer for reading tracks one and two, with a timeout of 10 seconds (10 / 1/8th seconds = 80). The last option of the MCR command must be terminated with cr/lf.

```
! U1 MCR 80 T1 T2
```

Once card is swiped successfully, the following is sent to host:

```
T1:B4000001234562^PUBLIC JR/JOHN Q.MR^9209101999999999
<cr/lf>
T2:4000001234562=9209101999999999<cr/lf>
```

If the printer times out or the card can not be read, the printer will not return anything to the host (because ERRORS option is not specified in this example).

Example 2: This example sets the MCR in the printer for reading tracks one and two, with a timeout of 10 seconds (10 / 1/8th seconds = 80). The ECHO command will instruct the printer to print the MCR data (in addition to sending that data to the host). The ERRORS option

instructs the printer to report read errors, time-outs, or cancellations, if any. The last option of the MCR command must be terminated with cr/lf.

```
! U1 MCR 80 ECHO T1 T2 ERRORS
```

Once card is swiped successfully, the following is sent to host and printed:

```
T1:B4000001234562^PUBLIC JR/JOHN
Q.MR^9209101999999999          <cr/lf>
T2:4000001234562=9209101999999999<cr/lf>
```

If the printer times out before a successful read, it will send the following message to host and printed on the printer:

Error:T1:Time-out
Error:T2:Time-out

Example 3: This example illustrates the PREFIX, EPREFIX, POSTFIX, and DEL (DELimiter) options:

```
! U1 MCR 80 ECHO T1 T2 ERRORS PREFIX START
EPREFIX ERR POSTFIX END DEL ()
```

Once card is swiped successfully, the following is sent to host and printed:

```
START(1)B4000001234562^PUBLIC JR/JOHN
Q.MR^9209101999999999END<cr/lf>
START(2)4000001234562=9209101999999999END<cr/lf>
```

If the printer times out before a successful read, it will send the following message to host and printed on the printer:

```
ERR(1)Time-out<cr/lf>
ERR(2)Time-out<cr/lf>
```

MCR-QUERY Command

The MCR can be put in a query mode by specifying the “QUERY” option in the MCR command, where it will

report MCR data only when it is asked to via the MCR-QUERY command. By default, the printer will report card swipe data immediately after a successful read.

The printer will not respond to MCR-QUERY if it does not have any data, either the track data or an error message. Multiple MCR-QUERY commands may be issued, keeping in mind that the response, or lack thereof, to the MCR-QUERY reflects the card swipe data at the moment MCR-QUERY is issued.

Format

{command}

Where:
{command}: MCR-QUERY

MCR-QUERY Command Example

In this example the printer is instructed to report the MCR read operation results.

```
! U1 MCR-QUERY
```

MCR-CAN Command

The MCR-CAN command will terminate the current MCR activity, and if error message reporting is enabled via the MCR command ERROR option, will transmit the “Cancel” error message to host.

Format

{command}

Where:
{command}: MCR-CAN

MCR-CAN Command Example

In this example the printer is instructed to deactivate the printer MCR and cancel any pending read operations.

```
! U1 MCR-CAN
```

DENSO BHT COMMANDS

NOTE: All commands in this section are compatible with ACKNAK-IT v 6.1 or greater.

Comtec printers can have an application downloaded which will support communications to the Denso Barcode Handy Terminal (BHT).

The different methods of communications between a BHT and a Comtec printer are per the table below:

Connection/protocol type		Refer to	
Cable	IR	Autoexec.bat File	pg. no
Cable w/BHT		PROBHT.CBL	P10-49
	IR w/BHT	PROBHT.IR	P10-49
Cable w/BHT-IR		PROBHTIR.CBL	P10-50
	IR w/BHT-IR	PROBHTIR.IR	P10-50

Data is sent to the printer formatted as a BHT .DAT file. The printer will check for the ".DAT" filename extension. Since the format of data using this protocol is not always what the printer requires, there are several modes of operation that can be set within the printer to specify how record data fields delivered by the protocol are interpreted.

Setting the Data Format

If the default mode (strip trailing spaces then add CR/LF for each data field) is not desired, any of the operating modes can be set by loading an AUTOEXEC.BAT command file into the printer's flash memory.

Note that any commands intended to modify the BHT operating modes must be placed between the SRF-ACCESS and the END-SRF-ACCESS command lines.

Mode 1: (Default) Strip trailing spaces and add CR/LF
Each data field in a record has any trailing spaces

stripped and a CR/LF (carriage return/line feed) pair added. The data field (stripped of trailing spaces and with an appended CR/LF pair) is then passed to the printer for processing. This is the default way to handle data sent to the printer.

Mode 2: Include all characters in each field (RAW)
The second mode accepts all characters that make up each data field. Using this method, any trailing spaces in a data field will be INCLUDED as data for the printer to process.

Mode 3: Use the first byte as a count of characters to send to the printer.
This mode is based on the first character of a data field being interpreted as a count of data bytes that follow. It is similar to a 'counted string' as found in the Pascal programming language String data type. In the 'counted string' mode, the number of bytes specified by the count will be transferred into the printer. The count byte's maximum value is limited to the data field size minus 1.
Note that in this example the count byte is shown as ASCII, but would actually be sent as a binary number.

M**ode 4:** Strip trailing spaces
This mode eliminates trailing spaces in each data field of a record. Note that unlike Mode 1 no CR/LF pair is added to the data after stripping the trailing spaces.
Refer to pgs. P10-44 thru P10-47 for examples using the BHT-MODE commands

IR or Cable Interface Selection

The printers are normally self configuring for IR or cable data transmission. Plugging in the communications cable will disable the IR interface; removing the cable will make it active. This default method of interface selection can

be overridden by means of a BHT-PROTOCOL command in an autoexec. bat file. (Refer to the examples at the end of this section on setting the mode for IR or cable data transmissions.) BHT-PROTOCOL CABLE enables BHT or BHT-IR protocol via cable. If the BHT-PROTOCOL CABLE command is *not* present, data is sent "raw" over the serial cable; if it is detected BHT-PROTOCOL BHT sends data in the BHT mode; BHT-PROTOCOL BHTIR sends data in the BHT-IR mode. Performing a two-key reset of the printer will return it to its normal, automatic selection mode. (Refer to pg. P1-2 in this manual for the reset procedure.)

BHT-BAUD Command

The printers support 9600 and 19200 BPS baud rates. The default baud rate for communication with Comtec printers is 9600 BPS. Configuring a BHT baud rate will set both the IR and the cable transmit speeds to that rate. You must insure that both the printer and the BHT are set to the same baud rate. This can also be changed in an AUTOEXEC.BAT file with the BHT-BAUD <baud-rate> command. (Refer to the BHT-BAUD command example at the end of this section.)

Setting the Mode for IR Data Transmission

Autoexec.bat files can configure the printer for IR data transmission modes. The following notes apply:

- Unless specified otherwise with a BHT-PROTOCOL CABLE command, communication over a serial cable does not use the BHT or BHT-IR protocol. In this default ('raw") mode, any Comtec utility for reprogramming, adding or deleting files will be in effect.
- If the printer has been configured to include the BHT or BHT-IR protocol with the serial cable the following procedure may be used to force the printer to com-

municate “raw” (i.e. using no protocol) for the purposes of reprogramming, or the adding or deleting of files.

1. Connect the printer to a host terminal loaded with the desired communications software.
2. Turn the printer on by starting the terminal’s communications program.
3. Turn the printer off, then, while holding the “Feed” key down, turn the printer back on.
4. The printer will print a status report. When the printout is complete, the communications protocol will be in the “raw” mode.

(Refer to the BHT-MODE Commands examples at the end of this section.)

Configuring the BHT for IR transmission

The BHT must also be configured correctly to communicate with Comtec printers using the BHT or BHT-IR protocol. All parameters that must be set are found in the BHT Set System Æ Communication menu area. The following notes apply:

- The pulse width for optical communications must be set to 1.63µS.
- The protocol is set to match the protocol in the printer. Default for the printer is BHT-IR unless specified as BHT by an AUTOEXEC.BAT file.
- The baud rate (if the printer uses the default) is set to 9600 BPS
- The Serial Number is set to ON.
- Comtec printers use a ID number of 9. You should avoid using this ID value for the terminal.
- The Com port is set to OPTICAL.

BHT-IR File Transmission

Assume that a label file CIS.DAT has been loaded into the BHT. (Refer to pg. _ for details on loading a *.DAT file.) The steps to transmit this file via IR are as follows:

1. Depress the 1 / PW / SF keys all at once, then release. The SYSTEM MENU should be displayed. This menu has 6 options. This power on sequence must be performed every time to boot to the SYSTEM MENU. A regular power on (PW key) will boot into the BHT application.

2. Select option 3:UPLOAD. Press the ENT key. The UPLOAD menu should now be displayed. Select option 1:DRIVE A. Press the ENT key.
Use the arrow keys, F5, and F6, to scroll through the menu options to the desired file. (CIS.DAT in our example) When CIS.DAT is highlighted, press the ENT key.

3. The IR on the BHT should be pointed towards the IR window on the printer. The transfer of data will power on the printer if it is not already on. When file transfer is complete, ** Completed ** is displayed for the user and a beep sounds. The file will then print. Press the C key to exit this screen.

4. Continue to press the C key to exit back 1 menu at a time (if desired). The SYSTEM MENU is the top-level menu.

5. After a successful UPLOAD, the CIS.DAT file must be re-selected again to transfer to the printer.

NOTES: The arrow keys or a numeric keypress can be used to select a menu option. The arrow keys are F5 through F8. A press of the ENT key usually needs to accompany the option selection. Hold down the PW key for 1 – 2 seconds to turn the BHT off, or let it power down automatically.

Communication errors can occur infrequently during BHT-IR transmission. If a transmission is unsuccessful, the data. should be re-sent.

Configuring the BHT for Cable Transmission

The BHT can also be configured for cable communications with Comtec Printers using either the BHT or the BHT-IR protocol. All parameters that must be set are found in the BHT Set System Æ Communication menu option. The following notes apply:

- The protocol type is set to BHT or BHT-IR protocol. Printer default is BHT-IR
- The SET CONNECTOR menu options:
  1. The TRANSMIT SPEED (if the printer uses the default) is set to 9600 BPS
  2. The PARITY BIT is set to NONE
  3. The DATA BIT is set to 8 BITS
  4. The STOP BIT is set to 1 BIT
  5. The Serial Number is set to ON
  6. The Horizontal Parity is set to ON
- The COM PORT is set to IFC (on the BHT-5000 set COM DEFAULT to CONNECTOR)

Loading a .DAT file into the BHT

Comtec printers will only print files with the .DAT extension sent from the BHT. The following example file CIS.DAT contains a label file that can be loaded from a PC into the BHT via a serial cable using the TU3.EXE utility provided by DENSO. The file can then be sent to a Comtec printer to produce a label. Files with a .DAT extension contain records composed of one or more record fields. For CIS.DAT, each record will be specified as having only one field with a width of 40 bytes. The 40 byte width was chosen because no single line in the example CIS.DAT file exceeds 40 bytes. The resulting records transferred to the BHT will each contain a single field 40 bytes wide. The BHT and BHT-IR protocol will add space characters as needed to any field containing less than 40 bytes. Thus the CIS.DAT file contains 24 lines that will be interpreted as 24 records containing a single field of 40 bytes.

Note that while the example uses the TU3.EXE utility to download the file from a PC, these files could also be created under control of an application program executing directly on the BHT.

The command line that will transfer this file into the BHT is: `TU3 +MPC +B9600 CIS.DAT +F40`

The BHT will now contain the file CIS.DAT.

Sending The Example Label File to the Printer

Comtec printers with the BHT application will print files with the .DAT extension. As noted before, the default action of the printer when a record is received is to strip all trailing spaces from all record fields, then append a carriage return/line feed pair to the data. Using this default, the trailing spaces in every field of the example CIS.DAT file will be stripped, and a CR/LF pair will be added.

Interrupted Transmissions

If a transmission from the terminal to the printer is interrupted, the user should wait at least five seconds before attempting another transmission. During this delay, the printer will discard any partial data received from the interrupted transmission and reset itself to receive a new transmission.

Example .dat File

Input (CIS.DAT)

```
! 0 200 200 581 1
;MEDIA P/N LD-E9QT7S
LABEL
CONTRAST 0
TONE 0
SPEED 3
PAGE-WIDTH 240
BAR-SENSE
;// PAGE 0000000002400600
;// TEXT 0 1 3 560 DEPT 34
TEXT90 4 3 36 288 $22.88
TEXT90 5 2 163 273 SWEATSHIRT
VBARCODE UPCA 2 1 45 139 576 04364503284
TEXT90 7 0 191 511 043645032841
TEXT90 5 0 4 524 COMPARE AT
TEXT90 4 0 30 508 $ 30.00
TEXT90 5 0 115 575 ZD-180-KL
TEXT90 5 2 119 269 ALL COTTON
TEXT90 7 0 114 389 01/17/98
TEXT90 0 0 208 173 EA00-732-00560
TEXT90 5 0 82 519 ELSEWHERE
BOX 189 358 217 527 1
FORM
PRINT
```

Output:

SRF-ACCESS and END-SRF-ACCESS Commands

Commands intended to modify the BHT operating modes must be placed between the SRF-ACCESS and the END-SRF-ACCESS command lines.

SRF- and END-SRF-ACCESS Command Example

This autoexec.bat file uses the SRF-ACCESS and END-SRF-ACCESS command pair to establish BHT communication via the BHT-CABLE protocol.

```
Input: PROBHT.CBL

! DF AUTOEXEC.BAT
! UTILITIES
SRF-ACCESS
BHT-PROTOCOL BHT
BHT-PROTOCOL CABLE
BHT-BAUD 19200
BHT-MODE STRIP-ADD-CRLF
END-SRF-ACCESS
PRINT
```

BHT-BAUD Command

This command will set both the IR and the cable transmit speeds. You must insure that both the printer and the BHT are set to this same baud rate. This can also be changed in an AUTOEXEC.BAT file with the BHT-BAUD <baud-rate> command.

Format:
{command} {baud rate}
where:

{command}:	BHT-BAUD
{baud rate}:	9600,19200. Default value for Comtec printers is 9600 BPS.

BHT-BAUD Command Example

The following example sets BHT-IR communications to use a baud rate of 19200 BPS and to strip spaces from received data fields.

```
! DF AUTOEXEC.BAT
! UTILITIES
SRF-ACCESS
BHT-MODE STRIP-SPACES
BHT-BAUD 19200
END-SRF-ACCESS
PRINT
```

BHT MODE Commands

This command sets one of several modes of operation to specify how record data fields delivered by the BHT protocol are interpreted.

Format:
{command} {mode}
where:

{command}: BHT-MODE

{mode}: STRIP-ADD-CRLF
Each data field in a record has any trailing spaces stripped and a CR/LF (carriage return/line feed) pair added. The data field (stripped of trailing spaces and with an appended CR/LF pair) is then passed to the printer for processing. This is the default way to handle data sent to the printer.

RAW
The RAW mode accepts all characters that make up each data field. Using this method, any trailing spaces in a data field will be INCLUDED as data for the printer to process.

COUNTED STRING
This mode is based on the first character of a data field being interpreted as a count of data bytes that follow. In the 'counted string' mode, the

number of bytes specified by the count will be transferred into the printer. The count byte's maximum value is limited to the data field size minus 1.

STRIP-SPACES
This mode eliminates trailing spaces in each data field of a record. Note that unlike the STRIP-ADD-CRLF mode, no CR/LF pair is added to the data after stripping the trailing spaces.

BHT MODE Command Examples

In the following examples a record is defined to be made up of 3 data fields. Fields 1, 2 and 3 are 10, 8 and 20 bytes long, respectively. The "^" character indicates a space character. Other combinations of fields and field lengths can be used as long as they conform with the BHT protocol's data file record field formats.

Example 1: Strip Spaces and Add CR/LF:

```
Input: ADDCRLF.BAT

! DF autoexec.bat
! UTILITIES
SRF-ACCESS
BHT-MODE STRIP-ADD-CRLF
END-SRF-ACCESS
PRINT
```

Output:

Data Record:

```
|----10----|---8----|--------20----------
|
 Comtec^^^^ ^Info^^^ ^Systems^^^^^^^^^^^^
```

Data Sent to printer:

```
|Comtec<CR><LF>^Info<CR><LF>^Systems<CR><LF>|
```

BHT MODE Command Examples (continued)

Example 2: Send Raw Data

```
Input: SENDRAW.BAT

! DF autoexec.bat
! UTILITIES
SRF-ACCESS
BHT-MODE RAW
END-SRF-ACCESS
PRINT
```

Output:

Data Record:

```
|----10----|---8----|--------20----------
|
 Comtec^^^^ ^Info^^^ ^Systems^^^^^^^^^^^
```

Data Sent to printer:

```
 Comtec^^^^ ^Info^^^ ^Systems^^^^^^^^^^^
```

Example 3: Counted String

```
Input: COUNTED.BAT

! DF autoexec.bat
! UTILITIES
SRF-ACCESS
BHT-MODE COUNTED-STRING
END-SRF-ACCESS
PRINT
```

Data Record:

```
|----10----|---8----|--------20----------
|
 6Comtec^^^ 6^Info^^ 7Systems^^^^^^^^^^^
```

Data Sent to printer:

```
 Comtec^Info^Systems
```

BHT MODE Command Examples (continued)

Example4: Strip Spaces

```
Input: SSPACES.BAT

! DF autoexec.bat
! UTILITIES
SRF-ACCESS
BHT-MODE STRIP-SPACES
END-SRF-ACCESS
PRINT
```

Output

Data Record

```
|----10----|---8----|--------20----------
|
 Comtec^^^^ ^Info^^^ ^Systems^^^^^^^^^^^
```

Data Sent to printer

|Comtec^Info^Systems|

BHT PROTOCOL Command

This command placed in an autoexec. bat file overrides the default method of communication.
BHT-PROTOCOL CABLE enables BHT or BHT-IR protocol via cable. If the BHT-PROTOCOL CABLE command is *not* present, data is sent "raw" over the serial cable, if it is detected the BHT-PROTOCOL BHT command sends data in the BHT mode; BHT-PROTOCOL BHTIR sends data in the BHT-IR mode.

Format:	{Command} {mode}
where:	
{Command}:	BHT-PROTOCOL
{mode}:	BHT This mode establishes that data will be sent in BHT protocol mode
	CABLE This mode enables BHT or BHT-IR protocol via cable. If the BHT-PROTOCOL CABLE command is detected the BHT-PROTOCOL BHT command sends data in the BHT mode
	BHTIR Sends data in the BHT-IR mode.

BHT PROTOCOL Command Examples

Example: Cable printing with BHT protocol

Using Cable with BHT protocol

```
Input: PROBHT.CBL

! DF AUTOEXEC.BAT
! UTILITIES
SRF-ACCESS
BHT-PROTOCOL BHT
BHT-PROTOCOL CABLE
BHT-BAUD 19200
BHT-MODE STRIP-ADD-CRLF
END-SRF-ACCESS
PRINT
```

EXAMPLE: IR printing with BHT protocol

Print using IR communications with BHT protocol

```
INPUT: PROBHT.IR

! DF AUTOEXEC.BAT
! UTILITIES
SRF-ACCESS
BHT-BAUD 19200
BHT-PROTOCOL BHT
BHT-MODE STRIP-ADD-CRLF
END-SRF-ACCESS
```

BHT PROTOCOL Command Examples (continued)

Example: Cable printing with BHT-IR protocol

Print using cable BHT-IR protocol

Input: PROBHTIR.CBL

```
! DF AUTOEXEC.BAT
! UTILITIES
SRF-ACCESS
BHT-BAUD 19200
BHT-PROTOCOL BHTIR
BHT-PROTOCOL CABLE
BHT-MODE STRIP-ADD-CRLF
END-SRF-ACCESS
PRINT
```

Example: IR printing with BHT-IR protocol

Print using IR with BHT-IR protocol

Input: PROBHTIR.IR

```
! DF AUTOEXEC.BAT
! UTILITIES
SRF-ACCESS
BHT-BAUD 19200
BHT-PROTOCOL BHTIR
BHT-MODE STRIP-ADD-CRLF
END-SRF-ACCESS
PRINT
```

PRINTER ESCAPE COMMANDS

SET AND READ CODE COMMAND

Printer command sessions normally start with the '!' character. When the printer is used in Generic Text Mode (or Line-Print Mode) and if the user expects to print the '!' character in that mode, then the CCL code must be changed. This is done via the Redefine CCL Code command.

Send the following sequence to redefine the code:

Set CCL Code: ESC (0x1b) '}' (0x7D) 'W' (0x57) '1' (0x31) <new code> (SETCCL.LBL)

Where {new CCL code} is a one-byte character representing the new CCL code.

Once the CCL Code is changed, all CCL sessions should be started with the new CCL code. For example, if CCL code is changed to '~', then instead of issuing a '! UTILITIES' command to the printer, '~ UTILITIES' should be issued.

The printer will retain the new CCL code for as long as it remains powered. If the printer is powered off and back on again, it will revert back to normal operations and will expect '!' as the CCL code.

Send the following sequence to read the CCL code:

Read CCL Code: ESC (0x1b) '}' (0x7D) 'R' (0x52) '1' (0x31) (READCCL.LBL)

After the above command is issued, the printer will return the one-character CCL code.

PRINTER ESCAPE COMMANDS FORMAT

Format:

{escape} {command} [parameters]

where:

{escape}:	The ESC character (0x1b).
{command}:	Choose from the escape commands in this section.
[parameters]:	Parameters for the escape commands.

Note: The escape commands should not be used while in a control/utility session ("! UTILITIES... PRINT" OR "! 0... PRINT").

STATUS/INFORMATION

Get Printer Status ESC (0x1b) 'h' (0x68)

This command requests a status byte from the printer. If one is returned, it indicates that the printer is operational and has finished processing the previous label. It should be called before loading or printing a label in order to make sure that the host software is synchronized with the printer. If bit 3 of the status byte is high, it indicates that the battery is low. If bit 4 is high, it indicates that the printer has been powered on and reset (see the Reset Status function). This command requests a status byte from the printer with the following format:

Bit	Description
4	Printer reset (0=reset cleared, 1=printer reset)
3	Battery status (0=voltage OK, 1=low battery)
2	Latch status (0=latch closed, 1=latch open)
1	Paper status (0=paper present, 1= out of paper)
0	Printer status (0=printer ready, 1=printer busy)

Acknowledge Printer Reset ESC (0x1b) 'N' (0x4e)

This command clears the reset bit that is set on power-up and reported by the 'get printer status' function. The reset information returned by ESC 'h' may be used by the host to perform its printer power-up initializations, such as form-feeding. Once the host completes its initialization of the printer, it may call this function to tell the printer to clear its reset bit.

STATUS/INFORMATION (continued)

Get Printer Information (GPI.LBL)
ESC (0x1b) 'v' (0x76)

This command instructs the printer to return a null-terminated string containing its model number, firmware revision and serial number. In practice, a search for the NUL character should be used, since the length of the string returned may change.

Get Extended Printer Status ESC(0x1b) 'i' (0x69)

This command requests the extended status byte from the printer. Printer program versions 24 and higher respond to this command. The extended status byte returned from the printer is in the following format:

Bit	Description
7	Ribbon status: 0=ribbon detected, 1=no ribbon detected
6	Paper supply status: 0=paper supply is OK 1=paper supply is low
5	Presentation (peeler) sensor: 0=last label removed, 1=last label not removed yet
4	Reserved
3	Reserved
2	Reserved
1	Reserved
0	Reserved

Note: Mask all reserved bits when reading status.

USER LABEL COUNT

Get User Label Count (GULC.LBL)
ESC (0x1b) 'J' (0x4a) 'R' (0x52) 'U' (0x55)

This command requests the current user label count from the printer. The response consists of two bytes, most significant byte first. This count represents the total number of labels printed since the last time the count was reset to zero.

Reset User Label Count (RULC.LBL)
ESC (0x1b) 'J' (0x4a) 'W' (0x57) 'a' (0x61) 'c' (0x63) 'c' (0x63) 'N' (0x4e) 'V' (0x56) 'M' (0x4d) 'U' (0x55)

This command instructs the printer to clear its user label count to zero.

POWER OFF COMMAND

Off Command (OFF.LBL)
ESC (0x1b) 'p' (0x70)

This function instructs the printer to shut off. This function can be used instead of lowering DTR causing the printer to shut down.

PRINTER CONFIGURATION AND SETUP

MINIATURE PRINTER UTILITIES PROGRAM

Included with each Programming Manual is the Comtec Miniature Printer Utilities Disk for DOS and Windows.

The MP Utilities (MPU) Disk allows the user to:

1. Reprogram Printer.
2. Read Printer Status
3. Set Printer Parameters, such as:
   - Baud Rate
   - Inactivity Timeout
   - User Label Count
   - Enable/Disable DTR Power off
   - Enable XON/XOFF Mode
4. Print Test Labels

To run MPU from DOS, execute the program directly from disk or copy files to a sub-directory on your hard drive, then type MENU.

For Windows, type SETUP from Program Manager. A Comtec program group and icon will be created.

Before reprogramming printer, utilize Option 2 - Read Printer Status, to verify communication settings (baud rate, etc.). To reprogram printer for installation of Bar Code/Font/Command files, turn printer on and install media. The Error LED flashes until the file is successfully written into nonvolatile memory. When error LED extinguishes, the printer prints a report of version numbers and available fonts.

MP UTILITIES PROGRAM (continued)

If your printer does not already include MaxiCode or PDF417 , it is necessary to load either of these bar codes into Flash Memory.

TO REPROGRAM PRINTER FLASH

1. Type “Menu” from MPU Program .

2. Select appropriate COM Port and Baud Rate:
   1 = COM1 19,200 3 = COM1 9600
   2 = COM2 19,200 4 = COM2 9600

3. Verify PC to printer communication by selecting option #2 (Read Printer Status).

4. Press Enter to continue.

5. Select Option #1 (Reprogram Printer) from Main Menu.

6. Prompt will ask “Please enter file name:”. Type in the appropriate hex file name. % indicator will display percentage of reprogramming progress.

7. When programming is complete, printer will print out “Reprogrammed successful” message.

8. Press Enter to continue.

9. Turn printer Off then On again.

Printer is now reprogrammed. It is recommended to now select #4 (Print Test Labels) and enter ????.LBL file to test printer.

MP UTILITIES PROGRAM (continued)

To load UPS MaxiCode, use Option 1 (as described above) and enter MPUPSXX.HEX at the prompt for file name. The printer is now programmed with all the standard fonts, bar code symbologies and UPS MaxiCode. Follow the same procedure for PDF417 by sending MPPDFXX.HEX file. The printer is now programmed with all the standard fonts, bar code symbologies and PDF417.

Also included on the Miniature Printer Utilities disk are label files for all the examples in this manual. The file name associated with each example is referenced in the example box. They are identifiable by .LBL extension.

POWER MANAGEMENT

The printer contains the following power management features:

1. Synchronized On/Off control via DTR line from the attached host (DTR On/Off Control).
2. Programmable inactivity timeout.
3. Programmable low-battery timeout.
4. Shutdown command.
5. Enters low-power mode when idle.

SYNCHRONIZED ON/OFF CONTROL VIA PRINTER'S DSR LINE

The DSR line is an input to the printer and is controlled by the attached host's DTR line. The DSR line controls the power on/off and the operation of the short-range radio module.

1. Power On/Off Control

A low (inactive) to high (active) transition on this line will cause the printer to turn ON. A high (active) to (low) transition on this line will cause the printer to turn OFF only if it is configured to do so. The factory default configuration enables power-off on high to low transition on DSR. This setting can be changed using the MP Utilities program.

The printer will stay ON for as long as DSR is high (active) unless it reaches low-battery shut down point, or it receives a command to shut down. Please note that the inactivity time-out is disabled while DSR is high (active).

2. Short-Range Radio Control

The short range radio is enabled when DSR line is low (inactive) or when the host-to-printer cable is disconnected. In this case, the printer will attempt to communicate to the host over the short-range radio if the printer is equipped with a short-range radio. In its active (high) state, the DSR line will disable the short-range radio, if any, and will communicate over the RX/TX line of the serial port. For printers that are not equipped with a short-range radio, the host terminal must still keep the DSR line at an active state.

The printer can be programmed to time-out, and shut itself off, upon reaching a predefined inactivity period. The factory default inactivity time-out is 2 minutes. This setting can be changed using the MP Utilities program or by sending the following command to the printer.

```
! UTILITIES
TIMEOUT n
END
```

Where n is the inactivity timeout in 1/8 of a second. For example, n=960 for a 2 minute inactivity timeout (120 seconds X 8).

A timeout of 0 disables inactivity timeout.

PROGRAMMABLE LOW-BATTERY TIMEOUT

The printer can be programmed to timeout, and shut itself off, upon reaching a predefined period after low battery condition is detected. The factory default low battery timeout is 1 minute. This setting can be changed using

POWER MANAGEMENT (continued)

SHUTTING OFF THE PRINTER REMOTELY

The printer can be turned off by sending the following shut down command:

Off Command ESC (0x1b) 'p' (0x70)

This function can be used instead of lowering DTR causing the printer to shut down.

ENTERING LOW-POWER MODE WHEN IDLE

The printer automatically enters low-power mode when it is idle in order to conserve power. All resident data and images will be preserved while the printer is in low-power mode.

BATCH FILES

The printer flash file system can be used to store a start-up file titled AUTOEXEC.BAT. When the printer is powered on, this file will be searched for and, if present, the printer will execute the commands found in this file. The following example shows how to create an AUTOEXEC.BAT file and load it into the printer.

```
! DF AUTOEXEC.BAT
! UTILITIES
SETLP 5 1 40
PRINT
```

Line one uses the (D)efine (F)ile command to label this file as AUTOEXEC.BAT. The end of an AUTOEXEC.BAT file is indicated by a PRINT command as shown in line four.

When this file is executed, the printer will select font number 5, size 0 as the default front for line printer mode, and the printer will advance 40 dots for every carriage-return (0x0d) received.

Note: Every time a file is created on the printer by using the "! DEFINE-FORMAT...", "! DF..." or the Visual Label Utility application the file information is written to flash memory. Unlike RAM, flash memory does not require battery power for retaining data, and is immune to data corruption due to static discharge. Although flash memory is superior to RAM for safe guarding file contents, it is limited to an average of 10,000 write cycles (i.e. file creations). T
he user should exercise the file creation commands only when needed to ensure this write cycle limit is not exceeded.

RUN.BAT Command/File

The printer will execute the RUN.BAT file on power-up, if one exists. This file may be a format or label file. The only time RUN.BAT file is not executed is after a two-key reset (hold feed key down, turn printer on, release feed key when printer starts printing).

The following example demonstrates the use of the RUN.BAT file for an application that uses a bar code scanner connected to the printer's serial port for data input. The scanner must be set to the same baud rate as the printer, and be configured for 8 data bits, 1 stop bit, no parity. Scanned data must be terminated with both carriage return and line feed (0x0D 0x0A) characters.

The following RUN.BAT file is first sent to the printer's flash file system. When the printer is turned ON this file is found and executed. The keyword RE-RUN is used in the file to instruct the printer to execute this file repeatedly. In this case, the BARCODE command data will be taken from the serial input.

The printer will produce a label each time a bar code is scanned.

RUN.BAT Example

Input (RUN.LBL):

```
! DF RUN.BAT
! 0 200 200 210 1
CENTER
BARCODE-TEXT 7 0 5
BARCODE 128 1 1 50 0 90 \\
RE-RUN
FORM
PRINT
```

Input from scanner (SCANRUN.LBL)

Output:

RE-RUN Command

The RE-RUN command instructs the printer to execute the current file after an end-of-file is encountered. Any format or command file in the printer, with the exception of AUTOEXEC.BAT and CONFIG.SYS, may use the RE-RUN command.

GAP-SENSE & BAR-SENSE Commands

These commands are used to instruct the printer as to which means of top-of-form detection should be employed. Printers default to BAR-SENSE if no command is specified. Printers that are not equipped with a gap-sensor will attempt a pseudo gap-sense.

Format:

{command}

where:

{command}: Choose one of the following:
GAP-SENSE # (0-255)
BAR-SENSE # (0-255)

Gap and Bar Sense commands can be followed by a number to adjust sensitivity. This is useful for gap sense stock from vendors other than Comtec.

GAP-SENSE Command Example:

The following example configures the printer for gap-sensing. In addition, it specifies that the distance from top-of-form to the gap is zero.

```
Input (GAPSENSE.LBL):

! UTILITIES
GAP-SENSE
SET-TOF 0
PRINT
```

Index

Index

Symbols

A

B

G

I

J

L

M

N

O

P

R

S

T

U

V

W

X

Appendices

APPENDIX A-Frequently Asked Questions

These FAQ's are in no particular order. They cover the most commonly encountered questions asked about our printers.

Q. *The first label I print is fine, but the next one starts printing not as close to the top of the label form, then sometimes skips a label, then prints a good label.*

A. Make sure that you only have one carriage return/ line feed pair after the PRINT statement in your label. If there is more than one CR/LF pair, the printer prints the label, then sees the extra CR/LF pairs as data to be printed in line print mode. This advance causes the next label to be registered incorrectly when a new label file is sent.

Q. *I send a label to the printer, and the label begins to print. Before all of the label is printed, there is a feed to the next label. This happens for 2 attempts before the printer stops.*

A. Make sure that the label length specified in your command line is shorter than the distance between your eye sense marks on the label stock. If you are asking to print, say a 400 dot long label and the label stock eye sense marks are 300 dots apart, the printer 'sees' an eye sense mark before the entire label has printed and assumes the label stock was not at top of form. It then feeds to the next top of form and tries again. After 2 tries, the printer advances to the next top of form and stops.

Q. *When I send a label to the printer , the label file gets printed, and not the label itself.*

A. If you are getting the label file printed instead of the label, the syntax of the first line may be incorrect. Check that the syntax is something like: ! 0 200 200 210 1 where the 210 is the label length and 1 is the quantity. If the printer detects an invalid first line, the command mode is not entered. Instead, the printer acts as a generic line printer, and the remaining lines are treated as raw text to be printed.

Q. *My printer doesn't seem to respond when I send anything to it.*

A Make sure that you have the same communications parameters for both the printer and the device connected to it. You can determine the printer's settings by turning the printer off, then while holding the feed key down, turning the printer back on. It should respond by printing a report, pause for about 3 seconds, then print a second report. If you look through these reports, you will find an entry for the current baud rate setting. A good progression for finding problems like this is to use the MPU.EXE utility. Connect the cable, then turn the printer off. Now run the MPU utility. When this utility starts, it should turn the printer on. If the printer will not turn on, check the com port and cable. Now select option 2 (Get printer status) from the menu. If there is no response, you probably have mismatched baud rates.

Q. *There seems to be missing or garbled characters sent to the printer.*

A. Some systems may alter characters before they are sent to the printer. The '\' character for example may

be taken as an escape for the following character. You can place the printer in a 'DUMP' mode to see exactly what characters are being received. In this mode, any character that comes into the printer is printed in both its ASCII form and as a hex value. To place the printer in dump mode, turn the printer off, hold down the FEED key, and turn the printer on. When a report begins to print, release the feed key. After the first report there will be a pause, then a second report. At the end of this report, you will have about 3 seconds to press the feed key to enter Communications Diagnostic (DUMP) Mode. (Refer to pg. P1-4) If you are successful, you will see the message "Dump Mode Entered" on the label along with the Com port settings. You can now send your data to the printer, then examine the resulting label to determine if all characters received are as expected.

Q. *The labels I am producing have fields missing, but the command syntax to produce these fields seems to be correct.*

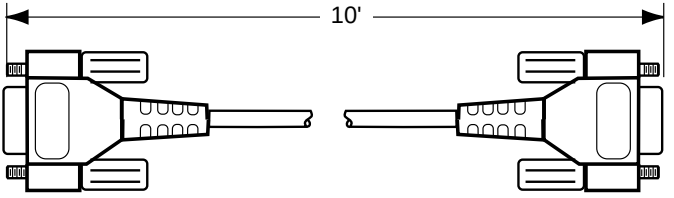
A. Insure that the entire text or barcode fields you are trying to print are positioned within the label borders when printed. Some commands will print the requested field only if it will 'fit' on the label.
Also insure that the requested font and size specified in your label file is resident in the printer. Some applications make use of the flash file system to store custom fonts. If these fonts are missing, the result is a blank field. This also applies for any .PCX image files that may be used in your label files.

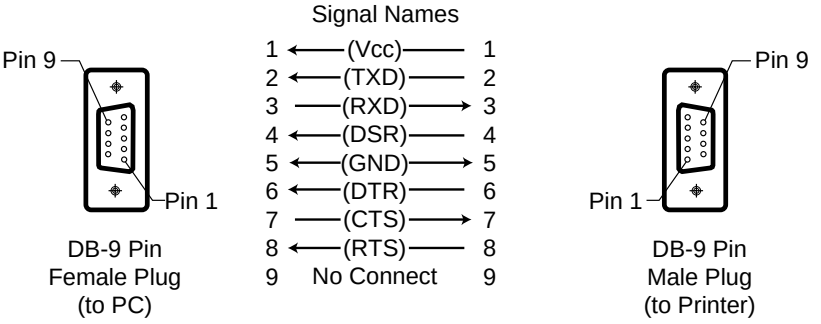
APPENDIX B- Interface Cables

BIDIRECTIONAL INTERFACE CABLE

P/N BL13402-1

Use only with Bravo printers configured with standard DB9 serial I/O connector

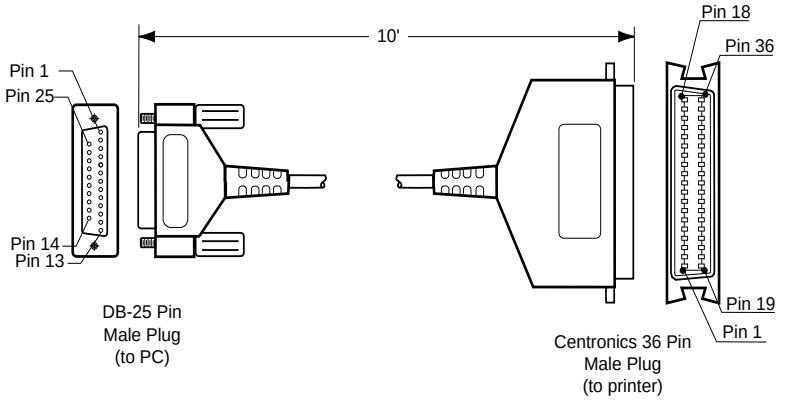




UNIDIRECTIONAL INTERFACE CABLE

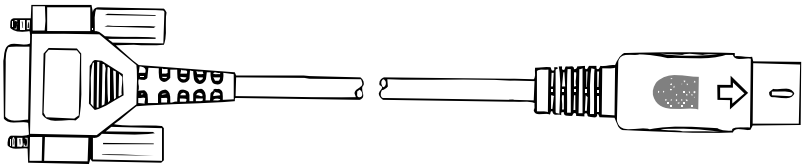
P/N BL13403-1

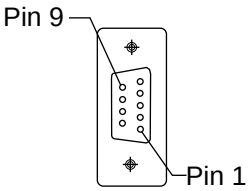
Use only with printers configured with parallel Centronics type I/O connector



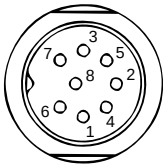
MP SERIES BIDIRECTIONAL INTERFACE CABLES FOR USE WITH A PERSONAL COMPUTER

Comtec Part Number BL11757-000
8-Pin DIN to 9-Pin DB PC Cable





DB-9 Pin
Female Plug



8 Pin
Male Plug

PC Signal Names			Printer Signal Names
(TXD)	3	→ 1	(RXD)
(RXD)	2	← 2	(TXD)
(RTS)	7	→ 3	(CTS)
(CTS)	8	← 4	(RTS)
(GND)	5	— 5	(GND)
(DTR)	4	→ 7	(DSR)
(DSR)	6	← 8	(DTR)
Shield		↔	Shield

MP SERIES BIDIRECTIONAL INTERFACE CABLES
(continued)

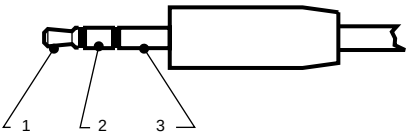
NORAND TERMINALS	Comtec Part Number
RT 1100 and RT1700 Series	BL13309-1, 8-Foot Coiled Cable, 6-Pin Mini DIN to 8-Pin DIN (Supports remote on/ off from terminal)
	BL13298-1 8-Foot Coiled Cable, 6-Pin Mini DIN to 8-Pin DIN Locking, over-molded. (Supports remote on/ off from terminal)
	BL11537-1 8-Foot Coiled Cable, 6-Pin Mini DIN to 8-Pin DIN, over-molded
	BL11537-2, 12-Foot Coiled Cable, 6-Pin Mini DIN to 8-Pin DIN, overmolded
	BL12804-1, 8-Foot Coiled Cable, 6 Pin Mini DIN with Hirose Locking Connector to 8-Pin DIN
RT5900 Series	BL12803-1, 8-Foot Coiled Cable, 15-Pin D Sub to 8-Pin DIN

MP SERIES BIDIRECTIONAL INTERFACE CABLES (continued)

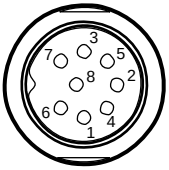
TELXON TERMINALS	**Comtec Part Number**
TELXON 960	BL11122-1, 8-Foot Coiled Cable, RJ45 to 8-Pin DIN
TELXON PTC1140 TELXON 960SL	BL13232-1, 8-Foot Coiled Cable, ITT Cannon 15-Pin Sub Micro D to 8-Pin DIN
TELXON 860 and 912	CL11314-000, 8-Foot Coiled Cable, DB25 to 8-Pin DIN
SYMBOL TERMINALS	**Comtec Part Number**
LDT/LRT 3800 Series	CC11371-5, PIM Cable PIM Optical to 8-Pin DIN
PDT3100 Series	BL12093-1, 8-Foot Coiled Cable, RJ45 to 8 Pin-DIN
PDT3300 Series	BL11391-000, 8-Foot Coiled Cable, DB25 Male to 8-Pin DIN
PPT4100 Series	BL11757-000 or BL11162-1, 6-Foot DB9 Female to 8-Pin DIN *(Note: Must be used with Symbol RS232 Adapter Cable p/n 25-12059-01)*
PPT4600 Series	BL13253-1, 8-Foot Coiled Cable 9 pin Sub Micro D to 8-Pin DIN

DENSO BHT CABLE INTERFACE

BHT- 3 Pole
Mini-Stereo Plug

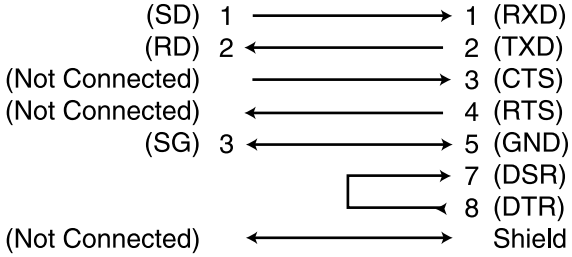


Printer- 8 Pin
Male DIN Plug



BHT Signal Names

Printer Signal Names



APPENDIX C- Character Tables

ASCII TABLE HEX VALUES

Hex	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	sp	0	@	P	`	p								
1	!	1	A	Q	a	q								
2	"	2	B	R	b	r								
3	#	3	C	S	c	s								
4	$	4	D	T	d	t								
5	%	5	E	U	e	u								
6	&	6	F	V	f	v								
7	'	7	G	W	g	w								
8	(	8	H	X	h	x								
9	)	9	I	Y	i	y								
A	*	:	J	Z	j	z								
B	+	;	K	[	k	{		¢						
C	,	<	L	\	l	|								
D	-	=	M	]	m	}								
E	.	>	N	^	n	~								
F	/	?	O	_	o									

INTERNATIONAL ISO SUBSTITUTION CHARACTERS

This table references the ISO symbol set substitutions performed for various countries. The ISO symbol set contains the same characters as the ASCII symbol set except for the character values listed in the table.

	----HEX Character Values---											
---Country---	23	24	40	5b	5c	5d	5e	60	7b	7c	7d	7e
USA	#	$	@	[	\	]	^	`	{	|	}	~
United Kingdom	£	$	@	[	\	]	^	`	{	|	}	-
France	£	$	à	°	ç	§	^	µ	é	ù	è	¨
Germany	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
Italy	£	$	§	°	ç	é	^	ù	à	ò	è	ì
Sweden	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
Spain	£	$	§	¡	Ñ	¿	^	`	°	ñ	ç	~
Norway	#	$	@	Æ	Þ	Å	^	`	æ	ý	å	-

CP-850 CHARACTER SET

Hex	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		0	@	P	`	p	Ç	É	á			ð	Ó	
1	!	1	A	Q	a	q	ü	æ	í			Ð	ß	±
2	"	2	B	R	b	r	é	Æ	ó			Ê	Ô	
3	#	3	C	S	c	s	â	ô	ú	|		Ë	Ò	
4	$	4	D	T	d	t	ä	ö	ñ			È	õ	
5	%	5	E	U	e	u	à	ò	Ñ	Á		ı	Õ	§
6	&	6	F	V	f	v	å	û	ª	Â	ã	Í	µ	
7	'	7	G	W	g	w	ç	ù	º	À	Ã	Î	þ	´
8	(	8	H	X	h	x	ê	ÿ	¿	©		Ï	Þ	°
9	)	9	I	Y	i	y	ë	Ö	®				Ú	¨
A	*	:	J	Z	j	z	è	Ü	¬				Û	·
B	+	;	K	[	k	{	ï	ø					Ù	
C	,	<	L	\	l	|	î	£					ý	
D	-	=	M	]	m	}	ì	Ø	¡	¢			Ý	
E	.	>	N	^	n	~	Ä		«	¥		Ì	¯	
F	/	?	O	_	o		Å		»		¤		´	

LATIN 1 CHARACTER SET

Hex	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		0	@	P	`	p				°	À	Ð	à	ð
1	!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
2	"	2	B	R	b	r			¢		Â	Ò	â	ò
3	#	3	C	S	c	s			£		Ã	Ó	ã	ó
4	$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
5	%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
6	&	6	F	V	f	v			¦		Æ	Ö	æ	ö
7	'	7	G	W	g	w			§	·	Ç	×	ç	÷
8	(	8	H	X	h	x			¨	¸	È	Ø	è	ø
9	)	9	I	Y	i	y			©	¹	É	Ù	é	ù
A	*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
B	+	;	K	[	k	{			«	»	Ë	Û	ë	û
C	,	<	L	\	l	|			¬		Ì	Ü	ì	ü
D	-	=	M	]	m	}					Í	Ý	í	ý
E	.	>	N	^	n	~			®		Î	Þ	î	þ
F	/	?	O	_	o				¯	¿	Ï	ß	ï	ÿ

APPENDIX D - Font Information

FONT HEIGHTS

The following table contains the font heights. The height values are in pixels.

Font # / Font Size --->	0	1	2	3	4	5	6	7
0	9	9	18	18	18	36	36	
1	48							
2	12	24						
4	47	94	45	90	180	270	360	450
5	24	48	46	92				
6	27							
7	24	48						

FIXED-WIDTH FONTS

The following table contains the font widths for the fixed-width fonts. Only one width is given for each font/size combination since every character in that font/size combination has the same width. The proportional-width fonts follow, with a separate table for each. The space character will be substituted for empty values in these tables. The width values are in pixels.

Font # / Font Size --->	0	1	2	3	4	5	6	7
0	8	16	8	16	32	16	32	
1 (see separate table)								
2	20	20						
4 (see separate tables)								
5 (see separate tables)								
6	28							
7	12	12						

PROPORTIONAL WIDTH FONTS

FONT WIDTH IN DOTS-FONT 1, SIZE 0

Hex	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	15	19	22	26	29	15								
1	17	16	26	26	17	15								
2	19	23	26	26	16	11								
3	21	20	20	23	15	16								
4	21	20	25	28	19	12								
5	23	23	22	25	14	18								
6	23	21	20	23	12	16								
7	10	22	18	28	17	23								
8	14	21	23	25	16	16								
9	19	19	16	20	9	18								
A	17	8	21	25	8	17								
B	18	13	24	16	16	13		14						
C	10	19	17	26	11	14								
D	20	21	28	12	26	13								
E	10	18	26	27	17	13								
F	17	19	23	24	15									

FONT WIDTH IN DOTS-FONT 4, SIZES 0-1

Hex	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	12	23	43	28	14	24								
1	13	23	28	32	24	24								
2	15	23	29	30	24	15								
3	23	23	30	27	22	21								
4	23	23	30	26	24	13								
5	37	23	28	29	23	23								
6	28	23	26	27	13	21								
7	8	23	32	39	24	30								
8	14	23	30	27	23	21								
9	14	23	12	28	10	21								
A	17	11	21	25	10	20								
B	25	11	28	12	22	14		23						
C	11	25	23	12	10	12								
D	14	25	35	12	35	14								
E	11	25	31	21	23	25								
F	12	24	32	23	24									

FONT WIDTH IN DOTS-FONT 4, SIZES 2-7

Hex	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	26	51			31									
1	31	51												
2	44	51												
3		51												
4	40	51												
5	82	51												
6		51												
7	22	51												
8	31	51												
9	31	51												
A	36	31												
B	54	31						40						
C	26	54				26								
D	31	54												
E	26	54												
F	26	56												

FONT WIDTH IN DOTS-FONT 5, SIZES 0-1

Hex	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	6	12	23	15	8	14								
1	8	12	18	19	12	14								
2	12	12	16	18	13	11								
3	13	12	18	14	11	10								
4	12	12	18	16	14	8								
5	19	12	17	18	11	14								
6	21	12	15	18	8	12								
7	7	12	19	24	12	18								
8	8	12	19	18	14	12								
9	8	12	10	18	7	12								
A	13	8	12	17	7	11								
B	14	8	19	8	15	10		14						
C	10	14	16	7	7	5								
D	16	14	24	8	21	10								
E	10	14	18	10	14	10								
F	7	12	19	12	13									

FONT WIDTH IN DOTS, FONT 5, SIZES 2-3

Hex	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	12	21	39	23	14	22								
1	14	21	30	30	20	22								
2	17	21	28	28	22	15								
3	21	21	28	23	18	16								
4	21	21	30	25	22	13								
5	35	21	26	30	19	22								
6	33	21	23	30	14	21								
7	10	21	30	40	20	30								
8	14	21	31	30	22	21								
9	14	21	15	30	12	21								
A	21	12	17	26	12	18								
B	24	12	30	14	21	20		21						
C	11	24	26	12	11	8								
D	14	24	37	14	33	20								
E	11	24	30	20	22	22								
F	12	19	30	21	21									

APPENDIX E-BARCODE QUICK REFERENCE

Barcode Symbology	Barcode Type	Input Length	Characters	Ideal Wide/Narrow Ratio	Ideal Narrow Dot Width	Checksum Calculation
UPC-A	UPCA	11 or 12 digits*	0-9 only	2:1	2	mod 10
UPC-A plus 2	UPCA2	13 digits*	0-9 only	2:1	2	mod 10 (UPC-A)
UPC-A plus 5	UPCA5	16 digits*	0-9 only	2:1	2	mod 10 (UPC-A)
UPC-E	UPCE	6, 7 or 11 digits*	0-9 only	2:1	2	mod 10
UPC-E plus 2	UPCE2	8 or 13 digits*	0-9 only	2:1	2	mod 10 (UPC-E)
UPC-E plus 5	UPCE5	11 or 16 digits*	0-9 only	2:1	2	mod 10 (UPC-E)
EAN/JAN-13	EAN13	12 or 13 digits*	0-9 only	2:1	2	mod 10
EAN/JAN-13 plus 2	EAN132	14 digits*	0-9 only	2:1	2	mod 10 (EAN13)
EAN/JAN-13 plus 5	EAN135	17 digits*	0-9 only	2:1	2	mod 10 (EAN13)
EAN/JAN-8	EAN8	6, 7 or 8 digits*	0-9 only	2:1	2	mod 10
EAN/JAN-8 plus 2	EAN82	9 digits*	0-9 only	2:1	2	mod 10 (EAN8)
EAN/JAN-8 plus 5	EAN85	12 digits*	0-9 only	2:1	2	mod 10 (EAN8)
Code 39 (3 of 9)	39	Variable	See notes*	2.5:1	2	none
	39C	Variable	See notes*	2.5:1	2	mod 43
	F39	Variable	See notes*	2.5:1	2	none
	F39C**	Variable	See notes*	2.5:1	2	mod 43
Code 93 (9 of 3)	93	Variable	128 ASCII	1.5:1	1	two mod 47
Interleaved 2 of 5	I2OF5	See notes*	0-9 only	2.5:1	2	See notes*
Interleaved 2 of 5 w/check digt	I2OF5C	See notes*	0-9 only	2.5:1	2	mod 10
Industrial 2 of 5	INDUST 2OF5	See notes*	0-9 only	2.5:1	2	See notes*
Industrial 2 of 5 w/ check digit	INDUST 2OF5C	See notes*	0-9 only	2.5:1	2	mod 10
Code 128 /A/B/C/Auto	128	Variable	See notes*	1.5:1	1	mod 103
UCC-128Std.	UCCEAN 16	See notes*	See notes*	3:1	3	mod 103
Codabar	CODABAR	Variable	0-9,A-D symbol	2.5:1	2	none
	CODABAR 16	Variable	0-9,A-D, symbol	2.5:1	2	mod 16

Barcode Symbology	Barcode Type	Input Length	Characters	Ideal Wide/Narrow Ratio	Ideal Narrow Dot Width	Checksum Calculation
MSI Plessey	MSI	13 digits max	0-9 only	2:1	2	none
	MSI10	13 digits max	0-9 only	2:1	2	mod 10
	MSI1010	13 digits max	0-9 only	2:1	2	two mod 10
	MSI1110	13 digits max	0-9 only	2:1	2	mod 11 mod 10
Postnet	POSTNET	5, 9, 11 digits	0-9 only	3.5:1	3	mod 10
Facing Ident Mark	FIM	A, B, or C only	A, B, or C	1.5:1	6	N/A

* Refer to the discussion of this particular barcode in Section Five for more information.

**To make a HIBCC compliant bar code, use bar code type F39C. See the document "The Health Industry Bar Code (HIBC) Supplier Labeling Standard" from the Health Industry Business Council, 5110 N. 40th Street, Suite 250, Phoenix, Arizona 85018.

MEDIA SUPPLIES

To insure maximum printer life and consistent print quality and performance for your individual application, it is recommended that only media produced by Comtec be used. These advantages include:

- Consistent quality and reliability of media products.
- Large range of stocked and standard formats.
- In-house custom format design service.
- Large production capacity which services the needs of many large and small media consumers including major retail chains world wide.
- Media products that meet or exceed industry standards.

For more information call Comtec Information Systems, Inc. at (401) 739-5800 and ask to speak to a Media Sales Representative.

MAINTENANCE SUPPLIES

In addition to quality media provided by Comtec, it is recommended that the print head be cleaned as prescribed in the User's Manual for individual models. The following items are available for this purpose:

- Print Head Cleaning Pads (10 pack), Reorder No. AN11207-1
  (*For use with all Comtec printers.)*

- Print Head Cleaning Cards (10 pack), Reorder No. AN11208-1
  (Recommended for use with MP50XX, M2, and M4 printers.)

- Cleaning Kit with Alcohol, Cleaning Card, and Cotton Swabs, Reorder No. AT700
  (Recommended for use with MP50XX, M2, and M4 printers.)

- Cleaning Kit with Alcohol, Cleaning Pad, and Cotton Swabs, Reorder No. AT700-2
  (Recommended for use with RP3 printer.)

- Cleaning Kit with Cleaning Pen and and (5) Cleaning Swabs, Reorder No. AT702
  (Recommended for use with Bravo printers)

- Cleaning Pen 10 Pack, Reorder No. AN11209-1
  (Recommended for use with Bravo and Encore printers)

TECHNICAL SUPPORT

Contact the Comtec Product Support Help Desk at:

Comtec Information Systems, Inc.
30 Plan Way
Warwick, Rhode Island 02886-1012
USA
Telephone: (401) 739-5800
Ask for the Help Desk
Fax: (401) 732-0145
E Mail: support@comtecis.com

In Europe:

Comtec Europe B.V.
Nuenenseweg 167
5667 KP Geldrop
The Netherlands
Telephone: +31-40-286 29 80
Fax : +31-40-286 27 36

Ask for Customer Support

Comtec Information Systems, Inc.

30 Plan Way, Warwick, Rhode Island 02886, USA
401/739-5800
800/556-7266
Fax: 401/732-0145
EMail: support@comtecis.com

Comtec Europe B.V.

Nuenenseweg 167, 5667 KP Geldrop,The Netherlands +31 40 286 29 80
Fax: +31 40 286 27 36

Visit Our Website
www.comtecinfosys.com